Predictive Analysis of protein structures: Parsing PDB data for Structural and Sequential Insight and Active Site Prediction in Protein-Protein Interaction at residue level

A Thesis Submitted to

FACULTY OF ENGINEERING AND TECHNOLOGY JADAVPUR UNIVERSITY

Submitted in partial fulfillment of the requirements for the degree of,

MASTER OF TECHNOLOGY IN COMPUTER TECHNOLOGY

By

Godhuli Das

Examination Roll: M6TCT24023

Registration No:160099 of 2021 – 2024

Under the guidance of

Prof. Subhadip Basu

Jadavpur University

188, Raja S.C. Mallick Rd,

Kolkata - 700032, West Bengal, India

CERTIFICATE OF RECOMMENDATION

This is to certify that the work embodied in this thesis entitled “Predictive Analysis of protein structures: Parsing PDB data for Structural and Sequential Insight and Active Site Prediction in Protein-Protein Interaction at residue level '' has been satisfactorily completed by Godhuli Das (Registration Number 160099 of 2021 - 2024; Class Roll No. 02110504028; Examination Roll No. M6TCT24023). It is a bona fide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata, for partial fulfillment of the requirements for the awarding of the Master of Technology in Computer Technology degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2021 - 2024.

Prof. (Dr.) Nandini Mukherjee
Head of the Department Department of Computer Science and Engineering Jadavpur University

---

Prof. (Dr.) Subhadip Basu,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Prof. (Dr.) Dipak Laha,
DEAN, Faculty of Engineering and Technology,
Jadavpur University

CERTIFICATE OF APPROVAL

This is to certify that the work embodied in this thesis entitled “Predictive Analysis of protein structures: Parsing PDB data for Structural and Sequential Insight and Active Site Prediction in Protein-Protein Interaction at residue level'' has been satisfactorily completed by Godhuli Das (Registration Number 160099 of 2021 - 2024; Class Roll No. 02110504028; Examination Roll No. M6TCT24023). It is a bonafide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata, for partial fulfillment of the requirements for the awarding of the Master of Technology in Computer Technology degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2021 - 2024.

Signature of Examiner 1
Date:

Signature of Examiner 2
Date:

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHIC

I hereby declare that the thesis entitled “Predictive Analysis of protein structures: Parsing PDB data for Structural and Sequential Insight and Active Site Prediction in Protein-Protein Interaction at residue level” contains literature survey and original research work by the undersigned candidate, as a part of his degree of Master of Technology in Computer Technology, Jadavpur University. All the information has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Godhuli Das

Examination Roll No. M6TCT24023

Registration No.: 160099 of 2021 - 2024

Thesis Title: Predictive Analysis of protein structures: Parsing PDB data for Structural and Sequential Insight and Active Site Prediction in Protein-Protein Interaction at residue level

Signature of the Candidate:

Godhuli Das

Acknowledgment

I would like to express my deepest gratitude to my supervisor Prof. Subhadip Basu for allowing me to carry out my research work under his supervision. His insight in the field of Bioinformatics inspired me to undertake my own journey in this interdisciplinary domain of science. I shall always remain grateful to him for his insightful guidance, patient supervision, and his continuous encouragement for independent exploration and research. I am thankful to my seniors, classmates, Lab mates and my fellow friends for their help and support at different points of time. Their wealth of experience has been a source of strength for me throughout the duration of my work. I would like to express my gratitude and indebtedness to my parents and all my family members for their unbreakable belief, constant encouragement, moral support, and guidance.

Godhuli Das

Abstract

The prediction of active sites in protein-protein interactions (PPIs) is a crucial aspect of computational biology, aiding in the understanding of molecular mechanisms and facilitating drug discovery. This study presents an integrated workflow utilizing DenseNet201, a deep learning architecture, for predicting active sites in PPIs. The process begins with the retrieval and parsing of PDB files to extract chain sequences, followed by the computational generation of interatomic distance matrices of protein chains at the residue level. Sequence information and distance matrices, representing structural data for both positive and negative PPI datasets, are considered. Heatmaps generated from these distance matrices visually depict the proximity of active sites, with a color spectrum ranging from green (indicating shorter PPI distances) to red (indicating longer PPI distances), which can be used for future validation purposes. A color encoding strategy maps bigram interactions of amino acids to a lookup table, resulting in the generation of images from protein sequence information parsed from PDB data. Using a sliding window approach with specific dimensions of 32x32 and a stride of 2, sub-images are created and fed into the DenseNet201 model for training and testing. The classification strategy evaluates PPIs based on confidence scores compared against a threshold of 0.5. The workflow includes the generation of performance metrics, ensuring a robust evaluation of the model's predictive capabilities. This comprehensive approach enhances the visualization and prediction of active sites in PPIs, significantly contributing to advancements in bioinformatics. Currently, the workflow, which includes PDB data parsing and active site prediction by training image data using the DensePPI model, has been trained and tested, resulting in an overall accuracy of 95.84%. Additionally, this research will contribute to another project focusing on protein-protein interaction prediction, employing Graph Convolutional networks and graphlet features.

Table of Contents

List of Figures

List of Table

Chapter 1

Introduction

Protein complexes are assemblies of multiple protein molecules that interact with each other to perform specific biological functions. These complexes can involve homomeric interactions (where the interacting proteins are identical) or heteromeric interactions (where the interacting proteins are different). Protein complexes play crucial roles in various cellular processes such as signal transduction, gene expression, metabolism, and cell communication.

The Research Collaboratory for Structural Bioinformatics Protein Data Bank (**RCSB PDB**) (website - https://www.rcsb.org/)is a comprehensive resource for the three-dimensional structures of biological macromolecules, including proteins, nucleic acids, and complex assemblies such as protein-protein complexes. It provides free access to experimentally determined structures obtained through techniques such as X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and electron microscopy.

The RCSB PDB serves as a central repository for **structural data** and offers a wide range of tools and resources for researchers to explore, analyze, and visualize protein structures. It plays a critical role in structural biology, bioinformatics, drug discovery, and molecular biology research by providing valuable insights into the structure-function relationships of biomolecules and facilitating the development of new therapeutic interventions and biotechnological applications. Additionally, RCSB PDB offers a suite of tools and resources for structure visualization, analysis, and education, making it an invaluable resource for the scientific community.

UniProt serves as a comprehensive resource encompassing protein sequence and functional information, comprising two main components: the UniProt Knowledgebase (UniProtKB) and UniProt Reference Clusters (UniRef). UniProtKB serves as a central repository, meticulously curated by expert annotators, housing protein sequences and rich functional details including names, sequences, functions, subcellular locations, interactions, post-translational modifications, and literature references. Updated regularly, it integrates new findings from scientific research. On the other hand, UniRef offers clustered sets of protein sequences from UniProtKB and other databases, aimed at reducing redundancy by grouping similar sequences and providing representative ones for each cluster. Widely utilized by researchers, bioinformaticians, and

biologists, UniProt facilitates protein annotation, sequence analysis, functional characterization, and comparative genomics, thereby offering valuable insights into the properties and functions of proteins across diverse species and biological contexts. The UniProt website (https://www.uniprot.org/) serves as the primary interface for accessing the UniProt Knowledgebase (UniProtKB) and its associated resources.

1.1 Distinct Yet Interconnected: Contrasting Data Offered by PDB and UniProt for Comprehensive Protein Understanding

Proteins are multifaceted biomolecules and understanding them requires insights from various angles. The Protein Data Bank (PDB) and UniProt are two prominent resources offering different perspectives on proteins, yet they are intricately connected, enriching each other's data.

Protein Data Bank (PDB):

- PDB primarily provides structural information about proteins, nucleic acids, and complex assemblies. It includes data on the three-dimensional (3D) coordinates of atoms within these molecules, obtained through experimental techniques such as X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and electron microscopy.
- PDB entries offer insights into the shapes, sizes, and spatial arrangements of proteins and their interactions with other molecules. This structural data is crucial for understanding protein folding, function, and dynamics.
- PDB does not typically provide detailed functional annotations or sequence information about proteins beyond what is necessary for structural determination.

UniProt:

- UniProt, on the other hand, is a comprehensive repository of protein sequence and functional information. It includes data on protein names, sequences, functions, subcellular locations, interactions, post-translational modifications, and literature references.
- UniProtKB, the central component of UniProt, offers detailed functional annotations curated by expert annotators, providing insights into the biological roles and activities of proteins.
- UniProt also provides cross-references to other databases, including PDB, allowing users to access

structural information for proteins listed in UniProtKB.

Variation and Connection:

- While PDB focuses on structural data, UniProt emphasizes functional and sequence information. This difference reflects their distinct but complementary roles in protein research.
- Despite these differences, PDB and UniProt are connected through cross-references and data integration efforts. UniProt entries often include links to corresponding PDB structures, enabling users to access structural data for specific proteins listed in UniProtKB.
- Researchers can leverage both PDB and UniProt to gain a comprehensive understanding of proteins, integrating structural, functional, and sequence data to explore their properties, interactions, and biological roles.

In summary, PDB and UniProt offer varied but interconnected information about proteins, providing complementary insights into their structures, functions, and relationships within biological systems.

1.2 What are Protein-protein interactions (PPIs)?

Protein-protein interactions (PPIs) refer to the physical contacts established between two or more proteins within a biological system. These interactions are crucial for virtually all cellular processes, including signal transduction, gene regulation, metabolic pathways, and cell-to-cell communication. PPIs govern the formation of protein complexes, which are dynamic assemblies of proteins that work together to perform specific functions.

There are several types of protein-protein interactions, including:

1. Enzyme-substrate Interactions: Enzymes interact with specific substrate molecules to catalyze biochemical reactions.
2. Receptor-ligand Interactions: Cell surface receptors interact with signaling molecules (ligands) to initiate intracellular signaling cascades.
3. Protein Complex Formation: Proteins interact with each other to form stable complexes with defined structures and functions.

4. Protein Binding: Proteins bind to other proteins or molecules to regulate their activity, localization, or stability.

Understanding protein-protein interactions is essential for deciphering the molecular mechanisms underlying biological processes and disease pathways. Experimental techniques such as yeast two-hybrid assays, co-immunoprecipitation, and fluorescence resonance energy transfer (FRET) are commonly used to study PPIs. Additionally, computational methods, including protein docking simulations and co-evolutionary analysis, are employed to predict and analyze protein interactions on a large scale. Overall, elucidating protein-protein interactions provides valuable insights into the organization, regulation, and dynamics of cellular networks.

1.3. What is Active site prediction in Protein complexes?

Active site prediction in protein complexes involves identifying and characterizing the specific region of a protein where catalytic activity or binding interactions with other molecules occur. This is crucial for understanding the function of the protein and its role in various biological processes.

Several computational methods are used for active site prediction in protein complexes. These methods often involve structural analysis, molecular docking, and bioinformatics techniques. Here's a simplified overview of the process:

1. Structure Determination: Experimental techniques like X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy are used to determine the three-dimensional structure of the protein complex.

2. Sequence and Structure Analysis: Bioinformatics tools are employed to analyze the sequence and structure of the protein complex. This includes identifying conserved motifs or residues known to be involved in catalysis or binding.

3. Molecular Docking: Computational techniques such as molecular docking are used to predict the binding interactions between the protein complex and potential ligands or substrates. This helps in identifying potential active sites based on the complementarity of the protein surface to the ligand.

4. Functional Annotation: Once potential active sites are identified, functional annotation methods are applied to further characterize these sites and predict their roles in biological processes.

5. Validation: Predicted active sites are experimentally validated through techniques like mutagenesis studies or enzyme assays to confirm their functional significance.

Active site prediction in protein complexes is essential for drug discovery, enzyme engineering, and understanding the molecular mechanisms underlying biological processes. It helps in designing targeted interventions and elucidating the structure-function relationships of proteins.

1.4. Comparing Protein-Protein Interaction Prediction and Active Site Prediction in Protein Complexes: Relationship and Distinctions

Protein-protein interactions (PPIs) and active site prediction in protein complexes are both essential aspects of protein structure and function analysis, but they focus on different aspects of protein behavior.

Relation:

1. Functionality: Both PPI and active site prediction are critical for understanding protein function. PPIs determine how proteins interact within a cellular context to carry out specific tasks, while active site prediction identifies regions within proteins that are involved in catalysis or binding interactions with other molecules.

2. Complex Formation: Protein-protein interactions often lead to the formation of protein complexes, where multiple proteins bind together to perform a biological function. Active sites within these complexes can mediate the interactions between the proteins, facilitating their function.

3. Biological Processes: Many biological processes rely on both protein-protein interactions and active sites within protein complexes. For example, enzymatic reactions often involve the formation of complexes between enzymes and their substrates, where the active site of the enzyme catalyzes the reaction.

Difference:

1. Focus: Protein-protein interaction prediction primarily focuses on identifying pairs or groups of proteins that physically interact with each other within a cellular environment. Active site prediction, on the other hand, specifically targets regions within proteins that are involved in catalytic activity or binding interactions with other molecules.

2. Methods: Different computational and experimental methods are used for predicting protein-protein interactions and active sites. Protein-protein interaction prediction methods often involve network analysis, co-evolutionary analysis, and machine learning algorithms. Active site prediction methods typically rely on structural analysis, sequence conservation analysis, molecular docking, and functional annotation techniques.

3. Scope: Protein-protein interaction prediction tends to focus on the global interactome, aiming to identify all possible interactions within a given proteome or biological system. Active site prediction, however, zooms in on specific regions within individual proteins to identify functional sites involved in catalysis or binding.

In summary, while both protein-protein interaction prediction and active site prediction in protein complexes are essential for understanding protein function, they differ in their focus, methods, and scope. However, they are interconnected as protein-protein interactions often occur within protein complexes where active sites mediate functional interactions.

1.5. Understanding Distinctions Between Active Binding Sites and Ligand Binding Sites in Proteins

In living things, proteins are the molecular workhorses that coordinate a wide range of biological functions that are vital to life. Specialized areas on their surfaces called binding sites, each of which plays a unique role in protein activity, are essential to their functionality. The two most important ones among them are the lig[1]and binding site and the active binding site, which provide vital interactions that control protein activity and propel biological events. We explore the essential distinctions between ligand binding sites, which tiny molecules attach to to control protein function, and active binding sites, which catalyze reactions. Through comprehending the distinct

attributes and functions of various binding sites, we can acquire a deeper grasp of the complex systems that oversee protein function and regulation.

Yes, there is a difference between an active binding site and a ligand binding site in protein complexes.

1. **Active Binding Site**:

- The active binding site, also known as the active site, is a region on the surface of a protein where catalytic activity occurs.
- This site is crucial for the protein's function, as it is involved in interactions with substrate molecules, facilitating biochemical reactions.
- Active sites often contain specific amino acid residues that directly participate in catalysis or substrate binding, and they are typically highly conserved across evolutionary related proteins with similar functions.
- In enzymes, the active site provides a microenvironment that promotes the catalytic reaction by stabilizing transition states or facilitating substrate binding and orientation.

2. **Ligand Binding Site**:

- A ligand binding site is a region on the protein surface where small molecules, called ligands, can bind reversibly.
- Ligands can include substrates, cofactors, inhibitors, or other molecules that interact with the protein and modulate its activity.
- Unlike the active site, which is specifically tailored for catalysis, ligand binding sites may have diverse functions, such as signal transduction, allosteric regulation, or structural stabilization.
- Ligand binding sites can be located at various locations on the protein surface, and they may undergo conformational changes upon ligand binding to induce functional responses.

While there can be overlap between active binding sites and ligand binding sites, especially in enzymes where the substrate itself acts as a ligand, they serve distinct roles in protein function. The active site is primarily associated with catalysis, while ligand binding sites are involved in a broader range of interactions and functions, including regulation and signaling (Figure 1).

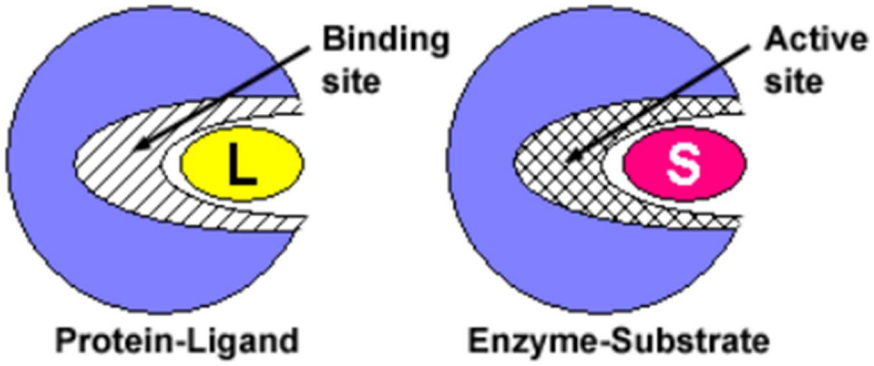

Figure 1: Compare the protein-ligand interaction to the enzyme-substrate interaction. Notice that both binding proteins and enzymes have binding sites for their ligands (L) and substrates (S), respectively. This area of the enzyme is called the active site because it also contains amino acids that are important for the conversion of substrate to product.

1.6. Predictive Tools and Visualization Methods for Protein Active Sites and Complexes

Various tools and methods are available to predict active sites on protein structures. Commonly used ones include:

- CASTp (Computed Atlas of Surface Topography of proteins): A web-based tool that identifies and measures the volume and area of protein binding sites, aiding in the localization of potential active sites.
- SiteMap: A module within the Schrödinger Suite that forecasts binding sites based on attributes like hydrophobicity, size, and charge, widely employed in structure-based drug design.
- PocketFinder: Another web-based tool for forecasting ligand-binding pockets on protein structures.
- DoGSiteScorer: Developed by BIOVIA, this software detects pockets in protein structures and scores them according to druggability.
- POCKET 2.0: Standalone software designed to pinpoint and analyze binding pockets in protein structures.
- LigSite: Identifies pockets in 3D structures by comparing their characteristics to known binding

sites.

- Concavity: An algorithm for projecting concave surface regions likely to be binding sites.
- Molegro Virtual Docker (MVD): A software package encompassing tools for cavity detection and binding site prediction.
- PLIP (Protein-Ligand Interaction Profiler): While its primary focus is protein-ligand interactions, it can also identify crucial residues involved in binding sites.
- AutoDock Vina: Primarily a docking software, but it can also uncover potential binding sites on protein structures.

Additionally, online servers such as Scfbio, Zhang Lab, and COFACTOR provide platforms for predicting active sites in protein complexes. Tools like PyMOL, ChimeraX, VMD, UCSF Chimera, Jmol, NGL Viewer provide researchers with the ability to visualize and analyze protein complexes in detail, aiding in the interpretation of complex molecular structures and interactions. Different instances of visualizing protein complexes in UCSF camera are shown below in Fig 2 (a, b, c, d).

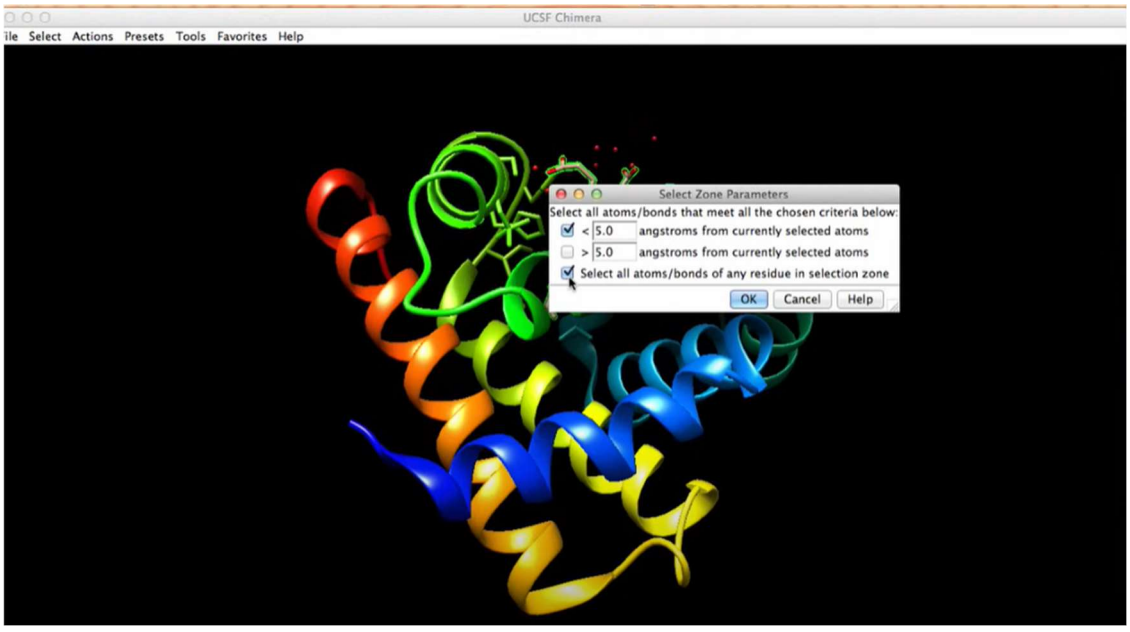

Figure 2 (a): Selecting the residues within 5 Angstrom in the vicinity of the haem group of myoglobin structure with PDB ID "1mbo" in UCSF Chimera

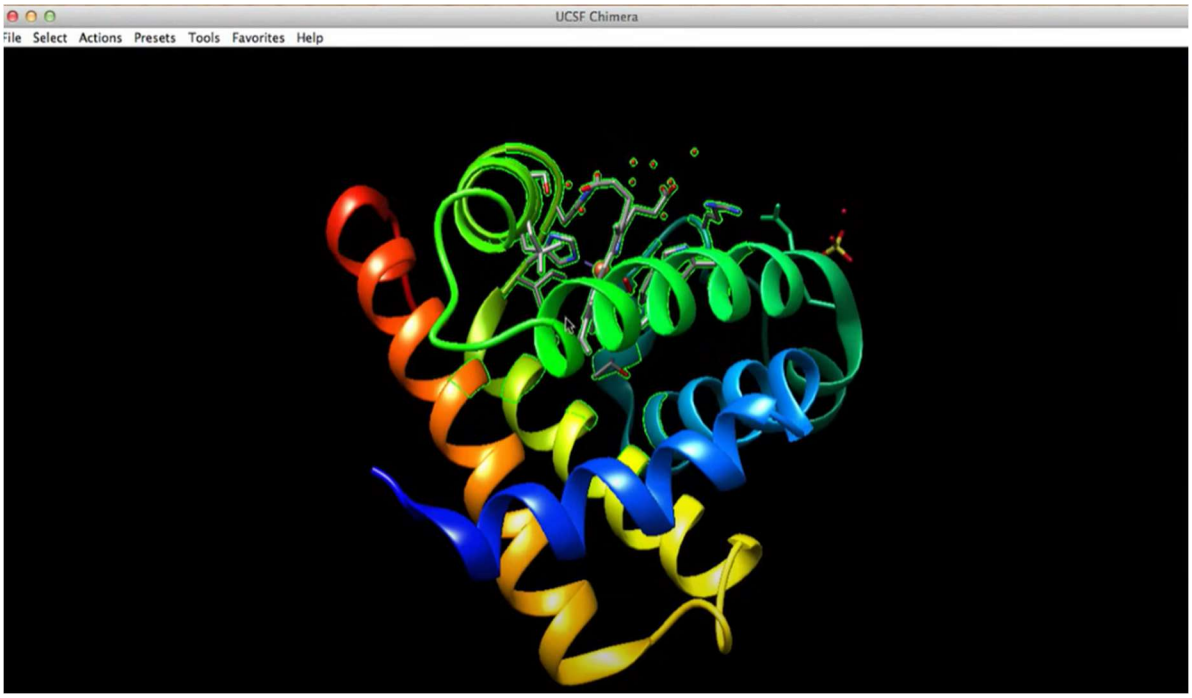

Figure 2(b): Highlighting the residues by element within 5 Angstrom (stick representation in ribbon structure within CPK coloring) in the vicinity of the haem group of myoglobin structure with PDB ID "1mbo" in UCSF Chimera

Figure 2(c): Structural Analysis of PDB ID "1mbo"

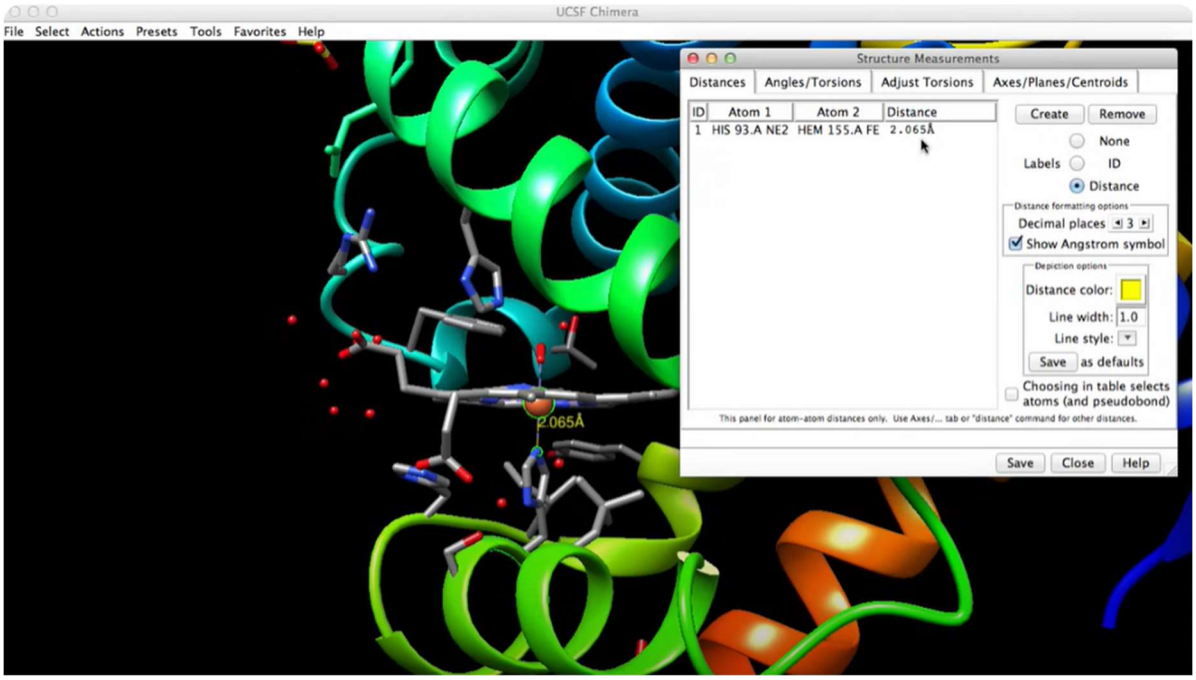

Figure 2(d): Measuring distances by selecting atoms in UCSF Chimera. Distance between Histidine and Heme group of 1mbo measured as 2.06 Angstrom.

1.7. Advancements in Predicting Active Sites in Protein Complexes: A Comprehensive Overview

Accurately predicting active sites in protein complexes is pivotal for understanding biological functions and facilitating drug discovery endeavors. Leveraging computational methodologies, researchers have made significant strides in this domain. Here, we present a comprehensive overview of various approaches proposed and employed for predicting active sites in protein complexes.

Free Energy-Based Simulations and Machine Learning-Based Scoring Functions:
Free energy-based simulations and machine learning-based scoring functions stand out as promising methodologies for predicting protein-ligand binding affinities. These methods offer the potential for highly accurate predictions, albeit with distinct computational strategies. Free energy-based simulations follow thermodynamic cycles, while machine learning-based scoring functions utilize feature-representation taxonomies. Recent advancements in deep learning have introduced hierarchical feature representations, further enhancing predictive capabilities.

Deep Learning Technologies for Active Site Binding Prediction:
Deep learning technologies have emerged as powerful tools for active site binding prediction in proteins, owing to their ability to extract complex patterns directly from raw data. Various deep learning architectures have been proposed and applied to this task:

1. **Convolutional Neural Networks (CNNs):** CNNs analyze protein sequences and structures to capture spatial dependencies and identify crucial motifs associated with active binding sites.

2. **Recurrent Neural Networks (RNNs):** RNNs, particularly Long Short-Term Memory (LSTM) networks, excel in sequential data analysis, capturing long-range dependencies and temporal relationships in protein sequences.

3. **Graph Neural Networks (GNNs):** GNNs model complex interactions in protein structures represented as graphs, effectively capturing the 3D spatial arrangement of atoms and residues.

4. **Attention Mechanisms:** Integrating attention mechanisms into deep learning models enables dynamic weighting of relevant regions or residues in proteins, enhancing active site prediction accuracy.

5. **Transformer-based Models:** Transformer architectures, including variants like BERT and GPT, leverage global contextual information and dependencies across protein sequences and structures for active site prediction.

6. **Protein Representation Learning:** Deep learning techniques for protein representation learning encode proteins into low-dimensional vector representations, preserving structural and functional information for downstream tasks.

7. **Multi-Modal Learning:** Integrating multiple modalities of protein data, such as sequences, structures, and evolutionary information, allows for more comprehensive analysis and prediction of active binding sites.

Recent Advances and Future Directions:

Recent studies have demonstrated the efficacy of deep learning technologies in predicting active sites in protein complexes. However, challenges persist, including the need for improved interpretability, scalability, and generalization across diverse protein structures. Future research directions may focus on addressing these challenges through advancements in model architectures, data integration, and algorithmic innovations.

In summary, the integration of computational methodologies and deep learning technologies offers promising avenues for advancing the prediction of active sites in protein complexes, thereby accelerating drug discovery and protein engineering endeavors.

1.8. A Detailed Discussion on PDB

What is PDB?

Protein Data Bank (PDB) is a database for the three-dimensional structural data of large biological

molecules, such as proteins and nucleic acids

• The data, is typically obtained by X-ray crystallography, NMR spectroscopy, or, increasingly, cryoelectron microscopy

• The data is freely accessible on the Internet via the websites of its member organizations (PDBe,PDBj, RCSB, and BMRB)

• The PDB is supervised by an organization called the Worldwide Protein Data Bank, wwPDB. A glance of the RCSB PDB Home page *(Figure 3)* is given below.

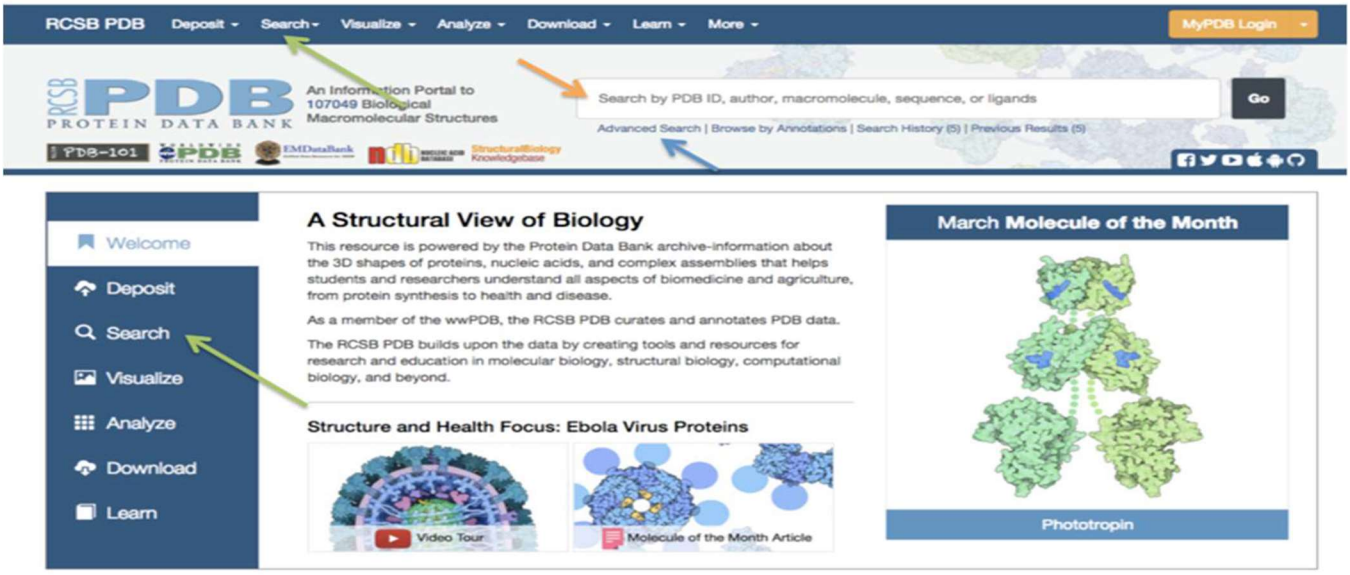

Figure 3: RCSB PDB Home page

Information extracted from PDB data

In the PDB archive, biological molecule coordinate files make up the majority of the data. Each protein's atoms are listed in these files together with their 3D spatial location. These files are accessible in PDB, mmCIF, and XML formats. The sequence and a lengthy list of the atoms and their coordinates are usually found after the huge "header" portion of text that summarizes the protein, the citation information, and the specifics of the structural solution. The experimental data used to establish these atomic coordinates is likewise preserved in the archive. A glance of the Atomic Coordinates file *(Figure 4)* downloaded in PDB format is given below.

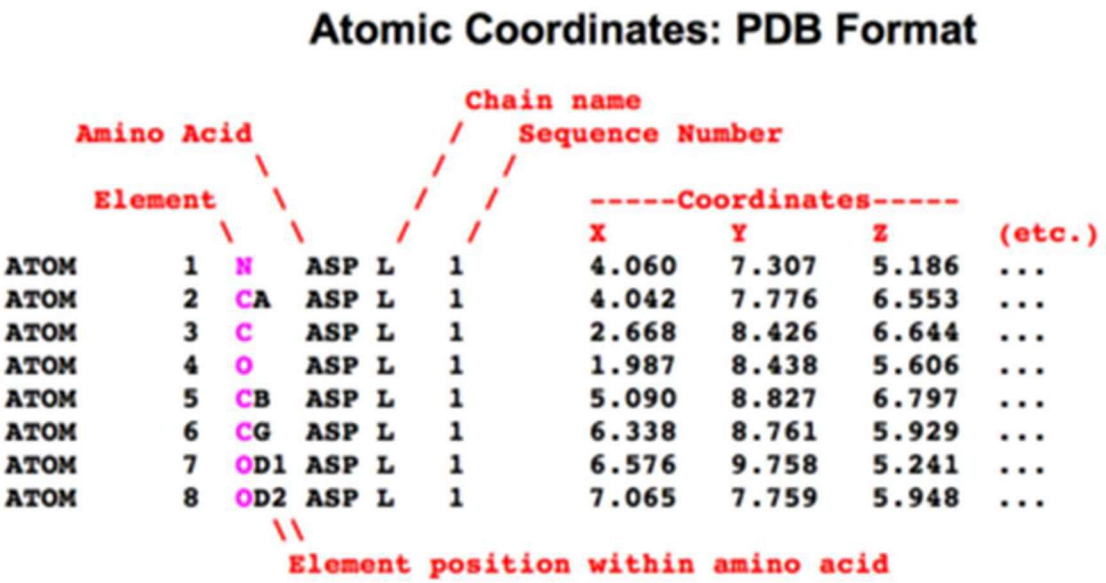

Figure 4: Atomic Coordinates file when downloaded from RCSB PDB website in PDB format

PDB Parsing: Hierarchical Representation of Structural Information

Model in Structure:

- A PDB file may contain multiple models, each representing a potential conformation or assembly of the macromolecular structure. Models are typically used to represent different states or stages of the structure, such as snapshots from molecular dynamics simulations or alternative conformations determined by X-ray crystallography.

Chain in Model:

- Within each model, biological macromolecules like proteins or nucleic acids are represented as chains. Chains are labeled with unique identifiers (typically alphabetic characters) and correspond to distinct polymer sequences within the structure. For example, in a protein complex, each protein component would typically be represented by a separate chain.

Residue in Chain:

- Chains consist of residues, which are the building blocks of proteins and nucleic acids. Residues are linked together to form polymer chains, with each residue representing a

specific amino acid or nucleotide. Residues are identified by their sequence number and may include additional information such as insertion codes to account for structural variations.

Atom in Residue:

Residues are composed of atoms, which represent the individual atomic constituents (e.g., carbon, nitrogen, oxygen) of the amino acids or nucleotides. Each atom is uniquely identified by its name and is associated with coordinates that specify its position in three-dimensional space. Additional information such as atom type and occupancy may also be provided.

This hierarchical representation of structural information in PDB files enables the detailed description and analysis of macromolecular complexes, facilitating a wide range of structural biology studies and computational analyses.

Comparative Analysis of mmCIF and PDB Formats in Structural Biology

mmCIF (macromolecular Crystallographic Information File) and **PDB** (Protein Data Bank) format are both widely used in structural biology to represent the three-dimensional structures of biological macromolecules like proteins and nucleic acids. While they serve similar purposes, they have some differences in how they structure and present the data.

1. **Data Organization**:

- mmCIF: Organizes data in a tabular format, with different tables representing various aspects of the macromolecular structure, such as atom coordinates, experimental details, and annotations.

- PDB: Uses a flat file structure with specific fields for each entry, making it more straightforward but potentially less flexible for representing complex structural information.

2. **Flexibility**:

- mmCIF: Offers flexibility in representing complex structures, accommodating large proteins with multiple chains, non-standard residues, and experimental details more comprehensively.

- PDB: Has a more rigid structure, which may limit its ability to represent certain types of structural information efficiently, especially for complex or non-standard structures.

3. **Completeness**:

- mmCIF: Tends to provide more comprehensive information, including experimental conditions, metadata, and annotations, alongside atomic coordinates.

- PDB: Focuses primarily on atomic coordinates and basic structural information, with fewer details about experimental methods and conditions.

4. **Usage**:

- mmCIF: Preferred by some researchers and databases for its structured and comprehensive representation of structural data, particularly for large and complex biomolecular structures.

- PDB: Remains widely used and supported across various software tools and databases due to its long-standing history and simplicity, especially for routine tasks and basic structural analyses.

In summary, while both mmCIF and PDB formats serve the purpose of representing macromolecular structures, they differ in their approach to data organization, flexibility, completeness, and usage. Researchers may choose between them based on their specific needs, the complexity of the structure, and the compatibility with existing software tools and databases.

1.9. Synergizing PDB Parsing with Active Site Prediction in Protein Complexes

1. **Acquiring Structural Data**:
    - PDB parsing entails extracting intricate structural details from Protein Data Bank (PDB) files, capturing the spatial arrangement of atoms in biological macromolecules. This includes atomic coordinates, chain designations, and residue properties.
2. **Identifying Protein Complexes**:
    - Leveraging PDB parsing facilitates the recognition and extraction of protein complexes from structural repositories. Through chain analysis and atomic interactions, researchers can delineate protein-protein interfaces, shedding light on complex composition and

architecture.

3. **Analyzing Residues at a Granular Level**:
    - Effective active site prediction demands a meticulous examination of protein structures at the residue level. PDB parsing enables the extraction of specific residues within protein complexes, honing in on critical functional sites crucial for ligand binding or catalytic functions.
4. **Extracting Informative Features**:
    - PDB parsing serves as the foundation for feature extraction in active site prediction models. Attributes like amino acid types, spatial coordinates, and local structural properties are derived from parsed PDB data, enriching the characterization of residues and their microenvironments.
5. **Harnessing Machine Learning and Deep Learning Techniques**:
    - Parsed PDB data form the basis for training machine learning and deep learning models tailored for active site prediction. These models leverage extracted features to discern patterns associated with active sites, facilitating the classification or regression of residues based on their functional roles.
6. **Integrating Structural Context:**
    - PDB parsing facilitates the incorporation of structural context into active site prediction algorithms. Insights into neighboring residues, protein-protein interactions, and conformational dynamics within complexes enrich prediction accuracy by capturing the local environment of potential active sites.
7. **Validating and Evaluating Predictions:**
    - Parsed PDB data play a pivotal role in the validation and evaluation of active site prediction methodologies. Predicted sites are benchmarked against experimentally determined active sites from the PDB, enabling rigorous assessments of prediction accuracy and performance metrics.

By harmonizing PDB parsing with active site prediction in protein complexes, researchers can harness structural insights to advance computational methodologies for identifying functional sites, thereby catalyzing drug discovery, enzyme engineering, and unraveling biological mechanisms.

Chapter 2
Literature Review

2.1. Computational Approach of Parsing PDB fetched from RCSB PDB by BioPython Library

Proteins participate in various essential processes in vivo via interactions with other molecules. Finding the residues involved in these interactions is crucial for drug discovery as well as offering biological insights for research on protein function. As a result, bioinformatics and computer-aided drug development have long been heavily researching protein-ligand binding site prediction [1]. In particular, amino acid residues at particular locations within the protein—typically found in pocket-like regions—are responsible for intermolecular interactions between proteins and ligands, such as tiny molecules. The term "ligand binding sites" (LBSs) refers to these particular essential amino acid residues. Molecular docking, drug-target interactions, chemical design, ligand affinity prediction, and even molecular dynamics have all shown a great deal of interest in LBSs[2-6].According to the definition given in BioLip, if the distance between any one of the atoms in the ligand molecule and at least one of the atoms in the amino acid residue of the protein does not exceed the sum of the radii of these two atoms plus 0.5 Å, the amino acid residue is regarded as a ligand binding residue[1].Through application programming interfaces (APIs), FTP downloads, and experimentally determined 3D structures of biomolecules integrated with over 40 external data resources, the US Research Collaboratory for Structural Bioinformatics Protein Data Bank (RCSB PDB) provides services to millions of unique users worldwide. The RCSB PDB offers data delivery services' architectural redesign, which expands on the PDBx/mmCIF data schemas that are currently in use. The PDB archive data may now be delivered efficiently thanks to new data access APIs (data.rcsb.org). An innovative GraphQL-based API offers an intuitive REST API in addition to customizable, declarative data retrieval. The PDB archive's sophisticated new search system, search.rcsb.org, smoothly combines several search methods. Text features, protein or nucleic acid sequences, small-molecule chemical descriptors, three-dimensional macromolecular structures, and sequence motifs can all be combined in a search [8]. The "Structural View of Biology" offered by RCSB PDB resources facilitates a greater comprehension of significant biological procedures and mechanisms in relation to PDB archive 3D structural data. The RCSB PDB website combines PDB data with several open access primary and derived data resources that have been created and made available by the community in order to achieve this goal. Individual Structure Summary pages provide access to external

data, which is frequently used to create searches and create tabular reports[9]. The PDB data fetched from RCSB PDB is thereby parsed to computationally retrieve the structural and sequential information of intra and inter-species protein data. The same is done by a collection of Python-based bioinformatics tools that is offered by the biopython project. Recently, a collection of modules pertaining to macromolecular structure were added to biopython. A PDB file parser for Biopython is now included, providing atomic information in a user-friendly yet robust data structure. The parser and data structure handle aspects like as anisotropic B factors, multiple models, insertion codes, atom and residue disorder (if point mutations are present in the crystal), that are frequently ignored or handled insufficiently by other tools. To find glaring mistakes, the parser also does some sanity checking. The Biopython distribution, which includes source code and documentation, can be downloaded for free from http://www.biopython.org under the terms of the Biopython license[7].

2.2. PPI prediction and Active site prediction

The majority of computational methods for predicting protein-protein interactions (PPIs) utilize machine learning (ML) techniques, with feature extraction and algorithm selection being critical for success. Shen et al. proposed the conjoint triad (CT) method, using the frequency of three consecutive amino acids (AAs) as features and an SVM for classification [11]. Guo et al.'s Auto Covariance (AC) method considered interactions of residues a few AAs apart, converting them to numerical values for SVM prediction [12]. Yang et al. [13] and Zhou et al. [14] employed local descriptors such as composition, transition, and distribution of AA triplets for prediction. You et al. used Wavelet Transform (WT), Discrete Cosine Transformation (DCT), and Global Encoding (GE) to extract features from AA sequences and fed these to a weighted sparse classifier [15]. A comparative study found Random Forest classifiers most effective among various methods, including Bayesian Networks and k-Nearest Neighbors (kNN) [32]. Wang et al.'s CNN-FSRF method combined a CNN with a Feature-Selective Rotation Forest (FSRF) classifier, using weighted feature values calculated by the chi-square method [31]. Tian et al. combined Pseudo-AA Composition (PseAAC), AC, and Encoding Based Grouped Weight (EBGW) methods for SVM classification [24]. To reduce training time, You et al. used a MapReduce framework for parallel and distributed SVM training [25]. Yang et al.'s Signed Variational Graph Auto Encoder (S-VGAE) utilized a graph convolutional network (GCN) encoder for converting proteins into embeddings, followed by a simple inner product decoder and a feed-forward neural network classifier [30]. Colonnese et al. employed the Spectral Graph Wavelet Transform (SGWT) for estimating local connectivity patterns, representing

the prediction task as a graph learning problem using Bayesian estimation [26]. Various feature extraction methods include multi-scale local feature representation (MLD) [27], multi-scale continuous and discontinuous (MCD) [22], and autocross-covariance (ACC) [12]. Halder et al. introduced JUPPI for obtaining high-quality negative data and performing three-level pairwise cross-validation using a random forest classifier [18]. Tsukiyama et al. used the LSTM model with word2vec, called LSTM-PHV, for predicting PPIs between humans and viruses, efficiently learning despite uneven sample ratios [16]. Yuan et al. proposed the GraphPPIS framework for PPI site prediction, transforming it into a graph node classification task using deep learning techniques [17]. SVM variants, Random Forests, neural networks, and gradient boosting decision trees are effective for PPI prediction, and ensemble learning methods can enhance prediction robustness by combining different feature sets and algorithms [18].

Although it's still a challenging subject, accurately predicting protein-ligand binding affinities can greatly aid in the drug discovery process. Several computer techniques have been developed to address the problem. Among these techniques, machine learning-based scoring functions and simulations based on free energy have the ability to produce precise forecasts. During research, two groups of approaches, using a feature-representation taxonomy for the machine learning-based scoring functions and several thermodynamic cycles for the free energy-based simulations. Also more recent deep learning-based predictions have been evaluated, which typically extract a hierarchy of feature representations. Comparatively, the advantages and disadvantages of the two kinds of approaches are explored, along with potential avenues for future development[10]. Halsana et al. propose the DensePPI model, which demonstrates superior performance compared to state-of-the-art methods across various evaluation metrics. The improved results highlight the efficiency of using an image-based encoding strategy for sequence information within a deep learning framework for PPI prediction. Furthermore, the enhanced performance of DensePPI on diverse test sets underscores its significance not only for predicting intra-species interactions but also for cross-species interactions. This paper introduces DensePPI, a novel deep convolutional strategy designed for predicting protein-protein interactions (PPIs) by leveraging 2D image maps generated from interacting protein pairs. The methodology includes a unique color encoding scheme that translates the bigram interactions of amino acids into RGB color space, thereby enhancing the model's learning and prediction capabilities. Overall, this study showcases the potential of DensePPI in advancing PPI prediction through innovative use of deep learning and image-based encoding strategies, marking a significant improvement over existing approaches[19].

Chapter 3
Proposed Methodology

The proposed methodology involves predicting active sites in protein-protein interactions (PPIs) using the DenseNet201 deep learning model. The process starts with retrieving and parsing PDB files to extract chain sequences and generate interatomic distance matrices at the residue level. These matrices are visualized as heatmaps using a color encoding strategy that maps bigram interactions of amino acids. Sub-images are generated using a sliding window approach and fed into the DenseNet201 model for training and testing. The classification strategy relies on averaging confidence scores and applying a threshold to determine active sites in PPI at residue level. Performance metrics are generated to robustly evaluate the model's predictive capabilities, ensuring a comprehensive analysis and validation framework.

3.1. Data Acquisition and Preparation

Downloading PDB/mmcif Files:

The initial phase of data acquisition involves the systematic retrieval of PDB files in PDB and mmcif format from the RCSB PDB database. Each PDB file is uniquely identified by its alphanumeric PDB ID, serving as a key for accessing the corresponding structural data. The `download_pdb` and ‘download_mmCIF’ functions, nested within the `urllib.request` module, orchestrates this retrieval process seamlessly. Through HTTP requests, the function accesses the database's vast repository, facilitating the efficient retrieval of PDB and cif files with minimal latency.

Retrieval of UniProt IDs from MMCIF Parsing followed by PDB Parsing:

Subsequent to PDB retrieval, the focus shifts to the extraction of UniProt IDs, pivotal in annotating and characterizing protein structures. The `get_uniprot_ids` function, a constituent of the `Bio.PDB.MMCIFParser` module, plays a pivotal role in this endeavor. Leveraging the macromolecular Crystallographic Information File (mmCIF) format, renowned for its comprehensive representation of structural data, the parsing process unfolds meticulously.

Upon receiving a PDB file, the `MMCIFParser` and 'PDBParser' meticulously dissect its contents, extracting crucial metadata including UniProt IDs associated with the protein structures. This intricate parsing mechanism delves deep into the structural nuances encapsulated within the PDB files, extracting key identifiers imperative for subsequent analyses. Through a systematic traversal of the mmCIF-formatted data, UniProt IDs are meticulously cataloged, paving the way for their seamless integration into downstream analyses and annotations.

Retrieval of Protein Chain Information by PDB Parsing:

1. **Retrieving Chain Count and Names:**

- The function calls the `get_uniprot()` function to obtain UniProt IDs and strand IDs associated with the given PDB identifier.
- Strand IDs are combined into a dictionary (`chains_by_uniprot`) where the keys are UniProt IDs and the values are lists of associated strand IDs.
- Using the `combinations` function from the `itertools` module, the function generates all possible combinations of chains from the extracted strand IDs.

Table 1 shows the comprehensive data parsing from PDB and UniProt sources

Data Source	Information Parsed
From PDB Data	
PDB Files	Downloaded PDB files using their unique identifiers
Distance Matrix	Calculated distance matrix between pairs of protein chains
Chain Lengths	Determined the lengths of protein chains
Interaction Classification	Classified interactions between protein chains into gold, silver, and non-interacting sets based on calculated distances
From UniProt Data (mmCIF files)	
UniProt IDs	Extracted UniProt accession IDs associated with protein structures
Strand IDs	Extracted strand IDs from mmCIF files, representing different chains within the protein structure

Table 1. Comprehensive Data Parsing from PDB and UniProt Sources

2. **Processing Chain Combinations**:

For each chain combination, the function performs the following steps:

- Downloads the corresponding PDB file using `download_pdb()` and parses it using `PDBParser()` to obtain the protein structure.
- Calculates the coordinates of alpha carbon atoms ('CA') for each chain and generates a distance matrix between them using the `generate_distance_matrix()` function.
- Evaluates the lengths of the protein chains and checks for insufficient lengths (less than 16 alpha carbon atoms).

3. **Interaction Classification and Dataset Summary**:

- Classifies the interactions between chain pairs based on the overall minimum distance. Interaction types include **gold (distance < 5 Å), silver (5 Å ≤ distance ≤ 10 Å), and non-interacting (distance > 10 Å).**
- Constructs a dictionary (`entry`) containing information such as PDB ID, UniProt IDs, chain combinations, and interaction classifications.
- The dataset summary is being prepared that includes distances and interaction classifications as mentioned above and the same is written to a master CSV file specified by `master_csv_filename`, using the `csv.DictWriter()` module.
- There might be some PDB records from the input PDB bulk, that cannot be processed so only the processed records are dumped in a separate file (`processed_ids_filename`) for reference.

These functions collectively retrieve UniProt IDs and chain information from mmCIF files and analyze protein-protein interactions, providing valuable insights into the structural characteristics of proteins as shown in Figure 5.

	A	B	C	D	E	F	G	H	I	J
1	PDB ID	Uniprot pair IDs	Chain pairs	Chain lengths of chain pairs	Gold Data Set chain pair	Gold Dataset Distance Value (Inter-atomic Euclidean distance < 5 Å)	Silver Data Set chain pair	Silver Dataset Distance Value (5 Å ≤ Inter-atomic Euclidean distance ≤ 10 Å)	Non-interacting protein chain pairs	Non-Interacting Dataset Distance Value (Inter-atomic Euclidean distance > 10 Å)
51	6SF3	P37023, O95393	A, B	76, 104			A & B	5.433411598		
52		Q15561, Q8N9Y4	L, A	16, 205			L & A	6.193887711		
53		Q15561, Q8N9Y4	L, B	16, 213					L & B	13.87968445
54		Q15561, Q8N9Y4	M, A	16, 205					M & A	13.89482975
55		Q15561, Q8N9Y4	M, B	16, 213			M & B	6.282700539		
56	6SEO	Q15561, A6NEQ2	L, A	16, 205			L & A	6.195219994		
57	6SF1	P37023, O95393	A, B	76, 104			A & B	5.305719376		
58		P24941, P20248	C, D	261, 257	C & D	4.851637363				
59		P24941, P20248	C, B	261, 258					C & B	11.46545887
60		P24941, P20248	A, D	262, 257					A & D	12.08349609
61		P24941, P20248	A, B	262, 258	A & B	4.875376701				
62	6SJM	P19793, Q15596	A, B	212, 13			A & B	6.526742458		
63	6SJZ	P30419, Q96NN9	E, A	8, 391					E & A	25.59142494
64		P30419, Q96NN9	E, B	8, 389	E & B	4.001670361				
65		P30419, Q96NN9	A, F	391, 7	A & F	4.278978348				
66		P30419, Q96NN9	F, B	7, 389					F & B	25.41548729
67	6SK2	P30419, Q96NN9	F, A	8, 392					F & A	27.05076599
68		P30419, Q96NN9	F, B	8, 390	F & B	4.321949959				
69		P30419, Q96NN9	A, D	392, 8	A & D	4.007955551				

Figure 5: Computational Analysis of Protein-Protein Interactions: Structural Insights and Characteristics

3.2. Methodology for Distance Calculation and Matrix Generation

1. **Distance Calculation Function**:

The function `calculate_distance(coord1, coord2)` is designed to compute the Euclidean distance between two sets of coordinates, `coord1` and `coord2`. The Euclidean distance represents the straight-line distance between two points in space. It is calculated using the formula:

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{n}(coord1_i - coord2_i)^2}$$

Formula for Calculating Euclidean Distance

This function employs the NumPy library's `np.linalg.norm()` function, which efficiently computes the Euclidean norm.

2. **Distance Matrix Generation**:

The function `generate_distance_matrix(chain1, chain2)` is responsible for generating a distance

matrix that encapsulates the pairwise distances between each coordinate pair in the sets `chain1` and `chain2`.

The methodology employed involves the following steps:

- **Initialization**: A square matrix of zeros is initialized, with dimensions corresponding to the lengths of `chain1` and `chain2`.
- **Iterative Computation**: Nested loops are employed to iterate over every pair of coordinates in `chain1` and `chain2`. For each pair, the `calculate_distance()` function is invoked to compute the distance, which is then stored in the appropriate position of the distance matrix.
- **Result**: The resulting matrix encapsulates the distances between all pairs of coordinates in `chain1` and `chain2`.

3. **Saving Distance Matrix to CSV:**

The function `save_distance_matrix_to_csv(filename, matrix)` is responsible for persisting the computed distance matrix to a CSV (Comma-Separated Values) file. This operation enables further analysis or visualization of the distance data. The function utilizes the NumPy library's `np.savetxt()` function, allowing the matrix to be efficiently written to the specified file in CSV format. The delimiter parameter ensures proper formatting of the CSV file, with commas separating individual values.

This methodology elucidates the systematic approach employed to calculate distances between coordinates and subsequently generate and store the resulting distance matrix. It forms an integral part of the overall analysis pipeline for protein structure characterization.

3.3. Computational Approach for Generation of Protein-Protein Interaction Distance Statistics

Once distances between protein chains are computed, various statistical measures are derived to characterize the spatial relationships within protein structures. This step involves the following procedures:

1. Computes various distance statistics including:
- Minimum distance
- Maximum distance

- Average distance
- Standard deviation of distances

2. Provides quantitative insights into the proximity and distribution of protein chains within structures.
3. Facilitates the identification of key structural features and interaction patterns.

The distance statistics of protein-protein interaction is shown in Figure 6 below

Processing PDB ID: 6SEN

PDB ID	Uniprot IDs	Chain Combinations	Chain Lengths	Min Distance	Max Distance	Avg Distance	Std Distance
6SEN	Q8N9Y4	L, M	16, 16	50.09290313720703	80.0859375	68.11367671191692	6.475562166765172
	Q15561, Q8N9Y4	L, A	16, 205	6.193887710571289	50.36792755126953	25.520631960556575	8.46275381677224
	Q15561, Q8N9Y4	L, B	16, 213	13.879684448242188	79.78758239746094	53.879479508444474	10.631983360385464
	Q15561, Q8N9Y4	M, A	16, 205	13.894829750061035	77.90702056884766	53.71449374222174	10.696292468457338
	Q15561, Q8N9Y4	M, B	16, 213	6.282700538635254	50.40880584716797	25.444259910656253	8.444938476689932
	Q15561	A, B	205, 213	4.486532688140869	74.564697265625	40.728356466868844	12.225306711829537

Processing PDB ID: 6SEO

PDB ID	Uniprot IDs	Chain Combinations	Chain Lengths	Min Distance	Max Distance	Avg Distance	Std Distance
6SEO	Q15561, A6NEQ2	L, A	16, 205	6.195219993591309	50.90471649169922	25.422548953352905	8.484600073139037

Processing PDB ID: 6SF1

PDB ID	Uniprot IDs	Chain Combinations	Chain Lengths	Min Distance	Max Distance	Avg Distance	Std Distance
6SF1	P37023, O95393	A, B	76, 104	5.305719375610352	69.28028869628906	33.12193432112454	13.633931023174945

Processing PDB ID: 6SG4

PDB ID	Uniprot IDs	Chain Combinations	Chain Lengths	Min Distance	Max Distance	Avg Distance	Std Distance
6SG4	P24941	C, A	261, 262	4.534749984741211	106.1712875366211	55.429230947708724	16.528376914302253
	P24941, P20248	C, D	261, 257	4.851637363433838	78.36771392822266	41.171099918413304	12.165386246368897
	P24941, P20248	C, B	261, 258	11.465458869934082	112.5570068359375	68.20252703811683	16.51817317340537
	P24941, P20248	A, D	262, 257	12.08349609375	113.85488891601562	68.71120125922995	16.63941770760128
	P24941, P20248	A, B	262, 258	4.875376701354805	78.50322723388672	41.22303147721315	12.258984789199173
	P20248	D, B	257, 258	18.373516082763672	112.1229248046875	65.00248620408918	17.370013670986374

Processing PDB ID: 6SJM

PDB ID	Uniprot IDs	Chain Combinations	Chain Lengths	Min Distance	Max Distance	Avg Distance	Std Distance
6SJM	P19793, Q15596	A, B	212, 13	6.526742458343506	49.22702407836914	25.73833990979437	8.26317520555778

Processing PDB ID: 6SJZ

PDB ID	Uniprot IDs	Chain Combinations	Chain Lengths	Min Distance	Max Distance	Avg Distance	Std Distance
6SJZ	P30419, Q96NN9	E, A	8, 391	25.5914249420166	96.96086120605469	57.15929945533538	12.676619887692969
	Q96NN9	E, F	8, 7	48.1519660949707	81.8750991821289	64.11359460013253	8.37038590891826
	P30419, Q96NN9	E, B	8, 389	4.001670360565185 5	49.21124267578125	23.636793939344066	8.2293240278242
	P30419, Q96NN9	A, F	391, 7	4.27897834777832	48.38760757446289	22.95249874085721 6	7.817924534537114
	P30419	A, B	391, 389	4.396884918212891	110.3869857788086	50.029541706544954	15.307364240970763
	P30419, Q96NN9	F, B	7, 389	25.41548728942871	94.85710906982422	55.77279985366916	12.31302379565813

Figure 6: Comprehensive Analysis of Protein Chain Interactions Distance Statistics

3.4. Batch Processing of PDB IDs

To streamline the analysis of multiple protein structures, a batch processing approach is adopted, allowing efficient exploration and characterization of diverse protein datasets.

1. Iterates through a list of PDB IDs stored in a text file, processing each ID sequentially.
2. Executes UniProt ID retrieval, distance statistics generation, and data recording for each PDB ID.
3. Ensures systematic and comprehensive analysis of protein structures at scale.

By following this comprehensive methodology, researchers can gain valuable insights into the structural organization, biological context, and spatial relationships within protein structures. The integration of computational techniques with biological data analysis enables a deeper understanding of protein-protein

interactions and molecular mechanisms, contributing to advancements in fields such as structural biology, drug discovery, and protein engineering. The workflow diagram of protein data processing is shown in Figure 7.

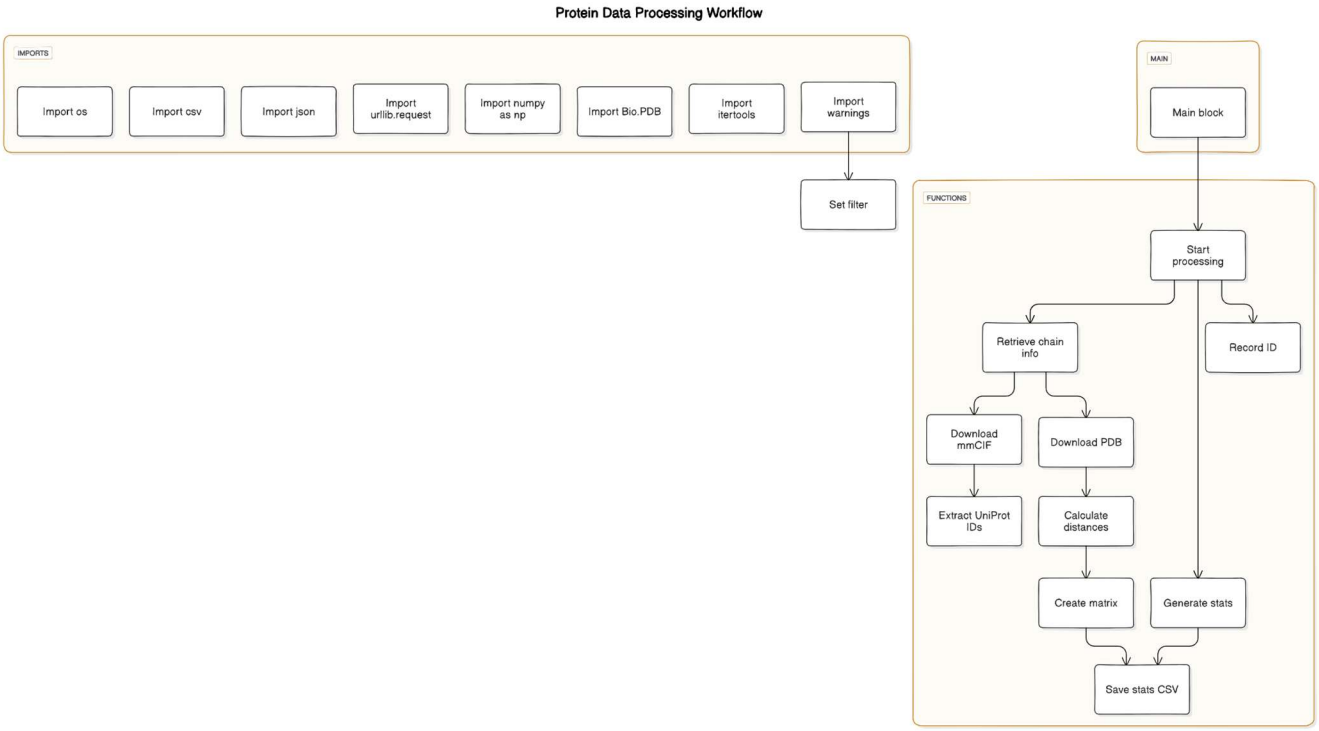

Figure 7: Workflow diagram of Protein Data Processing

3.5. Validation using distance matrix heatmaps

In the pursuit of assessing the accuracy and reliability of protein structure predictions, validation plays a pivotal role in confirming the fidelity of computational models. One such validation technique involves the generation of distance matrix heatmaps, which offer a visual representation of the spatial relationships between alpha carbon atoms in protein chains. These heatmaps serve as a powerful tool for scrutinizing the consistency between predicted and experimentally determined protein structures.

Methodology:

The validation process begins with the retrieval of protein structures from the Protein Data Bank (PDB)

using their unique identifiers. Subsequently, the alpha carbon coordinates of each protein chain are extracted, forming the basis for distance matrix generation. Leveraging the numpy library, distance matrices are computed for pairwise combinations of alpha carbon coordinates between different protein chains. The process flow diagram of protein structure validation using distance matrix heatmaps is shown in Figure 8.

For each pair of protein chains within a given structure, a distance matrix heatmap is generated using seaborn and matplotlib libraries. The heatmap visualizes the distances between alpha carbon atoms, with a color gradient representing varying degrees of proximity. Specifically, the **heatmap employs a color spectrum ranging from green (indicating shorter distance PPIs) to red (indicating longer distances PPIs)**, allowing for intuitive interpretation of spatial relationships within protein structures (Figure 9).

Heatmap Analysis:

The generated distance matrix heat maps provide valuable insights into the structural integrity and conformational consistency of protein models. By visually comparing the heatmaps of predicted protein structures with experimentally resolved structures, discrepancies and inaccuracies can be readily identified. Consistent patterns of distances between alpha carbon atoms across multiple chains signify robust and reliable structural predictions, corroborating the validity of computational models.

Validation of Correctness of proposed methodology:

The inclusion of distance matrix heatmaps as a validation methodology within the thesis serves multiple purposes. Firstly, it demonstrates a rigorous approach to evaluating the accuracy of computational protein structure predictions. Secondly, it provides visual evidence supporting the conclusions drawn from computational analyses. Lastly, it showcases the proficiency in utilizing advanced computational tools and techniques for structural biology research.

In summary, distance matrix heatmaps serve as a powerful validation tool, augmenting the credibility and trustworthiness of computational protein structure predictions. Through their integration into the thesis, a comprehensive validation framework is established, underscoring the meticulousness and rigor applied to

structural biology research endeavors.

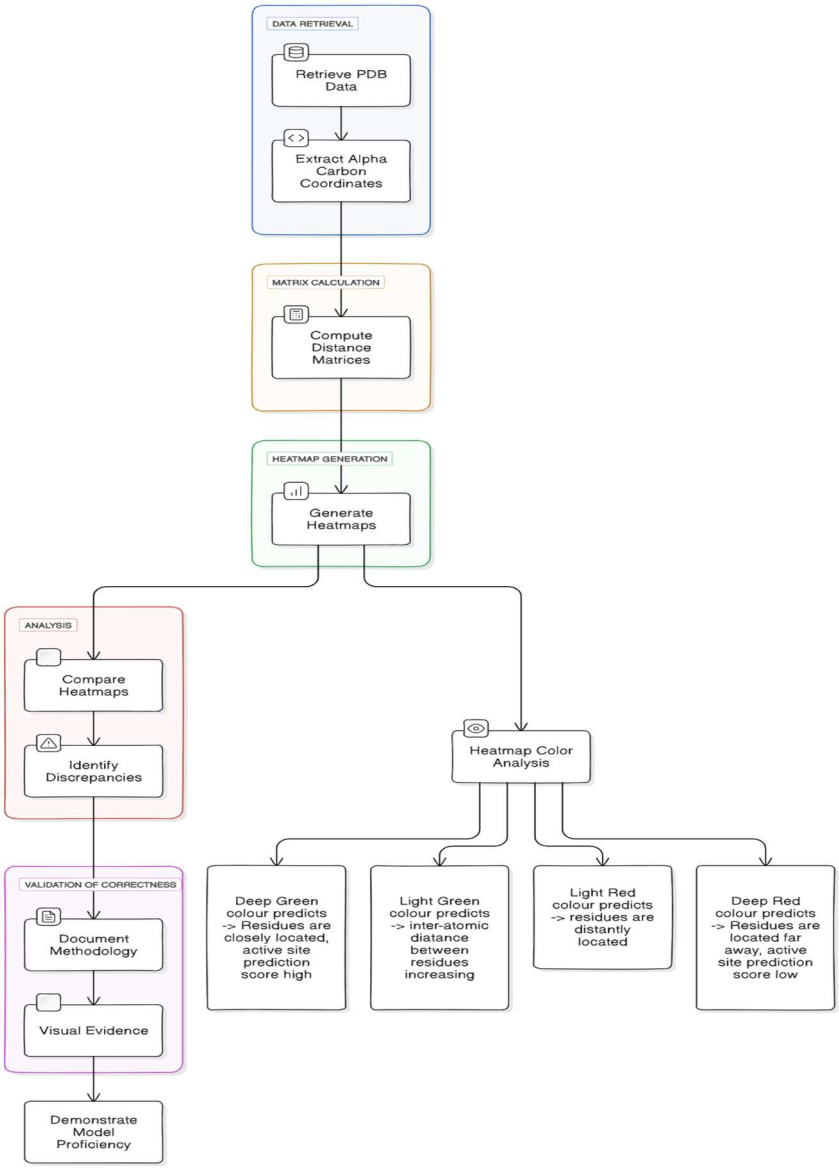

Figure 8: Process flow diagram of Protein structure validation using distance matrix heatmaps

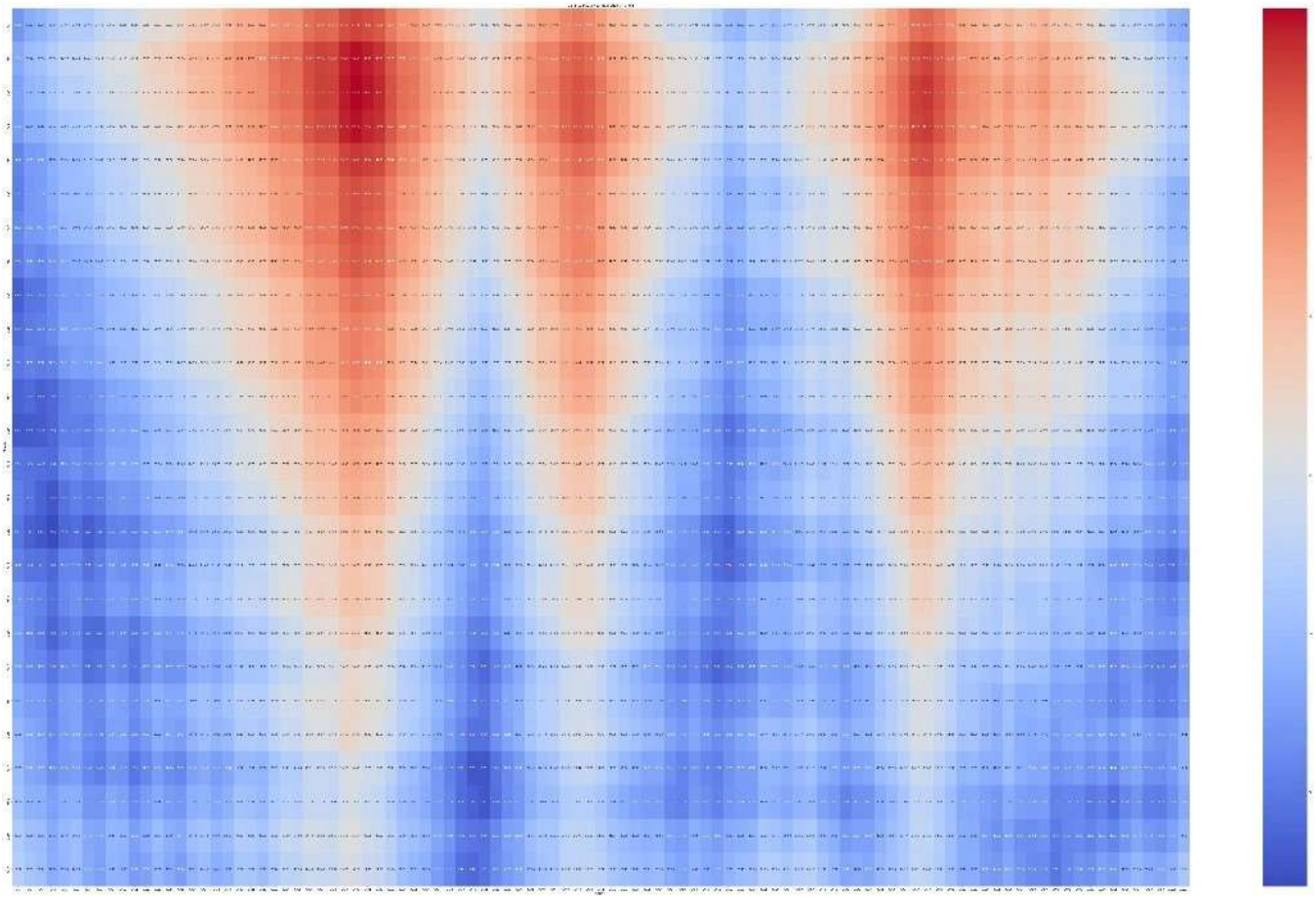

Figure 9: Computationally generated heatmaps with color spectrum ranging from green (indicating shorter distance PPIs) to red (indicating longer distances PPIs)

3.6. Revised Procedure of Protein Data Acquisition, Dataset Preparation and Data Validation by Normalization technique

Retrieval and Analysis of Protein Structure Data

In the provided Python scripts, we conduct a thorough analysis of protein structures retrieved from the Protein Data Bank (PDB) database. This analysis encompasses the extraction of structural information, calculation of pairwise distances between protein chains, and the generation of heatmaps for visualizing the spatial relationships within the protein structures.

Retrieval and Preprocessing of Protein Structures

The initial phase involves the systematic retrieval of PDB and mmCIF files corresponding to the

specified PDB IDs. These files serve as the primary sources of structural data for subsequent analyses. The `download_pdb` and `download_mmCIF` functions facilitate the downloading of PDB and mmCIF files, respectively, from the RCSB PDB database. Once downloaded, these files are parsed using the Biopython library to extract essential structural details, including chain IDs, UniProt IDs, and sequence information.

Calculation of Pairwise Distances

With the extracted structural data, pairwise distances between alpha carbon atoms (`CA`) of protein chains are calculated. The `generate_distance_matrices_combinations` function iterates over all possible combinations of protein chain pairs within each structure. For each pair, the Euclidean distance between corresponding alpha carbon coordinates is computed to construct a distance matrix. This matrix quantitatively represents the spatial separation between residues in the protein chains.

Importance and Methodology of Normalization

Normalization of the distance matrix is imperative to ensure the comparability and interpretability of distance values across different protein structures. Normalization scales the distance values to a common range, typically between 0 and 1, by accounting for variations in the magnitudes of distances observed in different protein structures. This process facilitates the identification of structural patterns and relationships independent of absolute distance values.

To achieve normalization, the following formula is applied to the distance matrix:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where:

- X represents the original distance matrix.
- X_{min} is the minimum distance value in X.
- X_{max} is the maximum distance value in X.
- $X_{normalized}$ denotes the normalized distance matrix.

Normalized distance matrices are generated successfully.

Generation of Normalized Heatmaps

Normalized distance matrices are visualized using heatmaps to provide intuitive insights into the spatial proximity of protein residues. The `generate_distance_matrices_combinations` function generates heatmaps from the normalized distance matrices using the Seaborn library in Python. The workflow diagram of normalized heatmap generation is shown in Figure 10. These heatmaps depict the relative distances between residues, with warmer colors like red indicating greater separation and cooler colors like green representing closer proximity (Figure 11).

Comprehensive analysis and visualization of protein structures, coupled with normalization techniques, offer a robust yet uniform framework for validating and interpreting structural data. These methodologies enhance our understanding of protein folding, interactions, and functional dynamics, thus contributing to advancements in structural biology and drug discovery research.

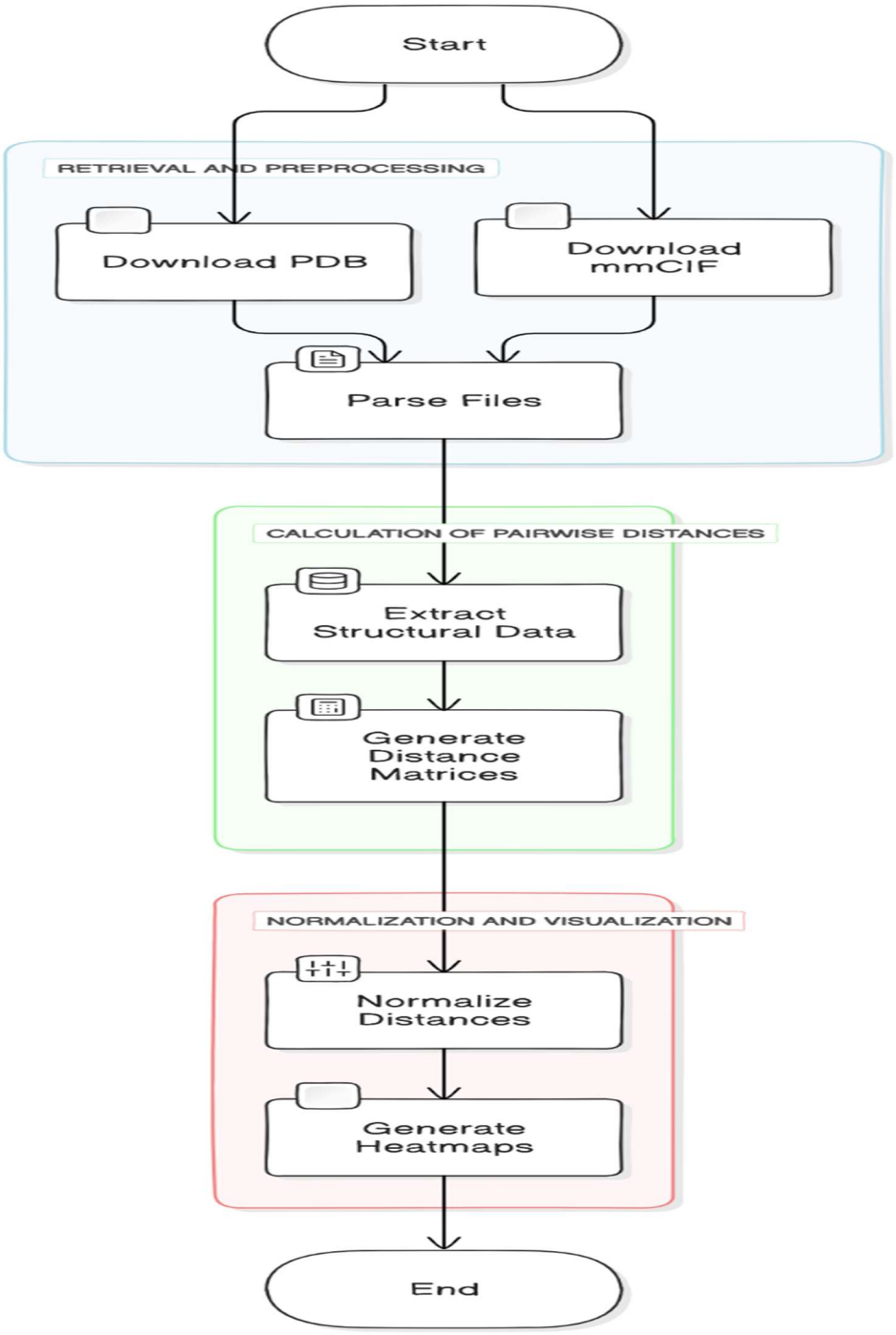

Figure 10: Workflow diagram of Revised Procedure of Protein Data Acquisition, Dataset Preparation and Data Validation by Normalization technique

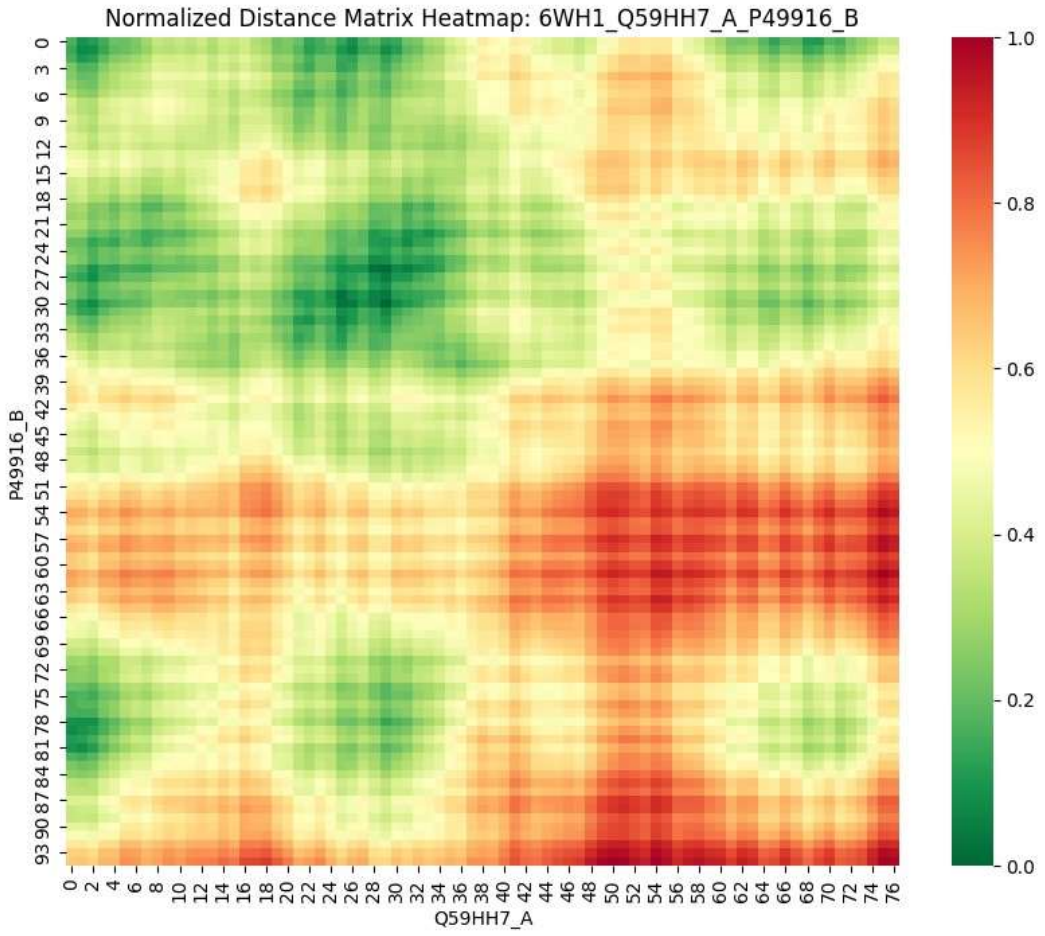

Figure 11: Computationally generated normalized heatmaps with color spectrum ranging from green (indicating shorter distance PPIs) to red (indicating longer distances PPIs)

3.7. Computational Approach Protein Sequence Information Generation

The Python script facilitates the retrieval and analysis of protein structure sequences from the Protein Data Bank (PDB) database. Utilizing the Biopython library, the script downloads PDB files corresponding to specified PDB IDs and extracts essential sequence information from these files. This analysis aids in understanding the primary structure of proteins, including amino acid sequences and associated UniProt IDs.

Retrieval and Processing of Protein Structure Sequences

The script begins by reading PDB IDs from a text file, allowing for bulk retrieval of sequence information for multiple protein structures. For each PDB ID, the script retrieves the corresponding PDB file and parses it using a PDBParser object from Biopython. Additionally, the script utilizes an MMCIFParser to extract UniProt IDs associated with the protein chains, providing valuable metadata for further analysis. The process flow diagram of protein sequence information generation is shown in Figure 12. The output of computational generation of protein sequence information by parsing PDB data is shown in Figure 13.

Analysis and Presentation of Sequence Information

Upon successful retrieval and parsing, the script generates a summary of sequence information for each protein chain within the given PDB IDs. This includes the UniProt ID and amino acid sequence for each chain, facilitating downstream analyses such as sequence alignment and functional annotation.

The systematic retrieval and analysis of protein structure sequences offer valuable insights into the primary structure of proteins, enriching our understanding of protein function and evolution. By leveraging computational tools and publicly available databases, researchers can expedite the process of sequence analysis and accelerate discoveries in structural biology and bioinformatics.

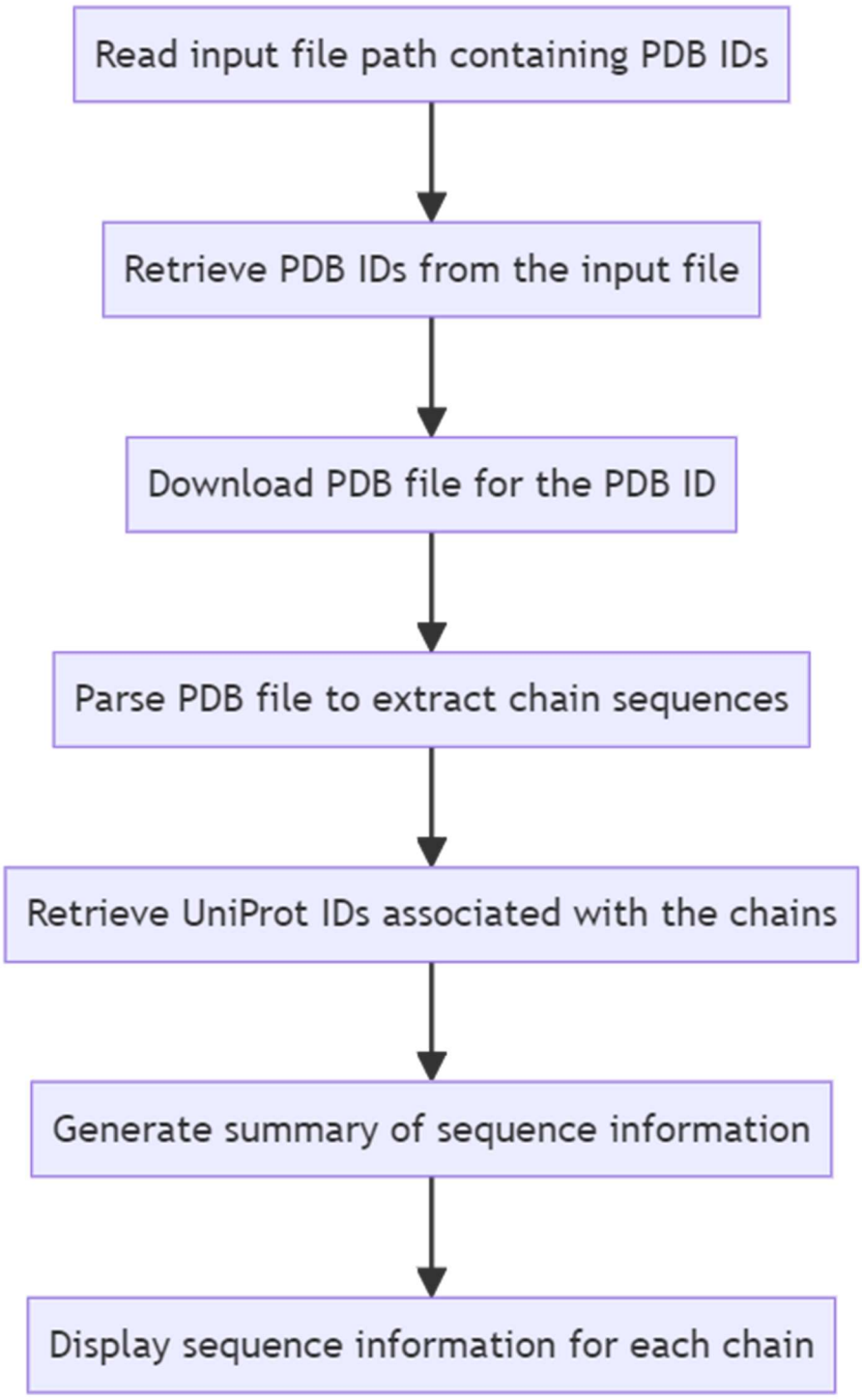

Figure 12: Process Flow diagram of Protein Sequence Information Generation

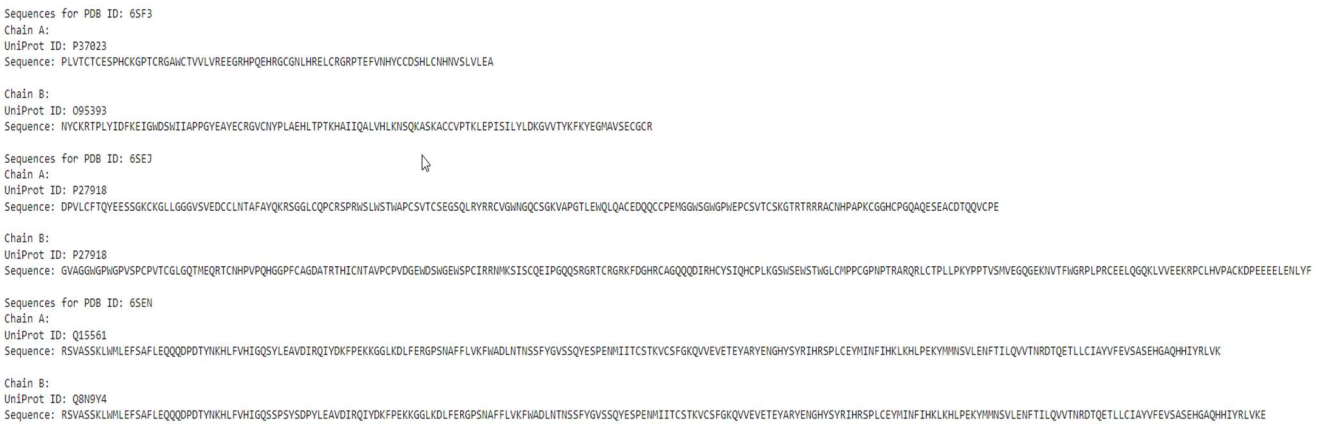

Figure 13: Computational Generation of Protein Sequence Information by parsing PDB data

3.8. Active Site Prediction in Protein-Protein Interaction at Residue Level

This study outlines a comprehensive workflow for predicting active sites in protein-protein interactions (PPIs) using the DenseNet201 deep learning model. The process begins with initializing parameters and importing necessary libraries, followed by meticulous data preparation, defining transformations, and configuring global device settings. A custom PPIModel class incorporating DenseNet201 is defined for model architecture.

Training involves loading training data, initializing the dataloader, model, and optimizer, and performing iterative forward and backward passes to optimize the model. Checkpoints are saved periodically to ensure intermediate results can be resumed. The prediction phase includes loading test data, initializing the dataloader, and evaluating the model to compute loss and accuracy.

Metrics generation is a critical phase where positive and negative test dictionaries are used to generate final predictions. A confusion matrix is calculated to assess performance, followed by the computation of accuracy and other crucial metrics such as Matthew's Correlation Coefficient (MCC), sensitivity, specificity, and precision. The workflow concludes with saving the predictions and metrics, providing a robust framework for PPI active site prediction.

This structured approach enhances the reliability and interpretability of active site predictions, contributing significantly to the field of computational biology and bioinformatics.

1. **Data Preparation**:
    - Input Data: The workflow begins with parsing PDB data to extract interatomic distance matrices at the residue level and sequence-level information of protein chains.
    - Output Format: The parsed data is stored in CSV format and consists of both positive and negative PPIs.
2. **Structure Information**:
    - Pickle Files: Pickle files containing positive and negative PPI datasets are generated for further processing.
    - Heatmaps: Based on the PPI distance visualization, heatmaps using a color spectrum (green indicating shorter distances, red indicating longer distances) are created for future validation purposes.
3. **Color Encoding Strategy**:
    - Bigram Interaction Mapping: A color encoding strategy maps the bigram interaction of amino acids to a lookup table. This encoding helps in visualizing interactions between amino acids.

      This work introduces a color encoding scheme to represent protein sequences as images. The method estimates the interaction possibilities between amino acid (AA) pairs using a color map (CMAP). Each AA is assigned one of 26 distinct colors in the RGB spectrum, ensuring equal importance for each AA interaction. The color matrix CMAP of dimension 26 × 26 represents interactions between residues, defined as:

$$\mathrm{CMAP}_{ij} = \left[\sqrt{\frac{r_i^2+r_j^2}{2}}, \sqrt{\frac{g_i^2+g_j^2}{2}}, \sqrt{\frac{b_i^2+b_j^2}{2}}\right]$$

where $[r_i, g_i, b_i]$ and $[r_j, g_j, b_j]$ are the unique RGB colors for residues i and j. The final image, CMAPP, representing interactions between the AA sequences of two proteins $P1$ and $P2$, is created by combining CMAP entries for all AA pairs:

$$\mathrm{CMAPP}_{ij} = \mathrm{CMAP}_{ij} \; \forall i \in \mathrm{AA}[P1] \text{ and } \forall j \in \mathrm{AA}[P2]$$

These images are used for training and validating a deep-learning model for protein-protein interaction prediction (Figure 14).

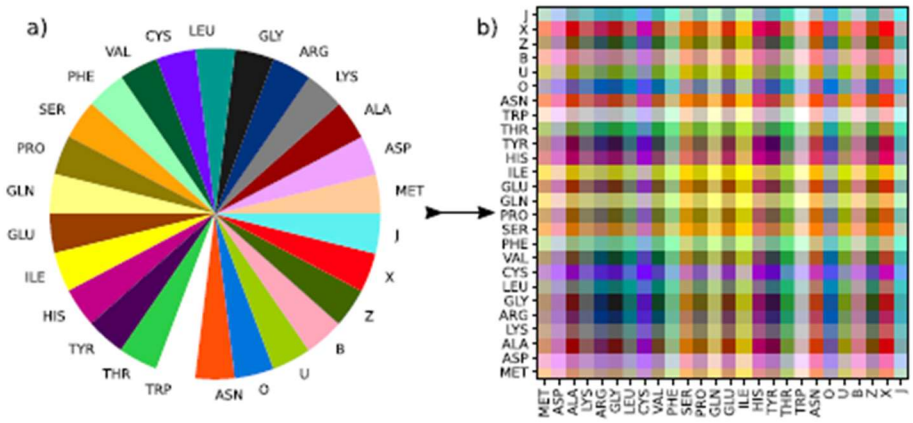

Figure 14: Assigned colours and colour maps to produce images from AAs in PPIs. a) Colour assigned to each amino acid and the unrecognizable amino acids. b) The colour map used for generating images from two proteins using amino acid pairs [19].

4. **Image Generation**:

- PNG Images: PNG images are generated for both positive and negative PPIs using the encoded color strategy.

5. Sub-Image Generation:

- Sliding Window Approach: A sliding window approach with fixed dimensions (32x32 pixels) and stride (2 pixels) is used to create sub-images from the original PPI images.
- Sub-Image Generation for Training and Testing: The generated sub-images are categorized into positive and negative PPIs and are prepared for training and testing the model. Figure 15 shows the sub-image generation by sliding window approach.

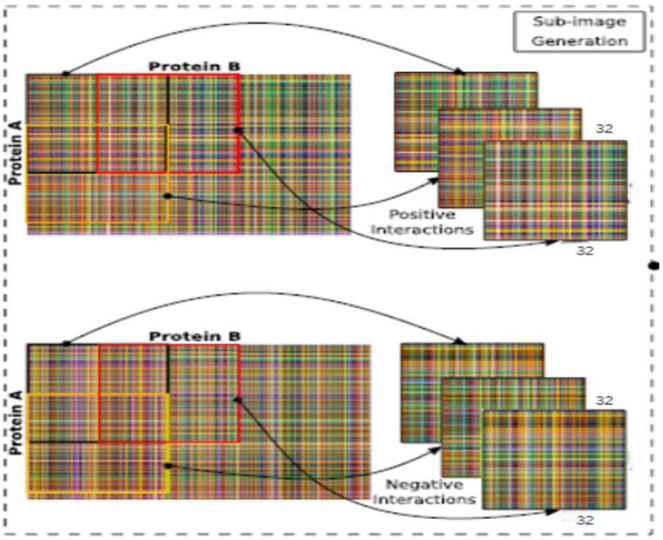

Figure 15: Sub-Image Generation by Sliding Window approach [19]

6. **Deep Learning Model**:
    - DenseNet201 Architecture: The sub-images are fed into the DenseNet201 model. The DenseNet201 architecture is depicted, showcasing its dense blocks, transition layers, and output layer.
7. **Classification Strategy**:
    - Confidence Score and Thresholding: The classification strategy involves averaging the confidence scores for each sub-image. A threshold of 0.5 is applied to determine the final class label for the original PPI.
    - Performance Metrics: Prediction evaluation and performance metrics are generated based on the model's outputs, helping assess the accuracy and reliability of the predictions.

Overall Workflow diagram for Active site prediction in PPI at residue level has been shown in Figure 16.

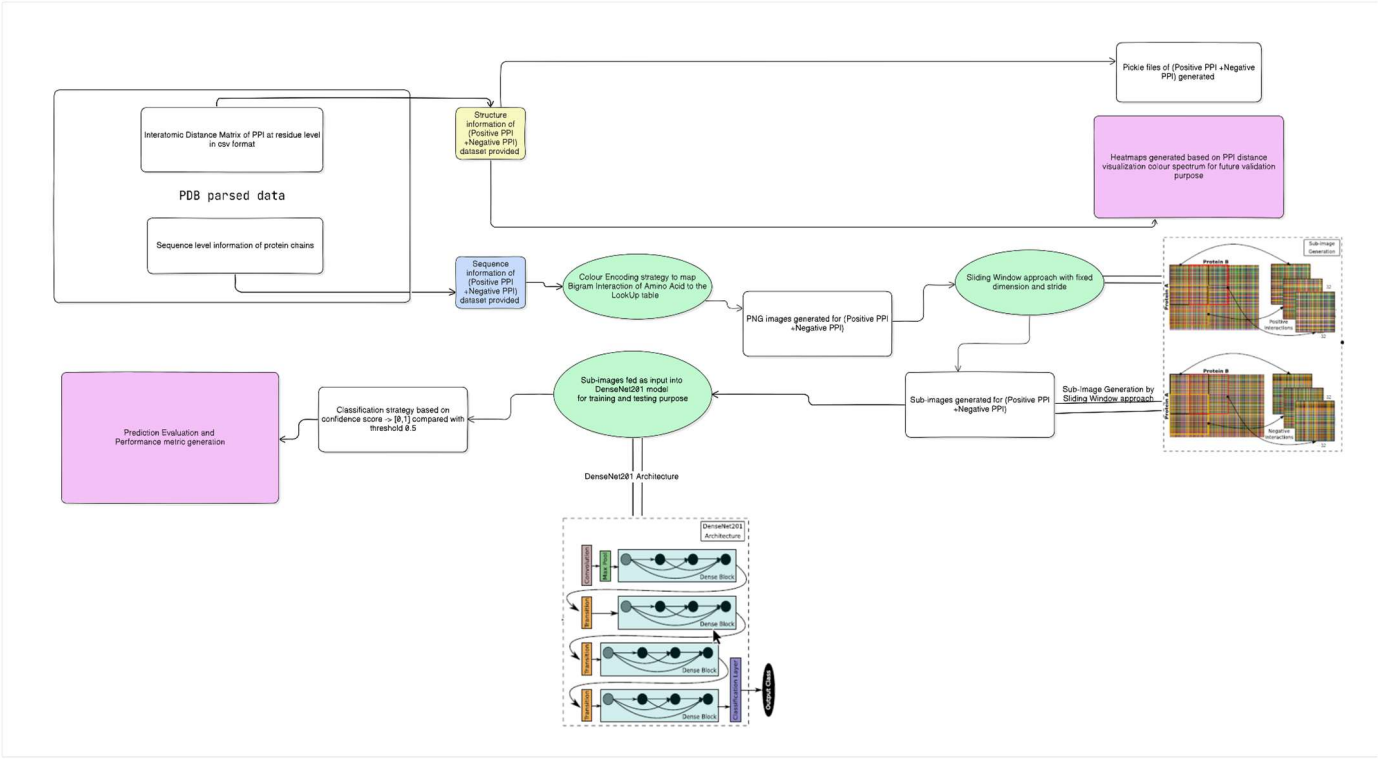

Figure 16: Overall Workflow diagram for Active site prediction in PPI at residue level

3.9. Brief Discussion on DenseNet201 Architecture

1. Convolutional Neural Networks (CNNs):

- Dominant approach in visual object detection.
- Examples: LeNet5 (5 layers), VGG (19 layers), Highway Networks, Residual Networks (ResNets) surpassing 100 layers.

2. Challenges in Deep CNNs:

- Vanishing-gradient problem.
- Information and gradients diminish as they pass through multiple layers.

3. DenseNet:

- Developed by Huang et al. to address deep CNN challenges.
- Each layer is connected to every other layer in a feed-forward manner.
- DenseNet architecture results in L(L+1)/2 direct connections in an L-layer network.

- Layers use feature maps from all previous layers as inputs, promoting feature reuse and improving feature propagation.

Schematic Diagram of DenseNet201 Architecture is shown in Figure 17

4. DenseNet Layers:

- Each layer defined as: $L_i = f_i([L_0, L_1, ..., L_{i-1}])$.
- Composite function f_i: Batch Normalization (BN), Rectified Linear Unit (ReLU), and 3×3 Convolution (Conv).

5. Efficiency and Compactness:

- DenseNet allows easy access to earlier layer feature maps, making the network smaller and more compact.
- Transition layers (convolution and pooling) are used between dense blocks for downsampling.

6. Growth Rate Adjustment:

- Controls the number of feature maps to manage computational expense.
- Growth rate k: Feature maps in the ith layer = $k \times (i - 1) + k_0$.

7. DenseNet201 Usage:

- Chosen architecture: DenseNet201.
- Excluded protein pairs with sequence length < 128 to avoid squeezing or expanding images.

8. Training Configuration:

- Average pooling between layers.
- Learning rate: 0.001.
- Momentum: 0.9.
- Binary classification with categorical crossentropy loss function.

9. Categorical Crossentropy Loss Function:

- Defined as:

$$J_{CCE} = -\frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N} y_j^i \times \log(h_\theta(x_j, i))$$

  - N: Number of training examples.
  - C: Number of classes.
  - x_j: jth input vector.
  - y_j^i: Target label for class i.
  - h_θ: Model with network weights θ.

10. Optimizer and Training:
    - Stochastic Gradient Descent (SGD) optimizer.
    - Mini-batch gradient descent with batch size of 32 and 10 epochs.

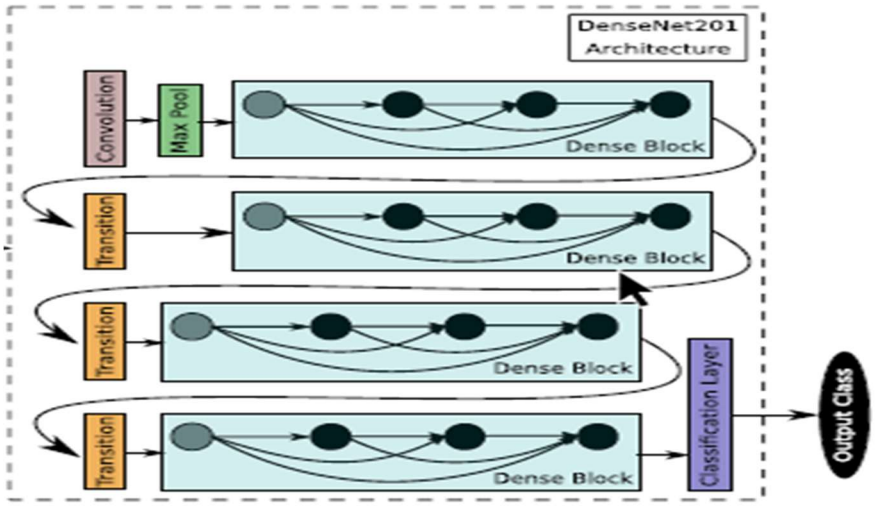

Figure 17: Schematic Diagram of DenseNet201 Architectur

Chapter 4
Experiment And Evaluation

The purpose of this project is to develop and evaluate a deep learning model for active site prediction in protein-protein interaction (PPI) prediction at residue level using PDB parsed data. This methodology section outlines the process of data preparation, model training, evaluation, and metrics generation. The project leverages the DenseNet-201 architecture from the torchvision library for image classification tasks.

4.1. Data Preparation

1. **Data Pre-processing and Transformations**:

Images generated in png format by a color encoding strategy that maps bigram interactions of amino acids to a lookup table from 40 sets of positive protein-protein interactions and 40 sets of protein-protein negative interactions computationally generated from parsing PDB data. Sub - images generated in png format by sliding window with specified stride and dimension from these images. The image data undergoes preprocessing transformations to convert them into tensors suitable for model input. Specifically, the images are converted to grayscale and then to tensors using the `transforms` module from `torchvision`.

2. **Data Loading**:

The training and testing datasets are loaded using the `ImageFolder` class from `torchvision.datasets`. The data is then loaded into data loaders for batching, shuffling, and efficient processing during training and evaluation.

4.2. Model Architecture

The model architecture used in this project is a modified DenseNet-201:

- **DenseNet-201**:

DenseNet-201 is a convolutional neural network that connects each layer to every other layer in a feed-forward fashion. For this project, the final classification layer of DenseNet-201 is replaced

with a linear layer with two output features, corresponding to the binary classification task (interaction vs. no interaction).

4.3. Training

The training process involves the following steps:

1. **Model Initialization**:

   The `PPIModel` class initializes the DenseNet-201 model and modifies its classifier.

2. **Optimizer and Criterion**:

   The **Stochastic Gradient Descent (SGD)** optimizer is used with a learning rate of $(1\times10^{(-4)})$ and a momentum of 0.9. The loss function used is **Cross Entropy Loss**, which is suitable for classification tasks.

3. **Training Loop**:

   The training loop iterates over a specified number of epochs (10 in this case). For each epoch, the model is set to training mode and processes the training data in batches. The optimizer updates the model weights based on the computed gradients. The loss and accuracy are tracked and printed for each epoch.

4. **Checkpointing**:

   The `ModelCheckpoint` class is used to save the model state after each epoch, ensuring that the best-performing model is retained.

4.4. Evaluation and Metrics Calculation

The evaluation process involves the following steps:

1. **Model Loading**:

   The trained model is loaded from the saved checkpoints.

2. **Prediction Loop**:

   The model is set to evaluation mode and processes the test data in batches. Predictions are made for each batch, and the loss and accuracy are tracked.

3. **Metrics Calculation**:

The true labels and predicted labels are collected to compute various performance metrics. These metrics include accuracy, confusion matrix, Matthews correlation coefficient (MCC), sensitivity, specificity, and precision.

4.4.1. Discussion on Training /Prediction Loss/Accuracy and Confusion Matrix Calculation

Training Phase:

The model was trained over 10 epochs. During training, both the loss and accuracy were monitored and the best model was saved based on the highest accuracy achieved. The training accuracy steadily increased over the epochs, reaching a final value of 98.78%. The corresponding loss also decreased, indicating a successful training process.

Testing Phase:

The model was tested using a separate test dataset. The prediction accuracy on the test set was 95.44%, and various performance metrics were calculated.

Training and Testing Accuracy:

- The training accuracy improved steadily over the epochs, starting from 84.83% in the first epoch and reaching up to 98.78% in the tenth epoch.
- The testing accuracy achieved was 95.44%, indicating the model's ability to generalize well to unseen data.

Training Loss:

- The training loss consistently decreased over the epochs, starting from 0.3681 in the first epoch and reducing to 0.0277 by the tenth epoch. This indicates effective learning and convergence of the model.

Accuracy and Loss Over Epochs has been shown graphically in Figure 18

Epoch	Training Loss	Training Accuracy
1	0.3681	0.8483
2	0.1651	0.9457
3	0.0806	0.9735
4	0.0525	0.9821
5	0.0410	0.9855
6	0.0367	0.9869
7	0.0338	0.9869
8	0.0290	0.9885
9	0.0270	0.9894
10	0.0277	0.9878

Table 2: Training Loss and Accuracy for Each Epoch

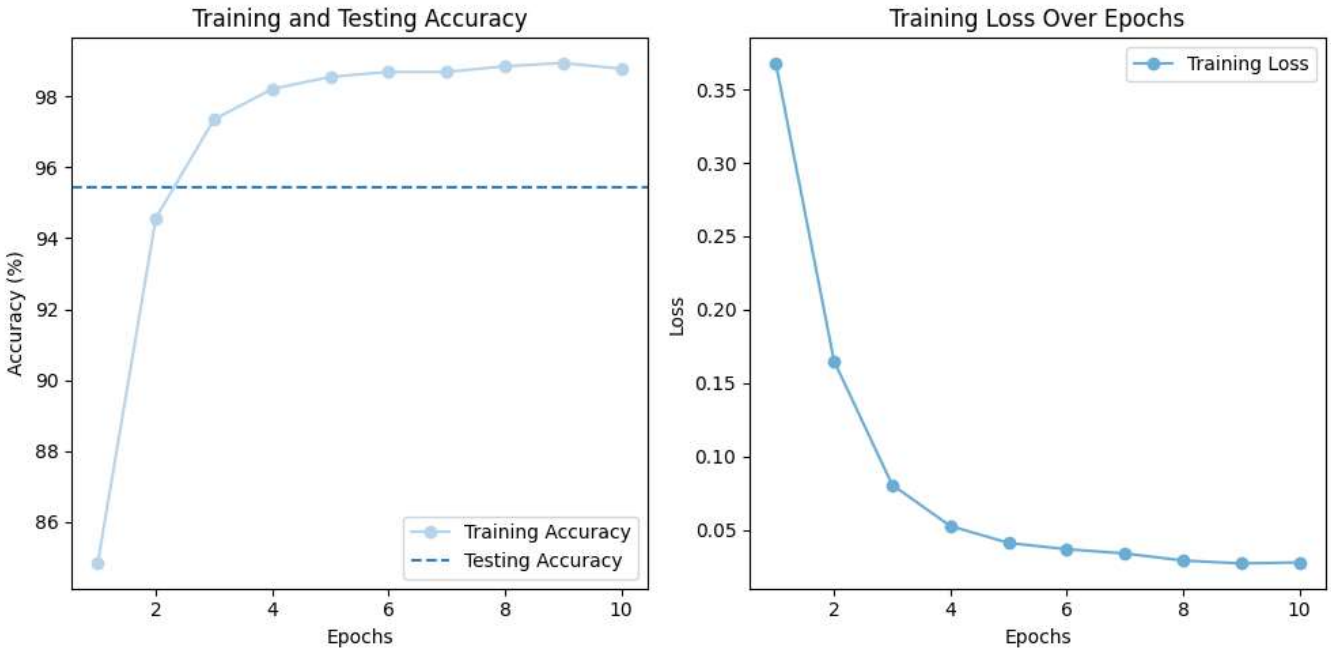

Figure 18: Model Training Performance: Accuracy and Loss Over Epochs

Prediction Loss:

This is the loss calculated during the prediction phase of your model. It measures how well your model performs in terms of discrepancy between predicted and actual values. In this case, the prediction loss is 0.1414.

Prediction Accuracy:

This is the accuracy of your model's predictions on a test dataset. It measures the proportion of correctly classified instances out of the total instances. Here, the test accuracy is 0.95445, indicating that about 95.44% of the predictions were correct.

Now, let's discuss True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), which are commonly used to evaluate the performance of a classification model.

- **True Positive (TP)**: These are the cases where the model correctly predicts the positive class. In other words, the instances that are actually positive and are correctly classified as positive by the model.
- **True Negative (TN):** These are the cases where the model correctly predicts the negative class.

In other words, the instances that are actually negative and are correctly classified as negative by the model.

- **False Positive (FP):** These are the cases where the model incorrectly predicts the positive class when it's actually negative. In other words, the instances that are actually negative but are incorrectly classified as positive by the model.
- **False Negative (FN):** These are the cases where the model incorrectly predicts the negative class when it's actually positive. In other words, the instances that are actually positive but are incorrectly classified as negative by the model.

The confusion matrix (Figure 18) is calculated to derive true negatives (TN), false positives (FP), true positives (TP), and false negatives (FN). The confusion matrix summarizes the performance of the model as follows:

	Predicted Negative	Predicted Positive
Actual Negative	$TN(5142)$	$FP(142)$
Actual Positive	$FN(85)$	$TP(92)$

Table 3: Summarization of TN, FP, TP, FN values

This matrix shows that the model correctly identified 5142 true negatives and 92 true positives, while making 142 false positive and 85 false negative errors.

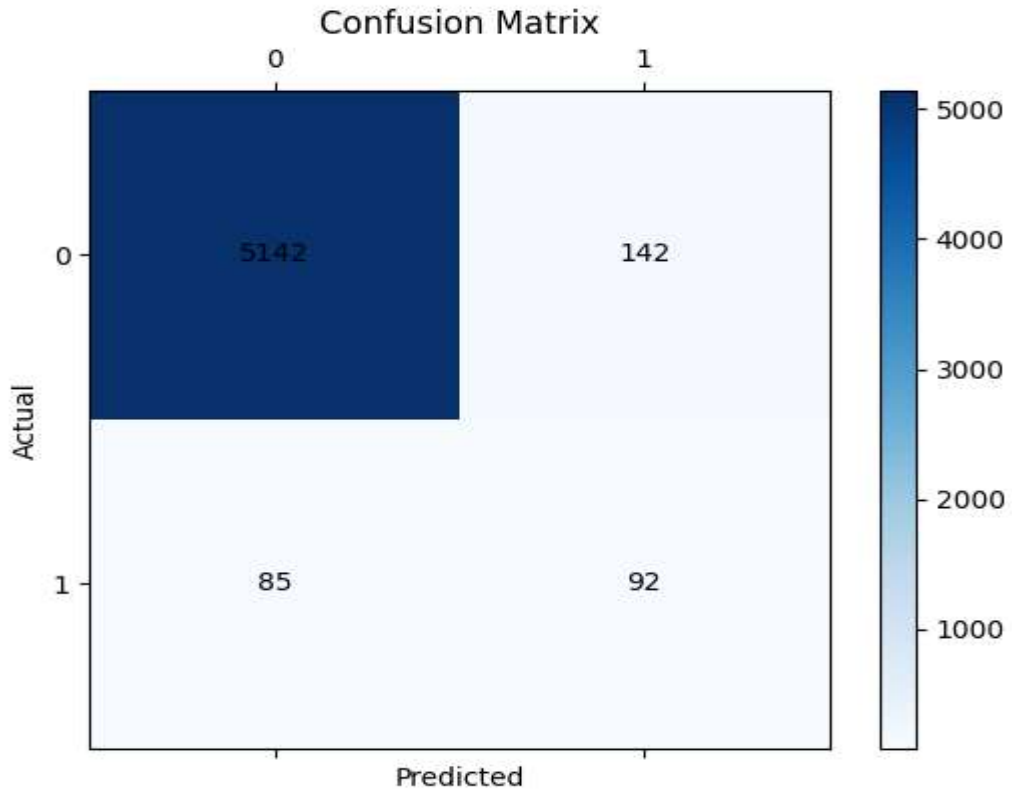

Figure 19: Confusion Matrix Heatmap

These metrics are crucial for understanding the strengths and weaknesses of your model. They are often used to calculate other evaluation metrics like accuracy, precision, recall, and F1-score.

Performance Metrics:

Various performance metrics are computed from the confusion matrix and predictions:

- **Accuracy**:

The ratio of correct predictions to total predictions. The model achieved an overall accuracy of 95.84%, indicating that the model correctly predicts both positive and negative interactions most of the time.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{92 + 5142}{92 + 5142 + 142 + 85} = \frac{5234}{5461} \approx 0.9584$$

-**Matthews Correlation Coefficient (MCC)**:

A balanced measure that takes into account true and false positives and negatives. The MCC value of 0.4310 indicates a moderate correlation between the actual and predicted

classes, suggesting that the model performs better than random guessing but still has room for improvement.

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} = \frac{(92 \times 5142) - (142 \times 85)}{\sqrt{(92 + 142)(92 + 85)(5142 + 142)(5142 + 85)}} \approx 0.4310$$

- **Sensitivity (Recall)**:

The ratio of true positives to the sum of true positives and false negatives. The recall of 51.98% indicates that the model correctly identifies about half of the actual positive interactions, reflecting moderate performance.

$$\text{Sensitivity} = \frac{TP}{TP + FN} = \frac{92}{92 + 85} \approx 0.5198 \ (51.98\%)$$

- **Specificity**:

The ratio of true negatives to the sum of true negatives and false positives.A high specificity of 97.31% shows that the model is excellent at correctly identifying negative interactions, which is particularly valuable when the cost of false positives is high.

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{5142}{5142 + 142} \approx 0.9731 \ (97.31\%)$$

- **Precision**:

The ratio of true positives to the sum of true positives and false positives. With a precision of 39.32%, the model shows that a significant proportion of the positive predictions are incorrect, highlighting the need for improvement in identifying positive interactions.

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{92}{92 + 142} \approx 0.3932 \ (39.32\%)$$

Summary of Results:

- **Accuracy**: 0.9584
- **Matthews Correlation Coefficient (MCC)**: 0.4310
- **Sensitivity (True Positive Rate)**: 0.5198
- **Specificity (True Negative Rate)**: 0.9731
- **Precision:** 0.3932

Evaluation of DensePPI Model Performance Metrics showing Accuracy, MCC, Sensitivity, Specificity, Precision score has been shown in Figure 20.

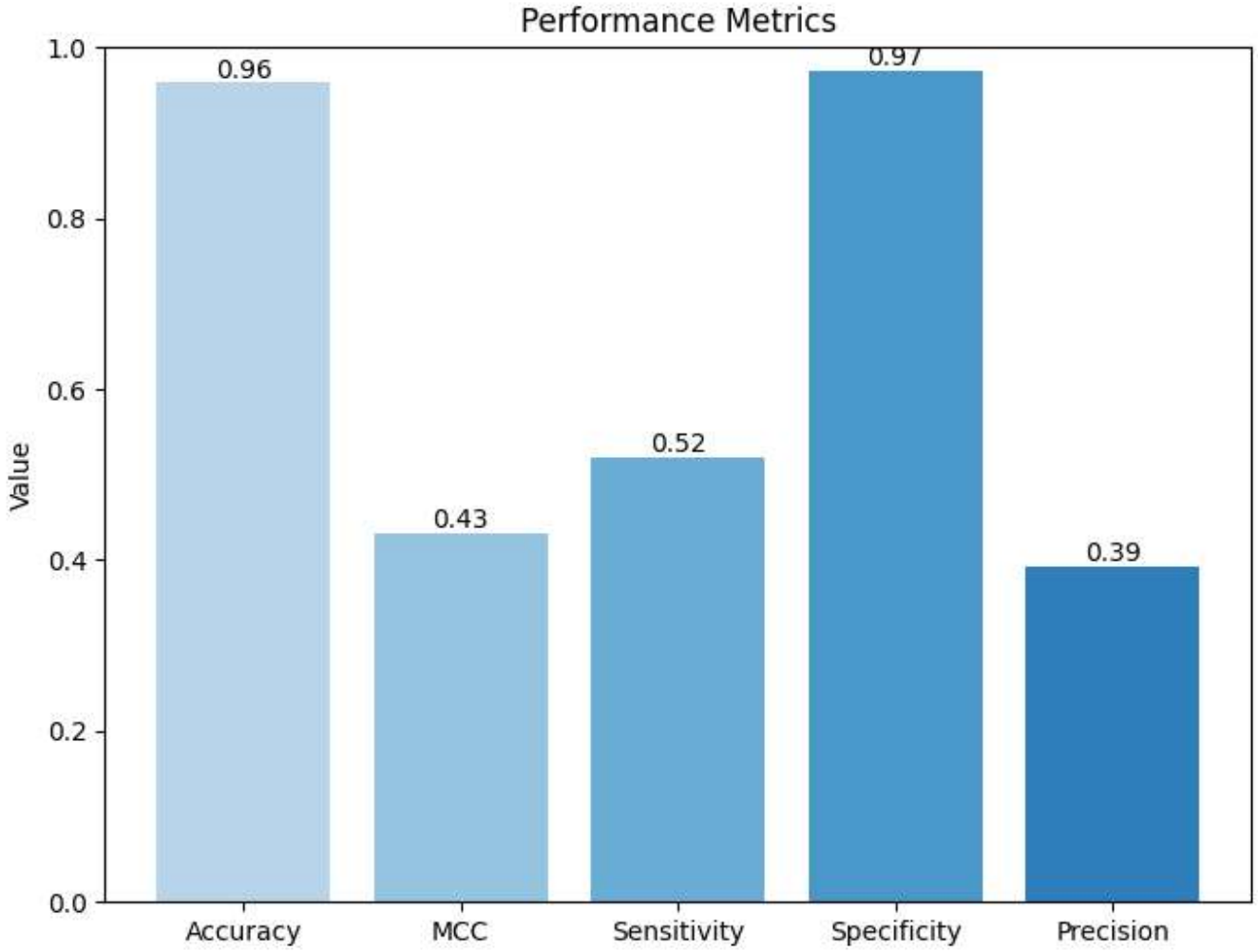

Figure 20: Evaluation of DensePPI Model Performance Metrics showing Accuracy, MCC, Sensitivity, Specificity, Precision score

This study outlines a comprehensive approach for predicting active sites at the residue level in protein-protein interactions (PPI) of parsed PDB data for a sample of *Homo sapiens* species using deep learning techniques, specifically leveraging the DenseNet201 architecture. The methodology involves several critical steps: data preparation, model training, evaluation, and metrics generation. The model achieved an impressive overall accuracy of 95.84%, demonstrating its strong capability

in correctly identifying interactions. The high accuracy is complemented by a high specificity (true negative rate), indicating that the model excels at correctly identifying non-interacting pairs. This is crucial for applications where false positives can be particularly detrimental.

4.6. Implementation Details

The implementation of the PDB Data Acquisition and parsing was conducted using a Google Colab setup, leveraging its cloud-based computing resources for efficient experimentation. The setup included a Google Colab runtime environment equipped with 12.7GB of RAM and a Google Cloud disk size of 107.7GB, providing ample computational resources for processing large-scale datasets and conducting experiments. The DensePPI model training and prediction was conducted using server system device "ppin1-HP-Z840-Workstation" of Jadavpur University CMATER Lab with Processor Intel® Xeon(R) CPU E5-2695 v4 @ 2.10GHz × 72 and memory of 503.8 GiB.

The core bioinformatics functionality is powered by the Biopython library, while numerical computations and machine learning tasks are handled by NumPy and PyTorch, respectively. Using a conda environment helps manage these dependencies effectively and ensures that the code runs smoothly without conflicts. Below is a comprehensive list of the Python libraries and tools used, along with the suggested environment setup for executing the code efficiently.

Python Libraries

General Libraries

1. os: Used for interacting with the operating system, including file and directory management.
2. csv: Utilized for reading from and writing to CSV files.
3. json: Used for parsing JSON formatted data.
4. urllib.request: Used for downloading files from the internet.
5. warnings: Used for handling warnings, specifically ignoring certain types of warnings.
6. itertools.combinations: Utilized for generating combinations of items.
7. random: Used for generating random numbers and performing random operations.
8. pickle: For serializing and deserializing Python object structures.

Numerical and Scientific Computing Libraries

1. numpy: A powerful library for numerical operations and handling arrays.
2. torch: PyTorch library for building and training deep learning models.
3. torchvision: Provides datasets, transforms, and models specifically for computer vision.

Bioinformatics Libraries

1. Bio.PDB.MMCIFParser: For parsing mmCIF files.
2. Bio.PDB.PDBParser: For parsing PDB files.
3. Bio.PDB.PPBuilder: For building polypeptides from chains.
4. Bio.PDB.MMCIF2Dict: For converting mmCIF files to dictionaries.
5. Bio.PDB.PDBExceptions.PDBConstructionWarning: For handling specific warnings from Biopython.

Machine Learning and Evaluation Libraries

1. sklearn.metrics: Provides evaluation metrics such as ROC AUC, precision-recall curve, F1 score, confusion matrix, etc.

Data Visualization Libraries

1. matplotlib.pyplot: For plotting graphs and visualizing data.

Miscellaneous Libraries

1. tqdm: For displaying progress bars during training and evaluation.

Chapter 5
Conclusions

This study presents a comprehensive approach utilizing deep learning techniques, specifically the DenseNet201 architecture, to identify active sites at the residue level in protein-protein interactions (PPI) using parsed PDB data from Homo sapiens species. The methodology encompasses critical processes such as data preparation, model training, assessment, and metrics generation. The model demonstrated strong performance, achieving an impressive overall accuracy of 95.84%. It excelled in correctly identifying non-interacting pairs, evidenced by its high specificity. This capability is particularly important for applications where minimizing false positives is crucial.

5.1. Conclusions of the Present Work

This study presents a comprehensive approach utilizing deep learning techniques, specifically the DenseNet201 architecture, to identify active sites at the residue level in protein-protein interactions (PPI) using parsed PDB data from Homo sapiens species. The methodology encompasses critical processes such as data preparation, model training, assessment, and metrics generation. The model demonstrated strong performance, achieving an impressive overall accuracy of 95.84%. It excelled in correctly identifying non-interacting pairs, evidenced by its high specificity. This capability is particularly important for applications where minimizing false positives is crucial.

- Despite high training accuracy, the model exhibited discrepancies in sensitivity and precision during testing, suggesting potential overfitting to the training data.
- The identification of positive interactions was less accurate, indicating a need for further refinement and balancing of the model.
- The current model may not generalize well to diverse datasets, highlighting the limitations in robustness and general applicability.

5.2. Limitations of the Present Work

However, the evaluation metrics also reveal areas for improvement. The Matthews Correlation Coefficient (MCC) of 0.4310 suggests a moderate level of correlation between the predicted and

actual labels. While this is a positive outcome, it highlights that there is room for enhancing the model's predictive power. More notably, the sensitivity (true positive rate) is relatively low, indicating a higher rate of false negatives. This suggests that the model may miss some true interactions, which is a critical area for improvement. Similarly, the precision, which measures the proportion of true positive predictions among all positive predictions, stands at a moderate 39.32%. This indicates that a significant proportion of the predicted interactions are not true interactions, highlighting the need for better discrimination by the model.

5.3. Scope for Future Works

The development and optimization of protein-protein interaction (PPI) prediction models is a dynamic and evolving field, with substantial opportunities for enhancing model performance and expanding applications. The following sections outline several promising directions for future work, focusing on optimization strategies, model tuning, architectural exploration, data augmentation, validation, and application expansion. These efforts aim to improve predictive accuracy, robustness, and generalizability of PPI models.

- **Learning Rate Scheduling**

  One of the fundamental aspects of training deep learning models is the optimization of the learning rate. Future work can involve implementing dynamic learning rate scheduling techniques, such as learning rate annealing, learning rate warm-up, or cyclic learning rates. These techniques adjust the learning rate during training, potentially improving convergence rates and achieving better performance.

- **Data Augmentation**

  Data augmentation is a crucial technique to enhance model robustness by artificially increasing the diversity of the training dataset. Future studies can explore advanced augmentation techniques such as rotation, flipping, scaling, translation, and normalization of protein structures. Additionally, synthetic data generation using generative adversarial networks (GANs) or other data synthesis methods could be employed to further diversify the training data.

- **Optimizer Selection**

  The choice of optimizer significantly affects the convergence and performance of the model. While stochastic gradient descent (SGD) is widely used, alternative optimizers like Adam, RMSprop, or AdaGrad may offer better convergence properties. Future work can involve extensive experimentation with different optimizers and their hyperparameters to identify the most effective optimization strategy for PPI prediction.

- **Increasing Training Epochs**

  Training the model for more epochs can lead to better convergence and improved performance. However, this needs to be balanced with the risk of overfitting. Future work can explore strategies to increase training epochs while incorporating regularization techniques, such as dropout, to prevent overfitting.

- **Regularization Techniques**

  Regularization is essential to improve model generalization and prevent overfitting. Future efforts can include the integration of dropout layers, weight decay, and batch normalization. Experimentation with different dropout rates and other regularization parameters can help in identifying the optimal configuration for robust model performance.

- **Model Tuning**

  Fine-tuning the model's thresholds and hyperparameters is critical to achieving a balance between sensitivity and specificity, thereby reducing false negatives and improving overall predictive accuracy. Future work can involve comprehensive hyperparameter optimization using techniques such as grid search, random search, or Bayesian optimization.

- **Architecture Exploration**

  Investigating alternative neural network architectures or incorporating ensemble methods can potentially enhance the model's robustness and accuracy. Future research can explore advanced architectures like Transformers, Graph Neural Networks (GNNs), or Convolutional Neural Networks (CNNs) tailored for PPI prediction. Ensemble methods, which combine predictions from multiple models, can also be investigated to improve predictive performance.

- **Extensive Cross-Validation and Testing**

  Conducting extensive cross-validation and testing on independent datasets is crucial for evaluating and refining the model's performance. Future studies can involve rigorous cross-validation protocols, such as k-fold cross-validation, and testing on diverse, independent datasets to ensure the model's generalizability and robustness.

- **Application Expansion**

  The model's application can be expanded to other types of protein interactions, including inter-species interactions. Integrating the model into broader bioinformatics workflows can extend its utility and effectiveness. Future work can involve exploring the application of the model in different biological contexts, such as drug-target interactions, enzyme-substrate interactions, and multi-protein complex formation.

- **Visual Detection and Validation**

  Developing coding utilities to visually detect active sites based on PPI predictions can enhance the interpretability and usability of the model. Future work can include designing visualization tools to highlight areas where active sites are proximate, using heatmaps generated from distance matrices of PPI. These visual tools can be validated against computationally generated heatmaps, providing novel insights into the structural basis of PPIs at the residue level.

The outlined future work encompasses a broad range of strategies and methodologies aimed at enhancing the performance and application scope of PPI prediction models. By revising PDB data parsing procedures to ensure consistent generation of normalized heatmaps and integrating detailed protein sequence and structure information, we can significantly improve the accuracy of active site predictions. Furthermore, implementing the other proposed improvements—such as dynamic learning rate scheduling, advanced data augmentation techniques, alternative optimizers, extended training epochs, regularization methods, comprehensive model tuning, architectural exploration, extensive validation, and application expansion—will collectively contribute to achieving more accurate, robust, and generalizable predictions. These advancements will enhance our understanding of protein interactions and their implications in various biological processes and diseases, ultimately driving forward the field of bioinformatics and its applications in biomedical research.

References

[1] Zhao, J., Cao, Y. and Zhang, L., 2020. Exploring the computational methods for protein-ligand binding site prediction. Computational and structural biotechnology journal, 18, pp.417-426.

[2] Durrant, J.D. and McCammon, J.A., 2011. Molecular dynamics simulations and drug discovery. BMC biology, 9, pp.1-9..

[3] Öztürk, H., Özgür, A. and Ozkirimli, E., 2018. DeepDTA: deep drug–target binding affinity prediction. Bioinformatics, 34(17), pp.i821-i829.

[4] Ballester, P.J. and Mitchell, J.B., 2010. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. Bioinformatics, 26(9), pp.1169-1175.

[5] Seco, J., Luque, F.J. and Barril, X., 2009. Binding site detection and druggability index from first principles. Journal of medicinal chemistry, 52(8), pp.2363-2371.

[6] Heo, L., Shin, W.H., Lee, M.S. and Seok, C., 2014. GalaxySite: ligand-binding-site prediction by using molecular docking. Nucleic acids research, 42(W1), pp.W210-W214.

[7] Hamelryck, T. and Manderick, B., 2003. PDB file parser and structure class implemented in Python. Bioinformatics, 19(17), pp.2308-2310.

[8] Rose, Y., Duarte, J.M., Lowe, R., Segura, J., Bi, C., Bhikadiya, C., Chen, L., Rose, A.S., Bittrich, S., Burley, S.K. and Westbrook, J.D., 2021. RCSB Protein Data Bank: architectural advances towards integrated searching and efficient access to macromolecular structure data from the PDB archive. Journal of molecular biology, 433(11), p.166704.

[9] Rose, P.W., Prlić, A., Altunkaya, A., Bi, C., Bradley, A.R., Christie, C.H., Costanzo, L.D., Duarte, J.M., Dutta, S., Feng, Z. and Green, R.K., 2016. The RCSB protein data bank: integrative view of protein, gene and 3D structural information. Nucleic acids research, p.gkw1000.

[10] Wang, D.D., Zhu, M. and Yan, H., 2021. Computationally predicting binding affinity in protein–ligand complexes: free energy-based simulations and machine learning-based scoring functions. Briefings in bioinformatics, 22(3), p.bbaa107.

[11] Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y. and Jiang, H., 2007. Predicting protein–protein interactions based only on sequences information. Proceedings of the National Academy of Sciences, 104(11), pp.4337-4341.

[12] Guo, Y., Yu, L., Wen, Z. and Li, M., 2008. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. Nucleic acids research, 36(9), pp.3025-3030.

[13] Yang, L., Xia, J.F. and Gui, J., 2010. Prediction of protein-protein interactions from protein sequence using local descriptors. Protein and peptide letters, 17(9), pp.1085-1090.

[14] Zhou, Y.Z., Gao, Y. and Zheng, Y.Y., 2011. Prediction of protein-protein interactions using local description of amino acid sequence. In Advances in Computer Science and Education Applications: International Conference, CSE 2011, Qingdao, China, July 9-10, 2011. Proceedings, Part II (pp. 254-262). Springer Berlin Heidelberg.

[15] You, Z.H., Huang, W.Z., Zhang, S., Huang, Y.A., Yu, C.Q. and Li, L.P., 2018. An efficient ensemble learning approach for predicting protein-protein interactions by integrating protein primary sequence and evolutionary information. IEEE/ACM transactions on computational biology and bioinformatics, 16(3), pp.809-817.

[16] Tsukiyama, S., Hasan, M.M., Fujii, S. and Kurata, H., 2021. LSTM-PHV: prediction of human-virus protein–protein interactions by LSTM with word2vec. Briefings in bioinformatics, 22(6), p.bbab228.

[17] Yuan, Q., Chen, J., Zhao, H., Zhou, Y. and Yang, Y., 2022. Structure-aware protein–protein interaction site prediction using deep graph convolutional network. Bioinformatics, 38(1), pp.125-132.

[18] Li, H., Gong, X.J., Yu, H. and Zhou, C., 2018. Deep neural network based predictions of protein interactions using primary sequences. Molecules, 23(8), p.1923.

[19] Halsana, A.A., Chakroborty, T., Halder, A.K. and Basu, S., 2023. DensePPI: A Novel Image-based Deep Learning method for Prediction of Protein-Protein Interactions. IEEE Transactions on NanoBioscience.

[20] Sanner, M.F., Duncan, B.S., J. CARRILLO, C. and Olson, A.J., 1999. Integrating computation and visualization for biomolecular analysis: an example using python and AVS. In Biocomputing'99 (pp. 401-412).

[21] Zardecki, C., Dutta, S., Goodsell, D.S., Voigt, M. and Burley, S.K., 2016. RCSB protein data bank: a resource for chemical, biochemical, and structural explorations of large and small biomolecules.

[22] You, Z.H., Zhu, L., Zheng, C.H., Yu, H.J., Deng, S.P. and Ji, Z., 2014, December. Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set. In BMC bioinformatics (Vol. 15, pp. 1-9). BioMed Central.

[23] Bandyopadhyay, S.S., Halder, A.K., Saha, S., Chatterjee, P., Nasipuri, M. and Basu, S., 2023. Assessment of GO-based protein interaction affinities in the large-scale human–coronavirus family interactome. *Vaccines*, *11*(3), p.549.

[24] Tian, B., Wu, X., Chen, C., Qiu, W., Ma, Q. and Yu, B., 2019. Predicting protein–protein interactions by fusing various Chou's pseudo components and using wavelet denoising approach. Journal of Theoretical Biology, 462, pp.329-346.

[25] You, Z.H., Yu, J.Z., Zhu, L., Li, S. and Wen, Z.K., 2014. A MapReduce based parallel SVM for large-scale predicting protein–protein interactions. Neurocomputing, 145, pp.37-43.

[26] Colonnese, S., Petti, M., Farina, L., Scarano, G. and Cuomo, F., 2021. Protein-protein interaction prediction via graph signal processing. IEEE Access, 9, pp.142681-142692.

[27] You, Z.H., Chan, K.C. and Hu, P., 2015. Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest. PloS one, 10(5), p.e0125811.

[28] Gao, M., Nakajima An, D., Parks, J.M. and Skolnick, J., 2022. AF2Complex predicts direct physical interactions in multimeric proteins with deep learning. Nature communications, 13(1), p.1744.

[29] Tubiana, J., Schneidman-Duhovny, D. and Wolfson, H.J., 2022. ScanNet: an interpretable geometric deep learning model for structure-based protein binding site prediction. Nature Methods, 19(6), pp.730-739.

[30] Yang, F., Fan, K., Song, D. and Lin, H., 2020. Graph-based prediction of protein-protein interactions with attributed signed graph embedding. BMC bioinformatics, 21, pp.1-16.

[31] Wang, L., Wang, H.F., Liu, S.R., Yan, X. and Song, K.J., 2019. Predicting protein-protein interactions from matrix-based protein sequence using convolution neural network and feature-selective rotation forest. Scientific reports, 9(1), p.9848.

[32] Kösesoy, İ., Gök, M. and Öz, C., 2019. A new sequence based encoding for prediction of host–pathogen protein interactions. Computational Biology and Chemistry, 78, pp.170-177.