Master's Programme in Health Informatics
Spring Semester 2019
Degree thesis, 30 Credits

Feature optimization of contact map predictions based on inter-residue distances and U-Net++ architecture

Author: Aditi Adesh Shenoy

Author: Aditi Adesh Shenoy

Main supervisor: Professor Arne Elofsson, Department of Biochemistry and Biophysics, Stockholm University

Examiner: Professor Uno Fors, Department of Computer and Systems Sciences (DSV), Stockholm University

Affirmation

I hereby affirm that this Master thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text. This work has not been submitted for any other degree or professional qualification except as specified; nor has it been published.

Stockholm, 20^{th} May 2019

Aditi Adesh Shenoy

Feature optimization of contact map predictions based on inter-residue distances and U-Net++ architecture

Abstract

Background: Determination of the three-dimensional structure of proteins has been a scientific challenge for decades. Since there are many hurdles associated with experimental determination of protein structures, there are attempts for making in-silico predictions of protein structures more accurate. Recent advances include prediction of contacts between residues in a protein using convolutional neural networks. Based on the results of the latest global competition for protein structure prediction (called CASP13), inter-residue distances are found to contain key information required for structure prediction.

Aim: The primary objective of this study was to determine if inter-residue distance-based classification could improve predictions of contacts between residues in a protein. The secondary objective was to determine whether contact predictions based on U-Net, a popular architecture for biomedical image segmentation could be improved by using an alternate nested architecture called U-Net++.

Methods: Using the inputs from state-of-the-art contact prediction model PconsC4, the distance between residue pairs in proteins were used to predict contacts as a classification problem as compared to the regression approach used in PconsC4. The output of this study was a distance-based probability distribution as compared to S-score (a single quantity of distance measure). Additionally, the U-Net architecture was replaced and tested with U-Net++ architecture. The results from all the models deployed during this study were evaluated using performance metrics.

Results: Inter-residue contact distance predictions were calculated for three bin ranges - 7, 12 and 26. These showed low precision values between 0.07 and 0.3 while the absolute and relative error did not shown much variation among the models. The U-Net++ implementation showed improved precision from 0.59 to 0.67 for all residues for top L contacts. Despite recall slightly falling from 0.35 to 0.30 and no visual difference between the contact maps, there is more accurate determination of correctly predicted distances against the actual distances while testing 210 proteins.

Conclusions: Contrary to CASP13 results which showed improved contact predictions by using inter-distance, implementing distance-based classification on PconsC4 did not improve the predictions of contacts. However, U-Net++ architecture showed improved precision values than U-Net architecture for PconsC4 model of contact predictions.

Key Words: Protein Folding; Convolutional Neural Networks; Contact Map; Protein Structure Prediction; Ab-initio protein folding

Acknowledgment

I would like to extend my deepest gratitude to Arne Elofsson for his guidance and supervision during this project. I am grateful to him for giving me this unique opportunity to step into and explore the field of proteins bioinformatics and deep learning. I would like to thank him for his understanding and constant support throughout my project.

I would also like to thank Uno Fors for his kind support and understanding during my project. I am grateful for him reviewing my project and making it possible to submit my thesis. I would specially like to thank Maria Olsson for being there for me over the past couple of stressful months. I am very grateful for her positivity and heartfelt support till the very end. I would also like to thank Sabine Koch, Maria Hägglund, Nadia Davoody and Stefan Möller for all their help during my programme.

I am very grateful to David Menendez Hurtado, Erik Sjölund, John Lamb and Claudio Bassot for taking out the time to answer my questions and helping me when I was stuck during my project. I am grateful to Sudha Govindarajan, Marco Salvatore, Patrick Bryant, Gabriele Pozzati, Katarina Elez, Wensi Zhu and Saman Ashtiani for their valuable suggestions and support.

I would like to give a special thanks to Mukund Kabbe for his constant encouragement, support and valuable inputs.

I am grateful to my entire family for all their love and support. This is dedicated to my dear uncle who will always be remembered.

And mostly importantly, I am extremely grateful to my wonderful parents. This project would not have been possible without their unconditional love, support and guidance. Their valuable suggestions were crucial for this project. I owe my everything to them.

List of abbreviations

PDB	Protein Data Bank
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
NMR	Nuclear Magnetic Resonance
DM	Distance Matrix / Distance Map
CM	Contact Map
CASP	Critical Assessment of Protein Structure Prediction
MSA	Multiple Sequence Alignment
DCA	Direct Coupling Analysis
MI	Mutual Information
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
RGB	Red, Green, Blue
FCN	Fully Convolution Network
PPV	Positive Predicted Value
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
AE	Absolute Error
RE	Relative Error
ReLU	Rectified Linear Units
ELU	Exponential Linear Units

List of Figures

List of Tables

Contents

1

Introduction

Prediction of the three-dimensional structure of a protein from its amino acid sequence has been an unsolved problem for decades. A major hurdle in solving this problem was the vast amount of computational resources required to parse through all possible conformations of a protein. The number of these possible conformations can be limited if we predict which residues in a protein are in contact. Direct Coupling Analysis (DCA) has been a major breakthrough for protein contact predictions [1]. Recent advances which use deep learning for contact predictions [2][3][4] have shown significantly better results in solving the protein folding problem. Inter-residue contact prediction have been shown to be key elements to predict protein structures [5][6][7]. This thesis study explores ways to optimize protein contact predictions using inter-residue distances and by employing enhanced architectures for deep learning.

1.1 Central Concepts

Proteins are complex, large molecules which play an important role in most biological functions. These bio-molecules are the essential, working elements of a cell and are responsible for the regulation and sustenance of cellular activities. Proteins are macro-molecules made of a chain of small molecules (or monomers) called amino acids. There are 20 amino acids which are the building blocks of a protein and join together in varying linear sequences to form a variety of different protein chains. These protein chains fold into unique stable shapes depending on the interactions between residues in the amino acid sequences [8]. An overview on different levels of protein structures is shown in Figure 1.1[1]

The structural orientation of a protein provides key information regarding the function of the protein. When a protein folds correctly into its 3D form, it has the capacity to function optimally. Contrarily, if a protein is incorrectly folded (or misfolded), the functions of the protein can get compromised [8]. Diseases affecting respiratory function in humans, such as cystic fibrosis or diseases leading to neurological degeneration, such as Alzheimer's, Parkinson's and Huntington's disease, are believed to be caused by misfolded proteins [5]. In order to develop effective

[1]Protein Data Bank (PDB) is a repository of experimentally determined structures of large bio-molecules like proteins.

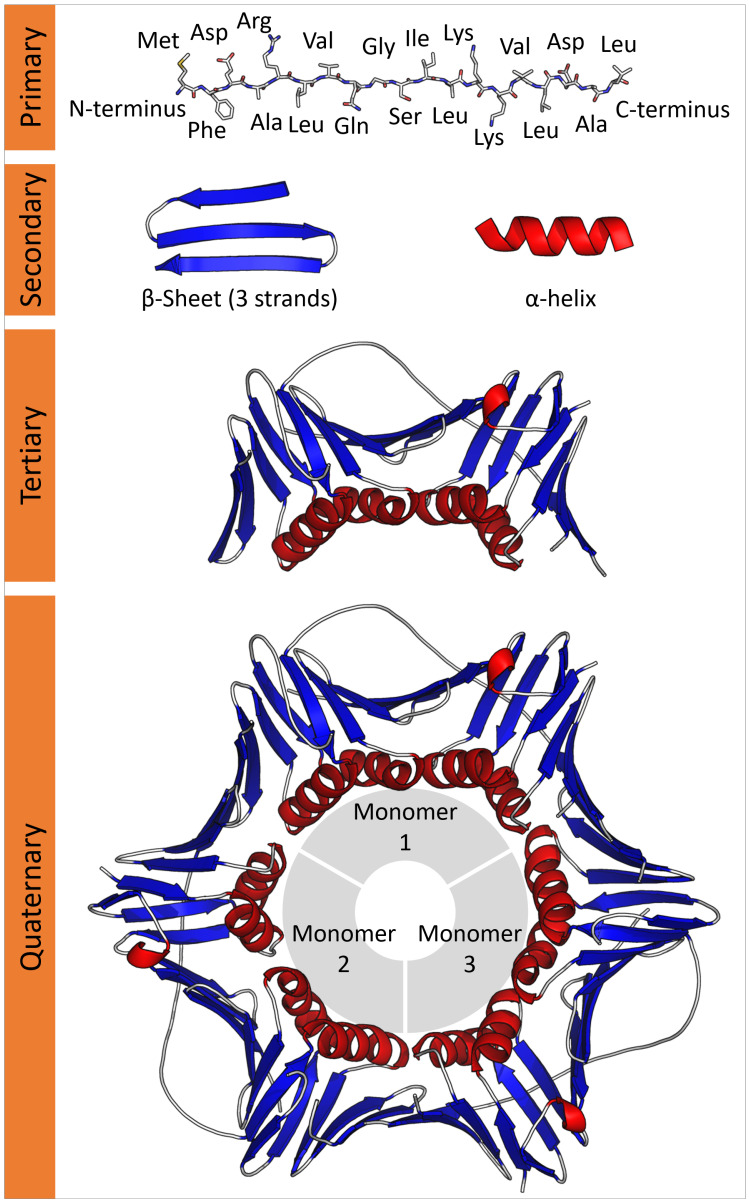


Figure 1.1: Protein structure levels shown by using an example of an antigen protein (PDB: 1AXC): (a) The linear sequence of amino acids forms the **primary** protein structure (b) The local segments of the proteins, which are represented in three dimensions, form the **secondary** structure of the protein (alpha helices and beta sheets) (c) The entire protein represented in three dimensions forms the **tertiary** structure of the protein (d) A **quaternary** structure of a protein is formed when multiple tertiary protein structures join together. Figure created by Thomas Shafee and licensed under CC BY 4.0 license.

cures for these diseases, it is imperative to understand the biological function of the associated proteins. Since the protein's function is closely associated with the protein's structure, how the primary amino acid sequence of the protein folds into a 3D structure needs to be understood. Once the structure of the correctly folded protein is known, we can determine the mutation leading to misfolding of the protein. This will improve our understanding of the disease state and allow for accelerated drug discovery and design for the diseases.

1.1.1 Protein Structure Prediction

Determination of the 3D structure of the protein from its linear primary amino acid sequence is referred to as 'protein structure prediction'. The 3D structure of proteins can be experimentally determined by X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, cryo-electron microscopy or two-dimensional infrared spectroscopy. However, these experimental methods have proved successful only for certain types of proteins. For example, structure prediction for membrane proteins using X-ray crystallography still remains quite challenging [9]. The gap between the number of known protein sequences and their known protein structures is increasing and the experimental methods are not able to meet the growing needs. Using computational tools for determining the structures of these proteins have therefore become increasingly popular to meet the demands of this increasing gap. Algorithms could be designed to determine the 3D structure of a protein either using the information from known structures of similar proteins (homologs) (for example, homology modelling, fold recognition also called protein threading) or without using any information from similar proteins with known structures (for example, ab-initio alternatively known as de novo methods). Ab-initio structure prediction is based on trying to find the conformation having the least energy and is solely based on physics principles [10]. During this thesis study, the ab-initio approach has been used, where the tertiary structure of the protein is predicted from the primary amino acid sequence without using any known structures of similar proteins for prediction.

Critical Assessment of Protein Structure Prediction (CASP)

CASP is a global community-wide experiment held biennially to provide researchers an independent and objective assessment of their protein structure prediction methods. This global competition displays what methods are currently being used and highlights what direction future efforts could be focused in. The first CASP competition was held in 1994 and was called CASP1 with CASP2 held two years later and so on. Assessing contact prediction and contact-assisted protein structure prediction in addition to producing a ranking of state-of-the-art methods has been a major component of CASP over the last decade [11]. The most recent competition was CASP13 and it was held in December 2018.

1.1.2 Contact Map Prediction

Each atom in the structure of a protein macro-molecule can be represented in a three-dimensional space by three cartesian coordinates (x, y and z position). So if a protein has n number of atoms, we will have 3n elements to represent the 3D structure of the protein. An alternative way to represent the 3D protein structure is by using contact maps (CM). Contact maps are 2D representations of 3D structures which capture all the essential information needed for protein folding as illustrated in Figure 1.2.

In order to create a contact map, a distance matrix (also known as distance map) (DM) for the protein must be calculated. Each element in the DM is determined by the Euclidean distance between two residues in a protein structure. From the DM, a contact (or a dot) in the contact map can be defined as follows:

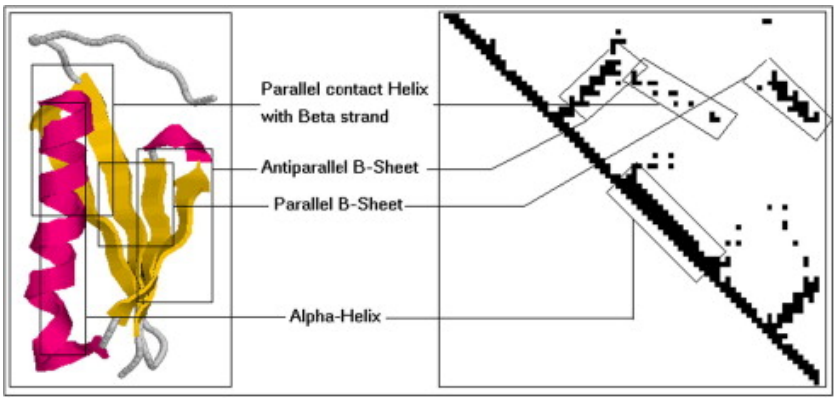


Figure 1.2: Representation of a contact map (a) Left: 3D representation of a protein (b) Right: Contact Map - 2D representation of (a) with threshold set at 8 Å and with residues on the x and y axis [12].

$$CM[i,j] = 1, if DM[i,j] < T$$

$$CM[i,j] = 0, if DM[i,j] >= T$$

where T is the threshold to define whether the two residues are in contact or not, CM[i,j] is an element in the contact map (matrix of contacts for the residue pairs) and DM[i,j] is an element of the distance matrix [13].

Therefore if a protein has n number of atoms, n^2 elements will be needed to represent a 3D structure of the protein. However, since the distance between i and j residues is equal to the distance between j and i residues, the accurate number of elements that is needed to be represented for the 3D structure is $n(n-1)/2$ [13]. Since CM are more compact and their binary nature (contact or non-contact) make it a classical classification task, CM predictions are becoming increasingly popular for protein structure prediction, especially since CASP 11 in 2014 [6].

Usually, the inputs for protein contact prediction models include primary amino acid sequence of a protein and features that can be extracted from the protein sequence. These features could include information about (a) a collection of protein sequences, also called Multiple Sequence Alignment (MSA) (e.g. self-information, partial entropy) (b) positions in a single sequence (e.g. sequence profile) (c) a pair of residues in the sequence (e.g. Direct Coupling Analysis (DCA), Mutual Information (MI)).

Multiple Sequence Alignment (MSA)

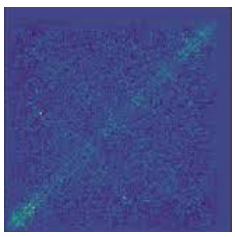
A MSA is an alignment of three or more biological sequences stacked horizontally one on top of each other. This is done to visualize which parts of the sequence are common to each other and understand shared linkage (evolutionary origins) between homologous sequences. A MSA output can give information about similar protein sequences and therefore, similar protein structures. For similar protein structures, a MSA can contain protein sequences with minor amino acid substitutions. To compensate for these substitutions while maintaining the overall structure of the

protein, the sequence may have an amino acid variation somewhere else in the protein sequence. Information about these amino acid substitutions and compensatory substitutions is important for contact map predictions. For example, if there is a mutation in a single protein sequence, there could be another mutation in a different region of that protein sequence to retain the overall protein structure. Therefore, a large and high-quality MSA is crucial for obtaining good contact predictions [11][14].

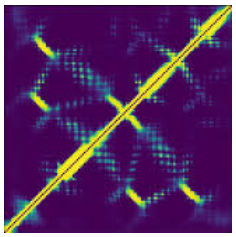
Direct Coupling Analysis (DCA)

Correlated substitution patterns in protein families are extensively used to extract information about protein structures and are being utilized to determine residue-residue contacts [15]. To elucidate the connection between substitution patterns and residue contacts, let us take an example of a protein sequence where residue i and residue j are in contact. If residue i destabilizes and is substituted by residue i^*, then in order to conserve the contact between the two residues i and j, residue j would be expected to substitute to residue j^* to compensate for residue i's initial substitution. On using co-variance analysis to analyze the interactions between the correlated residues in the proteins, it was found that in addition to partially predicting correct contacts within the protein, pairs of residues which were far apart were also predicted as a contact [15]. To overcome the limitation of predicting secondary correlations between non-interacting residues, DCA focused on eliminating indirect relationships which show high correlation [1][15]. Therefore, the DCA method statistically quantified the direct relationship between two positions in a sequence by excluding the influence of indirect relationships [1]. The use of DCA has been a breakthrough for protein contact predictions and the protein structure predictions have significantly improved [11].

In addition to utilizing high-quality MSA and correlated mutations in protein sequences, recent advances in contact predictions employ deep learning techniques [11][14]. In Figure 1.3, the distinct separation of contacts for the model using deep learning along with GaussDCA (a modification of DCA) as compared to the model using only GaussDCA can be observed.



(a) GaussDCA



(b) Deep learning and GaussDCA

Figure 1.3: Contact Maps generated using DCA and Deep Learning

1.2 Health Informatics Context

The future of health care is personalized medicine. The term personalized (or precision medicine) is used to describe an individualized treatment for patients based on multiple characteristics i.e. clinical symptoms, severity of the symptoms, genetic information, lifestyle, patient history and so on. It is often also referred to as P4 Medicine where health care is predictive, personalized, preventive and participatory, and embraces the use of technologies to customize and deliver quality care [16].

The field of health informatics aims to utilize informatics and information technology to deliver quality health care services efficiently. It also involves effectively using bio-medical data to increase knowledge for scientific inquiry, solving medical problems and decision making with the intention of improving health of the public [16]. Therefore, with the revolutionizing sector of health care, it becomes imperative to transfigure the scope of health informatics to encompass the rising demands for quality health services.

Health informatics can facilitate personalized medicine by enabling translational research i.e. integration of scientific discoveries in clinical practice [16][17]. This thesis study deals with using computational and data science techniques to maximally extract information from biological data and predict contact maps which are used for protein structure prediction. This project contributes to different facets of health informatics as shown below:

Firstly, this study could provide an understanding on protein structures and therefore protein function as described in Section 1.1. When we know the structure of a correctly folded protein and its function, we will have improved understanding about misfolded proteins and the diseases they cause. In turn, our understanding of the manifestation and treatment of diseases at a molecular level would increase. Therefore, this research field could produce enhanced therapeutics and diagnostics for provision of better health care.

Secondly, the deep learning methods that are used in this thesis study contribute to the field of bio-image informatics [18]. Bio-image informatics is a sub-field of health informatics and it includes advances from information technology like image processing, pattern recognition and computer vision applied for biological images [17][18]. Image segmentation of histo-pathological images and medical image processing for tumour or glaucoma detection are popular applications of computer vision on bio-medical image sets [17]. The principles used for bio-medical image processing is common, so the methodology of this research study can be adopted for other bio-images as well.

Therefore, in addition to providing more information about disease states, this study would significantly contribute to the field of bio-image informatics. Hence, this study is an enabling resource for accelerating integration of scientific discoveries in clinical practice.

1.2.1 Deep Learning

In order to improve contact map predictions, this thesis study has used deep learning to learn complex abstract features for prediction of visual patterns in the secondary structure of proteins. Since predicting contacts is a structured learning problem, using deep learning would be beneficial since this approach can learn feature hi-

erarchy to define high-level concepts [14]. The following sections will describe the fundamental concepts of deep learning that have been used in this project. This involves understanding the basic functional unit of deep learning - artificial neuron, how artificial neurons connect with each other to form multilayer feed-forward neural networks and how deep learning can be applied to the context of images. Thereafter, an explanation on convolutional neural networks (CNNs) and components making up CNNs will be described. Lastly, architectures that are commonly used for bio-medical image processing (U-Net and U-Net++) (which have also been used in this project) will be described.

Artificial neuron

The most fundamental unit of a neural network is an artificial neuron which is inspired by the function of a biological neuron. When a signal in a biological neuron exceeds a threshold value, the neuron emits a chemical signal which is passed on to the connected neuron. Similarly, as illustrated in Figure 1.4, when the value of the input signals multiplied by the corresponding weights is higher than a defined threshold, it is passed through an activation function which determines the value of the output of the artificial neuron.

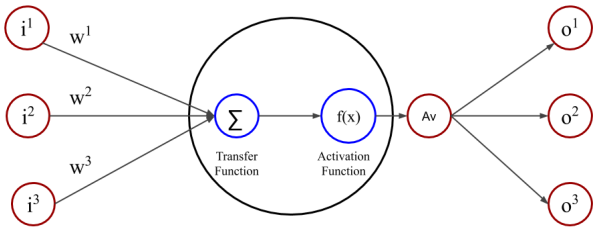


Figure 1.4: Illustration of a single artificial neuron: i^1, i^2, i^3 are the input connections and w^1, w^2, w^3 are the corresponding weights for each of the input connections. A_v is the value after passing the summation of the input connections through the activation function. o^1, o^2, o^3 are the output connections.

Multilayer Feed-Forward Networks

In a feed-forward neural network, much similar to biological neural networks, the signals are passed along the network to subsequent neurons in a single direction. When multiple artificial neurons are grouped together, they form a layer. A typical mutilayer feed-forward neural network architecture consists of three main components as illustrated in Figure 1.5:

1. Input layer: Through this single input layer, the network is fed the input values (x^1, x^2, x^3). The number of neurons in this layer is determined by the number of input features to be fed to the network [19].

2. Hidden layers: There could be one or more layers in between the input and output layers of a network. The weights associated with these intermediate connections determines how the network extracts information and learns from the input values. In fully connected networks, every neuron in a single layer is individually connected to all the neurons in the preceding and succeeding layers.

3. Output layers: Through this layer, the prediction of the network can be obtained. Depending on which activation function (e.g. sigmoid, softmax) is used, the output value (y^1, y^2, y^3) maybe a real number (for regression problems) or an array of probabilities for each class label (for classification problems) [19].

In order to train the neural network, the learning process occurs by iterative re-adjustment of the weights associated with each input feature. Some input features are more strongly correlated with the output class labels as compared to others. During each training iteration (epoch), the network produces an output signal. The difference between the output value and the true value is the error. An attempt to reduce the error value (or loss) is made using loss functions (e.g. Mean Squared Error, Cross Entropy) where correct predictions are rewarded and incorrect predictions are penalized. This type of iterative learning for optimization by re-adjusting the weights to reduce the loss value is called back-propagation learning [19].

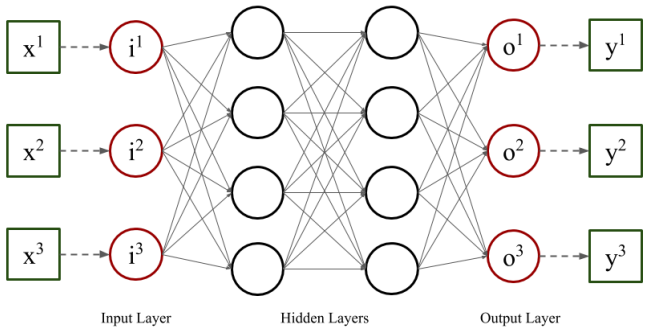


Figure 1.5: Illustration of a multilayer feed-forward neural network: x^1, x^2, x^3 are the input features, i^1, i^2, i^3 are the input neurons, o^1, o^2, o^3 are the output neurons, y^1, y^2, y^3 are the output values (real number or probabilities)

Deep Learning for Image Classification

Multilayer feed-forward neural networks take a one-dimensional vector as an input and produce an output value after transforming the data vector through one or more hidden layers. If we provide image data as the input, the number of weights per neuron would be significantly high and the fully-connected network may lead to over-fitting [20]. For example, let the dimensions of an input image be 32 pixels (width) * 32 pixels (height) * 3 colour channels (depth). So the number of weights for a fully-connected neuron in the hidden layer would be 32*32*3 = 3072. Images with more pixels and networks with more hidden layers would result in more weights per neuron, so the multilayer feed-forward architecture does not scale well for images. In order to extract maximum information from images and prevent over-fitting, the standard neural network architecture was modified to consider the inputs as a 3D volume of neurons instead of a 1D array of feature values. Therefore, in Convolutional Neural Networks or ConvNet (CNN), the (image width), (image height) and (RGB channels of the image) are arranged in a neuron as a 3D volume with (width), (height) and (depth).

Convolutional Neural Networks (CNN)

CNN are capable of learning complex high-order features through a sequence of layers using convolutions [20]. A typical CNN architecture contain a combination of the following feature extraction layers between the input layer and the output classification layer:

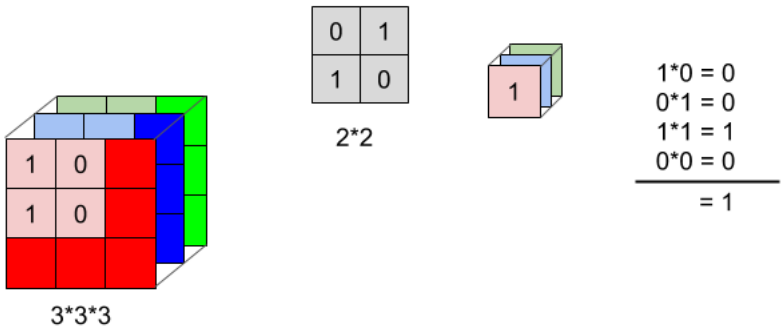
1. **Convolution Layer:** This layer forms the core building block of a CNN which takes a region from the input layer, performs a dot product of the input region and a set of weights (a kernel filter) and produces an output feature map. The 2D kernel filter passes through (convolves) the 3D space of the input layer and produces a dot product with each slice of the 3D volume. The output of this operation is a 2D activation map which contain the relevant features of the image. The convolution operator is the feature detector for the CNN [20]. The kernel filter acts as a filter to allow only certain features to be represented in the output feature map. For example, if the edge kernel is used, only edge information is found in the resulting activation map.

   In Figure 1.6, the 1D 2*2 matrix represents the kernel filter. Figure 1.6a illustrates how the convolution operation works. The 3*3*3 matrix is the input image data which has 3 pixels as height, 3 pixels as width and 3 colour channels - Red, Blue and Green. If we consider the first section of 2*2 dimensions in the red channel and perform a dot product with the 2*2 filter, a 1*1 dimensional red activation map is obtained. If the same is performed for the blue and green channel, a 1*3 dimensional activation map is obtained. If there are 10 filters in the convolutional layer, the output volume would be of dimension 3*3*10. If we observe the second section of 2*2 dimensions in the red channel as shown in Figure 1.6b, we can see that the section considered for convolution is moving horizontally by one pixel i.e. with stride = 1. The value of the stride can be changed depending on what features the output feature map is expected to contain.

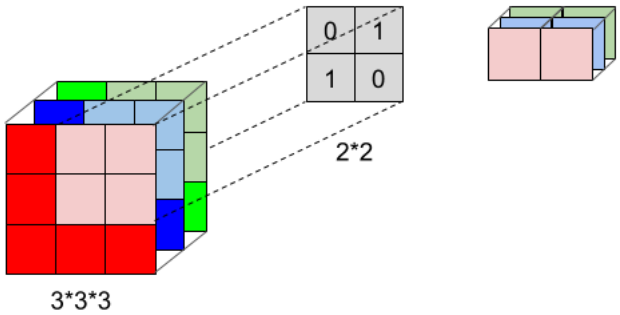
2. **Pooling Layer:** The aim of this layer is to progressively reduce the dimensionality of the data i.e. the width and height of the input image and prevent over-fitting [20]. If an input image of 32 pixels width * 32 pixels height is taken, the pooling layer would result in an image of size 16 pixels width * 16 pixels height.

   If a kernel filter of 2*2 is considered, the maximum value of the four dot products is considered in the output feature map. This operation is called max-pooling.

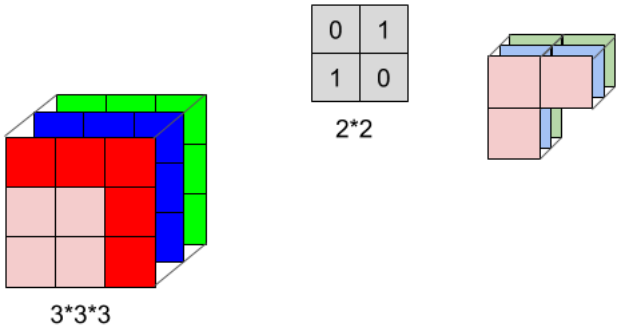
3. **Normalization Layer:** In order to increase the speed of training the input data, this layer normalizes the activations of the previous layer. This layer transforms the mean activations to as close to 0 and activation standard deviation to as close to 1 [20]. Batch-normalization is a normalization method which can be introduced within the network architecture to normalize the inputs at each mini-training step [21]. This is used to increase the speed of the network.



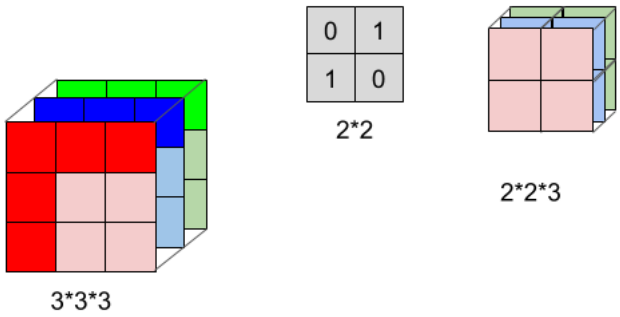
(a) Representation of how one section of a 3*3*3 dimensional input image and the 2*2 dimensional kernel filter performs a dot product to give a 1*1*3 dimensional output



(b) Representation on how the filter overlaps with a certain section of the input image by one-to-one mapping to perform dot product and obtain the corresponding output map



(c) Representation on how convolution on the third slice of the 3D input image obtains the third part of the activation map.



(d) Complete representation on how a 3*3*3 dimensional input image results in a 2*2*3 output feature map after undergoing 12 convolutions

Figure 1.6: Illustration explaining the steps involved in a convolution operation.

4. **Fully-connected layer:** The neurons in this layer are connected to every neuron from the previous layer. This layer is usually used at the end of the network to compute the output of the classes. The dimension of the output of this layer is 1*1*N where N is the number of output class labels [20].

5. **Non-linearity Layer:** Non-linear neurons allow the network to learn complex features from the input data. Neurons can employ non-linearity by passing their inputs through an activation function [22]. The activation function used to determine the output of the contact map in context of this thesis study are:

   - *Sigmoid*: When the value of the input to this function is very small, the output of this logistic neuron would be close to 0. When the value of the input is very small, the output of this function would be close to 1 [22]. All the output values from sigmoid function are sandwiched between 0 and 1 and the values in between take a S-shape crossing through 0.5 as seen in Figure 1.7. This function is defined as follows:

   $$f(x) = \frac{1}{1+e^{-z}}$$

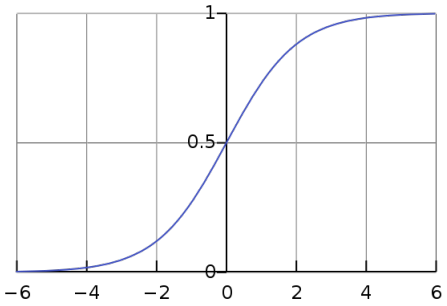
   

   Figure 1.7: Illustration of output of sigmoid function with varying input values. Figure is available on the public domain.

   - *Softmax*: In order to express the output as a probability distribution to for a classification problem, softmax activation can be used. This function gives more information regarding how confident the model is in predicting a certain class. If the model is confident about the predicted class label, the output value of that particular class label will be close to 1 else it will be close to 0. If the model is unsure about which class label the output belongs to, multiple class labels will have a certain probability associated with it. All the probabilities of the classes sums up to 1.
     The softmax function is defined by:

     $$y_i = \frac{e^{-z_i}}{\sum_j e^{-z_j}}$$

   - *Rectified Linear Units (ReLU)*:This activation transforms the input only if it is above a certain threshold. If the input is below 0, the output value would also be 0. If the input is above 0, the output would increase linearly with the input dependent variable. This can be visualized in Figure 1.8. This function is defined as follows:

     $$f(x) = max(0, z)$$

- *Exponential Linear Units (ELU)*: This activation function showed to increase the speed of deep learning networks and to produce more accurate classification predictions [23]. ELU can be observed in Figure 1.8. In ReLU, the value of gradient descent diminishes for positive values but in ELU, the derivative of x with respect to x becomes 1. For $\alpha > 0$, this function is defined as follows when x > 0:

$$f(x) = x$$
$$f'(x) = 1$$

and is defined as follows when x <= 0:

$$f(x) = (\alpha)(exp(x) - 1)$$
$$f'(x) = f(x) + (\alpha)$$

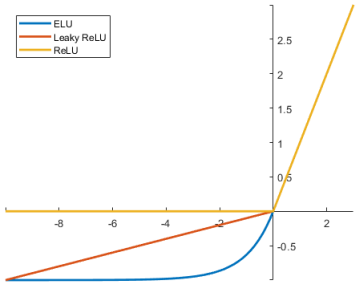


Figure 1.8: Illustration of output of ELU vs ReLU vs Leaky ReLU function with varying input values. Figure is available on the public domain.

CNN architectures for bio-medical image segmentation

Deep learning methods, such as CNNs, have shown great potential to solve medical image processing and analysis problems. The advantages CNN has, over other techniques, are its proprieties for parameter sharing and local-connectivity [24]. By using common kernel filters across different sections of the input volume, the number of parameters are shared within a single network. Additionally, since the neurons in the hidden layers of a CNN are not fully connected to the neurons from the preceding and succeeding layers and are locally connected, the low number of overall connections within the architecture can prevent over-fitting [24].

Fully Convolutional Networks (FCNs) omit the fully connected layers to retain the high spatial locality and resolution in the output feature map [25]. Since this approach can take inputs of varying dimensionality and produce outputs or corresponding sizes, the inessentiality of padding or cropping images, makes this method advantageous for contact map predictions [4].

A skip is a connection between non-sequential layers in a network. So, the features from a layer at a higher level can be transferred to another layer in the network

by skipping layers in between. Architectures with FCN add skips between layers to fuse the coarse and fine details of an image [25].

In bio-medical image processing, semantic segmentation is critical i.e. each pixel needs to be assigned a class label [26]. This type of precise segmentation of tumour lesions or abnormalities in medical images is of prime importance since even the slightest error in segmentation would lead to loss of credibility of the designed system in clinical settings [27]. Additionally, it is challenging to strike a balance between global features (which determine deep, semantic and course details of the image) and local features (which determine low-level and fine-grained details of the image). Another challenge associated with the bio-medical image processing is the lack of sufficient amount of training data. Similarly in contact maps, semantic segmentation and localization are important. The following architectures [26][27] use FCNs and have been implemented for contact map predictions to overcome the aforementioned challenges:

1. **U-Net Architecture**: This architecture has yielded significant improvement in performances for contact map predictions [3]. This fully convolutional network takes into consideration good localization of images for semantic segmentation by retaining the wide context of use. Additionally, this architecture produces precise semantic segmentation with higher resolution of output despite having limited amount of training data [26].

   In the Figure 1.9, the U-Net architecture is pictorially explained with the left side showing the contracting path and the right side showing the expanding path. Each blue rectangle represents one feature map having multiple channels. At the top of each box, the number of channels in the respective feature map is mentioned.

   For the contracting path (left side), a standard sequence of two 3*3 convolutions (each followed by a ReLU) and a 2*2 max-pooling layer is applied for down-sampling (ReLU activation function described in Section 5). At each down-sampling step, the number of feature channels is doubled. For the expanding path (right side), a 2*2 convolution and concatenation with the corresponding contracting feature map (having the same number of channels) takes place and a standard unit of 3*3 convolutions (each followed by ReLU is applied. At each up-sampling step, the number of feature channels is halved. The skip connections from the contracting which are concatenated with expanding path help retain the finer details of the input image.

2. **U-Net++ Architecture**: This architecture is a recent advancement to the U-Net architecture by implementing a series of nested and dense skip connections to the original network [27]. Instead of the plain skip connections implemented in U-Net, the re-designed skip pathways and deep supervision in U-Net++ are expected to bring the feature maps from contracting and expanding paths closer to each other to bridge the semantic gap, thereby making the learning process easier [27]. During this thesis study, only the re-designed skip pathways have been adapted to produce contact map predictions. The difference between the U-Net and U-net++ is clear from Figure 1.10.

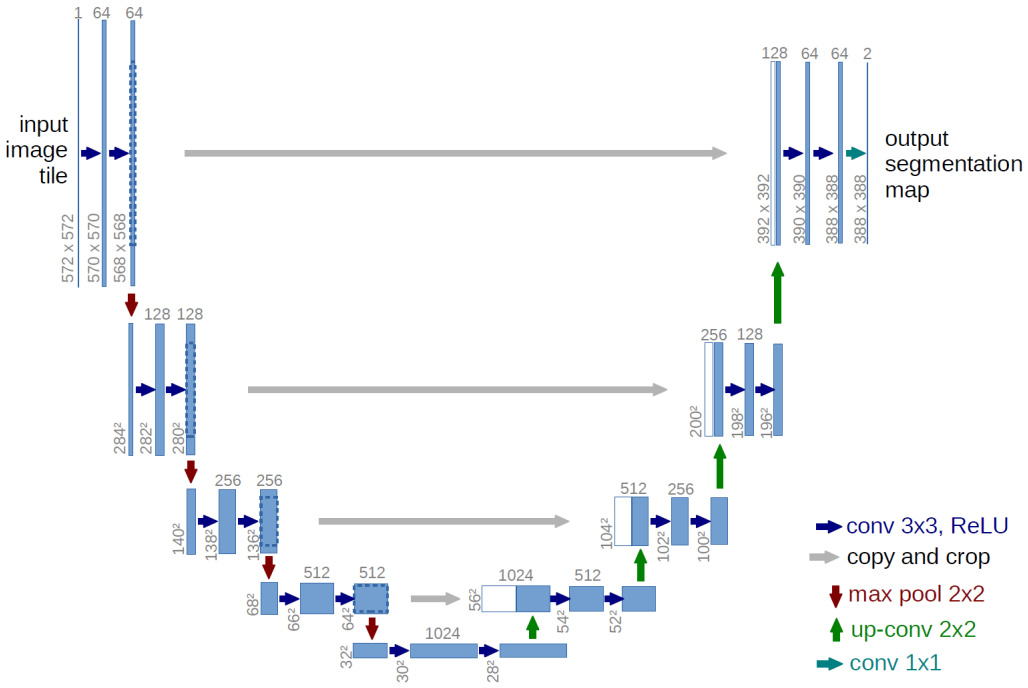


Figure 1.9: Representation of the U-Net architecture [26]. The left side is the contracting path where the number of channels in the feature map increases with each down-sampling step and the right side is the expanding path where the number of channels in the feature map decreases with each up-sampling step

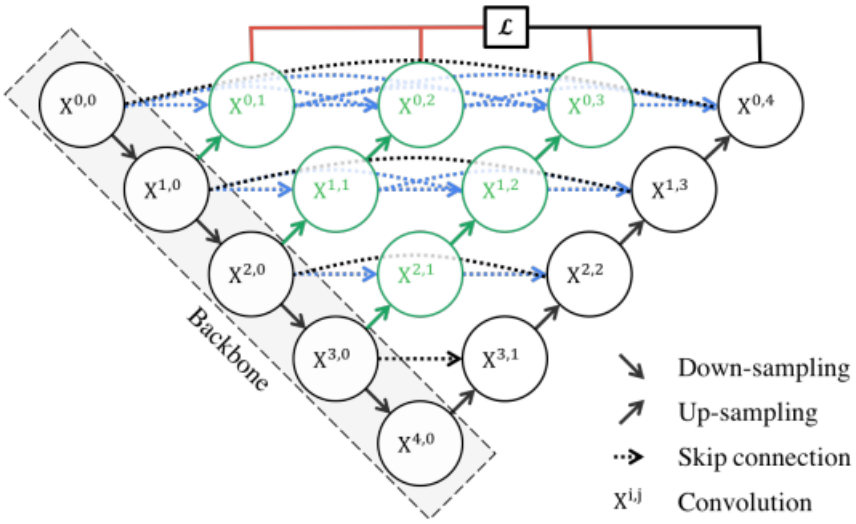


Figure 1.10: Representation of the U-Net++ architecture [27]. The circles in black represent the original U-Net architecture. The circles in green and the blue dotted lines represent dense convolutional blocks on the skip pathways and the red represents deep supervision

Re-designed skip pathways: Instead of the direct concatenation of the feature map in the contracting path with the feature map in the expanding path, a dense convolution block with n convolution layers is added in the skip pathway.

If the value of each neuron in the dense convolution block is $X^{i,j}$ (where i is the index for the down-sampling layer and the j is the index for the convolution layer in the dense block), the value of a convolution layer followed by an activation function is $C_a(z)$ and the value of the up-sampling layer is $U(z)$, the output value of each node in the dense convolution block is:

$$x^{i,j} = C_a(x^{i-1,j})$$

where j = 0 (nodes which receive only one input from the previous layer) and

$$x^{i,j} = C_a[[x^{i,k}]_{k=0}^{j-1}, U(x^{i+1,j-1})]$$

where j >0 (nodes with j >1 will receive j+1 inputs from the previous layer)

From the output of these convolutions in the dense block, the features will accumulate and travel to the top left node in the network. This allows for the feature map from contracting path to be semantically similar to the feature maps from the expanding path.

1.3 Related research

The most recent and significant breakthrough in the field of contact predictions was AlphaFold (by Google's DeepMind). Alphafold was the CASP13 winner and this system revolutionized the use of deep learning for protein structure prediction by predicting distances between every pair of amino acids in a protein. Their networks predicted the distance between residues and the angles between chemical bonds that connect those amino acids. The distance distribution was combined into a score which helped determine the accuracy of the proposed structure. A separate network was additionally trained to calculate the aggregate distance of entire protein to give an estimate on how far the true structure is from structure of the predicted protein [5].

MULTICOM[2] used contact distance prediction using CNNs and deep learning driven model selection for the CASP13 experiment and showed significantly higher results as compared to their CASP12 submission. Hou et al [6] study suggests that distance based predictions are key parameters for protein structure prediction.

ResNet architecture was used by RaptorX-Contact, the CASP12 winner, for producing the most accurate contact prediction [2]. Xu [7] introduced distance matrix prediction and used distance-based classification for protein structure predictions using a similar ResNet architecture.

[2]A state-of-the-art protein structure prediction system

Xu's study supports DeepMind's hypothesis that distance-based protein folding produces better 3D structures of proteins than contact-based protein folding [7].

FCN-based architectures are also popular among prediction systems predicting contacts between residues of proteins. Jones et al [4] used FCNs in DeepCov and showed improved precision for small sequence families with minimal sequence features for training. Mirco et al [3] have shown fast contact predictions using U-net architecture in PconsC4.

PconsC4

State-of-the-art protein contact prediction model PconsC4 provides fast and hassle-free contact prediction by using GaussDCA and U-Net architecture. Figure 1.11 describes the pipeline of PconsC4. The inputs for training this model are 1D features obtained from protein sequences and secondary structures and 2D features like mutual information, normalized mutual information, cross entropy, GaussDCA output. This model, developed by the Elofsson Lab[3] produces contact maps at three different thresholds (6, 8 and 10 Å) and distance output as S-score. S-score is a distance measure defined as:

$$S - score = \frac{1}{1+(d_{ij}/d_0)^2}$$

where d_{ij} is the distance between the amino acid i and amino acid j and d_0 is the threshold distance set as the reference to determine whether the amino acids are in contact or not (e.g. 8Å).

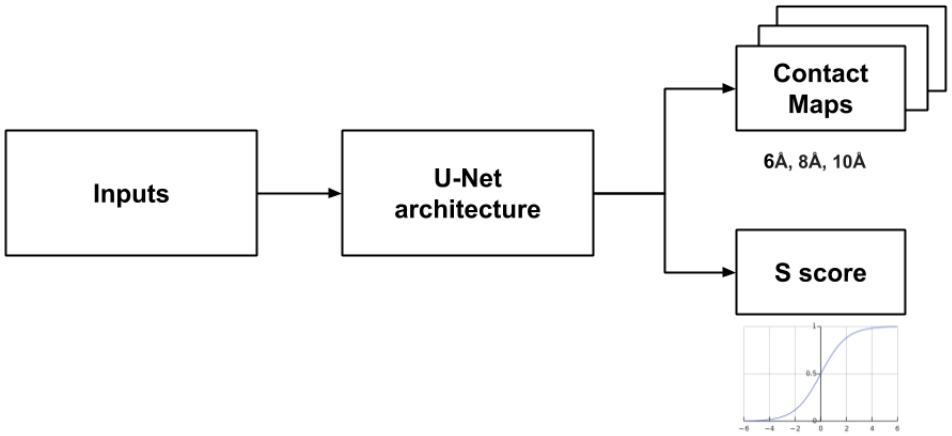


Figure 1.11: Current pipeline for PconsC4 model for contact prediction [3]

[3]This thesis project was hosted at Elofsson Lab at Science For Life Laboratory, Sweden

1.4 Scientific gap

PconsC4 uses multi-dimensional input to provide quick contact predictions by solving for S-score as a regression problem. The recent advances in the area of protein structure prediction [5][7][6] (as highlighted in Section 1.3) suggest that distance-based prediction provide more information than contact map prediction for protein structure prediction. Based on these new insights, it is unknown whether the multi-dimensional inputs of PconsC4 could provide better contact predictions by solving for distance as a classification problem. The major difference between the classification and regression approach is that the former predicts class labels whereas the latter predicts specific quantities.

Therefore, this thesis study attempts to explore how inter-residue contact distance prediction would affect the outputs of contact prediction of the PconsC4 contact prediction model.

Additionally, this thesis project explored ways to optimize the CNN architecture that was used in PconsC4.

1.5 Aims and Objectives

- To investigate the effect of modifying PconsC4 from a regression problem to a classification problem
- To compare predictions from distance-based outputs as compared to S-score-based outputs
- To evaluate whether probability distribution of distance measures can give more information for protein structure determination as compared to S-score predictions
- To study if using U-Net++ architecture instead of U-Net architecture would improve contact predictions

1.6 Research Questions

1. Can inter-residue contact distance prediction (a classification problem) give better results of contact predictions than S-score based contact prediction (a regression problem) in PconsC4 contact prediction model?
2. Can U-Net++ architecture improve contact predictions of PconsC4 as compared to predictions based on U-Net architecture?

2

Methods

2.1 Research Approach

The current thesis project utilized a quantitative study design. This study used analytic and data science tools to prove a hypothesis through numerical experiments and the outcomes were quantified using evaluation metrics. Based on related research (refer Section 1.3) and advancements in protein structure prediction, the theory of using distance-based prediction of contacts is becoming popular. This study adopted a confirmational strategy to test if implementing the above theory allowed for better contact predictions in PconsC4 [28]. While designing this experimental study, one variable (distance-based classification or core architecture) had been independently manipulated while keeping all the other variables constant [28][29]. This was done to look for individual differences in the outcome due to a particular manipulation. Descriptive studies include collection, analysis and interpretation of data [29]. The study was descriptive as it only aimed to answer if distance-based contact prediction produced better results than previous methods for generating contact prediction without delving into how or why one may be better than the other [28]. This thesis study can be classified as a methods development or an improvement project since it provides a enhancement to solutions to solve the existing problem of protein structure prediction [30].

2.2 Data-sets

This thesis study has used a protein dataset of 2891 proteins that had been used to train PconsC4 [3]. These proteins were obtained from PDB[1] and were selected if they had minimum resolution 2 Å and maximum R-factor 0.3. Sequences having more than 20% identity were not included. To make sure that there was no overlap in the data used for training and testing, the sequences having same ECOD H-groups[2] [32] were excluded. From the 2891 proteins used for training, 100 of them were selected randomly to make the validation set and these were strictly not used during the training process. An independent set of 210 proteins were used for testing.

[1]Protein Data Bank (PDB) is a repository of experimentally determined structures of large bio-molecules like proteins [31]

[2]The proteins belonging to the same ECOD H-group have homologous links to each other. These proteins could be similar based on sequence, structural or functional similarity.

The PDB codes for the proteins used for training are available in Table A.1, the PDB codes for validation are available in Table A.2 and the PDB codes that were used for testing can be found in Table A.3 in Appendix A.

2.3 Inputs for training the models

The training inputs for both the research objectives of this study are the same as those of PconsC4 [3]. A representation of the 1D and 2D inputs that were used in this study are shown in Figure 2.1. There are three types of primary (1D) inputs to the model: Protein sequence[3] and MSA; Self Information and Partial Entropy. From the columns in the MSA, the self-information can be calculated as follows:

$$I_i = \log_a \frac{p_i}{pavg_i}$$

And the partial entropy can be calculated as:

$$S_i = p_i * \log_a \frac{p_i}{pavg_i} = p_i * I_i$$

where p_i is the probability of each amino acid (or gap) which can be calculated from the columns in the MSA and $pavg_i$ is the average frequency of an amino acid based on its occurrence in Uniref50 dataset. The total number of states that are considered are 20 amino acids, one gap state, B (asparagine or aspartic acid) and X (for unknown residues). From each alignment, the probability of the gap state is estimated [3].

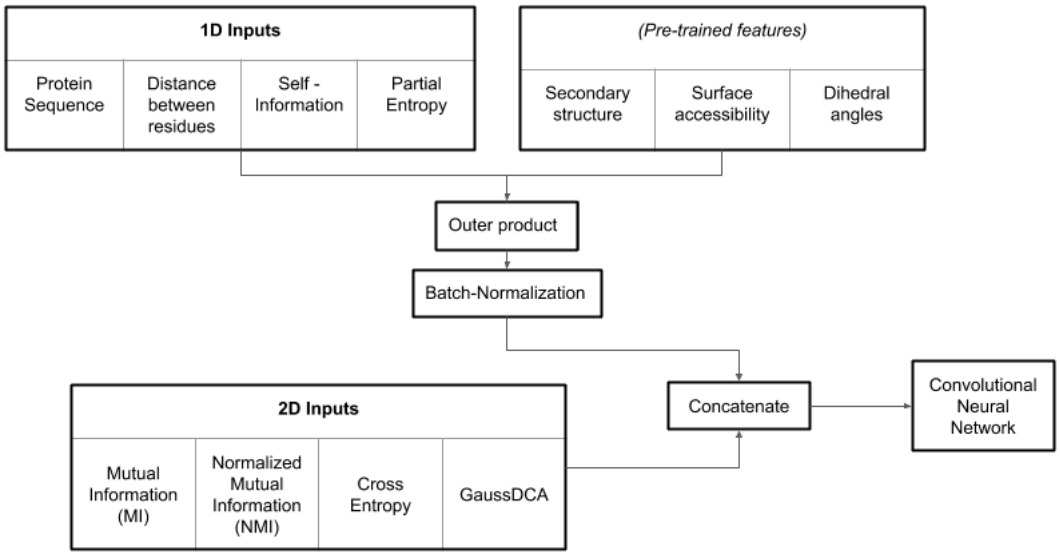


Figure 2.1: Representation of 1D and 2D training inputs used in this study

The 2D inputs comprise of mutual information (MI), normalized mutual information (NMI), cross entropy (H) and GaussDCA as defined as below:

$$MI(x,y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

[3]one-hot encoding was used on this input

$$NMI(x,y) = \frac{MI(x,y)}{\sqrt{S(x)S(y)}}$$

where MI is similar to the covariance and NMI is similar to Pearson correlation coefficient where $S(x)$ and $S(y)$ are the standard deviations of x and y respectively. Average Product Correction (APC) was applied to both of the above [33].

$$H(x,y) = S(x) + S(y) - MI(x,y)$$

A faster python implementation of GaussDCA was implemented for PconsC4 which based on the Julia implementation by Baldassi et al [34].

The 1D inputs were passed through a pre-trained model called ProQ4 [35] to obtain pre-trained features of secondary structure and surface accessibility for each amino acid pair [3]. An outer product of the 1D inputs was calculated and after a batch-normalization step, these inputs were concatenated with the 2D inputs before feeding them into the network architecture.

2.4 Approaches used for training

2.4.1 Inter-residue contact distance prediction

In order to approach contact prediction as a distance-based classification problem, the distances between the residues had to be converted from a continuous variable to categorical variable. The distances were categorized into bins using the *to_categorical* function from the Utils package in Keras[4] as distance = to_categorical(distance, num_classes = no_bins) where no_bins is the number of bins. During this study, models were generated for number of bins as 7, 12 and 26. If the number of bins were 7, the bins were categorized from 4 Å till 14 Å with increments of 2 Å (bins were defined as [4, 6, 8, 10, 12, 14]). If the number of bins were 12, the bins were categorized from 5 Å till 15 Å with increments of 1 Å (bins were defined as [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]). And lastly, if the number of bins were 26 [7], the bins were categorized from 4 Å till 16 Å with increments of 0.5 Å (bins were defined as [4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5, 14, 14.5, 15, 15.5, 16]).

The inputs were passed through the CNN with U-Net architecture (described in Section 1). After each convolutional layer, a ELU activation was added (described in Section 5) [23]. Subsequently a batch-normalization layer (described in 3 [21] and Dropout Layer[5] with probability 0.1 was added [36]. The weights used for training the network were initialized using 'he_normal' distribution as described in [37]. For regularization the weight decay that was used for L2-norm was of value 10^{-12}.

The outputs were contact maps with thresholds 6 Å, 8 Å and 10 Å and the predicted distance between each amino acid pair. The three contact maps were calculated using binary cross entropy loss and distance was calculated as a classification problem using categorical cross entropy loss. To obtain the outputs for

[4]Keras is a Deep Learning Library in Python

[5]Dropout is a technique where connections between neurons are left out to increase regularization and reduce over-fitting.

distance-based classification as a probability distribution, the output layers were passed through the softmax function (described in Section 5). Therefore, the output of this layer would give an array with the probability associated with each bin. So for the model generated with 7 bins, the predicted output would be an array of 7 values with probabilities that sum up to 1.

The model was trained for 50 epochs using the Adam optimizer. The initial learning rate that was used was equal to 0.001. Using the Keras callback ReduceLROnPlateau, the value of training loss of distance measure was monitored and the learning rate was reduced by a factor of 0.5 if the loss did not decrease within 5 epochs. After every epoch, the order of entries within the training dataset was reshuffled. Finally, the model having the least loss on distance measure on the validation dataset was selected (epoch 25 for model with 7 bins, epoch 32 for model with 12 bins and epoch 25 for model with 26 bins.

In order to calculate the predicted distance from the array of predicted probabilities, the mid point of each bin range was taken and it was multiplied by the probability of that respective bin. For example (illustrated in Figure 2.2), for the model generated with 7 bins, the midpoint of each bin would give the array [2, 5, 7, 9, 11, 13, 15]. If we take an example of predictions for this model as an array of [0.02045989, 0.33463833, 0.28087327, 0.1577469, 0.07977389, 0.02461922, 0.10188856], then the predicted distance is calculated as (2*0.02045989 + 5*0.33463833 + 7*0.28087327 + 9*0.1577469 + 11*0.07977389 + 13*0.02461922 + 15*0.10188856) which is equal to 7.825 Å. The actual distance of this example was 7.013 Å. Since both of these distances are less than 8 Å, they are both classified as a contact. Therefore, this example had a probability distribution which correctly predicted a contact.

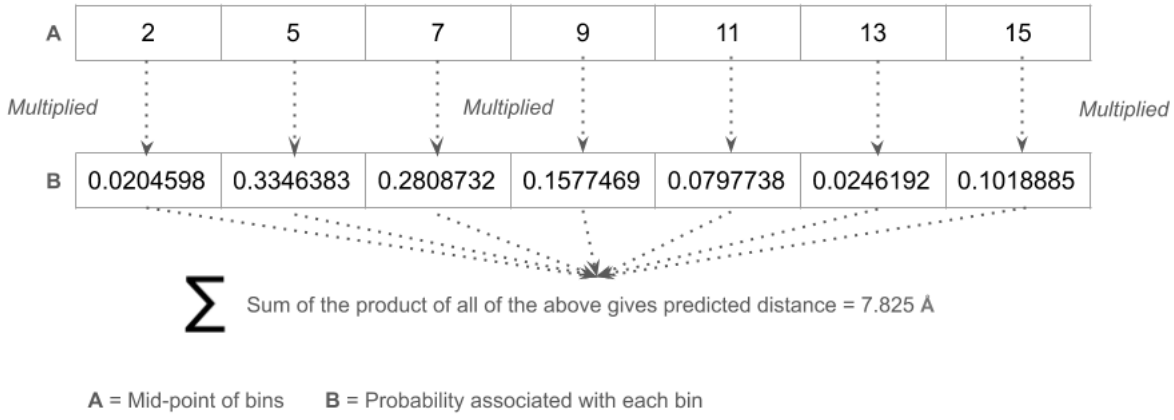


Figure 2.2: Illustration describing how the distance was calculated from the predicted probability distribution of distances

2.4.2 U-Net++ Architecture Implementation

In order to access whether U-Net++ architecture improved contact predictions as compared to results of PconsC4 which used U-Net architecture, the contact predic-

tion was defined as a regression problem.

The inputs were passed through the U-Net++ architecture (described in Section 2). Each convolutional layer was succeeded by a dropout layer with probability 0.1 was added [36]. This was followed by an ELU activation layer (described in Section 5) [23] and then a batch-normalization layer (described in 3) [21]. As described in [37], the weights used for training this network were also initialized using 'he_normal' distribution. The weight decay used for regularization (using L2-norm) was of value 10^{-12}.

The outputs were contact maps with thresholds 6 Å, 8 Å and 10 Å and the S-score similar to PconsC4 model [3]. The contact maps of varying thresholds were calculated using binary cross entropy loss and S-score was resolved as a regression problem using Mean Absolute Error loss as done in PconsC4 model [3].

The model was trained for 100 epochs also using the Adam optimizer. The learning rate had also been initialized to 0.001. The Keras callback ReduceLROnPlateau was used to monitor the value of training loss on S-score. If the S-score loss on the training set did not decrease within 5 epochs, the learning rate was decreased by a factor of 0.5. Similar to distance-based classification, the training data was shuffled following every epoch. The model having the least S-score loss on the validation est was chosen (epoch 75).

In order to calculate the distance from the model prediction (predicted distance d_{ij} the following equation is utilized:

$$d_{ij} = d_0 \sqrt{\frac{1}{S-score} - 1} \tag{2.1}$$

Where d_0 is the reference distance which was set to 8 Å and S-score as defined in Section 1.3

2.5 Evaluation Metrics

Based on the confusion matrix shown in Table 2.1, the following evaluation metrics were calculated: Precision (Positive Predicted Value or PPV), Recall, F1 Score, Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). Absolute Error and Relative Error were calculated for only the classification models.

- The commonly used metric to evaluate performance of contact predictions is Positive Predicted Value (PPV) or Precision [11][2][3]. Precision is calculated as the percentage of correctly predicted contacts out of the total number of predicted contacts. For a contact to be correctly predicted, the predicted distance and the actual distance should be less than 8 Å.

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

  So, for this project, PPV or Precision was calculated as: Residue pairs with predicted distance $<$ 8 Å AND actual distance $<$ 8 Å divided by all residue pairs with predicted distance $<$ 8 Å.

Table 2.1: Confusion Matrix to represent contacts and non-contacts where TP is True Positive, FP is False Positive, FN is False Negative and TN is True Negative

		Predicted Class	
		Contact (d_{ij} < 8 Å)	**Non - Contact** (d_{ij} > 8 Å)
Actual Class	**Contact** (d_{ij} < 8 Å)	True Positive (TP)	False Positive (FP)
	Non - Contact (d_{ij} > 8 Å)	False Negative (FN)	True Negative (TN)

- Recall or True Positive Rate (TPR) or Sensitivity is calculated as the percentage of correctly predicted contacts (when predicted distance <8 Å) out of the number of true contacts (when actual distance < 8 Å).

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

So, for this project, Recall was calculated as: Residue pairs with predicted distance < 8 Å AND actual distance < 8 Å) divided by all residue pairs with Actual distance < 8 Å.

- F1 Score is another measure to evaluate the model's accuracy and information retrieval. It is a single score by combining the precision and recall as follows:

$$F1Score = \frac{2*Precision*Recall}{Precision+Recall}$$
$$F1Score = \frac{2*TP}{2TP+FN+FP}$$

- Mean Squared Error (MSE) is a measure of the quality of the predictor and is calculated by taking the average of the squares of the errors for every residue pair in the protein. The error for each residue pair is the difference between the actual and predicted distance. This was calculated as follows:

$$MSE = \frac{1}{n}\sum_{i-1}^{n} *(PredictedDistance–ActualDistance)^2$$

- Root Mean Squared Error (RMSE) is a more standard measure of quality of the predictor like MSE. It is analogous to standard deviation and is calculated by taking the square root of the MSE. This was calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i-1}^{n} *(PredictedDistance–ActualDistance)^2}$$
$$RMSE = \sqrt{MSE}$$

All of the above metrics were calculated for the top L contacts for all the models from Section 2.4.1 and Section 2.4.2 (where L is the length of the protein sequence). These metrics can be calculated for top $L/2$, $L/5$ or $2L$ contacts as the number of

actual contacts is proportional to the length of the protein sequence that is under consideration. The metrics can be presented for short (with index of residue pairs between 5 and 12), medium (with index of residue pairs between 12 and 23), long (with index of residue pairs between 23 and 10000) or all (entire range) of contacts. The following two metrics were calculated for models from Section 2.4.1.

- Absolute Error (AE) is defined as the absolute of the difference between the predicted and actual distance for the residue pairs in contact [7]. This was calculated as:

$$AE = |PredictedDistance–ActualDistance|$$

- Relative Error (RE) is defined as AE normalized over the average of the predicted and actual distance [7]. This was calculated as:

$$RE = \frac{AE}{(1/2)*(PredictedDistance+ActualDistance)}$$

In this study , 'top L' contacts were calculated by two different methods:

- In the first method: For distance-based classification models, the sum of all the probabilities with class labels less than 8Å was taken and then sorted in the reverse order. The first L residue pairs (contacts) from this list were chosen as the top L contacts. So for a protein with length L = 176, the top L contacts for that protein would be 176 and the total number of top L contacts for the testing set would be the sum of all the protein sequences from the entire test set. For regression models calculating S-score as output, the predictions were sorted in reverse order and the first L contacts from the list were selected.

- In the second method: For classification models with distance class label as the output, the predicted distance of each residue pair was calculated as shown in 2.2. These calculated distances between the residue pairs were sorted and the first L residue pairs with the shortest distances were taken for further calculating the metrics. For the regression models, the predicted distance was calculated using equation 2.1. These were also sorted and evaluated using the first L shortest distances as done for the classification problem.

2.6 Development environment

The complete development for this project was done using Python (version 2.7.15rc1). This was a suitable programming language since it is easily interpretable and has a large variety of libraries and packages to solve deep learning problems. Moreover, the development of PconsC4 was done using Python so the inputs could easily be introduced into the models of the current thesis study. The backend library used to implement the deep learning models was TensorFlow (version 1.13.0) [38] and the application programming interface (API) that allowed for easy communication with TensorFlow was the Keras package (version 2.2.4) [39]. Keras contains the high-level directive commands which allow developers to focus on the problem instead of figuring out the low-level computations in the backend. The python packages NumPy

(version 1.16.1) and SciPy (version 1.2.1) were used for scientific computing. This project was developed on a Linux-based Ubuntu system (Ubuntu 18.04.2 LTS) which used one Graphics Processing Unit (GPU) (Nvidia GeForce GTX 1070) for faster computations. In order to run Python and the associated libraries on the GPU, TensorFlow-GPU (version 1.13.1) was installed to use CUDA[6]. To safely download, operate and maintain these libraries and packages, a singularity container was used with an Ubuntu operating system (Ubuntu 16.04.5 LTS).

2.7 Ethical Considerations

All the data used for this project has been taken from Protein Data Bank (PDB) [31], a public repository for protein information. None of the data entries have an association with any particular individual and is completely desensitized. The entire project was carried out in-silico with publicly available scientific data and therefore, informed consent and medical ethics clearance did not fall under the scope of this project. The purpose of this study is purely academic to progress the field of contact predictions to enable accurate protein structure prediction. The author hereby declares that no conflict of interest exists.

[6]CUDA is a parallel computing platform created by Nvidia to enable general computing and processing on the GPU.

3

Results

The python files for creating, training and testing the models for 2.4.1 and 2.4.2 are available at github.com/aditishenoy/unetplus. This chapter summarizes the results from the distance-based classification models followed by the results of implementing the alternative U-Net++ architecture.

3.1 Inter-residue contact distance prediction

A detailed comparison on the models generated with different number of bins (7, 12 and 26) for distance predictions between a pair of amino acids in a protein is presented in this section. The results for the model generated with 7 bins are given below, whereas those generated for 12 and 26 bins can be found in Appendix B. The results from the evaluation metrics (as described in Section 2.5) are shown below in Table 3.1:

Table 3.1: Precision, Recall, F1 Score, Absolute error, Relative Error for the top L contacts, where L is the sequence length, at different sequence thresholds (short, medium, long, all)

Number of bins	Sequence thresholds	Precision	Recall	F1 Score	Absolute Error	Relative Error
7	all	0.0725	0.0182	0.2332	3.1694	0.2985
	short	0.3167	0.1152	0.1659	2.5322	0.2422
	medium	0.3133	0.0888	0.1513	3.3176	0.2837
	long	0.0557	0.0147	0.2066	4.4780	0.3421
12	all	0.0700	0.0189	0.2437	2.5901	0.2660
	short	0.3175	0.1158	0.1666	2.5188	0.2417
	medium	0.3117	0.0888	0.1514	3.1894	0.2780
	long	0.0537	0.0151	0.2107	3.6016	0.3024
26	all	0.0699	0.0193	0.2508	2.7025	0.2723
	short	0.3185	0.1161	0.1673	2.5490	0.2446
	medium	0.3116	0.0891	0.1523	3.2305	0.2812
	long	0.0541	0.0154	0.2108	3.8403	0.3171

Table 3.2: Precision, Recall, F1 Score for the top L contacts for state-of-the-art contact prediction method PconsC4 (based on U-Net architecture)

	Sequence thresholds	Precision	Recall	F1 Score
PconsC4 (with U-Net)	all	0.5992	0.3532	0.4347
	short	0.2895	0.1517	0.1955
	medium	0.2750	0.1467	0.1891
	long	0.4118	0.2314	0.2949

The evaluation metrics for the classification models with 7, 12 and 26 bins have been reported in Table 3.1. These models show low PPV in the range of 0.07 and 0.3. There is no significant difference in values of AE and RE among these models. If we compare the evaluation metrics calculated for state-of-the-art contact prediction model PconsC4 (as shown in Table 3.2) with the inter-residue distance classification models, the precision values for all-range and long-range contacts are fairly lower for the classification models. However, there is a slight improvement in PPV values in the classification models for short-range and medium-range contacts as compared to PconsC4.

In order to visualize the predicted distance with respect to the actual distance, scatter plots have been plotted between the predicted distances and actual distances between residues in a protein. In an ideal case where all the predictions were perfectly accurate and precise, we could expect a straight line from the bottom left to top right corner of the plot. Every point in the graph is for a specific residue pair and is represented as x = Predicted distance and y = Actual distance. Since there are a large number of residue pairs in a protein, different colours have been used to represent the density of points in a particular region of the plot. For example, the region with the highest density have the points in yellow and with the least density are in purple. The gradient from yellow to purple shows the corresponding decreasing density of points.

The following figures 3.1 and 3.2 represent scatter plots between the predicted distance < 8 Å) and actual distance < 8 Å) for classification model generated with 7 bins. Only distances which are less than 8 Å have been taken into consideration. This model appear to predict values mostly in the first category of < 4 Å to 4 Å which means it is under-predicting the actual inter-residue distance.

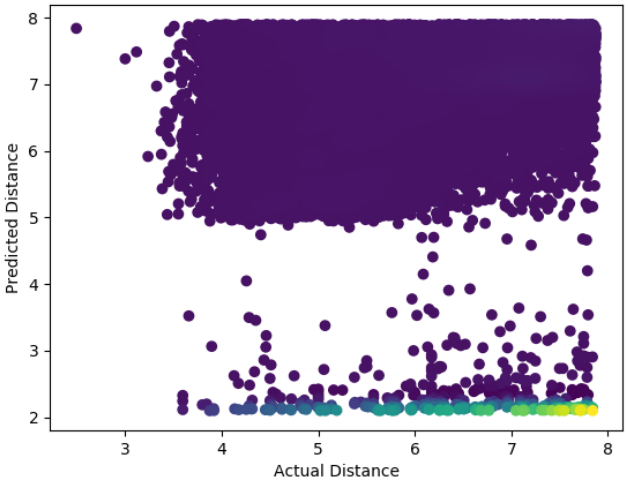


Figure 3.1: Predicted Distance < 8 Å vs Actual Distance < 8 Å for Classification Model with 7 bins for all contacts in the protein.

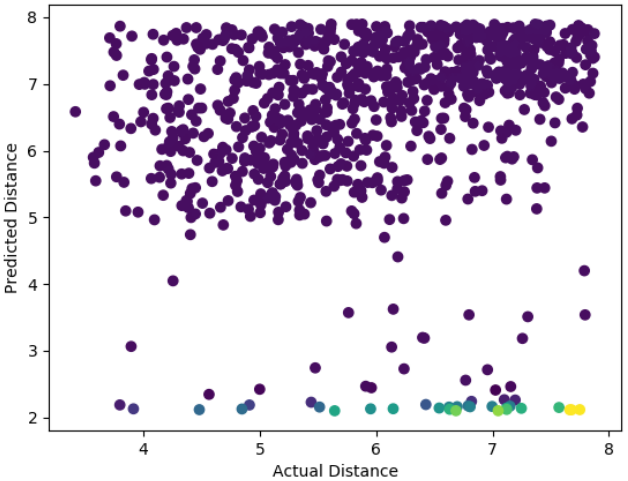


Figure 3.2: Predicted Distance < 8 Å vs Actual Distance < 8 Å for Classification Model with 7 bins for top L contacts in the protein.

The following figures 3.3 and 3.4 represent scatter plots between all the predicted distances and all the actual distances for classification model generated with 7 bins. The high density region marked in greenish yellow shows that most inter-residue distances are above 20 Å for all contacts and less than 10 Å for top L contacts.

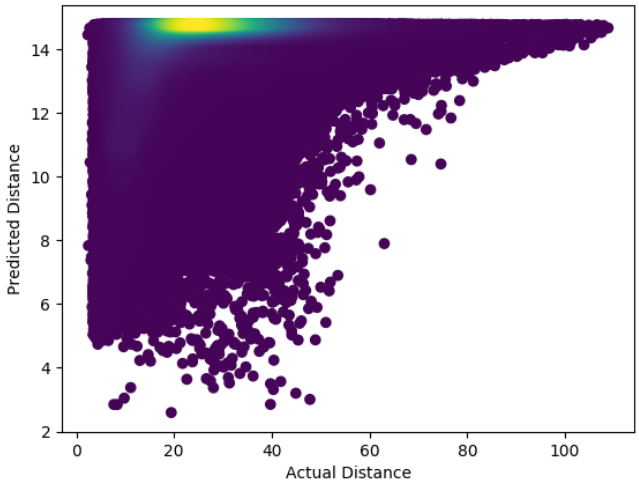


Figure 3.3: All the predicted distances vs all the actual distances for Classification Model with 7 bins for all contacts in the protein.

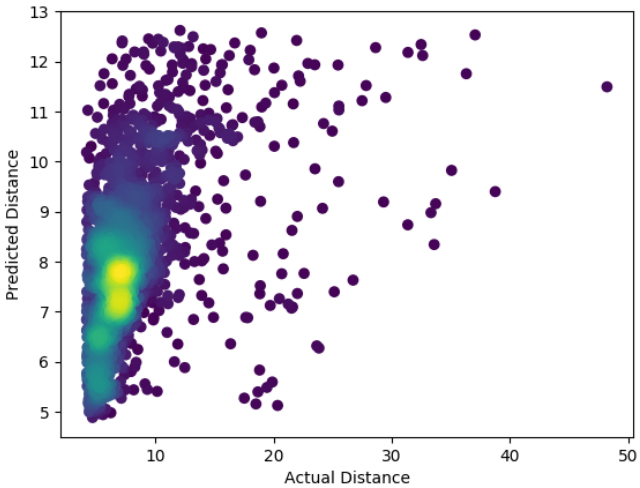


Figure 3.4: All the predicted distances vs all the actual distances for Classification Model with 7 bins for top L contacts in the protein.

The following figures 3.5 and 3.6 represent the probability distributions for correctly and incorrectly predicted contact for the classification model generated with 7 bins.

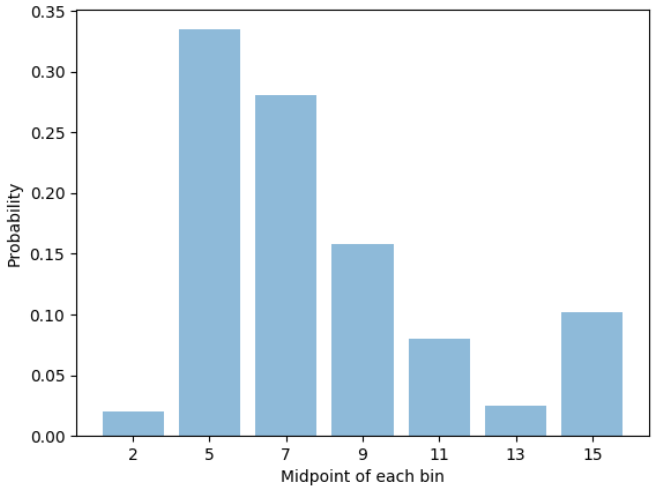


Figure 3.5: Probability distribution of correctly predicted contact (Actual value = 7.013, Predicted value = 7.825)

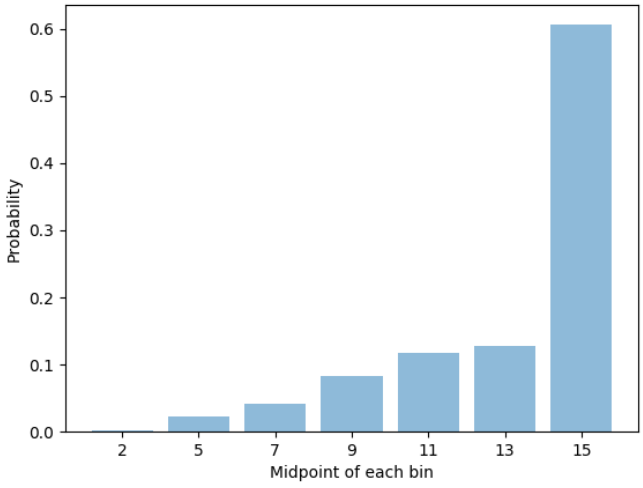


Figure 3.6: Probability distribution of incorrectly predicted contact (Actual Value = 4.775, Predicted Value = 13.201)

The objective of predicting inter-residue contact distance was to obtain more information from a distance probability distribution as shown in Figures 3.5 and 3.6. (refer to bar plots for 12 and 26 bins in Appendix B) From these probability distributions, how the models classify the distance in the form of categorical distance bins can be visualized.

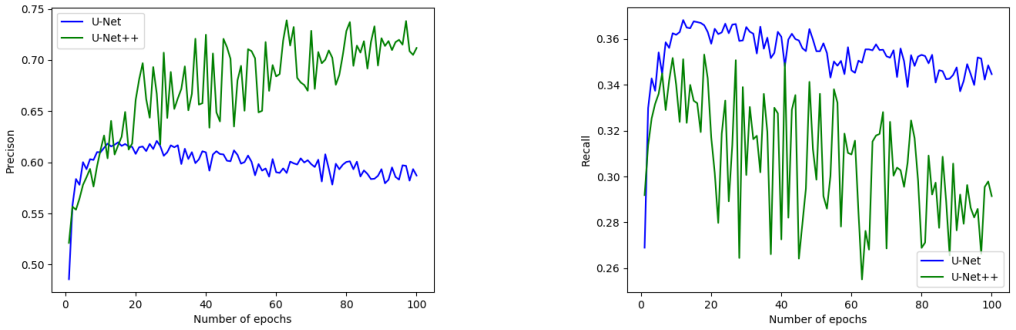
3.2 U-Net++ Architecture Implementation

The evaluation metric results for the regression models (using U-Net architecture (PconsC4) and using U-Net++ architecture) are shown in Table 3.3.

Table 3.3: Precision, Recall, F1 Score for the top L contacts, where L is the sequence length, at different sequence thresholds (short, medium, long, all)

	Sequence thresholds	Precision	Recall	F1 Score
PconsC4 (with U-Net)	all	0.5992	0.3532	0.4347
	short	0.2895	0.1517	0.1955
	medium	0.2750	0.1467	0.1891
	long	0.4118	0.2314	0.2949
Regression model generated using Unet++	all	0.6736	0.3079	0.4053
	short	0.3043	0.1540	0.2006
	medium	0.3005	0.1432	0.1907
	long	0.5246	0.1921	0.2719

By implementing U-Net++ architecture instead of U-Net in the PconsC4 model, a significant increase in the PPV values is seen from Figure 3.3. For these regression models, the long-range and all-range residues in contact perform better than short-range and medium range residues. This also can be seen in the following figures 3.7a and 3.7b which represent the precision vs epochs and the recall vs epochs for the regression model trained with U-Net (blue) and U-Net++ (green) architecture. In the Figure 3.7a the precision value for U-Net++ increases against each epoch as compared to the U-Net architecture. The value of recall is slightly compromised as seen in Table 3.3 and Figure 3.7b.



(a) Precision vs Epochs

(b) Recall vs Epochs

Figure 3.7: Precision and Recall vs Epochs

The following figures 3.8 and 3.9 represent scatter plots between the predicted distance < 8 Å) and actual distance < 8 Å) for regression model with U-Net and U-Net++ architecture for all contacts. These plots are visually similar for for all contacts for both the architectures.

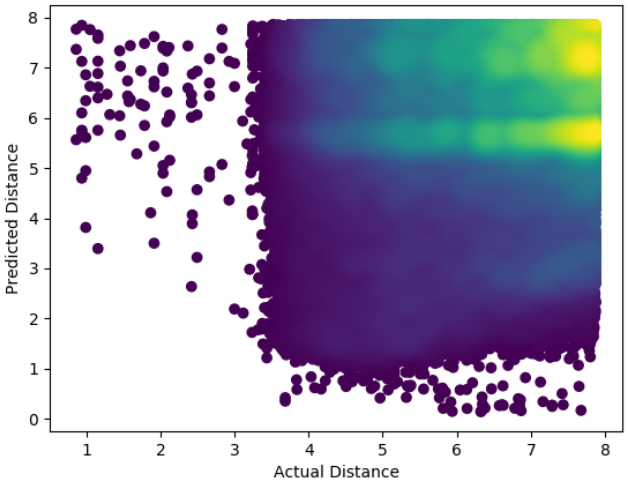


Figure 3.8: Predicted Distance < 8 Å vs Actual Distance < 8 Å for regression model with U-Net architecture for all contacts in the protein.

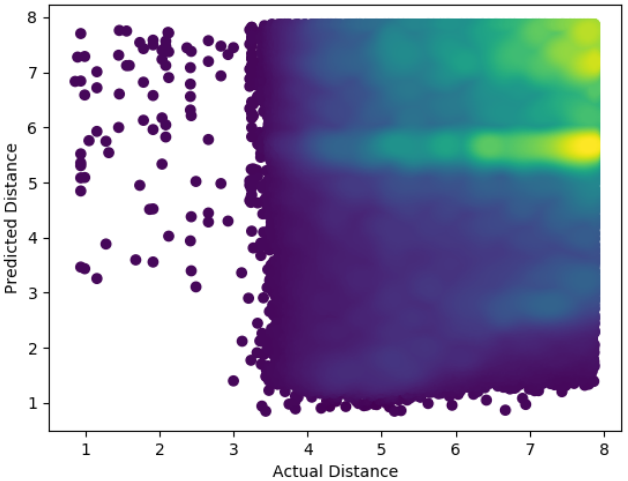


Figure 3.9: Predicted Distance < 8 Å vs Actual Distance < 8 Å for regression model with U-Net++ architecture for all contacts in the protein.

The following figures 3.10 and 3.11 represent scatter plots between the predicted distance < 8 Å) and actual distance < 8 Å) for regression model with U-Net and U-Net++ architecture for top L contacts. In these figures below, a more horizontally dispersed area can be observed for U-Net whereas in U-Net++ a more focused area around 2 Å can be seen.

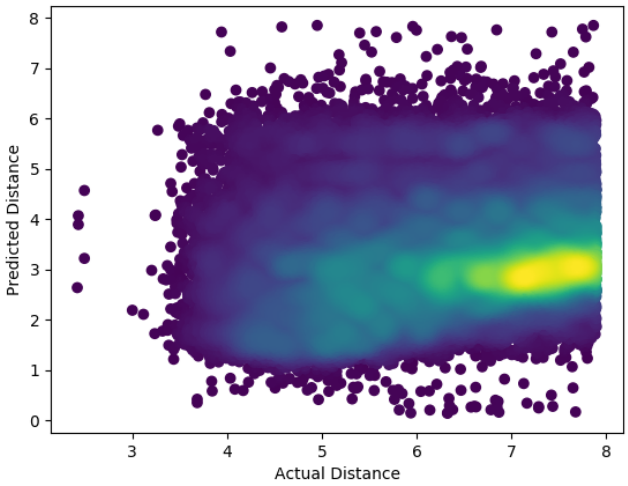


Figure 3.10: Predicted Distance < 8 Å vs Actual Distance < 8 Å for regression model with U-Net architecture for top L contacts in the protein.

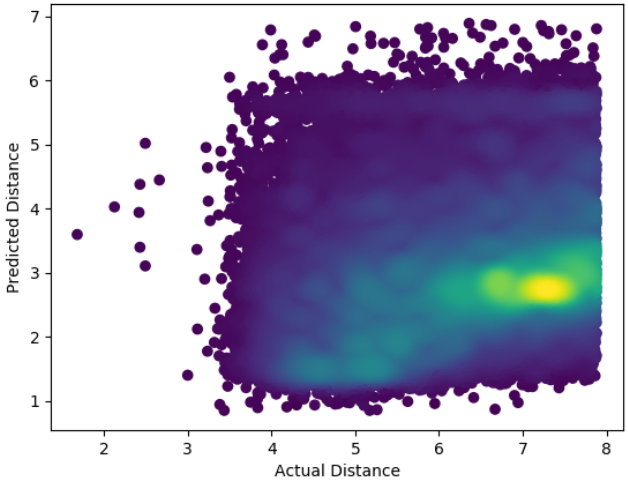


Figure 3.11: Predicted Distance < 8 Å vs Actual Distance < 8 Å for regression model with U-Net++ architecture for top L contacts in the protein.

The following figures 3.12 and 3.13 represent scatter plots for top L contacts between all the predicted distances and all the actual distances for regression model with U-Net and U-Net++ architecture. In these figures, U-Net model generates a more focused area around 8 Å. In U-Net++ model, the density is more dispersed and begins to represent a straight diagonal line between predicted and actual distance. So as the actual distance increases, the predicted distance also correspondingly increases for U-Net++.

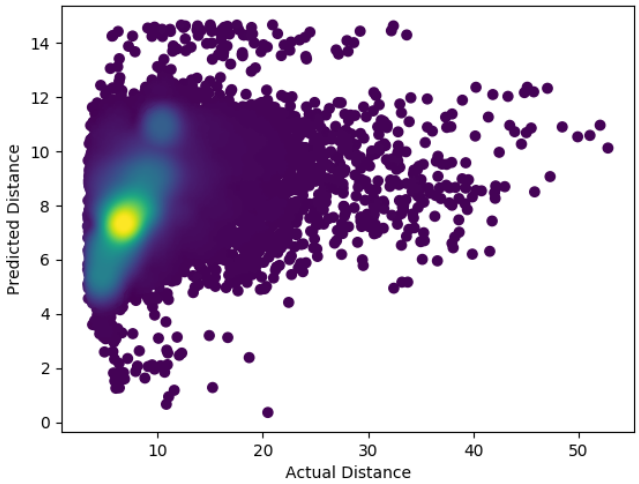


Figure 3.12: All the predicted distances vs all the actual distances for regression model with U-Net architecture for top L contacts in the protein.

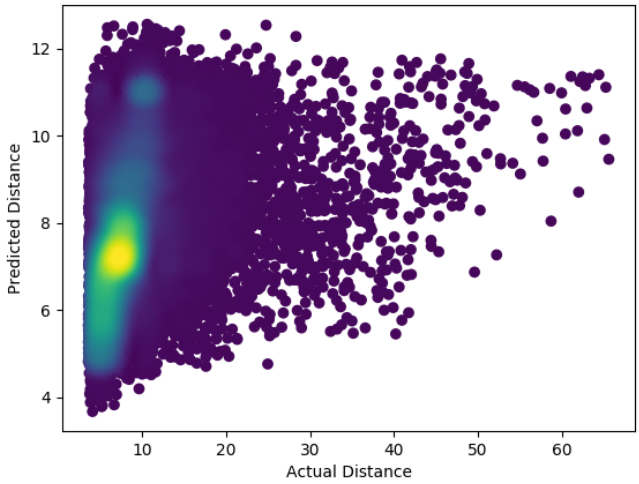


Figure 3.13: All the predicted distances vs all the actual distances for regression model with U-Net++ architecture for top L contacts in the protein.

The following figures show the loss on distance score on the validation dataset and the learning rate for both classification (Section 2.4.1) and regression (Section 2.4.2). In Figure 3.14 and 3.15, the Clas Model represent the models trained for inter-residue distance prediction and the Reg Model indicate the models trained with U-Net and U-Net++ architecture.

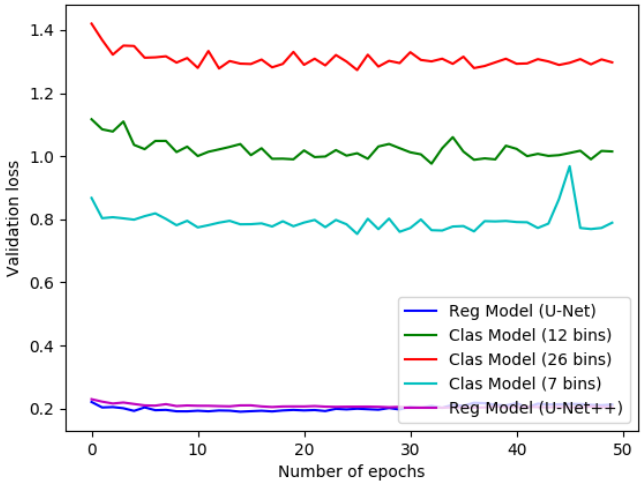


Figure 3.14: Loss of distance score for classification models and loss of s-score on regression models on validation set for 50 epochs.

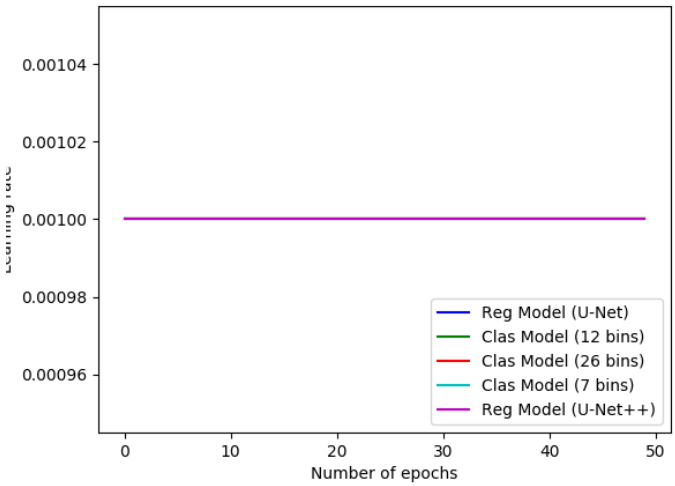


Figure 3.15: Learning rate for all models (classification and regression) on validation set for 50 epochs.

In Figure 3.14, the validation loss of the classification models based on inter-residue distance are higher than the regression models with U-Net (PconsC4) and U-Net++. During the first 50 epochs, it appears that no great improvements in the learning takes place as the learning rate is constant as seen in Figure 3.15.

The following figures show the loss on distance score on the validation dataset and the learning rate for only the regression models trained with U-Net++ architecture with S-score as the output. In Figure 3.16 and 3.17, U-Net indicate the PconsC4 model which used U-Net architecture and U-Net++ indicates the model generated during this thesis project which used U-Net++ architecture.

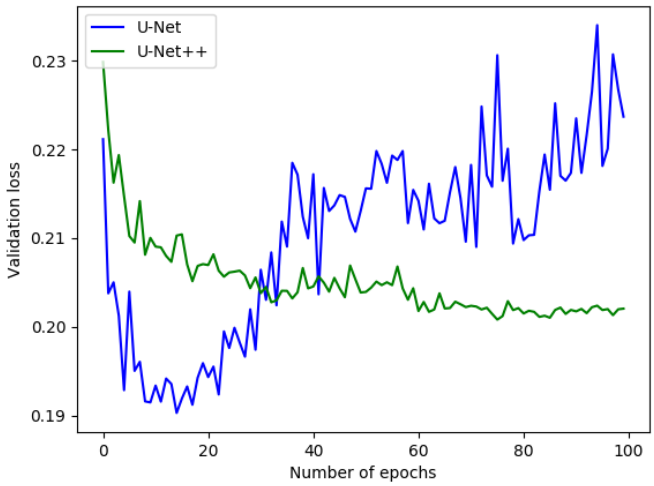


Figure 3.16: Loss of s-score on regression models (trained with U-Net architecture (PconsC4) and on U-Net++ architecture on validation set for 100 epochs.

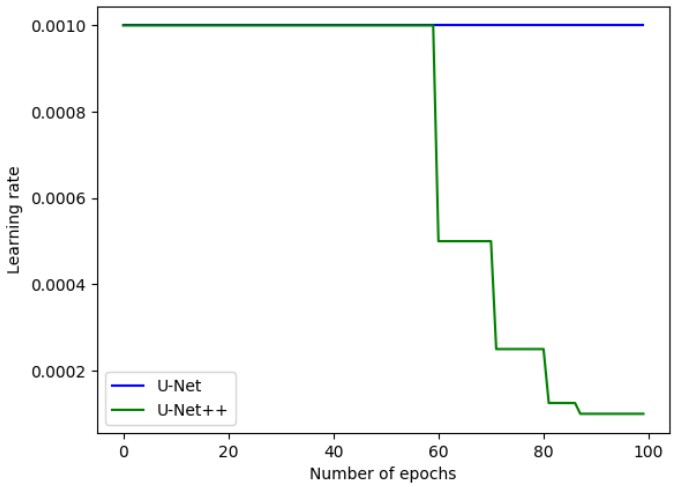


Figure 3.17: Learning rate of s-score on regression models (trained with U-Net architecture (PconsC4) and on U-Net++ architecture on validation set for 100 epochs.

In figure 3.16 (which is closeup of Figure 3.14), it can be seen that the model trained with U-Net architecture (PconsC4) was over-fitting and the U-Net++ architecture showed a better gradual decrease in loss over 100 epochs. Figure 3.17 also showed that after epoch 60, the learning rate reduced in small steps which can signify that the model was approaching the minimum value of loss for the function.

The following table 3.4 gives a comparative overview of the values for RMSD and MSE for all the classification models with 7, 12 and 26 bins as well as the regression models with U-Net (PconsC4) and U-Net++ architecture. The value of the classification models are higher than the regression models.

Table 3.4: RMSD and MSE values for models trained with methods mentioned in 2.4.1 and 2.4.2

	Number of bins	RMSD	MSE
Distance based Classification	7	4.234	3.925
	12	3.834	3.558
	26	4.296	4.022
PconsC4 Regression (U-Net)	-	1.378	1.051
Regression (U-Net++)	-	1.297	0.988

The following table 3.5 shows the variation in PPV values depending on the the way the 'Top L contacts' were calculated for the distance-based classification models and the contact-based regression models.

Table 3.5: PPV values for two ways of calculating 'Top L contacts': Using Top L proteins with shortest distances and using Top L number of contacts from each protein sequence.

	Number of bins	Top L using shortest distance		Using Top L number of contacts
		PPV for threshold 8 Å	PPV for threshold 15 Å	PPV for threshold 8 Å
Distance based Classification	7	0.070	0.093	0.072
	12	0.069	0.094	0.070
	26	0.071	0.091	0.069
PconsC4 Regression (U-Net)		0.576	0.961	0.599
Regression (U-Net++)		0.585	0.940	0.673

The regression models performed better for top L number of contacts as compared to when top L shortest proteins were taken. Moreover, it can be seen that

when the threshold value is increased to 15 Å, the PPV correspondingly increases since the number of contacts that are correctly predicted less than 15 Å are significantly higher than the number of contacts less than 8 Å. Finally, it can be highlighted that the U-Net++ model performs better than the other classification models and PconsC4 which uses U-Net architecture.

The Figure 3.18 shows the contacts maps (for protein 1H97A) generated by regression models with U-Net and U-Net++ architectures respectively. The Figure 3.19 shows the distance maps (for protein 1H97A) generated by classification model with 7 bins.

(a) U-Net (b) U-Net++

Figure 3.18: Contact maps for Protein 1H97A generated by regression models using U-Net and U-Net++ architecture

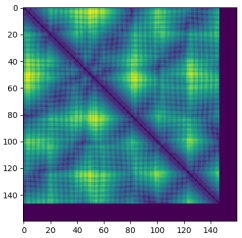


Figure 3.19: Distance maps for Protein 1H97A generated by distance classification models with 7 bins

These inter-residue distance maps and contact maps would give more information during ab-initio folding later on in the protein folding process.

4

Discussion

4.1 Major Findings

The aims of this thesis study were twofold. Firstly, it was to evaluate whether inter-residue contact distance prediction by classification analysis provided a better basis for contact prediction as compared to state-of-the-art methods using regression analysis. Secondly, it was to compare and evaluate whether a nested U-Net architecture (U-Net++) could retain features from the contracting path to the expanding path as compared to U-Net architecture. The U-Net++ architecture was expected to bridge semantic gap and thereby produce better predictions of contact maps.

The following sections will revisit the two-fold research objectives of this study. Along with an interpretation of the results from the two experiments conducted, the major findings of this study have been illustrated. Additionally, an evaluation of these findings in the context of the field have also been described.

Inter-residue contact distance prediction

Based on results from studies presented during CASP13 as described in Section 1.3, inter-residue distance based classification was an important advancement for prediction of contact maps. To implement this distance-based classification in PconsC4 contact prediction model, the distance between the residues were categorized into bins and a CNN was developed to predict the respective distance class label. This distance probability distribution was expected to give more information as compared to the binary output as presence or absence of a contact between a residue pair in a protein. The probability distribution could also give information about how sure the model was while predicting distance between two contacts. For example, if the probability of the bin having class label 8 Å is 0.8, then it could be said that the model was fairly sure of the prediction of the distance between that residue pair. However, if the probability of the bin having class label 8 Å is less than or equal to 0.5, the model is taking a random guess at making the prediction and has not learned enough to make a prediction of distance with respect to that specific residue pair. Therefore, these residue pairs can then be isolated and further studied to understand what key elements of information the model is failing to learn.

However, the results of this study do not seem to show an improvement in performance of contact predictions on using inter-residue distance classification. When the PPV values of the classification analysis were compared to the PPV values calculated for state-of-the-art PconsC4, it was found that long-range and all-range contacts showed lower precision than PconsC4. Short-range and medium-range contacts, however, showed improved precision values. The RMSD values for the classification models were also significantly higher than those of regression analysis. Since the distances for these models were calculated for all the contacts, it is possible that the predicted distance was in the range of approximately 15 Å so if actual distances from PDB were higher than 20 Å for a residue pair, the difference between the predicted and actual distance would be high and therefore, the MSE and RMSD values would also be high. A better range of values could be obtained if the contacts which were less than 4 Å and greater than 8 Å were filtered out.

U-Net++ Architecture Implementation

U-Net++ architecture was designed as a dense and more nested network with redesigned skip pathways. This architecture was expected to bring the features from the contracting and expanding maps closer together and thereby improve the prediction of bio-medical images. This study aimed to validate this hypothesis and compare the use of U-Net and U-Net++ architectures for protein contact image segmentation. For the implementation of U-Net++ architecture, regression analysis on S-score was done. Based on the results obtained from this study, it was found that by using U-Net++ architecture the PPV value was increased to 0.67 as compared to 0.59 when U-Net architecture was used for all-range contacts. There was an improvement in short, medium and long range contacts as well. This could suggest that the U-Net++ model was better able to predict correct contacts that were actually correct (precision increased). However, the ability of the model to predict actual contacts correctly reduced on using U-Net++ (recall decreased).

In the scatter plots of predicted distance against actual distance, it was found that there was a linear increase between actual and predicted distance. This means, as compared to U-Net model, the predicted distance appeared to increase along with increasing values of actual distance. Top L contacts predicted from the contact prediction model are used in the pipeline for protein structure prediction, so the improved patterns for U-Net++ model make it more desirable for use as compared to U-Net model.

Moreover, the RMSD and MSE values were lower for U-Net++ model. This means that the error of each distance prediction was lower for U-Net++ as compared to any of the other models, thereby suggesting the accuracy of these predictions were better.

Finally, to summarize, the following section will concisely answer the research questions in the context of this study:

1. *Can inter-residue contact distance prediction (a classification problem) give better results of contact predictions than S-score based contact prediction (a regression problem) in PconsC4 contact prediction model?*

   Contrary to results of CASP13, the global competition for protein structure prediction, inter-residue distance based classification did not improve the results of contact prediction model PconsC4. PconsC4, being a fast state-of-the-art contact prediction software is trained on S-score and solved using regression. When PconsC4 was trained on inter-residue distances using classification, it showed lower values of evaluation metrics - precision, recall, F1 score and higher values of AE and RE than the original PconsC4 model. This could either be because the inputs for this experiment were the same as those used for PconsC4 and were not altered in context of this study or because of insufficient amount of training data used to train the models on inter-residue distance. Therefore, this study showed that solving PconsC4 as a classification problem did not improve the results of contact prediction as compared to solving PconsC4 as a regression problem.

2. *Can U-Net++ architecture improve contact predictions of PconsC4 as compared to predictions based on U-Net architecture?*

   This thesis study confirmed that use of U-Net++ architecture improved the precision of contact prediction model PconsC4 as compared to the original PconsC4 which used U-Net architecture. This could be because the features of the contracting path were close to that of the expanding path. The network is dense and the skip pathways used in the U-Net++ were redesigned for more efficiency. Therefore, this study showed that U-Net++ architecture showed improved contact predictions as compared to the predictions based on U-Net architecture.

The two experiments were conducted independently of each other. For inter-residue distance classification, U-Net architecture was used and the training parameters of PconsC4 were used. The output in this case were distance based probabilities. In U-Net++ architecture implementation, the training parameters of PconsC4 were again used by only changing the architecture and certain hyper-parameters for optimization. The output in this case was the S-score. The suggestive next step would be to combine both the experiments into one so the PPV values for short- and medium-range contacts could be increased by inter-residue distance classification and the PPV values for all- and long-range contacts could be increased by using U-Net++ architecture.

The top L predicted contacts are used for protein structure prediction as illustrated in Figure 4.1. This pipeline for structure prediction is an example of ab-initio protein folding as the primary amino acid sequences are used without using known structures of protein homologs. A challenge associated with protein folding is that there are subtle differences in local and global alignment in protein sequences and

there have been limited scientific advances to predict these subtle differences accurately [10]. To overcome this challenge and reduce the entropy of possible protein confirmations, ab-initio approach has been used in this study.

4.2 Comparison with other studies in the field

Based on the related research and recent advances in CASP13 as described in Section 1.3, inter-residue distance contact prediction showed to improve contact predictions and accuracy of overall protein structure prediction [5][6][7]. The results of this thesis study, however, does not corroborate with the results of the above described studies. The inter-residue contact prediction by classifying the inputs into categorical variables reduced the performance of PconsC4 significantly. In Hou et al's study [6], the input sequence was used to create input features matrices which were passed through a CNN in DNCON2[1] [40] to produce contact distance maps. These distance probability distributions were concatenated with the input feature matrices and then passed through another CNN to produce the final contact map using probabilities 8 Å [6]. During the current thesis study, the concatenated inputs were directly converted into categorical variables without passing them through any convolutions.

It is suggested that distance maps can give more accurate information regarding Cα atoms[2] than contact maps [41]. However, the distance maps of classification models of this study with 7, 12 and 26 bins produce similar distance maps with little contextual information that can be used for protein folding.

In Xu's study [7], the inter-atom distances were discretized into 26 bins and the sum of probabilities of the first 9 labels gave information about whether there was a contact or not between the residues. The predicted inter-atom distance along with the secondary structure information and torsion angles are used to build the 3D protein structure. Since concatenated 1D and 2D inputs were used for categorization in this thesis study, the misclassification of class labels might have occurred.

U-Net++ architecture has shown to improve the prediction of finer-details of images by highly dense and packed skip connections [27]. This can be observed in the adaptation of U-Net++ for the current thesis study as well. Although there is no enhancement in terms of the contact maps visually, there is an improvement of performance metrics. This could be because the model is learning the finer aspects of the contact map. So if the number of epochs for training the U-Net++ architecture is increased, an improvement in terms of performance metrics may be observed.

[1] A state-of-the-art contact prediction method which uses two convolutional neural networks

[2] The first carbon atom where the the functional group of an amino acid is attached. Understanding the position of this atom gives key information in protein folding.

4.3 Limitations and Practical Implications

The current study has provided the scientific community with a case under which inter-residue distance would not improve contact predictions contrary to results of recent studies [6][7]. There are certain limitations in this thesis study which need to be noted for future research.

Firstly, since the inputs (1D and 2D) were the same as the ones used for PconsC4 without any alteration, the concatenated inputs may not have been optimum for the current CNN network. Secondly, the number of epochs chosen to train the network may not have been optimal. The reason the classification models were set to only 50 epochs whereas the regression models were set to 100 epochs is because of its long training time (approximately 10-12 hours for 50 epochs. This hyper-parameter might not have been optimal for distance classification. Moreover, the hyper-parameters for the models were tested by trial and error method. Using hyper-parameter tuning methods like grid search, random search or optimization methods using packages like Talos (for Keras), scikit-learn or hyperopt could be an alternative to arrive with optimal initial parameters for training. Lastly, it is possible that the amount of training data was insufficient for the models to learn the finer details of the contact maps. Inter-residue distance prediction and U-Net++ architecture implementation are attempts to enhance contact predictions. Contact predictions, along with secondary structure information forms the input for protein structure prediction softwares, like CONFOLD [42] as described in Figure 4.1. The quality of these contact predictions determines the output of the overall protein structure prediction. If the protein structure can be accurately determined and the protein folding problem can be solved, the rate of our understanding of proteins and the diseases caused by misfolded proteins would substantially increase.

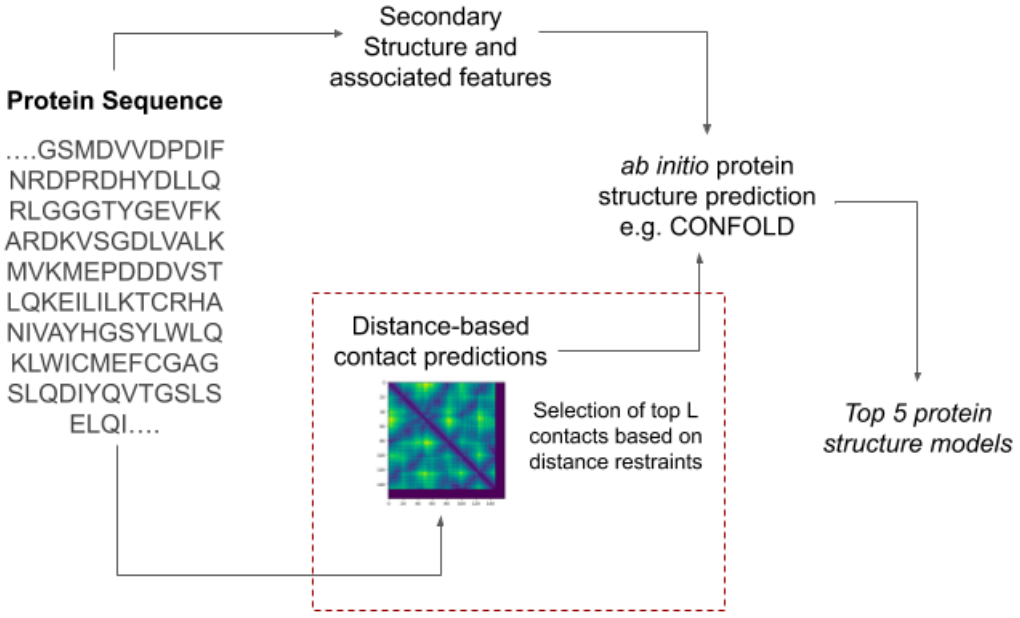


Figure 4.1: An example of an entire pipeline of protein structure prediction from sequence to 3D protein structure adapted from [6]

4.4 Future Research

As done in Hou et al's study [6], if the input features were pre-trained with a CNN before concatenation with secondary structure information and 2D inputs, the input feature matrices may include more information. Alternatively, only categorizing the distance inputs without pre-training with CNN may also impact the quality of predictions as done in Xu's study [7].

Additionally, a deeper look at the nature of inputs into the CNN would also had more value to the study. Studies like DeepCov [4] show that contact predictions comparable to state-of-the-art can be obtained by minimal sequence features. So it would be interesting to explore which features can be done without in PconsC4. Using minimal features which used categorical distance inputs and U-Net++ architecture could significantly improve the results of this study even further.

5

Conclusion

This thesis study aimed to optimize contact predictions which were based on inter-residue distance measure. The inputs for this model were 1D and 2D features generated from the protein sequence and the output was a probability distribution of each distance class label. These predictions were compared with contact predictions of state-of-the-art contact prediction method PconsC4. PconsC4 was trained on the same inputs as this study but the output was S-score which was trained as a regression problem.

The hypothesis before starting this thesis study was that inter-residue contact distance would give more information about the contacts and produce better protein contact maps. This hypothesis was based on studies described in 1.3. Since these contact maps were the inputs for protein structure prediction methods like CONFOLD, the accuracy of the contact maps would impact the accuracy of the 3D protein structures that were predicted. This study, however, showed that inter-residue distance-based classification did not provide more information or better results that S-score contact maps solved by regression. The inconsistent results with the other studies could be because of the difference in methods used to process the input features before passing them in to the prediction networks as described in 4.2.

Therefore, this thesis project contributes to the knowledge base in the field of contact map predictions. It provides a specific use case for inter-residue distances not providing the expected enhancement in output. These results help in designing future experiments for contact predictions that could be used for more accurate protein structure predictions. By solving the problem of contact map and protein structure prediction, our understanding of proteins would be revolutionized. This would not only result in more informed research of misfolded proteins and diseases but also transform our outlook towards diagnosis and treatment of diseases for better health care.

Additionally, this thesis project aimed to implement an enhanced version of the current U-Net architecture that is used in PconsC4 with U-Net++ architecture. The U-Net++ architecture showed improved detection of fine details of images because of re-designed skip pathways and was thus considered useful to apply to PconsC4. The PPV results of using U-Net++ showed a significant improvement as compared to the U-Net architecture despite not showing much of a visual difference in the

respective contact maps. So this thesis project provides a confirmatory test on how U-Net++ architecture produces a better outcome than U-Net architecture for bio-medical image segmentation. It also ensures the validity of using U-Net++ for application in other bio-medical image processing problems. The deep learning methods that are used for bio-image processing are constantly being upgraded so it is essential to be updated with the recent advances regarding these methods. This study helps in designing future studies in bio-image informatics and provides a comparison on results of U-Net and U-Net++ architectures. Thus, future researchers would be more informed in choosing an architecture more suited for their application.

Therefore, as described in 1.2, this thesis study is an aiding resource for bringing basic scientific advances closer to clinical applications.

References

[1] Weigt M, White RA, Szurmant H, Hoch JA, Hwa T. Identification of direct residue contacts in protein-protein interaction by message passing. Proceedings of the National Academy of Sciences of the United States of America. 2009 jan;106(1):67–72. Available from: http://www.ncbi.nlm.nih.gov/pubmed/19116270http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2629192http://www.ncbi.nlm.nih.gov/pubmed/19116270{%}0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2629192.

[2] Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model [Article]. PLOS COMPUTATIONAL BIOLOGY. 2017 jan;13(1).

[3] Michel M, Menéndez Hurtado D, Elofsson A. PconsC4: fast, accurate and hassle-free contact predictions. Bioinformatics. 2018 dec;Available from: https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/bty1036/5259184.

[4] Jones DT, Kandathil SM. High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. Bioinformatics. 2018 oct;34(19):3308–3315. Available from: https://academic.oup.com/bioinformatics/article/34/19/3308/4987145.

[5] Senior A, Jumper J, Hassabis D. Blog: AlphaFold: Using AI for scientific discovery; 2018. Available from: https://deepmind.com/blog/alphafold/.

[6] Hou J, Wu T, Cao R, Cheng J. Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. Proteins: Structure, Function, and Bioinformatics. 2019 feb;p. 552422. Available from: https://www.biorxiv.org/content/10.1101/552422v1.

[7] Xu J. Distance-based Protein Folding Powered by Deep Learning. bioRxiv. 2018;(Dl):465955. Available from: https://www.biorxiv.org/content/early/2018/12/18/465955.

[8] Lodish et al. Molecular Cell Biology (4th edition) Harvey Lodish, Arnold Berk, S. Lawrence Zipursky, Paul Matsudaira, David Baltimore and James Darnell; Freeman & Co., New York, NY, 2000, 1084 pp., list price $102.25, ISBN 0-7167-3136-3. vol. 29; 2001. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1470817501000236.

[9] Yonath A. X-ray crystallography at the heart of life science. Current Opinion in Structural Biology. 2011 oct;21(5):622–626. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0959440X11001096.

[10] Guo, Jun-tao; Ellrott KX, Ying. A Historical Perspective of Template-Based Protein Structure Prediction. In: Bystroff MZ, C, editors. Methods in Molecular Biology, vol. 413: Protein Structure Prediction. 2nd ed. Totowa, NJ: Humana Press Inc.; 2008. .

[11] Schaarschmidt J, Monastyrskyy B, Kryshtafovych A, Bonvin AMJJ. Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age. Proteins: Structure, Function and Bioinformatics. 2018;86:51–66. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5820169/pdf/PROT-86-51.pdf.

[12] Abu-Doleh AA, Al-Jarrah OM, Alkhateeb A. Protein contact map prediction using multi-stage hybrid intelligence inference systems. Journal of Biomedical Informatics. 2012 feb;45(1):173–183. Available from: https://www.sciencedirect.com/science/article/pii/S1532046411001742.

[13] Bartoli L, Capriotti E, Fariselli P, al E. The Pros and Cons of Predicting Protein Contact Maps. In: Bystroff MZ, C, editors. Methods in Molecular Biology, vol. 413: Protein Structure Prediction. 2nd ed. Totowa, NJ: Humana Press Inc.;. .

[14] Skwark MJ, Raimondi D, Michel M, Elofsson A. Improved Contact Predictions Using the Recognition of Protein Like Contact Patterns. PLoS Computational Biology. 2014 nov;10(11):e1003889. Available from: https://dx.plos.org/10.1371/journal.pcbi.1003889.

[15] Morcos F, Pagnani A, Lunt B, Bertolino A, Marks DS, Sander C, et al. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. Proceedings of the National Academy of Sciences. 2011 dec;108(49):E1293–E1301. Available from: http://www.ncbi.nlm.nih.gov/pubmed/22106262http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3241805http://www.pnas.org/cgi/doi/10.1073/pnas.1111471108.

[16] Overby CL, Tarczy-Hornoch P. Personalized medicine: challenges and opportunities for translational bioinformatics. Personalized medicine. 2013 jul;10(5):453–462. Available from: http://www.ncbi.nlm.nih.gov/pubmed/24039624http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3770190.

[17] Kuznetsov V, Lee HK, Maurer-Stroh S, Molnár MJ, Pongor S, Eisenhaber B, et al. How bioinformatics influences health informatics: Usage of biomolecular sequences, expression profiles, and automated microscopic image analyses for clinical needs and public health. In: Omics in Clinical Practice: Genomics, Pharmacogenomics, Proteomics, and Transcriptomics in Clinical Research. vol. 1. Springer; 2014. p. 203–244. Available from: http://www.ncbi.nlm.nih.gov/pubmed/25825654http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4336111.

[18] Coelho LP, Glory-Afshar E, Kangas J, Quinn S, Shariff A, Murphy RF. Principles of Bioimage Informatics: Focus on Machine Learning of Cell Patterns. Springer, Berlin, Heidelberg; 2010. p. 8–18. Available from: http://link.springer.com/10.1007/978-3-642-13131-8{_}2.

[19] Patterson J, Gibson A. Foundations of Neural Networks and Deep Learning. In: Loukides M, McGovern T, editors. Deep Learning A Practitioner's Approach. 1st ed. O'Reilly Media, Inc.; 2017. .

[20] Patterson J, Gibson A. Major Architectures of Deep Networks. In: Loukides M, McGovern T, editors. Deep Learning A Practitioner's Approach. 1st ed. O'Reilly Media, Inc.; 2017. .

[21] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015;Available from: http://arxiv.org/abs/1502.03167.

[22] Buduma N, Lacascio N. The Neural Network. In: Loukides M, Cutt S, editors. Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms. 1st ed. O'Reilly Media, Inc.; 2017. .

[23] Clevert DA, Unterthiner T, Hochreiter S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). 2015;Available from: https://arxiv.org/pdf/1511.07289.pdfhttp://arxiv.org/abs/1511.07289.

[24] Dolz J, Desrosiers C, Ben Ayed I. 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study; 2018. Available from: https://arxiv.org/pdf/1612.03925.pdf.

[25] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015;39(4):640–651. Available from: https://people.eecs.berkeley.edu/{~}jonlong/long{_}shelhamer{_}fcn.pdf.

[26] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2015;9351:234–241.

[27] Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J. Unet++: A nested u-net architecture for medical image segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2018;11045 LNCS:3–11.

[28] Bordens, Abbott. Research design and methods: A process approach .. vol. Eighth edi; 2002. Available from: http://informahealthcare.com/doi/abs/10.1080/08880010290057291{%}5Cnhttp://psycnet.apa.org/psycinfo/2001-18329-000.

[29] Bordens, Abbott. Research design and methods: A process approach .. vol. Eighth edi; 2002. Available from: http://informahealthcare.com/doi/abs/10.1080/08880010290057291{%}5Cnhttp://psycnet.apa.org/psycinfo/2001-18329-000.

[30] Johannesson P, Perjons E. An introduction to design science. vol. 9783319106328; 2014.

[31] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Res. 2000;28:235–242.

[32] Cheng H, Schaeffer RD, Liao Y, Kinch LN, Pei J, Shi S, et al. ECOD: An Evolutionary Classification of Protein Domains. PLoS Computational Biology. 2014 dec;10(12):e1003926. Available from: http://www.ncbi.nlm.nih.gov/pubmed/25474468http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4256011https://dx.plos.org/10.1371/journal.pcbi.1003926.

[33] Dunn SD, Wahl LM, Gloor GB. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. Bioinformatics. 2008 feb;24(3):333–340. Available from: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btm604.

[34] Baldassi C, Zamparo M, Feinauer C, Procaccini A, Zecchina R, Weigt M, et al. Fast and Accurate Multivariate Gaussian Modeling of Protein Families: Predicting Residue Contacts and Protein-Interaction Partners. PLoS ONE. 2014 mar;9(3):e92721. Available from: https://dx.plos.org/10.1371/journal.pone.0092721.

[35] Hurtado DM, Uziela K, Elofsson A. Deep transfer learning in the assessment of the quality of protein models;. Available from: https://arxiv.org/pdf/1804.06281.pdf.

[36] Sutskever I, Hinton G, Krizhevsky A, Salakhutdinov RR. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 2014;15:1929–1958. Available from: http://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm{_}content=buffer79b43{&}utm{_}medium=social{&}utm{_}source=twitter.com{&}utm{_}campaign=buffer.

[37] Sun KHXZSRJ. Deep Residual Learning for Image Recognition arXiv:1512.03385v1. Enzyme and Microbial Technology. 1996;19(2):107–117. Available from: http://image-net.org/challenges/LSVRC/2015/.

[38] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al.. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems; 2015. Software available from tensorflow.org. Available from: http://tensorflow.org/.

[39] Chollet F, et al.. Keras; 2015. https://keras.io.

[40] Adhikari B, Hou J, Cheng J. DNCON2: improved protein contact prediction using two-level deep convolutional neural networks. Bioinformatics (Oxford, England). 2018;34(9):1466–1472. Available from: http://www.ncbi.nlm.nih.gov/pubmed/29228185http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5925776.

[41] Kukic P, Mirabello C, Tradigo G, Walsh I, Veltri P, Pollastri G. Toward an accurate prediction of inter-residue distances in proteins using 2D recursive neural networks. BMC Bioinformatics. 2014 dec;15(1):6. Available from: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-6.

[42] Adhikari B, Bhattacharya D, Cao R, Cheng J. CONFOLD: Residue-residue contact-guided ab initio protein folding. Proteins: Structure, Function, and Bioinformatics. 2015 aug;83(8):1436–1449. Available from: http://www.ncbi.nlm.nih.gov/pubmed/25974172http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4509844http://doi.wiley.com/10.1002/prot.24829.

Appendix A

Datasets

Table A.1: PDB code and chain identification for proteins in training data set

1A1XA 1A62A 1A73A 1A76A 1AF7A 1AH7A 1ALUA 1AOCA 1AOLA 1AYOA 1B8PA 1BGCA 1BGFA 1BKRA 1BM9A 1BX4A 1BX7A 1BXYA 1BYIA 1C1KA 1C7KA 1CC8A 1CDWA 1CEOA 1CFBA 1CHDA 1CMCA 1CQYA 1CUKA 1CV8A 1CXQA 1D0QA 1D2SA 1D2TA 1D4OA 1D9CA 1DCSA 1DD3A 1DD9A 1DG6A 1DJ7A 1DK8A 1DM9A 1DMGA 1DS1A 1DUSA 1DVOA 1DXGA 1DY5A 1DZFA 1E58A 1E7LA 1EAQA 1EB6A 1EG2A 1EGWA 1EJ8A 1ELKA 1EP0A 1EYEA 1EZGA 1EZWA 1F1EA 1F32A 1F39A 1F3VA 1F86A 1F9VA 1FCQA 1FCYA 1FIPA 1FLMA 1FOBA 1FSFA 1FUKA 1FVIA 1FX2A 1FYEA 1G2RA 1G2YA 1G3PA 1G5TA 1G6XA 1G8EA 1G8QA 1GAKA 1GMXA 1GNYA 1GPRA 1GS5A 1GS9A 1GSAA 1GV9A 1GVPA 1GWMA 1H2CA 1H4XA 1H8PA 1H8UA 1H97A 1H99A 1H9MA 1HCZA 1HDOA 1HH8A 1HQ0A 1HQ1A 1HUFA 1HUWA 1HXIA 1HXNA 1HXRA 1HZ6A 1HZTA 1I1WA 1I27A 1I2KA 1I2TA 1I4JA 1I71A 1I8AA 1IAPA 1ID0A 1IFGA 1IFRA 1IGQA 1IIBA 1IJVA 1IQZA 1IRQA 1ISUA 1IZCA 1IZMA 1J0PA 1J1TA 1J27A 1J33A 1J3AA 1J5PA 1J5UA 1J5XA 1J7XA 1J8BA 1J98A 1JB3A 1JBEA 1JBZA 1JEOA 1JERA 1JF4A 1JG1A 1JHJA 1JHSA 1JI7A 1JIDA 1JL1A 1JM1A 1JNIA 1JO0A 1JOSA 1JOVA 1JR7A 1JX6A 1JYHA 1K3XA 1K4IA 1K5CA 1K77A 1K7CA 1K7JA 1K8WA 1KGDA 1KHXA 1KMTA 1KNGA 1KNMA 1KOEA 1KP6A 1KPTA 1KQ6A 1KS9A 1KT6A 1KYFA 1KZFA 1L2PA 1L3KA 1L3PA 1L6PA 1L8JA 1LC0A 1LDDA 1LFPA 1LJOA 1LKIA 1LKKA 1LMIA 1LNIA 1LS1A 1LTMA 1LUZA 1LWBA 1LXJA 1LZLA 1M2DA 1M40A 1M4LA 1M65A 1M9ZA 1MAIA 1MC2A 1MIXA 1MJ5A 1MK0A 1MKKA 1MNNA 1MSCA 1MUNA 1MY7A 1N08A 1N12A 1N1FA 1N8VA 1N9PA 1NARA 1NEPA 1NFPA 1NG6A 1NIGA 1NIJA 1NJHA 1NJRA 1NKDA 1NKOA 1NKZA 1NMYA 1NNXA 1NRGA 1NRIA 1NU0A 1NWZA 1O22A 1O4WA 1O54A 1O6DA 1O7IA 1O8XA 1OAAA 1OD3A 1ODMA 1OH0A 1OHLA 1OI7A 1OISA 1OK0A 1OKSA 1OQJA 1OU8A 1OW1A 1OZ2A 1OZ9A 1P0HA 1P3CA 1P90A 1PG6A 1PQHA 1PSRA 1PSWA 1PUJA 1PV5A 1PZ4A 1Q08A 1Q0PA 1Q2HA 1Q42A 1Q4AA 1Q4CA 1Q4EA 1Q5YA 1Q73A 1Q8CA 1Q8DA 1Q9UA 1QCSA 1QCZA 1QJ8A 1QJPA 1QNRA 1QR0A 1QSTA 1QV1A 1QW2A 1QWGA 1QZMA 1R0UA 1R29A 1R3DA 1R4XA 1R5LA 1R6DA 1R6JA 1R75A 1R7JA 1RFYA 1RG8A 1RH6A 1RI6A 1RKIA 1RL6A 1RLHA 1RMMA 1RMOA 1RO2A 1ROCA 1RSSA 1RTQA 1RTTA 1RV9A 1RXIA 1RYLA 1RYQA 1S29A 1S2OA 1S2XA 1S3CA 1S4KA 1S7KA 1S7ZA 1S9UA 1SAUA 1SBXA 1SD4A 1SDIA 1SEIA 1SENA 1SF9A 1SFPA 1SFSA 1SFUA 1SH8A 1SQHA 1SR8A 1SRAA 1SZ7A 1SZHA 1T07A 1T3YA 1T4AA 1T6SA 1T8KA 1T92A 1T95A 1T9FA 1T9IA 1TAGA 1TFEA 1TIFA 1TIGA 1TKEA 1TP6A 1TQ5A 1TQGA 1TS9A 1TT8A 1TU1A 1TUAA 1TXJA 1TXLA 1U07A 1U6TA 1U9LA 1UCDA 1UCHA 1UCRA 1UCSA 1UDVA 1UFIA 1UFYA 1UI0A 1UJ8A 1UJCA 1UKFA 1UNKA 1UNQA 1UOYA 1US0A 1USGA 1UT7A 1UUYA 1UV7A 1UX6A 1UXOA 1UZ3A 1UZKA 1V05A 1V0AA 1V2ZA 1V6PA 1V74A 1V77A 1V96A 1V9MA 1VAJA 1VBIA 1VBWA 1VCCA 1VD6A 1VGJA 1VH5A 1VHNA 1VJFA 1VJLA 1VK1A 1VK4A 1VKKA 1VL7A 1VLSA 1VLYA 1VMBA 1VMGA 1VMHA 1VP8A 1VR7A 1VR8A 1VYIA 1VYKA 1VZMA 1W0HA 1W0NA 1W1GA 1W4SA 1W53A 1WCWA 1WEHA 1WERA 1WHIA 1WHOA 1WJXA 1WKCA 1WLJA 1WMHA 1WN2A 1WNAA 1WNHA 1WOUA 1WQ6A 1WV3A 1WWCA 1WWIA 1X0TA 1X2IA 1X3KA 1X6IA 1X6OA 1X6ZA 1X8QA 1XAUA 1XBIA 1XCLA 1XDZA 1XE1A 1XJ3A 1XJVA 1XKRA 1XLQA 1XMKA 1XMTA 1XQOA 1XSVA 1XT5A 1XTPA 1XUBA 1XW3A 1XYIA 1Y08A 1Y5HA 1Y63A 1Y6XA 1Y71A 1Y80A 1Y88A 1Y8AA 1Y9LA 1YB3A 1YD0A 1YDIA 1YFQA 1YG9A 1YGTA 1YI9A 1YJFA 1YLEA 1YLIA 1YLXA 1YN4A 1YOZA 1YQ5A 1YQEA 1YQHA 1YQSA 1YRKA 1YWMA 1YZVA 1Z0NA 1Z0PA 1Z21A 1Z67A 1Z6MA 1Z6NA 1Z9LA 1ZARA 1ZCEA 1ZD0A 1ZDYA 1ZGKA 1ZHVA 1ZI8A 1ZK4A 1ZK5A 1ZLDA 1ZM8A 1ZMAA 1ZMIA 1ZO2A 1ZPSA 1ZS9A 1ZT3A 1ZUUA 1ZWXA 1ZXXA 1ZZKA 2A0BA 2A26A 2A6ZA 2A72A 2AGKA 2AH5A 2AIBA 2AJ6A 2ANXA 2AP3A 2ASBA 2ASKA 2ATZA 2AWLA 2AWMA 2AXCA 2AXOA 2AXWA 2AYDA 2B0AA 2B18A 2B1YA 2B3PA 2B4WA 2B8IA 2B8MA 2B97A 2B9DA 2BCQA 2BDRA 2BF5A 2BFWA 2BI0A 2BJFA 2BJNA 2BKAA 2BKFA 2BKMA 2BL8A 2BOPA 2BT9A 2BTIA 2BW0A 2BZ1A 2BZGA 2C2IA 2C3VA 2C60A 2C71A 2CAYA 2CB8A 2CCQA 2CCVA 2CG7A 2CHHA 2CIUA 2CIWA 2CJJA 2CKKA 2CO3A 2CPGA 2CS7A 2CULA 2CVBA 2CVEA 2CW9A 2CWSA 2CWYA 2CX7A 2CXAA 2CXHA 2CXYA 2CY5A 2CYJA 2CZSA 2D2EA 2D48A 2D4PA 2D4XA 2D59A 2D5MA 2D68A 2D7VA 2D81A 2DB7A 2DDXA 2DEJA 2DFAA 2DLBA 2DNJA 2DP9A 2DPMA 2DQAA 2DQLA 2DVKA 2DWKA 2DXAA 2DXQA 2DYIA 2E12A 2E1FA 2E2OA 2E3HA 2E56A 2E5YA 2E6MA 2E6XA 2E7VA 2E85A 2E8EA 2EAQA 2EBEA 2EBNA 2EFVA 2EH3A 2EHPA 2EI9A 2EJ9A 2EK0A 2EKLA 2ELAA 2ENDA 2ERFA 2ERVA 2ESSA 2ET1A 2ETJA 2EVRA 2EW0A 2F1FA 2F1NA 2F22A 2F23A 2F46A 2F69A 2F9HA 2FA5A 2FB6A 2FB9A 2FBQA 2FCWA 2FD4A 2FHTA 2FHZA 2FI1A 2FJ8A 2FL4A 2FM9A 2FMAA 2FOZA 2FQ4A 2FSJA 2FSQA 2FSRA 2FSUA 2FUEA 2FUJA 2FUPA 2FVVA 2FVYA 2FWHA 2FWTA 2FYGA 2FZPA 2G0CA 2G1UA 2G3RA 2G40A 2G7OA 2G7SA 2GA1A 2GAUA 2GENA 2GJ3A 2GJLA 2GKEA 2GKGA 2GKPA 2GMQA 2GNOA 2GOMA 2GPEA 2GPIA 2GQTA 2GS5A 2GSVA 2GU3A 2GUDA 2GUIA 2GUKA 2GXQA 2GZQA 2GZSA 2H0UA 2H1VA 2H30A 2H5PA 2H6EA 2H7OA 2H8EA 2H9WA 2HALA 2HBAA 2HC8A 2HE7A 2HHCA 2HHZA 2HINA 2HJEA 2HJNA 2HJOA 2HKVA 2HL7A 2HL9A 2HLJA 2HLYA 2HNGA 2HP7A 2HQ4A 2HQQA 2HQZA 2HRSA 2HS1A 2HU9A 2HUHA 2HUJA 2HW4A 2HX0A 2HX5A 2HXXA 2HYTA 2HZCA 2I02A 2I2CA 2I53A 2I5HA 2I5UA 2I6DA 2I8DA 2I8TA 2I9CA 2I9IA 2I9WA 2IA7A 2IAYA 2IC2A 2IC6A 2IDLA 2IGPA 2II2A 2IIHA 2IJQA 2IKKA 2ILKA 2ILRA 2IM9A 2IMFA 2IN3A 2INWA 2IP6A 2IRXA 2IU1A 2IUWA 2IVNA 2IXMA 2IY2A 2IYVA 2IZXA 2J0PA 2J22A 2J6AA 2J6BA 2J73A 2J8KA 2J97A 2J9WA 2JAYA 2JDCA 2JE3A 2JEKA 2JFRA 2JFUA 2JG6A 2JH1A 2JKUA 2JLIA 2MCMA 2NLRA 2NLVA 2NNUA 2NPNA 2NQ3A 2NR7A 2NRRA 2NS0A 2NSAA 2NSFA 2NSZA 2NUHA 2NUJA 2NVHA 2NWFA 2NWHA 2NX2A 2NXFA 2NZCA 2O0AA 2O0MA 2O1QA 2O24A 2O2XA 2O30A 2O38A 2O4AA 2O4DA 2O4TA 2O5HA 2O6LA 2O7AA 2O8NA 2O8PA 2O8QA 2O90A 2O9SA 2OA2A 2OAFA 2OB5A 2OBLA 2OCTA 2OD0A 2OD5A 2OEBA 2OEEA 2OF3A 2OFCA 2OFZA 2OHWA 2OIXA 2OJHA 2OKMA 2OKTA 2OKUA 2OMDA 2OMLA 2OO3A 2OOCA 2OOKA 2OPCA 2OQBA 2OQZA 2OS0A 2OSOA 2OU3A 2OU5A 2OU6A 2OV0A 2OVGA 2OVJA 2OVSA 2OXLA 2OXNA 2OXOA 2OY9A 2OYAA 2OYNA 2OYRA 2OZEA 2OZHA 2OZJA 2OZNA 2OZTA 2P08A 2P0NA 2P17A 2P2VA 2P38A 2P4FA 2P51A 2P58A 2P5DA 2P5KA 2P65A 2P67A 2P6VA 2P84A 2P9WA 2PA7A 2PAGA 2PC1A 2PEBA 2PETA 2PFIA 2PH0A 2PIEA 2PK8A 2PMAA 2PN1A 2PN2A 2PORA 2PPVA 2PQ8A 2PR7A 2PRVA 2PRXA 2PSPA 2PU3A 2PV4A 2PVBA 2PWWA 2Q12A 2Q3TA 2Q3VA 2Q4MA 2Q6KA 2Q82A 2Q8PA 2Q9KA 2Q9RA 2QDJA 2QF4A 2QFEA 2QGUA 2QHQA 2QJLA 2QJZA 2QK1A 2QKVA 2QL8A 2QLTA 2QMLA 2QMQA 2QNGA 2QNIA 2QNKA 2QNLA 2QRUA 2QSBA 2QSKA 2QSWA 2QT1A 2QTDA 2QU1A 2QUDA 2QUOA 2QUPA 2QYWA 2QZ0A 2QZQA 2R01A 2R0XA 2R16A 2R2YA 2R31A 2R4GA 2R4QA 2R9FA 2RA9A 2RBKA 2RCIA 2RDCA 2RDQA 2RE2A 2RFAA 2RFRA 2RG8A 2RH2A 2RH3A 2RHFA 2RIQA 2RJ2A 2RJIA 2RK3A 2RKHA 2RKLA 2RKQA 2SAKA 2TGIA 2UU8A 2UURA 2UV4A 2UY2A 2UYOA 2UZ8A 2V03A 2V05A 2V1MA 2V1TA 2V33A 2V3GA 2V3SA 2V75A 2V79A 2V7FA 2V7SA 2V89A 2V9BA 2V9VA 2VB1A 2VC8A 2VDFA 2VDJA 2VEZA 2VGAA 2VH3A 2VHKA 2VJWA 2VK2A 2VLAA 2VMHA 2VOVA 2VPAA 2VPBA 2VQ2A 2VVWA 2VXGA 2VXNA 2VXZA 2VY8A 2VZCA 2W0GA 2W15A 2W1RA 2W2RA 2W31A 2W39A 2W3GA 2W3QA 2W47A 2W56A 2W7AA 2W9YA 2WAOA 2WBNA 2WCRA 2WDSA 2WF7A 2WFBA 2WFIA 2WFOA 2WH7A 2WJ5A 2WJRA 2WK1A 2WL1A 2WLTA 2WLVA 2WNFA 2WP7A 2WQFA 2WQPA 2WTAA 2WTPA 2WURA 2WVIA 2WWEA 2WY4A 2WZOA 2X2UA 2X3GA 2X3MA 2X46A 2X49A 2X4LA 2X55A 2X5CA 2X5GA 2X5NA 2X5RA 2X5XA 2X6UA 2X8WA

2X9ZA 2XBGA 2XCBA 2XETA 2XF7A 2XFVA 2XGRA 2XHFA 2XIOA 2XJ4A 2XLGA 2XM5A 2XODA 2XOMA 2XPWA 2XSEA 2XSKA 2XTPA 2XU3A 2XU8A 2XVSA 2XVYA 2XY4A 2XZ2A 2XZ7A 2Y0GA 2Y0OA 2Y1BA 2Y2ZA 2Y39A 2Y43A 2Y4XA 2Y5PA 2Y6CA 2Y6XA 2Y78A 2Y8GA 2Y8YA 2Y9UA 2YG2A 2YGOA 2YH9A 2YHCA 2YJMA 2YK4A 2YMVA 2YN0A 2YV4A 2YVQA 2YVTA 2YWJA 2YWWA 2YXFA 2YYOA 2YZTA 2YZYA 2Z0XA 2Z14A 2Z1CA 2Z2BA 2Z2NA 2Z51A 2Z5BA 2Z5EA 2Z5WA 2Z72A 2Z84A 2Z8LA 2Z8PA 2Z98A 2ZAYA 2ZCAA 2ZCUA 2ZCWA 2ZFGA 2ZGLA 2ZHJA 2ZOUA 2ZOVA 2ZPMA 2ZQOA 2ZVCA 2ZZJA 3A02A 3A0NA 3A0SA 3A0YA 3A1GA 3A27A 3A2ZA 3A4CA 3A4JA 3A57A 3A9FA 3AAFA 3ACHA 3ADOA 3ADYA 3AEIA 3AGNA 3AJ4A 3AJDA 3AKSA 3B09A 3B1VA 3B33A 3B42A 3B49A 3B4QA 3B6EA 3B79A 3B7CA 3B7HA 3BA3A 3BBYA 3BCYA 3BDIA 3BEDA 3BFMA 3BHNA 3BI7A 3BJNA 3BK5A 3BLEA 3BN0A 3BOEA 3BPJA 3BQAA 3BS4A 3BUTA 3BV4A 3BV8A 3BWZA 3BY8A 3C0SA 3C1QA 3C26A 3C2EA 3C4BA 3C5VA 3C6AA 3C70A 3C7XA 3C8IA 3C8LA 3C8MA 3C8XA 3C9PA 3CANA 3CB9A 3CBNA 3CBZA 3CCDA 3CD1A 3CECA 3CEXA 3CG6A 3CHJA 3CHMA 3CHVA 3CI3A 3CIMA 3CJEA 3CKKA 3CKMA 3CLAA 3CP0A 3CP3A 3CQBA 3CT5A 3CT6A 3CZ6A 3D06A 3D0FA 3D0JA 3D1PA 3D2YA 3D32A 3D33A 3D3BA 3D3MA 3D4EA 3D79A 3D7IA 3D9NA 3DA5A 3DASA 3DCZA 3DD6A 3DD7A 3DDJA 3DDTA 3DEFA 3DEWA 3DF7A 3DF8A 3DFFA 3DFGA 3DGTA 3DKMA 3DKRA 3DLCA 3DMNA 3DNJA 3DNPA 3DQYA 3DSMA 3DT5A 3DXEA 3DXYA 3DZ1A 3E0HA 3E3XA 3E8TA 3E99A 3E9AA 3E9VA 3EA6A 3EDHA 3EEAA 3EERA 3EF8A 3EGAA 3EHGA 3EIPA 3EJCA 3EJFA 3EJVA 3ELFA 3EMFA 3EMIA 3ENUA 3EO6A 3EOIA 3ER7A 3ESMA 3EUNA 3EURA 3EUSA 3EVFA 3EVPA 3EXMA 3EYEA 3EZHA 3F14A 3F2EA 3F2ZA 3F43A 3F4MA 3F5BA 3F5HA 3F5RA 3F6GA 3F7EA 3F9SA 3FAJA 3FB9A 3FBLA 3FBUA 3FCNA 3FDQA 3FF0A 3FF2A 3FGHA 3FGYA 3FHGA 3FILA 3FKEA 3FL2A 3FLBA 3FLJA 3FMYA 3FN2A 3FNCA 3FNDA 3FPNA 3FPRA 3FPWA 3FRHA 3FRRA 3FSOA 3FSSA 3FT7A 3FTDA 3FTJA 3FX7A 3FXHA 3FYMA 3FYNA 3FYRA 3FZEA 3G16A 3G1JA 3G2BA 3G36A 3G3TA 3G46A 3G5TA 3G7PA 3G85A 3G91A 3G98A 3G9RA 3GA4A 3GA8A 3GBWA 3GBYA 3GDWA 3GGMA 3GHAA 3GI7A 3GJYA 3GK6A 3GKJA 3GMGA 3GMXA 3GOEA 3GOHA 3GONA 3GP6A 3GPIA 3GPVA 3GQJA 3GRDA 3GS2A 3GS9A 3GT0A 3GVAA 3GWIA 3GXHA 3GZRA 3H0NA 3H1ZA 3H20A 3H36A 3H4OA 3H4XA 3H51A 3H5JA 3H6QA 3H74A 3H75A 3H79A 3H7IA 3HA9A 3HBMA 3HC7A 3HFTA 3HJZA 3HM4A 3HMSA 3HP7A 3HPDA 3HRGA 3HSHA 3HTVA 3HVWA 3HWUA 3HYNA 3HZ7A 3HZ8A 3HZPA 3I0WA 3I10A 3I6IA 3I7MA 3IB5A 3IB7A 3IBWA 3ICVA 3IDUA 3IE4A 3IEEA 3IEZA 3IG9A 3IHTA 3II2A 3IIDA 3ILSA 3IM1A 3IMKA 3INOA 3IO3A 3IP0A 3IPFA 3IPJA 3IQUA 3IR4A 3IRBA 3ISXA 3ITFA 3IUOA 3IUWA 3IV3A 3IV4A 3IVVA 3IWFA 3IX3A 3IXLA 3JSRA 3JTZA 3JU3A 3JUDA 3JX9A 3JXOA 3JYOA 3JZ9A 3K0ZA 3K1HA 3K1UA 3K29A 3K3VA 3K5JA 3K67A 3K69A 3K8UA 3KA5A 3KBGA 3KBQA 3KBYA 3KDWA 3KE7A 3KEVA 3KF6A 3KFFA 3KFOA 3KGKA 3KHIA 3KJHA 3KKBA 3KKGA 3KLQA 3KOGA 3KOJA 3KSNA 3KUVA 3KVHA 3KWRA 3KXTA 3KYZA 3L00A 3L0FA 3L1NA 3L23A 3L2CA 3L32A 3L39A 3L51A 3L60A 3L6BA 3L7YA 3L81A 3L8WA 3LAGA 3LAXA 3LAZA 3LB2A 3LD7A 3LFRA 3LHCA 3LHEA 3LHIA 3LHNA 3LI9A 3LLOA 3LLUA 3LMZA 3LO8A 3LPZA 3LQ9A 3LQNA 3LRUA 3LTIA 3LUYA 3LW3A 3LWAA 3LWXA 3LX3A 3LX7A 3LXYA 3LYDA 3LYEA 3LYGA 3LYHA 3LYWA 3LYYA 3LZWA 3M0BA 3M1XA 3M3PA 3M4IA 3M66A 3M7AA 3M7KA 3M86A 3M8JA 3M9QA 3MABA 3MAGA 3MAOA 3MC3A 3MCBA 3MCQA 3MDQA 3ME7A 3MHXA 3ML3A 3MMHA 3MQQA 3MQZA 3MR0A 3MSWA 3MTQA 3MTXA 3MVCA 3MVNA 3MVSA 3MW8A 3MWZA 3MX7A 3MXNA 3MXZA 3MZ0A 3N01A 3N08A 3N0RA 3N17A 3N1EA 3N2TA 3N4JA 3N6YA 3N72A 3NA6A 3NBCA 3NBMA 3NDQA 3NE0A 3NFTA 3NH4A 3NJCA 3NJNA 3NKGA 3NKLA 3NL9A 3NO2A 3NO7A 3NOJA 3NR5A 3NRFA 3NRLA 3NRWA 3NSOA 3NTKA 3NUFA 3NUQA 3NYMA 3O0LA 3O0PA 3O12A 3O15A 3O48A 3O4PA 3O6CA 3O7BA 3O7WA 3OBLA 3OCJA 3OCUA 3OEBA 3OHGA 3OIOA 3OIZA 3OJ0A 3OKXA 3OL3A 3OMDA 3OMYA 3ON9A 3ONHA 3ONJA 3OOSA 3OOUA 3ORUA 3OSDA 3OSEA 3OSTA 3OV5A 3OXPA 3P02A 3P0KA 3P2TA 3P4EA 3P4HA 3PE6A 3PESA 3PF6A 3PFGA 3PFTA 3PI7A 3PIWA 3PLUA 3PMSA 3POJA 3POWA 3PP2A 3PSHA 3PT3A 3PT5A 3PUCA 3PVHA 3PVVA 3PYFA 3PYWA 3Q13A 3Q1NA 3Q1XA 3Q20A 3Q2UA 3Q46A 3Q6BA 3Q7RA 3QAGA 3QC0A 3QC7A 3QDHA 3QH6A 3QHPA 3QL9A 3QLEA 3QM9A 3QP4A 3QPAA 3QR7A 3QWLA 3QX1A 3QY7A 3QZBA 3QZMA 3QZXA 3R15A 3R2QA 3R5TA 3R6DA 3R87A 3RAYA 3RC1A 3RFNA 3RJVA 3RKCA 3RKGA 3RL5A 3RLGA 3RLKA 3RLOA 3RLSA 3RM3A 3RMHA 3RMQA 3RN4A 3RNLA 3RNVA 3RO3A 3ROFA 3RPJA 3RPZA 3RQ4A 3RR6A 3RVCA 3RX9A 3S25A 3S2RA 3S4EA 3S6EA 3S6FA 3S83A 3S8IA 3S8SA 3S9XA 3SBMA 3SD2A 3SIBA 3SIGA 3SJMA 3SK7A 3SMVA 3SMZA 3SNOA 3SNZA 3SO6A 3SOJA 3SOVA 3SU6A 3SXMA 3SXUA 3SY1A 3T0HA 3T47A 3T4RA 3T7AA 3T7LA 3T7ZA 3T8JA 3T92A 3T9YA 3TBDA 3TBNA 3TE8A 3TEEA 3TEUA 3TG2A 3TJ8A 3TJMA 3TJYA 3TNTA 3TOWA 3TR0A 3TS9A 3TT9A 3TU8A 3TYSA 3TZ6A 3U0VA 3U1DA 3U1UA 3U25A 3U2AA 3U4GA 3U4VA 3U5SA 3U5VA 3U6GA 3U7ZA 3U81A 3U8VA 3U97A 3U9JA 3UAWA 3UC9A 3UEJA 3UFEA 3UFZA 3UI4A 3UJCA 3ULBA 3ULJA 3ULTA 3UMHA 3UMZA 3UP3A 3URGA 3URRA 3USHA 3UUEA 3UV0A 3UX2A 3V0EA 3V1EA 3V46A 3V68A 3V7BA 3VBCA 3VDJA 3VEJA 3VGIA 3VGPA 3VHJA 3VMVA 3VN5A 3VNEA 3VORA 3VP5A 3VPZA 3VQJA 3VTTA 3VUBA 3VURA 3VVVA 3VWCA 3VXVA 3VZHA 3W06A 3W07A 3W0EA 3W1OA 3W2ZA 3W43A 3WCQA 3WDCA 3WDNA 3WGXA 3WH1A 3WH2A 3WHJA 3WI9A 3WITA 3WJPA 3WJTA 3WMVA 3WS7A 3WURA 3WVAA 3WW9A 3WX4A 3WZ3A 3WZSA 3X0FA 3X0IA 3X0TA 3X2MA 3X34A 3ZBDA 3ZDBA 3ZFIA 3ZGHA 3ZH5A 3ZHIA 3ZJAA 3ZN3A 3ZN4A 3ZNVA 3ZRGA 3ZRXA 3ZSJA 3ZSUA 3ZUCA 3ZUDA 3ZUIA 3ZXCA 3ZXNA 3ZY7A 3ZYHA 3ZYPA 3ZZLA 3ZZOA 3ZZPA 3ZZYA 4A02A 4A29A 4A2VA 4A3PA 4A3ZA 4A4JA 4A56A 4A5UA 4A61A 4A6QA 4A7UA 4A8TA 4ABLA 4ACJA 4ADZA 4AE7A 4AEQA 4AFFA 4AFMA 4AIVA 4AIWA 4AL5A 4ALZA 4AQOA 4ATEA 4ATGA 4ATMA 4AU1A 4AVAA 4AXOA 4AXQA 4B1LA 4B21A 4B2FA 4B4CA 4B4DA 4B5OA 4B89A 4B8VA 4B8XA 4B9GA 4BFCA 4BFOA 4BGCA 4BGPA 4BHRA 4BI3A 4BJ0A 4BJAA 4BJIA 4BK0A 4BN4A 4BOQA 4BOUA 4BPFA 4BPSA 4BQYA 4BT7A 4BTBA 4BWOA 4BYZA 4BZAA 4BZPA 4C1WA 4C5PA 4C6AA 4C6SA 4C81A 4C84A 4C98A 4CBEA 4CC2A 4CCVA 4CCWA 4CD8A 4CFIA 4CG1A 4CHEA 4CICA 4CMFA 4CO8A 4CV7A 4CZ5A 4CZGA 4D01A 4D0QA 4D53A 4D5RA 4D6QA 4D8BA 4DB5A 4DB6A 4DDPA 4DE9A 4DI9A 4DJGA 4DK2A 4DLMA 4DNYA 4DOLA 4DQ9A 4DQJA 4DQNA 4DT5A 4DVCA 4DYQA 4DZOA 4E0AA 4E14A 4E1PA 4E1SA 4E40A 4E4RA 4EA9A 4EBGA 4EFOA 4EG9A 4EHCA 4EHSA 4EHXA 4EICA 4EKFA 4EL6A 4EO0A 4EP4A 4EPZA 4EQPA 4ERCA 4ERNA 4ERRA 4ERYA 4ES1A 4ES7A 4ESMA 4ESQA 4ETXA 4EULA 4EUNA 4EV1A 4EVUA 4EW5A 4EW7A 4EXOA 4EXRA 4EYCA 4EYSA 4EZ8A 4F0WA 4F2EA 4F54A 4F55A 4F67A 4F80A 4F87A 4F98A 4FBSA 4FD9A 4FDBA 4FE3A 4FF5A 4FGOA 4FH3A 4FK9A 4FLBA 4FTFA 4FUUA 4FVGA 4FXIA 4FZPA 4G08A 4G0XA 4G29A 4G3NA 4G3VA 4G4KA 4G54A 4G5AA 4G75A 4G78A 4G7XA 4G9QA 4G9SA 4GA2A 4GB5A 4GBMA 4GC3A 4GCNA 4GDZA 4GEIA 4GIOA 4GJZA 4GMQA 4GOFA 4GS3A 4GT8A 4GT9A 4GUCA 4GWBA 4GZCA 4GZJA 4H08A 4H14A 4H3UA 4H4NA 4H7WA 4H86A 4H8EA 4HCJA 4HCSA 4HDDA 4HE6A 4HEIA 4HEOA 4HFVA 4HHXA 4HI8A 4HIKA 4HJIA 4HKGA 4HLSA 4HLYA 4HNOA 4HP4A 4HROA 4HRVA 4HS1A 4HS2A 4HTGA 4HTLA 4HTUA 4HU2A 4HVYA 4HWFA 4HWMA 4HY4A 4HYLA 4HYQA 4HZOA 4I1FA 4I1KA 4I4OA 4I66A 4I6RA 4I6XA 4I71A 4I8HA 4I8IA 4I90A 4I95A 4I9OA 4IAUA 4IC3A 4IC9A 4IEJA 4IFAA 4IGIA 4II9A 4IIKA 4IILA 4IKCA 4IKGA 4IL7A 4IN0A 4IPVA 4IRFA 4IT6A 4IUSA 4IX7A 4J32A 4J37A 4J39A 4J42A 4J4RA 4J5RA 4J8AA 4J8CA 4J8SA 4JB3A 4JBDA 4JCCA 4JCYA 4JDEA 4JDUA 4JEJA 4JF5A 4JF8A 4JG2A 4JGLA 4JHTA 4JJ2A 4JMPA 4JNFA 4JNUA 4JP6A 4JQFA 4JTMA 4JXHA 4K0DA 4K0NA 4K4OA 4K7BA 4K82A 4K8WA 4KA9A 4KAGA 4KDDA 4KDRA 4KDWA 4KEFA 4KEXA 4KH8A 4KIIA 4KM6A 4KRUA 4KSNA 4KSYA 4KT3A 4KTWA 4KUIA 4KV2A 4KW4A 4KZKA 4L0JA 4L1IA 4L3UA 4L8AA 4L9EA 4L9NA 4LBAA 4LD1A 4LEBA 4LF0A 4LGJA 4LHSA 4LJ6A 4LLDA 4LMYA 4LPQA 4LQTA 4LQZA 4LRUA 4LTTA 4LUAA 4LUPA 4LWSA 4LWUA 4LZHA 4M0NA 4M5BA 4M7TA 4M8AA 4M91A 4M9KA 4MAIA 4MAKA 4MDAA 4MDWA 4MDYA 4ME2A 4MF5A 4MFKA 4MGQA 4MGSA 4MI7A 4MISA 4MJDA 4MJFA 4MLOA 4MNNA 4MNOA 4MP8A 4MT2A 4MT8A 4MTMA 4MTUA 4MUMA 4MVKA 4MXTA 4MYKA 4MZCA 4MZJA 4N1IA 4N30A 4N67A 4N6QA 4N77A 4N7CA 4N7FA 4NC6A 4NC7A 4NDJA 4NDSA 4NE3A 4NGDA 4NI6A 4NL9A 4NMIA 4NNOA 4NOAA 4NOHA 4NS5A 4NTDA 4NWBA 4NYQA 4O0AA 4O0KA 4O1TA 4O65A 4O66A 4O6GA 4O6UA 4O7JA 4O7KA 4OA3A 4OD6A 4OE9A 4OFAA 4OIEA 4OKEA 4ONRA 4OOXA 4OQ8A 4OQPA 4OSNA 4OTMA 4OTNA 4OUNA 4OUSA 4OVYA 4OY3A 4P0MA 4P1MA 4P3AA 4P3VA 4P5NA 4P7XA 4P98A 4P9IA 4PAUA 4PBDA 4PBOA 4PDNA 4PF3A 4PGRA 4PH2A 4PH8A 4PHJA 4PI8A 4PKMA 4PLZA 4PNOA 4PQDA 4PQQA 4PS2A 4PS6A 4PSFA 4PSYA 4PUIA 4PWWA 4PWYA 4PXVA 4PZ0A 4PZ1A 4PZ3A 4Q0PA 4Q2LA 4Q2SA 4Q34A 4Q53A 4Q63A 4Q6VA 4Q7OA 4QA8A 4QB0A 4QEKA 4QHJA 4QM6A 4QMAA 4QMHA 4QNDA 4QP5A 4QPNA 4QPTA 4QRKA 4QTQA 4QUKA 4QXLA 4QY7A 4R03A 4R1BA 4R1JA 4R2HA 4R3QA 4R5RA 4R7QA 4R9PA 4RAXA 4RAYA 4RBRA 4RD7A 4RDBA 4REOA 4RGDA 4RGIA 4RK4A 4RMLA 4RNAA 4RO3A 4RP3A 4RPTA 4RT1A 4RTIA 4RU3A 4RUVA 4RVQA 4RWUA 4RXIA 4RXVA 4RYOA 4RZ9A 4S1PA 4S24A 4TJVA 4TKBA 4TMDA 4TQ1A 4TQ2A 4TQRA 4TTWA 4TXRA 4TYZA 4U12A 4U3VA 4U6NA 4U6OA 4U89A 4U9BA 4UDSA 4UE8A 4UHHA 4UHQA 4UHTA 4UIOA 4UJ7A 4UMGA 4UNUA 4UOBA 4UONA 4UOSA 4UQWA 4UQXA 4UU3A 4UVQA 4UY5A 4UYIA 4UYRA 4V0KA 4V0SA 4V17A 4V1GA 4V1JA 4V1KA 4V23A 4V3IA 4W5XA 4W6TA 4W79A 4W7WA 4W8HA 4W8PA 4W8QA 4W97A 4W9ZA 4WBJA 4WCJA 4WDCA 4WE2A 4WEEA 4WH9A 4WHEA 4WHIA 4WIQA 4WJIA 4WJQA 4WJTA 4WN5A 4WNBA 4WOLA 4WPGA 4WPKA 4WPYA 4WRIA 4WSFA 4WT3A 4WTPA 4WUIA 4WUTA 4WV4A 4WWBA 4WWFA 4WY4A 4WY9A 4WYHA 4WZ0A 4WZXA 4X2RA 4X3IA 4X5MA 4X5PA 4X7GA 4X8QA 4X8YA 4X9RA 4X9XA 4X9ZA 4XABA 4XALA 4XB4A 4XCVA 4XDUA 4XDXA 4XEHA 4XEPA 4XEZA 4XFSA 4XINA 4XKBA 4XKZA 4XO1A 4XPCA 4XPLA 4XPXA 4XT6A 4XTBA 4XU4A 4XUOA 4XUWA 4XVVA 4XXXA 4XY5A 4XZFA 4Y6WA 4Y88A 4Y9IA 4YAAA 4YBGA 4YE7A 4YG0A 4YI8A 4YMYA 4YNHA 4YNXA 4YQDA 4YSIA 4YTKA 4YTLA 4YTVA 4YU8A 4YUCA 4YWZA 4YX1A 4YZ6A 4Z2NA 4Z3HA 4Z47A 4Z85A 4ZAVA 4ZBHA 4ZC3A 4ZCEA 4ZDSA 4ZF7A 4ZGFA 4ZGMA 4ZHWA 4ZJ9A 4ZJHA 4ZJUA 4ZLDA 4ZMKA 4ZNKA 4ZOTA 4ZQXA 4ZV5A 4ZVAA 4ZVCA 4ZVFA 4ZX2A 4ZY9A 4ZZ1A 5A0NA 5A1QA 5A3AA 5A3DA 5A4AA 5A67A 5A6WA 5A99A 5A9AA 5AE0A 5AG8A 5AGIA 5AGRA 5AIGA 5AIMA 5AIZA 5AJGA 5AJJA 5AMHA 5AOYA 5AZBA 5BOIA 5BP9A 5BPXA 5BTYA 5BVLA 5BXGA 5BY4A 5BY5A 5BY8A 5BYKA 5C12A 5C17A 5C1EA 5C2MA 5C30A 5C5GA 5C5ZA 5C6SA 5CE7A 5CKLA 5CL8A 5COFA 5COTA 5COWA 5CR9A 5CVWA 5CWGA 5CXUA 5CYVA 5D2FA 5D3KA 5D3XA 5D7UA 5DAWA 5DFSA 5DGJA 5DHDA 5DJOA 5DLBA 5DM2A 5DMMA 5DOMA 5DP2A 5DPGA 5DPOA 5DTCA 5DTXA 5E0LA 5E2CA 5E3QA 5E50A 5E5YA 5E6XA 5E9PA 5EC6A 5ECDA 5EIPA 5EJUA 5EKYA 5EMIA 5EP2A 5EPEA 5EQ0A 5EQVA 5EQZA 5EWOA 5EYRA 5EZQA 5F18A 5F47A 5F4CA 5F6RA 5FD9A 5FF3A 5FMTA 5FOTA 5FPZA

5FSVA 5HB7A 5HD9A 5HDWA 5HJ1A 5HJ9A 5HQHA 5HQTA 5HWVA 5HYAA 5I45A 5I9PA 5ICUA 5IDVA 5IG0A 5IGCA 5IGEA 5IGFA 5IIFA 5IO9A 5IOCA 5IODA 5IT3A 5IUCA 5J1ZA 5J2OA 5J3QA 5J8EA 5JBRA 5JE5A 5JJXA 5NULA 7A3HA

Table A.2: PDB code and chain identification for proteins in validation data set

1H97A 1J5XA 1N12A 1NFPA 1ODMA 1OZ9A 1QJPA 1SENA 1TQGA 1U6TA 1WLJA 1XYIA 1ZI8A 2CAYA 2HE7A 2HNGA 2HQQA 2IIHA 2NQ3A 2OA2A 2OO3A 2OSOA 2P58A 2P5DA 2PSPA 2RH2A 2RIQA 2W3GA 2WY4A 2YGOA 2Z8PA 2ZHJA 3A4CA 3D06A 3D1PA 3DT5A 3EJVA 3FILA 3FMYA 3G5TA 3GPIA 3H5JA 3H7IA 3HRGA 3INOA 3JU3A 3LO8A 3LZWA 3MC3A 3MMHA 3N6YA 3QPAA 3U4VA 3U8VA 3VGPA 3ZRXA 3ZZLA 4A02A 4A4JA 4BOQA 4CCVA 4ETXA 4EUNA 4F2EA 4HDDA 4I71A 4I95A 4JG2A 4K0NA 4KEXA 4L3UA 4N7CA 4NDSA 4ONRA 4Q2SA 4Q7OA 4RO3A 4TKBA 4V17A 4V3IA 4WHIA 4WJQA 4WNBA 4XKZA 4XVVA 4XXXA 4YBGA 4YE7A 4YTLA 4ZX2A 5A9AA 5AE0A 5C17A 5E0LA 5E50A 5IOCA 5J2OA

Table A.3: PDB code and chain identification for proteins in testing data set

1AHSC 1C2YD 1C9YA 1CCTA 1COZA 1DBRA 1DCHF 1EDIA 1EFDN 1F46B 1F68A 1FHIA 1FJRB 1FS0G 1G61A 1GJJA 1GLGA 1GPSA 1H68A 1I95E 1I97T 1IMBB 1IMXA 1IR1S 1IS9A 1JGPR 1JH0L 1K6LH 1KNVB 1KNYB 1KQPA 1LDIA 1LQKB 1M12A 1MB6A 1MFRP 1MR7A 1N2ZB 1N5BA 1N60C 1NQLB 1OAGA 1OTFF 1P3HE 1PCFA 1PDFE 1PS1A 1RD9D 1RH7C 1RL9A 1S3FB 1S68A 1SUDA 1SWXA 1SYHA 1TD4A 1TFKB 1TJLD 1UWZB 1VCRA 1VJNA 1VQZA 1W8AA 1W9GB 1WD5A 1WIGA 1WPVB 1X0PJ 1X48A 1X8HA 1X91A 1XBAA 1XQFA 1XS6A 1Y4HD 1Y60C 1YG6F 1YHQO 1YQFF 1YWSA 1Z7ME 1ZD7B 1ZJ0A 1ZWYC 2A84A 2A9KB 2AMCA 2AV5D 2B9NX 2BWEL 2C2OA 2CB6A 2CCCA 2CDMC 2CJRB 2CSMA 2D0PB 2D2CN 2DIOC 2E2AB 2EJNA 2F0RA 2FEEB 2FJCO 2GVIA 2H44A 2HGHA 2HI7B 2HJJA 2HL0A 2I9LI 2IA9E 2II9B 2J1KQ 2J3WA 2J8WB 2JOVA 2JYNA 2KYSA 2KZSA 2M0MB 2NQ2A 2NR9A 2OF5H 2OGFD 2OHCA 2OJ5C 2ONKC 2OPIA 2PAVP 2PLSF 2Q7RA 2QQDE 2QYFD 2RDOL 2RMRA 2RTBB 2VGRA 2VT8A 2WNKA 2WNYA 2XVTF 2Y9PB 2YADB 2YZOA 2ZITD 3A1JB 3ANZW 3AXGI 3B2UB 3B71B 3B7AA 3BLAB 3BP9B 3CPWT 3CVZC 3CXJC 3D2QD 3DBYF 3DKXB 3EB6B 3EW1A 3G74B 3GUVA 3GYVA 3GZFC 3H8DB 3H90A 3HPGL 3HTYJ 3I9OB 3IQZF 3K43B 3K8RB 3KZLA 3LW5L 3M71A 3MEZA 3N1GA 3NJSA 3O7JA 3OFEB 3OQIA 3P45J 3PC7B 3PJZA 3QE7A 3QNQA 3RBYB 3T3TB 3UD2B 3UWSA 3UYUB 3V3LB 3VHHB 3VX6A 3ZNUG 3ZUXA 4A5ZB 4AI3A 4ARDB 4AU0B 4DLHB 4E1YB 4E6FA 4F0DA 4HBRC 4IOSH 4IZJC 4J32B

Appendix B

Extended results

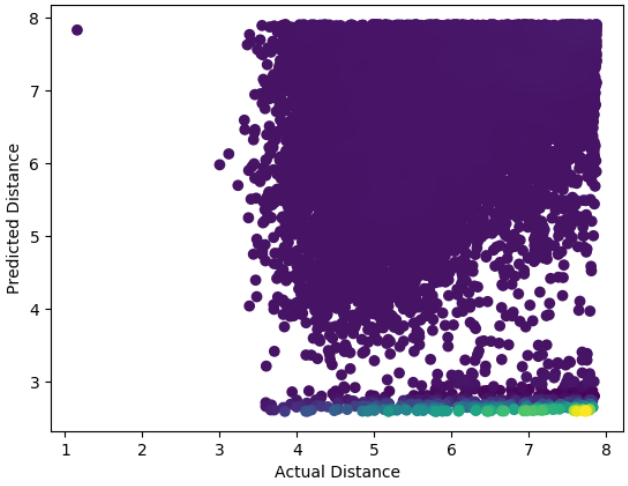


Figure B.1: Predicted Distance < 8 Å vs Actual Distance < 8 Å for Classification Model with 12 bins for all contacts in the protein.

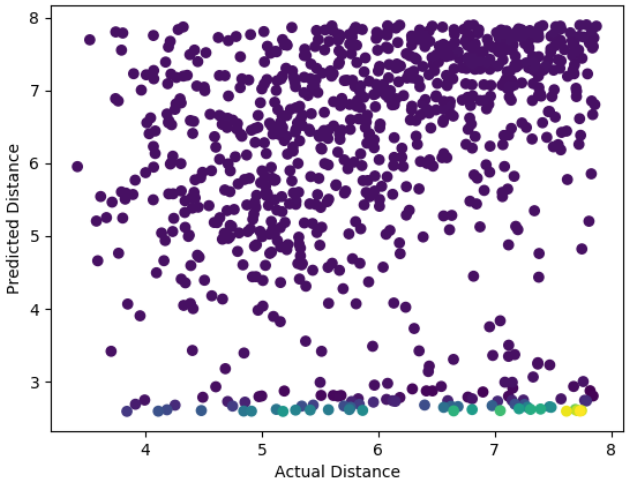


Figure B.2: Predicted Distance < 8 Å vs Actual Distance < 8 Å for Classification Model with 12 bins for top L contacts in the protein.

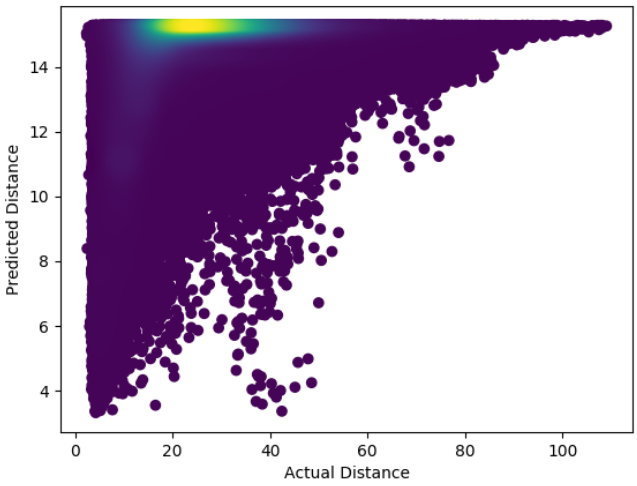


Figure B.3: All the predicted distances vs all the actual distances for Classification Model with 12 bins for all contacts in the protein.

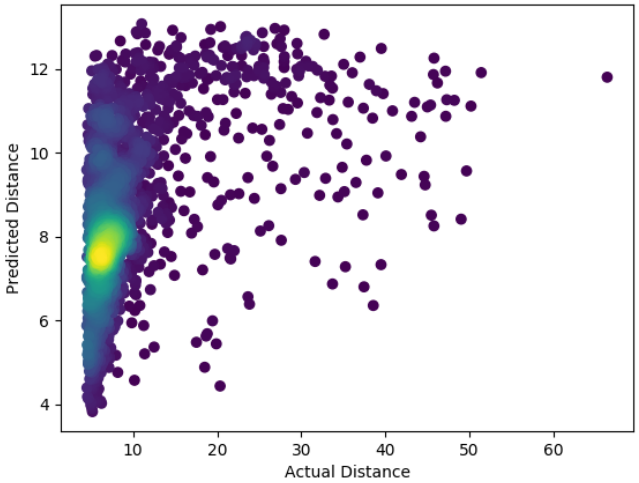


Figure B.4: All the predicted distances vs all the actual distances for Classification Model with 12 bins for top L contacts in the protein.

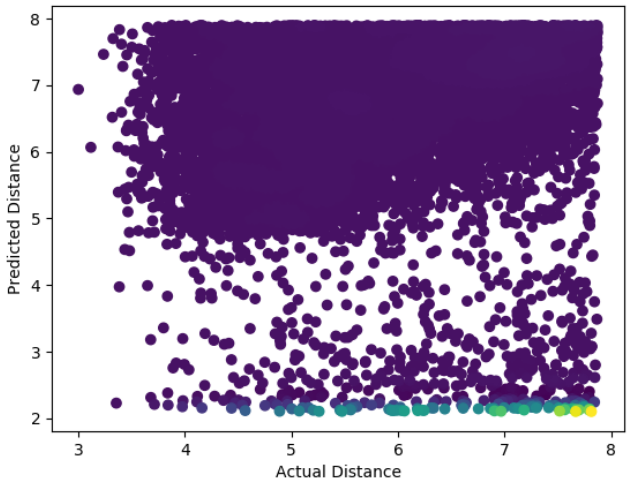


Figure B.5: Predicted Distance < 8 Å vs Actual Distance < 8 Å for Classification Model with 26 bins for all contacts in the protein.

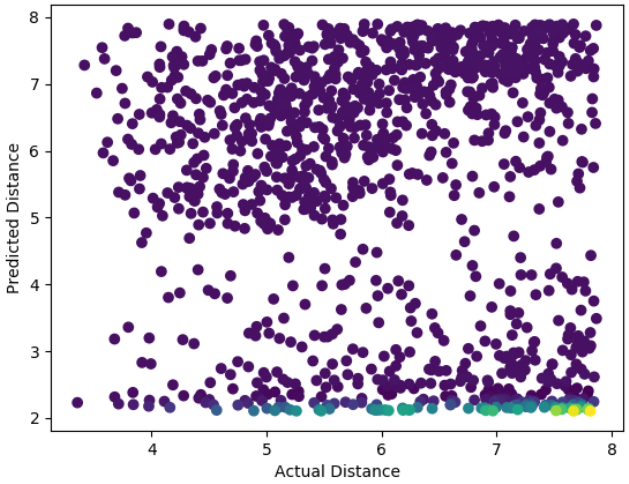


Figure B.6: Predicted Distance < 8 Å vs Actual Distance < 8 Å for Classification Model with 26 bins for top L contacts in the protein.

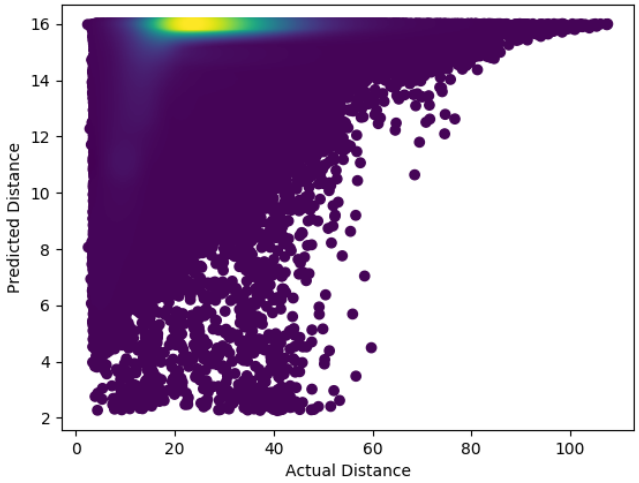


Figure B.7: All the predicted distances vs all the actual distances for Classification Model with 26 bins for all contacts in the protein.

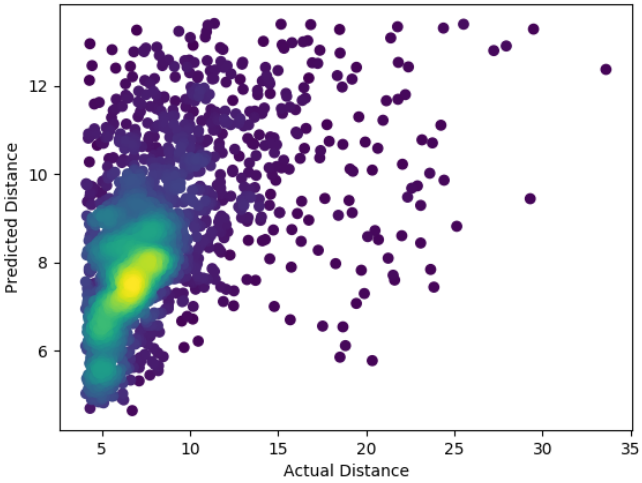


Figure B.8: All the predicted distances vs all the actual distances for Classification Model with 26 bins for top L contacts in the protein.

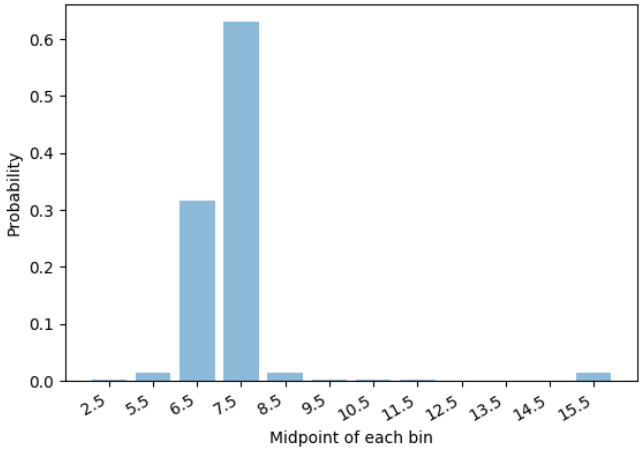


Figure B.9: Probability distribution of correctly predicted contact (Actual value = 7.181, Predicted value =7.295))

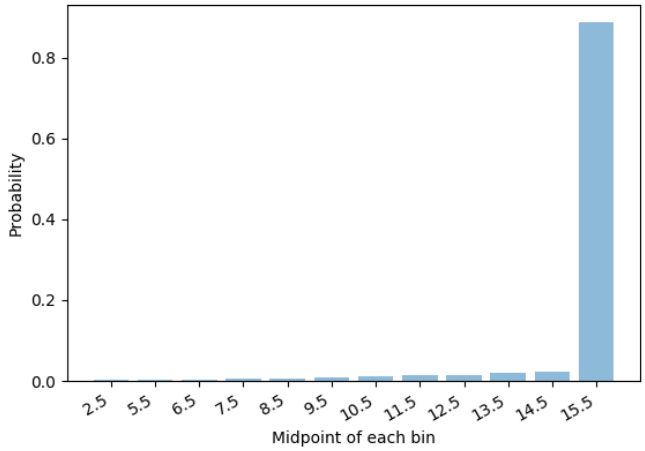
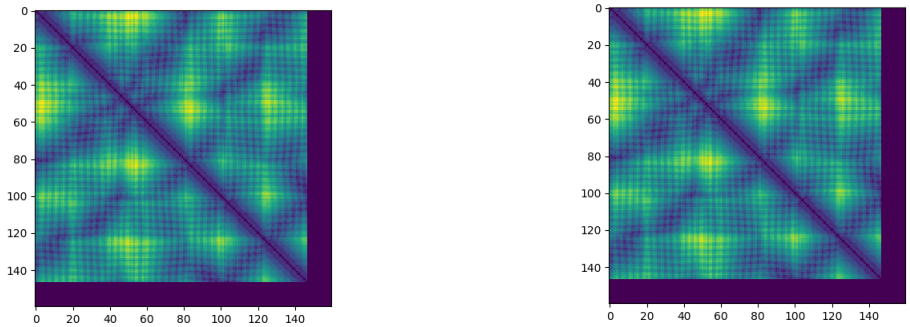


Figure B.10: Probability distribution of incorrectly predicted contact (Actual Value = 4.441, Predicted Value = 15.039)



(a) 12 bins

(b) 26 bins

Figure B.11: Distance maps for Protein 1H97A generated by distance classification models.

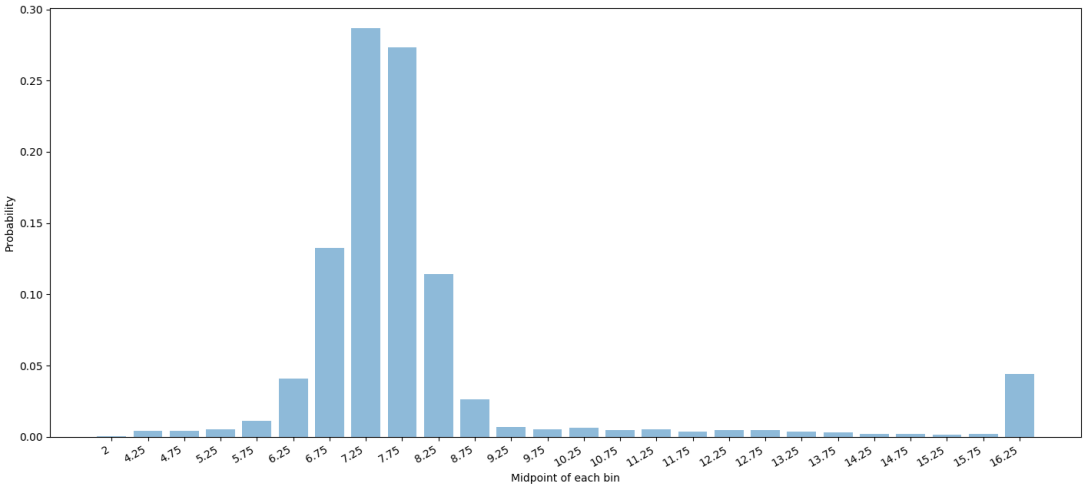


Figure B.12: Probability distribution of correctly predicted contact (Actual value = 7.012, Predicted value = 8.028)

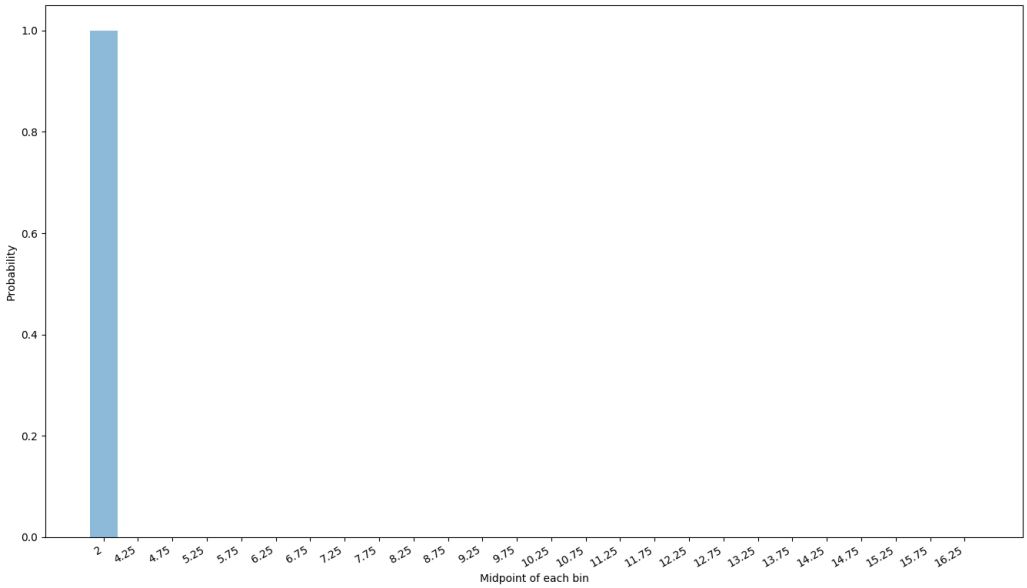


Figure B.13: Probability distribution of incorrectly predicted contact (Actual Value = 7.643, Predicted Value = 2.000)