Developing machine learning and deep learning models for predicting the folding rate of proteins and peptides.

A Project Report

by

REEGINA TYAGI

Submitted

in partial fulfillment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

To

COMPUTATIONAL BIOLOGY

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

21st May 2024

Certificate

This is to certify that the thesis titled "**Developing machine learning and deep learning models for predicting the folding rate of proteins and peptides**" being submitted by **Reegina Tyagi** to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

21st May 2024

Prof. (Dr.) N Arul Murugan

Department of Computational Biology

Indraprastha Institute of Information Technology Delhi

New Delhi 110020

Acknowledgement

My profound thanks goes out to my supervisor, Prof. (Dr.) N Arul Murugan, for all of his help, encouragement, and support during this research effort. This thesis would not have been possible without his knowledge and persistence.

Lastly, I want to express my gratitude to my family for their love and support over this whole journey. Their unfailing faith in me has served as a continual source of inspiration and drive. I also want to express my gratitude to the entire teachers and staff at IIIT Delhi and the Department of Computational Biology for their unwavering support during our time in college. I sincerely appreciate all of your assistance and backing.

Abstract

In order to comprehend the functionality and stability of proteins and peptides, it is essential to forecast their folding rates. This thesis explores the construction of sophisticated machine learning (ML) and deep learning (DL) models for this purpose. The study employs a comprehensive computational methodology that integrates a wide range of bioinformatics instruments to effectively navigate the intricacies of protein folding dynamics.

Using Pfeature, a programme created to extract a wide variety of features from protein sequences and greatly improve the input data quality for machine learning models, is the fundamental step in the feature engineering process. Additionally, to represent protein structures as networks and enable a more in-depth examination of the connections between residues that influence folding kinetics, the study makes use of Graph Signal Processing (GSP) techniques.

Amber23 facilitates molecular dynamics (MD) simulations, which are essential to the study since they model the atomic movements within proteins under varied settings and offer dynamic insights into protein behaviour. Understanding the energetic and structural alterations that take place during the folding process is made possible by this method, which also enriches the dataset with crucial parameters for precise model training.

The thesis uses a range of machine learning models, including advanced regressors, to interpret the intricate datasets that are produced. These models are able to capture the nuanced parameters that control protein folding rates because they are trained on features generated from both sequence data and MD simulation results. The incorporation of many data sources and analytical methods guarantees that the models created not only accurately forecast folding rates but also help to understand theory of protein Biophysics

This work considerably increases the predictive capacities in protein research by fusing data science, machine learning, and computational biology. It provides fresh insights into one of the most intricate biological processes and may find use in genetic and medication design research.

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Artificial intelligence (AI) and machine learning (ML) have seen a tremendous revolution in the last few decades, changing a wide range of application fields, including language translation and picture identification.[1] The ability of algorithms to extract complex patterns and non-trivial relationships from enormous volumes of data has been the driving force behind this revolution, since it has made it possible to abstract this information for the assessment of fresh data.[1]

Simultaneously, these instruments and notions have started to seep into basic scientific investigations, providing fresh perspectives by revealing latent patterns and connections that may inspire the development of novel overarching principles. The introduction of machine learning techniques has led to substantial breakthroughs in the fields of protein folding and dynamics.[1]

All the information required to ascertain the folded structures of proteins is contained in their sequences. Consequently, it makes sense to investigate if the concepts and methods that have worked well for labeling images may also be used to help associate a folded form with a specific protein sequence. The use of machine learning techniques has, in fact, given protein structure prediction a boost, as shown by the several teams' successes in the Critical Assessment of Structure Prediction (CASP) challenges.

Other concerns of protein dynamics have also demonstrated potential for being addressed by machine learning techniques. Energy functions are used in traditional physics-based approaches to protein folding to direct the dynamics of the protein from its unfolded to folded state. Although these methods have proven effective, they frequently have difficulty adequately simulating intricate multi-body interactions. The creation of energy functions that may include these complex interactions which are difficult to represent analytically is made possible by neural networks, providing a solution.

Additionally, there have been notable advancements in machine learning for the study of protein simulations. It is still difficult to extract relevant information and relate it to experimental observations, even with precise force fields [2], [3], [4] and long enough simulations to sample equilibrium distributions. Strong tools for deriving valuable insights from simulation data and bridging the gap between computational and experimental findings are offered by machine learning approaches.

All things considered, there is great promise for improving our comprehension of these intricate biological processes through the application of machine learning to the study of protein folding and dynamics. Researchers can uncover new information on the 102 relationships between protein structure and function and open the door to advancements in drug discovery, protein engineering, and biomedical research by utilizing AI and ML techniques.[5]

Understanding a protein's "functional native structure" is essential for understanding its biological roles and for forming associations with other molecules or proteins to perform structural and regulatory functions. Only a small portion of research has been able to unravel the latter portion of the genetic code, making the complexities of protein folding one of the most challenging enigmas. Folding occurs in an orderly fashion to produce a stable native structure. If a protein domain were to randomly navigate through all possible interactions in order to reach its natural state, it would take longer than the universe to expand, however most protein domains do so in a matter of $10^{(-6)}$ to $10^{(-1)}$ seconds. [5]

Hundreds of separate bonds constitute the covalent backbone of a typical protein, enabling free rotation and a multitude of conformations. Nonetheless, each protein serves a distinct chemical or structural purpose, underpinning a specific three-dimensional structure. A protein's conformation, whether in whole or in part, delineates the spatial arrangement of its atoms and encompasses any structural state achievable without bond rupture. Conformational changes, such as rotation about single bonds, contribute to the diverse states of proteins. Despite the myriad potential conformations for a protein, only one or a few typically prevail under biological conditions. [6]

Stability refers to a protein's tendency to maintain its inherent structure, arising from intricate interactions among various components. Native proteins typically exhibit a modestly stable Gibbs free energy (ΔG) gap between their unfolded and folded states, ranging from 20 to 65 kJ/mol. The unfolded state, characterized by a multitude of potential conformations, possesses high conformational entropy. Nonetheless, stabilizing forces favoring the native conformation are generated by chemical interactions. These forces encompass weak interactions such as disulfide bonds, hydrophobic and ionic interactions, as well as hydrogen bonds.[6], [7]

Protein folding relies on weak interactions, which facilitate the formation of secondary and tertiary structures.[5] Additionally, these weak interactions play a crucial role in the assembly of multiple polypeptides to form quaternary structures. Despite the strength of individual covalent bonds, weak interactions, abundant in protein structure, emerge as the primary stabilizing factor. Protein stability transcends the mere sum of weak interactions; for instance, hydrogen bonds enhance folding in aqueous solutions by augmenting entropy. Alongside van der Waals interactions, hydrophobic interactions, and salt bridges, these forces contribute to stabilizing proteins, defining a delicate equilibrium that shapes protein architecture. [7]

Hydrophobic interactions exert a significant influence on protein structure and dynamics, playing a crucial role in protein stability and folding. In aqueous solutions, the unique hydrogen-bonding network of water molecules facilitates the formation of highly structured solvation layers around hydrophobic molecules. This structured arrangement of water molecules represents an unfavorable thermodynamic state, associated with a reduction in entropy. Conversely, the clustering of nonpolar groups, such as hydrophobic amino acid side chains, leads to a decrease in the extent of the solvation layer, thereby favorably increasing entropy. This entropic effect is one of the primary driving forces behind the interaction of hydrophobic groups in aqueous solutions.

Hydrophobic side chains of amino acids exhibit a propensity to cluster together within proteins, forming a hydrophobic core that remains shielded from the surrounding water molecules. Common hydrophobic residues, such as leucine (Leu), isoleucine (Ile), valine (Val), phenylalanine (Phe), and tryptophan (Trp), are strategically positioned in amino acid sequences to facilitate this clustering during protein folding. The entropic consequences associated with burying hydrophobic surfaces represent a significant driving force behind the formation of hydrogen bonds in proteins under physiological conditions.[7]

The interaction of groups with opposing charges can have a profound impact on protein structures, either stabilizing or destabilizing them through the formation of ion pairs or salt bridges. Similar to hydrogen bonding, the unfolded state of charged amino acid side chains interacts with water and salts, necessitating consideration of the interactions lost during protein folding. Salt bridges significantly contribute to protein stabilization, particularly in environments with lower dielectric constants, such as the nonpolar interior of proteins.[8]

Furthermore, ionic interactions confer distinctiveness to protein structures by limiting structural flexibility and influencing the overall stability of folded proteins. Van der Waals interactions play a crucial role in the densely packed atomic environment of proteins. Despite their weakness, van der Waals interactions are instrumental in maintaining protein stability, especially in tightly packed protein structures or interactions with other molecules.[7], [8]

Through an examination of the correlation between changes in stability resulting from physicochemical, energetic, and structural aspects of amino acid residues, the researchers identified parameters influencing protein stability.[8]

Covalent bonds within the peptide backbone impose significant constraints on polypeptide conformation. The conformation of peptides is characterized by three dihedral angles—phi, psi, and omega—that represent rotation about each of the three repeating bonds in the peptide backbone. These angles describe the local spatial arrangement of main-chain atoms, leading to the formation of regular secondary structures such as helices, sheets, and turns.[8]

Peptide bonds, crucial for protein assembly, possess unique properties elucidated by Linus Pauling and Robert Corey in the late 1930s. They consist of an inflexible atomic arrangement, creating a partial double bond that restricts free rotation. Comprising six co-planarly organized atoms, the peptide bond features the carbonyl group's oxygen atom positioned opposite to the amide nitrogen's hydrogen atom, resulting in a small electric dipole.[6]

X-ray diffraction analyses have demonstrated that the C—N bond length in a peptide bond is shorter compared to a typical amine, implying the presence of resonance or partial sharing of electron pairs between the carbonyl oxygen and the amide nitrogen. This characteristic bond is represented as Cα−C=O−N−Cα. The polypeptide backbone is characterized by a series of rigid planes resulting from the ability to rotate around the N—Cα and Cα—C bonds, while rotation is restricted at the peptide C—N bonds due to their partial double-bond nature.

The conformation of the backbone is characterized by three dihedral angles (φ, ψ, and ω), representing rotation about the repeating bonds in the peptide backbone. These angles describe the orientation of successive planes formed by bond vectors in the peptide backbone. Secondary structure disregards side chains or interactions with other segments and focuses on the local spatial arrangement of main-chain atoms.[9]

α-helices and β-sheets represent examples of regular secondary structures, characterized by constant φ and ψ angles throughout a segment. Conversely, irregular segments may manifest as turns or randomly arranged coils. The rigidity of peptide bonds constrains the range of conformations available to polypeptide chains, exerting a fundamental influence on protein structure and function.

Proteins adopting the α-helix structure feature a tightly wound polypeptide backbone encircling an imaginary axis along the helix's length, with amino acid side chains extending outward. The turns of the helix are spaced approximately 5.4 Å apart along its longitudinal axis, akin to the periodicity observed by Astbury in the X-ray analysis of hair keratin. Typically, an α-helix comprises 3.6 amino acid residues in a single turn, with its conformation defined by distinct dihedral angles.[9]

Proteins commonly exhibit a right-handed α-helix configuration, although left-handed variations are less prevalent and theoretically less stable. α-helical segments may exhibit slight deviations from ideal dihedral angles. Notably, experimental systems indicate a significant propensity for helix formation in alanine residues.[8]

The relative positioning of amino acid residues is crucial, as interactions between side chains influence the stability of the α-helical structure.[8] For instance, negatively charged carboxyl groups on consecutive Glu residues repel each other, hindering α-helix formation. Similarly, at neutral pH, neighboring Lys and/or Arg residues with positively charged side chains oppose one

another, impeding α-helix formation. Additionally, residues such as Cys, Thr, Ser, and Asn, when densely packed, may introduce instability to α-helix assembly due to their bulkiness or structural features. [8], [9]

Presence of Proline in α-helices is uncommon due to its unique structural characteristics, particularly its rigid cyclic side chain, which disrupts the regular α-helical structure. Conversely, glycine, with its smaller side chain and greater conformational flexibility, is more commonly found in α-helices. Glycine's flexibility allows it to adopt a wide range of conformations, making it suitable for α-helical structures.

In contrast to proline, glycine residues are conducive to α-helix formation due to their ability to accommodate the tight packing characteristic of α-helices. Consequently, polypeptide chains with glycine residues are more likely to adopt α-helical configurations. These structural preferences play a crucial role in determining a polypeptide chain's propensity to form α-helices, which in turn influences protein folding and functionality.[9]

The β-conformation, as elucidated by Pauling and Corey in 1951, represents an elongated form of polypeptide chains characterized by a zigzag configuration instead of a helical one. This arrangement gives rise to β-sheets, consisting of multiple segments aligned side by side in the β-conformation. The sheet exhibits a pleated appearance due to hydrogen bonds forming between adjacent segments, facilitated by the side chains of protruding amino acids orienting in diverse directions.

Within a β-sheet, adjacent polypeptide chains may display slight discrepancies in hydrogen-bonding patterns and repeat lengths, showcasing either parallel or antiparallel orientations. In antiparallel β-sheets, interstrand hydrogen bonds align linearly, while they may exhibit a twisted configuration in parallel β-sheets. Analogous to α-helices, idealized structures demonstrate characteristic bond angles, although real proteins often exhibit variations that contribute to structural diversity.

Turns and loops within globular proteins facilitate the reversal of the polypeptide chain's orientation, linking consecutive segments of secondary structures such as α-helices or β-conformations. Among these, the β-turn is a frequently encountered turn comprising four amino acid residues, executing a 180° turn. Different types of β-turns are distinguished by unique bond angles, and due to their distinctive structural features, glycine and proline residues are commonly found in β-turns.[9]

Characteristic dihedral angles (φ and ψ) delineate secondary structures like α-helices and β-conformations, with most residues falling within anticipated regions in Ramachandran plots. Glycine, owing to its abbreviated side chain, can assume conformations beyond conventional protein domains, thereby augmenting the structural diversity of proteins. These secondary

structures profoundly influence the overall architecture and stability of proteins and play pivotal roles in protein folding and functionality.

A protein's complete three-dimensional arrangement, encompassing interactions between amino acids that may be distant in the polypeptide sequence and capable of assuming distinct secondary structures, is termed its tertiary structure. Certain residues within the polypeptide chain, such as Pro, Thr, Ser, and Gly, influence the positioning and angle of bends, including turns.

The typical spatial arrangements of interacting segments within the folded protein structure are stabilized by weak interactions and occasionally by covalent bonds like disulfide cross-links. Quaternary structure pertains to the organization of multiple subunits in proteins, which may be identical or different, within three-dimensional complexes.[10]

Fibrous and globular proteins constitute the two primary categories of proteins. Fibrous proteins consist of long strands or sheets of polypeptide chains, providing structural strength and flexibility. These proteins typically exhibit a simple tertiary structure predominantly composed of a single type of secondary structure.

Conversely, globular proteins feature polypeptide chains folded into a spherical or globular shape, often incorporating a diverse array of secondary structures. Typically, globular proteins play roles in enzymatic and regulatory processes, while fibrous proteins are primarily responsible for structural functions such as support, shape, and protection.[11], [12]

Collagen, silk fibroin, and α-keratin represent notable examples of fibrous proteins. Sharing common characteristics, these proteins contribute to the flexibility and strength of biological structures. Their insolubility in water stems from a significant concentration of hydrophobic amino acid residues, both internally and on their surface.[13]

Typically, these hydrophobic regions remain concealed when multiple polypeptide chains assemble to form complex structures. Fibrous proteins, owing to their structural simplicity, serve as valuable models for elucidating fundamental principles of protein structure.

A recognizable folding pattern composed of two or more secondary structure elements and their connecting regions is termed a "motif," also known as a fold or supersecondary structure. Motif diversity ranges from simple configurations like the β-α-β loop to more intricate structures such as the β-barrel. While "fold" typically denotes more complex patterns, the terms "motif" and "fold" are sometimes used interchangeably to describe them, whether they represent a small section or the entirety of a protein.[8]

Motifs are folding patterns rather than hierarchical systems, and their stability can vary. Examples include the globin fold observed in myoglobin and other globins, as well as the coil in

α-keratin. These motifs contribute to the overall tertiary structure of proteins and may not always function as independent entities.[13]

"Domain," a term introduced by Jane Richardson in 1981, denotes a segment of a polypeptide chain capable of independent movement within a protein or maintaining independent stability. Proteins often fold into multiple domains, each potentially serving a distinct function. Even when isolated from the rest of the protein, domains can retain their original structure, highlighting their stability and autonomy.[8], [14]

Protein structures rely on hydrophobic interactions, often necessitating the presence of at least two layers of secondary structure. In a folded arrangement, contiguous segments of an amino acid sequence typically aggregate, with helices and β-sheets occupying distinct structural planes. The orientation of β-sheets and polypeptide linkages is influenced by the right-handed twist of individual segments, impacting the stability of the β-conformation.

By adhering to specific principles, complex motifs can be constructed from simpler ones. An example of this is the β/α barrel motif, comprising several β-α-β loops organized to create a stable structure with right-handed connections. These motifs can be categorized into hierarchical levels of classification based on their folding configurations and evolutionary relationships, supporting the tertiary structure of proteins.[8]

The Structural Classification of Proteins (SCOP) database categorizes protein structures into four groups (all α, all β, α/β, and α+β), each comprising distinct folding configurations constructed from recognizable substructures. Proteins sharing similar tertiary structures and functions are grouped into families or superfamilies, providing insights into their evolutionary relationships and functional roles.

Quaternary structure refers to the arrangement of multiple polypeptide subunits, ranging from two to hundreds, within a protein. These subunits associate to perform various functions, including catalysis, structural support, regulatory roles, and complex multistep processes. Conformational changes in regulatory proteins induced by the binding of small molecules can impact their activity.[10]

While some multisubunit proteins consist of identical subunits or repeating groups arranged symmetrically, others exhibit asymmetric structures due to nonidentical subunits. The repeating structural unit of a multisubunit protein is referred to as a protomer, often designated by Greek letters such as α, β, γ, and so on. For instance, a multimeric protein may consist of subunits labeled α, β, γ, and more.[14]

One notable example of an oligomeric protein is hemoglobin, which comprises four polypeptide chains and four heme prosthetic groups. Each hemoglobin molecule consists of two α chains and

two β chains, arranged symmetrically in pairs. Despite the use of Greek letters to designate subunits, it's important to note that these designations may not always correspond directly to specific secondary structures.

Max Perutz, John Kendrew, and their colleagues successfully elucidated the three-dimensional structure of hemoglobin in 1959, revealing its tetrameric arrangement consisting of symmetric pairs of α and β subunits.[8]

The intricate process of protein folding plays a vital role in restoring unfolded proteins to their original, functional structures. This process is governed by both kinetics and thermodynamics. Two crucial aspects of protein folding are kinetic order and rate constant. Depending on kinetic order, the folding process may proceed through one or more intermediates, and the rate constant can vary significantly based on the protein's length, ranging from nanoseconds to hours. Generally, smaller proteins tend to fold more rapidly than larger ones.[11], [13]

While larger proteins containing over 100 amino acids often exhibit non-two-state kinetics with stable intermediates, smaller proteins typically demonstrate simple two-state kinetics. However, these folding dynamics can be influenced by alterations in the experimental conditions. Proteins that misfold or aggregate due to incorrect folding are implicated in various neurodegenerative diseases.[15], [16], [17]

Proteins are synthesized rapidly in living cells; for instance, at 37°C, E. coli cells can produce an entire protein molecule comprising 100 amino acid residues in approximately 5 seconds. However, protein synthesis alone is insufficient; proper protein folding is essential.

During the folding process, the polypeptide chain explores multiple conformations before attaining its native state. If each amino acid residue could adopt an average of 10 different conformations, there would be approximately 10^{100} alternative conformations for the polypeptide. It would require approximately 10^{77} years to explore every configuration, even at the fastest molecular vibration rate, a timescale far exceeding any realistic time frame. Levinthal's paradox underscores the need for more efficient folding mechanisms.[18], [19]

The folding process of a polypeptide chain is intricate yet hierarchical. Local secondary structures such as sheets and helices emerge first under certain constraints. Ionic and hydrophobic interactions play a significant role in the early folding steps and intermediate stabilization. Stable folded structures then form through longer-range interactions among various structural elements. Proteins with more complex folding patterns, involving numerous long-range interactions, typically fold more slowly than those with simpler folding patterns and close-range interactions. [18]

Protein folding is a highly coordinated process driven by both long- and short-range interactions, leading to the formation of complete domains and the folded protein structure. It's important to note that a protein's thermodynamic stability is not uniform throughout; some regions may exhibit relatively high stability, while others may have little or low stability. In regions of low stability, a protein may be capable of adopting several different conformational states.[18]

Protein folding, viewed through a thermodynamic lens, involves a transition from high free energy and structural entropy in unfolded proteins to a more constrained state as they approach their native structure. This process is depicted as a narrowing free-energy funnel, with minor deviations representing semi-stable intermediates. Ultimately, folding leads to convergence on one or a few native conformations.[18] The complexity of folding pathways, presence of intermediates, and potential for misfolded aggregates can impact the shape of this funnel. Some proteins require assistance from molecular chaperones for the spontaneous folding of proteins to occur. Chaperones interact with misfolded or partially folded polypeptides, aiding in the formation of proper folding pathways or creating favorable folding microenvironments. [20]

The two main families of chaperones are the chaperonins and the Hsp70 family. Hsp70 proteins aid unfolded polypeptides in folding by binding to hydrophobic residue-rich regions, preventing denaturation. Chaperonins, like GroEL/GroES in bacteria and Hsp60 in eukaryotes, assist non-spontaneously folding proteins by creating a regulated folding environment within a chamber. [20]

Additionally, certain proteins require enzymes catalyzing isomerization events; for instance, peptide prolyl cis-trans isomerase (PPI) converts proline peptide bond isomers, while protein disulfide isomerase (PDI) shifts disulfide bonds until the native conformation is achieved.

Protein misfolding contributes to various diseases, including amyloidoses such as Parkinson's, Alzheimer's, type 2 diabetes, and Huntington's disorders. Amyloidoses involve soluble proteins adopting a misfolded state, forming insoluble extracellular amyloid fibers. These fibers, characterized by a high β-sheet structure, are organized and unbranched, with the stability of their β-sheet core largely reliant on aromatic amino acid residues. Mutations that promote amyloid fibril formation can accelerate the onset of symptoms in these diseases.[17], [18], [20]

The intricate process of protein folding is essential for proteins to function in living organisms. Transitioning from an unfolded protein domain to its native, compact three-dimensional (3D) structure is necessary for a protein to carry out its biological functions. Understanding the variables influencing protein folding rates has always been a focal point of research in molecular biology and bioinformatics.[18]

The pursuit of deciphering the fundamental principles underlying protein folding has popularized the field of predicting protein-folding rates based on linear chain sequences. Research has

consistently revealed a strong association between the native structure of proteins and their folding rates, emphasizing the importance of topological packing in comprehending protein structure and folding dynamics. Consequently, investigating the relationship between protein folding rate and various structural variables has garnered significant attention. [21], [22]

Analyzing protein 3D structures to identify novel structural factors corresponding to folding rates represents a promising approach to exploring this relationship. In an attempt to elucidate the underlying principles governing protein folding kinetics, scientists have proposed various structural characteristics based on transition state (TS) and non-two-state (NTS) proteins. Understanding the intricate mechanics of protein folding necessitates comprehension of the kinetics and rate constants involved. Experimentally determining these characteristics is often laborious and time-consuming. [21] Consequently, bioinformatics tools have emerged as valuable alternatives. Notably, there is a relationship between folding kinetics and protein solubility, as the former determines whether a protein folds into its native, soluble state or forms insoluble inclusion bodies.

Over the past decade, numerous bioinformatics algorithms have been developed to predict the solubility or folding kinetics of proteins. These tools leverage data on the three-dimensional structure of proteins to generate predictions. Their efficacy in elucidating protein folding dynamics is evidenced by the strong correlation observed between their predictions and experimentally determined folding rates.[23]

Structural factors such as relative contact order (RCO), absolute contact order (ACO), chain length (Nres), size-modified contact order (SMCO), and native state geometry have shown promise in encapsulating the nuances of protein folding kinetics. These metrics provide insights into the folding process by revealing details about the topology and compactness of protein structures. However, it's worth noting that the majority of structural data stem from a limited set of TS and NTS proteins, potentially limiting their ability to predict folding rates for other protein types.[23], [24]

Due to this limitation, researchers are exploring machine learning (ML)-based approaches to predict folding rates with less data. While ML methods have shown promise, the underlying mechanisms of folding remain elusive.

Despite the challenges, ongoing research efforts are enhancing our understanding of the relationship between folding rate and protein structure. By integrating computational modeling, structural biology techniques, and experimental data, scientists are gaining new insights into the complex dynamics of protein folding. [23]

In computational and molecular biology, predicting protein folding rates from amino acid sequences is a significant challenge. Numerous studies have been conducted over the years to

comprehend and forecast these rates, often leveraging data from three-dimensional protein structures.

To understand the core concepts governing protein folding kinetics, research has explored various theories and models. One such concept is contact order, which assesses the proximity of amino acid residues in the protein's original structure. It has been observed that this parameter correlates with folding rates, providing insights into the folding process.[25]

Additionally, scientists have investigated the fundamental physical and chemical mechanisms underlying protein folding to deepen their understanding of the process. Folding kinetics have also been examined concerning long-range order, which describes the spatial organization of residues over significant distances in the protein structure. Similarly, simple statistical models have been developed to predict folding rates effectively while capturing the stochastic nature of protein folding.[26], [27]

Another approach to predict folding rates integrates stability considerations with contact order. Researchers have developed comprehensive models for predicting folding kinetics by incorporating data on topological features of protein conformations, total contact distance, and the number of native contacts.[22], [23], [25]

Recent advancements have expanded the repertoire of prediction techniques to include information on amino acid sequence, secondary structure, and structural class.[9] These methods enhance the accuracy of folding rate predictions by leveraging machine learning algorithms and statistical analysis to extract relevant information from protein sequences. The intricate network of interactions among amino acid residues, influenced by their physical, chemical, energetic, and structural properties, constitutes the fundamental aspect of protein folding. These characteristics are crucial for understanding the stability and folding kinetics of proteins and provide valuable insights into the relationships between their structure and function.[23]

Furthermore, researchers have extensively studied the structure of proteins in their transition states and the stability of proteins after mutations using amino acid characteristics. By integrating these features with advanced computational methods, significant progress has been made in estimating protein folding speeds and unraveling the intricate dynamics of protein folding.[27], [28]

Predictive techniques often estimate the natural logarithm of a protein's folding rate, commonly referred to as the log folding rate.[22]The correlation between the predicted and actual log folding rates serves as a key metric for evaluating these techniques. This evaluation measure, known as the "correlation of the model," assesses the performance of the model when applied to the training dataset. It has been widely acknowledged that the length of a protein significantly influences its folding speed, with models considering only protein length showing correlations as

high as 0.70. By incorporating the protein's architecture, the correlation can be further improved.[20]

With the continuous influx of experimental data, predicting protein folding rates has become more accessible through the development of statistical and machine learning techniques. However, recent evaluations of several models using current experimental data revealed much poorer prediction performance. This discrepancy was attributed to overfitting, a phenomenon where models perform well on training data but poorly on new data. Consequently, claims of strong correlations should be approached with caution, and future research should rigorously assess and guard against overfitting, potentially using techniques like learning curves.

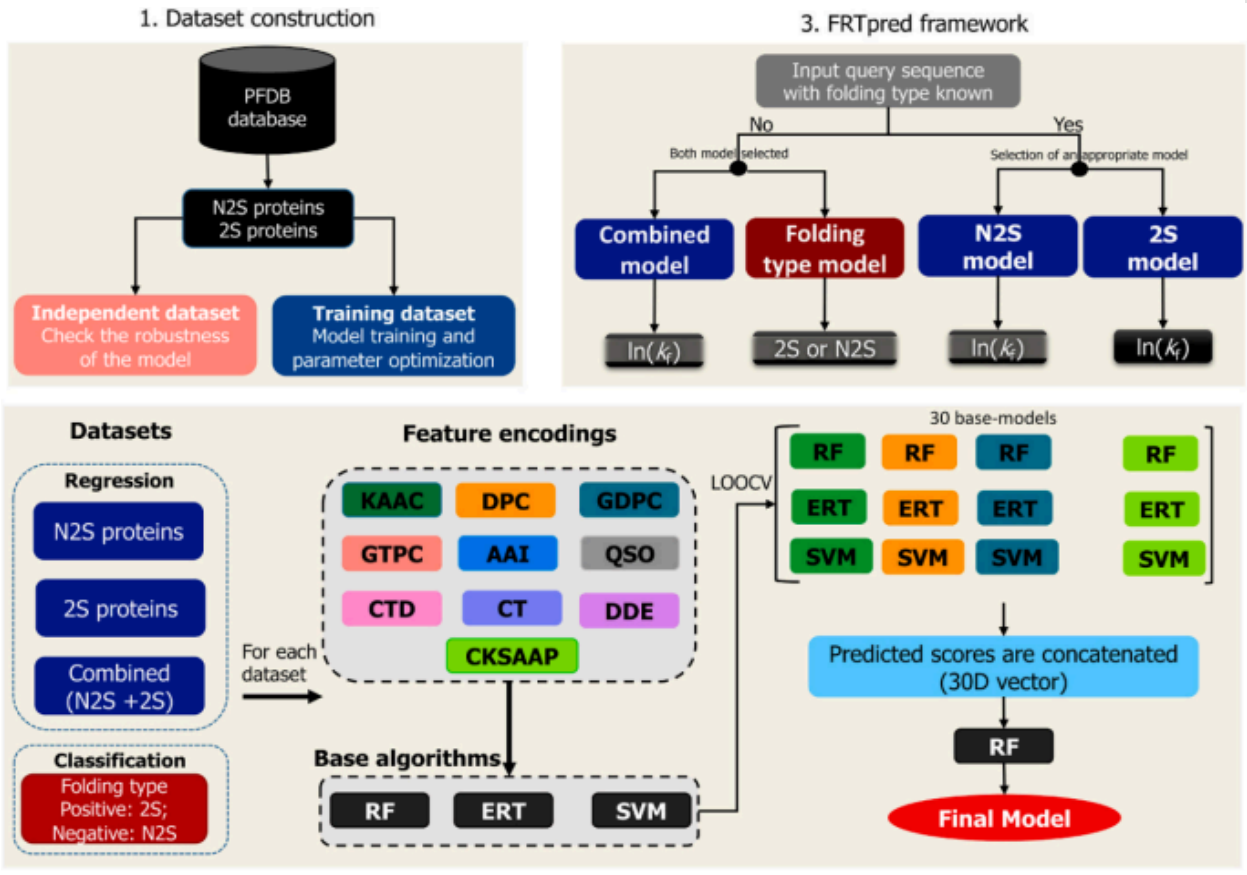


Fig. 1. An overview of the three steps in the FRTPred prediction framework are as follows: (i) creating a dataset using the PFDB database; (ii) creating a model based on a stacking framework by creating 30 baseline models with three different ML algorithms and ten different encodings; these models' predicted values were then fed into RF to create a meta-predictor; and (iii) assembling four developed models into the FRTpred framework.

SfoldRate[29], FOLD-RATE[28], [30], PredPFR, FoldRate[30], K-Fold, PRORATE, SWFoldRate[29], and SeqRate[31] are the instruments that were assessed in this analysis. The

fundamental techniques used by the protein-folding rate prediction tools are both statistical and machine learning algorithms. Simple visual aids for determining the correlation between input data and output predictions include statistical techniques like logistic regression and linear regression. Multiple linear regression is used by programs such as SFoldRate[29] and FOLD-RATE[30], which allow different input factors to be taken into account and given varying weights to indicate how important they are in terms of protein-folding rate prediction.

Machine learning techniques, such as decision trees and neural networks, provide automated information search and processing skills, enhancing prediction accuracy. PRORATE and K-Fold utilize support vector machine (SVM) algorithms, specifically support vector regression, to handle non-vector biological data like protein and nucleotide sequences.

In complex biological systems, ensemble predictors, like Pred-PFR and FoldRate, integrate multiple individual predictors to achieve improved performance. These ensemble predictors combine separate predictors based on various statistical models, leveraging the strengths of each component predictor.[32], [33]

SWFoldRate[29] and SeqRate[31] are prediction tools that combine both statistical and machine learning methods, offering a robust approach to protein-folding rate prediction. By integrating the strengths of both techniques, these tools aim to mitigate the limitations of each strategy, resulting in more robust and reliable prediction capabilities.[34]

During the intricate process of protein folding, a sequence of amino acids explores a wide range of conformations in its unfolded state before adopting a single, native three-dimensional (3D) [8] structure. To understand this complex process, researchers have delved into various theoretical frameworks, analyzed numerous structural variables, and examined their relationship to the natural logarithm of the protein folding rate (ln(kf)).[34], [35] However, these structural characteristics are often limited to a subset of proteins and may not reliably estimate ln(kf) for two-state (TS) or non-two-state (NTS) proteins.

In an attempt to address these limitations, many machine learning (ML)-based models have been developed,[36] but they have not consistently provided accurate predictions of folding processes. In our work, we have explored 48 unique amino acid characteristics within this domain.[37], [38], [39]

Table 1: A comparison of the protein-folding rate prediction tools reviewed

Tool[b]	SFoldRate	FOLD-RATE	Pred-PFR	FoldRate	K-Fold	PRORATE	SWFoldRate	SeqRate
Correlation coefficient	0.82	0.87[c]	0.88	0.88	0.74	0.85[d]; 0.88[e]	0.93	0.81[d];0.80[e]
Statistical deviation[f]	n/a	$x \pm 1.19^{g}$ (MAD); $x \pm 0.23^{h}$ (MAD)	$x \pm 2.03$	$x \pm 2.03$	$x \pm 1.2$ (SE); $x \pm 0.75^{i}$ (MAD)	$x \pm 1.95^{d,e}$; $x \pm 2.12^{e,d}$; $x \pm 1.34^{d,h}$; $x \pm 1.77^{e,h}$	$x \pm 2.27$ (SE)	$x \pm 0.79^{d}$ (MAD); $x \pm 0.68^{e}$ (MAD)
Application type	WB	WB	WB	WB	WB	WB	WB	WB, SA
Development method	Statistical method with multiple linear regressions	Statistical method with multiple linear regressions	Ensemble predictor with multiple statistical models	Ensemble predictor with multiple statistical models	SVM classifier with linear kernel	SVR with polynomial kernel	Non-linear SVM regression model with sliding window	Non-linear SVM classifier with radial basis Gaussian kernel and regression model
Performance Evaluation strategy	Leave-one-out cross-validation	Leave-one-out cross-validation and back-check prediction	Leave-one-out cross-validation	Leave-one-out cross-validation	Cross-validation	Leave-one-out cross-validation and back-check prediction	Leave-one-out cross-validation	Independent test
Experimental verification	No	No	No	No	No	No	No	Yes
Size of training data set (number of proteins)	80	77	80	80	63	80	79	54
Size of test data set (number of proteins)	n/a	n/a	n/a	n/a	n/a	n/a	n/a	7
Input format	FASTA	Plain sequence	Plain sequence	Plain sequence	PDB code/file	PDB file	FASTA	Plain sequence
Additional input	No	Structural class of protein	No	No	No	Protein-folding kinetic state	No	Protein-folding kinetic state
Outputs	Folding rate in natural logarithm	Residues composition, protein structural class and folding rate in natural logarithm	Folding rate in natural logarithm	Folding rate in natural logarithm, half-folding time	Contact order, reliability index, protein-folding kinetic state and folding rate in logarithm based 10	Topology parameters, network parameters, protein-folding kinetic state and folding rate in natural logarithm	Folding rate in natural logarithm	Protein-folding kinetic state, contact number, contact order and folding rate in logarithm based 10 and natural logarithm

The study here involved the utilization of five distinct datasets for the sequence data of proteins as 1D descriptor, each contributing unique information essential to the objectives of the study. One dataset contained a compilation of 140 protein sequences, coupled with their corresponding folding rates, sourced from the PFDB and different research materials. Another dataset comprised detailed data regarding 48 physicochemical properties [35], [38] attributed to the 20 amino acids. Furthermore, data on 30 physicochemical properties of the 140 proteins was generated through the utilization of pfeature command lines. Additionally, a dataset was dedicated to protein composition data for the aforementioned proteins.[40], [41] The study

computed the 48 normalised physicochemical properties of the proteins by summing all the values provided for each amino acid and dividing them by the number of amino acids in that particular protein, resulting in a dataframe with 48 properties for all proteins. This dataset served as the independent variable for predicting the dependent variable, folding rates. Subsequently, the data underwent standardization, normalization, and feature scaling to determine the most effective method and machine learning (ML) was used to build the model for prediction.[36], [37] Standardization involved the application of techniques such as StandardScaler, MinMaxScaler with the most effective approach selected based on performance. MinMaxScaler proved optimal for the 48 protein properties data.[35], [36], [37]

Further in the study for the 2D descriptor the dataset comprises 52 single-domain two-state folding proteins' pdb files, each associated with ln(kf) values retrieved from PFDB and other research papers. As the other protein which were non two state folding proteins and had multidomain didn't show any better correlation between LFC and ln(kf).[42]

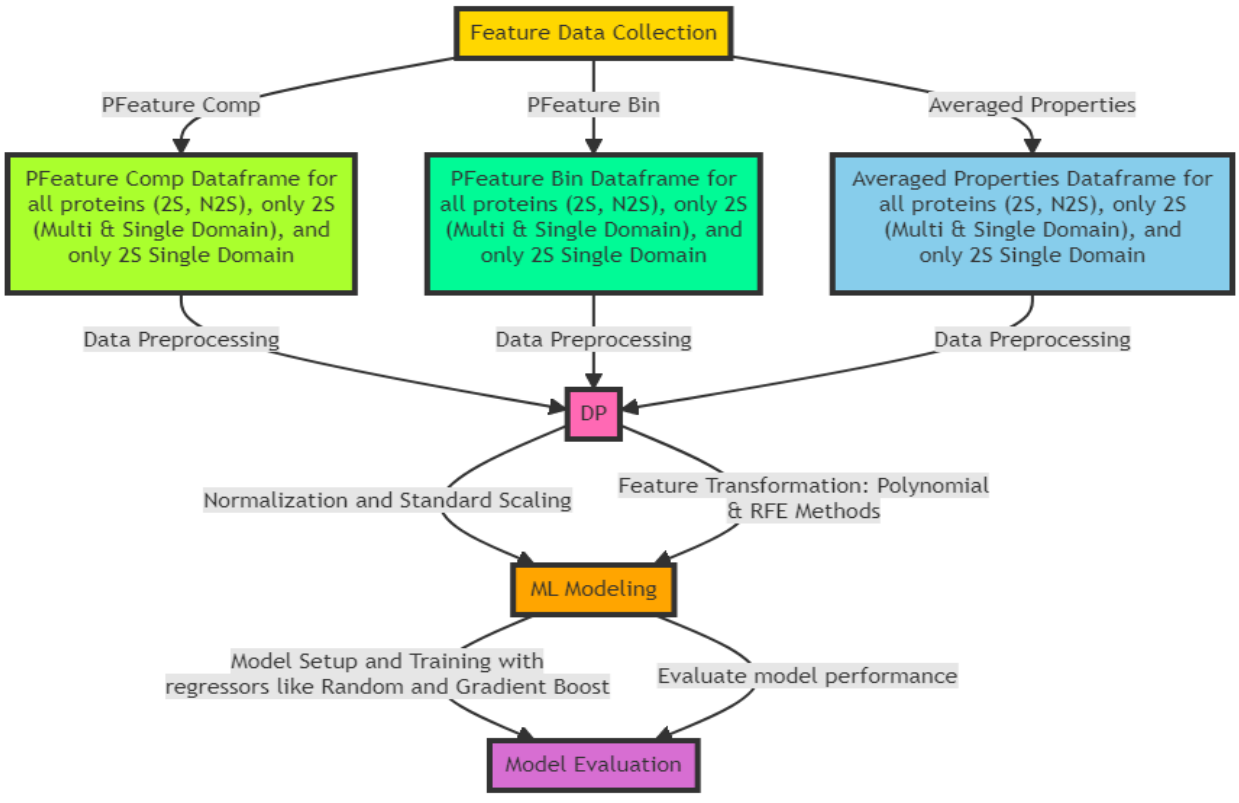


Fig. 2. Workflow for 1D descriptors.

Proteins are not only categorized by their functions but also by their distinctive physical and chemical properties. Among these, 2S proteins are particularly notable for their unique features and essential roles in plant development. These proteins are classified based on their sedimentation coefficients, indicated by '2S' which denotes their relative speed during ultracentrifugation, measured in Svedberg units. Typically small, 2S proteins consist of two subunits connected by disulfide bonds.[43]A defining characteristic of 2S proteins is their high content of sulfur-containing amino acids, particularly cysteine and methionine, which are crucial for human nutrition. This makes 2S proteins exceptionally valuable, notably in seeds like the Brazil nut, which is celebrated for its methionine-rich albumin. In the life cycle of plants, 2S proteins play a critical role as storage reserves, providing essential nutrients that support seed germination and early growth, vital for the survival of seedlings.

In contrast, non-2S proteins encompass a broad variety of protein types that differ significantly from 2S proteins in terms of solubility, size, and amino acid composition. This group includes other seed storage proteins such as prolamin, globulin, and glutelin, each classified by their solubility—prolamins in alcohol-water mixtures, globulins in salt solutions, and glutelins in dilute acids or bases.

These proteins fulfill various functions in plant physiology and seed nutrition. For instance, globulins constitute the majority of storage proteins in many seeds and are crucial for providing the amino acids necessary for the growth of the plant. A thorough understanding of these proteins' solubility and structural properties is essential for tailoring agricultural practices and food processing techniques to enhance the nutritional value of crop seeds. This deep knowledge is particularly crucial in breeding programs aimed at enhancing the dietary quality of plant-based foods, thereby more effectively addressing global nutritional needs.

Single-domain proteins are characterized by a single continuous polypeptide chain that forms a complete, independent structural and functional unit. These proteins tend to be smaller and simpler compared to their multi-domain counterparts. This simplicity offers several biological advantages:

Stability: Single-domain proteins often display enhanced stability, making them more resistant to denaturation under extreme conditions such as high temperatures or harsh chemical environments. This trait is particularly beneficial for organisms inhabiting extreme environments.

Efficient Folding: The straightforward structure of single-domain proteins allows them to fold into their functional shapes more quickly and efficiently. This rapid folding reduces the risk of misfolding, a common issue in more complex proteins, thus ensuring proper function in cellular processes that demand swift protein assembly.

Specialization: Typically, single-domain proteins are highly specialized. They are designed to perform specific tasks that do not require the modular versatility found in multi-domain proteins. Many enzymes and small hormone proteins are examples of single-domain proteins, each optimized to interact with precise molecular targets.

Contrasting with single-domain proteins, multi-domain proteins are composed of two or more polypeptide regions, each capable of folding into an independent structural unit. These proteins are generally larger and have a modular architecture, which confers several advantages:

Functional Diversity: The presence of multiple domains allows multi-domain proteins to engage in a broader range of functions within a single molecule. This versatility enables them to participate in complex interactions and execute intricate tasks within the cell.

Evolutionary Adaptability: Multi-domain proteins can evolve new functions through mechanisms such as exon shuffling or gene duplication. This flexibility aids organisms in developing novel protein capabilities, thereby enhancing adaptability and survival. This trait is particularly beneficial in processes like signaling pathways and gene regulation.

Allosteric Regulation: The domains within a multi-domain protein can influence each other's functions, facilitating allosteric regulation. This interaction enables a single multi-domain protein to act as a central hub in cellular networks, altering its conformation and activity in response to various signals. Proteins like kinases and phosphatases often contain regulatory domains that adjust their enzymatic activities based on cellular conditions, providing a dynamic response essential for maintaining cellular homeostasis.

The sum of absolute signal intensities in the lower eigenvalues was designated as the Low Frequency Component (LFC), with higher frequencies considered noise and lower frequencies deemed useful. Researchers aimed to determine the optimal cutoff (˜λcutoff) that maximizes the correlation between LFC and ln(kF), where kF represents the rate of protein folding. The cutoff frequency (ˆλcutoff) was adjusted incrementally in steps of 0.01 from 0.01 to λmax to achieve this goal. This cutoff value indicates the extent of the co-occurrence neighborhood of a related property. In the analysis of protein residue-network correlation with folding rate, the threshold signifies the breadth of the co-occurrence neighborhood of a property among connected residues. Preprocessing procedures involved the generation of sequence files and distance matrices.[42]

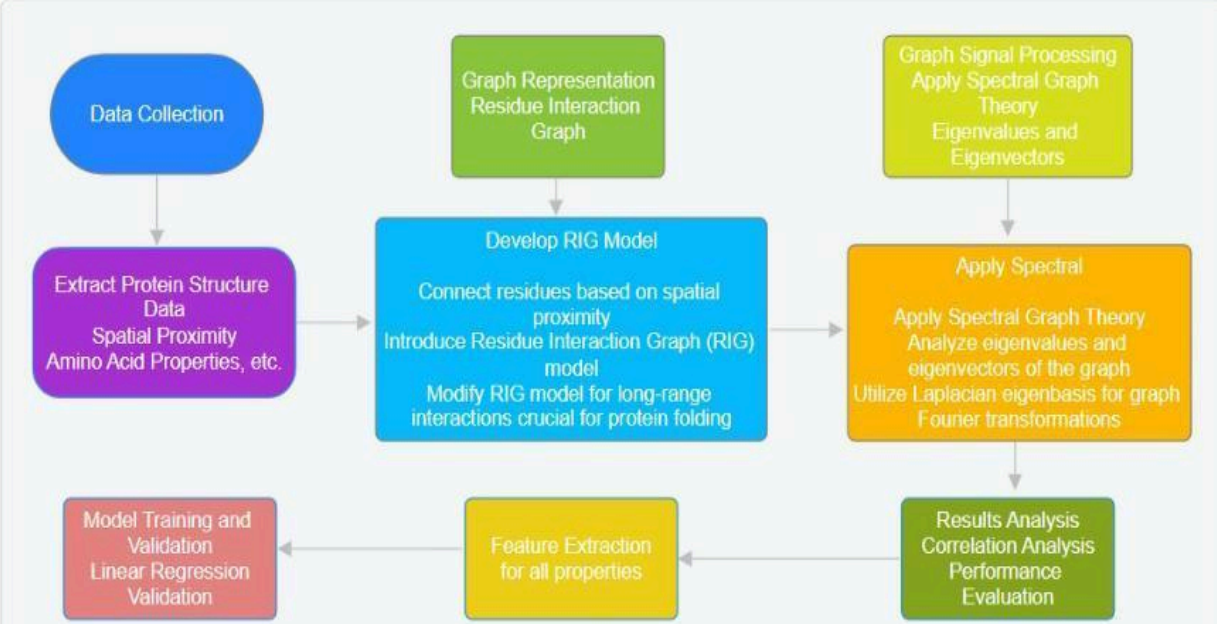


Fig. 3. Workflow for 2D descriptors.

Preprocessing entailed utilizing the 48 amino acid characteristics as signals for the 52 folding protein structures, which are single-domain and two-state, obtained from PFDB. Utilizing Bio.PDB and numpy, the coordinates from pdb files were extracted, facilitating the creation of distance matrices to illustrate interactions between "CA" residues in the proteins.[44]

Residues were interconnected based on their spatial proximity using the residue interaction graph (RIG) model. A modified RIG model was introduced to accommodate long-range interactions crucial for protein folding. To achieve this, the distance matrix was converted into binary matrices for rig-boolean[42], [43]

Various varieties of weighted-rig networks were examined. A predefined threshold was employed to categorize diverse thermodynamic parameters, such as bulkiness, surrounding hydrophobicity, and thermodynamic transfer hydrophobicity, as signals. The utilization of the graph Fourier approach facilitated the application of spectral graph theory in analyzing the graph's eigenvalues and eigenvectors. This enabled the exploration of informative low-frequency components of graph signals and their correlation with rates of protein folding. [44]

Among the 48 attributes assessed, 16 exhibited promising correlations with protein folding rates. These attributes, including Amino Molecular Mass and Unfolding Gibbs Free Energy Changes Chain, were selected for further scrutiny to enhance 1D descriptors. To enhance the prediction accuracy of protein folding speeds, the subsequent phase involves constructing a linear multivariate regression model that incorporates these attributes.[45]

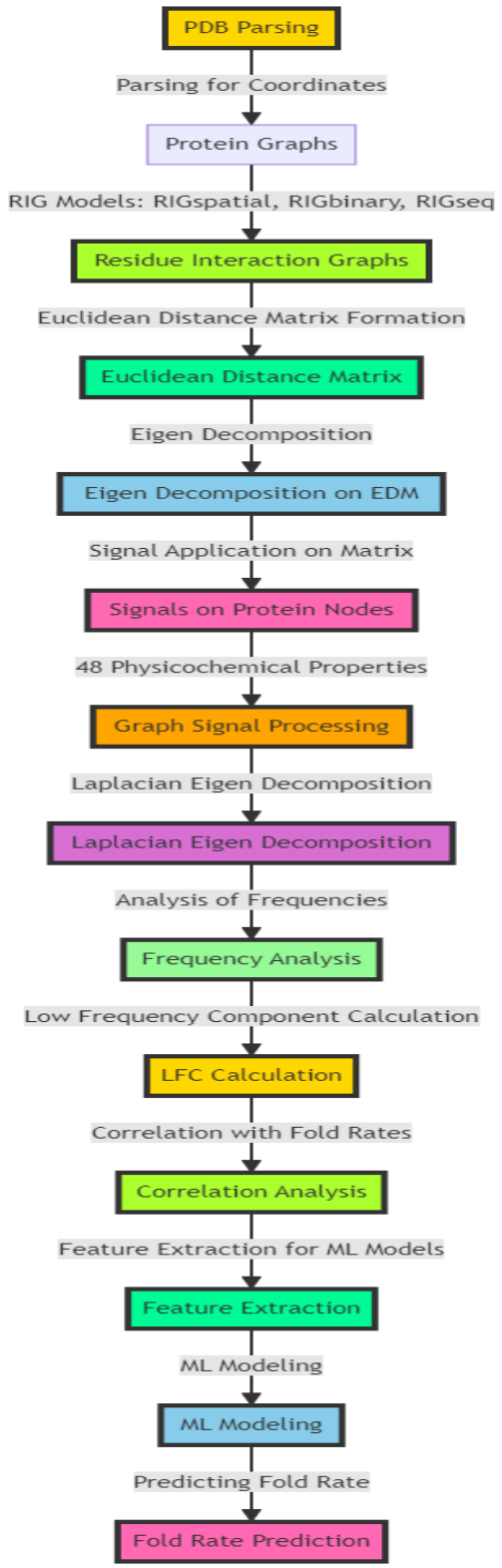


Fig.4. Pipeline for 2D descriptor

Further in the study all the proteins underwent implicit solvent MD Simulation using Amber23 as the Energy-based descriptor and the data from the production file was used to extract the average values of total energy, kinetic energy, potential energy, angles, bond, dihedrals, van der waals , electrostatic interactions, solvation(GB) energy, non bonded interactions and surface energy. From the coordinate file radius of gyration was also calculated. Molecular dynamics (MD) simulations are a fundamental tool in computational molecular studies, providing profound insights into the intricate behaviors of atoms and molecules across various scientific disciplines. Relying on the principles of classical mechanics, specifically Newtonian physics, MD simulations model the motion of particles within a system, enabling researchers to visualize and analyze molecular systems at an atomic scale over time. This capability often extends beyond what can be achieved through experimental approaches alone.[47]

At the core of MD simulations is their ability to accurately model molecular interactions by solving Newton's equations of motion for particles at discrete time intervals. This microscopic view into molecular dynamics allows for detailed exploration of energy landscapes and the structural transformations of biomolecules. Consequently, MD simulations are widely employed in fields such as biophysics, materials science, and pharmacology. They are instrumental in studying phenomena like protein folding, drug-receptor interactions, and the behavior of materials under different environmental conditions.[47], [48]

A pivotal element of MD simulations is the use of force fields, which mathematically describe the potential energy in a system as a function of atomic positions. These force fields are crucial for calculating the forces acting on atoms, guiding their trajectories within the simulated environment. Various potential energy functions, such as the Lennard-Jones potential, are utilized to accurately represent different types of interatomic forces, including van der Waals interactions and electrostatic forces.

The increasing power of computational resources and advancements in algorithms have significantly broadened the scope and enhanced the precision of MD simulations. The integration of high-performance computing platforms with advanced simulation techniques supports the modeling of larger and more complex molecular systems over extended periods. This advancement enriches our understanding of molecular mechanisms and interactions, which is essential for the development of new therapeutic drugs and innovative materials.[47], [48]

MD simulations stand as a cornerstone of contemporary scientific inquiry, bridging the gap between theoretical studies and practical applications. With ongoing advancements in MD methodologies and computational technologies, the potential of this essential scientific tool continues to grow, promising to propel further innovations across multiple domains of research.[47], [48]

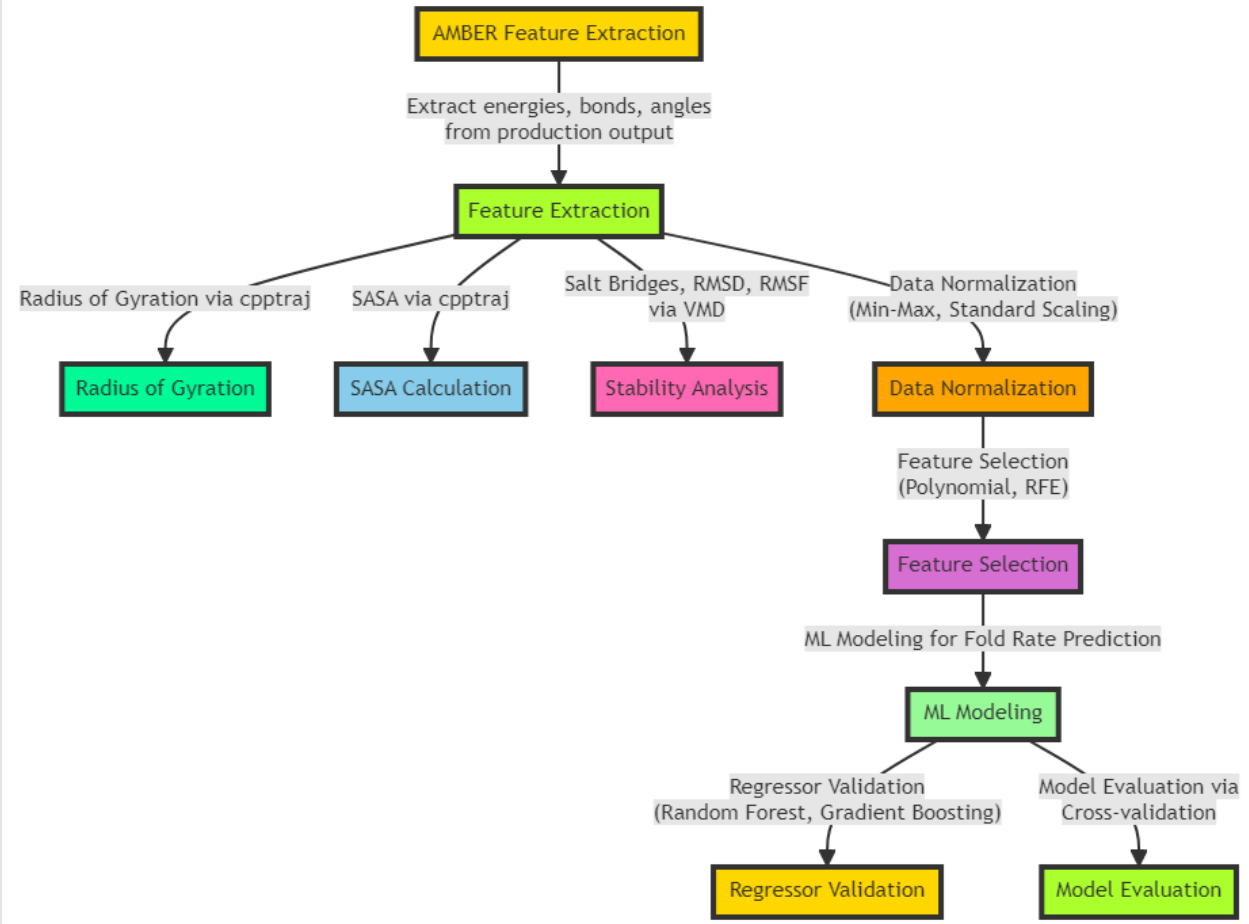


Fig.5. WorkFlow for Energy-based descriptor

CHAPTER 2

MATERIALS & METHODOLOGY

2.1 Data Collection

2.1.1 Collection of Protein Data

Some of the necessary information about the rate at which proteins fold was derived from PFDB datasets and earlier research. The standard protein folding kinetics database (PFDB) was created by the researchers by computing the logarithmic rate constants of all proteins that were observed at 25°C, the standard temperature. In order to account for proteins whose folding kinetics were initially determined at temperatures other than 25°C, a temperature adjustment based on the Eyring–Kramers equation was implemented to make them comparable. By comparing the estimated and empirically observed logarithmic rate constants of 14 distinct proteins at 25°C, this correction was verified and the quality of the database was found to have improved. PFDB is the largest database currently available on the kinetics of protein folding, with 141 single-domain globular proteins (89 two-state and 52 non-two-state proteins). Serving as a benchmark for the creation and evaluation of upcoming theoretical and predictive protein folding research is the primary goal of PFDB.[27]

Table 2: Fold rate data For 2S proteins

Protein short name	PDB	Class	ln(kf)
Arc repressor	1ARR	α	9.2
R16	1CUN (7-112)	α	4.8
R17	1CUN (113-219)	α	3.4
RAP1 (Human)	1FEX	α	8.2

c-Myb	1IDY	α	8.7
PAB	1PRB	α	13.8
Protein YjbJ	1RYK	α	9.1
IGBPA	1SS1	α	11.5
ACBP (Yeast)	1ST7	α	8.5
R15	1U4Q	α	11
PSBD (Bacillus stearothermophilus)	1W4E	α	10.2
PSBD (Pyrobaculum aerophilum)	1W4J	α	12.3
Rd-apocyt b562	1YYJ	α	8.4
De novo bundle a3d	2A3D	α	12.2
C-NPM1	2LLH	α	7
BBL	2WXC	α	11.7
SAP domain of THO1	2WQG	α	8.5
RNase HII (Thermococcus kodakaraensis)	1IO2	α/ß	-0.25
Myotrophin	2MYO	α+ß	4.8

Urm1	2QJL	α+ß	2.6
FKBP12	1D6O	α+ß	1.6
CksHs1 (Human)	1DKT	α+ß	4.5
LysM domain	1E0G	α+ß	7
U1A	1FHT	α+ß	4.6
Ribosomal protein L23 (Thermus thermophilus)	1N88	α+ß	2
ADAh2	1O6X	α+ß	6.8
raf RBD	1RFA	α+ß	7.7
NBR1-PB1	2BKF	α+ß	6.2
Ribosomal protein S6 (aquifex aeolicus)	2J5A	α+ß	7.3
ACYP1 (Human)	2VH7	α+ß	0.84
CI2	3CI2	α+ß	5.8
Apocytochrome b5 hydrophilic domain	1EHB	α+ß	3
Frataxin (Yeast)	2GA5	ß	5.4
NF-Kappa-B-complex	1NFI	ß	1.8

CheW (Thermotoga maritima)	1K0S	ß	7.4
Protein S (N-terminal domain)	1PRS (1-88)	ß	3
Protein S (C-terminal domain)	1PRS (91-173)	ß	-2
FBP28 WW domain	1E0L	ß	10.1
Prototype WW domain	1E0M	ß	8.9
Hisactophilin	1HCD	ß	1.3
abp1 SH3	1JO8	ß	2.5
Cafn2	1K85	ß	1.4
PDZ3 from PSD-95	1TP3	ß	3
GW1 of Internalin B	1M9S	ß	4
MTCP1 oncogene product P13	1QTU	ß	-0.36
FGF-1 (Human)	1RG8	ß	1.3
Fyn SH3	1AVZ	ß	4.9
Sho1 SH3 (Yeast)	2VKN	ß	2.1
9FNIII	1FNF	ß	-0.9

PDZ2 from PTP-BL	1GM1	ß	1
SPCp41	2JMC	ß	3.3
FGF-2 (Human)	1FGA	ß	-1.4
PDZ1 from PSD-95	3ZRT (3-93)	ß	1.3
PDZ2 from PSD-95	3ZRT (97-189)	ß	0.3
nNOS PDZ	1QAU	ß	1.8
Symfoil-4T	3O4B	ß	4.3
Symfoil-1	3O49	ß	1.7
Symfoil-4P	3O4D	ß	4.9
gpW	2L6R	ß	10.3
PDZ2 from SAP97	2X7Z	ß	0.74
Lambda-Repressor	1LMB	α	8.5
Im9	1IMQ	α	7.31
Horse Heart Cytochrome C	1HRC	α	8.76
Cytochrome B562	256B	α	12.2

VILLIN 14T	1VII	α	11.52
Staphylococcus areus Protein A	1BDD	α	11.75
Engrailed Homeodomain	1ENH	α	10.53
Dihydrolipoamide dehydrogenase	1EBD	α	9.68
FYN Proto-oncogene Tyrosine Kinase	1NYF	ß	4.54
PI3K SH3 domain	1PKS	ß	1.05
A SRC-homology 3 (SH3) domain	1SHG	ß	1.41
SRC Tyrosine Kinase SH3 domain	1SRL	ß	4.04
Twitchin	1WIT	ß	0.41
CspB (Bacillus subtilis)	1CSP	ß	6.98
Cspa	1MJC	ß	5.24
PI3 SH3	1PNJ	ß	1.1
FYN Tyrosine Kinase SH3 domain	1SHF	ß	4.5
CspB (Bacillus caldolyticus)	1C9O	ß	7.2
Cold shock-like Protein (Thermotoga maritima)	1G6P	ß	6.3

Sso7d	1C8C	ß	6.91
Alfa-spectrin SRC homology 3 domain	1AEY	ß	2.09
TI I27	1TIT	ß	3.47
Protein (cold-shock Protein A)	3MEF	ß	5.3
ACYP2 (Horse)	1APS	α+ß	1.48
HPr	1HDN	α+ß	2.7
Protein (U1A)	1URN	α+ß	5.73
B1 domain of Protein L (Finegoldia magna)	2PTL	α+ß	4.1
FK506 binding Protein	1FKB	α+ß	1.46
Chymotrypsin inhibitor 2	1COA	α+ß	3.87
Ribosomal Protein L9	1DIV	α+ß	6.58
Hybrid between Chymotrypsin inhibitor 2 and Helix E	1CIS	α+ß	3.87
Protein L	1HZ6	α+ß	4.1
B1 domain of Protein G (Streptococcal sp. group G)	1PGB	α+ß	6

Chymotrypsin inhibitor 2	2CI2	α+ß	3.9
Procarboxypeptidase A2	1AYE	α+ß	6.8
Ribosomal Protein S6 (Thermus thermophilus)	1RIS	α+ß	5.9
Histidine containing phophocarrier Protein HPR	1POH	α+ß	2.7
Barnase-Barstar complex	1BRS	α+ß	3.4
ACYL-phosphatase	2ACY	α+ß	0.92
Ubiquitin	1UBQ	α+ß	7.33
HIV-1 protease	1VIK	α+ß	6.8

Table 3: Fold rate data For N2S proteins

Protein short name	**PDB**	**Class**	**ln(kf)**
Apomyoglobin (Whale)	1A6N	α	1.1
Pit1	1AU7	α	9.7
4-helix bundle protein FRB	1AUE	α	5.4
IM7	1AYI	α	5.7
Apomyoglobin (Horse)	1DWR	α	2.9

FF domain from human HYPA/FBP11	1UZC	α	8
ACBP (Bovine)	1NTI	α	6.5
Phage 434 Cro	2CRO	α	3.7
Barstar	1BTA	α/ß	3.5
Apoflavodoxin (Anabaena)	1FTG	α/ß	2.3
HIV-1 RNase H	1HRH	α/ß	0.88
N-PGK (Bacillus stearothermophilus)	1PHP (1–175)	α/ß	2.3
C-PGK (Bacillus stearothermophilus)	1PHP (186–394)	α/ß	-4
DHFR	1RA9	α/ß	-0.37
Trp-synthase a-subunit (Escherichia coli)	1WQ5	α/ß	-2.1
RNase H (Escherichia coli)	2RN2	α/ß	-0.3
CheY	3CHY	α/ß	1
Apoflavodoxin (Desulfovibrio desulfuricans)	3F6R	α/ß	3.5
sIGPS (Sulfolobus solfataricus)	1IGS	α/ß	-4.5

RNase H (Chlorobaculum tepidum)	3H08	α/ß	1.6
HisF	1THF	α/ß	-3.2
GFP	1B9C	α+ß	-2.6
Barnase	1BNI	α+ß	2.7
p16INK4a	2A5E	α+ß	3.5
N-HypF	1GXT	α+ß	4.4
Monellin	1FA3	α+ß	4.1
T4 Lysozyme	1L63	α+ß	3.7
p13suc1	1PUC	α+ß	4.2
Villin 14T	2VIL	α+ß	4.2
ß-Lactamase (Staphylococcus aureus)	3BLM	α+ß	-6.6
ACYP (Sulfolobus solfataricus)	2BJD	α+ß	1.7
UCH-L3	1UCH	α+ß	-2.6
Ubq-UIM	2KDI	α+ß	2.3
Frataxin (Human)	1EKG	α+ß	3.5

ß-Lactamase (Bacillus licheniformis)	4BLM	α+ß	-4.7
CD2.d1	1HNG	ß	1.8
IL-1ß	1I1B	ß	-4
10FNIII	1TTG	ß	5.5
CRABPI (Mouse)	1CBI	ß	-3.2
CRBPII (Rat)	1OPA	ß	1.4
IFABP	1IFC	ß	4.3
Carbonic anhydrase II (Bovine)	1V9E	ß	-4.4
CRABPII (Mouse)	2FS6	ß	2.3
Pseudoazurin	1ADW	ß	0.69
SNase	2PQE	ß	2.2
IL-33	2KLL	ß	-1.4

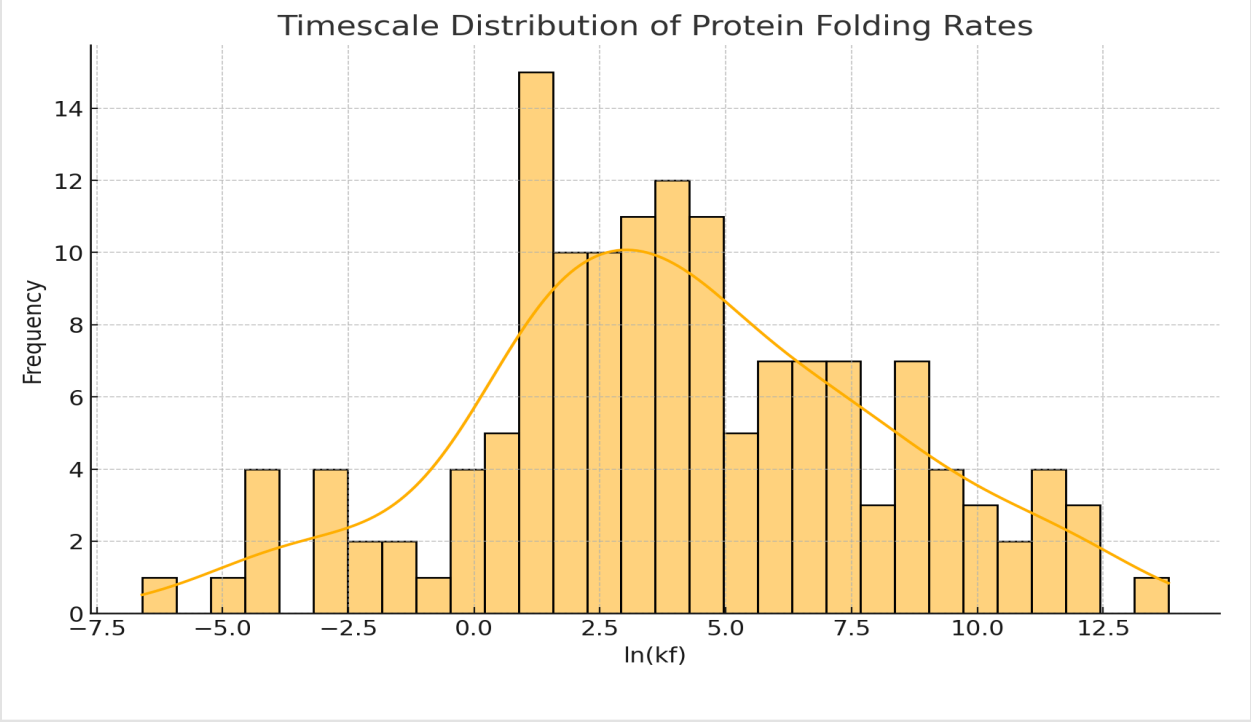
Compared to all the proteins combined or just 2s and n2s separately, 2S single domain proteins—which we gathered from research papers—provide superior results. The performance of single-domain 2s proteins was comparatively good.[28]

Table 4: Fold rate data For 2S single domain proteins

PDB ID	Ln (K_f)	Protein Family
1AEY	2.09	ß

1APS	1.48	α+ß
1AYE	6.8	α+ß
1BDD	11.75	α
1BRS	3.4	α+ß
1C8C	6.91	ß
1C9O	7.2	ß
1CIS	3.87	α+ß
1COA	3.87	α+ß
1CSP	6.98	ß
1DIV	6.58	α+ß
1EBD	9.68	α
1ENH	10.53	α
1FKB	1.46	α+ß
1G6P	6.3	ß
1HDN	2.7	α+ß
1HRC	8.76	α
1HZ6	4.1	α+ß
1IMQ	7.31	α
1LMB	8.5	α
1LOP	6.6	ß
1MJC	5.24	ß
1NYF	4.54	ß
1PBA	6.8	α+ß
1PCA	6.8	α+ß
1PGB	6	α+ß
1PIN	9.44	ß
1PKS	1.05	ß
1PNJ	1.1	ß
1POH	2.7	α+ß
1RIS	5.9	α+ß
1SHF	4.5	ß
1SHG	1.41	ß

1SRL	4.04	ß
1TEN	1.06	ß
1TIT	3.47	ß
1UBQ	7.33	α+ß
1URN	5.73	α+ß
1VII	11.52	α
1VIK	6.8	α+ß
1WIT	0.41	ß
1YCC	9.62	α
256B	12.2	α
2ABD	6.55	α
2ACY	0.92	α+ß
2AIT	4.2	ß
2CI2	3.9	α+ß
2HQI	0.18	α+ß
2PDD	9.8	α
2PTL	4.1	α+ß
2VIK	6.8	α+ß
3MEF	5.3	ß



This timeframe distribution curve shows protein fold rate distribution. The histogram here displays how frequent the folding rates (ln(kf)) are for all these proteins, with kernel density estimate(KDE) overlay for smoother illustration of the distribution. Bin width is 0.68 units of ln(kf) approx. dividing the data in 30 intervals. where, lnkf ranges from -6 to 14 and most protein clusters between -4 and 12. The range where frequency is highest and most proteins lie there is 0.88 to 1.56 .

2.1.2 Data collection of 48 properties of amino acids from research papers

Numerical values of 48 selected Physico-chemical, Energetic and Conformational Properties of the 20 amino acids/residues.

No	Property	Ala	Asp	Cys	Glu	Phe	Gly	His	Ile	Lys	Leu	Met	Asn	Pro	Gln	Arg	Ser	Thr	Val	Trp	Tyr
1	K^0	-25.50	-33.12	-32.82	-36.17	-34.54	-27.00	-31.84	-31.78	-32.40	-31.78	-31.18	-30.90	-23.25	-32.60	-26.62	-29.88	-31.23	-30.62	-30.24	-35.01
2	H_t	0.87	0.66	1.52	0.67	2.87	0.10	0.87	3.15	1.64	2.17	1.67	0.09	2.77	0.00	0.85	0.07	0.07	1.87	3.77	2.67
3	Hp	13.05	11.10	14.30	11.41	13.89	12.20	12.42	15.34	11.01	14.19	13.62	11.72	11.06	11.78	12.40	11.68	12.12	14.73	13.96	13.57
4	P	0.00	49.70	1.48	49.90	0.35	0.00	51.60	0.10	49.50	0.13	1.43	3.38	1.58	3.53	52.00	1.67	1.66	0.13	2.10	1.61
5	pH_i	6.00	2.77	5.05	5.22	5.48	5.97	7.59	6.02	9.74	5.98	5.74	5.41	6.30	5.65	10.76	5.68	5.66	5.96	5.89	5.66
6	pK'	2.34	2.01	1.65	2.19	1.89	2.34	1.82	1.36	2.18	2.36	2.28	2.02	1.99	2.17	1.81	2.21	2.10	2.32	2.38	2.20
7	M_w	89.00	133.00	121.00	147.00	165.00	75.00	155.00	131.00	146.00	131.00	149.00	132.00	115.00	146.00	174.00	105.00	119.00	117.00	204.00	181.00
8	B_1	11.50	11.68	13.46	13.57	19.80	3.40	13.67	21.40	15.71	21.40	16.25	12.82	17.43	14.45	14.28	9.47	15.77	21.57	21.61	18.03
9	R_f	9.90	2.80	2.80	3.20	18.80	5.60	8.20	17.10	3.50	17.60	14.70	5.40	14.80	9.00	4.60	6.90	9.50	14.30	17.00	15.00
10	μ	14.34	12.00	35.77	17.26	29.40	0.00	21.81	19.06	21.29	18.78	21.64	13.28	10.93	17.56	26.66	6.35	11.01	13.92	42.53	31.55
11	H_{nc}	0.62	0.90	0.29	-0.74	1.19	0.48	-0.40	1.38	-1.50	1.06	0.64	-0.78	0.12	-0.85	-2.53	-0.18	-0.05	1.08	0.81	0.26
12	E_{sm}	1.40	1.16	1.37	1.16	1.14	1.36	1.22	1.19	1.07	1.32	1.30	1.18	1.24	1.12	0.92	1.30	1.25	1.25	1.03	1.03
13	E_1	0.49	0.35	0.67	0.37	0.72	0.53	0.54	0.76	0.30	0.65	0.65	0.38	0.46	0.40	0.55	0.45	0.52	0.73	0.83	0.65
14	E_t	1.90	1.52	2.04	1.54	1.86	1.90	1.76	1.95	1.37	1.97	1.96	1.56	1.70	1.52	1.48	1.75	1.77	1.98	1.87	1.69
15	P_α	1.42	1.01	0.70	1.51	1.13	0.57	1.00	1.08	1.16	1.21	1.45	0.67	0.57	1.11	0.98	0.77	0.83	1.06	1.08	0.69
16	P_β	0.83	0.54	1.19	0.37	1.38	0.75	0.87	1.60	0.74	1.30	1.05	0.89	0.55	1.10	0.93	0.75	1.19	1.70	1.37	1.47
17	P_t	0.66	1.46	1.19	0.74	0.60	1.56	0.95	0.47	1.01	0.59	0.60	1.56	1.52	0.98	0.95	1.43	0.96	0.50	0.96	1.14
18	P_c	0.71	1.21	1.19	0.84	0.71	1.52	1.07	0.66	0.99	0.69	0.59	1.37	1.61	0.87	1.07	1.34	1.08	0.63	0.76	1.07
19	C_a	20.00	26.00	25.00	33.00	46.00	13.00	37.00	39.00	46.00	35.00	43.00	28.00	22.00	36.00	55.00	20.00	28.00	33.00	61.00	46.00
20	F	0.96	1.14	0.87	1.07	0.69	1.16	0.80	0.76	1.14	0.79	0.78	1.04	1.16	1.07	1.05	1.13	0.96	0.79	0.77	1.01
21	B_f	0.38	0.14	0.57	0.09	0.51	0.38	0.31	0.56	0.04	0.50	0.42	0.15	0.18	0.11	0.07	0.23	0.23	0.48	0.40	0.26
22	R_a	3.70	2.60	3.03	3.30	6.60	3.13	3.57	7.69	1.79	5.88	5.21	2.12	2.12	2.70	2.53	2.43	2.60	7.14	6.25	3.03
23	N_s	6.05	4.95	7.86	5.10	6.62	6.16	5.80	7.51	4.88	7.37	6.39	5.04	5.65	5.45	5.70	5.53	5.81	7.62	6.98	6.73
24	α_n	1.59	0.53	0.33	1.45	1.14	0.53	0.89	1.22	1.13	1.91	1.25	0.53	0.00	0.98	0.67	0.70	0.75	1.42	1.33	0.58
25	α_c	1.44	2.13	0.76	2.01	1.01	0.62	0.56	0.68	0.59	0.58	0.73	0.93	2.19	1.20	0.39	0.81	1.25	0.63	1.40	0.72
26	α_m	1.22	0.56	1.53	1.28	1.13	0.40	2.23	0.77	1.65	1.05	1.47	0.93	0.00	1.63	1.59	0.87	0.46	1.20	0.46	0.52
27	V^0	60.46	73.83	67.70	85.88	121.48	43.25	98.79	107.72	108.50	107.75	105.35	78.01	82.83	93.90	127.34	60.62	76.83	90.78	143.91	123.60
28	N_m	2.11	1.80	1.88	2.09	1.98	1.53	1.98	1.77	1.96	2.19	2.27	1.84	1.32	2.03	1.94	1.57	1.57	1.63	1.90	1.67
29	N_1	3.92	2.85	5.55	2.72	4.53	4.31	3.77	5.58	2.79	4.59	4.14	3.64	3.57	3.06	3.78	3.75	4.09	5.43	4.83	4.93
30	H_{gm}	13.85	11.61	15.37	11.38	13.93	13.34	13.82	15.28	11.58	14.13	13.86	13.02	12.35	12.61	13.10	13.39	12.70	14.56	15.48	13.88
31	ASA_D	104.00	132.20	132.50	161.90	182.00	73.40	165.80	171.50	195.20	161.40	189.80	134.90	135.10	164.90	210.20	111.40	130.40	143.90	208.80	196.40
32	ASA_N	33.20	62.40	17.90	81.00	33.10	29.20	57.70	28.30	107.50	31.10	41.30	60.50	60.70	71.50	94.50	48.70	52.00	28.10	39.50	50.40
33	ΔASA	70.90	69.60	114.30	80.50	148.40	44.00	107.90	142.70	87.50	129.80	147.90	74.00	73.50	93.30	116.00	62.80	78.00	115.60	167.80	145.90
34	ΔG_h	-0.54	-2.97	-1.64	-3.71	-1.06	-0.59	-3.38	0.32	-2.19	0.27	-0.60	-3.55	0.32	-3.92	-5.96	-3.82	-1.97	0.13	-3.80	-5.64
35	G_{hD}	-0.58	-6.10	-1.91	-7.37	-1.35	-0.82	-5.57	0.40	-5.97	0.35	-0.71	-6.63	0.56	-7.12	-12.78	-6.18	-3.66	0.18	-4.71	-8.45
36	G_{hN}	-0.06	-3.11	-0.27	-3.62	-0.28	-0.23	-2.18	0.07	-1.70	0.07	-0.10	-3.03	0.23	-3.15	-6.85	-2.36	-1.69	0.04	-0.88	-2.82
37	ΔH_h	-2.24	-4.54	-3.43	-5.63	-5.11	-1.46	-6.83	-3.84	-5.02	-3.52	-4.16	-5.68	-1.95	-6.23	-10.43	-5.94	-4.39	-3.15	-8.99	-10.67
38	$-T\Delta S_h$	1.70	1.57	1.79	1.92	4.05	0.87	3.45	4.16	2.83	3.79	3.56	2.13	2.27	2.31	4.47	2.12	2.42	3.28	5.19	5.03
39	ΔC_{ph}	14.22	2.73	9.41	3.17	39.06	4.88	20.05	41.98	17.68	38.26	31.67	3.91	23.69	3.74	16.66	6.14	16.11	32.58	37.69	30.54
40	ΔG_c	0.51	2.89	2.71	3.58	3.22	0.68	3.95	-0.40	1.87	-0.35	1.13	3.26	-0.39	3.69	5.25	3.42	1.74	-0.19	5.59	6.56
41	ΔH_c	2.77	4.72	8.64	5.69	11.93	1.23	7.64	4.03	3.57	3.69	7.06	3.64	1.97	4.47	6.03	5.80	4.42	3.45	13.46	14.41
42	$-T\Delta S_c$	-2.25	-1.83	-5.92	-2.11	-8.71	-0.55	-3.69	-4.42	-1.70	-4.04	-5.93	-0.39	-2.36	-0.78	-0.78	-2.38	-2.68	-3.64	-7.87	-7.95
43	ΔG	-0.02	-0.08	1.08	-0.13	2.16	0.09	0.56	-0.08	-0.32	-0.08	0.53	-0.30	-0.06	-0.23	-0.71	-0.40	-0.24	-0.06	1.78	0.91
44	ΔH	0.51	0.18	5.21	0.05	6.82	-0.23	0.79	0.19	-1.45	0.17	2.89	-2.03	0.02	-1.76	-4.40	-0.16	0.04	0.30	4.47	3.73
45	$-T\Delta S$	-0.54	-0.26	-4.14	-0.19	-4.66	0.31	-0.23	-0.27	1.13	-0.24	-2.36	1.74	-0.08	1.53	3.69	-0.24	-0.28	-0.36	-2.69	-2.82
46	v	1.00	4.00	2.00	5.00	7.00	0.00	6.00	4.00	5.00	4.00	4.00	4.00	3.00	5.00	7.00	2.00	3.00	3.00	10.00	8.00
47	s	0.00	2.00	0.00	3.00	2.00	0.00	2.00	1.00	0.00	2.00	0.00	2.00	0.00	3.00	5.00	0.00	1.00	1.00	2.00	2.00
48	f	0.00	2.00	1.00	3.00	2.00	0.00	2.00	2.00	4.00	2.00	3.00	2.00	0.00	3.00	5.00	1.00	1.00	1.00	2.00	2.00

We gathered information on 48 characteristics of amino acids from several research papers. These characteristics include polarity (P), which is expressed in Debye, surrounding hydrophobicity (Hp), which is likewise expressed in kcal/mol, and thermodynamic transfer hydrophobicity (Ht), which is measured in kcal/mol. Both the Equilibrium Constant for the ionization of COOH (pK′) and the Isoelectric Point (pHi) are expressed in pH units. The unit of measurement for molecular weight (Mw), often known as "Mw" or "Molecular Weight," is

grammes per mole or Daltons. Square angstroms (Å²) are used to depict bulkiness (Bl), while the dimensionless Chromatographic Index (Rf) is commonly represented by the symbol 'Rf'. Furthermore dimensionless is the Refractive Index (α), emphasizing its significance for light refraction as opposed to compressibility. Additionally, the unit of measurement for Normalised Consensus Hydrophobicity (Hnc) is kcal/mol. Measured in kcal/mol/atom, the energy measurements include Total Non-bonded Energy (Esm + El), Long Range Non-bonded Energy (El), and Short and Medium Range Non-bonded Energy (Esm). Pα, Pβ, Pt, and Pc are dimensionless metrics that indicate the α-helical, β-structure, turns, and coil inclinations, respectively. They lack specific symbols. The Mean r.m.s. Fluctuational Displacement (F) is expressed in angstroms (Å), whereas the Helical Contact Area (Ca) is expressed in square angstroms (Å²). In addition, but without any particular symbols, are Buriedness (Br), Solvent Accessible Reduction Ratio (Ra), and Average Number of Surrounding Residues (Ns). Dimensionless, the positional powers at the N-terminal, C-terminal, and middle of an α-helix are represented by the symbols αn, αc, and αm, respectively. Measured in cubic metres per mol (m^3/mol x 10^{-6}), the partial molar volume (V0) is quantified. Nm and Nl, which are likewise dimensionless, stand for Average Medium and Long Range contacts. Both the globular and membrane environments' combined surrounding hydrophobicity (Hgm) is expressed in kcal/mol. Measured in square angstroms (Å²), the surface areas of unfolding, native, and denatured proteins (ASAN, ΔASA, and ASAD) are listed. Gibbs Free Energy, Enthalpy, and Entropy changes that occur during protein hydration and unfolding processes (ΔGh, GhD, GhN, ΔHh, -TΔSh, ΔCph, ΔGc, ΔHc, -TΔSc, ΔG, ΔH, -TΔS) are expressed in kcal/mol with the exception of the unfolding hydration heat capacity change (ΔCph), which is expressed in cal/mol/K. In summary, 'v' denotes the volume of side-chain atoms that are not hydrogen, and 's' denotes the shape associated with a branch's position point; both are dimensionless in a side-chain.The flexibility, denoted by f, is measured in terms of side-chain dihedral angles. It is stated as a dimensionless quantity and without a distinct symbol, which reflects its relationship to molecular flexibility.

2.1.3 Data collection from Pfeature software

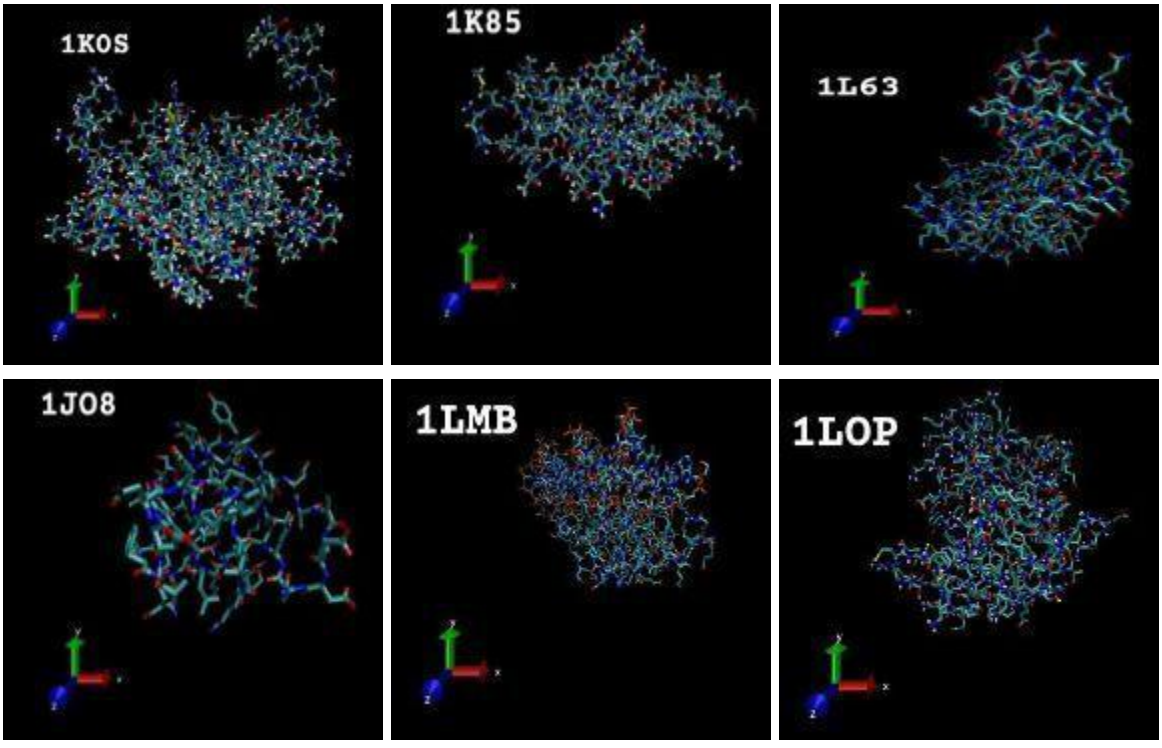
We downloaded the fasta files from the protein data bank using the PDB IDs, then we examined them to determine the different protein compositions. Physico-chemical Properties Composition (PCP), Atomic Composition (ATC), Tripeptide Composition (TPC), Dipeptide Composition (DPC), and Amino Acid Composition (AAC) are a few of these. Furthermore, we assessed the Shannon Entropy of Protein (SEP), Residues (SER), and Physico-chemical Properties (SPC), as well as Pseudo Amino Acid Composition (PAAC), Amphiphilic Pseudo Amino Acid Composition (APAAC), Quasi Sequence Order (QSO), Sequence Order Coupling Number (SOC). From these findings, we created three dataframes and combined all of the compositions into one dataframe for all proteins, 146 rows and 8,593 columns in size, with 2s and n2s included. In order to compare results, we additionally created separate dataframes for 2s proteins

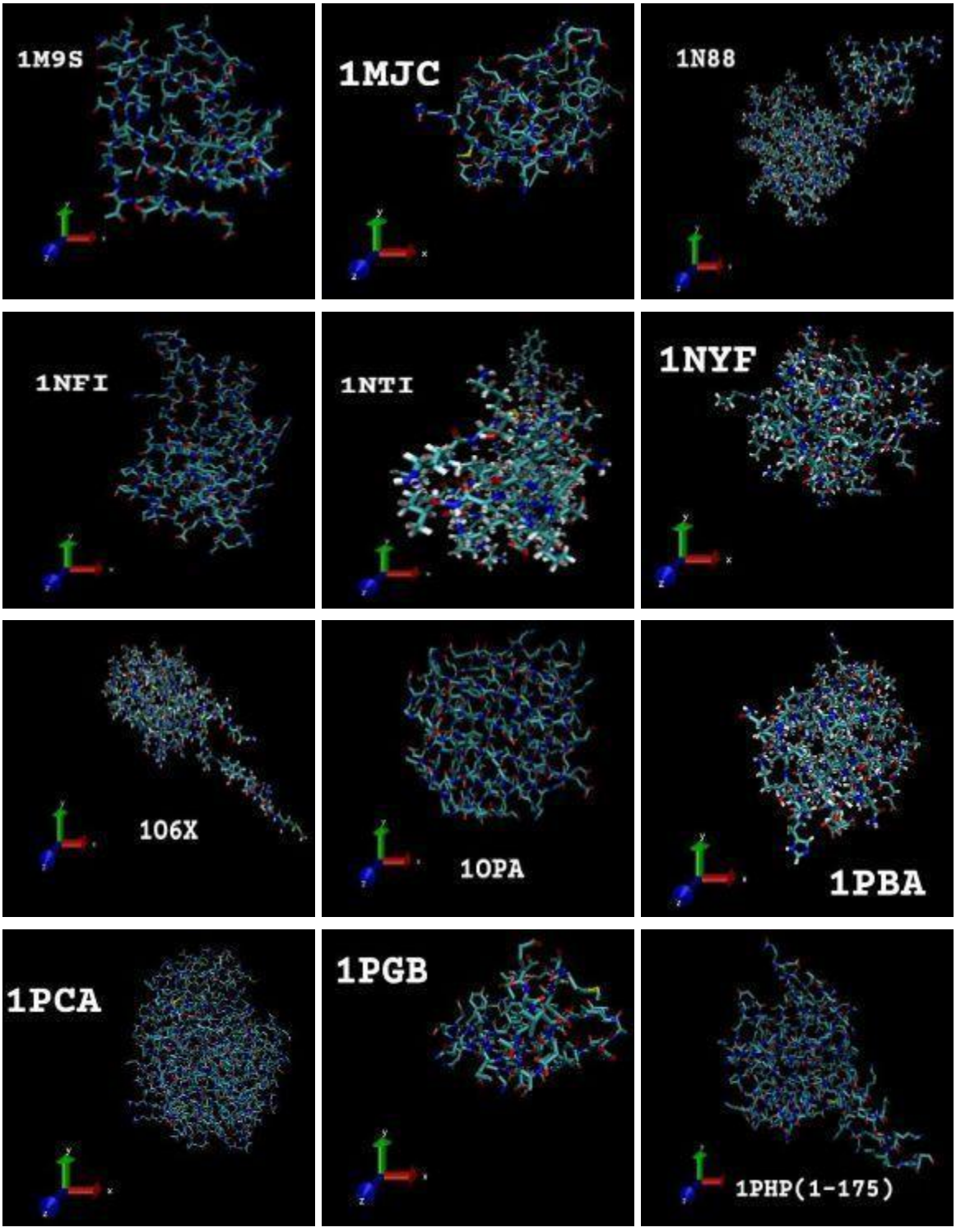
with several domains (101 rows x 8,593 columns) and another particularly for 2s proteins with a single domain (52 rows x 8,593 columns). [23], [49]

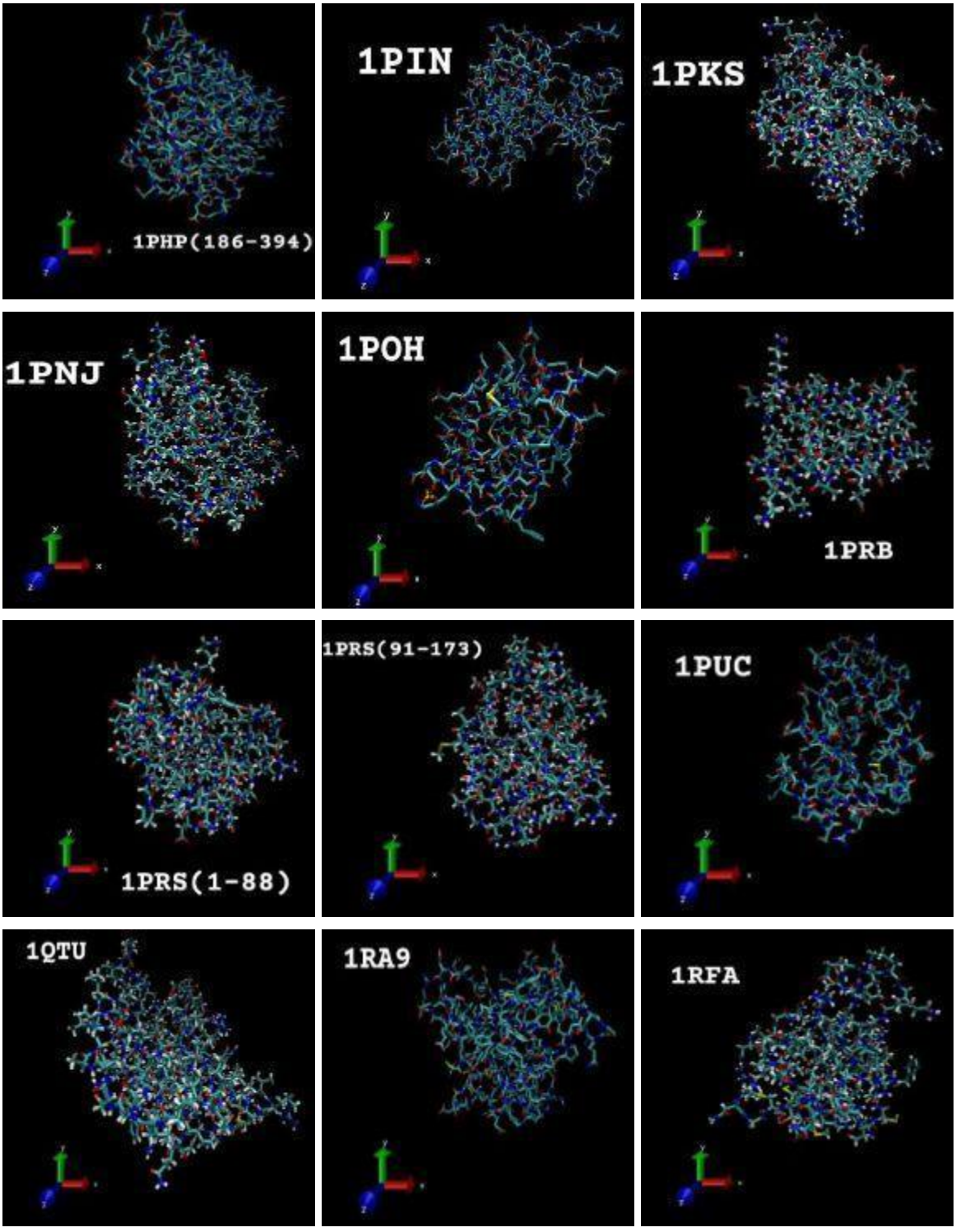
After normalizing these values using min_max scaling, we averaged them by adding the normalized values of one property for each of the complete amino acid sequences of a single protein and dividing the result by the total number of amino acids in the sequence. In this manner, we created three additional datasets, one for each of the proteins, all 2s proteins, and all 2s single domain proteins, each with 48 columns and a different number of rows based on protein count in datasets[23]

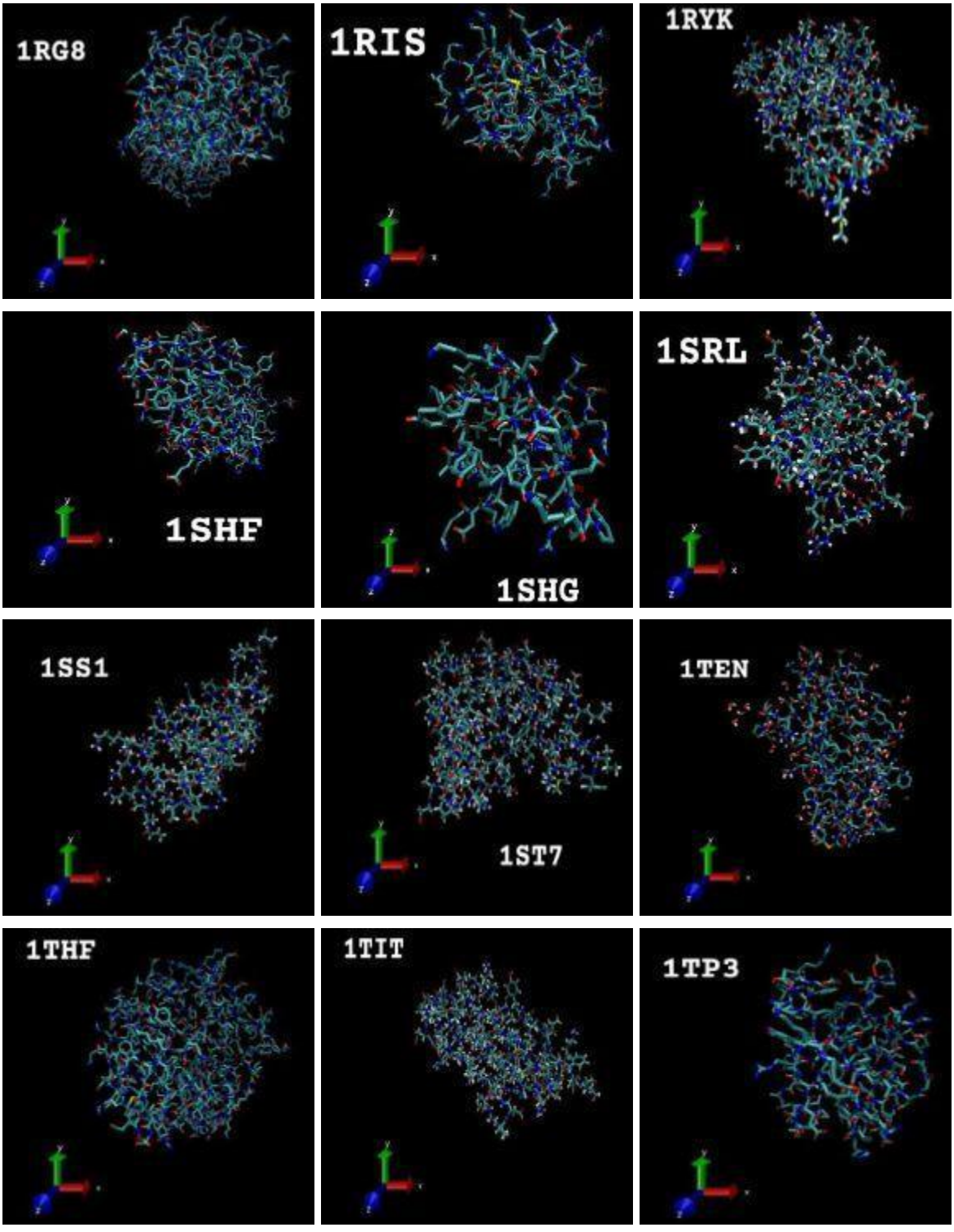
2.1.4 Data collection of protein PDB files for GSP and MD Simulation.

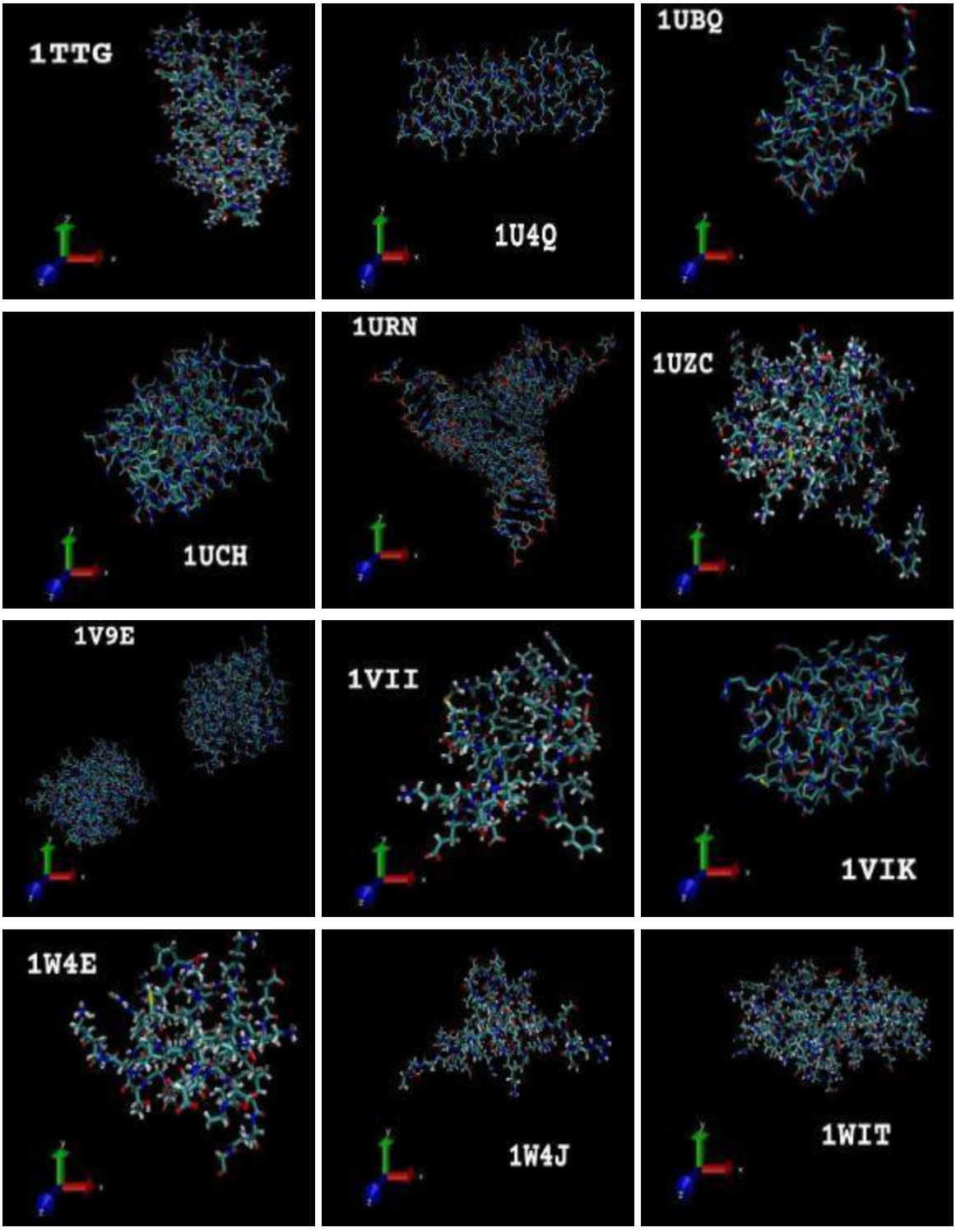
In order to process graph signals over protein PDB structures, the sequence data has been used in conjunction with the PDB structures, which have been parsed for both the sequences and the distance matrix. Here, the amino acid characteristics are treated as the node signals and the residues of the protein network were represented as graphs, with each protein represented by a network based on inter-residue closeness. The 48 attributes we previously possessed plus the five based on the network, which we extracted from the research articles, were the properties employed here. All of the information for each protein is shown above. Next, we used these pdb files to do MD Simulation for implicit solvent in Amber.

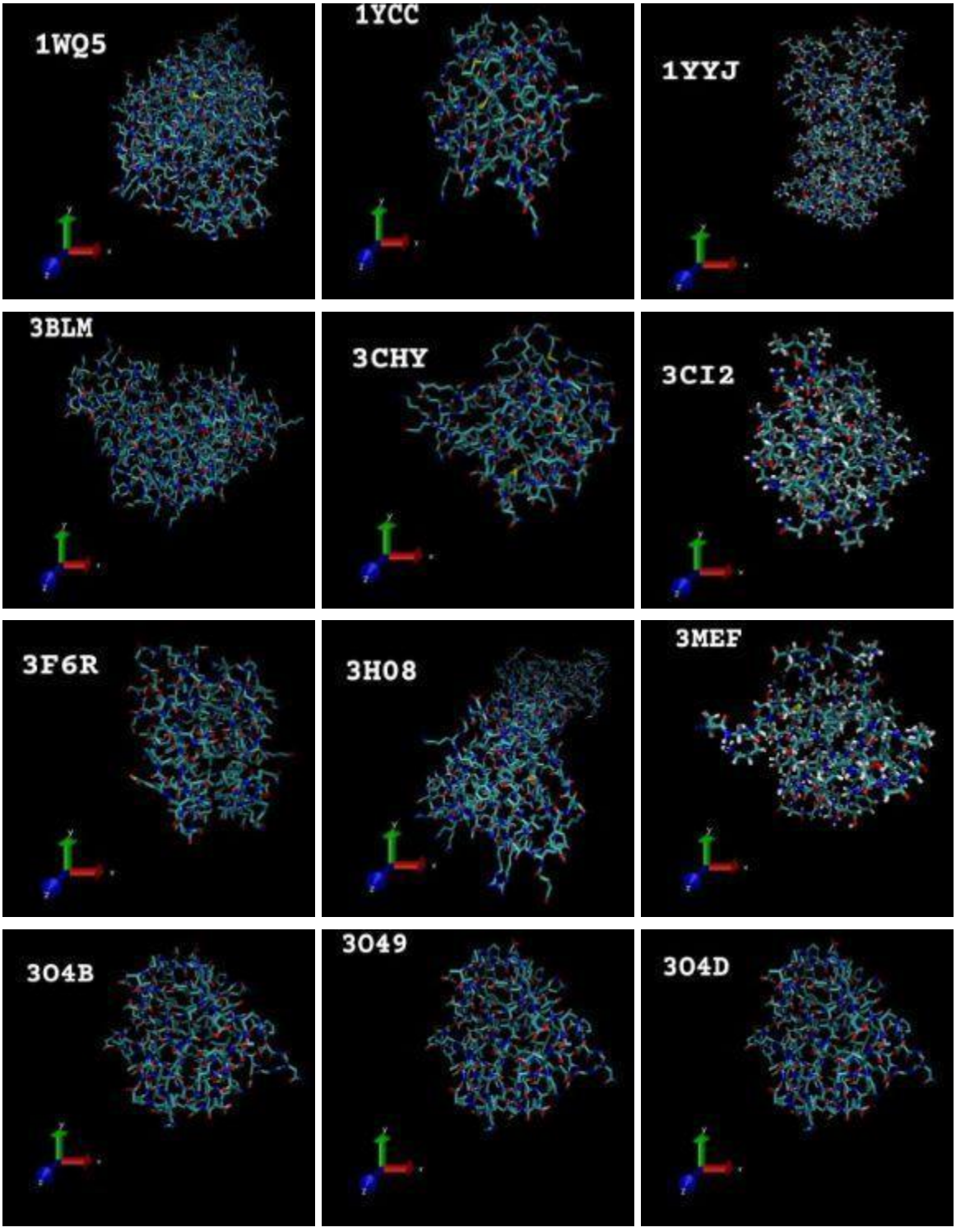


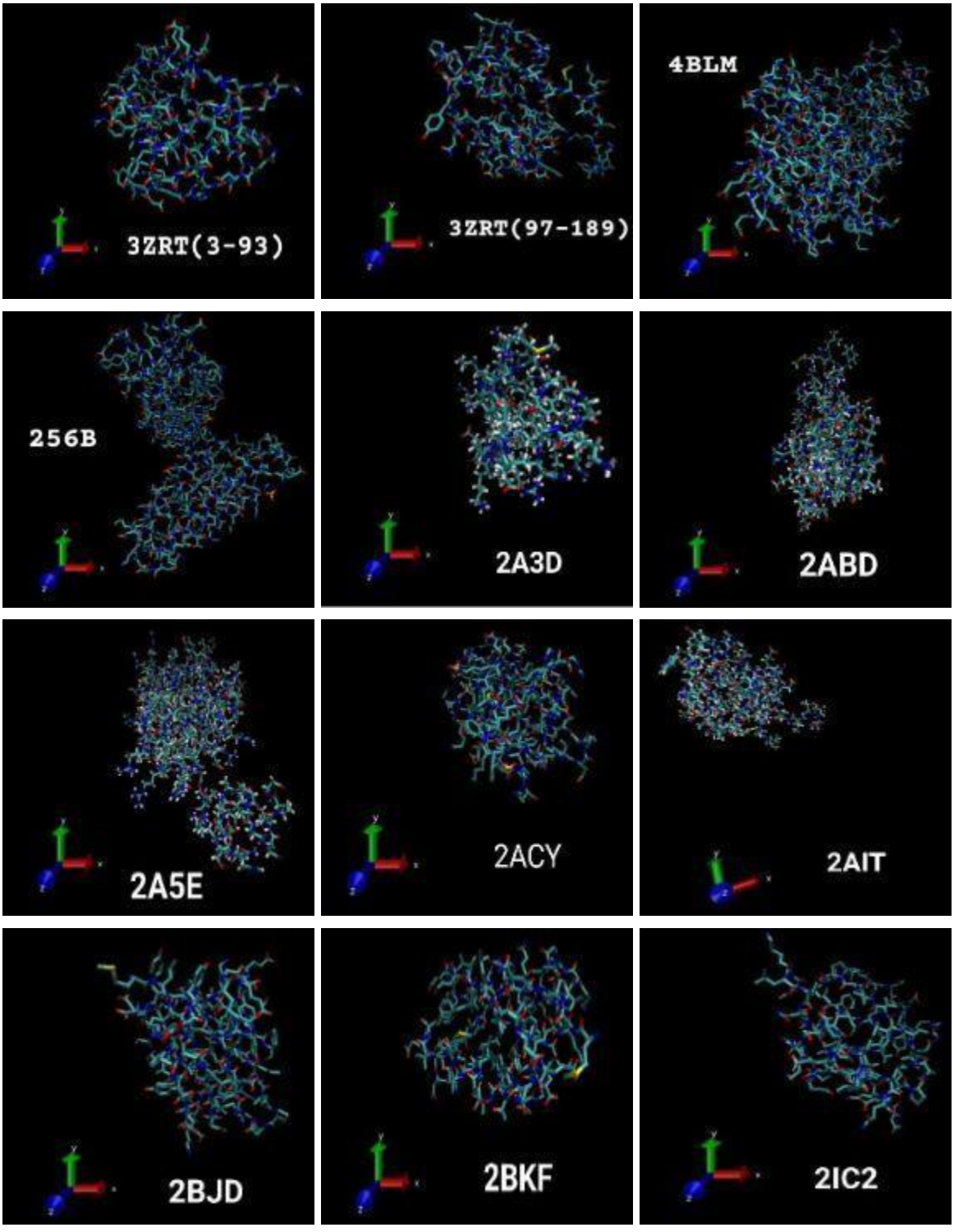

1M9S
1MJC
1N88
1NFI
1NTI
1NYF
106X
1OPA
1PBA
1PCA
1PGB
1PHP(1-175)

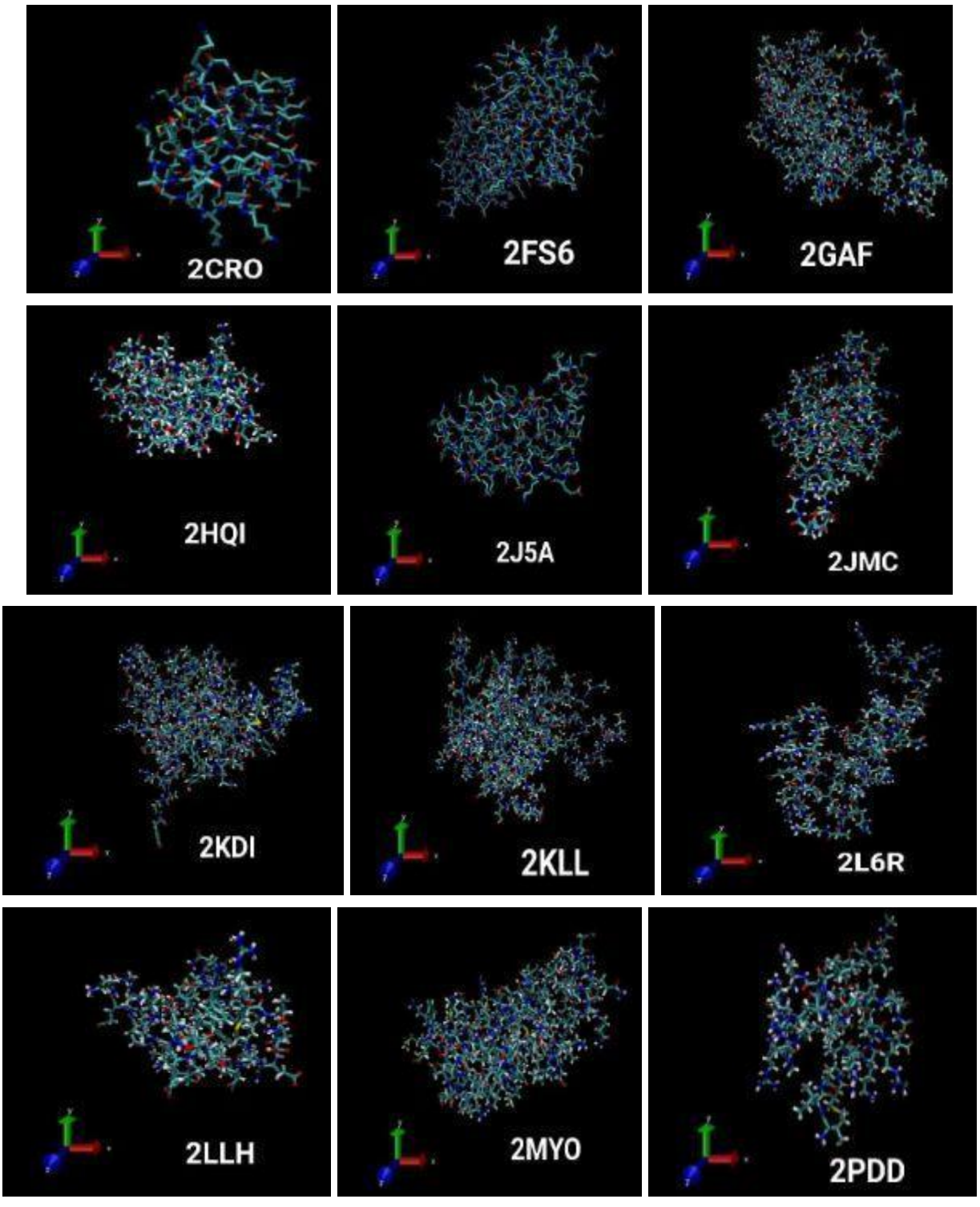

1PHP(186-394)
1PIN
1PKS
1PNJ
1POH
1PRB
1PRS(1-88)
1PRS(91-173)
1PUC
1QTU
1RA9
1RFA

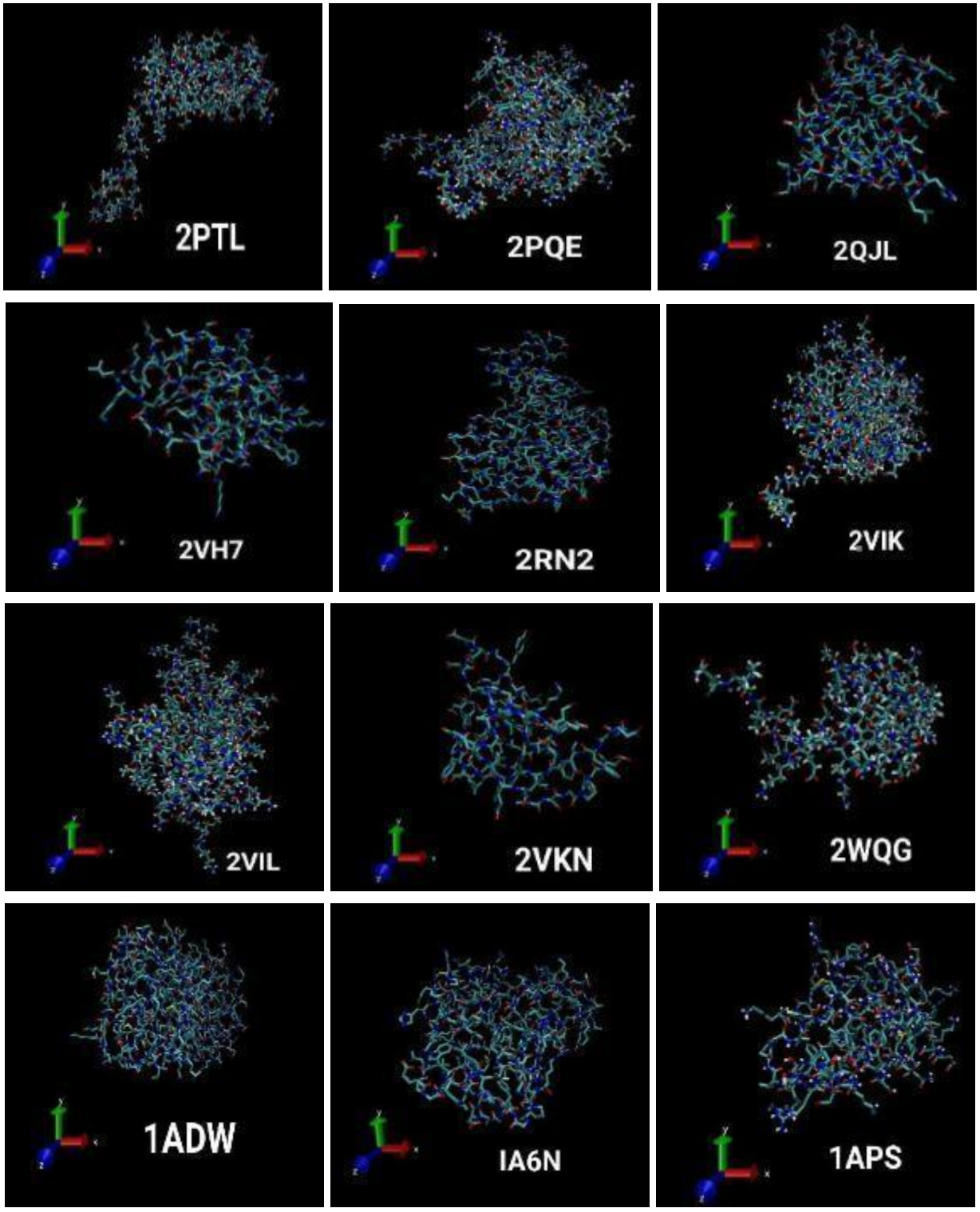

1RG8
1RIS
1RYK
1SHF
1SHG
1SRL
1SS1
1ST7
1TEN
1THF
1TIT
1TP3

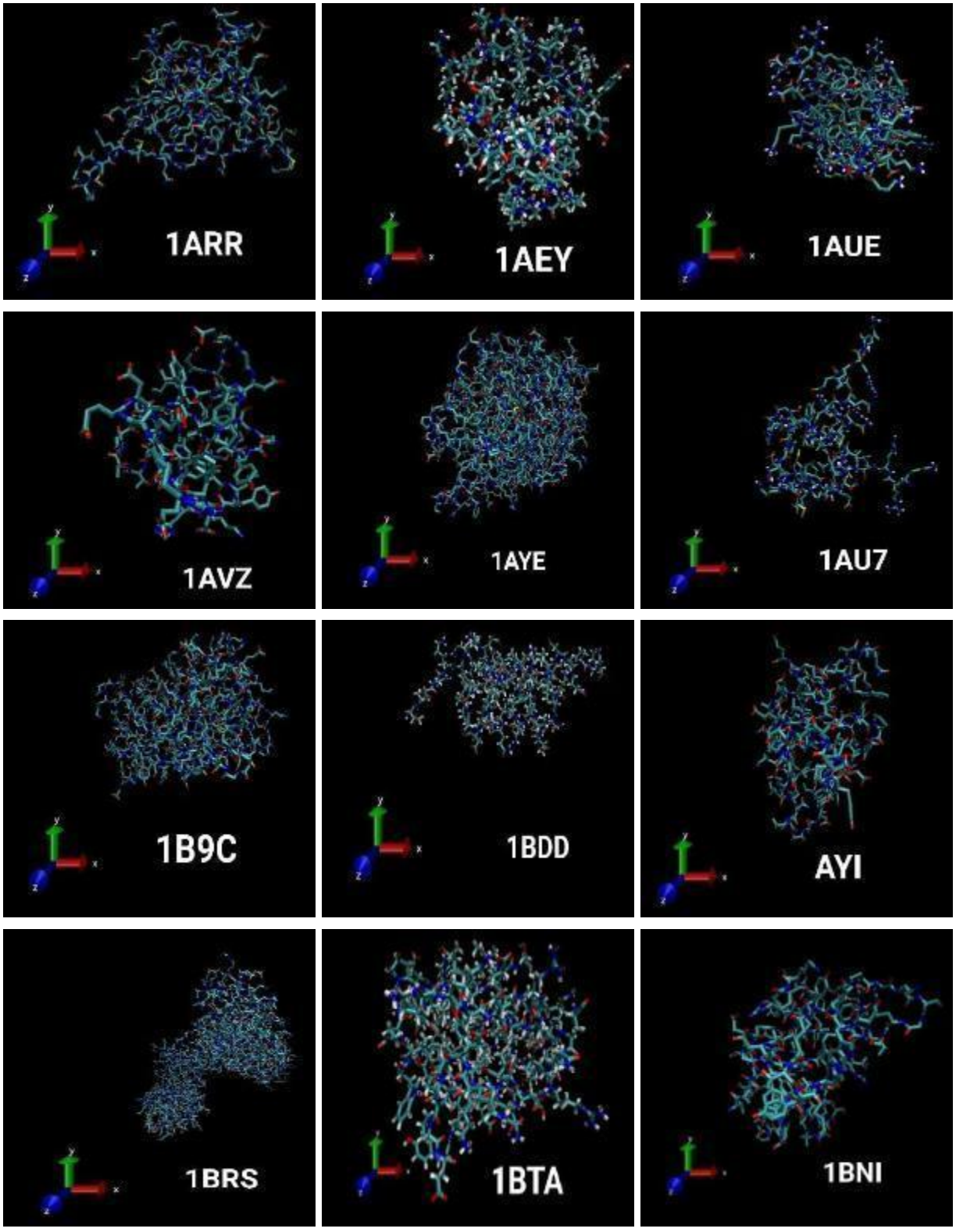

1TTG
1U4Q
1UBQ
1UCH
1URN
1UZC
1V9E
1VII
1VIK
1W4E
1W4J
1WIT

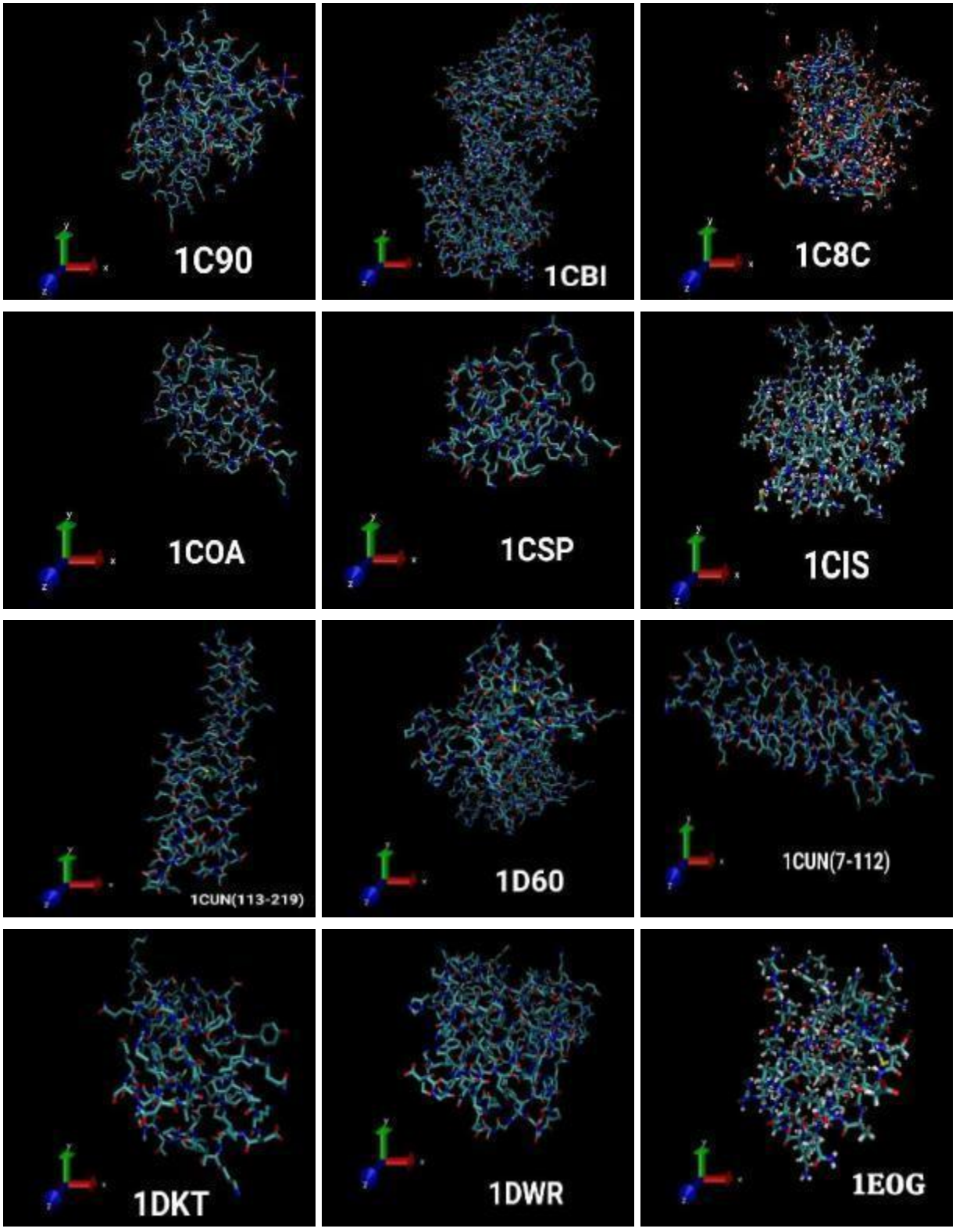

1WQ5
1YCC
1YYJ
3BLM
3CHY
3CI2
3F6R
3H08
3MEF
3O4B
3O49
3O4D

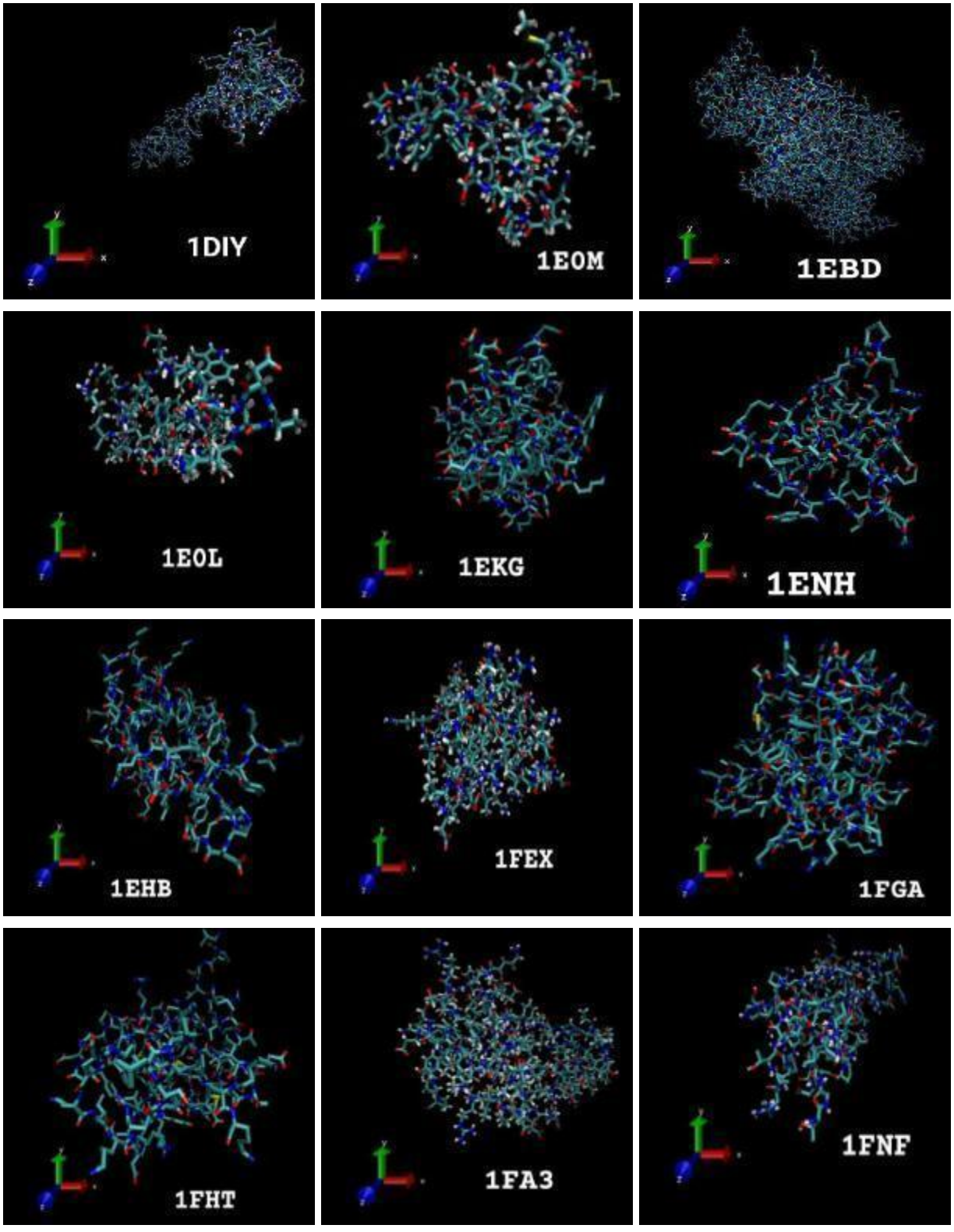

3ZRT(3-93)
3ZRT(97-189)
4BLM
256B
2A3D
2ABD
2A5E
2ACY
2AIT
2BJD
2BKF
2IC2

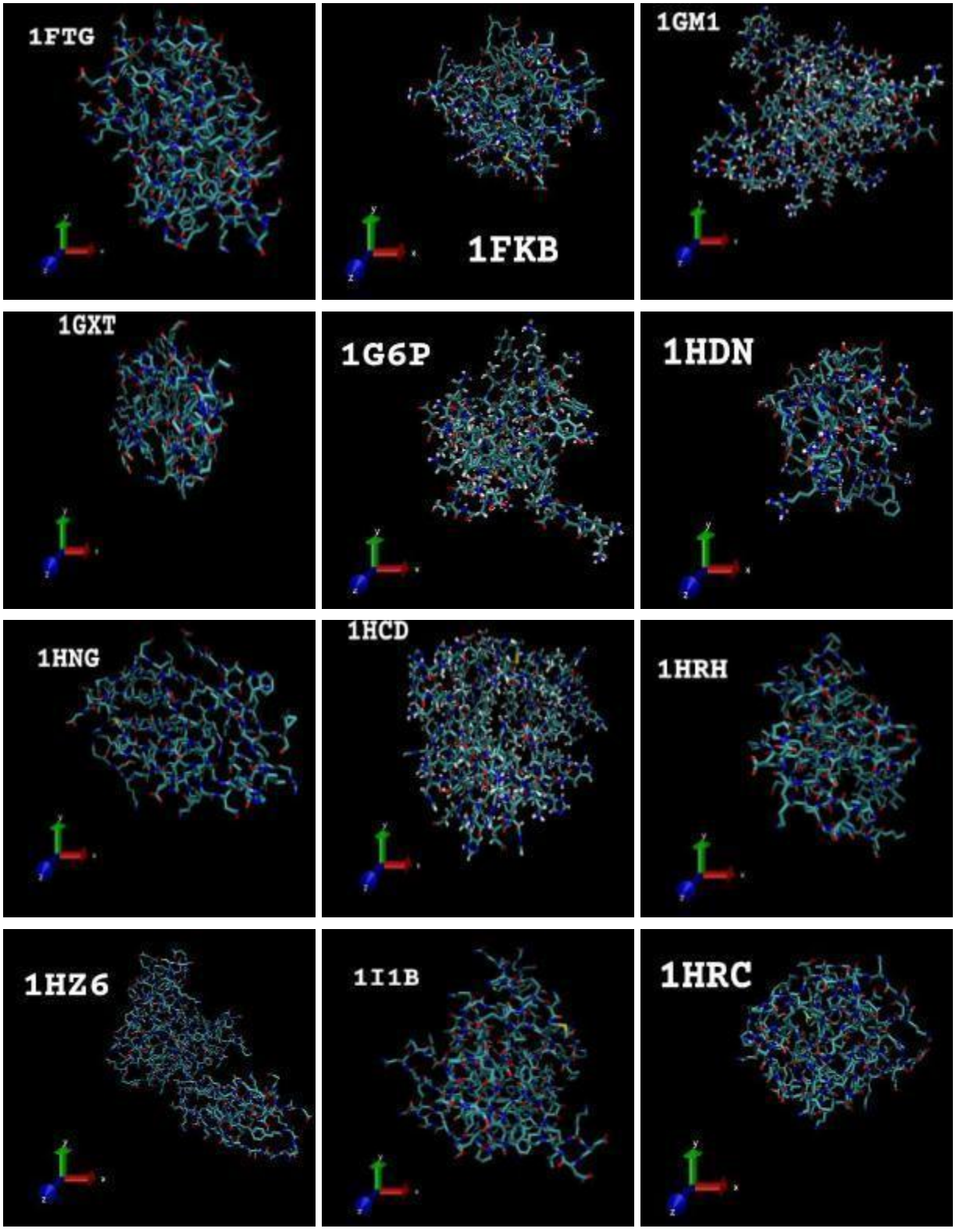

2CRO
2FS6
2GAF
2HQI
2J5A
2JMC
2KDI
2KLL
2L6R
2LLH
2MY0
2PDD


2PTL
2PQE
2QJL
2VH7
2RN2
2VIK
2VIL
2VKN
2WQG
1ADW
IA6N
1APS


1ARR
1AEY
1AUE
1AVZ
1AYE
1AU7
1B9C
1BDD
AYI
1BRS
1BTA
1BNI


1C90
1CBI
1C8C
1COA
1CSP
1CIS
1CUN(113-219)
1D60
1CUN(7-112)
1DKT
1DWR
1EOG


1DIY
1E0M
1EBD
1E0L
1EKG
1ENH
1EHB
1FEX
1FGA
1FHT
1FA3
1FNF


1FTG
1FKB
1GM1
1GXT
1G6P
1HDN
1HNG
1HCD
1HRH
1HZ6
1I1B
1HRC

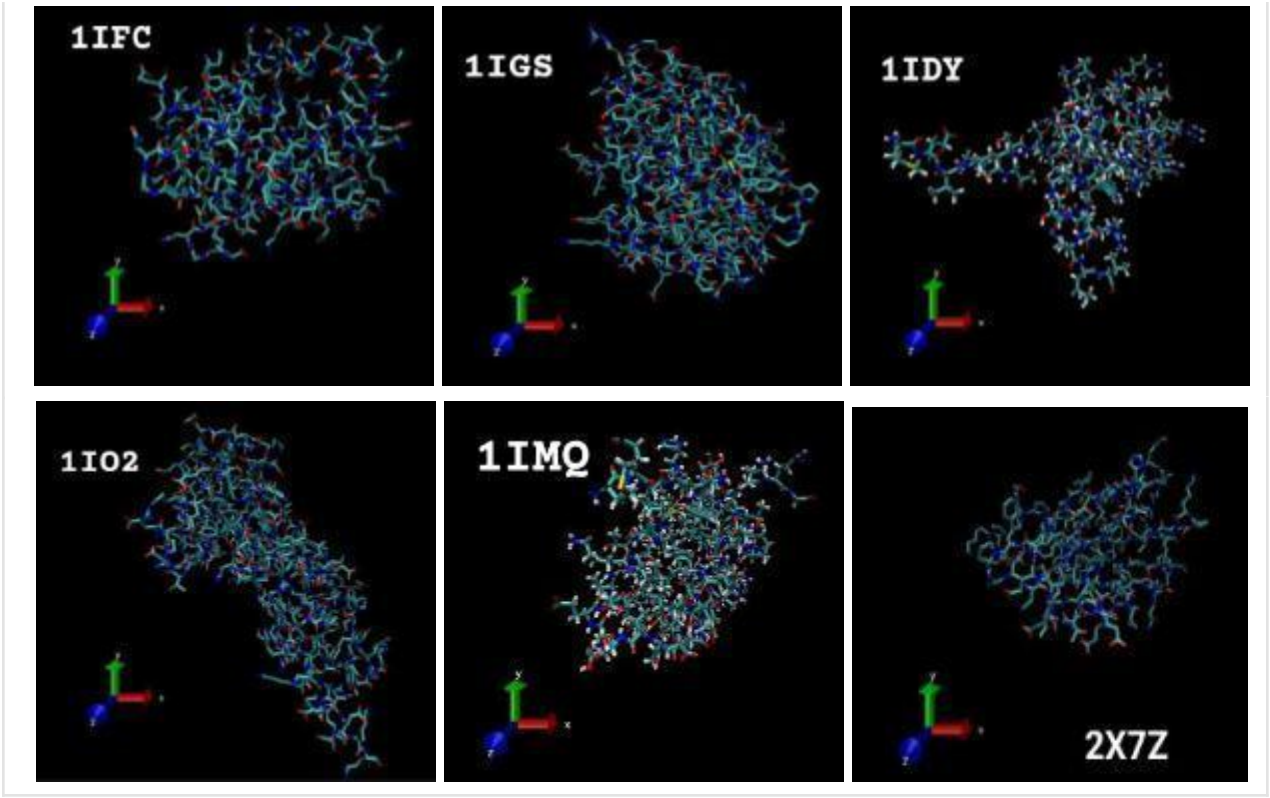


Fig.6. Snips of the pdb structures

2.2 Computational Requirements

The system requirements and configuration used for the inquiry are covered in this section. At IIIT Delhi, data processing was done with the open-source, Unix-based Linux operating system . Even though we preprocessed the data on Linux, using a Python compiler allowed the scripts to work on Windows as well. We needed an internet connection, a Linux operating system, and a computer system with administrative access in order to set up the Python environment. Neurocare at IIIT Delhi offered these prerequisites. Python is a straightforward and adaptable programming language that was originally made available in 1991. It was named after the British comedy group Monty Python. The goal of the developers' work was to produce a fun programming language.

We used the command line to install and configure Python after configuring the workstations. We upgraded the system with apt-get to make sure the Python version was current. After the upgrade was finished, we looked to see if Python 3 was installed on the system. We then used pip or the conda package to install all the required Python libraries and packages, including Pandas and Numpy. Furthermore, Linux provides the ability to create a Virtual Environment, which is a user-specified sub-environment inside the Linux desktop. More control over projects and the capacity to run several package versions on the same workstation are made possible by virtual

environments. Installing the venv module, which is a component of the Python 3 default library, first allows us to construct as many virtual environments as necessary. Next, we select the directory in which to create the Python environment. The activation command, which invokes the activate script, can be used to start the virtual environment after it has been generated and all prerequisites have been installed.

2.2.1 Jupyter Notebook Installation

An easy-to-use, web-based, open-source tool for producing and sharing documents with live code, equations, images, and narrative text is Jupyter Notebook. It is highly preferred in domains like scientific computing, machine learning, and data analysis because of its strong coding, data visualization, and documentation capabilities—all included in a single environment. The.ipynb extension makes it easier to save notebook files. Jupyter Notebook is especially well-liked by researchers because of its effective and potent data analysis and visualization capabilities, which also make cooperation easier. In the context of education, teachers use it to create self-contained, interactive modules that improve student engagement by allowing them to experience learning firsthand. It's simple to install Jupyter Notebook: use the pip command to download and install the Notebook and any required dependencies. To begin, launch it from the terminal. Making a new notepad is easy once it's opened. Press the 'New' button, select the preferred kernel (Python or R), and start working on your projects. You can use a variety of tools for data analysis and visualization, write code, and annotate using markdown.

2.2.2 Setting Up Python Environment

Depending on your needs, there are various ways to configure a Python environment on your computer.

Python distribution that is stand-alone: Perfect for general-purpose programming without the need for particular libraries. A package management (like pip), a basic library set, and an interpreter for Python are provided by solutions like Anaconda or Python.org.

Virtual Environment: Ideal for demands unique to a project, particularly when several projects call for various library versions. Project dependencies can be efficiently isolated with virtual environments made with tools like virtual env for Python's venv module.

Platforms for containerization: Applications and their dependencies can be contained within a container by platforms such as Docker, which allows the programme to run on any machine that has Docker installed. This allows for greater scalability and promotes sharing and cooperation. Pandas, NumPy, Matplotlib, and Seaborn were among the pre-processing and data visualisation technologies utilised in this project. A streamlined approach was ensured by installing these libraries using different commands in the same environment.

2.3 Tools for Modeling and Screening

2.3.1 PFEATURE

A computational software programme called Pfeature is essential for extracting a wide range of features from protein sequences. To carry out these extractions, it makes use of both the sequence-based and physiochemical features of proteins. This technology is essential to the discipline of bioinformatics, especially in the areas of computational biology and proteomics, where it greatly helps with the prediction of protein structure, function, and protein-protein interactions.[49]
The data was processed using the following command lines after installing Pfeature's standalone version:
To calculate all composition data features:
Python pfeature_comp.py -i protein.seq -o output_comp.csv -j ALLCOMP

2.3.2 AMBER

A key component of the AMBER software suite, AmberTools23 is designed for molecular dynamics simulations of biological molecules. With new tools and features including sophisticated force fields and molecular dynamics and energy reduction techniques, this version improves on its predecessor and offers more accurate simulation capabilities.

We utilized AmberTools23's implicit solvent model to effectively simulate complex biomolecular systems. By using this method, the computational burden of directly modeling solvent molecules is circumvented. Many implicit solvent models are supported by AmberTools23, including the updated Poisson-Boltzmann (PB) and Generalised Born (GB) models. We used a GB model from AmberTools23 that uses electrostatic descreening to approximate solvent effects.

This model was selected because it strikes a compromise between the precision of the electrostatic interactions between the solvent and the solute and computational efficiency. We chose the PB model for several simulations in order to solve the Poisson-Boltzmann equation and more precisely handle certain electrostatic interactions.

Parameters for Simulation The specific parameters used in these simulations are described in detail in the methodology

Field of Force: The simulations were performed with AmberTools23's ff14SBonlysc force field, which is renowned for its excellent predictions of protein backbone conformations.

Configuration for Simulation: We carefully configure the temperature control using Langevin dynamics, the treatment of non-bonded interactions, and the integration time step. Every simulation was run with constant pressure and temperature to replicate physiological circumstances.

The primary motivation for implementing an implicit solvent model was to minimize computational burden and optimize the system without significantly compromising the precision of solute dynamics.

Computational Efficiency: Larger systems or longer simulation timeframes are made possible by implicit solvent models, which significantly reduce the number of atoms in simulations.

System Simplification: The system's setup and subsequent analysis are made simpler by the removal of explicit water molecules, which keeps the focus on crucial biomolecular interactions.

Steps Taken

Environment Configuration: We set our installation directory for AmberTools23 as the value for the AMBERHOME environment variable. This configuration guarantees that every software suite component runs correctly and can access the necessary libraries and tools.

Getting the Protein Data Ready: Using pdb4amber, the protein data file was preprocessed to remove unnecessary atoms and molecules from the raw PDB file, leaving only the necessary protein structure for simulation.

Creating Simulation Input Files: To create the required coordinate files and topology, we used tleap. In order to correctly represent the physical and chemical properties of the protein, we loaded the protein structure, applied the ff14SBonlysc force field, and modified the system's radii to satisfy the needs of the implicit solvent model.[47], [48]

Hydrogen Mass Distribution Adjustment: To stabilize simulation dynamics, we repartitioned the mass of the hydrogen atoms, striking a balance between computing efficiency and dynamic correctness, particularly for high-frequency vibrations of hydrogen atoms.

System Energy Minimization: We carried out an energy minimization stage to guarantee stability and avoid physically implausible conformations. Unfavourable contacts, such as overlapping atoms or strained bonds, that were introduced during setup are lessened by this technique. In order to balance simulation accuracy with computing demands, we carefully chose the minimization parameters. In order to accurately describe the solvation effects and the ionic strength of the environment, the parameters for the implicit solvent model were also modified.

Heating and Equilibration: In order to avoid thermal shocks that could upset molecular conformations, we progressively raised the system's temperature from a low starting point (0 K) to the desired physiological temperature, which is approximately 310 K. This was set to run on a thermostat for a predetermined amount of time (e.g., Langevin dynamics). To guarantee a steady rise, we kept a careful eye on the system's temperature, pressure, and potential energy during the heating phase. After then, the system reached a state of equilibrium in which all of its macroscopic characteristics—temperature, pressure, and volume—fluctuated roughly about their mean values with little to no drift. Before going to production, this stage makes sure the system operates consistently under predetermined circumstances.

Production Run: To replicate the real-time dynamics of the protein in the implicit solvent environment, we started a production run after setting up the system and finishing the energy minimization. This stage is essential for tracking the behavior of the molecular system and acquiring information about its attributes and operations. After extracting every energy average from the production output file, we performed a study of the radius of gyration using cpptraj.[47]

Analysis of Molecular Dynamics Simulations CPPTRAJ

Radius of Gyration: This measure assesses how compact a protein or biomolecular structure is by computing the mass-weighted distribution of atomic positions around the centre of mass. Monitoring the radius of gyration during simulations makes it easier to evaluate folding behaviours and structural integrity under a variety of environmental conditions. CPPTRAJ uses trajectory data to perform this computation; instructions typically call for trajectory files, topology files, and an output destination.

Solvent-Accessible Surface Area (SASA): It is a crucial metric for assessing protein folding, stability, and solvent-protein interactions. It quantifies the surface area of a biomolecule that is exposed to a solvent.

Using geometric methodologies to estimate solvent exposure or the rolling probe technique, which simulates a water molecule rolling across a protein surface, CPPTRAJ determines SASA.

2.3.3 Visualization with VMD (Visual Molecular Dynamics)

3D Animation and Rendering: Using a variety of drawing styles and color schemes to distinguish between components such as atoms, residues, and secondary structures, VMD excels at rendering intricate 3D molecular structures. It has the ability to animate the paths of molecular dynamics, enabling users to interactively play, pause, and scrub through simulations in order to see how molecules evolve over time.

Visual Representations: With a wide range of graphical possibilities, VMD highlights protein secondary structures by displaying molecules in styles that range from basic lines to intricate ribbons and cartoons.

Interactive Analysis Tools

Salt Bridges: During simulations, VMD's visualisation tools can track the development and disruption of salt bridges, emphasising the residues involved and dynamically displaying these interactions.

Root Mean Square Deviation, or RMSD: It is a measure of molecule stability or conformational changes over time. VMD can compute and plot RMSD values along a trajectory.

Root Mean Square Fluctuation, or RMSF: It is a tool used in VMD computations to find molecular areas with high flexibility. It is similar to RMSD in that it uses color-coded representations to indicate different degrees of fluctuation.

Extensions & Customisation: With Tcl scripting, VMD offers a high degree of customisation that enables users to create custom analyses and visuals. Moreover, it combines biophysical data—such electrostatic potential maps—with structural data to improve our comprehension of molecular interactions like those found in salt bridges.

2.4 Data Preparation

In this investigation, we used a range of datasets and computational models:

Graph Signal Processing (GSP): We concentrated on proteins that show spatial proximity defined by Euclidean distances between residues obtained from 3D coordinates obtained from the Protein Data Bank (PDB).[42], [45]

Graph Models (RIG): A basic network model in which two residues are connected if their separation is less than a predetermined threshold, usually 5–9 Å.

By adding sequence-based inter-residue lengths as weights and emphasizing longer distances, RIGseq improves the RIGbinary model. [45]

Signal Representation on Graphs: We used the graph nodes (residues) to represent 48 physicochemical and thermodynamic parameters as well as aspects of the protein network as signals.[42], [44]

Graph Fourier Transform (GFT): This signal analysis technique is used. The eigen decomposition of the graph's Laplacian matrix yields the Fourier basis, which makes it easier to convert residue signals from the spatial to the frequency domain. Since we believed that the low-frequency components of the GFT capture important biophysical features, including protein folding rates, we concentrated on them. [42], [43], [45]

Building Dataframes: For every signal, thresholds were determined at which the highest association with protein fold rates was noted. After then, these signals were applied to every protein to produce an extensive dataframe.

Using Amber and PDBFixer: We used Python to clean PDB data, eliminating unnecessary molecules like waters and HETATOM.

The Python module PDBFixer was used to handle missing residues; it also makes the following possible:

- Elimination of undesirable heterogens (molecules other than proteins and nucleic acids)
- heavy atoms that are absent are added
- uniformity in protonation states
- loop areas and absent hydrogens are added
- Non-standard residue conversion, like changing selenomethionine into methionine

Feature Deletion: Following cleanup, we used cpptraj and VMD to extract a number of features from the production files, such as values for bonds, angles, dihedrals, non-bonded interactions, van der Waals forces, electrostatics, generalized Born approximation, and solvent accessible surface area. We also extracted total energy (Etot), kinetic energy (EKtot), potential energy (EPtot), and other features.

Dataframe Compilation: For all proteins, including those with single and multiple domains inside the 2s classification, we generated distinct averaged properties dataframes and GSP's signal dataframes, similar to the pfeature dataframes. The pdb we created with PDBFixer and AmberTools 23 were used for the Amber MD Simulation. The pdb underwent simulation and was then processed to produce all energy features, the radius of gyration, the number of salt bridges, and the rmsd and rmsf values.[47], [48],[49]

2.4.1 Data Preprocessing

Data cleaning: The first step in our data preparation process was to clean the dataframes by eliminating any missing values and any pertinent columns. This guarantees the accuracy and relevance of the data used in later analyses.

Data Normalization: We used two primary scaling techniques, each selected according to the unique needs of the data, to normalize the data:

Min-Max Scaling (Normalization): This technique modifies the features of the data to fit within a given range, usually [0,1] or occasionally [-1,1]. To calculate the range (difference between the maximum and minimum values of a feature), subtract the minimum value of the feature. This method is especially helpful for preserving the integrity of distances inside the feature space, which is essential for gradient descent-based algorithms like neural networks and distance-based algorithms like K-nearest-neighbors (KNN).

Xnorm = (Xmax−Xmin)/(X−Xmin) is the formula, where X is the original value, Xmin, Xmax, and Xmax are the minimum and maximum values from the feature column, respectively, and Xnorm is the normalized value.

Standardization: Also known as standard scaling, is a scaling technique that aligns the data distribution with a standard normal distribution by transforming the features into a mean of zero and a standard deviation of one. For techniques that assume a normal distribution or optimization algorithms like Principal Component Analysis (PCA) and Support Vector Machines (SVM), where feature variance is important, this standardization is essential.

The formula for Xstd is **Xstd** = (X−μ)/σ, where X is the original value, μ is the feature column mean, σ is the feature column standard deviation, and Xstd is the standardized value. These preprocessing processes are crucial for accurately and efficiently analyzing the data in computational biology applications such as molecular dynamics.

2.4.2 Transformation of Features

Logarithmic transformation: this mathematical operation modifies the distribution of numbers and is especially helpful in machine learning, statistics, and data analysis, particularly in cases where the variables span multiple orders of magnitude or the data is skewed. These are the main justifications and techniques for using logarithmic transformation:

Reducing skewness: Cut DownThe majority of data points in real-world datasets are right-skewed, which means that a long tail extends upward and most clusters at the lower end of the scale. By compressing higher values more than lower ones, the logarithmic transformation helps to normalize the distribution and lessen skewness. Many statistical methods and machine learning models that assume data normalcy are consistent with this symmetry.

Stabilizing Variance: A logarithmic transformation can assist in stabilizing a dataset whose variance rises with a mean condition referred to as heteroscedasticity. The efficacy of several statistical tests that assume constant variance across data ranges depends on this stabilization.

Converting Multiplicative Relationships into Additive: Linear regression is a statistical model that assumes linear relationships between variables, implying additive changes between them. By converting multiplicative relationships (such as exponential growth) within data into additive ones, logarithmic transformation makes modeling with linear techniques easier.

Increasing Interpretability of Coefficients: Logarithmically transforming variables improves the interpretability of model coefficients in econometrics and regression analysis. For example, the coefficients represent elasticities in log-log models (where both dependent and independent variables are log-transformed); that is, a 1% change in one variable corresponds to a percentage change in another, as indicated by the coefficient.

The logarithmic transformation formula : Value Transformed = Logb(x), where x is the initial data value, and b is the base of the logarithm is typically 10, e, or 2. This transformation is an essential step in getting data ready for analyses that make use of linear relationships and normalized distributions.

2.4.3 Feature Extraction Methods

Feature extraction is essential for converting unprocessed data into useful, effective features that affect how well machine learning models perform. Here are a few often used techniques:

PCA, or principal component analysis: By converting original variables into a new set of variables (called principal components), which are a linear combination of the original ones, PCA is a technique for reducing dimensionality. These parts are ranked according to the amount of variation they extract from the data and are orthogonal. In polynomial feature extraction, the feature space is expanded by generating higher-degree terms and interactions between existing features. The method entails computing the covariance matrix of the data, extracting its eigenvalues and eigenvectors, and utilizing these to build the principal components. It works especially well at capturing intricate, non-linear relationships between features and the target variable that traditional line models might miss. The inclusion of terms like squares and products of features (e.g., x12, x22, x1×x2) allows models to fit detailed patterns in the data more

accurately. But because there are more features, this expansion may result in problems like model overfitting, especially when there is little data, and higher computational costs.

Recursive feature elimination, or RFE: It is a powerful feature selection technique that improves a model's simplicity and effectiveness by gradually building a model and eliminating its least important features in accordance with how well it performs. This method concentrates on keeping the most advantageous aspects, which aids in avoiding overfitting and expedites model training and prediction.

2.4.4 Feature Selection Methods

In order to reduce computational complexity, improve model performance, and identify the most relevant features for use in a model, feature selection is essential. Here are a few specific methods:

Filter Methods: These approaches evaluate the relationship between the target variable and the input variable using statistical techniques. Commonly used metrics to assess the significance of data include correlation coefficients, chi-square scores, and mutual information scores.nce of features.

Wrapper Methods: These techniques use many models, each utilizing a different subset of features, and choose the combination of features that yields the best-performing model. These techniques need several rounds of model training, which makes them computationally demanding.

Embedded Methods: These techniques optimize both feature selection and model training simultaneously by incorporating feature selection into the training process. Methods such as Lasso and Elastic Regression model coefficients are normalized by net use regularization, which effectively reduces some coefficients to zero to eliminate some features.

2.4.5 Model Selection

In order to successfully capture complicated patterns in the data, we combined individual and ensemble methods for creating our prediction models. Due to their reputation for being reliable and adaptable to a wide range of data formats, RandomForestRegressor, XGBRegressor, Gradient Boosting, and SVM were selected as the techniques to use. We also used a StackingRegressor, which increases model generalisation and lowers the chance of overfitting by utilising a variety of base estimators.

Detailed Overview of Selected Models

Random Forest: The 'forest' formed by several decision trees, each trained on different random subsets of the data and features, is an approach that improves on the simplicity of decision trees. Every tree votes during the prediction phase, with the final prediction being determined by the majority vote (for classification) or the average vote (for regression). By reducing the variance of individual trees, this ensemble method efficiently combats overfitting. Because of its versatility, Random Forest can handle jobs involving both regression and classification with no need for hyperparameter modification.

Gradient Boosting: Gradient Boosting is another strong ensemble strategy that develops models step-by-step, emphasising the correction of previous models' errors in each new model. Usually, it uses decision trees as base learners and improves them by controlling model complexity through the optimisation of a loss function and a regularisation term.

eXtreme Gradient Boosting, or XGBoost: A sophisticated gradient boosting solution, XGBoost is built for both efficiency and adaptability. It can handle a variety of regularisation strategies to prevent overfitting and is extremely scalable, making it appropriate for a broad spectrum of data science problems. Moreover, XGBoost has functions for managing missing data and pruning trees.

Support Vector Regression (SVR): SVR is a regression technique that applies the ideas of Support Vector Machines to regression scenarios. Its goal is to discover a function that, within a given threshold, stays as flat as feasible for improved generalisation but keeps as close as possible to the actual outputs. Unlike regular SVM, SVR optimises for all deviations from the margin and may adjust to a range of data distributions by using different kernel functions.

Stacked generalisation, or stacking: A two-level method is used in this meta-ensemble technique: a meta-model trained on the outputs of the base models is used at the second level, while many base models trained independently make up the first level. By utilising the advantages of each model, stacking often produces simpler or more effective results than any one model alone.

Voting Regressor: An ensemble technique to improve overall prediction accuracy and stability by combining the predictions of numerous regression models. This technique successfully leverages the various capabilities of the associated models to achieve higher performance by reducing prediction variance and increasing robustness by averaging or weighting the outputs of several base regressors.

These carefully chosen techniques offer a thorough approach to solving problems related to predictive modelling, guaranteeing that the resulting ensemble is resilient against many types of data anomalies and modelling hazards in addition to being accurate.

2.4.6 Model Training

A significant approach in machine learning is the train-test split, which divides the dataset into two parts: one for testing the model's performance (20–30%) and the other for training the model (usually 70–80% of the data). This segment is essential for evaluating a model's ability to generalise to fresh, untested data and aids in preventing overfitting, a phenomenon in which a model performs remarkably well on training data but badly on untested data. The split needs to be randomised in order to guarantee fairness and randomness as well as to remove any potential bias brought on by the data's ordering. Simple functions like train_test_split, available in Python's Scikit-Learn module, let you choose the percentage of data to be utilised for testing (test_size) and guarantee repeatability (random_state). The predictive capacity and resilience of the model can be assessed by training it on a sizable section of the dataset and testing it on a different set, allowing for any necessary improvements to be made prior to deployment. This method reduces the chance of overfitting and helps optimise the model by helping to identify the optimal model among several contenders. The resulting model is therefore more appropriate for real-world use.[1], [3], [5]

2.4.7 Hyperparameter Tuning

A vital step in the creation of machine learning models is hyperparameter tweaking. It entails determining the ideal values for the hyperparameters that control learning and specify the general behaviour of the model. In contrast to model parameters, which are determined by training, hyperparameters preset and control variables like model complexity, convergence speed, and variance to bias ratio. Typical techniques for adjusting hyperparameters consist of:

Grid Search: Looks through all potential combinations of hyperparameters in detail.

Random Search: To investigate a wider range of options, hyperparameter values are sampled at random.

Bayesian Optimisation: Iteratively refines the search based on previous findings, using a probabilistic model to forecast how various hyperparameter settings would perform.

The methods and effectiveness of these approaches differ. For example, Grid Search is quite comprehensive, but it might be computationally expensive and not the best option for larger datasets or more complicated models. Even though it is less methodical, random search can find suitable parameters fast, particularly in high-dimensional domains. Because it attempts to reduce the number of assessments by learning from past results, Bayesian optimisation is especially useful in situations when configuration evaluations are expensive. More complex techniques are also employed, such as gradient-based optimisation and evolutionary algorithms, which simulate evolutionary processes and use gradient descent to optimise hyperparameters, respectively.

It emphasises the necessity for a strategic approach to improve model performance without incurring excessive computational costs. The choice of hyperparameter tuning method frequently depends on the particular model, the type of data, and the available computer resources. In order to maintain robustness and avoid overfitting, cross-validation is frequently used. This emphasises how difficult and crucial it is to handle hyperparameters well in machine learning workflows.

2.4.8 Model Validation

A critical stage in the construction of a model is validation, which makes sure that the model's performance is fairly evaluated in comparison to unobserved data. The data is divided into training (usually 70–80%) and testing (20–30%) sets during this phase. For our models, an 80-20 split was employed in order to preserve objectivity while assessing the model's capacity for generalization.

K-Fold Cross-Validation

This technique divides the data into 'k' equal-sized folds or segments, improving the basic holdout method. Using the remaining 'k-1' folds as the training set and cycling through each fold as the test set, the model is trained and tested 'k' times. By using this method, every data point may be validated exactly once, giving a reliable indicator of the model's efficacy across a range of data subsets.The average of the performances over all 'k' trials is usually used to assess the final model's performance.[5]

2.4.9 Insights from Learning Curves

Plotting the training size or iterations versus the performance on both the training and validation sets is done graphically using learning curves. These curves are very useful for identifying problems with the behaviour of the model, such as overfitting or underfitting, and indicating areas where the model's performance could be enhanced by more training data. For example, deliberate feature selection and regularisation reduced the initial overfitting seen in the GSP signal dataframes of 52 proteins, increasing the model's robustness and consistency in both training and validation sets.

2.4.10 Actual vs. Predicted Graphs

In machine learning, actual vs. projected graphs are vital tools for comparing model predictions with actual results visually. These graphs, which are commonly presented as scatter plots with actual values on the Y-axis and anticipated values on the X-axis, are essential for assessing the correctness of the model. Perfect predictions from an ideal model would be closely aligned with a diagonal line. Prediction mistakes are indicated by deviations from this line, potential bias or variance problems are indicated by the spread of points, and anomalies or places where the model is inadequate are highlighted by outliers. Data scientists can evaluate several models, fine-tune the behaviour of the model, and identify particular performance enhancements that the model requires by examining these graphs.

When combined, these validation techniques provide an exhaustive and efficient evaluation of the accuracy and reliability of the model, directing modifications and enhancements prior to the model's finalization for use in real-world scenarios.

2.4.11 Model Refinement and Final Evaluation

Further model refining can be required after assessing the initial model performance and finishing hyperparameter adjustment. This could entail making a number of changes, like changing the model architecture, choosing different features, or going over and refining the preprocessing stages to better prepare the data. Because model refining is by its very nature an iterative process, it frequently takes several iterations of tuning and validation to achieve the required performance levels.

After the model has been suitably adjusted and improved, a last assessment is carried out. A distinct validation set that hasn't been used for any part of the training or tuning phases should ideally be used for this evaluation. By ensuring that the model performs effectively, robustly, and consistently across several data samples, this procedure validates that the model can generalise

successfully to new, untested data. Confirming the model's applicability and dependability in real-world circumstances requires this last step.

CHAPTER 3

RESULTS & DISCUSSION

3.1 1D DESCRIPTOR

We normalized the data of 48 amino acid properties that included physicochemical, thermodynamic and structure based property values and then we averaged these for all the proteins, only 2s proteins with both single and multidomain and for 2s proteins with only single domain to compare the results.

3.1.1 Averaged 48 properties

All proteins dataset

It has 51 columns total, of which 48 are the attributes, 1 is the goal variable (logarithmic protein fold-rates), or ln(kf), and the remaining 2 are PDB IDs and sequences. contains 146 rows for every characteristic. The dataset under analysis included several characteristics linked to proteins, such as hydrophobicity, intrinsic pH, and molecular weight.

Many important features with moderate correlations with ln(kf) were found and selected as a result of the correlation threshold being adjusted. ASAN (0.336) and Nm (0.304), two positively correlated features, and Pß (-0.305), Et (-0.317),ΔCph (-0.336), Pr (-0.344), Ns (-0.367), Hnc (-0.368), Rf (-0.370), El (-0.387), and Nl (-0.403), two negatively correlated characteristics were selected because to their capacity to affect protein folding rates via intricate, non-linear connections that are not sufficiently represented by straightforward linear models. With these features, the first attempt at a Linear Regression model produced an R-squared (R^2) of -0.40 and a Mean Squared Error (MSE) of 15.92, highlighting the model's inefficiency as it performed worse than a straightforward horizontal mean line.This led to more research into the use of non-linear models to better handle the dataset's complexities. However, the Decision Tree Regressor did not perform well, most likely because of overfitting, producing an MSE of 23.69 and a R^2 of -1.09. With an MSE of 11.03 and a marginal R2 of 0.03, a second Random Forest Regressor indicated modest progress, but it was still unable to adequately predict the data. The Elevation of an MSE 12.64 and an R2 of -0.11 indicated that the Boosting Regressor had difficulties as well, pointing to possible problems with feature selection and emphasizing the need for additional parameter adjustment to maximize model performance.

2s proteins with both multiple and single domain dataset

The target variable, ln(kf) (logarithmic protein fold-rates), is contained in one of the 51 columns (which contain 48 attributes), while the other two contain PDB IDs, sequences, and 100 rows for each protein. The first step in data exploration was evaluating important variables such as ln(kf), which showed a complicated, skewed distribution suggestive of complex links in the data. Thorough statistical research revealed the fundamental distributions and structure of these attributes, emphasising their variability and central tendencies.

Numerous features with significant positive and negative correlations to ln(kf) over ±0.3 were found using correlation analysis. Features such as Pa (0.45), a (0.38), and ASAN (0.37) showed positive correlations, indicating that rising values of these variables correlate with rising values of ln(kf), possibly pointing to factors that speed up protein folding. On the other hand, features like Nl (-0.44), El (-0.41), and Hnc (-0.39) showed significant negative correlations, suggesting that a drop in these variables corresponds to a greater ln(kf). This could suggest that there are suppressive or inhibitory effects on folding rates.

The current investigation employed a rigorous feature selection process to identify attributes that demonstrated moderate to strong correlations with ln(kf). The objective of this approach was to optimize model performance through the management of dataset dimensionality. These particular features were used in the Linear Regression model that started the regression analysis. The model's failure to capture the intricate relationships underlying the data was evident from the model's negative R-squared (R^2) of -0.40 and its Mean Squared Error (MSE) of 15.92. In order to handle these difficulties, the analysis moved to more complex techniques and investigated a number of non-linear models, including a Decision Tree Regressor, which produced an MSE of 23.69 and an R2 of -1.09, indicating overfitting; an MSE of 11.03 and a R^2 of 0.03, indicating a slightly positive fit; and a Gradient Boosting Regressor, which also encountered difficulties, as shown by an MSE of 12.64 and a R^2 of -0.11, indicating possible problems with feature selection or the requirement for additional parameter tuning. Furthermore, with an MSE of 11.72 and an R2 of 0.204, the Lasso and Elastic Net regression models—both with an ideal alpha of 0.1—showed enhanced prediction accuracy. The two models exhibit consistent performance, indicating their efficacy in mitigating feature complexity and improving interpretability. This validates the significance of the chosen features in predicting ln(kf).

2s proteins single domain dataset

52 samples and 51 unique features make up the dataset; 48 of the features are characteristics, one column has the goal variable, ln(kf) (logarithmic protein fold-rates), and the remaining two comprise PDB IDs and sequences. 'Hnc' showed a clear negative correlation of -0.343,

confirming an inverse association with 'lnkf', whilst 'Nm' showed a positive correlation of 0.482, indicating a strong direct relationship with 'lnkf'.

With 'lnkf' showing a mean of 5.505 and a range from 0.180 to 12.200, descriptive statistics demonstrated the dataset's heterogeneity. This suggests significant variation in the protein folding speeds among the samples. Further information about the asymmetry of the data came from skewness analysis. For example, 'Hnc' displayed a large left skew of -1.744, indicating a concentration of values at the top end of the range.

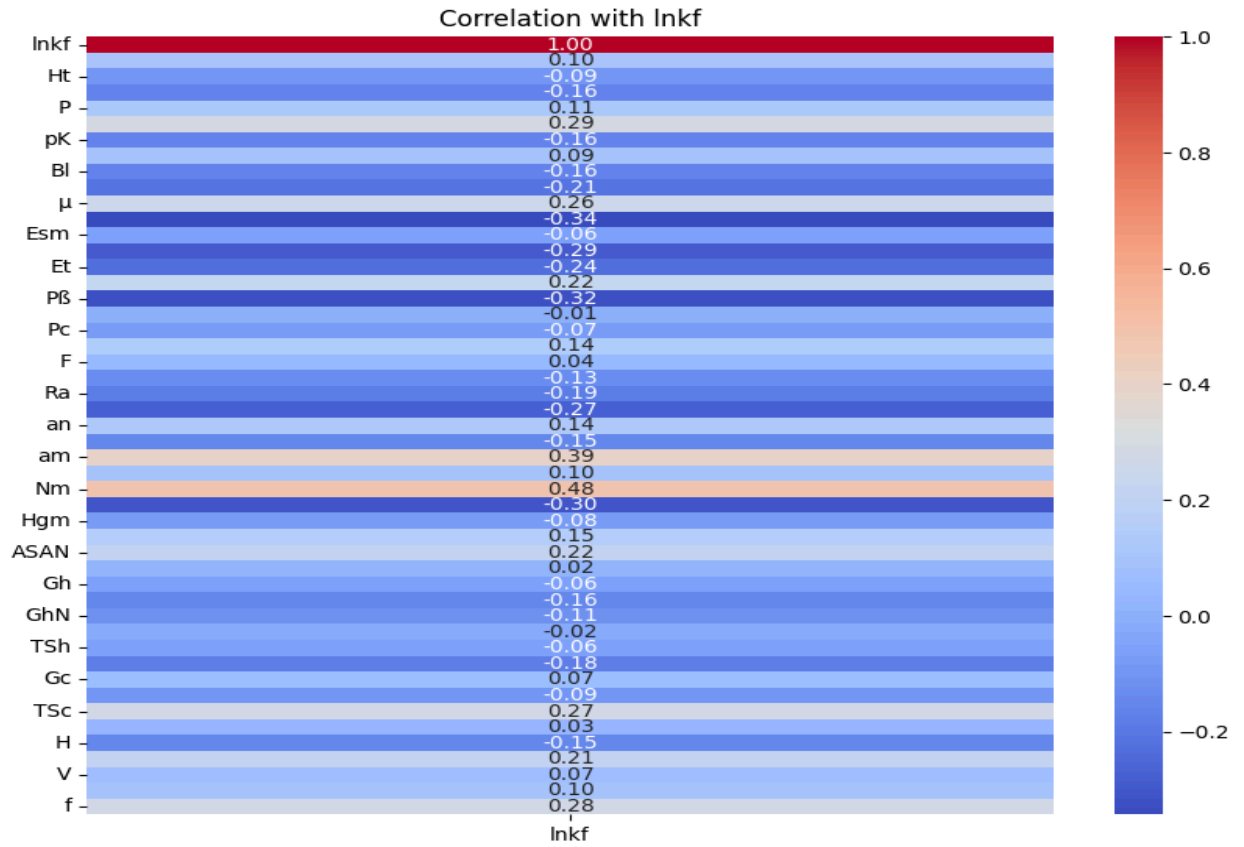


Fig.7.A heatmap showing relationships between several variables and the variable "lnkf" is shown in the first image. The correlation between each variable and "lnkf"—whether positive or negative—is displayed in this heatmap. An ideal positive correlation is denoted by a correlation value of 1, an ideal negative correlation by a correlation value of -1, and no correlation by a correlation value of 0. stronger positive correlations are shown by colours that are closer to red, and stronger negative correlations are shown by colours that are closer to blue. For instance, "Nm" and "lnkf" have a fairly positive correlation of 0.48, indicating that "lnkf" tends to increase along with "Nm."

A thorough understanding of the prediction power of each of the five regression models was obtained through their examination. These features—"Nm," "am," "Nl," "Pß," and "Hnc"—were obtained after feature selection and utilised to construct the regression models. With the lowest

Mean Squared Error (MSE) of 4.646 and the highest R-squared (R^2) value of 0.677, the Gradient Boosting Regressor was found to be the best effective model; this means it could account for roughly 67.7% of the variation in 'lnkf'. Strong validation scores, such as a Mean Absolute Error (MAE) of 1.670 and a Root Mean Squared Error (RMSE) of 2.155, which highlight the model's accuracy and dependability in estimating the kinetic folding rates of proteins, further supported the model's supremacy. While the training data MAE score is 0.06202949329347448 , MSE score is 0.005756236040101326 , RMSE score is 0.07586986252855166 and R2 score is 0.9992804952881523 .These results demonstrate the potential utility of the Gradient Boosting Regressor for research purposes and investigations also in predicting 'lnkf' values in similar datasets later.

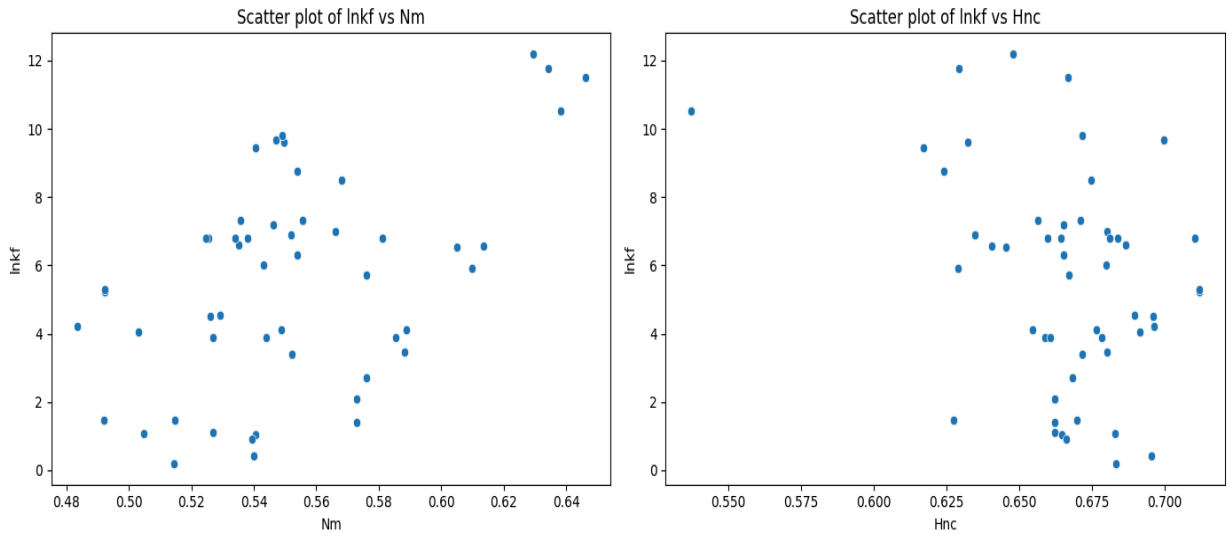


Fig.8. The association between "lnkf" and "Nm" and "lnkf" and "Hnc" is displayed in two scatter plots. The correlation values displayed in the heatmap are visually confirmed by these plots, where "Nm" exhibits a tighter, more defined upward trend with "lnkf" as opposed to the more diffuse association with "Hnc".

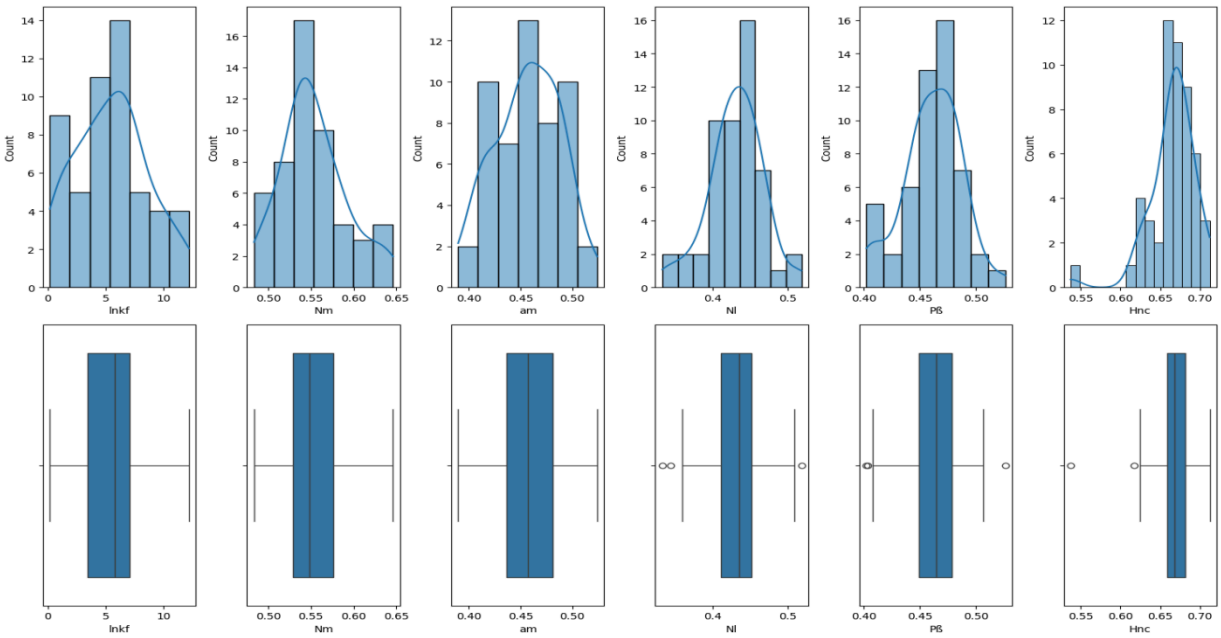


Fig.9. The distributions and summaries of "lnkf" and a number of additional variables ("Nm", "am", "Nl", "Pß", "Hnc") are shown in the last set of histograms and box plots. While box plots summarize the data by displaying the median, quartiles, and any outliers, histograms display the data's distribution across a range of values.

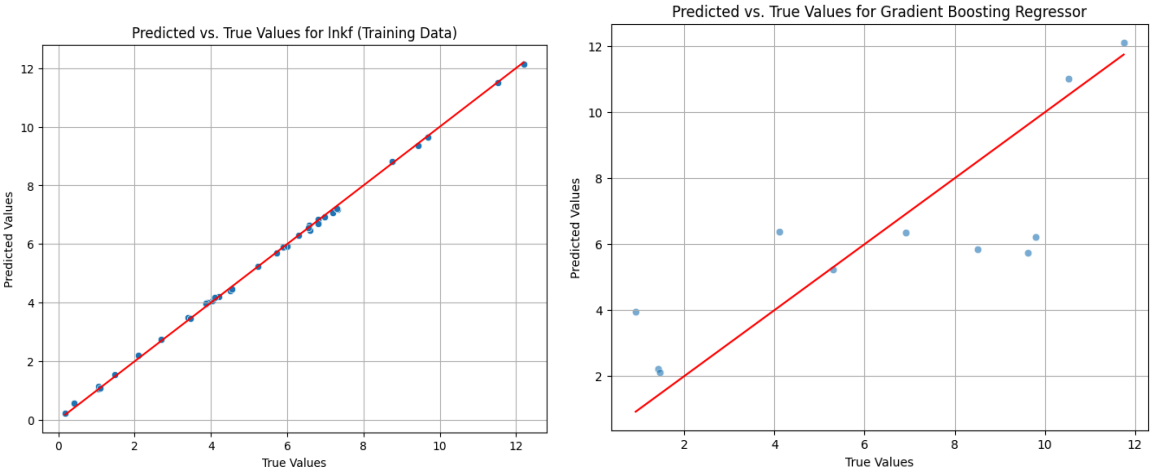


Fig.10. Plot contrasts the model's predicted values with the actual values. Points should ideally fall on the red line, which symbolises ideal forecasts. The model's predictions are more accurate the closer the points are to this line. Given that the majority of the points are closest to the line, the Gradient Boosting Regressor seems to work the best. (a) is for training data (b) for test data

3.1.2 pfeature Composition datasets

All proteins dataset

The characteristics that were used to identify the best correlations with 'ln(kf)' include a variety of factors, each of which may have an impact on comprehending the kinetic factor's ('ln(kf)') natural logarithm. The characteristic 'PCP_Z1' had a moderate positive association with 'ln(kf)', as indicated by its correlation coefficient of 0.336730. Correlations between 'DPC1_RR', 'TPC_TVE', 'TPC_ADN', 'TPC_GTG', and 'DPC1_NA' were also observed; these indicated varied degrees of positive correlation with the target variable. Positive associations were also shown by other features, including 'APAAC1_K', 'PAAC1_K', 'AAC_K', 'DPC1_NL', 'TPC_RRL', 'TPC_LND', 'TPC_WAE', 'TPC_QQN', 'TPC_QAP', 'TPC_PNL', 'APAAC1_A', and 'AAC_A'. Together, these characteristics help to clarify how the kinetic factor's natural logarithm and the predictors relate to one another. Using these associated features, prediction algorithms may be able to provide more accurate estimates for target 'ln(kf)' also gives insight into influential mechanisms behind kinetic factors.

The regression study included a number of models, all of which were evaluated using the Mean Squared Error (MSE) and R-squared Score metrics. Comparable findings were obtained using Linear Regression, Ridge Regression, and Lasso Regression; MSE values ranged from 10.8278 to 10.8738, and R-squared scores were between 0.0419 and 0.0459. The efficiency of these models in capturing the variance in the data was found to be restricted. On the other hand, the linear-based models were surpassed by Random Forest Regression and Gradient Boosting Regression, which produced higher R-squared scores and lower MSE values. When measured against other models, Random Forest Regression has the lowest mean square error (MSE) of 8.5591 and the highest R-squared score of 0.2458. Moreover, gradient boosting regression showed encouraging results, with an R-squared value of 0.2319 and an MSE of 8.7170.

A grid search utilising 108 potential hyperparameter combinations was carried out during the model fitting procedure, yielding a total of 324 fits. With the following parameters, the best-performing model was found:'max_depth' of 10,'min_samples_leaf' of 2,'min_samples_split' of 2, and 'n_estimators' of 300. The mean squared error (MSE) for this arrangement was negative, coming in at -12.3807. 324 fits were obtained by examining 108 potential hyperparameter combinations during the model fitting procedure. With a 'learning_rate' of 0.01;'max_depth' of 3,'min_samples_leaf' of 2,'min_samples_split' of 2, and 'n_estimators' of 300, the best combination was found. A negative mean squared error (MSE) of -11.2385 was obtained with this arrangement.

Additional analysis revealed a consistent training process, indicating well-tuned parameters, and a drop in validation loss, indicating the model's capacity to generalise well on fresh data. Although the outcomes were encouraging, there were still areas that might be improved by

adjusting hyperparameters, experimenting with more layers or alternative designs, and lengthening the training period. A focus on deployment readiness was also emphasised, underscoring the need of guaranteeing model correctness and efficiency in a production setting.

2s proteins with both multiple and single domain dataset

The training scores provide important information about how well each regression model is performing. Gradient Boosting had a greater capacity to fit the training data when compared to Random Forest and Support Vector Regression (SVR), with the lowest Root Mean Squared Error (RMSE) at 2.8917 and the highest R-squared value of 0.4322. With an R-squared value of 0.3499 and an RMSE of 3.0941, Random Forest trailed closely behind Gradient Boosting, showing respectable performance but somewhat higher error and less explained variation. On the other hand, SVR's low R-squared value of 0.1550 and highest RMSE of 3.5276 indicate that it was not very successful in identifying the underlying patterns in the training set.

Gradient Boosting performed somewhat better in individual model testing, with an RMSE of 2.931 and a R^2 of 0.416, compared to Random Forest's RMSE of 3.047 and R^2 of 0.370. Conversely, SVR trailed behind with an R2 of 0.069 and an RMSE of 3.703. These conclusions were supported by cross-validation data, which showed that Random Forest had a CV RMSE of 2.67 and a CV R2 of 0.346, whereas Gradient Boosting had a CV RMSE of 2.76 and a CV R2 of 0.317. With a CV R^2 of 0.181 and a CV RMSE of 3.03, SVR's metrics remained lower.

Overall findings point to Random Forest's strong performance in both initial testing and cross-validation, which suggests good generalization across various data subsets and consistency in performance. Despite having the highest R2 at first, Gradient Boosting's cross-validation performance lagged below Random Forest's, which may indicate overfitting to the training set or sensitivity to the train-test split. When compared to tree-based models, SVR consistently showed poorer performance metrics in both scenarios, suggesting that it is not as good at capturing complicated patterns or non-linear interactions.

2s proteins with single domain

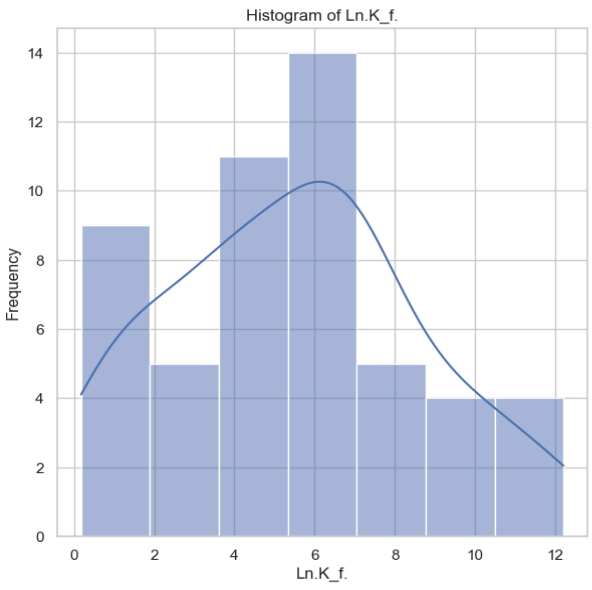
Early observations on the target variable ('Ln.K_f.') in the statistical analysis and correlation exploration showed low skewness, indicating a highly symmetric distribution. Notable relationships were revealed by feature correlations: 'DPC1_AF' and 'SEP_SS_ST', for example, showed large positive correlations with 'Ln.K_f.', whereas other features showed strong negative correlations. The following features were selected for the feature set: 'ATC_O, 'QSO1_G_A', 'SER_V', 'DPC1_AF', 'SEP_SS_ST', 'PCP_SS_ST', 'DPC1_GD', 'TPC_ILT', 'DPC1_DY']. These characteristics were selected based on their correlation and importance.

When it came to the model's performance, linear regression first produced findings that were mediocre, with an MSE of 2.62 and an R2 of 0.52and linear training data scores have MSE: 1.9017666550886811 and R^2: 0.5479260265604406

Then came the Tuned Random Forest Regressor and Gradient Boosting Regressor, whose R2 values were 0.5313523457989384 and 0.5241868038827561 and MSEs were 2.5954224013849014 and 2.615188912123639, respectively. Random forest showed score for training data as MSE: 1.419807717290628 and R^2: 0.7480269383931879 Best score for Gradient Boosting: 6.995098176392446 for Simplified Random Forest MSE: 2.7806988657203036 and R^2: 0.46205451172995726 with Training scores as MSE: 0.9680300212034848 and R^2: 0.8828687709407753 then for Ridge Regression Training score MSE: 1.9668574288286502 and R^2: 0.5164506442933895.

Support Vector Regression (SVR) was one possible use that was pointed out, highlighting its investigation of non-linear interactions that might act differently in this situation.There were notable differences in the performance of Ridge Regression both with and without the elimination of outliers. With an RMSE of 3.7876 and a R^2 of 0.7365, it first showed the best results.

Cross-validation and validation provided important information. A negative average R2 across folds (-0.2974) was seen in the cross-validation scores following the elimination of outliers, suggesting potential overfitting and consistency problems with the model. Inadequate feature representation, outlier removal impact, and model complexity are some of the issues that might lead to low validation scores. Poor validation scores may have been caused by overfitting, insufficient feature representation, outliers, poor data quality, and a cross-validation technique. These factors will need to be further investigated and refined in later modeling iterations.



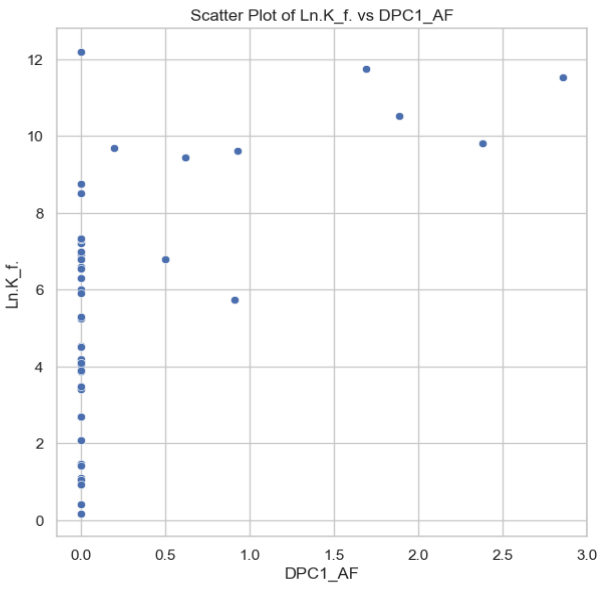
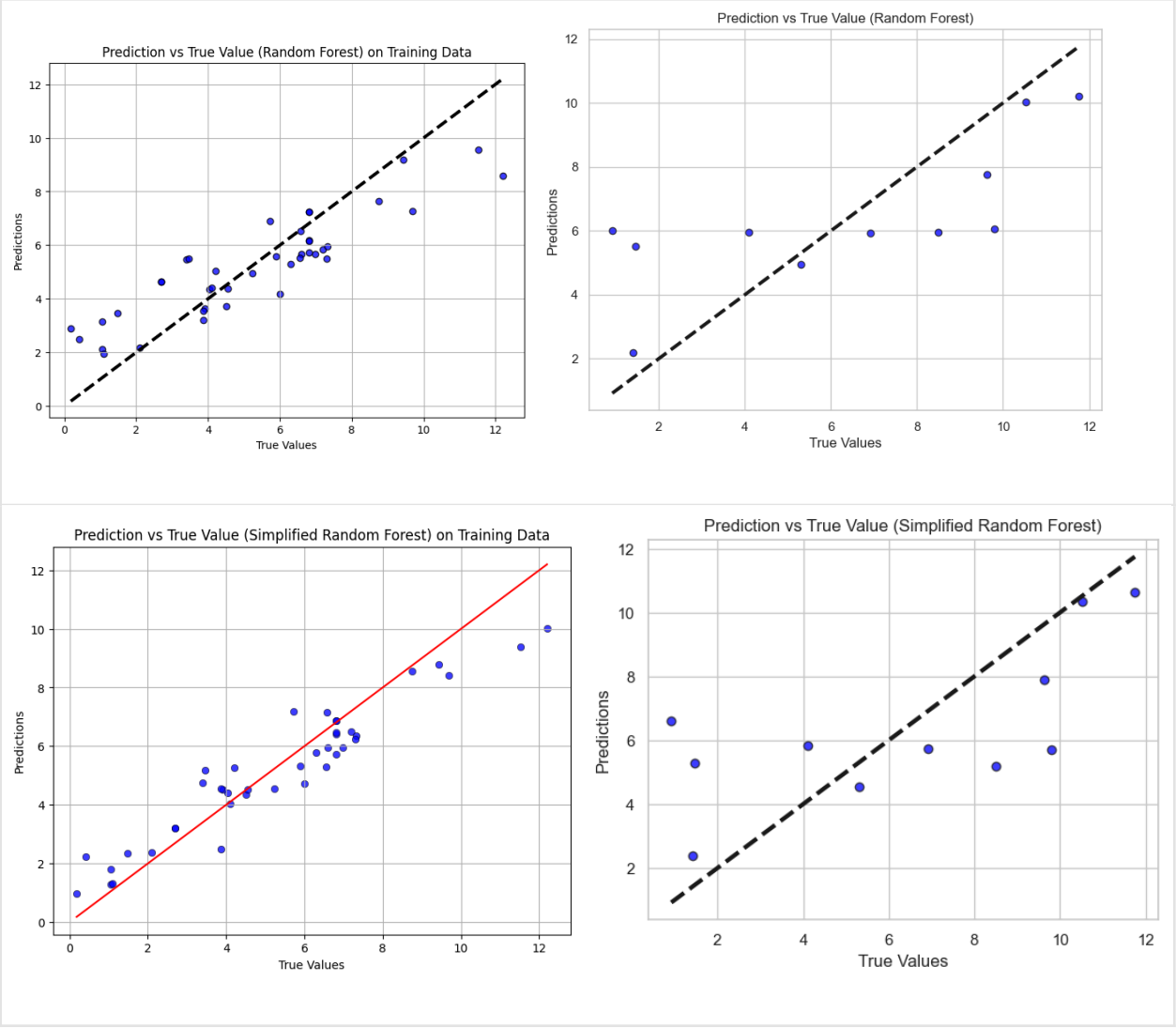


Fig.11.(a)The frequency distribution of "LnK_f" is displayed in the histogram. An attempt to evaluate the data's normality is indicated by the normal curve that is superimposed. "LnK_f" seems to be fairly regularly distributed, although the tail on the right suggests that there may be skewness or outliers.(b)A clear pattern can be seen in the scatter plot against "DPC1_AF," where "LnK_f" grows significantly at particular values of "DPC1_AF" (most notably, 2.5 to 3.0). This implies a non-linear relationship or threshold impact of some kind.



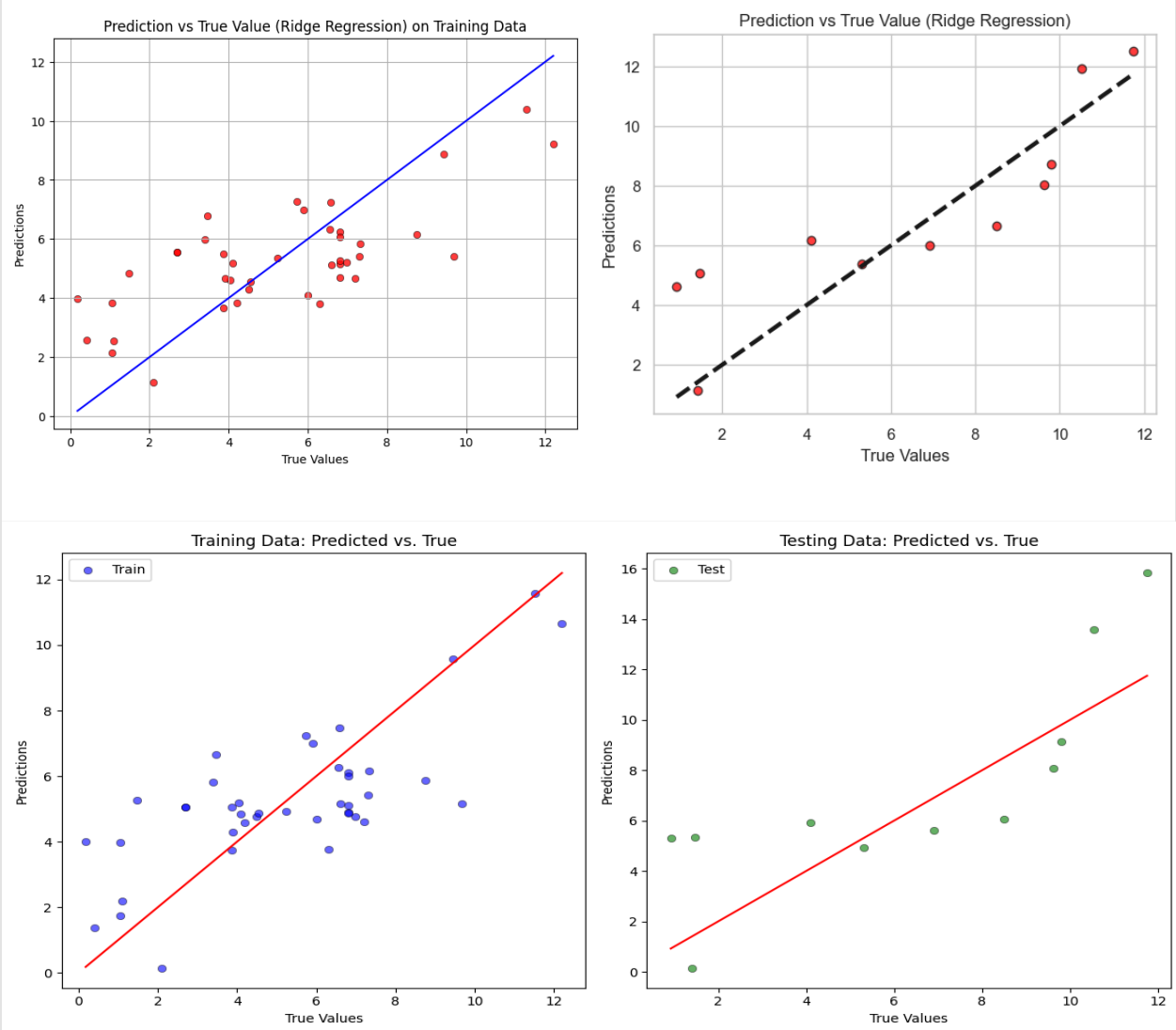


Fig.12. Random Forest, Simplified Random Forest, Ridge Regression and Linear Regression are four regression models whose projected values are plotted against the true values for training and testing datasets. The line of perfect prediction, where anticipated values equal true values, is represented by a dashed line. For the most part, the Random Forest models—particularly the basic model—show an excellent fit, indicating strong predictive performance. Ridge Regression demonstrates a strong fit as well, suggesting that it is capable of managing problems with the data, such as possible multicollinearity.

3.2 2D DESCRIPTOR

3.2.1 GSP all proteins (2s and n2s both) dataset

The investigation carried out on the protein binding affinity dataset shows a thorough investigation of different machine learning methods for regression assignments. The dataset was

first scaled according to conventional procedures, then the distribution of features and the goal variable, ln(kf), were analyzed using elementary statistics. The distribution of the data was subsequently visualized using histograms. The next step involved doing three sets of modeling experiments: the first set used the original features, the second set included polynomial features, and the third set applied non-linear transformations such as PCA and log transformation.

Random Forest, Gradient Boosting, SVR, Elastic Net, Voting Regressor, and Stacking Regressor were among the regression models that were trained and assessed in the initial set of tests. Although there was variation in the models' performance, none of them were remarkably accurate; their R2 ratings ranged from -0.015 to 0.136, suggesting that their predictive potential was restricted. To capture possible non-linear correlations, polynomial characteristics were added in the second set of tests. With R2 scores ranging from -0.067 to 0.136, this helped to increase performance somewhat, but it was still insufficient. Non-linear transformations were used in conjunction with PCA in the last series of studies. With R2 scores ranging from -0.045 to 0.093, the prediction performance remained subpar even though these strategies improved the models' interpretability.

Next, an ensemble method that combined Random Forest, Gradient Boosting, and SVR with a Voting Regressor was optimized using GridSearchCV. Performance improved slightly as a result of this optimisation; the top Voting Regressor had an RMSE of 3.25 and an R2 score of 0.071. Finally, hyperparameter-tuned XGBoost—a potent gradient boosting algorithm—was used. With an R2 score of 0.263 and an RMSE of 3.29 following optimisation, the XGBoost model demonstrated better performance, demonstrating its efficacy in identifying the underlying patterns in the dataset.

3.2.2 GSP 2s proteins (single and multi domain) dataset

Random Forest and Gradient Boosting: The dataset was used to train and assess the Random Forest and Gradient Boosting models. With an RMSE of 3.19 and an R2 score of 0.31, the Random Forest model demonstrated moderate predictive performance. Similarly, an R2 score of 0.26 and an RMSE of 3.31 were obtained with the Gradient Boosting model. These findings imply that while both models may benefit from additional optimisation, they are both capable of capturing a portion of the variance in the target variable.

Random Forest Grid Search: To maximise the Random Forest model's hyperparameters, a grid search was carried out. The search revealed that the configuration with the highest performance had a maximum depth of None, a minimum samples split of 2, a minimum samples leaf of 2, and 300 estimators. With these parameters, the Random Forest model performed marginally better than the original model, with an R2 score of 0.23 and an RMSE of 3.36 after retraining.

XGBoost: To determine the ideal hyperparameters, XGBoost was trained via grid search. 200 estimators, a learning rate of 0.01, a maximum depth of 3, a 0.8 subsample ratio, and a 0.7 column subsampling ratio were the optimal settings. With a R^2 score of 0.23 and an RMSE of 3.38 on the test data, the final model outperformed the Random Forest and Gradient Boosting models in terms of performance.

Enhanced XGBoost: A more complicated model with a higher maximum depth and more estimators was developed in order to significantly improve XGBoost's predictive power. With an R2 score of 0.27 and an RMSE of 3.28, this modified XGBoost model outperformed the original model and was performing similarly to the Random Forest and Gradient Boosting models.

3.2.3 GSP 2s single domain proteins dataset

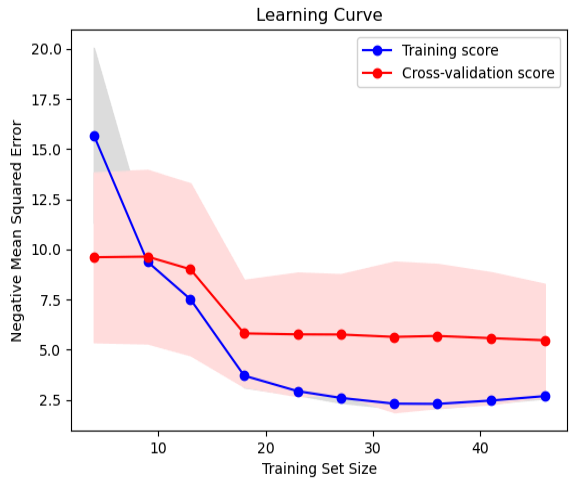
These were the correlation coefficient values we got before feature extraction.

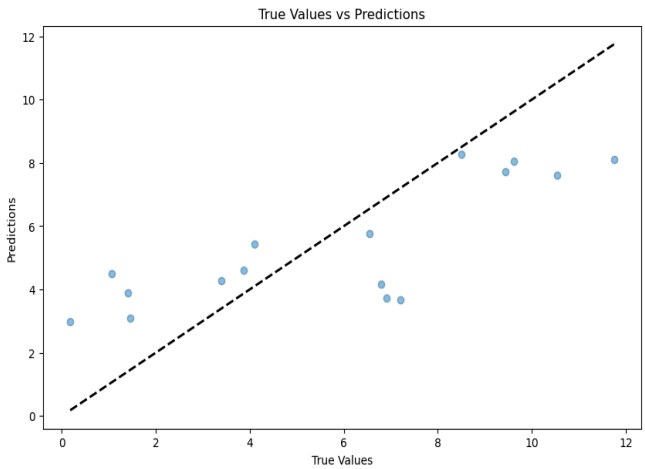
Table 5: Correlation coefficient values for 2S proteins with single domain

SIGNALS	LFC cutoffs	Max Correlation
Thermodynamic Transfer Hydrophobicity	0.19	0.67
Surrounding Hydrophobicity	0.19	0.67
Bulkiness	0.19	0.67
Refractive Index	0.19	0.71
Partial Molar Volume	0.22	0.68
Average Medium Contacts	0.11	0.67
Hgm Combined Surrounding Hydrophobicity	0.19	0.66
Solvent Accessible SA Denatured	0.22	0.67
Solvent Accessible SA Unfolding	0.19	0.67
Unfolding Hydration Enthalpy Change	0.18	0.68
Unfolding Hydration Entropy Change	0.19	0.70
Unfolding Hydration Heat Capacity Change	0.19	0.66
Unfolding Enthalpy Changes of Chain	0.18	0.70
Amino Molecular Mass	0.22	0.67
Volume	0.22	0.68
Unfolding Gibbs Free Energy Changes Chain	0.22	0.67

The five regression models we used were Gradient Boosting Regression, Random Forest Regression, Lasso Regression, Elastic Net, and Ridge Regression. The Ridge Regression model produced an R-squared value of roughly 0.50 and a mean squared error (MSE) of roughly 6.26. An MSE of roughly 6.46 and an R-squared value of roughly 0.49 were produced by the Lasso Regression model. The Elastic Net model produced an R-squared value of roughly 0.48 and an MSE of roughly 6.62. The R-squared value of approximately 0.52 and an MSE of approximately 6.06 were obtained by the Random Forest Regression model. Ultimately, an R-squared value of roughly 0.52 and an MSE of roughly 6.11 were obtained from the Gradient Boosting Regression model.

Additionally, GridSearchCV was used to optimize a Random Forest Regression model in order to determine the ideal hyperparameters. The model produced an R-squared value of roughly 0.54 and an MSE of about 5.86. The maximum depth of None, the square root of the number of features for the maximum features, two for the minimum samples split, and one for the minimum samples leaf were found to be the ideal values for the Random Forest model. In order to reduce complexity, the Random Forest model's parameters were adjusted. For example, the number of estimators was set to 100, the maximum tree depth to 10, and the minimum samples needed for splitting and leaf nodes were set to 10 and 4, respectively. The square root of the total features for splitting were taken into consideration. The training data MSE scores for this model is 2.8637411168111324, R2 score is 0.6479928918715597. The basic model performed well after training on the given dataset and producing predictions, as shown by a Mean Squared Error (MSE) of 5.566 and an R-squared (R^2) value of 0.560, which show a strong match to the data. In order to evaluate the model's generalization capacity, cross-validation was also used. The results showed an average MSE of 5.570 with a standard deviation of 2.826, which demonstrated the consistency of performance across different data subsets. These outcomes demonstrate how well the improved Random Forest model captures the underlying patterns in the dataset while keeping complexity under control.





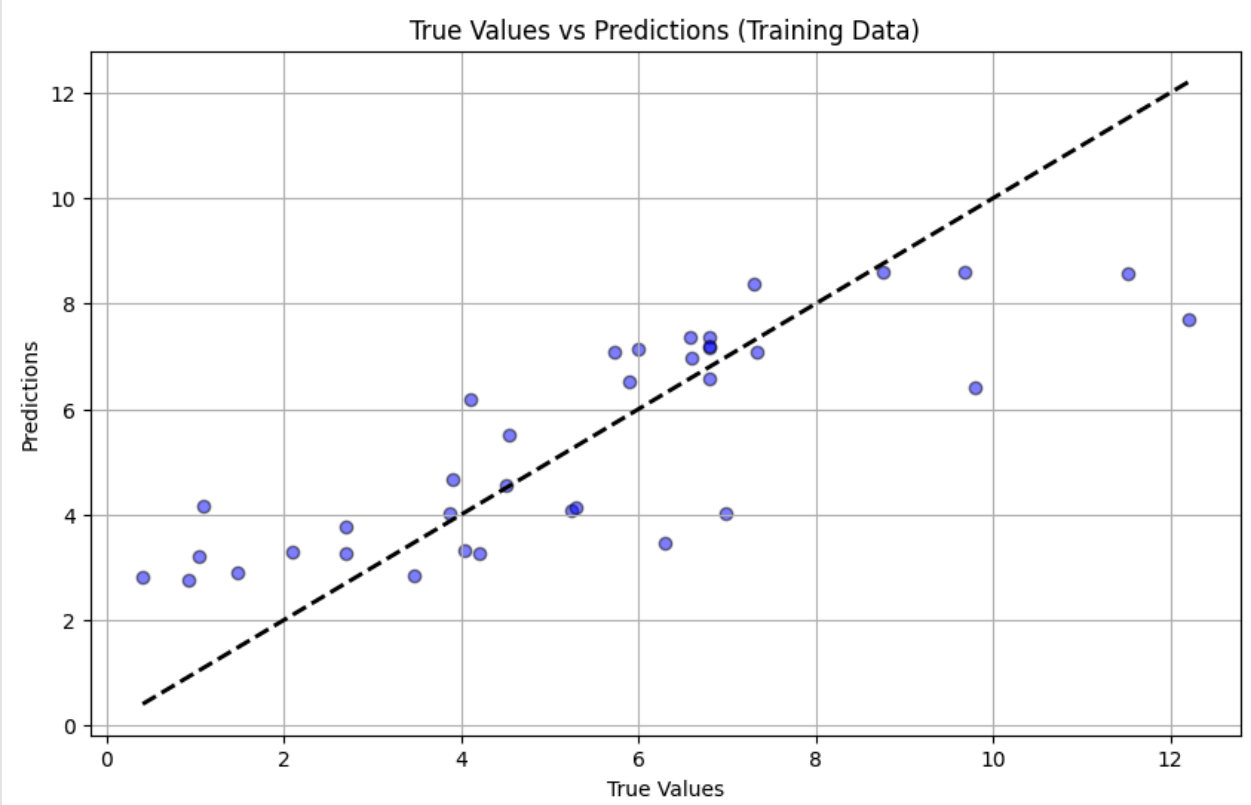


Fig.13.(a) **Learning Curve:** On both a training set and a cross-validation set, the learning curve graph shows the correlation between training sizes and the model's performance. A typical statistic for regression problems, the negative mean squared error is represented on the y-axis; larger values (nearer zero) denote greater performance. As additional data is used, the blue line, or training score, steadily declines from its extremely high beginning point, which denotes little error and thus good performance. This pattern is common because the model fits a small dataset very well, but it becomes harder to generalize as the dataset size grows. With the usage of additional training data, the cross-validation score (shown by the red line) improves, suggesting that additional data improves the model's ability to generalize. The training and cross-validation lines eventually converge, indicating that adding more data beyond this point may not result in a meaningful improvement in the model's performance because it will have plateaued. The variability (confidence interval) of the scores is shown by the darker areas surrounding each line, and it appears to stabilize with increasing data utilization. (b) **Predicted Values against True for testing data:** The target variable's genuine values are plotted against the values that the model predicts. Perfect predictions are indicated by all points falling on the dashed line. Different levels of prediction error are indicated by the dispersion of points. Most of the points fall below the line, indicating that the model tends to underestimate the real values. especially noticeable at larger true values. If the model continuously underpredicts higher values, for example, it may need to be adjusted or given new features in order to capture the variability at higher ranges of the target variable. This figure can assist in identifying any biases or tendencies in the model. (c) **Predicted Values against True for training data**

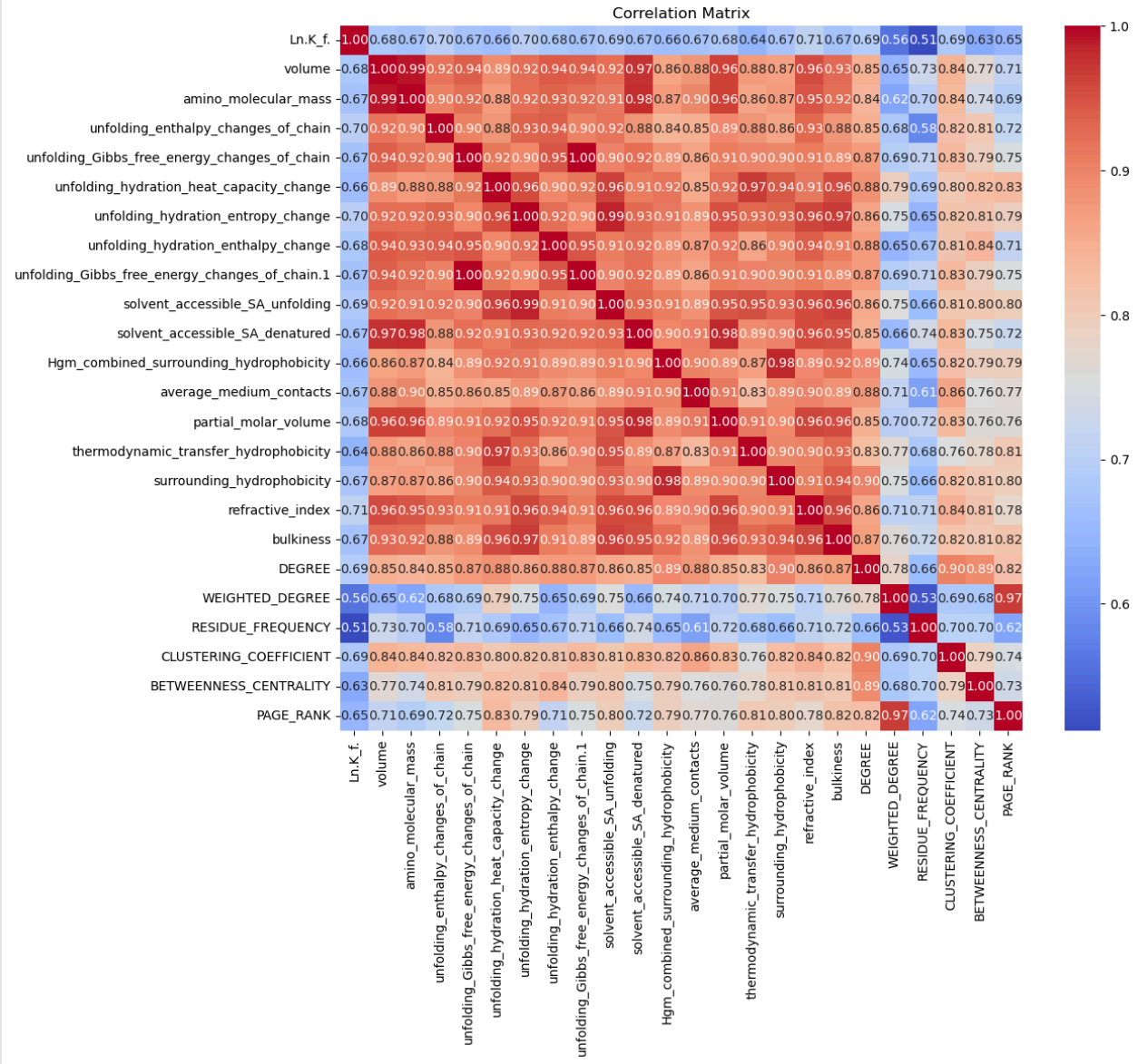


Fig.14. The correlation matrix shows High Inter-correlation: A number of variables exhibit a high degree of correlation, particularly those pertaining to structural traits and molecular unfolding. Given that several variables may be recording comparable processes or qualities, such as the hydration and energy changes that occur during molecule unfolding, the high degree of correlation points to redundancy. Moderate Correlation with LnK_f: Several parameters, especially those pertaining to solvent accessibility and hydration, show a moderate correlation with the key variable LnK_f, suggesting that these factors may have an impact on the kinetics of molecular folding. The intricate nature of molecular interactions and how they affect folding rates are shown by this relationship. Structural and Interaction Variables: Strong correlations between variables pertaining to molecular structure and interactions demonstrate their interconnectivity. This demonstrates how several molecular characteristics interact and affect the biochemical processes under study as a whole.

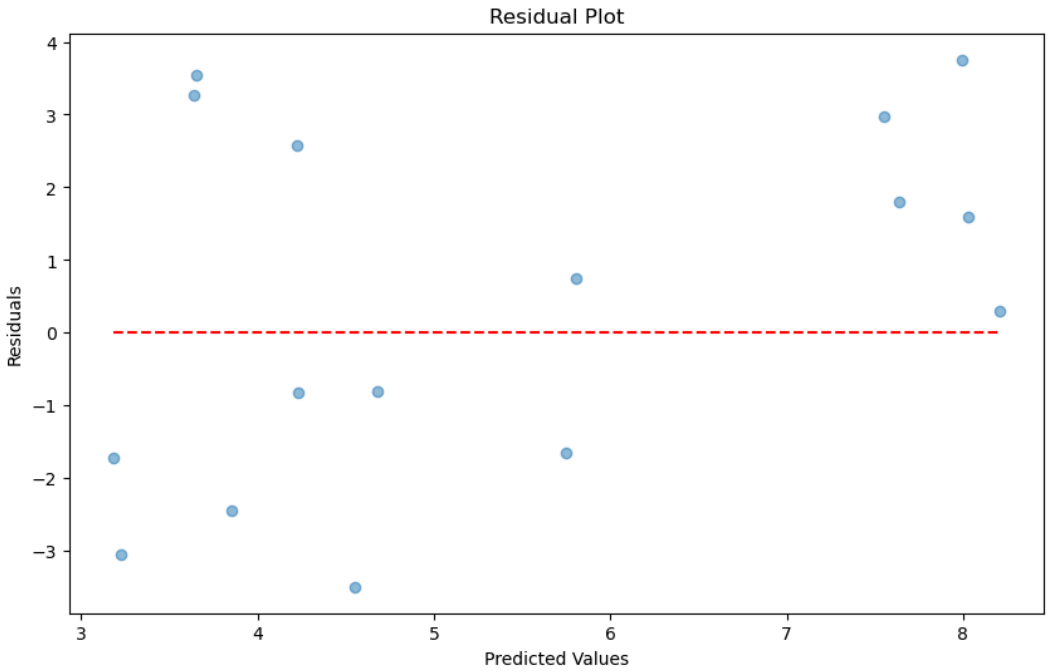


Fig.15. Residual Plot

3.3 Energy-based DESCRIPTOR

3.3.1 All proteins (2s and n2s both) post simulation dataset

Notably, the dataset includes a variety of energy components, including interaction energies like BOND, ANGLE, DIHED, VDWAALS, and EELEC, as well as total potential energy (EPtot) and total kinetic energy (EKtot). Significant departures from normal distributions were found in a number of features via skewness computation, which may be a symptom of underlying data asymmetries.

Support Vector Regression (SVR) and Gradient Boosting Regression (GBR) techniques were used in predictive modeling to forecast the natural logarithm of the rate constant (ln(kf)). Gradient Boosting produced somewhat predictive results, with a Mean Squared Error (MSE) of roughly 8.90 and an R-squared (R2) value of roughly 0.22. On the other hand, SVR produced an MSE of around 9.69 and an R2 score of about 0.15 with optimized hyperparameters (C=10, gamma='auto'). These results highlight the difficulties in estimating the rate constant from the available energy features, which are probably influenced by the intricacy and non-linearity of the

underlying physical processes. However, the analysis offers insightful information for additional investigation and improvement of predictive models in energy-related research.

3.3.2 2s proteins (single and multi domain) post simulation dataset

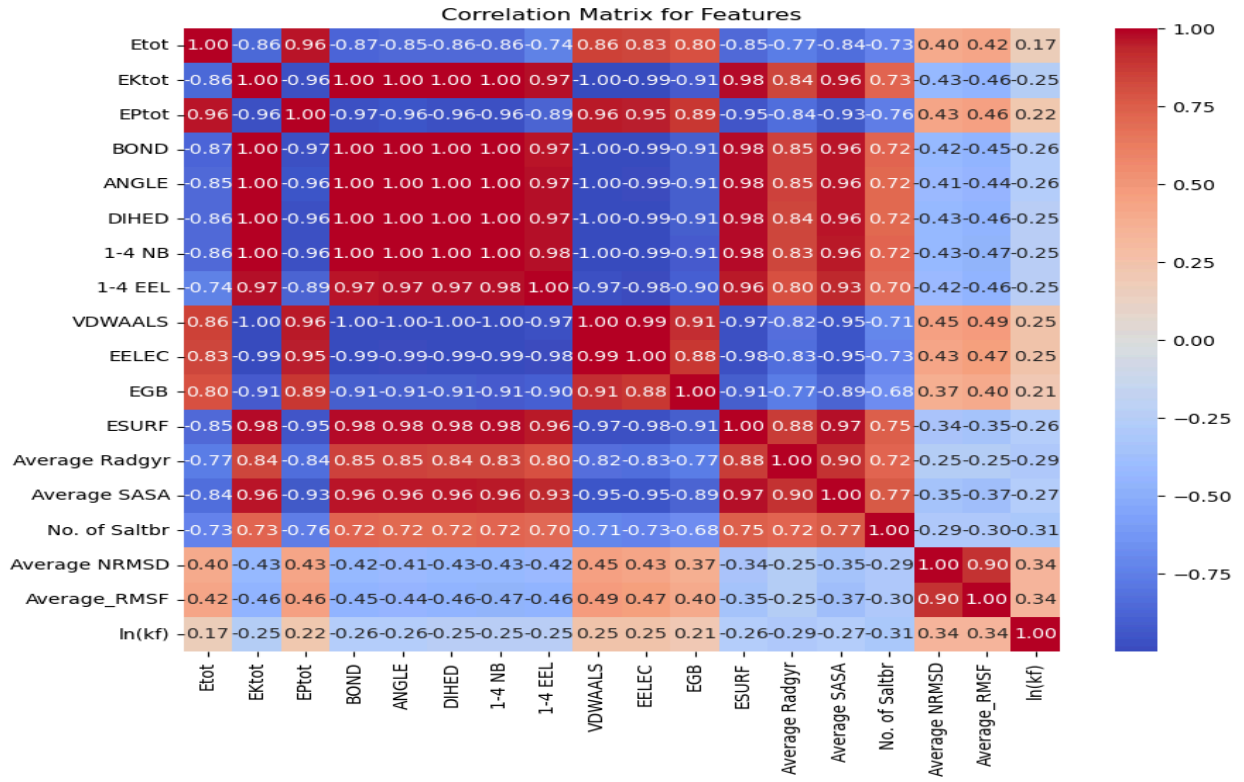


Fig.16. The Correlation Matrix displays the pairwise correlations among the various variables in your sample. Perfect positive correlation (value 1) and perfect negative correlation (value -1) are the two ranges of correlation values. Exceptionally high absolute values near -1 or 1 signify robust associations. Features such as ETOOT, EKtot, EPtot, BOND, ANGLE, DIHED, 1-4 NB, 1-4 EEL, VDWAALS, EELEC, EGB, and ESURF have strong correlations with one another, indicating that they are either produced from similar computations or measure related underlying processes. The variable ln(kf) exhibits a comparatively low correlation with the majority of attributes, indicating that a combination of factors rather than a single, significant one influences it.

Analyzing two regression models—Support Vector Regression (SVR) and Gradient Boosting—on a particular dataset yields informative results. Gradient Boosting is not as predictively accurate as the Optimised SVR model. Compared to Gradient Boosting, which has an MSE of 11.26, Optimised SVR has an MSE of 9.46. Moreover, Optimised SVR has a training data MSE score of 5.159242720304614, and R2 of 0.5904023418210322 and for test data

significantly higher coefficient of determination (R2 score) of 0.46 while Gradient Boosting only manages an R2 score of 0.36.

But further investigation using cross-validation reveals other subtleties. The Optimised SVR model shows a mean cross-validated R2 score of 0.159 with a standard deviation of 0.232, despite its better performance in the direct evaluation. Conversely, however, Gradient Boosting shows a mean cross-validated R2 score of only 0.025, but with a bigger standard deviation of 0.281, while initially doing less well.

These cross-validation results imply that both models would have trouble generalizing to new data, even if the Optimised SVR model shows higher predicted accuracy on the test set. It can be necessary to look at feature engineering, model complexity, or other alternatives in order to boost overall performance and guarantee robustness in a range of scenarios and datasets.

The pairwise relationships between various features are displayed in the correlation matrix. A direct relationship is implied by positive correlations (values closer to 1) while an inverse relationship is suggested by negative correlations (values closer to -1). High absolute correlation values between features indicate a high likelihood of mutual effect on the model predictions.

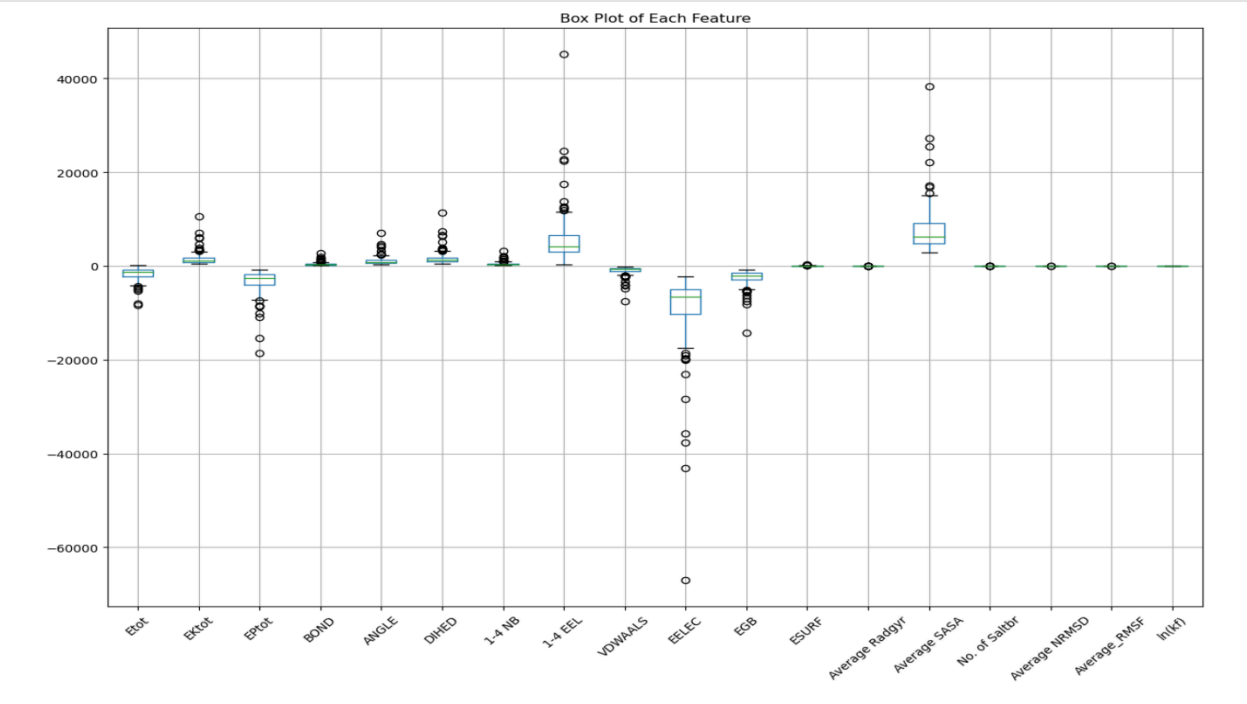


Fig.17. The box plots, which display the medians, quartiles, and outliers, provide a visual summary of the distribution of every feature. Finding characteristics with extreme values that could have an impact on model training is especially helpful. The points outside the box plots' whiskers indicate which attributes have a high number of outliers. Etott, EPtot, and VDWAALS are examples of this. Outliers may indicate that additional data cleansing, outlier removal, or the application of robust modelling approaches are necessary.

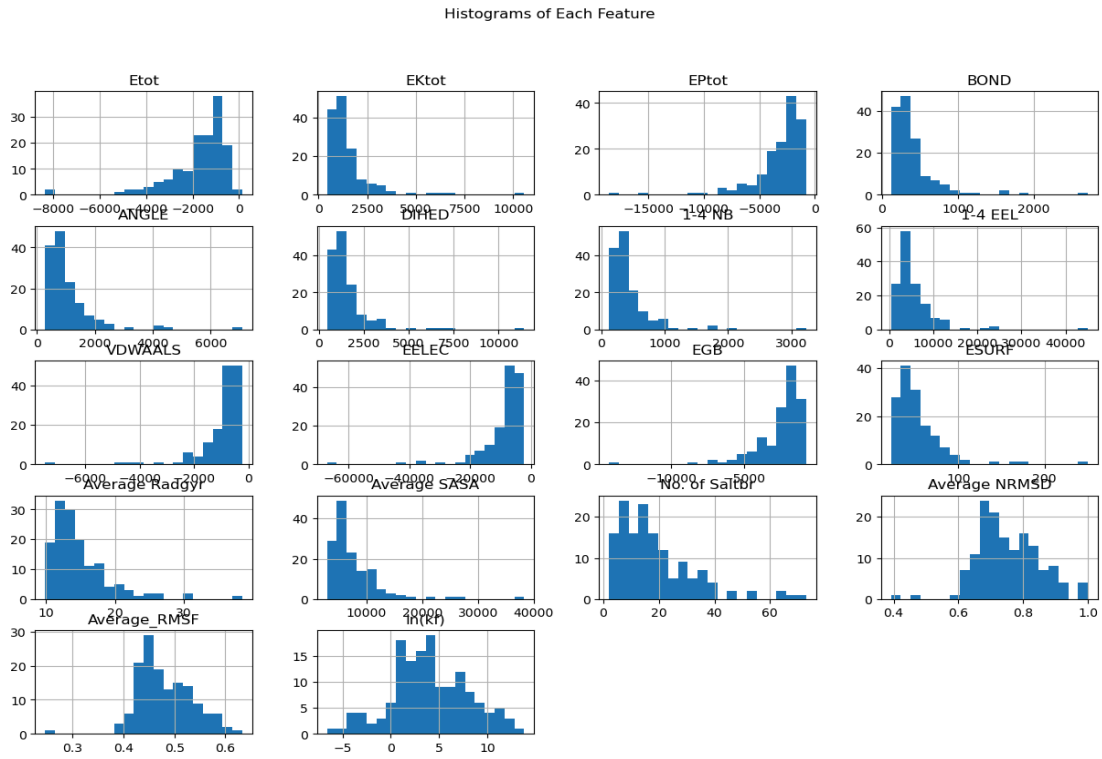
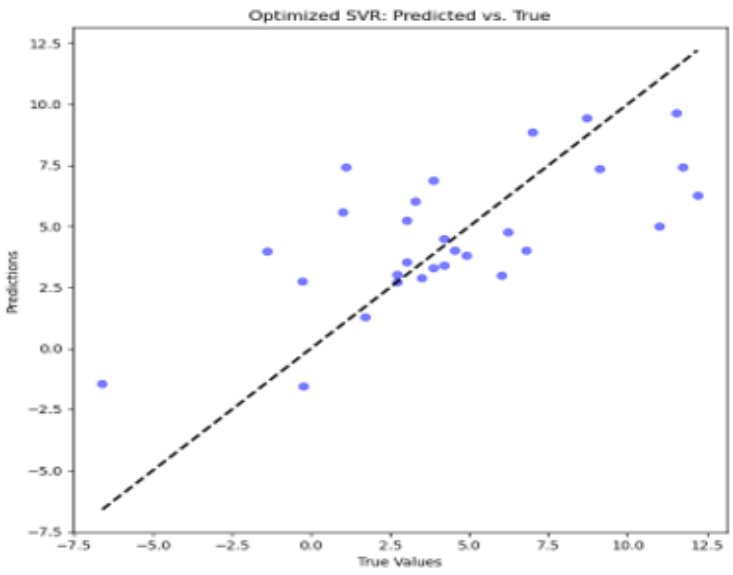


Fig.18. The distribution of values for every feature is shown by these histograms. Many characteristics exhibit skewness, either to the left or the right, which can affect model performance and require adjustments to produce more normal distributions, depending on the machine learning algorithms or statistical techniques being used. For instance, the distributions of Etot and EPtot are skewed left, while those of ESURF and EGB are skewed right.



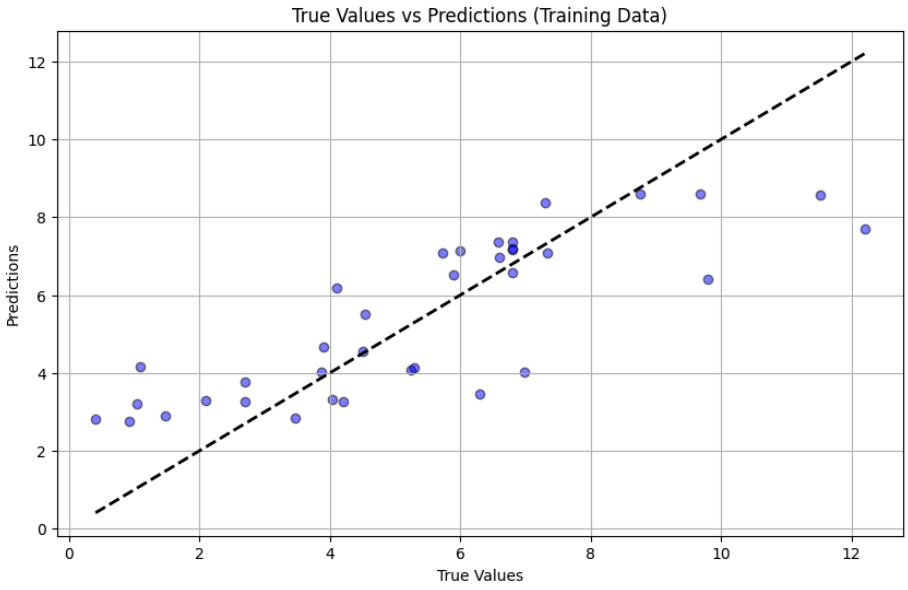


Fig.19.(a) Predict versus true graphs for test data -Gradient Boosting: A less accurate prediction is shown in many cases because the points are dispersed and do not line up perfectly with the diagonal line. This scatter points to a model that might be overly simplistic or one with a large variance. The model underestimates the higher true values, therefore there are obvious variances. Optimised SVR: Compared to Gradient Boosting, the points are closer to the diagonal line, indicating improved performance in terms of alignment with true values. While some outliers or extreme values are still not fully predicted, the SVR model appears to be more consistent across the range. Interpretation The closeness of the points to the diagonal which indicates that the Optimised SVR model performs better and provides predictions that are more accurate represent perfect prediction. **(b) Predict versus true graphs for test data.**

3.3.3 2s single domain proteins post simulation dataset

First, the Linear Regression model showed an R2 score of roughly -2.07 and an RMSE of roughly 4.54. This negative R2 result suggests a lack of fit between the model and the data, indicating that the model's predictions were less accurate than only utilising the target variable's mean as a predictor.

Upon switching to the Random Forest Regression model, it showed an approximately 3.54 RMSE, but its R^2 score was still negative at -0.86. This negative R2 result suggests that the model's fit to the data is not ideal, even with the RMSE lowered.

Finally, the optimized XGBoost model demonstrated an RMSE of roughly 3.01 and a marginally better but still negative R^2 score of about -0.15 following fine-tuning with GridSearchCV. The model's predictions may not have been able to adequately capture the variance in the target variable, as indicated by the continually negative R^2, even though the RMSE decreased in comparison to the previous models, indicating improved predictive accuracy.

CHAPTER 4

CONCLUSION

Impact on Machine Learning Predictions

The inclusion of the average of normalized 48 properties to machine learning models, Resulted in predicted accuracy to greatly increase from a multi-dimensional standpoint. The thesis has shown that the complex biological characteristics of amino acids may be efficiently measured and included into computational predictions by training the models on this dataset. This method not only increases the models' accuracy in folding rate prediction, but it also expands our knowledge of the ways in which particular features of amino acids affect the dynamics of proteins.

Moreover, the composition dataset's comprehensiveness makes it possible to investigate intricate relationships between characteristics, providing insights into how various amino acid combinations impact protein stability and folding rates. Various understandings are essential for theoretical biophysics research as well as real-world applications in genetic and pharmacological design, where a greater understanding of how various properties interact can result in more targeted interventions.

These datasets enable the models to capture a wider range of influencing elements, from simple compositional data to more intricate interaction patterns, by offering a more complete set of attributes. With the use of this enhanced feature set, models that can handle a wider variety of protein sequences which carries greater versatility .

Additionally, by employing Pfeature to extract features in an organized manner, high-quality and relevant data is supplied into the models, reducing noise and enhancing learning efficiency. This is especially crucial in the field of bioinformatics, as study conclusions can be greatly impacted by how accurate predictions are made.

Incorporating GSP gives the predictive models a huge boost by giving a greater grasp of how interactions between amino acids affect protein folding. This strategy makes it possible to investigate protein dynamics at a level of detail that is not possible with more conventional techniques, particularly when it comes to identifying the minute details of spatial relationships and interaction patterns inside proteins.

GSP's spectral analysis component has a significant influence because it makes it possible to identify important frequency components that are correlated with folding rates. These elements

can draw attention to particular protein sections or interactions that are essential for stability and effective folding, offering predicted insights that are not always obvious from other analytical techniques.

The incorporation of the Amber dataset offers a dynamic viewpoint on protein behavior, which considerably improves the predictive models:

Temporal Perspectives: The Amber dataset offers a time-series examination of protein folding, in contrast to static datasets that only present a single point in time. This makes it possible for the models to include kinetic data, which is essential for comprehending folding mechanisms and includes folding rates and transition stages.

Analysis of Energy and Stability: The collection contains comprehensive data on energy variations that occur during the folding process. Characteristics like total energy, kinetic energy, and potential energy are especially useful for forecasting stability and the possibility of achieving the folded state in various scenarios.

Structural Dynamics: The models can forecast how local and global structural changes influence folding results by analyzing changes in distances, angles, and torsional states. This involves figuring out which protein residues or areas are important for the folding process.

Implications for Future Research

The averaged 48 amino acids properties dataset offers a strong foundation for additional protein folding research. To gain a deeper understanding of the predictive power of the dataset, future research may investigate the application of more complex data analysis methods like principal component analysis (PCA) or machine learning algorithms like deep learning. Furthermore, adding dynamic features obtained from protein folding simulations to this dataset may yield a more dynamic and temporally accurate model of protein behaviors.

Future research in the field of protein folding will have a solid foundation thanks to the use of Pfeature composition. By combining these datasets with other kinds of biological data, such expression data, future research could build on this methodology. to create more thorough models that are able to forecast folding timeframes as well as the functional effects of folding variations. Furthermore, sophisticated machine learning methods like deep learning could be used to explore the intricate patterns and nonlinear relationships that these datasets conceal.

This thesis's use of GSP establishes a standard for next computational investigations into protein dynamics and provides other avenues for further study:

Multi-Scale Analysis: By adding multi-scale graph studies that take into account several levels of protein structure, from primary to quaternary structures, future study could build on GSP.

Integration with Other Data kinds: Models that predict not only folding durations but also the functional effects of mutations or alterations may be produced by combining GSP-derived features with other data kinds, such as genetic variation or post-translational modifications.

More Advanced GSP Methods: Investigating more sophisticated GSP methods, like machine learning algorithms made especially for graph data or adaptive graph filtering, may improve the precision and usefulness of prediction models.

Using the Amber dataset brings up a number of possibilities for further study and model improvement

Multi-Factorial Analysis: To examine the effects of multifactorial influences on folding kinetics and stability, future research could incorporate factors such as the influence of mutations or chemical alterations into the Amber simulations.

Improved Sampling Methods: Using improved sampling methods in the Amber framework, such as accelerated MD or metadynamics, may offer more in-depth understanding of conformations that are biologically significant but infrequently visited.

Hybrid Models: By fusing static structural data, experimental results, and dynamic data from Amber, hybrid models that predict protein behaviors more accurately and robustly may be created.

Remarks

The averaged 48 characteristics dataset establishes a standard for the thorough examination of amino acid properties in protein research and enhances the machine learning models created in this thesis. This dataset's clever integration highlights how computational techniques may be used to decipher the complexity of protein folding, opening new avenues for research into biological processes and the creation of innovative treatment approaches.
This thesis emphasizes how feature-rich, detailed datasets have the potential to transform computational biology's predictive modeling through the strategic application of Pfeature composition. The thesis makes a substantial contribution to the field by efficiently utilizing these datasets and offering instruments that can predict protein folding behaviors with high precision. This helps to progress disease research, drug creation, and our general understanding of protein functions.

Graph Signal Processing offers a powerful and advanced method for signal processing and network theory-based protein structure analysis. This thesis presents a successful integration of

GSP into the predictive modeling of protein folding, which advances our theoretical understanding of protein biophysics and creates new opportunities for applying this knowledge in molecular biology, drug design, and genetic research.

This thesis increases our understanding of the molecular dynamics underpinning protein folding and advances computational predictions of protein folding times by including the Amber dataset. This improved knowledge is crucial for domains like drug design, where understanding protein folding helps guide the creation of more potent medicinal compounds.

Bibliography

[1] F. Noé, G. De Fabritiis, and C. Clementi, "Machine learning for protein folding and dynamics," *Current Opinion in Structural Biology*, vol. 60. Elsevier Ltd, pp. 77–84, Feb. 01, 2020. doi: 10.1016/j.sbi.2019.12.005.

[2] K. Lindorff-Larsen, P. Maragakis, S. Piana, M. P. Eastwood, R. O. Dror, and D. E. Shaw, "Systematic validation of protein force fields against experimental data," *PLoS One*, vol. 7, no. 2, Feb. 2012, doi: 10.1371/journal.pone.0032131.

[3] P. Robustelli, S. Piana, and D. E. Shaw, "Developing a molecular dynamics force field for both folded and disordered protein states," *Proc Natl Acad Sci U S A*, vol. 115, no. 21, pp. E4758–E4766, May 2018, doi: 10.1073/pnas.1800690115.

[4] "Media Connections." [Online]. Available: www.courses.bfwpub.com/lehninger6e.

[5] S. Nithiyanandam, V. K. Sangaraju, B. Manavalan, and G. Lee, "Computational prediction of protein folding rate using structural parameters and network centrality measures," *Comput Biol Med*, vol. 155, Mar. 2023, doi: 10.1016/j.compbiomed.2022.106436.

[6] A. E. P. Durumeric *et al.*, "Machine Learned Coarse-Grained Protein Force-Fields: Are We There Yet?," 2023.

[7] F. R. Salemme, "STRUCTURAL PROPERTIES OF PROTEIN fi-SHEETS II. GEOMETRICAL AND SYMMETRY PROPERTIES OF /~-SHEETS Ill. THE GEOMETRY OF TruSTED fi-S~EETS IN PROT| :INS 1. Parallel Sheets 2. Antiparallel fl-Sheets," 1983.

[8] J. C. Kendrew, "The Three-dimensional Structure of a Protein Molecule," vol. 205, no. 6, pp. 96–111, 1961, doi: 10.2307/24937166.

[9] C. B. (Christian B. Anfinsen, J. T. (John T. Edsall, and F. M. (Frederic M. Richards, *Advances in protein chemistry. Volume 34*. Academic Press, 1981.

[10] M. Beeby, B. D. O'Connor, C. Ryttersgaard, D. R. Boutz, L. J. Perry, and T. O. Yeates, "The genomics of disulfide bonding and protein stabilization in thermophiles," *PLoS Biol*, vol. 3, no. 9, pp. 1549–1558, 2005, doi: 10.1371/journal.pbio.0030309.

[11] A. K. Dunker and R. W. Kriwacki, "the orderly chaos of proteins," vol. 304, no. 4, pp. 68–73, 2011, doi: 10.2307/26002479.

[12] J. H. Brown, "Breaking symmetry in protein dimers: Designs and functions," *Protein Science*, vol. 15, no. 1, pp. 1–13, Jan. 2006, doi: 10.1110/ps.051658406.

[13] A. Herrá Ez, "Articles Biomolecules in the Computer Jmol TO THE RESCUE," 2006. [Online]. Available: www.jmol.org,

[14] C. P. Ponting and R. R. Russell, "The natural history of protein domains," *Annual Review of Biophysics and Biomolecular Structure*, vol. 31. pp. 45–71, 2002. doi: 10.1146/annurev.biophys.31.082901.134314.

[15] F. Chiti and C. M. Dobson, "Protein misfolding, functional amyloid, and human disease," *Annual Review of Biochemistry*, vol. 75. pp. 333–366, 2006. doi: 10.1146/annurev.biochem.75.101304.123901.

[16] K. A. Dill, S. B. Ozkan, M. S. Shell, and T. R. Weikl, "The protein folding problem," *Annual Review of Biophysics*, vol. 37. pp. 289–316, 2008. doi: 10.1146/annurev.biophys.37.092707.153558.

[17] E. Gazit, "Mechanisms of amyloid fibril self-assembly and inhibition: Model short peptides as a key research tool," *FEBS Journal*, vol. 272, no. 23. pp. 5971–5978, Dec. 2005. doi: 10.1111/j.1742-4658.2005.05022.x.

[18] F. U. Hartl, A. Bracher, and M. Hayer-Hartl, “Molecular chaperones in protein folding and proteostasis,” *Nature*, vol. 475, no. 7356. pp. 324–332, Jul. 21, 2011. doi: 10.1038/nature10317.

[19] J. W. M. Höppener and C. J. M. Lips, “Role of islet amyloid in type 2 diabetes mellitus,” *International Journal of Biochemistry and Cell Biology*, vol. 38, no. 5–6. pp. 726–736, 2006. doi: 10.1016/j.biocel.2005.12.009.

[20] H. H. Kampinga and E. A. Craig, “The HSP70 chaperone machinery: J proteins as drivers of functional specificity,” *Nature Reviews Molecular Cell Biology*, vol. 11, no. 8. pp. 579–592, Aug. 2010. doi: 10.1038/nrm2941.

[21] J. Tyedmers, A. Mogk, and B. Bukau, “Cellular strategies for controlling protein aggregation,” *Nature Reviews Molecular Cell Biology*, vol. 11, no. 11. pp. 777–788, Nov. 2010. doi: 10.1038/nrm2993.

[22] K. W. Plaxco, K. T. Simons, and D. Baker, “Contact Order, Transition State Placement and the Refolding Rates of Single Domain Proteins.”

[23] M. M. Gromiha and S. Selvaraj, “Comparison between long-range interactions and contact order in determining the folding rate of two-state proteins: Application of long-range order to folding rate prediction,” *J Mol Biol*, vol. 310, no. 1, pp. 27–32, Jun. 2001, doi: 10.1006/jmbi.2001.4775.

[24] K. Kamagata, M. Arai, and K. Kuwajima, “Unification of the folding mechanisms of non-two-state and two-state proteins,” *J Mol Biol*, vol. 339, no. 4, pp. 951–965, Jun. 2004, doi: 10.1016/j.jmb.2004.04.015.

[25] C. Soto, ; Lisbell, and D. Estrada, “Protein Misfolding and Neurodegeneration,” 2008. [Online]. Available: http://archneur.jamanetwork.com/

[26] B. Ptitsyn, “MOLTEN GLOBULE AND PROTEIN FOLDING.”

[27] M. M. Gromiha, A. M. Thangakani, and S. Selvaraj, “FOLD-RATE: Prediction of protein folding rates from amino acid sequence,” *Nucleic Acids Res*, vol. 34, no. WEB. SERV. ISS., Jul. 2006, doi: 10.1093/nar/gkl043.

[28] “Media Connections.” [Online]. Available: www.courses.bfwpub.com/lehninger6e.

[29] X. Cheng, X. Xiao, Z. C. Wu, P. Wang, and W. Z. Lin, “Swfoldrate: Predicting protein folding rates from amino acid sequence with sliding window method,” *Proteins: Structure, Function and Bioinformatics*, vol. 81, no. 1, pp. 140–148, Jan. 2013, doi: 10.1002/prot.24171.

[30] B. Manavalan, K. Kuwajima, and J. Lee, “PFDB: A standardized protein folding database with temperature correction,” *Sci Rep*, vol. 9, no. 1, Dec. 2019, doi: 10.1038/s41598-018-36992-y.

[31] G. N. Lin, Z. Wang, D. Xu, and J. Cheng, “SeqRate: sequence-based protein folding type classification and rates prediction,” 2009. [Online]. Available: http://www.biomedcentral.com/1471-2105/11/S3/S1

[32] H.-B. Shen, J.-N. Song, and K.-C. Chou, “Prediction of protein folding rates from primary sequence by fusing multiple sequential features,” 2009. [Online]. Available: http://www.scirp.org/journal/jbise

[33] C. C. H. Chang, B. T. Tey, J. Song, and R. N. Ramanan, “Towards more accurate prediction of protein folding rates: A review of the existing web-based bioinformatics approaches,” *Brief Bioinform*, vol. 16, no. 2, pp. 314–324, Mar. 2015, doi: 10.1093/bib/bbu007.

[34] M. Corrales *et al.*, “Machine learning: How much does it tell about protein folding rates?,” *PLoS One*, vol. 10, no. 11, Nov. 2015, doi: 10.1371/journal.pone.0143166.

[35] M. M. Gromiha, “A statistical model for predicting protein folding rates from amino acid sequence with structural class information,” *J Chem Inf Model*, vol. 45, no. 2, pp. 494–501, 2005, doi: 10.1021/ci049757q.

[36] M. Punta and B. Rost, “Protein folding rates estimated from contact predictions,” *J Mol Biol*, vol. 348, no. 3, pp. 507–512, May 2005, doi: 10.1016/j.jmb.2005.02.068.

[37] M. M. Gromiha and S. Selvaraj, “Inter-residue interactions in protein folding and stability,” *Progress in Biophysics and Molecular Biology*, vol. 86, no. 2. pp. 235–277, Oct. 2004. doi: 10.1016/j.pbiomolbio.2003.09.003.

[38] M. M. Gromiha, M. Oobatake, H. Kono, H. Uedaira, and A. Sarai, “Role of structural and sequence information in the prediction of protein stability changes: comparison between buried and partially buried mutations,” 1999. [Online]. Available: http://www.rtc.riken.go.jp/~gromiha/table.html

[39] M. M. Gromiha, “Importance of Native-State Topology for Determining the Folding Rate of Two-State Proteins,” *J Chem Inf Comput Sci*, vol. 43, no. 5, pp. 1481–1485, Sep. 2003, doi: 10.1021/ci0340308.

[40] M. M. Gromiha, M. Oobatake, and A. Sarai, “Important amino acid properties for enhanced thermostability from mesophilic to thermophilic proteins,” 1999.

[41] M. M. Gromiha, M. Oobatake, H. Kono, H. Uedaira, and A. Sarai, “Importance of surrounding residues for protein stability of partially buried mutations,” *J Biomol Struct Dyn*, vol. 18, no. 2, pp. 281–295, 2000, doi: 10.1080/07391102.2000.10506666.

[42] D. Srivastava, G. Bagler, and V. Kumar, “Graph Signal Processing on protein residue networks helps in studying its biophysical properties,” *Physica A: Statistical Mechanics and its Applications*, vol. 615, Apr. 2023, doi: 10.1016/j.physa.2023.128603.

[43] W. Yan, J. Zhou, M. Sun, J. Chen, G. Hu, and B. Shen, “The construction of an amino acid network for understanding protein structure and function,” *Amino Acids*, vol. 46, no. 6. Springer-Verlag Wien, pp. 1419–1439, 2014. doi: 10.1007/s00726-014-1710-6.

[44] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Process Mag*, vol. 30, no. 3, pp. 83–98, 2013, doi: 10.1109/MSP.2012.2235192.

[45] Z. Elftmaoui and E. Bignon, “Robust AMBER Force Field Parameters for Glutathionylated Cysteines,” *Int J Mol Sci*, vol. 24, no. 19, Oct. 2023, doi: 10.3390/ijms241915022.

[46] P. Smardz, M. Anila, P. Rogowski, M. S. Li, B. Różycki, and P. Krupa, “Protocols for Multi-Scale Molecular Dynamics Simulations: A Comparative Study for Intrinsically Disordered Amyloid Beta in Amber & Gromacs on CPU & GPU”, doi: 10.1101/2023.10.24.563575.

[47] A. Pande *et al.*, “Computing wide range of protein/peptide features from their sequence and structure Equal contribution * Corresponding authors”, doi: 10.1101/599126.

[48] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Frequency analysis,” *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014, doi: 10.1109/TSP.2014.2321121.