Sequence-Based Predictions of Binding Residues and Secondary Structures of Proteins using Deep Learning

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Natural Sciences
by Research

by

Vineeth Chelur
201564080
ravindrachelur.v@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2021

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Sequence-Based Predictions of Binding Residues and Secondary Structures of Proteins using Deep Learning**" by Vineeth Chelur, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. U. Deva Priyakumar

To peace and tranquility

Acknowledgments

I want to express my gratitude to my mentor, Dr U Deva Priyakumar, for allowing me to work on the projects that I was interested in and guiding me throughout the duration of my research work at IIIT Hyderabad.

I would also like to express my gratitude to my friends and lab-mates at IIIT who kept me company and helped with the various projects, especially Hemanth Vemuri, Nikhil Vemuri, Vaishnavi Reddy, Arpita Dash, Yashaswi Pathak, Siddhartha Laghuvarapu, Oishika Pradhan, Shanmukh Alle and Yashas Samaga. A big shout out to my friends at home, Rakshith Venkatesh, Amith Bharadwaj, Akshay Seshagiri, Srinand Shivalingaiah, Shashank Bharadwaj, and Swaroop Sateesh, for filling my life with joy and laughter during these challenging times. A special thanks to gerzytet for proofreading the research paper and verifying code reproducibility.

I would also like to thank my uncle, Dattananda Chelur and aunt, Uma Rao, for their continued support and advice through writing the research paper and thesis.

Above all, I would like to thank my family, Ravindra Chelur, Nalinakshi Srikantiah, Reshma Chelur, Madan Embar, and Drithi Embar, for their unconditional love and support through the difficult times of the pandemic.

A big thank you to all the front-line workers for keeping us safe through the pandemic.

Abstract

With the number of protein sequences increasing rapidly, it becomes imperative to have a basic idea of the function and structure of a protein before the three-dimensional structure becomes available. Protein-drug interactions play essential roles in many biological processes and therapeutics. Prediction of the active binding site of a protein helps discover and optimise these interactions leading to the design of better ligand molecules. The secondary structure provides clues to the shape that the protein can be expected to take. It tells us whether an amino acid belongs to a coil turn, alpha-helix or beta-sheet structure. Deep Learning is a class of machine learning algorithms that progressively uses multiple layers to extract higher-level features from raw input. Deep learning methods eliminate feature engineering for supervised learning tasks by translating the raw inputs into intermediate representations that capture the more abstract and composite information, removing redundancies in the original input. The rapid adoption and success of deep learning algorithms in various sections of structural biology beckon deep learning algorithms for accurate binding site detection and secondary structure prediction.

Protein-drug interactions play essential roles in many biological processes and therapeutics. Prediction of the active binding site of a protein helps discover and optimise these interactions leading to the design of better ligand molecules. The tertiary structure of a protein determines the binding sites available to the drug molecule. To quickly and accurately predict the binding site from sequence alone without utilising the three-dimensional structure is challenging. In the first study, a Residual Neural Network (leveraging skip connections) [1] is implemented to predict a protein's most active binding site. An Annotated Database of Druggable Binding Sites from the Protein DataBank, sc-PDB [2], is used for training the network. Features extracted from the Multiple Sequence Alignments (MSAs) of the protein generated using DeepMSA, such as Position-Specific Scoring Matrix (PSSM), Secondary Structure (SS3), and Relative Solvent Accessibility (RSA), are provided as input to the network. A weighted binary cross-entropy loss function is used to counter the substantial imbalance in the two classes of binding and non-binding residues. The network performs very well on single-chain proteins, providing a pocket that has good interactions with a ligand.

Secondary structure predictions predict three classes: C (Coil Turn), H (Alpha Helix) and E(Beta Sheet). In the second study, a Transformer (based on the multi-attention mechanism) network is used to train on the TR4590 dataset, which contains 4590 proteins with a sequence similarity cut off 25% and X-ray resolution better than 2.0Å. Ten models are trained across 10-fold cross-validation, and a weighted cross-entropy loss function is used. The ten trained models are run on the test set containing 1199 sequences. The mean of the probabilities of each class is taken, and then the class with the maximum probability is considered the class to which the amino acid belongs. The model achieves an accuracy of 82.51%.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Protein and its Structures

Proteins are macromolecules that carry out vital functions in all biological processes in the human body, such as DNA replication, providing structure to cells, transporting molecules, etc. They are comprised of one or more long chains of amino acid residues (known as a polypeptide chain). Amino acids are organic compounds that contain an amino group ($-NH_2$), a carboxyl group ($-COOH$), and a side chain group (R). Although there are around 500 naturally occurring amino acids, only 20 appear in the genetic code of life. There are four levels of amino acid organisation in a protein: primary, secondary, tertiary and quaternary structures. The primary structure refers to the linear sequence of amino acid residues held together by peptide bonds between the amino acids. The secondary structure is a course-grained descriptor of the local structure of the polypeptide backbone, containing highly regular local sub-structures. The secondary structure involves hydrogen bonds along the backbone that cause the long chain to fold into local shapes, mainly α-helices, β-sheets and coils. Tertiary structure is the three-dimensional of a single protein molecule (polypeptide chain). The α-helices and β-pleated-sheets are folded into a compact globular structure, driven by non-specific hydrophobic interactions, the burial of hydrophobic residues from water, salt bridges, hydrogen bonds, and the tight packing of side chains and disulfide bonds. Quaternary structure is the aggregation of two or more individual polypeptide chains that operate as a single functional unit, stabilised by non-covalent interactions. A binding site is a region of the protein to which a ligand binds, with specificity. Often, the binding is accompanied by a conformational change that alters the protein's function.

1.2 Modern Machine Learning Methods

Machine learning is the process of using a computer algorithm to learn from data. The goal of machine learning is to find patterns in data that are not explicitly given, which is done by finding the best fit of a model to the data, by training a mathematical model, validating it along the way and then testing it to see how well it performs on unknown data. Deep learning imitates the workings of the human brain in processing data and creating patterns for use in decision making [7]. There has been a great deal of progress in deep learning. Some examples of architectures include: ANNs (Artificial Neural Networks) [8], RNNs (Recurrent Neural Networks) [9], LSTM (Long Short-Term Memory) Networks [10], ResNets (Residual Neural Networks) [1], Transformers [5], GANs (Generative Adversarial Networks) [11], SOMs (Self-Organizing Maps) [12], Boltzmann Machines [13], and VAEs (Variational Autoencoders) [14].

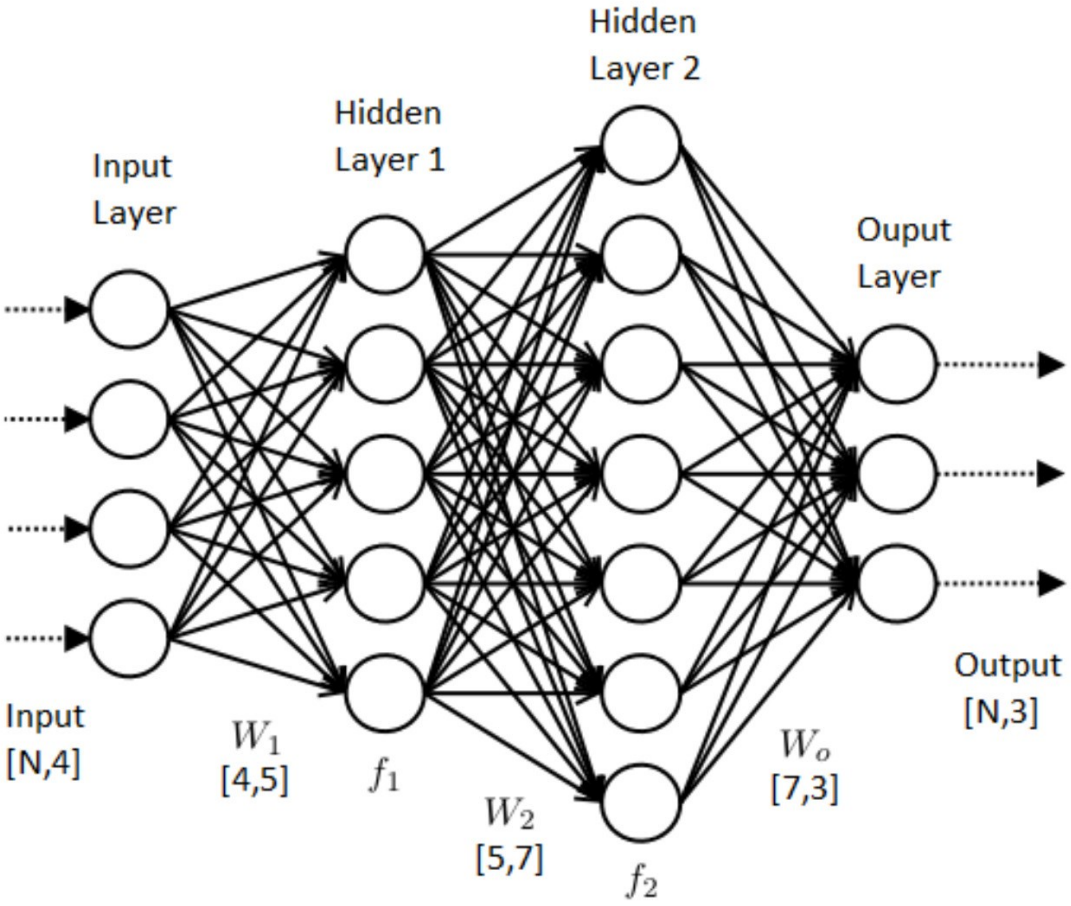

Figure 1.1: Architecture of a simple Artificial Neural Network model. Source: [3]

An Artificial Neural Network (ANN) is a straightforward machine learning model designed to simulate how the human brain analyses and processes information. It is the foundation of Artificial Intelligence and Deep Learning and can solve impossible or complex problems. They have self-learning capabilities, enabling them to produce better results as more data becomes available. An ANN has a collection of connected nodes called artificial neurons. Each connection/edge, like the synapses in a brain, can transmit signals (a real number computed by some non-linear function of a mathematical computation of its inputs) to another neuron. Each connection has a weight that adjusts as learning proceeds and increases or decreases the strength of the signal at a connection. Figure 1.1 shows a simple ANN model with an input layer, two hidden layers and an output layer.

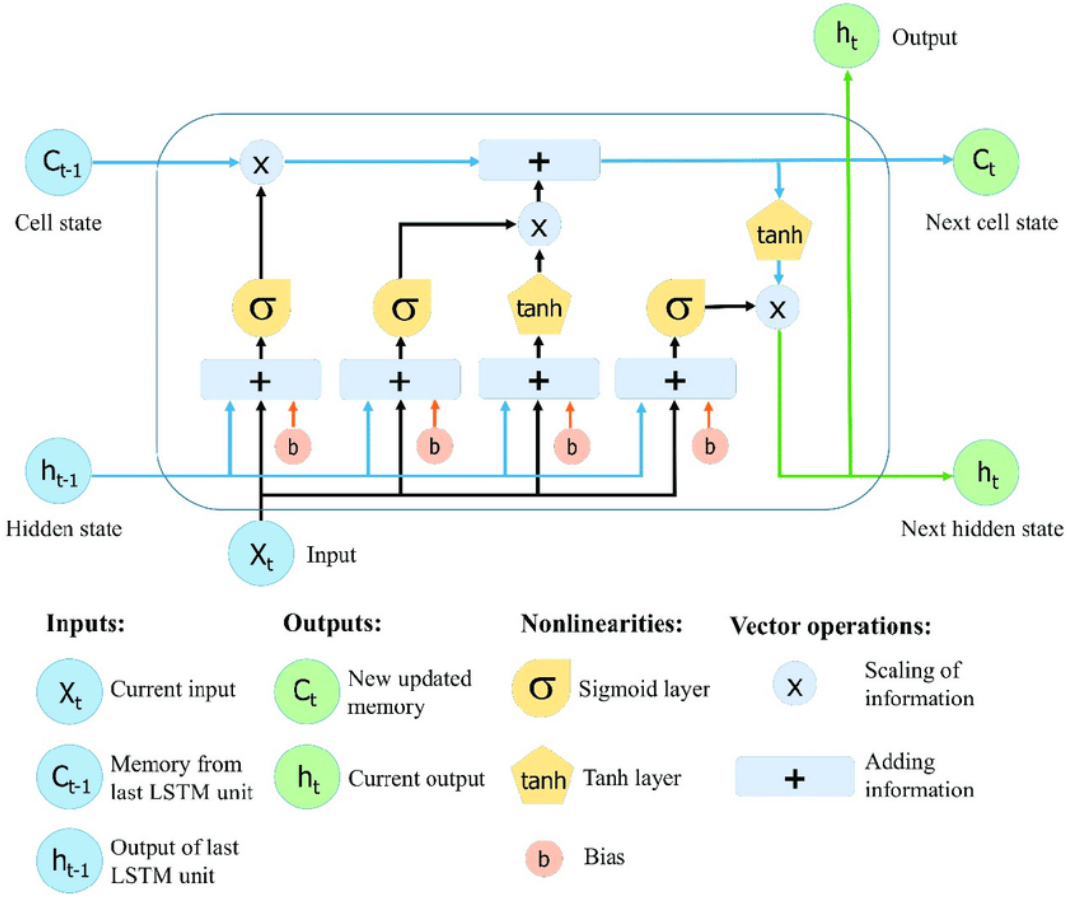

Figure 1.2: The structure of the Long Short-Term Memory (LSTM) cell. Source: [4]

LSTM [10] is a Recurrent Neural Network (RNN) [9], which has feedback connections, meaning that it can not only process single data points but a sequence of data points. RNNs are networks with loops in them which allows for them to persist past information. However, a problem of RNNs is that they remember only recent information and not long-range dependencies. LSTMs solve this problem by adding gates in their memory cell, allowing for the LSTM to remove or add information to the cell state based on whether or not the information is valuable or not. Figure 1.2 shows the structure of a single LSTM cell.

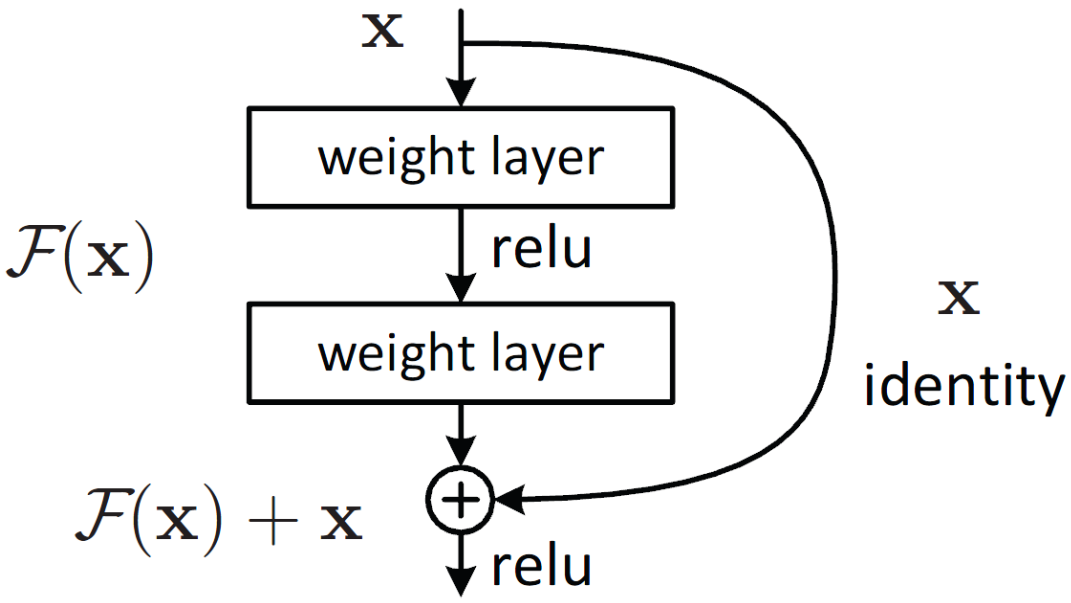

Figure 1.3: Residual learning of a building block of the Resnet. Source: [1]

CNNs (Convolutional Neural Networks) take in an input image, assign importance through learnable weights and biases to various aspects of the image, and differentiate one from another [15]. It is similar to a neural network but in higher dimensions. It uses the available surrounding information to capture spatial and temporal dependencies and make predictions. CNNs need not only be applied to images but also any matrix of information. ResNet [1] is a particular type of CNN that uses skip connections between layers to persist the original available information, allowing for the network to learn more complex features and to model long-range dependencies, making them very popular for classification problems that require modelling long-range dependencies. Figure 1.3 shows the skip connection that is added to a building block that helps the network persist the original information.

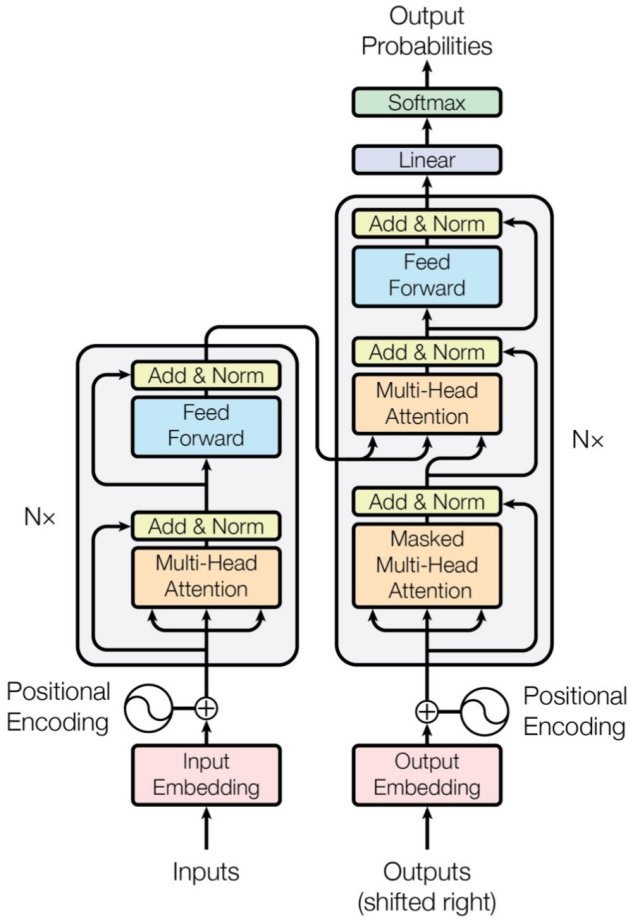

Figure 1.4: Model architecture of a transformer. Source: [5]

Transformer Neural Network [5] aims to solve sequence-to-sequence tasks while handling long-range dependencies. It introduces an encoder-decoder architecture based on multi-head attention layers, RNNs, embedding space, positional encoding, and feedforward networks. Attention tells us which part of the input is essential and should be focused on. A multi-head attention layer focuses on multiple parts of the input. Embedding space is used to convert the input into a dictionary that the model can understand. Positional encoding is a technique to provide a vector that gives context according to the position of, say, a word in a sentence. A feedforward network is just a plain artificial neural network that helps learn about the input. The encoder creates a representation of all the words provided until now, and the decoder will decode that information to predict the next word in the sequence. Transformers can be used for various tasks such as speech synthesis, machine translation, image captioning, and question answering. Figure 1.4 shows the architecture of a transformer network.

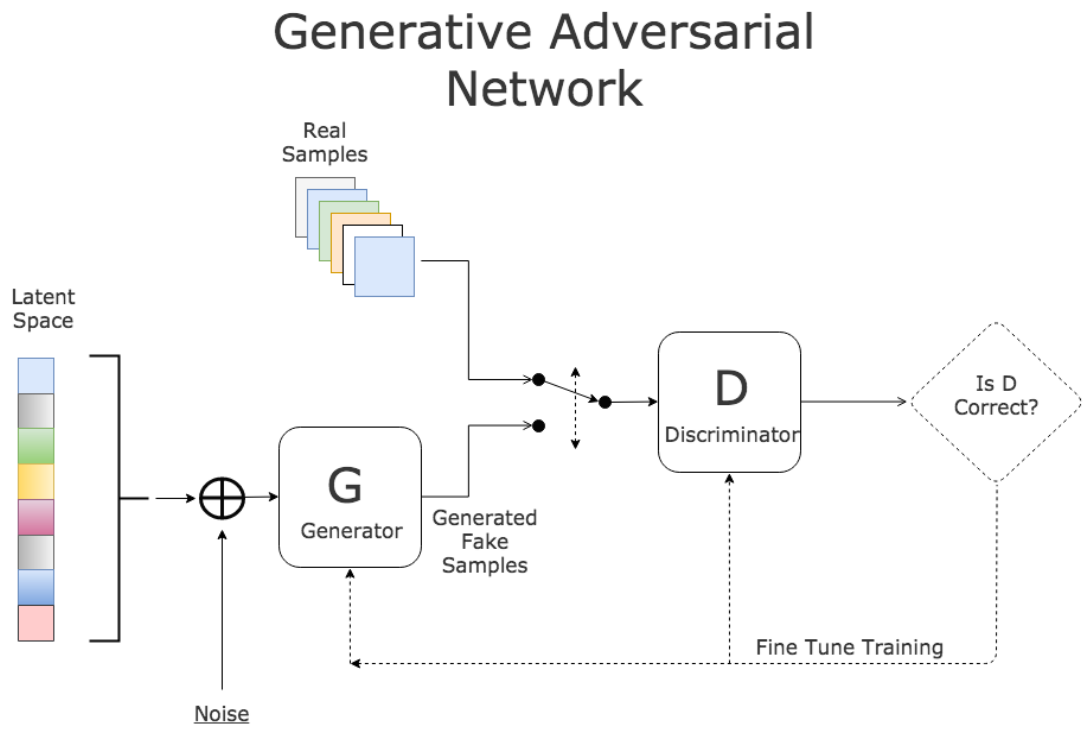

Figure 1.5: Architecture of a Generative Adversarial Network. Source: [6]

GAN (generative adversarial network) [11] consists of two neural networks that compete with each other in a zero-sum game. The generator learns to generate data similar to actual data by using information from a learned latent space representation. The discriminator tries to learn to distinguish between the actual data and generated data. Though it was intended for unsupervised learning, where the goal is to generate data similar to provided data, it has been very successful in semi-supervised, fully supervised and reinforcement learning. In figure 1.5, the architecture of a simple GAN is shown.

1.3 Role of Machine Learning in Protein Predictions

Over the years, many supervised and unsupervised machine learning methods have been applied to tackle protein prediction problems and have significantly contributed to advancing state-of-the-art protein predictions. The predictions can broadly be classified into protein structure predictions and protein function predictions.

Since its inception, deriving a protein's structure from its sequence alone has been an unsolved problem due to the large conformational space of a protein chain and the lack of accurate energy functions to model the folding process. Hence, it becomes necessary to solve more straightforward problems such as the determination of the secondary structure, torsion angles, contact map and distance map of the protein.

Protein secondary structure refers to the local conformation of the polypeptide backbone of proteins. It consists of 3 class classification: H (α-helix), E (β-sheet), and C (coil), as well as 8 class classification: H(α-helix), G(3-10-helix), I(π-helix), E(β-strand), B(isolated β-bridge), T (turn), S(bend), and C (others). DSSP [16] and STRIDE [17] are used for calculating the secondary structure from known protein structures and used for creating the datasets. PHD [18] and PSIPRED [19] were one of the earliest methods to use MSAs and neural networks for predictions. SPARROW [20] used two stages of multilinear regression and a neural network. PORTER 4.0 [21], SPIDER2 [22] leveraged Recurrent Neural Networks along with more information derived from MSAs to incorporate information of surrounding residues. PORTER 5 [23] and MUFOLD-SS [24] used more complex deep learning models like CNNs and inception networks to achieve state-of-the-art results.

The torsion angles are ϕ, ψ, θ and τ and form the complimentary basis for local backbone structure. ϕ and ψ are the angles between the planes formed by three consecutive residues, whereas θ and τ are 3 and 4 residues, respectively. These angles have been predicted as both discrete states and continuous values. Kang et al. predicted the probabilities for phi-psi angles using the appropriate frequencies from a database of crystal structures and then applying a custom function on the data. SPINE-X[26] uses a guided-learning artificial neural network with a conditional random field model. ANGLOR [27] and TANGLE [28] used Support Vector Machines (SVM) [29] by considering PSSM information. SPIDER3 [30] captured the non-local interactions by using long short-term memory (LSTM) bidirectional RNNs for prediction. RaptorX-Angle [31], and SPOT-1D [32] used an ensemble of Recurrent and Residual Neural Networks for predicting the real-value angles.

A contact map of a protein is a matrix of zeros and ones that shows the existence of contacts between residues. A distance map is a matrix of distances between residues. CORNET [33], DISTILL [34] were one of the first methods to use neural networks along with evolutionary information to predict the contact map. SVMcon [35] and SVMSEQ [36] used SVMs, whereas PconsC2 [37] and RaptorX-Contact [38] used Deep learning methods such as RNNs and ResNets for more accurate predictions. SPOT-Contact [39], and TripletRes [40] are some of the more recent methods that have been very successful with distance predictions, and these predictions have helped immensely in the state-of-the-art protein structure prediction.

Critical Assessment of Protein Structure Prediction (CASP) [41] is a community-wide, worldwide experiment for protein structure prediction that takes place every two years and has become the standard for testing new methods in structure prediction [42]. Modern prediction methods comprise of four modules: an input module that takes a protein sequence to generate additional input features such as MSAs, a neural network module capable of pattern recognition, which transforms the input feature vectors into vectors with partial spatial information, an output module that converts the spatial information into an initial 3D structure, and finally, a refinement module that improves the 3D structure and produces all atomic coordinates. In previous CASP assessments, a mixture of physics-based energy functions, knowledge-based statistical reasoning, and heuristic algorithms was used in these modules [43]. However, the inclusion of neural networks into all the modules has vastly improved the quality of the predictions. Some of the more recent and successful methods include Deepmind's AlphaFold [44], AlphaFold2 [45], RaptorX [46], Robetta [47], FEIG-R2 [48], I-TASSER [49], MULTICOM [50], and QUARK [51].

The binding site of a protein is the pocket in the 3D structure of the protein where a ligand binds to and changes the conformation of the protein, making it functional. DeepCSeqSite [52] is a template-based method that uses seven characteristics (position-specific scoring matrix, relative solvent accessibility, secondary structure, dihedral angle, conservation scores, residue type and positional embeddings) of each residue to create a feature map, which is then used as an input to a convolutional neural network. DeepPocket [53] is a structure-based method that uses 3D Convolutional Neural Networks to generate a list of pocket probabilities and a segmentation model to elucidate shapes for the top-ranked pockets.

Binding affinity is the strength of the binding interaction between a protein to its ligand. KronRLS [54] used Kronecker-Regularized Least Squares, and SimBoost [55] used gradient boosting regression trees method to rank the binding of a set of drugs to a set of target proteins. DeepDTA [56] was the first Deep Learning Approach that used SMILES [57] representation as an input to a CNN architecture to predict the binding affinity value without using structural information. WideDTA [58], PADME [59], and DeepAffinity [60] later used more complex architectures such as ResNets, Graph Convolutional Neural Networks and RNNs, and more relevant input features for prediction.

1.4 Motivation

The tertiary structure of a protein provides important clues about its function. The rapid speed of sequencing attained with modern DNA sequencing technology has been instrumental in sequencing complete DNA sequences, leading to faster sequencing of proteins. Although there have been improvements in the determination of the three-dimensional protein structure by techniques such as X-Ray Crystallography, NMR Spectroscopy and Cryo-Electron Microscopy, they are expensive, labour-intensive and time-consuming, and sometimes not possible. The gap between the number of known protein sequences (214,406,399 UniProt sequences as of May 2021)[61], and the number of known structures (177,910 PDBs as of May 2021)[62][63] is increasing rapidly. Proteins perform a vast array of functions within organisms, and the tertiary structure of a protein can provide important clues about these functions. Although the main goal is to predict the 3-D structure, 1-D and 2-D predictions are of intrinsic interest and often used as inputs for 3-D coordinate predictors. Hence, predictions based on the protein sequence must be made to speed up the structure prediction process and provide clues towards the function of a protein.

Recent advances in deep learning and the development of new deep learning frameworks have enabled state-of-the-art models for predictions on protein sequences. Also, due to the availability of large amounts of protein sequence data, the models have been very successful in making highly accurate predictions. Hence, it becomes imperative to apply the latest machine learning techniques to various protein prediction problems.

1.5 Thesis Structure

This thesis tackles the problems of protein binding site prediction and secondary structure prediction solely based on the sequence alone. The first study introduces BiRDS[64], a ResNet-based model for predicting the binding site of a protein. The second study deals with the prediction of the secondary structure of a protein using a Transformer network. Both studies have the following sections: Introduction, Methods (datasets used, features generated, model architecture, and evaluation metrics), Results and Discussion, and Conclusion. The thesis concludes with how protein sequences can derive valuable insights, how machine learning has helped with the predictions, and future possibilities.

Chapter 2

Binding Site Prediction

2.1 Introduction

Protein-ligand complexes are functionally important in crucial mechanisms such as DNA replication, metabolism, catalysis, defence against viruses, and signal transduction. A ligand can be any molecule that binds to the protein with high affinity where the interaction site is the active binding site of the protein. In drug design, a new drug is modelled to improve protein function after identifying a potential active binding site, thus aiding in these crucial mechanisms.

Ligand binding site prediction methods are broadly categorised into geometry-based, energy-based, template-similarity-based, traditional machine-learning-based and deep-learning-based prediction methods [65]. Geometry-based and energy-based methods maintain that most small ligand bindings occur in cavities on protein surfaces since large interfaces have a high affinity to small molecules. These methods locate the binding site by searching for spatial geometry or energy features by placing probes in protein structures. SITEHOUND[66] uses a carbon and phosphate probe inside a grid covering the entire protein. The grid points with higher interaction energies are clustered to determine the binding residues. A spatial geometric measurement method CURPOCKET[67] computes the curvature distribution of the protein surface and identifies clusters of concave regions. Other methods in this category include CASTp[68], LIGSITE[69], VISCANA[70], Fpocket[71], and Patch-Surfer2.0[72]. While these methods are widely used, they are invalid in certain cases due to their dependence on various factors, such as the resolution of the structure determination method and the presence of both ligand groups and external molecules.

Template-similarity-based methods consider that proteins evolved from structurally, functionally, or sequentially similar proteins, not as independent entities. S-SITE and TM-SITE[73] employ the Needleman-Wunsch algorithm to align the query protein to sequentially-similar proteins in the BioLip[74] database, a curated database for biologically relevant ligand-protein binding interactions. The frequently-occurring binding residues in the aligned proteins form the binding residues of the query protein. Methods such as ConSurf[75], FINDSITE[76], 3DLigandSite[77], FunFOLD[78], and COFACTOR[79] also employ similarity searching.

3D-structure-based and template-similarity-based methods complement each other very well. Traditional machine-learning-based methods build an analytical model based on protein data to identify patterns and structural similarities. Machine learning integrates the information of both the methods and applies mathematical functions to improve prediction accuracy. P2RANK[80][81] uses a random forest algorithm to predict ligandibility scores across the entire protein surface. Ligandibility score is the score given to a ligand for its ability to bind to specific points on the protein. The points with high scores are then clustered into a single binding pocket. SCRIBER[82] is a fast, sequence-based, two-layer architecture, machine learning predictor which predicts propensities of protein-binding, RNA-binding, DNA-binding, and ligand-binding residues. ConCavity[83], MetaPocket[84], RF-Score[85], NsitePred[86], NNSCORE[87][88], LigandRFs[89], COACH-D[90], and Taba[91] employ different machine learning models to predict the protein binding site.

Deep Learning is a subfield of machine learning based on artificial neural networks with feature learning. When a deep learning network is fed large amounts of data, it can automatically discover the representations needed for feature detection or classification. Deep learning has been hugely successful in the general areas of drug design, such as binding affinity predictions[92, 56], protein contact map predictions[39, 38], and protein-structure predictions[44, 93, 94]. Deep learning-based methods like DeepSite[95] and Kalasanty[96] model binding site prediction as an image processing problem. The protein 3D structure is divided into small grids, called voxels, through a process known as voxelisation. Each voxel's specific calculated properties are used to train a deep convolutional neural network that predicts whether a voxel belongs to a binding site. DeepPocket[53] is a structure-based method that uses 3D Convolutional Neural Networks to generate a list of pocket probabilities. A segmentation model then elucidates shapes for the top-ranked pockets.

The tertiary structure of a protein can provide essential clues about the binding sites of a protein. Even though there have been improvements in techniques such as X-ray Crystallography, NMR Spectroscopy, and Cryo-Electron Microscopy, the determination of the three-dimensional protein structure is time-consuming and expensive. Modern DNA sequencing technologies have sped up complete DNA sequencing, and in turn, protein sequencing. The gap between the number of known protein sequences (214,406,399 UniProt sequences as of May 2021)[97] and the number of known structures (177,910 PDBs as of May 2021)[62][63] is enormous. Predicting the binding site based on amino acid sequence alone is challenging. However, it helps to identify potential binding residues before the three-dimensional structure becomes available.

In this paper, a deep residual neural network (ResNet)[1] is trained to predict whether an amino acid residue in the sequence belongs to the most active binding site or not. The sc-PDB database identifies this site as the binding site most suitable for docking a drug-like ligand. Features are extracted from the MSAs generated by DeepMSA[98], whose robustness and usefulness have been studied extensively. BiRDS is trained on these features for all proteins in the training dataset. A weighted binary cross-entropy loss function is used for handling the severe class imbalance. The network outputs the final probabilities, which are converted to binary outputs. Most sequence-based prediction methods predict the binding site of a protein for specific ligands, while most popular 3D structure-based methods predict the ligandable binding sites of a protein. This paper bridges the gap between the two by providing a reliable method for predicting a protein's most active binding site from sequence information alone. SC6K, a novel test set, is used for comparing BiRDS with Kalasanty (a 3D structure-based method) and SCRIBER (a sequence-based method).

2.2 Methods

2.2.1 Dataset

An annotated database of druggable binding sites from the Protein Data Bank, known as sc-PDB (v.2017)[2], is used to train and validate BiRDS. The database takes samples from the Protein Data Bank[62, 99], creates prepared protein structures of biologically relevant protein-ligand complexes by filtering based on Uniprot annotations and prepared ligand templates. The most buried ligand, peptide or cofactor is found in the prepared structure, and the site of interaction is considered the most ligandable binding site. Thus each sample in the dataset contains the three-dimensional structure of one ligand, one protein, and one site.

The sc-PDB (v.2017) database is generally used to predict binding sites based on the available protein-ligand 3D structures. However, this paper deals with predicting the most active binding site using sequence information alone, for which the complete amino acid sequence of all the protein chains is required. The complete 3D structure is typically unavailable because some of the protein regions in the crystal under study are disordered and mobile. Hence the whole sequence cannot be extracted from the structure. Fortunately, the entire protein sequence is always available, and for this paper, it has been downloaded from the RCSB[63] website in FASTA file format. A one-to-one mapping of the amino acids in the downloaded sequence to the amino acids in the protein's 3D structure is required to know which amino acid is a binding residue. This mapping is done by first extracting the protein sequence from the 3D structure. Next, the Needleman-Wunsch dynamic programming algorithm[100] (implemented by Zhanglab's NW-Align program[101]) is utilised to align the sequence extracted from the structure file to the downloaded sequence. The protein structure file is reindexed based on this alignment to match the indices of the residues in the downloaded sequence. This reindexing allows for the labelling of binding residues in the downloaded sequence. Note that the protein sequence is the concatenation of all its chain sequences.

The training set consists of the downloaded sequence and the generated binding residue labels of every protein in the sc-PDB database, which has 17,594 PDB structures with 28,959 chain sequences, of which 9,419 are unique. For training using k-fold cross-validation, we must ensure that no two folds have proteins with sequence similarity greater than 25% to avoid data leakage between the training and validation set during network training. Hence, the pairwise sequence similarity of the 9,419 unique chain sequences was calculated using BLASTP (part of the BLAST$^+$[102] package from NCBI). SiLiX[103] package clustered these unique sequences into families with greater than 25% sequence similarity and over 80% overlap, leading to the creation of 2,039 clusters of chain sequences. Since BiRDS predicts the most active binding site of the complete protein, the protein sequence must also be clustered. The Union-Find algorithm[104] using a disjoint-set data structure was employed to make this clustering, where all the chains of a protein and their corresponding cluster were put in a single set, creating 1,744 sets. Protein sequences longer than 4,096 residues were removed. An equal sum K-partition algorithm put these sets into ten folds for cross-validation. One set had 2,009 proteins and was reduced to 1,642 to split the sets into ten even folds. Finally, this gave 16,450 proteins belonging to the training set, with each fold containing 1,645 proteins.

A separate test set SC6K was constructed using the PDB structures from January 2018 to February 2020. All PDBs with at least one ligand were run through pdbconv program from the IChem Toolkit[105]. The program used the exact filtering mechanism and site selection method as the scPDB[2] database. The entire test set consists of 2,274 PDB structures with 3,434 chain sequences, of which 1,889 are unique. However, there should be no data leakage between the test and training sets. Hence, the pairwise sequence similarity of the 1,889 test chain sequences with the 9,419 training chain sequences was calculated using BLASTP. Sequences with greater than 25% similarity and over 80% overlap were removed from the test set, giving a set of 576 chain sequences. Proteins with all their chain sequences in this set were considered for the reduced test set, leading to a final count of 530 protein sequences.

2.2.1.1 MSA Generation

Collections of multiple homologous sequences (called Multiple Sequence Alignments or MSAs) can provide critical information for modelling the structure and function of unknown proteins. DeepMSA[98] is an open-source method for sensitive MSA construction, which has homologous sequences and alignments created from multiple sources of databases through complementary hidden Markov model algorithms. DeepMSA profiles provided statistically significant improvements in residue-level contact prediction, homologous structure identification and secondary structure prediction. These improvements were achieved without retraining the parameters and neural-network models.

The search for alignments is done in 2 stages. In stage 1, the query sequence is searched against the UniClust30[106] database using HHBlits from HH-suite[107] (v2.0.16). If the number of effective sequences is < 128, Stage 2 is performed where the query sequence is searched against the Uniref50[108] database using JackHMMER from HMMER[109] (v3.1b2). Full-length sequences are extracted from the JackHMMER raw hits and converted into a custom HHBlits format database. HHBlits is applied to jump-start the search from the Stage 1 sequence MSAs against this custom database.

2.2.2 Features

The MSAs were generated for the unique chain sequences in the training(9,419), and test(1,889) sets using the method described in MSA Generation and stored in PSICOV[110] .aln format. The most commonly used features in sequence-based predictions were used. Token embeddings, Positional embeddings, and Segment embeddings were extracted from the sequence, while Position Specific Scoring Matrix, Information Content, Secondary Structure, and Solvent Accessibility were extracted from the generated, high-quality MSAs. The process for creating the feature map is shown in Figure 2.1

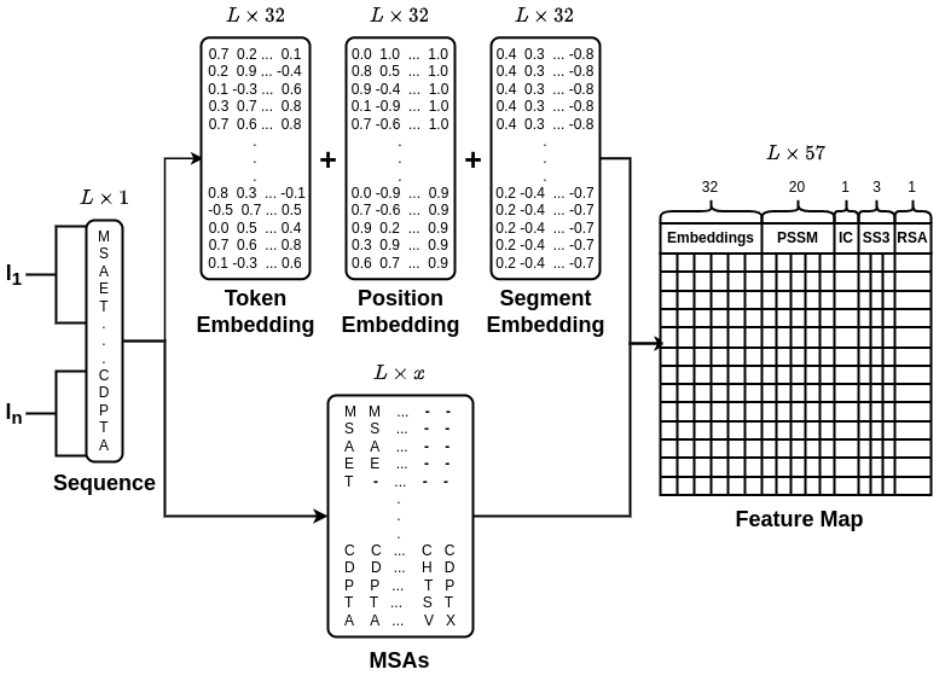

Figure 2.1: The process used for generating the feature map of BiRDS framework. Token, positional and segment embeddings are generated using just the sequence information. The features extracted from the MSAs of the individual protein chains created using DeepMSA, are concatenated to form the protein feature map

2.2.2.1 Token Embedding, Positional Embedding and Segment Embedding

There are 21 amino acids in the protein vocabulary of BiRDS, with the 20 standard amino acids labelled in alphabetical order from 1 to 20 and X, representing all non-standard amino acids, labelled as 0. Token embeddings help the model differentiate between the different types of amino acids. It is generated by an Encoding layer that uses the vocabulary label of each amino acid in the sequence. Positional Embeddings (PE) carry information about the absolute position of the amino acids in the sequence. Using the positional encoding layer of a Transformer network[5], these embeddings were unique for each position and generalised to long sequences without extra effort. A segment embedding was generated by using the chain number to which an amino acid belongs, to allow the model to differentiate between the multiple chains of a protein.

2.2.2.2 Position-Specific Scoring Matrix and Information Content

Position-Specific Scoring Matrix (PSSM) is a commonly used representation of patterns in biological sequences, derived as the log-likelihood of the probability that a particular amino acid occurs at a specific position. The PSSMs were derived from MSAs using Easel[111] and Heinikoff position-based weights so that similar sequences collectively contributed less to PSSM probabilities than diverse sequences. The information content (IC) of a PSSM gives an idea about how different the PSSM is from a uniform distribution. IC was also derived using Easel.

2.2.2.3 Secondary Structure and Solvent Accessibility

The secondary structure is defined by the pattern of hydrogen bonds formed between the amino hydrogen and carboxyl oxygen atoms in the peptide backbone. It gives an idea of the three-dimensional structure of the protein. The secondary structural elements are alpha helices, beta sheets and turns. PSIPRED (v4.0)[112] was used to predict the probability of each state of the 3-state secondary structure (SS3) for every amino acid in the sequence. The solvent-accessible surface area is the surface area of a biomolecule accessible to a solvent. SOLVPRED from MetaPSICOV 2.0[113] was used to predict the every amino acid's relative solvent accessibility (RSA). RSA can be calculated as
$RSA = ASA/MaxASA$, where ASA is the solvent-accessible surface area, and MaxASA is the maximum possible solvent accessible surface area for the amino acid residue.

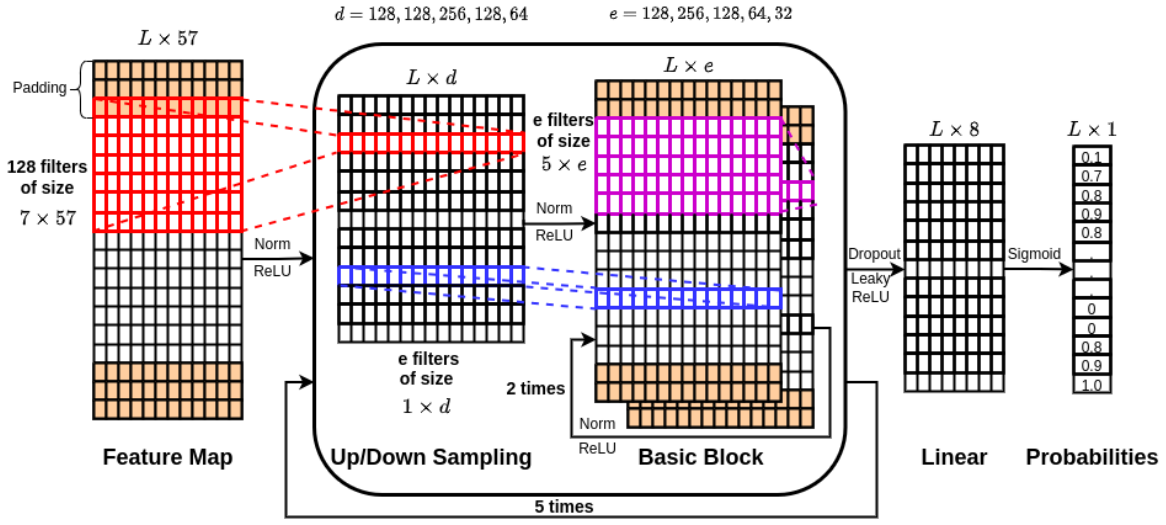

Figure 2.2: Architecture of the deep learning model, BiRDS

2.2.3 Model

2.2.3.1 BiRDS Architecture

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take an image as input, assign importance (learnable weights and biases) to various aspects/objects in the image, and differentiate one from the other. When multiple CNN layers are stacked on top of each other, Deep Neural Networks (DNNs) are formed. DNNs are challenging to train because of the vanishing gradient problem where the gradients become so small that the network's weights do not change, preventing further training. With the introduction of skip connections (shortcuts to jump over some layers) in CNNs, the vanishing gradient problem is avoided. CNNs with skip connections are known as Residual Neural Networks or ResNets[1]. ResNets use representation learning to extract the most important features for classification. They can also model long-range interactions and have been hugely successful in Computational Natural Sciences[44]. The architecture of the deep Residual Neural Network used here is shown in Figure 2.2.

Each sample protein in the dataset consists of one or more protein sequences. Let the length of the sequences be $l_1, ..., l_n$. Features are generated for each sequence in the protein (ordered by chain ID in PDB), leading to multiple vectors of shape $[l_i, 47]$ for the i^{th} sequence. These generated features are combined through simple concatenation, giving a final feature vector of shape $[L, 47]$ as input to the model, where $L = l_1 + ... + l_n$.

The feature vector is passed through the first level, consisting of a 1D convolutional layer with 128 filters of size 7, batch normalisation layer and ReLU (Rectified Linear Unit) activation function. The input is padded with zeroes to ensure that the length of the output vector remains the same. The filters of this layer stride along the length of the protein, considering the features of the three prior amino acids, the current amino acid, and the three subsequent amino acids (totalling 7). This stride allows for the extraction of the required information of the current amino acid based on the features of nearby amino acids.

The following five levels contain an up(down)sampling layer and two basic blocks. A basic block consists of a 1D convolutional layer, a batch normalisation layer, a ReLU activation function, a second 1D convolutional layer, a second batch normalisation layer, and a final ReLU activation function. The ResNet skip connection is made after the final ReLU activation, where the initial input to the first basic block is added to the output of the final ReLU activation. Usually, the input received by the first basic block will not match its required input size. Hence, an up(down)sampling layer ensures that the input to the first block has the required shape. The output of size $L \times d$ from the first level runs through e filters of size $1 \times d$ of the up(down)sampling layer to generate a vector of size $L \times e$. This vector is passed to the first basic block, which follows a similar stride policy as the first level but with a window size of 5. The process is repeated with the second basic block, and its output is sent back to the up(down)sampling layer. This process is repeated five times, with d going from $128 \rightarrow 128 \rightarrow 256 \rightarrow 128 \rightarrow 64$ and e going from $128 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32$. The multiple levels capture the long-range dependencies of amino acids since the filters help propagate information of one amino acid through its neighbours.

The last two levels contain simple, linear, fully connected artificial neural networks. The penultimate level has a LeakyReLU activation function with dropout to prevent sparse gradients. A sigmoid function at the end ensures that the model outputs values between $[0, 1]$, resulting in a vector of size L (length of the protein), denoting the probabilities of a residue being a part of the binding site.

2.2.3.2 Loss Function

There is a substantial imbalance in the two classes of binding and non-binding residues in this classification problem, where the percentage of binding residues is only 6%. Hence, a weighted binary cross-entropy loss function was used to train the model.

$$L(\hat{y}, y) = -(\alpha \hat{y} \log(y) + (1 - \hat{y}) \log(1 - y))$$

$\hat{y}$ is the vector of true labels, y is the model output probabilities, and α is the weight assigned to the rare class.

α heavily penalises the model if it incorrectly predicts binding residues as non-binding. α is calculated on the fly for every batch of inputs using $\alpha = \frac{n_{nbr}}{n_{br}}$, where n_{nbr} is the total number of non-binding residues in the batch and n_{br} is the total number of binding residues in the batch.

2.2.3.3 Implementation

The model is implemented using PyTorch Lightning[114], a wrapper on the popular open-source deep-learning library, PyTorch[115]. The model is trained in batches using an Adam Optimiser with the ReduceLROnPlateau scheduler and a learning rate warm-up where the learning rate is gradually increased to the actual learning rate. The implementation can be found at https://github.com/devalab/BiRDS.

2.2.4 Evaluation Metrics

2.2.4.1 Confusion Matrix

A confusion matrix is a table that allows for the visualisation of the performance of a supervised learning algorithm. The following terminologies can be defined in the binary classification of a residue as a binding residue (BR) or non-binding residue (NBR).

- True Positive (TP): Number of BRs predicted correctly as BRs.
- True Negative (TN): Number of NBRs predicted correctly as NBRs.
- False Positive (FP): Number of NBRs predicted incorrectly as BRs.
- False Negative (FN): Number of BRs predicted incorrectly as NBRs.

The following metrics can be derived from the confusion matrix

Accuracy: $ACC = \frac{TP+TN}{TP+TN+FP+FN}$
Precision: $PPV = \frac{TP}{TP+FP}$
Recall: $TPR = \frac{TP}{TP+FN}$
F1 score: $F_1 = \frac{2TP}{2TP+FP+FN}$
Intersection over Union: $IoU = \frac{TP}{TP+FN+FP}$
Matthews Correlation Coefficient: $MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

2.2.4.2 Accuracy, Precision, Recall

Accuracy (ACC) is the ratio of correct predictions to the total number of predictions. Precision (PPV) is the ability of a classifier to identify only relevant objects. Recall (TPR) is a metric which measures the ability of a classifier to find all the relevant cases (that is, all the ground-truths)

2.2.4.3 F_1 score, IoU

F_1 score is the harmonic mean of precision and recall. It maintains a balance between the precision and recall of the classifier. IoU, also called Jaccard index, is a metric that evaluates the overlap between the ground-truth and the predictions. It is commonly used in Object Detection.

2.2.4.4 MCC

The Matthew's Correlation varies from $[-1, +1]$, with $+1$ representing a perfect prediction, 0 representing no better than a random prediction and -1 representing total disagreement between the prediction and the observation. It is a common metric used in binary classification problems where there is a substantial imbalance in the class labels. It is a more reliable statistical rate which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset[116].

2.3 Results and Discussion

Ten models with the architecture described in BiRDS Architecture were trained through ten-fold cross-validation, where one fold formed the validation set while the remaining folds formed the training set in each iteration. The validation results are provided in Table 2.1 and the sum of confusion matrices in Figure 2.3. The Receiver Operating Characteristics (ROC) curve and the Precision-Recall (PR) curve of the models on their validation sets is provided in Figure 2.4 and 2.5. The description of the various metrics is provided in Evaluation Metrics.

Dataset	MCC	ACC	F1	IoU	PPV	TPR
Fold 1	0.354	0.920	0.394	0.582	0.359	0.437
Fold 2	0.606	0.931	0.633	0.695	0.545	0.755
Fold 3	0.521	0.896	0.565	0.641	0.474	0.700
Fold 4	0.270	0.898	0.323	0.544	0.296	0.355
Fold 5	0.324	0.892	0.367	0.556	0.293	0.490
Fold 6	0.338	0.884	0.373	0.555	0.282	0.550
Fold 7	0.324	0.902	0.368	0.562	0.309	0.456
Fold 8	0.340	0.924	0.380	0.578	0.355	0.407
Fold 9	0.380	0.918	0.421	0.591	0.378	0.475
Fold 10	0.355	0.917	0.391	0.579	0.332	0.476
Test (Full)	0.568	0.940	0.589	0.677	0.502	0.713
Test (Reduced)	0.440	0.951	0.464	0.626	0.497	0.436

Table 2.1: Validation and test results

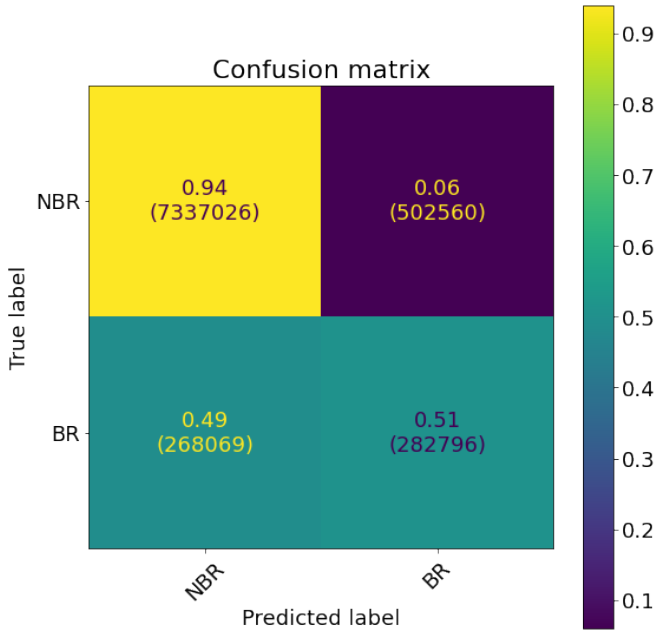

Figure 2.3: Sum of confusion matrices of the ten models on their corresponding validation sets

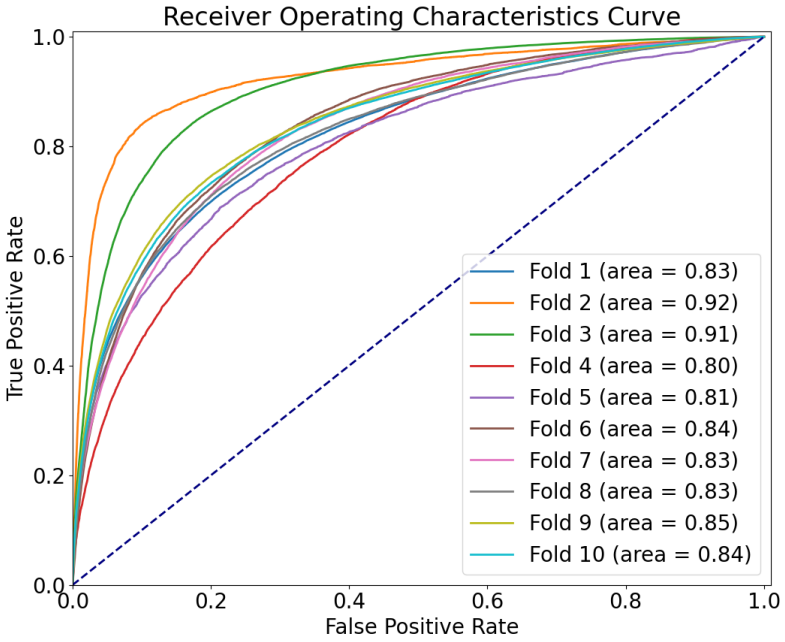

Figure 2.4: Receiving Operator Characteristics curve of the ten models on their corresponding validation sets

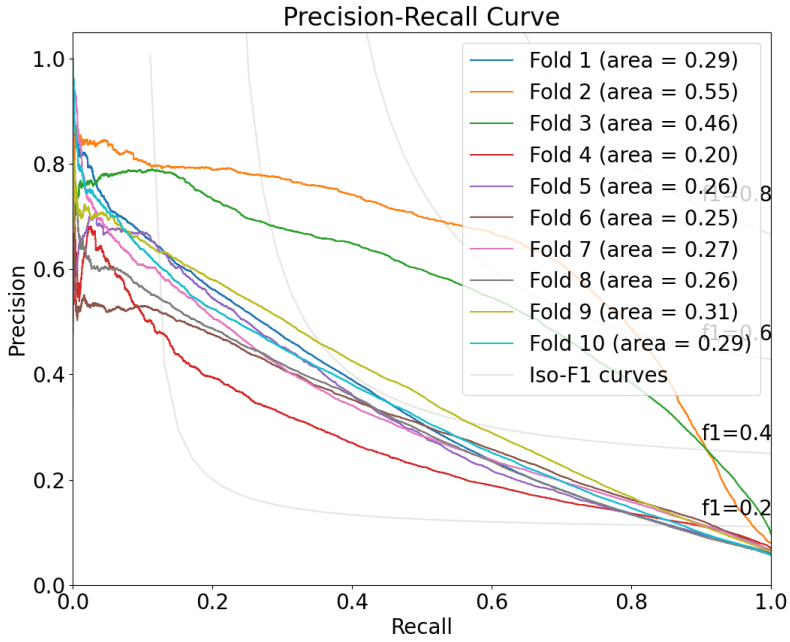

Figure 2.5: Precision-Recall Curve of the ten models on their corresponding validation sets

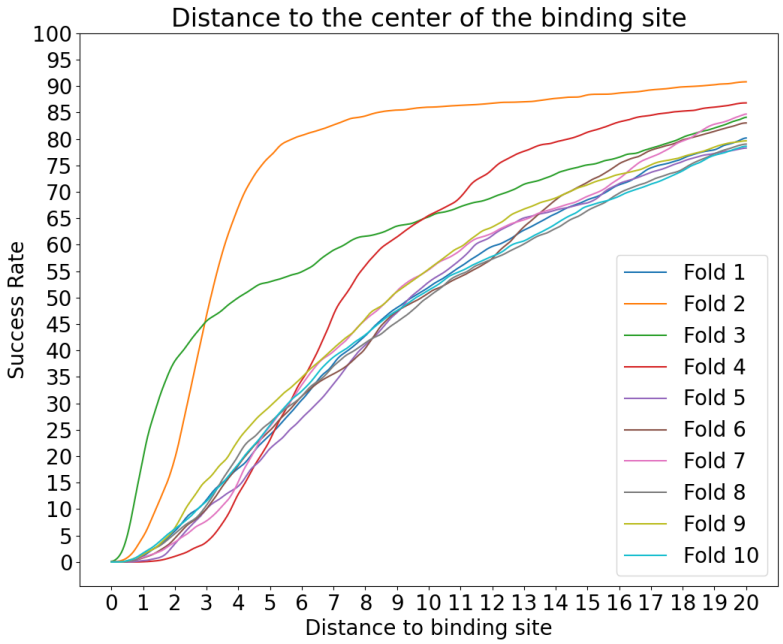

Figure 2.6: Success rate plot for various DCC thresholds of the ten models on their corresponding validation sets

The model predictions were also mapped back to the available 3D structures of proteins for DCC calculation. DCC is the distance between the centre of the predicted binding pocket and the centre of the actual binding pocket. It is commonly used for evaluating 3D-structure based models. The success rate of DCC is defined as the fraction of predictions below a given threshold. Pockets with DCC below 4Å are considered to be correctly predicted. Figure 2.6 denotes the success rate plot of the models' predictions on their validation set for various thresholds of the DCC metric. The success rate ranges from 15% to 75% when the threshold is 4Å. Fold 2 and Fold 3 models performed well on their validation sets since they contained only 1 to 5 protein families with similar sequence patterns. The presence of only a few families in these folds is due to the equal sum partition algorithm used to create these folds. It is a greedy algorithm that combines as many large clusters as possible, thus causing large families to appear in a single fold.

The ten trained models are run on the full and reduced test sets for testing. The models come to a consensus if five or more models predict a residue as belonging to the most active binding site of the proteins in a set. The test results, both full and reduced, are provided in Table 2.1 and the confusion matrix on the reduced test set in Figure 2.7 and on the full test set in Figure 2.8.

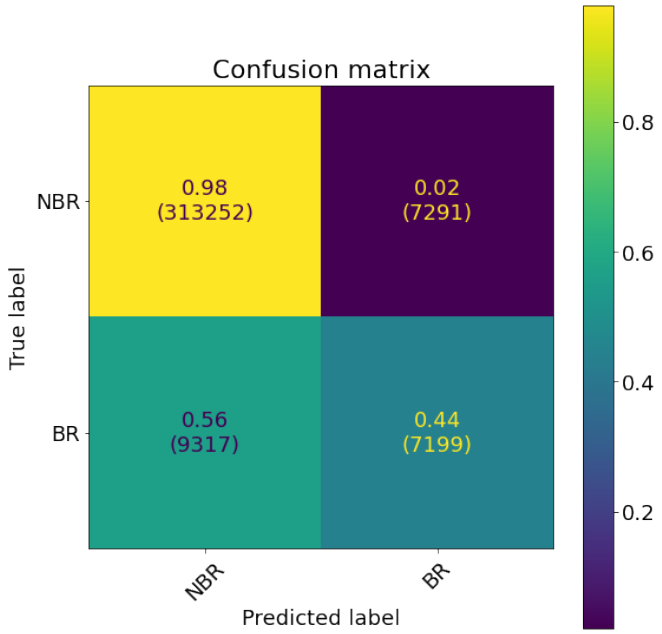

Figure 2.7: Confusion matrix on the reduced test set after consensus among models

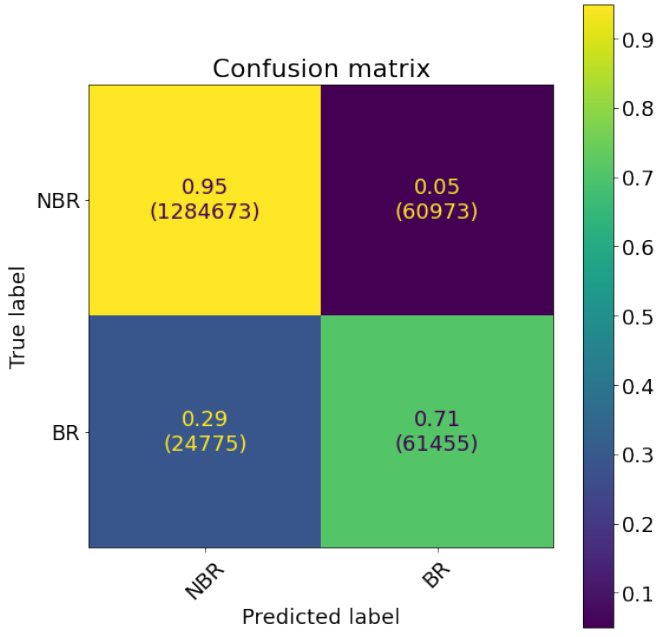

Figure 2.8: Confusion matrix on the full test set after consensus among models

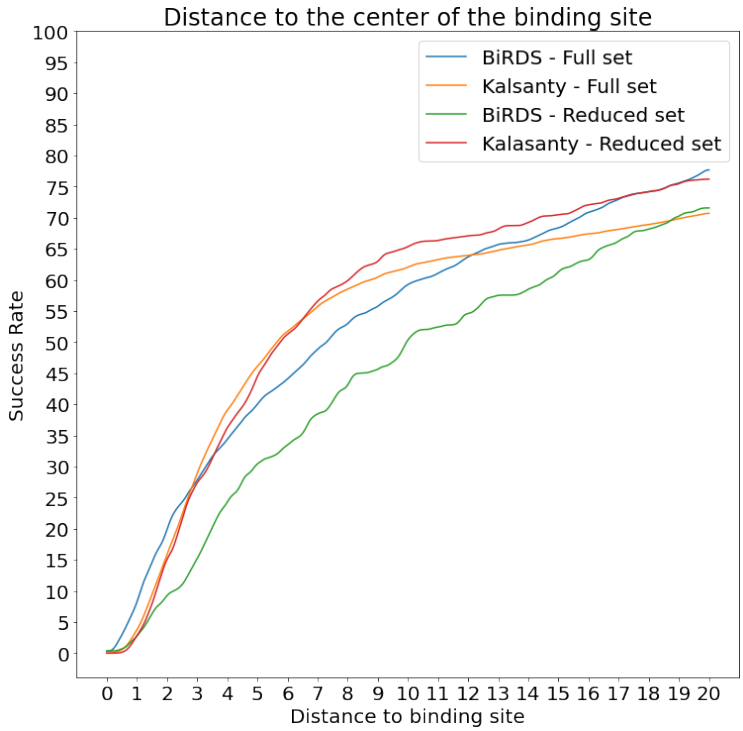

Figure 2.9: Success rate plot for various DCC thresholds on the test set after averaging the predictions of the 10 models

The performance of BiRDS on the novel SC6K test set was compared against Kalasanty[96] and SCRIBER[82]. Kalasanty is a 3D-structure-based method that uses a U-Net architecture[117] capable of protein binding site segmentation. The full test set was run on Kalasanty using their open-source code, and the DCC metric was calculated for the predicted pocket. The success rate plot of DCC is shown in Figure 2.9. BiRDS performs on par with Kalasanty on the full test set, which will have a lot of sequences similar to the training data. However, the performance on the reduced test set shows Kalasanty outperforming BiRDS. Nevertheless, BiRDS still performs well on the reduced test set for a sequence-based predictor, achieving a success rate of 25% at a 4Å cutoff for DCC. In other words, for 25% of the test data, the model has predicted the binding site such that the centre of the predicted binding site is within 4Å of the centre of the most ligandable binding site. As the threshold of DCC increases, the success rate also naturally increases. It should be noted that if the model predicts the whole binding site correctly and misses out on a couple of residues or predicts more residues, the centre of the predicted binding site may shift significantly.

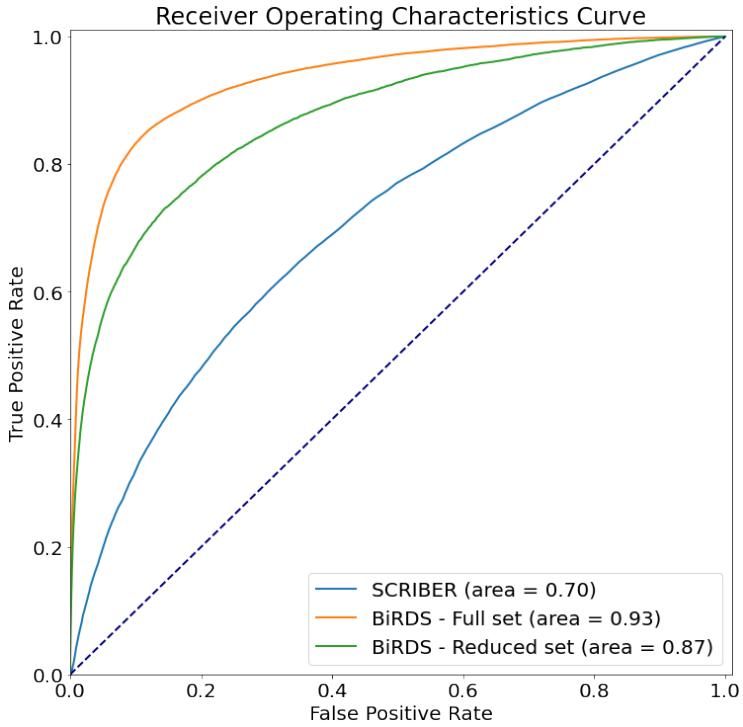

Figure 2.10: ROC curve of BiRDS and SCRIBER on the test sets

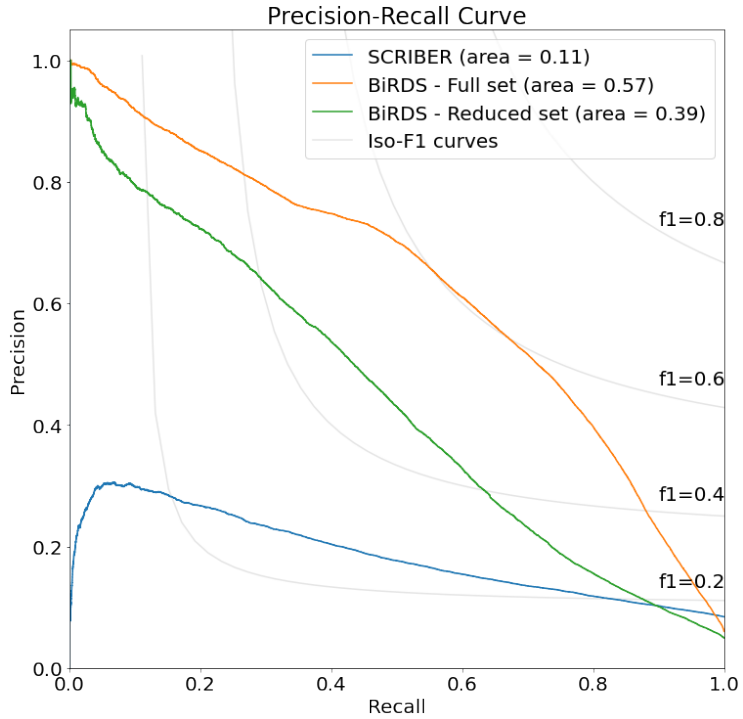

Figure 2.11: PR curve of BiRDS and SCRIBER on the test sets

SCRIBER is a sequence-based, two-layer architecture, machine learning predictor which predicts propensities of protein-binding, RNA-binding, DNA-binding, and ligand-binding residues. The predictor was trained on individual chain sequences of a protein, based on their Uniprot IDs. For a fair comparison with BiRDS and to speed up prediction time on their webserver, the 1,889 unique chain sequences of the test set were filtered; sequences with length greater than 1,024 and sequences with sequence similarity greater than 25% and over 80% overlap with the SCRIBER training set and SC6K test set were removed. SCRIBER predictions of RNA-binding, DNA-binding and ligand-binding residue propensities on the final 521 sequences were averaged and considered for comparison. The Receiver Operating Characteristic (ROC) curve and the Precision-Recall (PR) curve of BiRDS on the full and reduced test set, and SCRIBER on the 521 sequences, is shown in Figure 2.10 and 2.11.

A variety of more complex deep-learning models were trained to improve predictions. As described in the paper by Cui et al., a Complementary Generative Adversarial Network (CGAN) was implemented to mitigate the substantial imbalance in the prediction classes. However, a simple weighted binary cross-entropy loss function worked better than a CGAN with focal loss. A Deep Bidirectional Encoder Representations from Transformers (BERT)[118], a state-of-the-art model for token classification problems in NLP, was also implemented. It performed on par with the current BiRDS model but led to longer training times. Several different features to improve performance were also tried. Task Assessing Protein Embeddings (TAPE)[119] provided trained deep learning models which produced an embedding representation of the protein sequence input. The trained TAPE transformer model was added along with BiRDS architecture, but the training could not proceed due to a large-sized feature map and insufficient GPU memory. SPOT-1D[32] is a sequence-based predictor for predicting secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps. These predictions were used as inputs to BiRDS but did not provide any improvement over the features extracted from Deep MSAs.

Some case studies were undertaken to show that the model's performance is good, but the metrics do not rate it well due to the limitations of the dataset. The aggregated predictions of the ten models on the test set were mapped back to the three-dimensional structure of the protein-ligand complex. 3Dmol.js[120], a modern, object-oriented Javascript library for visualising molecular data, was used to visualise the protein's surface, with coloured residues representing the predicted and actual binding residues. In the following examples, red indicates an incorrect prediction of a non-binding residue as binding, blue indicates a binding residue that was not predicted as binding, and green indicates a correct prediction.

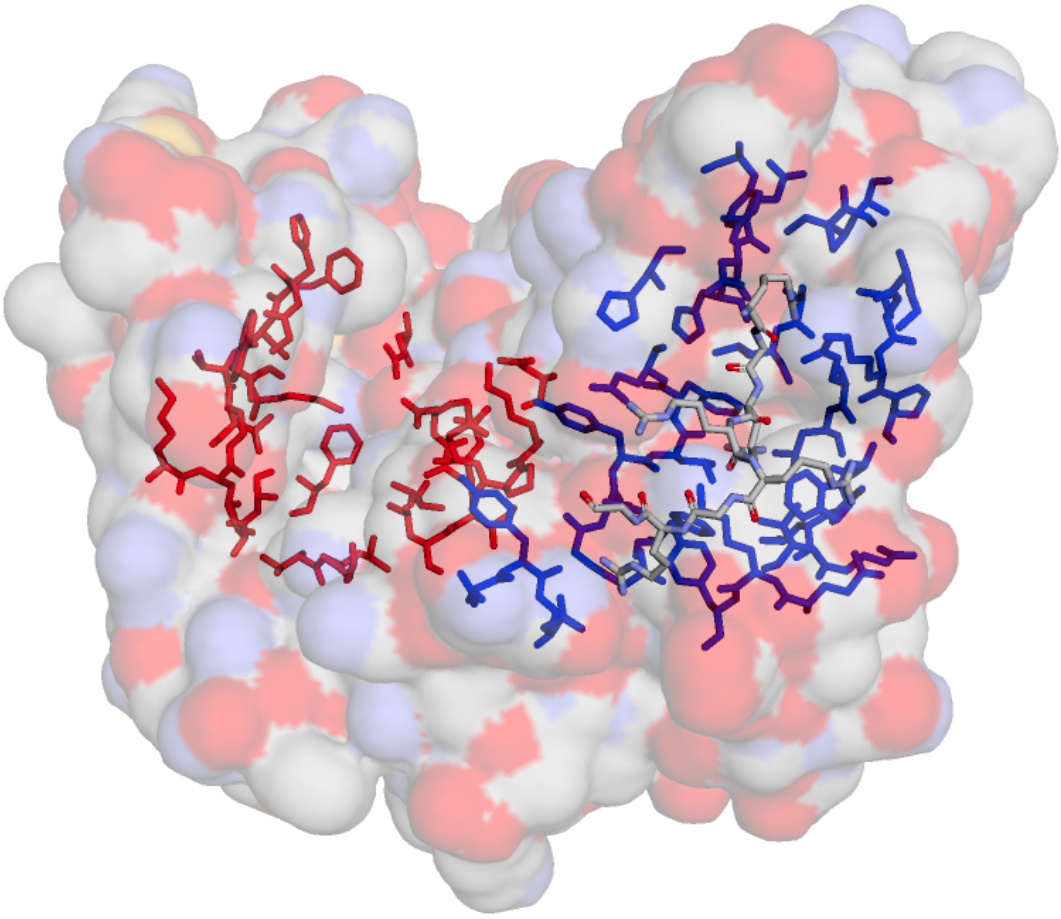

Figure 2.12: 6FAD - BiRDS seems to be incorrectly predicting the actual binding site (in blue), when in reality, it is predicting another binding site of the protein (in red)

In Figure 2.12, BiRDS seems to incorrectly predict all the binding residues for 6FAD[121]. However, it is predicting another binding site of the protein. The sc-PDB[2] dataset was generated through a series of filters, and the residues surrounding the most buried ligand was selected to be the most ligandable binding site. This selection, unfortunately, is a flaw of the dataset and the method used for predictions. There is no right way to cover cases like these where the model needs to be penalised less when it predicts a binding site that is not the most ligandable binding site. Hence, the evaluation metrics will generally give an abysmal score for such cases.

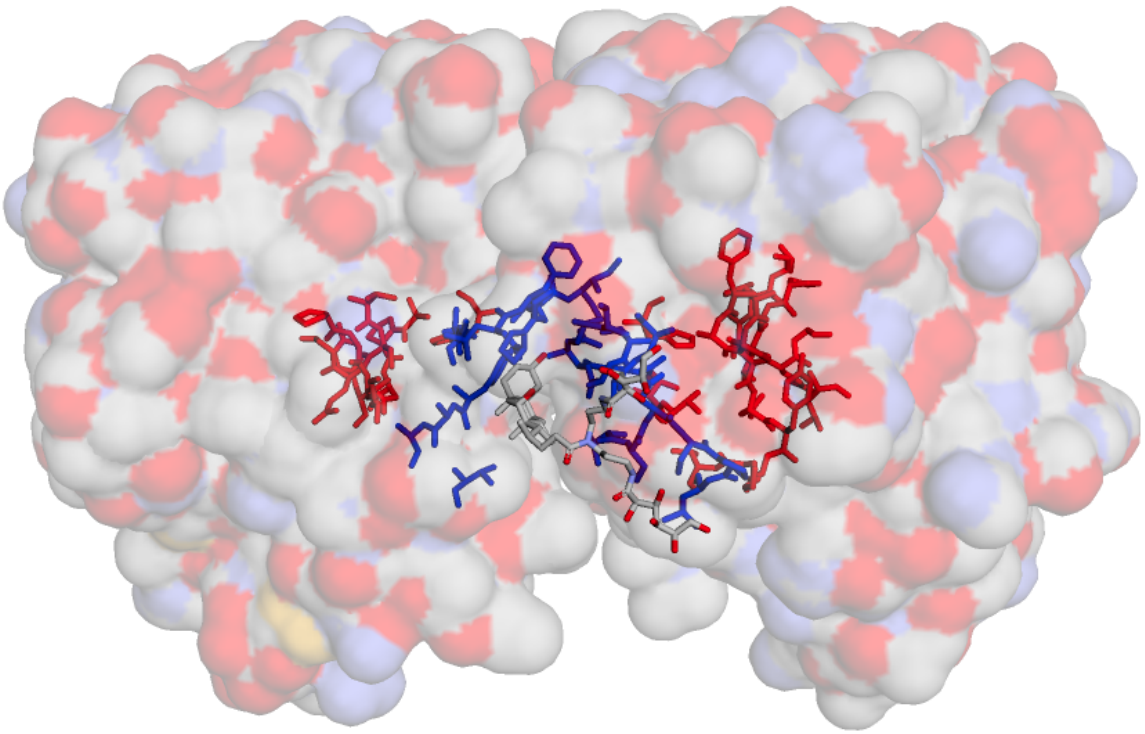

Figure 2.13: 6ISP - BiRDS is able to predict the binding site of individual chains (in red), but not the binding site formed due to the interaction between chains (in blue)

Figure 2.13 shows 6ISP[122], where BiRDS predicts individual binding sites of two same sequence chains of the protein. However, the model finds it challenging to predict the binding site created due to the interaction between the two chains. This may likely be due to the way the input features are generated. A simple concatenation of the features of individual chains to generate the protein sequence features is insufficient as it does not provide any information about the interaction among the multiple chains. These interactions scarcely occur in the training set, making it hard for BiRDS to learn.

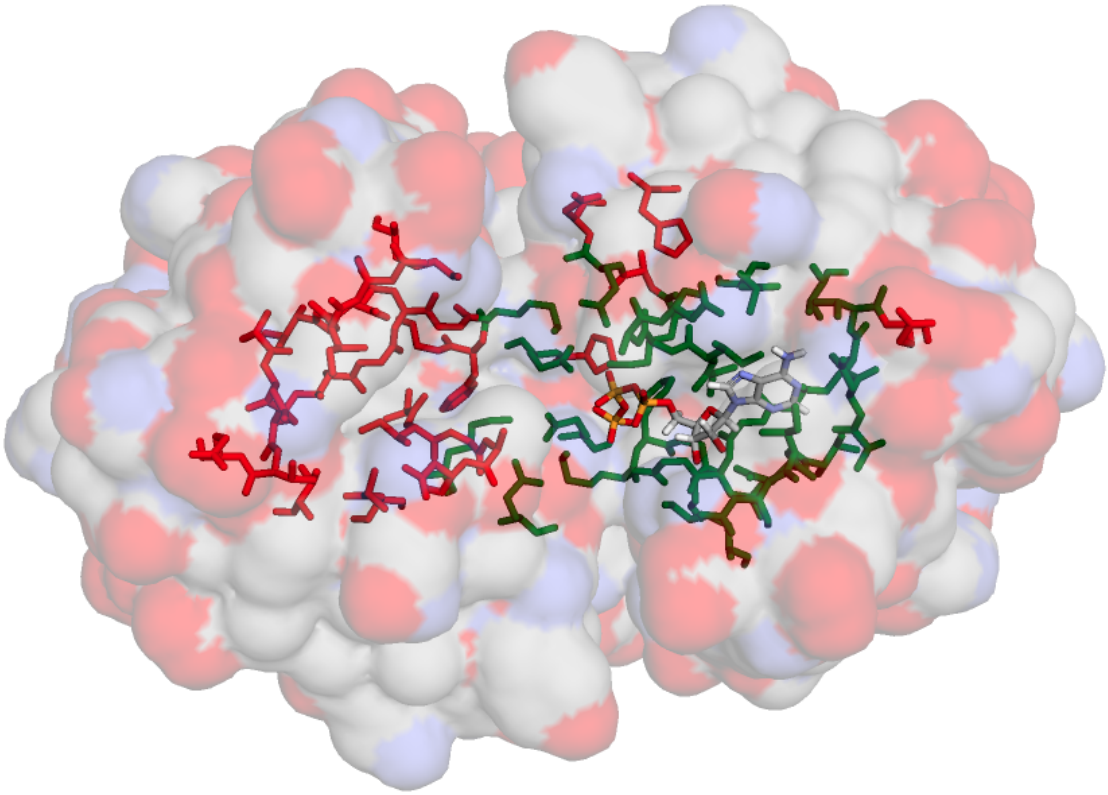

Figure 2.14: 6S2J - BiRDS predicts the binding site correctly, but due to the presence of same sequence protein chains, it predicts both the binding sites (in green and red)

Figure 2.14 shows 6S2J[123], where BiRDS predicts the binding site of a protein chain with high precision. It predicts most of the binding residues surrounding the ligand and a couple of outliers. However, the two protein chains have the same sequence, causing BiRDS to predict similar binding sites for both. Since sc-PDB selects only one active binding site during its selection process, the model predictions are compared against a single site for metrics calculation. The metrics do not do justice to these types of predictions, penalising BiRDS with a poor score.

2.4 Conclusion

In this study, a deep ResNet was implemented to predict a protein's most active binding site. A training set of ten folds was derived from the sc-PDB(v. 2017)[2] database containing data of a protein's most ligandable binding site. A novel test set SC6K was constructed from protein-ligand complexes of the PDB from January 2018 to February 2020. MSAs were generated for all unique protein chains in both the datasets using DeepMSA, and features such as Position-Specific Scoring Matrix, Secondary Structure and Solvent Accessibility were extracted. The individual features of the chains were concatenated to form the protein feature map, and BiRDS was trained using 10-fold cross-validation and a weighted binary cross-entropy loss function. BiRDS can accurately predict the most active binding site of a protein using only sequence information. It outperforms SCRIBER, a sequence-based protein-binding site predictor and performs on par with Kalasanty, a 3D-structure-based method. It becomes crucial to determine the pocket where the drug molecule binds with the protein in drug design. BiRDS can be used for early and quick determination of the binding site before the availability of the protein structure.

2.5 Data and Software Availability

The source code has been written in a modular fashion using PyTorch Lightning[114]. The method implementation, data and pretrained models can be found at
https://github.com/devalab/BiRDS.

Chapter 3

Secondary Structure Prediction

3.1 Introduction

The function of a protein is directly related to its native 3D structure, which often implies that similar structures have similar functionality[124]. There are four levels of amino acid organisation: primary, secondary, tertiary and quaternary structures. The primary structure of a protein refers to the linear sequence of amino acid residues. The secondary structure is a course-grained descriptor of the local structure of the polypeptide backbone. The secondary structure involves hydrogen bonds along the backbone that cause the long chain to fold into local shapes, mainly coils(C), sheets(E), and helices(H). Tertiary structure is the three-dimensional structure of a protein. The quaternary structure further stabilises the protein molecule by bonding with one or more similar tertiary structures.

There have been three generations of methods used for the prediction of the secondary structure of a protein. The first generation utilised statistical propensities of amino acids residues towards a specific secondary structure class. Chou-Fasman's method [125] is an example from this generation. The second generation of methods used sophisticated statistical methods such as graph theory, neural networks, logic-based machine learning and Bayesian statistics. A sliding window was used to take the information of neighbouring residues into account. Some examples from this generation include Garnier-Osguthorpe-Robson(GOR) method[126] and the Lim method[127]. The above two generations did not provide great accuracies. The third generation uses evolutionary information derived from multiple sequence alignments of the query sequence and advanced machine learning models. Some examples of such methods include PSIPRED [19], SPIDER3 [30], and ProteinUnet[128]. These new methods provide a high degree of accuracy, reaching up to 85%.

An ANN is an Artificial Neural Network created by at least 2-3 layers of neurons. The initial/input layer introduces input variables into the network. The hidden/inner layers are where the values of complex matrix computations are stored. The final layer is the output layer, which may contain units for carrying out output classification. Deep learning models stack layers of ANNs based on the principle of hierarchy of concepts which states that complex concepts are learned by building them from simpler ones [129]. Deep learning has surpassed other statistical methods in almost every domain, allowing to build such intricate relations from data, infeasible for traditional machine learning algorithms. It has had immense success in Computational Natural Sciences and has been used in many areas of science and engineering.

This study uses a Deep Neural Network called a Transformer to predict the protein secondary structure from its sequence.

3.2 Methods

3.2.1 Dataset

The dataset used is exactly the same as used in previous studies [30] [130]. The full dataset contains 5789 proteins with a sequence similarity cut off 25% and X-ray resolution better than 2.0Å. The dataset was split into two sets by Heffernan et al.: 4590 proteins were randomly selected to be the training set (TR4590), and the remaining 1199 were used as the independent test set (TS1199). In this study, the training set was further split into ten random sets for 10-fold cross-validation.

For visualisation of the predictions made by the model, the PDBs of all the proteins in the test dataset was downloaded from the RCSB [63] website. The Needleman-Wunsch dynamic programming algorithm[100] is used to align the sequence extracted from the structure file to the sequence present in the test set. The protein structure file is then reindexed, based on this alignment, to match the indexing of the sequence provided in the database.

	n_{prot}	aa_{coils}	aa_{sheets}	$aa_{helices}$
Train	4,590	393,265	234,267	376,313
Test	1,199	102,518	58,998	98,146

Table 3.1: Summary of the dataset used for training and testing for secondary structure prediction

3.2.2 Features

3.2.2.1 MSA Generation

Collections of multiple homologous sequences (called Multiple Sequence Alignments or MSAs) can provide critical information for modelling the structure and function of unknown proteins. DeepMSA [98] is an open-source method for sensitive MSA construction, which has homologous sequences and alignments created from multiple sources of databases through complementary hidden Markov model algorithms.

The search is done in 2 stages. In stage 1, the query sequence is searched against the UniClust30 [106] database using HHBlits from HH-suite[107] (v2.0.16). If the number of effective sequences is $<$ 128, Stage 2 is performed where the query sequence is searched against the Uniref50 [108] database using JackHMMER from HMMER [109] (v3.1b2). Full-length sequences are extracted from the JackHMMER raw hits and converted into a custom HHBlits format database. HHBlits is applied to jump-start the search from Stage 1 sequence MSA against this custom database.

3.2.2.2 Position Specific Scoring Matrix and Information Content

Position Specific Scoring Matrix (PSSM) is a commonly used representation of patterns in biological sequences. PSSMs are derived from MSAs using Easel [111] and Heinikoff position-based weights so that similar sequences collectively contributed less to PSSM probabilities than diverse sequences. The information content (IC) of a PSSM explains how different the PSSM is from a uniform distribution. IC is also derived using Easel.

3.2.2.3 Amino Acid Embeddings

Word2Vec [131] is a method in NLP (Natural Language Processing) for obtaining an efficient estimation of word representations in a vector space. The same methodology can be applied by considering amino acids as words, protein chains as sequences and the quarternary structure as a paragraph.

A shallow Common Bag of Words (CBOW) neural network is trained using a huge corpus of amino acid sequences, obtained from the RCSB website[62][63]. The network takes the context of each word as the input and tries to predict the word corresponding to the context. In this process of predicting the target word, the vector representation of the word is created. Gensim [132] is a fast library for training vector embeddings. Protein sequences were yielded as sentences to Gensim to train a Word2Vec model for generating the amino acid embeddings. The vector representation of an amino acid is projected into two dimensions using T-SNE, and the projection is visualised using a scatter plot.

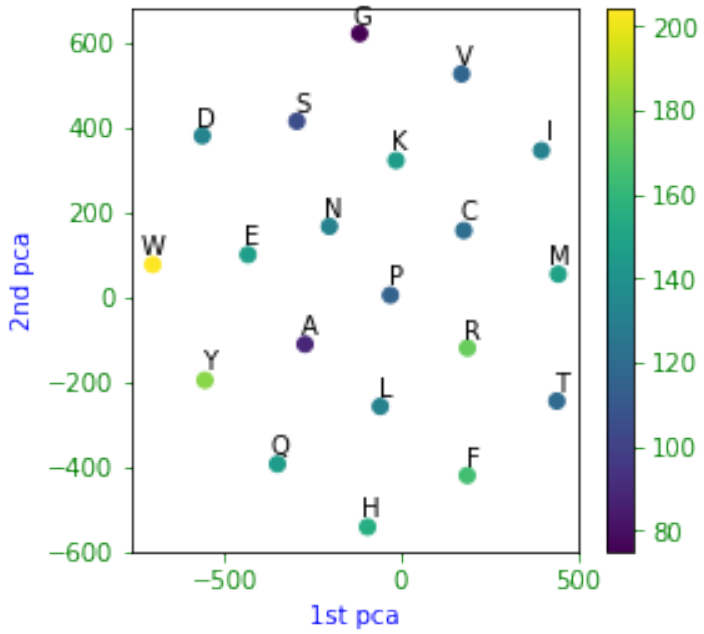

Figure 3.1: T-SNE 2D projection of the vector representation of amino acids coloured by mass

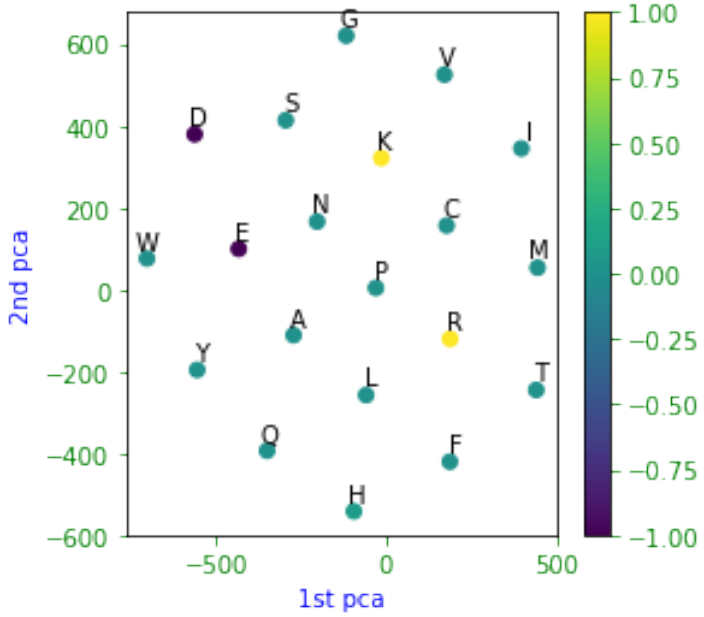

Figure 3.2: T-SNE 2D projection of the vector representation of amino acids coloured by charge

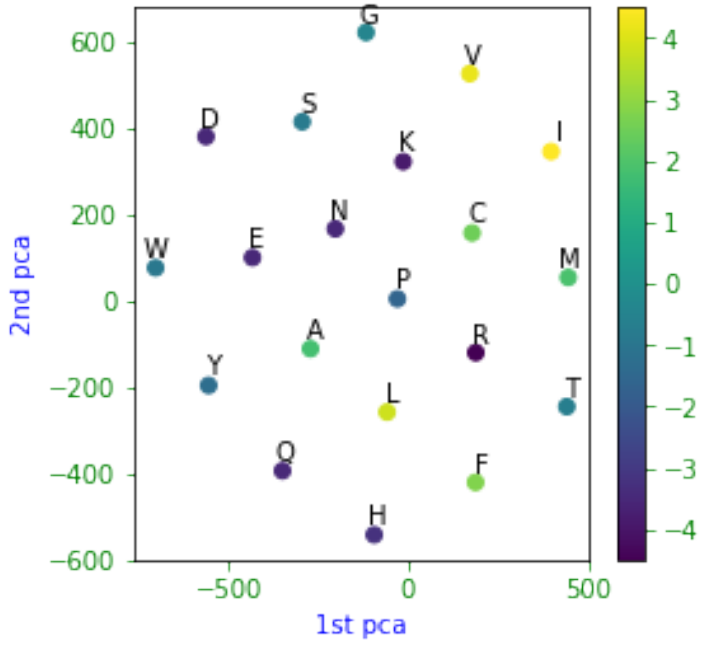

Figure 3.3: T-SNE 2D projection of the vector representation of amino acids coloured by hydrophobicity

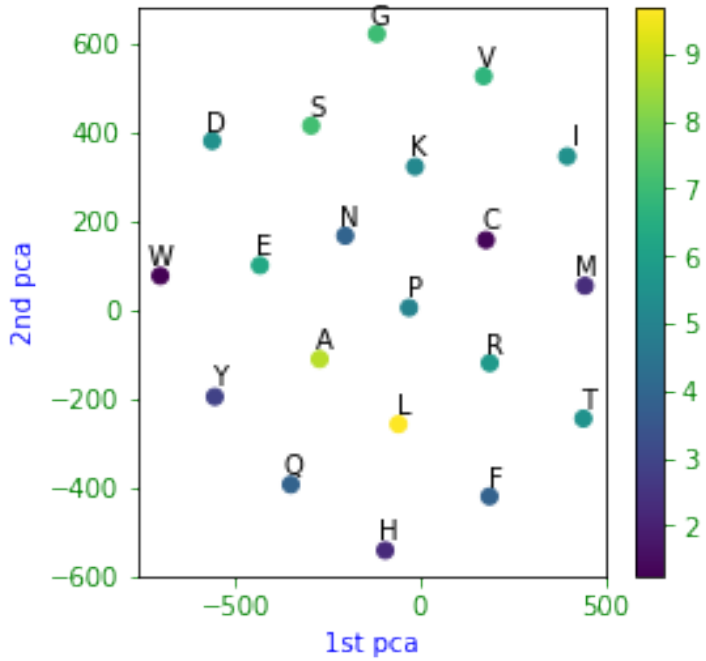

Figure 3.4: T-SNE 2D projection of the vector representation of amino acids coloured by occurrence

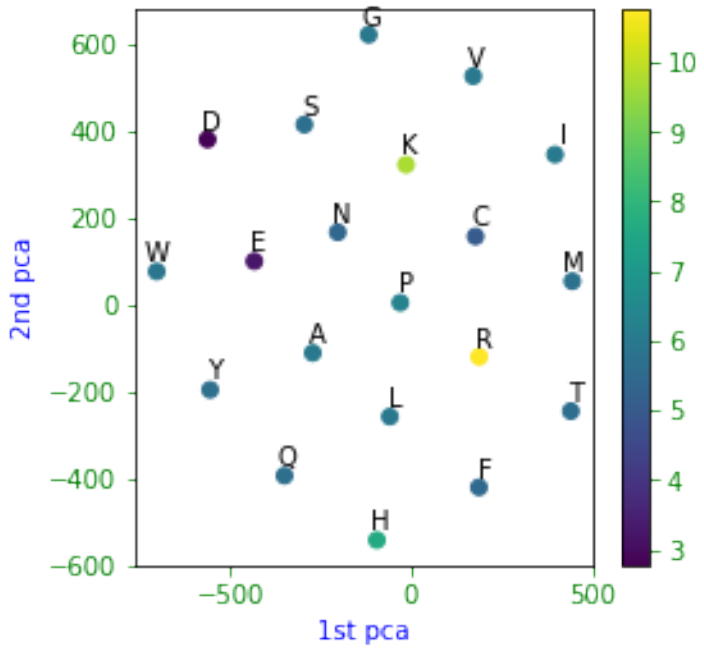

Figure 3.5: T-SNE 2D projection of the vector representation of amino acids coloured by isoelectric point

Figures 3.1 (mass), 3.2 (charge), 3.3 (hydrophobicity), 3.4 (occurrence), 3.5 (isoelectric point) represent the T-SNE projection of an amino acid vector representation. The colours represent the quantity of a particular property. It can be seen that amino acids with similar colours are closer together, indicating that the embeddings capture the properties of an amino acid very well, grouping amino acids with similar properties together.

3.2.3 Model

3.2.3.1 Architecture

A transformer neural network[5] is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies. It adopts the mechanism of attention, differentially weighing the significance of each part of the input data. The transformer architecture is composed of a stack of transformer layers, each containing a transformer sub-layer. The sub-layer is a multi-head self-attention mechanism that uses a combination of feedforward and convolutional layers to perform the necessary computations. A feedforward layer then transforms the output of the sub-layer to project it to the next layer.

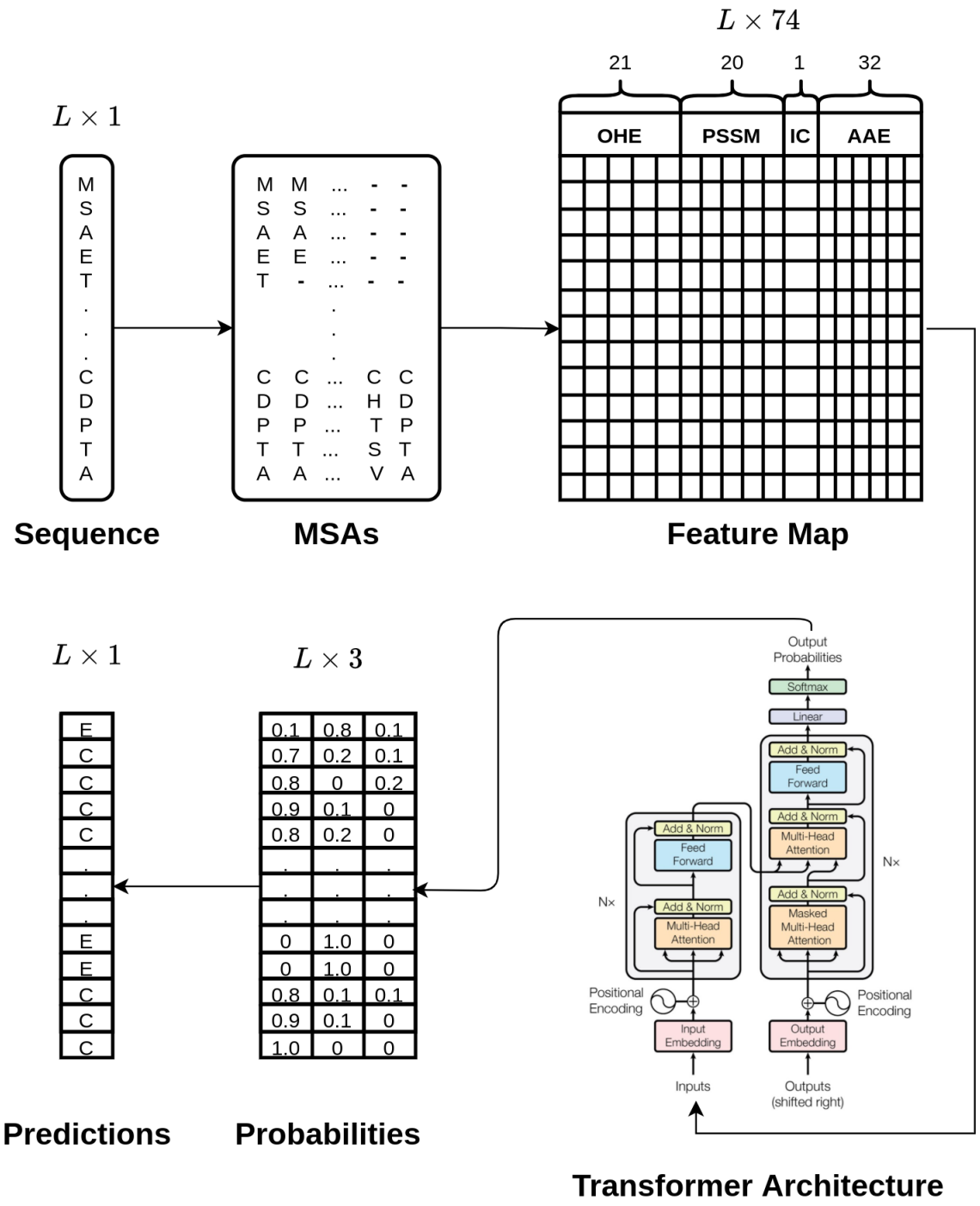

Figure 3.6: Model architecture for secondary structure prediction. Transformer image taken from [5]

Every sample in the dataset is a single protein chain, and the MSAs were generated for the same. PSSM was calculated from the generated MSAs. The feature map was created by concatenating the amino acid embeddings and the PSSM. The feature map was passed through a GRU (Gated Recurrent Unit) to overcome the vanishing gradient problem, and then positional encodings were added to the output. The GRU output with positional encoding was then provided to a Transformer network to capture the local and global properties of the sequence. The output of the Transformer network was then passed through a fully connected layer to project it to a single dimension and then through a softmax layer to project it to a probability distribution. The output of the softmax layer was then used to calculate the loss.

3.2.3.2 Loss Function

The loss function is calculated as the sum of the weighted cross-entropy loss of each class.

$$L(\hat{y}, y) = -\sum_{c=0}^{2} (\alpha_c \hat{y_c} \log(y_c)$$

c represents the class of amino acid, 0 representing a coil, 1 representing a beta-sheet, and 2 representing an alpha helix.

$\hat{y_c}$ is the vector of true labels where an amino acid belongs to class c.

y_c is the model output of probabilities of a residue belonging to class c

α_c is the weight that is assigned to the class c

3.2.4 Evaluation Metrics

3.2.4.1 Confusion Matrix

A confusion matrix is a table that allows for the visualisation of the performance of a supervised learning algorithm. The x-axis represents the predicted class, and the y-axis represents the true class. The diagonal elements represent the number of times a particular pair of classes were predicted correctly. In contrast, the non-diagonal elements represent the number of times a class was mistaken for another class. The sum of the diagonal elements represents the total number of correct predictions.

3.2.4.2 Accuracy, Precision, Recall

Accuracy (ACC) is the ratio of correct predictions to the total number of predictions. Precision (PPV) is the ratio of correctly predicted instances to the total number of predicted instances. Recall (TPR) is the ratio of correctly predicted instances to the total number of instances in the true class.chicco2020advantages

3.3 Results and Discussion

The training dataset was split into ten folds, and ten models with the same architecture were trained. One fold formed the validation set, and the remaining folds formed the training set for each model. The validation results are provided in Table 3.2, along with the confusion matrix in Figure 3.7. The table shows that the models learn almost similarly with accuracies ranging between 80.5% to 82%. This may be due to the random split of the dataset to generate the folds, causing similar protein sequences to be present in both the validation and train sets.

Dataset	ACC(%)	PPV(%)	TPR(%)
Fold 1	81.83	92.99	92.89
Fold 2	80.79	92.48	92.38
Fold 3	81.02	92.54	92.63
Fold 4	81.15	92.58	92.78
Fold 5	81.12	92.69	92.62
Fold 6	81.32	92.77	92.59
Fold 7	80.40	92.41	92.18
Fold 8	81.38	92.66	92.75
Fold 9	80.15	92.17	92.35
Fold 10	81.05	92.48	92.66
Test	82.51	93.19	93.23

Table 3.2: Validation results of all 10 trained models and test results

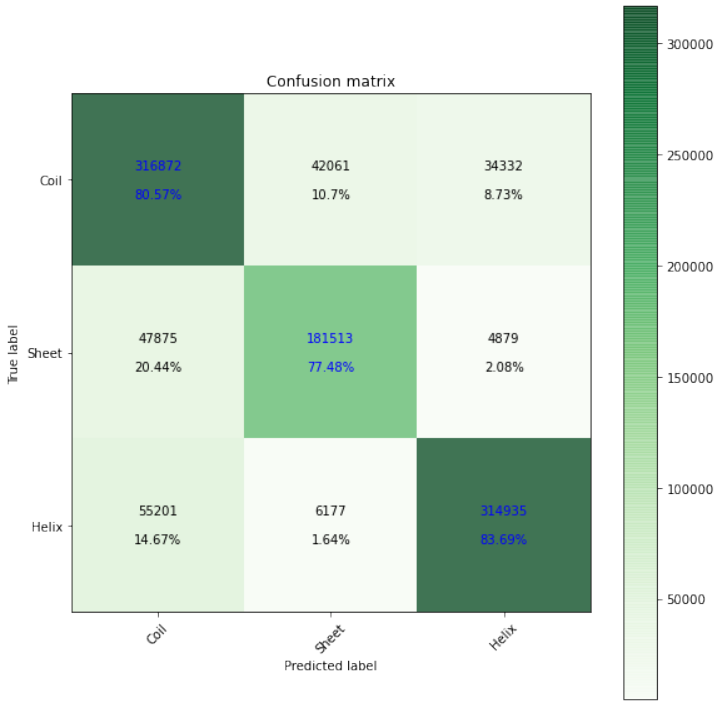

Figure 3.7: Sum of confusion matrices of the 10 models on their corresponding validation set

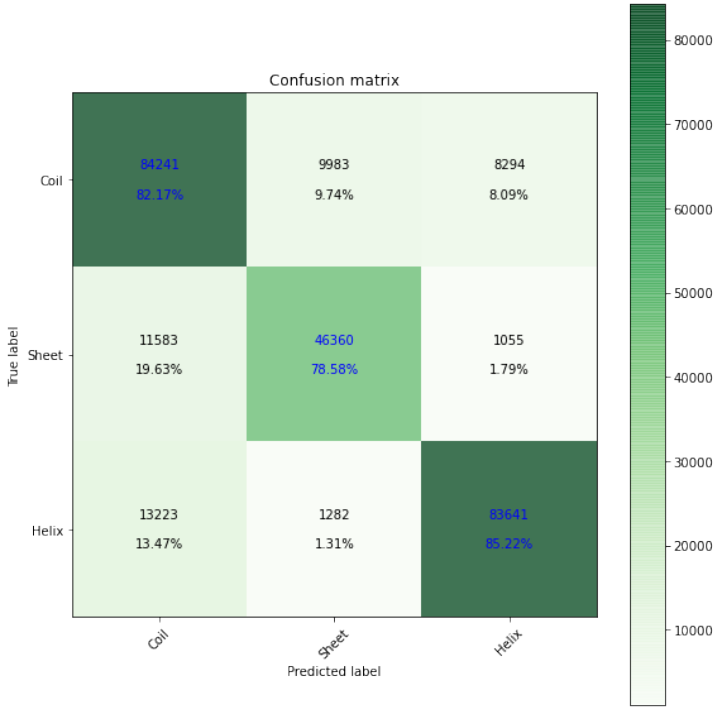

Figure 3.8: Confusion matrix on the test set after averaging the predictions of the 10 models

For testing, the consensus algorithm chosen is as follows. The ten trained models are run on the test set, the mean of the probabilities of each class is taken, and the class with the maximum probability is considered the class to which the amino acid belongs. The test results are also provided in Table 2.1, along with the confusion matrix in Figure 3.8. The consensus algorithm works well and provides a higher accuracy on the test set than the validation sets, indicating that the models are not suffering from overfitting on the training data.

SPIDER2 [22] employed a deep neural network consisting of three hidden layers with 150 hidden nodes in each layer. The weights were initialised by stacked sparse autoencoder and then refined by standard back-propagation through fine-tuned supervised training. SPIDER2 reported accuracy of 81.8% on the test set. An improved version of SPIDER2, called SPIDER3 [30] was released later on. SPIDER3 used long short-term memory (LSTM) bidirectional recurrent neural networks (RNNs) to capture non-local interactions for improving their predictions on the 3-state secondary structure. It recorded an accuracy of 84.16% on the test set.

Even as more data is becoming available and more sophisticated machine learning models are being trained, the accuracy of secondary structure predictions has not improved as much, meaning that the accurate prediction of the Q3 classification of secondary structures is possibly saturated. SPOT-1D [32], after using a predicted interresidue contact map as additional input and an ensemble of recurrent and residual convolutional neural networks was able to achieve 86% on the test set.

The transformer network was used for a token classification (named entity recognition) problem, where each amino acid (token) in the sequence was given a class. According to a recent survey on deep learning for Named Entity Recognition [133], Bidirectional Encoder Representations from Transformers (BERT) performs very well on this task and can be explored as a prospect for the secondary structure prediction task.

Some case studies were undertaken to see where the model performs well and where it performs poorly. In the following figures: A coil turn predicted incorrectly is indicated by Red, while Orange indicates a correct prediction. A beta-sheet predicted incorrectly is indicated by Green, while Yellow indicates a correct prediction. An alpha helix predicted incorrectly is indicated by Blue, while Purple indicates a correct prediction.

From the case studies in figures 3.9, 3.10, and 3.11, it is clear that the model does not perform well at the borders of conversion from one structure to another and on short bursts of beta-sheets and alpha-helices, indicating that the model has learnt to identify the general structure of the protein but finds it challenging to find the transition point from one secondary structure to another.

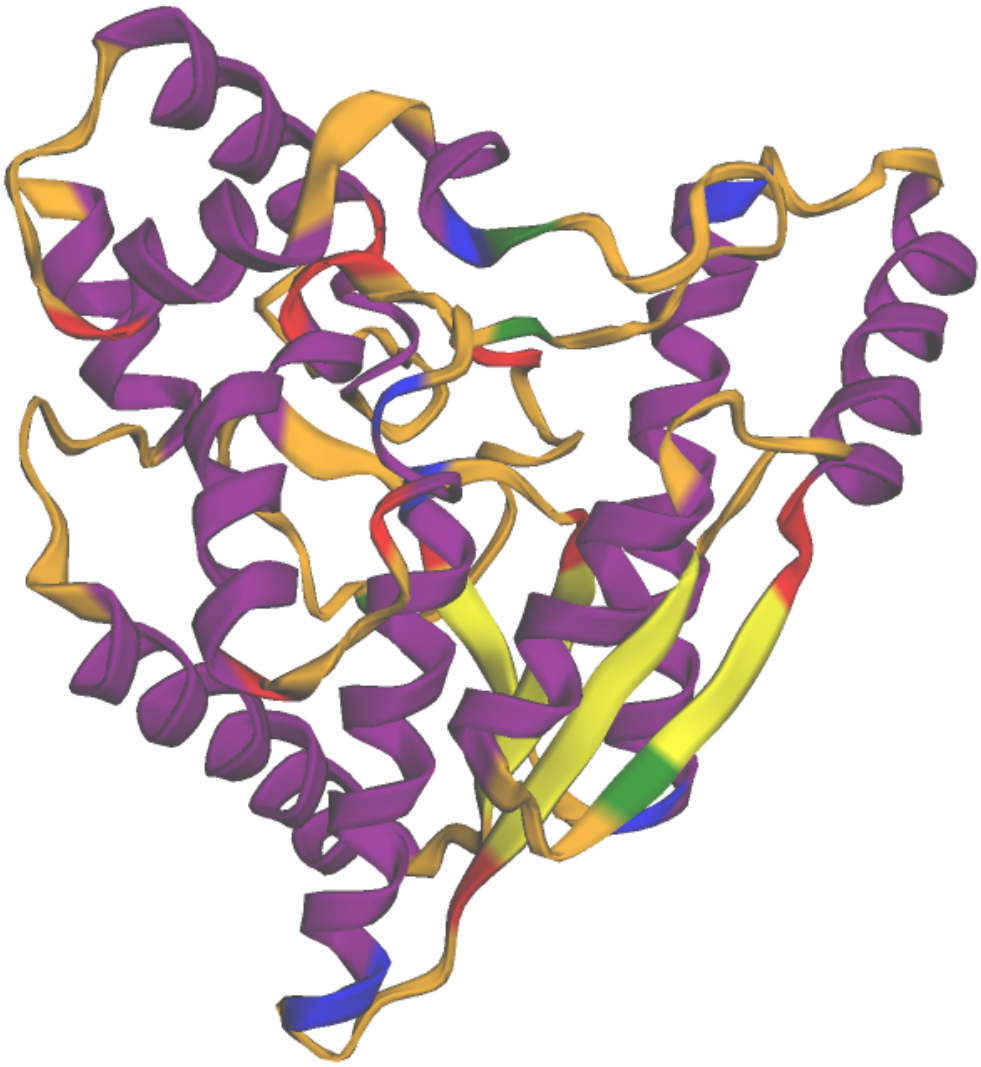

Figure 3.9: 1KQP:A - The protein chain consists of all the secondary structures (α-helices (purple), β-sheets (yellow) and coil turns (orange)). The model predicts correctly with high accuracy

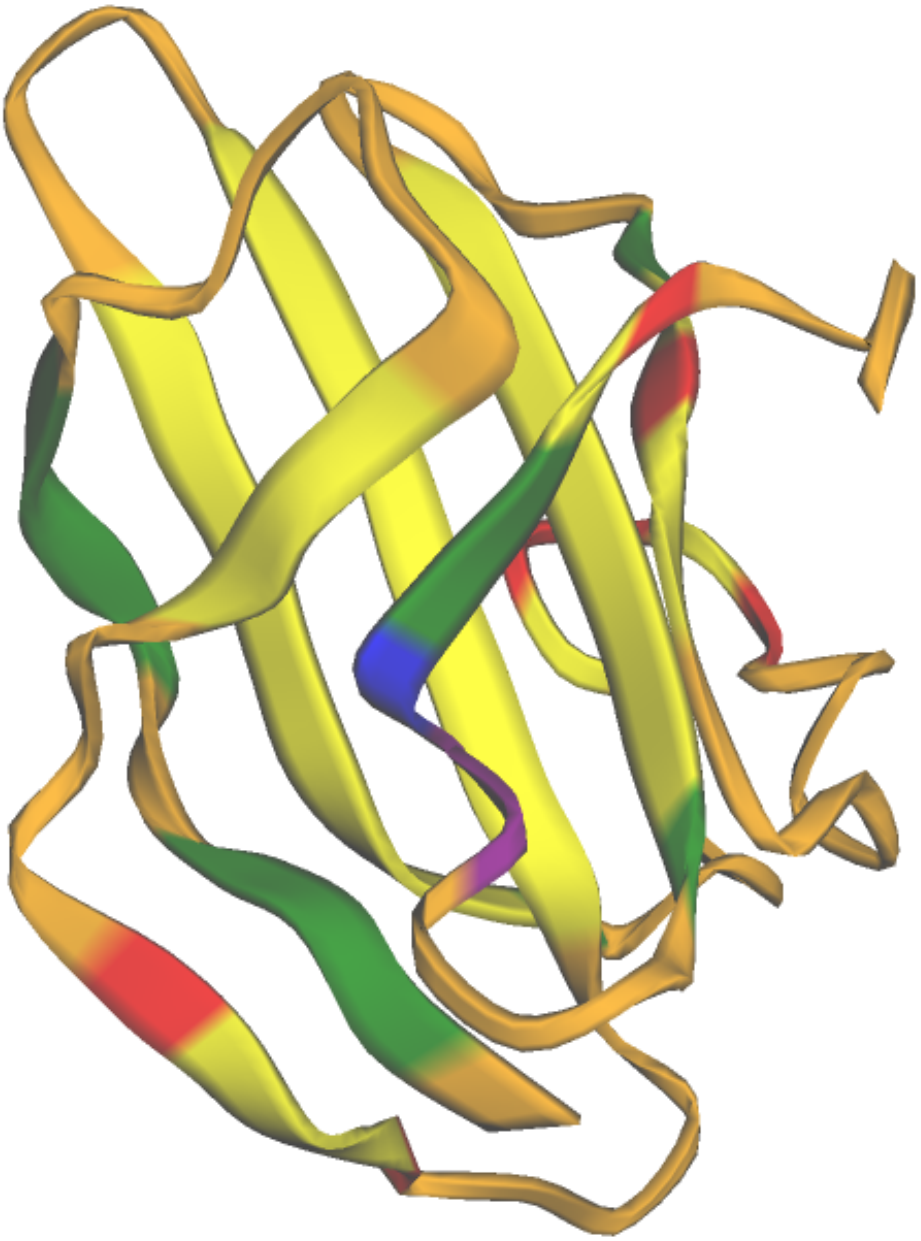

Figure 3.10: 1M9Z:A - The protein chain has bursts of small length β-sheets along its structure. The model is unable to identify these short bursts (green indicates incorrect prediction of β-sheets)

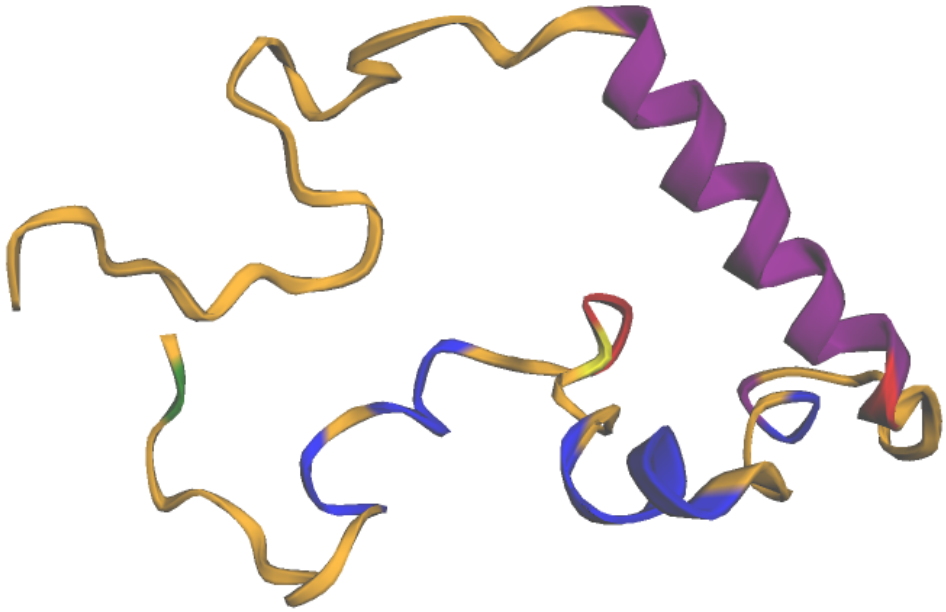

Figure 3.11: 1AOC:A - The protein chain has bursts of small length α-helices along its structure. The model is unable to identify these short bursts (blue indicates incorrect prediction of α-helices)

3.4 Conclusion

In this study, a Transformer network was used to predict the secondary structure. The dataset used was precisely the same as was used in previous studies [130] [30], where it was split into two parts: 4590 proteins were randomly selected to be the training set (TR4590), and the remaining 1199 were used as the independent test set (TS1199). MSAs were generated for all the proteins in the dataset using DeepMSA, and the Position-Specific Scoring Matrix and Information Content was extracted from it. Amino acid embeddings were generated using a Gensim Word2Vec model, which gave representations for each amino acid sharing similar qualities. Positional Encodings were generated and added to the amino acid embeddings. The network was trained on 10-fold cross-validation, and the test set was used to evaluate the model's performance. Even with limited data, the network achieved an accuracy of 82.5%.

Chapter 4

Conclusion

The rapid sequencing of proteins has led to the gap between the number of known protein sequences and the number of known structures to increase rapidly. The tertiary structure provides important clues about its function, but the techniques for structure determination are expensive, labour-intensive and time-consuming, and sometimes not possible. Although the primary goal of protein predictions is to predict the 3-D structure, 1-D and 2-D predictions are of intrinsic interest and often used as inputs for intrinsic structure and function predictors. It becomes essential that predictions based on the protein sequence are made to speed up the structure prediction process and provide clues towards the function of a protein. There has been swift progress happening in Deep Learning. It has been successful on many NLP tasks, including but not limited to image classification, text classification, object detection, and machine translation. Due to the similarity between protein prediction tasks and the listed tasks, applying the latest techniques for protein predictions becomes crucial.

This thesis applied two unique deep learning methods on different tasks, a protein function prediction task and a protein structure prediction task, using only protein sequence information. A deep ResNet was used to predict the binding site of a protein, and a Transformer network was used to predict the secondary structure. There are many prediction tasks, such as torsion angle, contact map, binding affinity, protein-protein interactions for which deep learning models can be applied. When more complicated techniques are developed, they can be applied to these tasks to achieve better results. Nevertheless, with the introduction of DeepMind's AlphaFold and AlphaFold2, there has been tremendous progress in protein structure predictions, where unprecedented levels of accuracy are being seen. Have we finally solved the age-old problem of predicting the structure of the protein from its sequence alone, or do we still have a long way to go?

Related Publications

1. **Vineeth Chelur**, U. Deva Priyakumar. BiRDS - Binding Residue Detection from Protein Sequences using Deep ResNets, Submitted to *Journal of Chemical Information and Modeling*.
   DOI: https://doi.org/10.33774/chemrxiv-2021-013gn

2. Rishal Aggarwal, Akash Gupta, **Vineeth Chelur**, C.V. Jawahar, and U. Deva Priyakumar. DeepPocket: Ligand Binding Site Detection and Segmentation using 3D Convolutional Neural Networks, *Journal of Chemical Information and Modeling*, **2021**.
   DOI: https://doi.org/10.1021/acs.jcim.1c00799

Bibliography

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Jérémy Desaphy, Guillaume Bret, Didier Rognan, and Esther Kellenberger. sc-pdb: a 3d-database of ligandable binding sites—10 years on. *Nucleic acids research*, 43(D1):D399–D404, 2015.

[3] Jayesh Bapu Ahire. The artificial neural networks handbook: Part 1. URL `https://dzone.com/articles/the-artificial-neural-networks-handbook-part-1-1`.

[4] Xuan Hien Le, Hung Ho, Giha Lee, and Sungho Jung. Application of long short-term memory (lstm) neural network for flood forecasting. *Water*, 11:1387, 07 2019. doi: 10.3390/w11071387.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[6] Al Gharakhanian. Generative adversarial networks. URL `https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html`.

[7] Marshall Hargrave. Deep learning. URL `https://www.investopedia.com/terms/d/deep-learning.asp`.

[8] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[9] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990. doi: 10.1109/5.58325.

[13] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455. PMLR, 2009.

[14] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[15] Sumit Saha. A comprehensive guide to convolutional neural networks. URL `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b116`

[16] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

[17] Matthias Heinig and Dmitrij Frishman. Stride: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucleic acids research*, 32(suppl_2):W500–W502, 2004.

[18] Burkhard Rost. Prediction in 1d: secondary structure, membrane helices, and accessibility. *Methods of biochemical analysis*, 44:559–588, 2003.

[19] Liam J McGuffin, Kevin Bryson, and David T Jones. The psipred protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.

[20] Francesco Bettella, Dawid Rasinski, and Ernst Walter Knapp. Protein secondary structure prediction with sparrow. *Journal of chemical information and modeling*, 52(2):545–556, 2012.

[21] Claudio Mirabello and Gianluca Pollastri. Porter, paleale 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics*, 29(16):2056–2058, 2013.

[22] Yuedong Yang, Rhys Heffernan, Kuldip Paliwal, James Lyons, Abdollah Dehzangi, Alok Sharma, Jihua Wang, Abdul Sattar, and Yaoqi Zhou. Spider2: A package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks. In *Prediction of protein secondary structure*, pages 55–63. Springer, 2017.

[23] Mirko Torrisi, Manaz Kaleel, and Gianluca Pollastri. Porter 5: fast, state-of-the-art ab initio prediction of protein secondary structure in 3 and 8 classes. *bioRxiv*, page 289033, 2018.

[24] Chao Fang, Yi Shang, and Dong Xu. Mufold-ss: New deep inception-inside-inception networks for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 86 (5):592–598, 2018.

[25] Hong Seok Kang, Natalya A Kurochkina, and B Lee. Estimation and use of protein backbone angle probabilities. *Journal of molecular biology*, 229(2):448–460, 1993.

[26] Eshel Faraggi, Tuo Zhang, Yuedong Yang, Lukasz Kurgan, and Yaoqi Zhou. Spine x: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of computational chemistry*, 33(3): 259–267, 2012.

[27] Sitao Wu and Yang Zhang. Anglor: a composite machine-learning algorithm for protein backbone torsion angle prediction. *PloS one*, 3(10):e3400, 2008.

[28] Jiangning Song, Hao Tan, Mingjun Wang, Geoffrey I Webb, and Tatsuya Akutsu. Tangle: two-level support vector regression approach for protein backbone torsion angle prediction from primary sequences. *PloS one*, 7(2):e30361, 2012.

[29] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[30] Rhys Heffernan, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou. Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics*, 33(18):2842–2849, 2017.

[31] Yujuan Gao, Sheng Wang, Minghua Deng, and Jinbo Xu. Raptorx-angle: real-value prediction of protein backbone dihedral angles through a hybrid method of clustering and deep learning. *BMC bioinformatics*, 19(4):73–84, 2018.

[32] Jack Hanson, Kuldip Paliwal, Thomas Litfin, Yuedong Yang, and Yaoqi Zhou. Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks. *Bioinformatics*, 35(14):2403–2410, 2019.

[33] Piero Fariselli, Osvaldo Olmea, Alfonso Valencia, and Rita Casadio. Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. *Proteins: Structure, Function, and Bioinformatics*, 45(S5):157–162, 2001.

[34] Alessandro Vullo, Ian Walsh, and Gianluca Pollastri. A two-stage approach for improved prediction of residue contact maps. *BMC bioinformatics*, 7(1):1–12, 2006.

[35] Jianlin Cheng and Pierre Baldi. Improved residue contact prediction using support vector machines and a large feature set. *BMC bioinformatics*, 8(1):1–9, 2007.

[36] Sitao Wu and Yang Zhang. A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. *Bioinformatics*, 24(7):924–931, 02 2008. ISSN 1367-4803.

[37] Marcin J Skwark, Daniele Raimondi, Mirco Michel, and Arne Elofsson. Improved contact predictions using the recognition of protein like contact patterns. *PLoS computational biology*, 10 (11):e1003889, 2014.

[38] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology*, 13(1):e1005324, 2017.

[39] Jack Hanson, Kuldip Paliwal, Thomas Litfin, Yuedong Yang, and Yaoqi Zhou. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics*, 34(23):4039–4045, 2018.

[40] Yang Li, Chengxin Zhang, Eric W Bell, Wei Zheng, Xiaogen Zhou, Dong-Jun Yu, and Yang Zhang. Deducing high-accuracy protein contact-maps from a triplet of coevolutionary matrices through deep residual convolutional networks. *PLoS computational biology*, 17(3):e1008865, 2021.

[41] Andriy Kryshtafovych, Bohdan Monastyrskyy, and Krzysztof Fidelis. Casp 11 statistics and the prediction center evaluation system. *Proteins: Structure, Function, and Bioinformatics*, 84: 15–19, 2016.

[42] Casp. URL `https://en.wikipedia.org/wiki/CASP`.

[43] Mohammed AlQuraishi. Machine learning in protein structure prediction. *Current Opinion in Chemical Biology*, 65:1–8, 2021.

[44] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, and Alex Bridgland. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[45] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, pages 1–11, 2021.

[46] Morten Källberg, Gohar Margaryan, Sheng Wang, Jianzhu Ma, and Jinbo Xu. Raptorx server: a resource for template-based protein structure modeling. In *Protein structure prediction*, pages 17–27. Springer, 2014.

[47] David E Kim, Dylan Chivian, and David Baker. Protein structure prediction and analysis using the robetta server. *Nucleic acids research*, 32(suppl_2):W526–W531, 2004.

[48] Lim Heo and Michael Feig. High-accuracy protein structures by combining machine-learning with physics-based refinement. *Proteins: Structure, Function, and Bioinformatics*, 88(5):637–642, 2020.

[49] Ambrish Roy, Alper Kucukural, and Yang Zhang. I-tasser: a unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4):725–738, 2010.

[50] Jie Hou, Tianqi Wu, Zhiye Guo, Farhan Quadir, and Jianlin Cheng. The multicom protein structure prediction server empowered by deep learning and contact distance prediction. In *Protein Structure Prediction*, pages 13–26. Springer, 2020.

[51] Dong Xu and Yang Zhang. Toward optimal fragment generations for ab initio protein structure assembly. *Proteins: Structure, Function, and Bioinformatics*, 81(2):229–239, 2013.

[52] Yifeng Cui, Qiwen Dong, Daocheng Hong, and Xikun Wang. Predicting protein-ligand binding residues with deep convolutional neural networks. *BMC bioinformatics*, 20(1):93, 2019.

[53] Rishal Aggarwal, Akash Gupta, Vineeth Chelur, CV Jawahar, and U Deva Priyakumar. Deeppocket: Ligand binding site detection and segmentation using 3d convolutional neural networks. 2021.

[54] Tapio Pahikkala, Antti Airola, Sami Pietilä, Sushil Shakyawar, Agnieszka Szwajda, Jing Tang, and Tero Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, 16(2):325–337, 2015.

[55] Tong He, Marten Heidemeyer, Fuqiang Ban, Artem Cherkasov, and Martin Ester. Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of cheminformatics*, 9(1):1–14, 2017.

[56] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

[57] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28 (1):31–36, 1988.

[58] Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. Widedta: prediction of drug-target binding affinity. *arXiv preprint arXiv:1902.04166*, 2019.

[59] Qingyuan Feng, Evgenia Dueva, Artem Cherkasov, and Martin Ester. Padme: A deep learning-based framework for drug-target interaction prediction. *arXiv preprint arXiv:1807.09741*, 2018.

[60] Mostafa Karimi, Di Wu, Zhangyang Wang, and Yang Shen. Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35(18):3329–3338, 2019.

[61] Uniprot: The universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2021.

[62] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28 (1):235–242, 2000.

[63] Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V Crichlow, Cole H Christie, Kenneth Dalenberg, Luigi Di Costanzo, and Jose M Duarte. Rcsb protein data bank: powerful new tools for exploring 3d structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic acids research*, 49(D1):D437–D451, 2021.

[64] Vineeth Chelur and U Deva Priyakumar. Birds-binding residue detection from protein sequences using deep resnets. 2021.

[65] Jingtian Zhao, Yang Cao, and Le Zhang. Exploring the computational methods for protein-ligand binding site prediction. *Computational and structural biotechnology journal*, 18:417–426, 2020.

[66] Marylens Hernandez, Dario Ghersi, and Roberto Sanchez. Sitehound-web: a server for ligand binding site identification in protein structures. *Nucleic acids research*, 37(suppl_2):W413–W416, 2009.

[67] Yang Liu, Maximilian Grimm, Wen-tao Dai, Mu-chun Hou, Zhi-Xiong Xiao, and Yang Cao. Cb-dock: a web server for cavity detection-guided protein–ligand blind docking. *Acta Pharmacologica Sinica*, 41(1):138–144, 2020.

[68] Joe Dundas, Zheng Ouyang, Jeffery Tseng, Andrew Binkowski, Yaron Turpaz, and Jie Liang. Castp: computed atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues. *Nucleic acids research*, 34(suppl_2):W116–W118, 2006.

[69] Manfred Hendlich, Friedrich Rippmann, and Gerhard Barnickel. Ligsite: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, 15(6):359–363, 1997.

[70] Shinji Amari, Masahiro Aizawa, Junwei Zhang, Kaori Fukuzawa, Yuji Mochizuki, Yoshio Iwasawa, Kotoko Nakata, Hiroshi Chuman, and Tatsuya Nakano. Viscana: visualized cluster analysis of protein- ligand interaction based on the ab initio fragment molecular orbital method for virtual ligand screening. *Journal of Chemical Information and modeling*, 46(1):221–230, 2006.

[71] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, 10(1):1–11, 2009.

[72] Xiaolei Zhu, Yi Xiong, and Daisuke Kihara. Large-scale binding ligand prediction by improved patch-based method patch-surfer2. 0. *Bioinformatics*, 31(5):707–713, 2015.

[73] Jianyi Yang, Ambrish Roy, and Yang Zhang. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, 29(20):2588–2595, 2013.

[74] Jianyi Yang, Ambrish Roy, and Yang Zhang. Biolip: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic acids research*, 41(D1):D1096–D1103, 2012.

[75] Fabian Glaser, Tal Pupko, Inbal Paz, Rachel E Bell, Dalit Bechor-Shental, Eric Martz, and Nir Ben-Tal. Consurf: identification of functional regions in proteins by surface-mapping of phylogenetic information. *Bioinformatics*, 19(1):163–164, 2003.

[76] Michal Brylinski and Jeffrey Skolnick. A threading-based method (findsite) for ligand-binding site prediction and functional annotation. *Proceedings of the National Academy of sciences*, 105 (1):129–134, 2008.

[77] Mark N Wass, Lawrence A Kelley, and Michael JE Sternberg. 3dligandsite: predicting ligand-binding sites using similar structures. *Nucleic acids research*, 38(suppl_2):W469–W473, 2010.

[78] Daniel B Roche, Stuart J Tetchner, and Liam J McGuffin. Funfold: an improved automated method for the prediction of ligand binding residues using 3d models of proteins. *BMC bioinformatics*, 12(1):1–20, 2011.

[79] Ambrish Roy and Yang Zhang. Recognizing protein-ligand binding sites by global structural alignment and local geometry refinement. *Structure*, 20(6):987–997, 2012.

[80] Radoslav Krivák and David Hoksza. Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. *Journal of cheminformatics*, 7(1): 1–13, 2015.

[81] Radoslav Krivák and David Hoksza. P2rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, 10(1):39, 2018.

[82] Jian Zhang and Lukasz Kurgan. Scriber: accurate and partner type-specific prediction of protein-binding residues from proteins sequences. *Bioinformatics*, 35(14):i343–i353, 2019.

[83] John A Capra, Roman A Laskowski, Janet M Thornton, Mona Singh, and Thomas A Funkhouser. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3d structure. *PLoS Computational Biology*, 5(12):e1000585, 2009.

[84] Bingding Huang. Metapocket: a meta approach to improve protein ligand binding site prediction. *OMICS A Journal of Integrative Biology*, 13(4):325–330, 2009.

[85] Pedro J Ballester and John BO Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010.

[86] Ke Chen, Marcin J Mizianty, and Lukasz Kurgan. Prediction and analysis of nucleotide-binding residues using sequence and sequence-derived structural descriptors. *Bioinformatics*, 28(3):331–341, 2012.

[87] Jacob D Durrant and J Andrew McCammon. Nnscore: a neural-network-based scoring function for the characterization of protein- ligand complexes. *Journal of chemical information and modeling*, 50(10):1865–1871, 2010.

[88] Jacob D Durrant and J Andrew McCammon. Nnscore 2.0: a neural-network receptor–ligand scoring function. *Journal of chemical information and modeling*, 51(11):2897–2903, 2011.

[89] Peng Chen, Jianhua Z Huang, and Xin Gao. Ligandrfs: random forest ensemble to identify ligand-binding residues from sequence information alone. In *BMC bioinformatics*, volume 15, pages 1–12. BioMed Central, 2014.

[90] Qi Wu, Zhenling Peng, Yang Zhang, and Jianyi Yang. Coach-d: improved protein–ligand binding sites prediction with refined ligand-binding poses through molecular docking. *Nucleic acids research*, 46(W1):W438–W442, 2018.

[91] Amauri Duarte da Silva, Gabriela Bitencourt-Ferreira, and Walter Filgueira de Azevedo Jr. Taba: A tool to analyze the binding affinity. *Journal of computational chemistry*, 41(1):69–73, 2020.

[92] José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

[93] Yang Li, Chengxin Zhang, Eric W Bell, Dong-Jun Yu, and Yang Zhang. Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in casp13. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1082–1091, 2019.

[94] Anirudh Tiwari and Nita Parekh. Network-based machine learning approach for structural domain identification in proteins. *bioRxiv*, 2020.

[95] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S Rose, and Gianni De Fabritiis. Deepsite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017.

[96] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Improving detection of protein-ligand binding sites with 3d segmentation. *Scientific reports*, 10(1):1–9, 2020.

[97] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100. URL https://doi.org/10.1093/nar/gkaa1100.

[98] Chengxin Zhang, Wei Zheng, SM Mortuza, Yang Li, and Yang Zhang. Deepmsa: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics*, 36(7):2105–2112, 2020.

[99] Helen Berman, Kim Henrick, and Haruki Nakamura. Announcing the worldwide protein data bank. *Nature Structural & Molecular Biology*, 10(12):980–980, 2003.

[100] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3): 443–453, 1970.

[101] Zhang lab nw-align. URL http://zhanglab.ccmb.med.umich.edu/NW-align.

[102] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. Blast+: architecture and applications. *BMC bioinformatics*, 10 (1):1–9, 2009.

[103] Vincent Miele, Simon Penel, and Laurent Duret. Ultra-fast sequence clustering from similarity networks with silix. *BMC bioinformatics*, 12(1):1–9, 2011.

[104] Dexter C Kozen. Union-find. In *The Design and Analysis of Algorithms*, pages 48–51. Springer, 1992.

[105] Franck Da Silva, Jeremy Desaphy, and Didier Rognan. Ichem: A versatile toolkit for detecting, comparing, and predicting protein–ligand interactions. *ChemMedChem*, 13(6):507–510, 2018.

[106] Milot Mirdita, Lars von den Driesch, Clovis Galiez, Maria J Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2017.

[107] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2012.

[108] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[109] L Steven Johnson, Sean R Eddy, and Elon Portugaly. Hidden markov model speed heuristic and iterative hmm search procedure. *BMC bioinformatics*, 11(1):431, 2010.

[110] David T Jones, Daniel WA Buchan, Domenico Cozzetto, and Massimiliano Pontil. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

[111] Simon C Potter, Aurélien Luciani, Sean R Eddy, Youngmi Park, Rodrigo Lopez, and Robert D Finn. Hmmer web server: 2018 update. *Nucleic acids research*, 46(W1):W200–W204, 2018.

[112] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202, 1999.

[113] David T Jones, Tanya Singh, Tomasz Kosciolek, and Stuart Tetchner. Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*, 31(7):999–1006, 2015.

[114] WA Falcon. Pytorch lightning. *GitHub*, 3, 2019. URL `https://github.com/PyTorchLightning/pytorch-lightning`.

[115] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[116] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.

[117] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[118] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[119] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32:9689, 2019.

[120] Nicholas Rego and David Koes. 3dmol. js: molecular visualization with webgl. *Bioinformatics*, 31(8):1322–1324, 2015.

[121] Richard B Tunnicliffe, William K Hu, Michele Y Wu, Colin Levy, A Paul Mould, Edward A McKenzie, Rozanne M Sandri-Goldin, and Alexander P Golovanov. Molecular mechanism of sr protein kinase 1 inhibition by the herpes virus protein icp27. *Mbio*, 10(5):e02551–19, 2019.

[122] Yixin Cen, Warispreet Singh, Mamatjan Arkin, Thomas S Moody, Meilan Huang, Jiahai Zhou, Qi Wu, and Manfred T Reetz. Artificial cysteine-lipases with high activity and altered catalytic mechanism created by laboratory evolution. *Nature communications*, 10(1):1–10, 2019.

[123] Celso M Teixeira-Duarte, Fátima Fonseca, and João H Morais-Cabral. Activation of a nucleotide-dependent rck domain requires binding of a cation cofactor to a conserved site. *elife*, 8:e50661, 2019.

[124] Frederick Sanger. Chemistry of insulin. *Science*, 129(3359):1340–1344, 1959.

[125] Peter Y Chou and Gerald D Fasman. Prediction of protein conformation. *Biochemistry*, 13(2): 222–245, 1974.

[126] Jean Garnier, David J Osguthorpe, and Barry Robson. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of molecular biology*, 120(1):97–120, 1978.

[127] VI Lim. Algorithms for prediction of α-helical and β-structural regions in globular proteins. *Journal of molecular biology*, 88(4):873–894, 1974.

[128] Krzysztof Kotowski, Tomasz Smolarczyk, Irena Roterman-Konieczna, and Katarzyna Stapor. Proteinunet—an efficient alternative to spider3-single for sequence-based prediction of protein secondary structures. *Journal of Computational Chemistry*, 42(1):50–59, 2021.

[129] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[130] James Lyons, Abdollah Dehzangi, Rhys Heffernan, Alok Sharma, Kuldip Paliwal, Abdul Sattar, Yaoqi Zhou, and Yuedong Yang. Predicting backbone cα angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network. *Journal of computational chemistry*, 35(28):2040–2046, 2014.

[131] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[132] Radim, Petr Sojka, et al. Gensim—statistical semantics in python. *Retrieved from genism. org*, 2011.

[133] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.