University of Reading

Development of consensus contact prediction methods for the improvement of protein 3D model quality estimates

A thesis submitted for the degree of

Doctor of Philosophy

School of Biological Sciences

University of Reading

Shuaa Muslih Awad Alharbi

November 2023

Declaration

I confirm that this is my own work and to the best of my knowledge, does not breach copyright law, and has not been taken from other sources except where such work has been cited and acknowledged within the text.

Shuaa Muslih Awad Alharbi:

Date: 22 /11/2023

Abstract

Proteins play a crucial role in the biological machinery of living organisms, with their structures dictating functions essential for life processes. Disruptions in protein function lead to diseases. Therefore, knowledge of proteins is vital for biomedical sciences and biotechnology. Protein structures are experimentally determined by NMR or X-ray crystallography, but computational methods have gained prominence due to their speed and accuracy. Recent advances in protein structure prediction provide high-accuracy 3D models, challenging quality estimation methods. Predicting residue contacts can be useful in obtaining significant information that may be used to improve the performance of quality estimation methods. Contact prediction methods have evolved, utilising diverse protein databases and approaches to enhance 3D protein model accuracy. However, challenges persist in modelling certain targets. This study proposes consensus approaches, combining data from deep learning-based contact prediction methods from CASP13 and CASP14, leading to measurable advancements in accuracy.

We then investigated the role of consensus contact prediction in improving the performance of ModFOLD9 using the CDA score. The experiment expanded to integrate various quality scores derived from the pure-single model and quasi-single model methods to further enhance ModFOLD9's accuracy. The consensus algorithms and contact prediction improved ModFOLD9's local quality estimations for tertiary structure models. This strategy was extended to enhance the IntFOLD7 and ModFOLDdockS servers. We analysed the performance of the improved servers using two gold-standard blind experiments: CAMEO and CASP15. The evaluation of these servers validated their improved performance and highlighted the impact of contact prediction on enhancing both local tertiary structure model quality estimations and quaternary structure model quality estimates for interface residues. Overall, our study demonstrated the importance of contact prediction in improving the performance of model quality estimation tools in the field of protein structure prediction.

Contents

List of Figures

List of Tables

List of Abbreviations

2D convolution	Tow-Dimensional Convolution
2D-BRLSTM	Two-dimensional Bidirectional Recurrent Long Short-Term Memory
3D model	Three-Dimensional model
AF2	AlphaFold2
ANNs	Artificial Neural Networks
ASE	Accuracy of Self-Estimate
AUC_PR	The area under the Precision-Recall curve
BFD	Big Fantastic Database
CAD	Contact Area Difference
CAMEO	Continuous Automated Model EvaluatiOn
CASP	The Critical Assessment of Structure Prediction
CDA	Contact Distance Agreement
CFS	Correlation-based Feature Selection
CN	Column Normalisation layer
CNN	Convolutional Neural Network
Cons3	Consensus 3 method
ConsA	Consensus A method
ConsB	Consensus B method
ConsC	Consensus C method
COV	Covariance Matrix
cryo-EM	cryo-Electron Microscopy
Cα	Alpha carbon
Cβ	Beta carbon
DBA	Disorder B-factor Agreement
DCA	Direct Coupling Analysis
DNs	Deep Networks
ELU	Exponential Linear Unit
EMA	Estimation of Model Accuracy
FM	Free Modelling
FPR	False Positive Rate
GDT_HA	Global Distance Test - High Accuracy
GDT-TS	Global Distance Test - Total Score
HMM	Hidden Markov Model
IN	Instance Normalisation layer
lDDT	local Distance Difference Test
LSTM	Long Short-Term Memory
MD	Molecular Dynamics
MILP	Mixed Integer Linear oPtimisation
MLP	Multi-layer Perceptron
MQAPs	Model Quality Assessment Programs
MSA	Multiple Sequence Alignment
NBC	Naïve Bayes Classifier

NMR	Nuclear Magnetic Resonance
NN	Neural Network
plDDT	predicted local Distance Difference Test
PLM	PseudoLikelihood Maximisation
PR	Precision-Recall curve analysis
PRE	PREcision matrix
PSSMs	Position-Specific Scoring Matrices
QA	Quality Assessment
QE	Quality estimation
ReLU	a Rectified Linear Unit
ResNet	Residual convolutional neural Network
RF	Random Forest
RN	Row Normalisation layer
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic curve
ROC AUC	The area under the Receiver Operating Characteristic curve
SE	Squeeze-and-Excitation block
SSA	Secondary Structure Agreement
S-score	Superposition-based score
SVM	Support Vector Machine
TBM	Template-Based Modelling
TM-score	Template Modelling score
TPR	True Positive Rate

Acknowledgement

Praise be to Allah, by whose grace good deeds are accomplished.

Praise be to Allah, who taught us what we did not know before.

I want to thank Allah for the knowledge and grace He has given me and for making the path easy for me to seek knowledge. Alhamdulillah.

I would like to thank my honourable supervisor, Professor Liam McGuffin, for his support and assistance throughout my doctoral studies. My supervisor was and still is a gentleman who never hesitated to furnish any information to enrich my knowledge in the field. He was generous in his guidance and advice to professionally write this thesis. I appreciated his efforts in facilitating the publication of our scientific studies in prestigious scientific journals and participating in the international conference. I am grateful for every moment I spent during this scientific journey under his supervision.

I would like to thank my fellows in our research group for their support and assistance in completing my research and participation in disseminating our scientific achievements. I want to thank them all without exception: Recep Adiyaman, Nicholas Edmunds, Ahmet Genc.. I also had the pleasure of mentoring Megan Hird for her dissertation, and I am grateful for her cooperation in completing certain parts of the research. Furthermore, I would like to thank our former colleagues - Ali Maghrabi, Fahd Aldowsari, and Limcy Philomina - for their invaluable advice during the early stages of my research.

A special thank you goes to my generous family, who still see in me what I cannot see in myself.

They have always believed in me and supported me throughout my journey. I am especially grateful to my father, who has been my primary source of inspiration and motivation in seeking knowledge. He has always encouraged me to pursue my academic goals and helped me in every way possible. I am proud to be his daughter and grateful for his unwavering support in my pursuit of a doctorate.

I am extremely grateful for my dear mother, who has always prayed for my success and well-being. She has been a constant source of encouragement for me to pursue a doctoral degree, and she believed in me every step of the way. Thanks, Mom, for your unwavering faith. Furthermore, I want to express my appreciation to my beloved siblings and friends for their support and prayers throughout my ongoing journey.

Finally, my sincere gratitude goes to the Saudi Arabian government and my university for their moral and material support in facilitating the necessary procedures to complete the doctoral study.

Thank you all.

Chapter 1 Introduction

Work presented in this chapter has been published in the following book chapter:

Shuaa Muslih Alharbi, and Liam James McGuffin (2023). Machine Learning Methods for Predicting Protein Contacts. In L. Kurgan, *Machine Learning in Bioinformatics of Protein Sequences* (pp. 155–181). WORLD SCIENTIFIC.

1.1 Proteins

Four major macromolecules sustain life in living organisms: polysaccharides, lipids, nucleic acids, and proteins. Proteins are essential components in the composition of all cells and tissues, constituting approximately 15.1 % of the body weight (Wang *et al.*, 2003; Ma *et al.*, 2022). Proteins are involved in most biological systems and are critical in many activities, such as immunological defence, structural support, the catalysis of chemical processes, and hormone regulation (Nahirñak *et al.*, 2012; Stollar and Smith, 2020; Ma *et al.*, 2022). The unique shapes of proteins determine their functions in different activities. Therefore, understanding the structure-function relationship of proteins is crucial for studying their roles in biological processes and developing therapeutic interventions.

1.1.1 The Native Structures of Proteins

Proteins are carbon-based structures, as carbon (C) atoms are the major constituents of the amino acid building blocks. Amino acids have three common components: an amino group (-NH2), a carboxylic acid group (-COOH) and an alpha carbon (Cα) atom. Amino acids are then linked by peptide bonds, constituting linear chains of polypeptides. The amino acids are distinguished by the R group or side chain attached to Cα atom. The R group has a varied chemical nature, giving each amino acid its unique properties, such as polarity and hydrophobicity (Figure 1.1). This group diversifies the amino acids' functionality, allowing proteins to perform a vast array of functions in the body (Stollar and Smith, 2020).

Protein structures fold through both covalent (disulfide) and non-covalent interactions between amino acids within the polypeptide chains. These non-covalent bonds are weak and reversible, which means they could be broken and reformed during the protein dynamic movements. The non-covalent bonds include hydrogen bonds, ionic bonds, van der Waals forces as well as hydrophobic interactions. The different chemical properties of the amino acid side chains allow

a protein to adopt its three-dimensional (3D) structures to perform the function needed (Stollar and Smith, 2020).

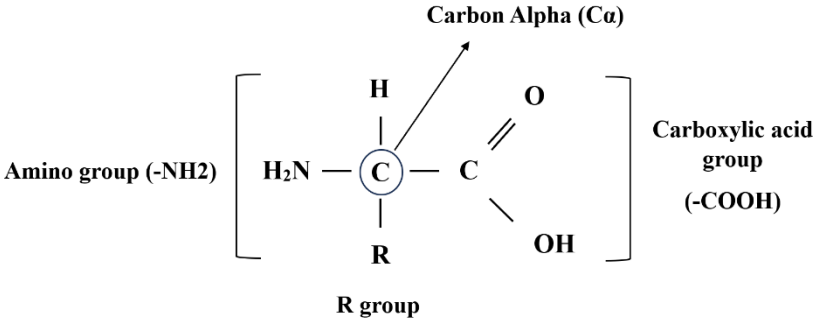


Figure 1.1. The general biochemical structure of amino acid. The amino acid is a carbon-based structure (carbon alpha) consisting of three chemical molecules: amino group (-NH2), carboxylic acid group(-COOH), and hydrogen (H). The amino acid has an R group attached to Cα, which makes it unique from other organic compounds.

Specific sequences of amino acids form proteins with specific shapes and functions. The interaction between side chain and backbone atoms guides the protein folding process, which shapes the protein into its specific 3D structure. The specific protein structure is crucial in function as it determines how it interacts with other molecules in biological systems. Proteins perform versatile functions, such as acting as enzymes in metabolic reactions, acting as messengers in genetic translation by serving as transcription factors that regulate the initiation of RNA synthesis, or providing structural scaffolds in cell building. Therefore, changes in protein structure can cause malfunctioning cellular systems, which can lead to disease. For example, sickle cell disease is caused by a mutation in the structure of haemoglobin, a blood protein responsible for carrying oxygen into cells. The mutation alters a single amino acid, resulting in a change in the shape of haemoglobin. The modified shape of sickle haemoglobin prevents it from effectively binding with oxygen, leading to various health complications. Hence, understanding protein structures can help us learn more about the biological systems for developing treatments and interventions for diseases caused by protein structure abnormalities (Stollar and Smith, 2020).

Protein structures form and fold in four levels: primary, secondary, tertiary and quaternary. The primary structure refers to protein sequence, the specific linear arrangement of amino acids. In protein sequence, each amino acid is a residue, and a series of peptide bonds between carbon and nitrogen atoms constitute the backbone chain of the protein. The secondary structure is formed by organising the primary structure due to hydrogen bonds between each residue's carbonyl (C=O) and amino group. The backbone chain can rotate around the Cα atoms, forming two types of secondary structures: alpha-helices and beta-sheets. Tertiary structures refer to the 3D shape or fold of the protein. At this level, the protein fold is stabilised with longer-range interactions. Hydrophobic interactions, ionic bonds, Van der Waals forces and disulfide bridges between the amino acid residues help fold up the chain and organise the secondary structures

in 3D. The specific 3D folds confer specific functions and allow specific interactions between chains. The quaternary structure level is composed of multiple folded subunits with two or more interacting polypeptide chains (Figure 1.2) (Stollar and Smith, 2020).

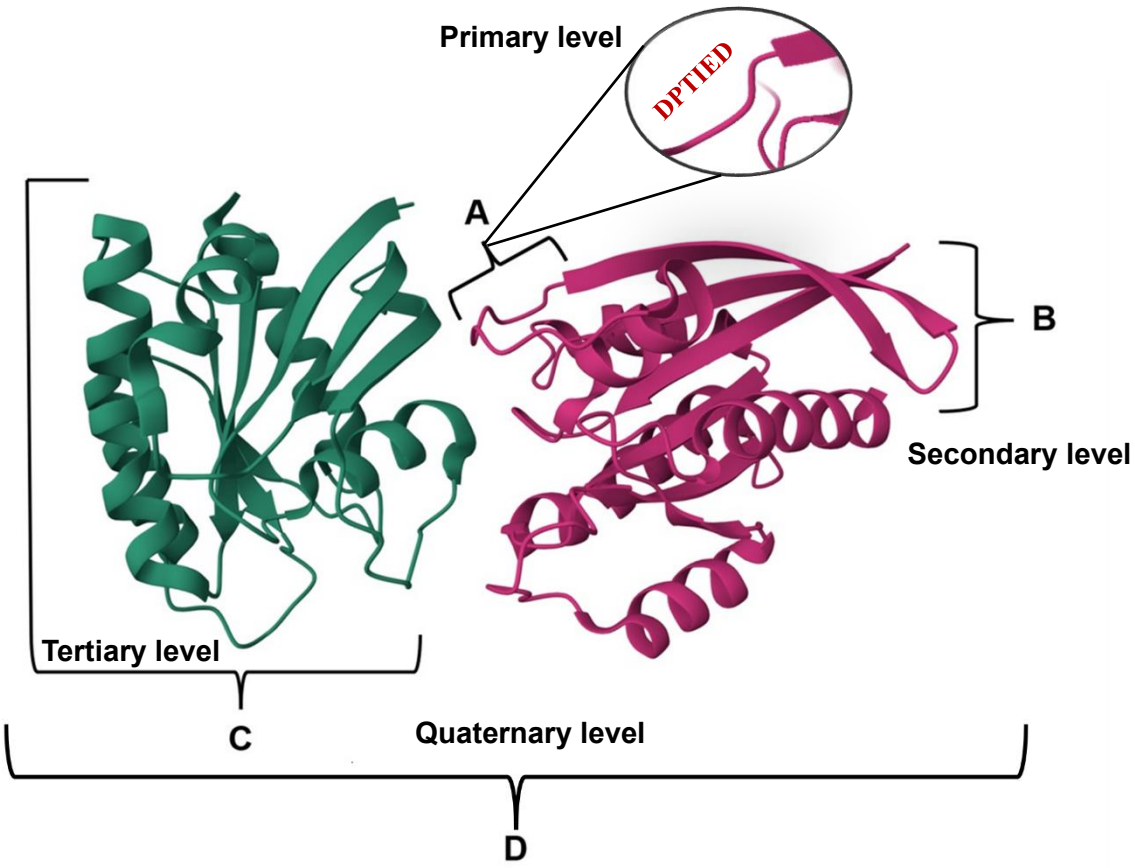


Figure 1.2. The levels of protein structure. A) A primary structure represents amino acid sequences, which is a simple level of protein structure. B) Beta sheet is one type of secondary structure that represents a local structure of protein and is determined by linking amino acid sequences through hydrogen bonds. C) Tertiary structure forms a 3D structure of Polypeptide chains created by interacting side chains of amino acid sequences. D) Quaternary level is a complex of secondary structure units linked by non-covalent interaction. The example query protein is an active KRAS G12D (GPPCP) dimer in a complex with BI-5747(PDB ID: 7ACA). The 3D structure was visualised by Mol*Viewer (Sehnal *et al.*, 2021). Adapted from (Kessel and Ben-Tal, 2018).

1.2 The Protein Folding Problem

Research on the folding of protein sequences into their tertiary structures has received much attention for decades because of its importance to biomedical sciences, biotechnology and other fields in the life sciences. Understanding how protein sequences fold into (3D) structures is fundamental and helps us to solve biological problems based on the sequence-structure-function paradigm. Using knowledge of protein 3D structures, researchers can better understand the pathways of biological systems and disease mechanisms. Experimental approaches, including cryo-electron microscopy (cryo-EM), X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy, have been developed to determine protein structures (Breda *et al.*, 2008; Rangwala and Karypis, 2010; Suh *et al.*, 2021; Bertoline *et al.*, 2023).

The cryo-EM uses a freezing method to analyse protein samples with an electron microscope. NMR applies a magnetic field to analyse the responses of atomic nuclei in protein samples. X-ray crystallography involves crystallising a protein sample and subjecting it to X-ray analysis. These techniques uncovered protein structures, enriching our understanding through experimentally derived structural data (Ma *et al.*, 2022). However, these methods have drawbacks, as they are expensive, and the effort required to resolve structures can be time-consuming, with some structures taking many years to solve. Conversely, obtaining DNA and protein sequences is comparatively very rapid and inexpensive. Therefore, as a result of this disparity, there is a notable gap between the number of protein sequence entries in protein databases (at the time of writing this chapter there are 251,600,768 sequences in UniProt: https://www.ebi.ac.uk/uniprot/TrEMBLstats) and the number of protein structures that experimental methods in the PDB have determined is 211,103 (https://www.rcsb.org/stats/growth/growth-released-structures) (Emerson and Amala, 2017; Li *et al.*, 2020; Pearce and Zhang, 2021b; Bertoline *et al.*, 2023).

This means that most protein sequences have unknown structures and functions, requiring extensive time and effort to resolve these all experimentally. Fortunately, computational methods have been developed, which provide a rapid and cost-efficient way to elucidate structures compared to experimental structure determination procedures. These methods can work quickly and reasonably accurately to predict three-dimensional (3D) models for protein sequences with unknown structures. Recent biomedical studies used computational methods for modelling protein-related diseases (Anderegg *et al.,* 2022; Bhojwani and Joshi, 2022; Fathi, Sakhteman and Solhjoo, 2023; Sathiyamani *et al.,* 2023)**.** Hence, continued computational research in protein structure and function will contribute to advancements in various fields, including medicine, biotechnology, and bioengineering.

1.3 Protein Structure and Function Prediction

Computational studies for protein structure and function prediction include three major categories: protein tertiary structure prediction (a.k.a single chain prediction), protein complex structure prediction, and protein function prediction. In each of these fields, computational tools were developed to predict or assess certain aspects related to the protein prediction field. The tertiary structure prediction field is interested in solving the prediction problem for single protein structures. The computational methods of tertiary structure prediction also involve the subcategories of modelling, quality assessment, refinement, and contact prediction (Farhadi, 2018; Pereira *et al.,* 2021; Huang *et al.,* 2023). Here, the prediction goal is to predict 3D coordinates of proteins from their target sequences with high accuracy (Jumper *et al.,* 2021b; Pereira *et al.,* 2021).

The protein complex structure prediction field intends to predict 3D models for protein interactions, including quaternary structures and the interactions with other biological macromolecules such as nucleic acids (Puton *et al.,* 2012). Various protein complex prediction

approaches have been developed to predict the structures and interaction interfaces of large assemblies (Puton *et al.*, 2012; Zahiri *et al.*, 2020). Finally, protein function prediction methods attempt to identify the functions and/or model the functional regions. or binding sites in protein structures. In this field, the predictors aim to design methods to predict the potential functional parts of protein in biological systems, such as the ligand-binding sites in 3D models (Roche, Buenavista and McGuffin, 2013; Bonetta and Valentino, 2020; Ma *et al.*, 2022). Our main initial focus will be to develop methods to help improve quality estimates for protein tertiary structure prediction, however, aspects of our approach might also be applied to quaternary structure prediction and function prediction in future.

1.4. Computational Methods for Tertiary Structure Prediction:

Classical computational studies sought to understand the native state of protein structures based on principles of physical law. The earlier computational analysis of protein structures started in the 1960s when Shneior Lifson extended molecular mechanics modelling to include large molecules (Hagler and Lifson, 1974; Hagler, Huler and Lifson, 1974; Wodak *et al.*, 2023). The technique aimed to compute a protein's physical and chemical characteristics in a vacuum. After that, computational investigations were conducted to study the behaviour of amino acid residues and determine their electrostatics in solution state by designing the computational models based on molecular mechanics and continuum electrostatics (Eisenberg and McLachlan, 1986; Gilson, Sharp and Honig, 1988; Onufriev, Case and Bashford, 2002; Marcu, Tăbîrcă and Tangney, 2022; Wodak *et al.*, 2023). This physical-based method aimed to use a force field function to estimate the forces between residues and the potential energy, which had limited accuracy in predicting the native states of protein structures (Onufriev, Bashford and Case, 2004; Ho and Dill, 2006; Wodak *et al.*, 2023).

The next move in modelling prediction was designing knowledge-based methods to employ

statistical algorithms such as coarse-grained potentials. Such algorithms were used to model small protein structures from their amino acids as well as been used for ranking and scoring the models (Levitt, 1976; Jernigan and Bahar, 1996; Shen and Sali, 2006; Kmiecik *et al.*, 2016; Marcu, Tăbîrcă and Tangney, 2022; Wodak *et al.*, 2023).

With the exponential growth of protein databases in the 1990s, the predictors started to exploit the benefits of experimental structure data to design computational modelling methods. The experimental structures were used as templates in modelling methods to predict the unknown protein structures from the same family. This kind of method was known as template-based modelling (TBM) as they predicted protein structures based on evolutionary similarity, or homology, to proteins with solved structures. In other words, the TBM methods were developed based on the assumption that homologous proteins with similar sequences will adopt similar structures. The methods were designed to align the target sequence of interest with the template and then copy the equivalent template residue coordinates to produce a predicted 3D model (Zhang, 2008; Zhang, 2009b; Kuhlman and Bradley, 2019; Dhingra *et al.*, 2020; Elofsson, 2023; Wodak *et al.*, 2023). These TBM approaches and available experimental structures can provide structural information for a significant portion of known protein families (Ovchinnikov *et al.*, 2017; Kuhlman and Bradley, 2019).

Protein targets, whose similar experimental structures were not discovered, were predicted based purely on their sequences to derive their physical and chemical features (Bonneau *et al.*, 2001). Such prediction approaches were traditionally called *de novo* prediction or *ab initio* prediction methods. As these methods did not use templates to apply structural similarity techniques, they are now known as template-free modelling (FM) methods in the tertiary structure prediction field (Kuhlman and Bradley, 2019; Dhingra *et al.*, 2020; Wodak *et al.*, 2023). Template-free modelling was one of the remaining challenges in protein structure prediction, as it was difficult to model protein structures without templates. Unlike TBM

methods, FM methods required intensive computational resources and often led to predicted models with lower accuracy than those obtained via TBM (Kuhlman and Bradley, 2019).

The accuracy of the prediction methodology was the foremost concern in the protein structure prediction field. Therefore, incremental development had significantly enhanced the predictive performance. For improving the ab initio prediction, fragment-based assembly procedures were introduced to assemble parts (fragments) derived from related protein structures into the model being studied (Bonneau *et al.*, 2001; Jones, 2001; Zhang, 2009a; Kuhlman and Bradley, 2019; Marcu, Tăbîrcă and Tangney, 2022; Elofsson, 2023; Wodak *et al.*, 2023).

A significant development in modelling approaches was incorporating the evolutionary information from multiple sequence alignments (MSAs). One of the proposed uses for this evolutionary data was to use it to derive predicted contacts between amino acids within the folded chain (Göbel *et al.*, 1994). This approach is based on the hypothesis that if two residues at different positions in the sequence show simultaneous mutations in the sequence alignments, then the mutations are correlated and therefore, these two residues are likely in contact with 3D structures (Kuhlman and Bradley, 2019; Wodak *et al.*, 2023). However, the performance was modest for these early "correlated mutation" based methods as this approach was affected by the transitive correlations in alignment, where two residues might be indirectly correlated with the third residue, leading to noisy results. To solve this issue, statistical approaches such as direct coupling analysis pseudolikelihood optimisation were introduced to minimise the noise in the alignment (Morcos *et al.*, 2011; Ekeberg *et al.*, 2013; Kamisetty, Ovchinnikov and Baker, 2013; Wodak *et al.*, 2023). However, even using statistical-based methods, the resulting predicted models had limited accuracy.

As the technology developed, modelling methods were improved by integrating the statistical-based methods with machine learning methods to improve contact prediction accuracy (Wang,

Sun, *et al.*, 2017; Pearce and Zhang, 2021a; Zheng *et al.*, 2021; Elofsson, 2023). The modelling methods integrating evolutionary-based approaches with deep learning neural networks, which are so-called 'meta-servers', advanced the modelling prediction and improved the accuracy of 3D models (Wodak *et al.*, 2023). The performance of such methods was attributed to the ability of deep learning algorithms to learn and extract the hidden patterns in the experimental data (Lee *et al.*, 2022). These algorithms subsequently advanced to such an extent that they gained phenomenal attention - it was claimed that the single protein chain prediction problem was effectively "solved" by AlphaFold2 (AF2) (Jumper *et al.*, 2021a). The second version of AlphaFold was developed using an end-to-end learning approach based on transformer models that incorporated the evolutionary data and geometric and physical restrictions (Lee *et al.*, 2022; Marcu, Tăbîrcă and Tangney, 2022; Wodak *et al.*, 2023). Following the lead of AF2, other methods, such as RoseTTAFold (Baek *et al.*, 2021), were developed by expanding its transformer architecture and adding a three-track neural network. These two methods predicted 3D models with comparable accuracy to experimental structures (Baek *et al.*, 2021; Wodak *et al.*, 2023). However, it must be stated that they still have significant local errors for many targets (Akdel *et al.*, 2022; Liang *et al.*, 2022; Pak *et al.*, 2023; Wodak *et al.*, 2023).

1.4.1 Quality Estimation Prediction

Protein structure modelling pipelines often involve several stages, including predicting contact and distance maps, building 3D models, scoring of model quality to identify any local errors and then model refinement to fix the errors. In the modelling process, often many alternative models are generated, and these need to be ranked based on their quality in order to select the ones closest to the native structures. Thus, this assessment and ranking by model quality is crucial in various stages of protein structure prediction, from refinement to model selection (Won *et al.*, 2019).

To assess the quality of the generated models, scoring methods known as quality estimation (QE) or quality assessment (QA) methods have been developed. These methods aim to evaluate the models at two levels: local and global. Local assessment focuses on detecting errors in specific localised regions of the model, i.e., how accurately the modelled residue coordinates are predicted to match up with their corresponding residue coordinates in the reference structures. Local quality assessment helps us to estimate discrepancies and errors in the local regions of the model, producing scores for each residue. Global assessment aims to evaluate the overall quality of the model, and these scores can be compared to rank alternative models. A single global score is applied, which considers each model in its entirety and estimates its overall similarity to the native structure (Won *et al.*, 2019).

Various scoring methods were developed to assess individual models of proteins using different algorithms. For example, Benkert *et al.* (2011) proposed a method to estimate the absolute quality of individual protein structure models. Their method combines various scoring functions to assess the local and global quality of the models (Benkert, Biasini and Schwede, 2011). Melo and Feytmans (1998) developed a non-local atomic interaction energy-based method to assess protein structures. This method considers the interactions between atoms in the protein structure to estimate its quality.

McGuffin's research group introduced the quasi-single model approach, which is effective in providing accurate assessments of model quality given only a single model. The approach starts with generating alternative conformations based on the target sequence and then compares them with the target model using a clustering-based approach (Roche, Buenavista and McGuffin, 2014). McGuffin's group was the pioneer of this approach, which was first applied in the third version of ModFOLD (McGuffin and Roche, 2010), a leading web server that is designed to estimate the accuracy of 3D model of proteins (McGuffin, 2008). These advancements were built upon to improve the predictive performance of ModFOLD in

subsequent versions, and the sixth version of ModFOLD used a combination of various alternative model quality scoring methods as inputs to a neural network. It used a sliding window of per-residue score inputs from each method and was trained to output a single quality score for each residue in a model (Figure 1.3) (Maghrabi and McGuffin, 2017).

The development of ModFOLD6 further demonstrated the potential of adding more data to neural networks to improve QE methods (see Chapter 3 for more details) (Maghrabi and McGuffin, 2017; Cheng *et al.*, 2019; Maghrabi, 2019; McGuffin *et al.*, 2021; McGuffin *et al.*, 2023). With the recent innovations in tertiary structure modelling, the accurate assessment of very high-quality models became a new challenge for QE methods, as it is harder to discriminate between them. However, with the development of many methods that are either on par with or surpass AF2 in terms of modelling performance, it becomes more important for users to be able to discriminate between models from different sources using consistent, unbiased and independent model quality estimates.

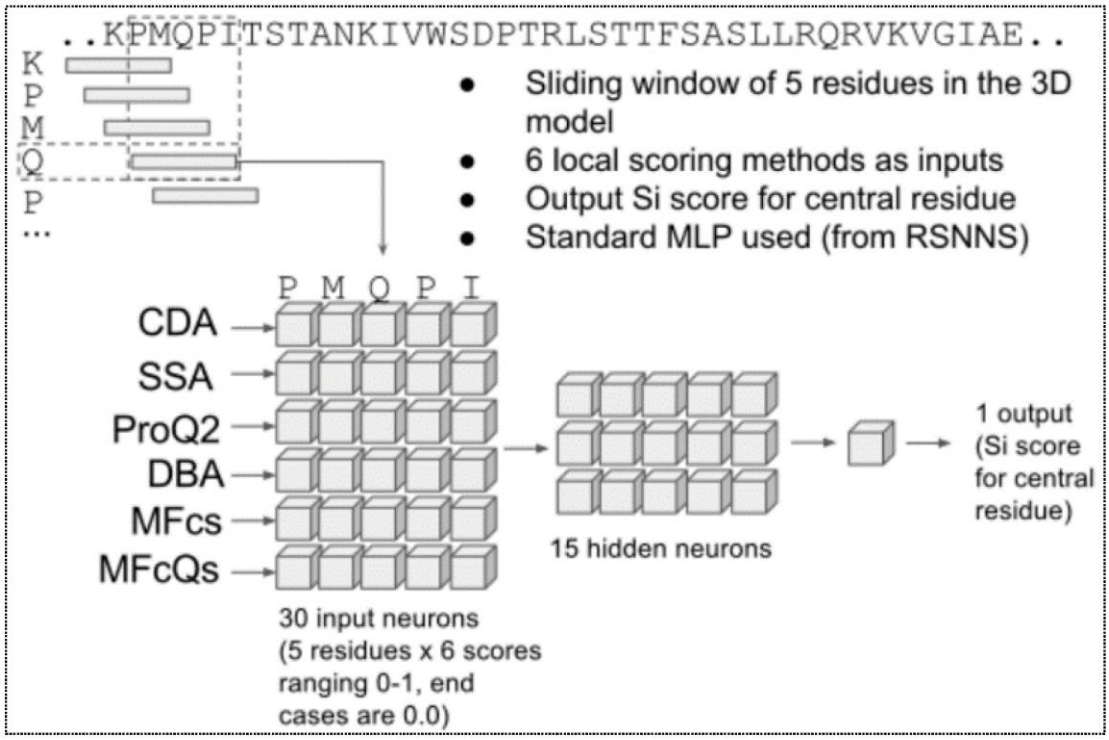


Figure 1.3. An illustration of a neural network-based approach using a sliding window to integrate the per-residue scores in ModFOLD6. The standard MLP is multi-layer perceptron neural network. The window size is 5 residues with 6 quality scores for each residue. The per-residue scores produced from six methods, which are ModFOLDclust_single (MFcs), ModFOLDclustQ_single (MFcQs), ProQ2, Contact Distance Agreement (CDA), Disorder B-factor Agreement (DBA), and Secondary Structure Agreement (SSA). The input of each residue is 30 scores (5 X 6) feeding the first layer of the neural network. The scoring procedure was conducted in hidden layers with 15 neurons. The neural network processes the input scores and generates a quality score (Si score) as an output for each residue in the protein model (Maghrabi and McGuffin, 2017; Maghrabi, 2019).

1.5. Residue-Residue Contact Prediction

As previously discussed, the quality of protein structure models is crucial if they are to be used in biomedical fields such as drug discovery. Therefore, protein structure prediction methods have been developed iteratively over the years in order to increase the accuracy of the 3D models that they produce (Heo and Feig, 2020; El-Rashidy *et al.*, 2021; Kryshtafovych, Moult, *et al.*, 2021). This development involves incorporating various protein features from their sequences and structures, such as the inter-residue contact maps, into protein structure prediction pipelines (Zheng *et al.*, 2019; Yang *et al.*, 2020; Pakhrin *et al.*, 2021). When protein sequences fold, the amino acid residues interact to form 3D structures by creating non-covalent bonds between their atoms (McMurry *et al.*, 2013). Thus, residue interaction predictions, or contact maps, can provide valuable information describing the tertiary structure, which can be exploited to reconstruct 3D models, leading to enhanced quality (Figure 1.4) (Konopka *et al.*, 2014; Hou *et al.*, 2019). This information is derived from predicting pairwise contacts in a protein sequence and is employed by many researchers to predict protein folding, for example, by restricting the conformational space of *ab initio* modelling (Lundström *et al.*, 2008; Wang, Sun and Xu, 2018; Adiyaman and McGuffin, 2019; Jing *et al.*, 2019). Moreover, protein contact prediction methods have been integrated in model quality estimation servers to detect both the local (per-residue) and global errors in models (Cheng *et al.*, 2019; Jing *et al.*, 2019; McGuffin *et al.*, 2021; McGuffin *et al.*, 2023). Furthermore, in refinement processes, contact prediction has been used as part of a "gradual restraint strategy" (Adiyaman and McGuffin, 2021). For transmembrane proteins, predicting contacts between the transmembrane alpha-helices helps to elucidate the protein fold, which can, in turn, help to predict functions (Fang *et al.*, 2020).

Due to their potential usefulness for predicting protein folding, methods for the prediction of contacts between residues have been in development since the early 1990s (Pearce and Zhang,

2021b). Covell and Jernigan (1990) used the lattice model to represent amino acid residue contacts for restricting a conformational space of globular proteins to predict all possible chain conformations. This approach was useful for predicting a small group of protein structures. In 1996, contact prediction was introduced as a part of the ab initio category for secondary and tertiary structure prediction in the second round of the Critical Assessment of Structure Prediction (CASP2) (see section 1.6 about CASP). In this experiment round, contact prediction methods were developed using the principle of correlated mutation of coevolutionary residues (Lesk, 1997; Monastyrskyy *et al.*, 2011).

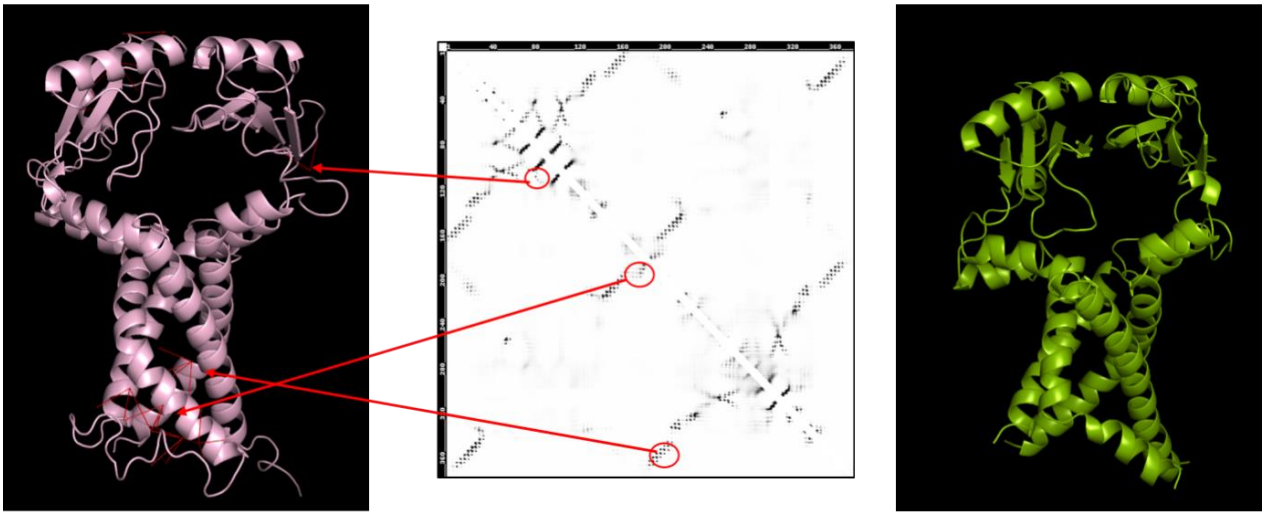


Figure 1.4. A diagram illustrates the role of a contact map. The role contact maps can play in enhancing the accuracy of 3D protein modelling. The diagram shows the experimental structure, the predicted 3D model, and the siderophore reductase FoxB contact map (PDB ID: 7awb). The protein's experimental structure was determined via X-ray diffraction. The contact map was predicted using RaptorX-Contact (Wang, Li, *et al.*, 2017; Wang, Sun, *et al.*, 2017; Wang, Sun and Xu, 2018; Xu, 2019; Xu and Wang, 2019), and residue pairs predicted to be in contact are visually represented by red circles. The 3D model of protein predicted by trRosetta (Yang *et al.*, 2020).The protein models were visualised using PyMOL (pymol.org). The Figure taken from (Alharbi and McGuffin, 2023).

1.5.1 Residue Contact Prediction Definitions

Contacting residue pairs in protein structures can be identified by calculating the distance between carbon atoms of the amino acid residues at a specific threshold. The distance threshold between carbon atoms of residue pairs in a protein structure has different values depending on the goal of the contact prediction. For predicting helix-helix interactions in transmembrane proteins, contacts between residues are defined as distances less than 5.5 Å between two heavy atoms of the side chain or backbone. An alternative definition considers the contact distance threshold to be less than 8 Å between beta carbon (Cβ) atoms of side chains (Wang *et al.*, 2011; Jing *et al.*, 2019). For modelling the 3D structures of proteins, contacts have been defined between Cβ atoms (or between Cα atoms in the case of Gly) using different distance cut-offs of between 7 and 11 Å (Duarte *et al.*, 2010; Wang *et al.*, 2011; Yuan, Chen and Kihara, 2012; Adhikari and Cheng, 2016). However, in the contact prediction evaluation process of the CASP experiments (see section 1.6), the formal definition is that residue pairs are in contact if the distance between their Cβ atoms (Cα in Gly) is less than 8 Å (Monastyrskyy *et al.*, 2011; Monastyrskyy *et al.*, 2014; Monastyrskyy *et al.*, 2016; Schaarschmidt *et al.*, 2018; Jing *et al.*, 2019; Shrestha *et al.*, 2019). All these threshold values are in a range that allows non-covalent interactions to be measured as protein sequences folded up into 3D shapes (Emerson and Amala, 2017).

1.5.2 Contact Maps

To represent contacts between residues computationally, “contact maps” have been devised as two-dimensional (2D) matrices (N × N), where N is the length of the protein sequence. The contacting residue pairs are set to 1 if the distance between their atoms is less than or equal to a given cut-off value; otherwise, 0 indicates the residue pairs that are non-contacting. The

distance between the same residue position is also set to 0 and represents the diagonal line in the contact matrix. Thus, a contact between each residue can be represented as a dot and the x- and y-axes represent residue positions along the sequence length (Figure 1.5) (Emerson and Amala, 2017; Jisna and Jayaraj, 2021; Suh *et al.*, 2021).

Furthermore, the type of contact is particularly important in determining protein structures. To classify contact types, the number of residues between two residue pairs that are predicted to be in contact determines the type of contact. In other words, if there are more than 24 separate predicted residue pairs, their contact is classified as being long-range; if there are more than 12 but less than 23 residues, the predicted contact is classified as medium-range; and if there are more than 6 residues but less than 12, predicted contacts are classified as short-range (Monastyrskyy *et al.*, 2011; Monastyrskyy *et al.*, 2014; Schaarschmidt *et al.*, 2018; Jing *et al.*, 2019; Shrestha *et al.*, 2019). The long-range contacts contribute to improving the quality of 3D models as they assist in positioning the secondary structures at the right distance. Therefore, this type of contact can be used as a restraint for conformational spaces in predicting the structures *ab initio* (Latek and Kolinski, 2008; Yuan, Chen and Kihara, 2012; Jing *et al.*, 2019; Jisna and Jayaraj, 2021). According to the CASP evaluation system, each residue pair predicted to be in contact can be assigned by calculating the probability score. The length of the target domain (L) with the greatest probability value is used to divide each contact range into subsets (L/5, L/2, L, FL, where FL indicates all predicted contacts in these sets). In this chapter, we will refer to the accuracy of contact prediction by machine learning approaches as L/5 long-range contacts for template-free or free-modelling (FM) targets (Schaarschmidt *et al.*, 2018; Shrestha *et al.*, 2019).

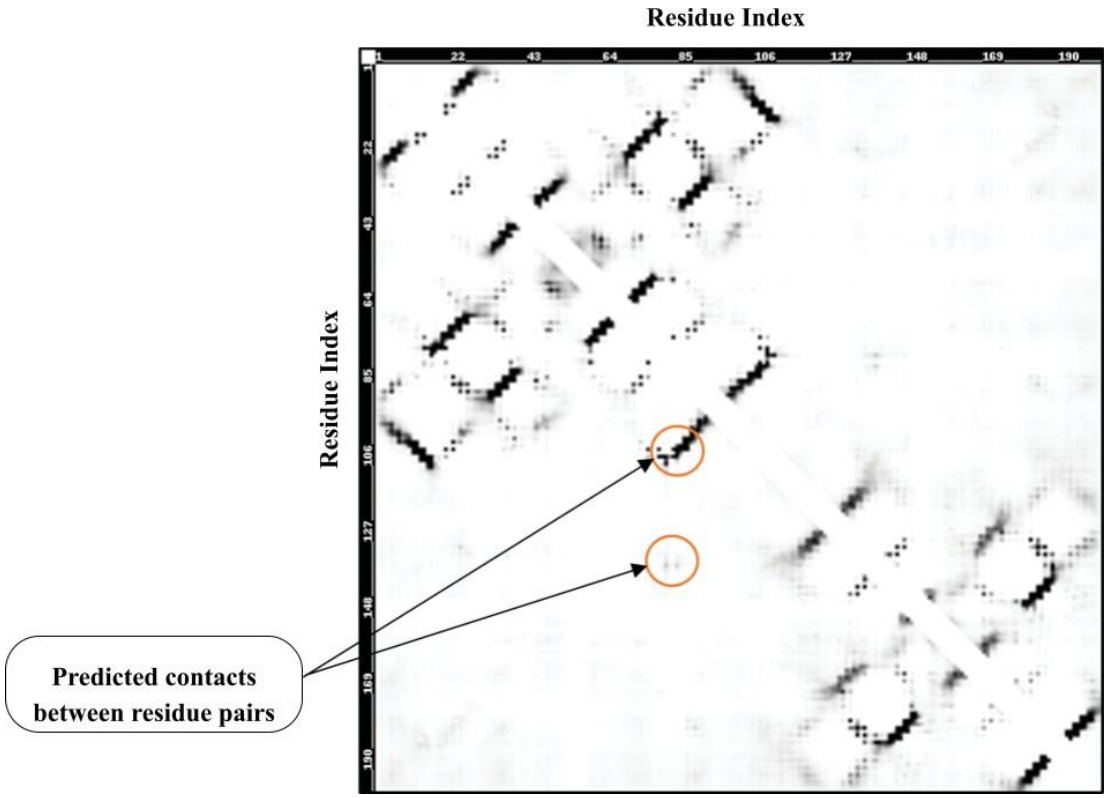


Figure 1.5. An illustration of a protein contact map. The contact map depicted here is for the TCR-017 ectodomain protein (PDB ID:7EA6). A diagonal line on the map indicates a residue that is in contact with itself and has a value of 0. Black dots indicate residue pairs that are in contact. The prediction of this contact map was conducted using RaptorX-Contact (Wang, Li, *et al.*, 2017; Wang, Sun, *et al.*, 2017; Wang, Sun and Xu, 2018; Xu, 2019; Xu and Wang, 2019; Xu, McPartlon and Li, 2021)**.** The Figure taken from (Alharbi and McGuffin, 2023).

1.6 The Critical Assessment of Protein Structure Prediction (CASP) Community

Numerous computational tools have been developed to predict protein structure, which raised concerns among scientists regarding the dependability and applicability of these methods in other related fields. This brought to light the necessity for unbiased and genuine evaluation materials to criticise the viability of protein structure prediction techniques. In 1994, John Moult and his colleagues introduced the CASP experiment as a large-scale test to assess computational methods in protein structure prediction. Every two years, the CASP experiment evaluates the performance of bioinformatic methods in predicting protein structures (Moult *et al.,* 1995).

The CASP experiments have been assessing and promoting developments in protein structure prediction for about thirty years. Since the beginning, the assessment of single protein chain structural modelling has been at the core of the CASP programme (Simpkin *et al.,* 2023). Focusing on the main category of tertiary structure prediction, the previous assessment additionally covers prediction techniques related to different subcategories, such as tertiary structure model, refinement, quality estimation and contact residue prediction in protein structures (Kryshtafovych *et al.,* 2019). The recent experiment (CASP15) categories have been adopted to address new challenges that have emerged after the success of AF2 in advancing solutions to the tertiary structure prediction problem.

The CASP is a blind test where participants are asked to predict models of protein sequences whose experimental structures have yet to be publicly released. The predictor participants are classified into two types of groups: the human groups and the automatic servers. The predictors who use their tools with human intervention are classified as human groups, whereas those who use their servers without human intervention are assigned to automatic servers. Official assessors then evaluate predicted models from all groups. The assessors use advanced evaluation methods that assess the performance of the prediction tools, which have

continuously improved over successive years (Wodak *et al.,* 2023). Hence, the CASP experiment has highlighted the progress in protein structure prediction, providing valuable insights into new ideas to elevate the performance of computational methods.

1.7. Advancements in Contact Prediction Methods through Successive CASP Experiments

Contact prediction was introduced in the early years of CASP. In CASP3, predicted residue contacts were introduced as a separate category, apart from the assessment methods for protein structure prediction (Orengo *et al.,* 1999). However, a renewed interest in predicted residues contacts occurred in 2008 with CASP8 (Ezkurdia *et al.,* 2009). Predictor groups developed their methods by using different approaches based on extracting correlated mutations in MSAs, applying machine learning on contact maps, or a combination of approaches. These earlier methods with low accuracy might be an aid in selecting the best models of FM targets, which might be employed by consensus prediction tools for predicting the harder targets (Tress and Valencia, 2010).

In CASP9, methods for predicting contacts in a protein structure had improved a little further. The same sorts of methods that had been assessed in CASP8 were further developed through a combination of the different approaches, where correlated mutation methods to predict residues had been integrated with machine learning methods (Monastyrskyy *et al.,* 2011). In addition, there was a method that used information, which was obtained from templates of homologous protein, e.g. HMMSTR-CM (Shao and Bystroff, 2003; Monastyrskyy *et al.,* 2011). Although most of the contact prediction methods have been steadily improved upon, the servers based on machine learning had the best accuracy overall in the contact prediction category (Monastyrskyy *et al.,* 2011).

Regarding CASP10, there was a considerable advance in contact prediction servers, where some servers had been improved by further integrating machine learning strategies with features of protein sequences (Monastyrskyy *et al.,* 2014). Interestingly, some studies indicated that the contact information was accurate enough to be used for the improvement of 3D modelling in tertiary structure prediction methods. However, the performance of contact prediction methods only achieved ~20 % accuracy, the same level as that in previous CASP experiments. In contrast, the accuracy of contact prediction on difficult targets took a leap forward, reaching 27 % in CASP11, meaning that the information from contacting residues was even more useful for the improvement of 3D models (Monastyrskyy *et al.,* 2016).

A major breakthrough in the improvement of contact prediction methods was seen in CASP12 with the advent of hybrid methods, representing a merger of coevolution information with machine learning (Schaarschmidt *et al.,* 2018), which the majority of the most successful methods exploited. Furthermore, some of the methods used the outputs of hybrid approaches as the inputs for deep networks, resulting in further substantial improvements in their performance. In addition, the exponential increases in the sizes of the databases of protein sequences allowed methods to extract substantially better evolutionary information from deeper alignments, which further boosted accuracy. Therefore, contact prediction has achieved an unprecedented 20 % increase in the percentage accuracy (to 47 %) in CASP12 (Schaarschmidt *et al.,* 2018). Since CASP12, predictors have been consistently working on the improvement of deep neural network-based methods, which have been the major focus in the development of contact prediction methods. By CASP13, the best method with the top performance was developed by exploiting advanced deep convolutional neural networks to interpret sequence alignment data. This outstanding combination produced a further unprecedented 23 % increase in the accuracy of contact prediction, reaching 70 % for the first time (Shrestha *et al.,* 2019).

The contact prediction methods assessed in CASP14 were significantly advanced in terms of their input features, MSA construction, and the training process of deep learning models. The accuracy of contact prediction depends on two key aspects: the quality of MSA analysis and the training phase of deep neural networks. Advanced MSA analysis approaches were used for predicting the distance between residues. In addition, deep neural networks were trained using distance information, increasing the accuracy of contact prediction. The use of distance and orientation prediction has significantly improved contact prediction methods, which in turn has helped improve the accuracy of 3D protein structure prediction (Senior *et al.,* 2020; Yang *et al.*, 2020; Li and Xu, 2021; Ruiz-Serra *et al.,* 2021). The distance matrix, which provides information on the distances between every pair of residues in a protein, offers more detailed and comprehensive data compared to a contact matrix. This increased level of detail in the distance matrix translates to a greater number of physical constraints and a more comprehensive training signal for protein structure prediction algorithms (Xu and Wang, 2019; Senior *et al.*, 2020; Ruiz-Serra *et al.,* 2021). Despite these advancements, the contact prediction accuracy in this round reached 64 %. CASP14 assessors stated that this accuracy may suggest a regression in contact prediction approaches; it is essential to consider the increased complexity of the CASP14 targets, which could have influenced the observed advancements (Ruiz-Serra *et al.,* 2021). In CASP15, contact prediction with two categories, refinement and model accuracy estimation for monomeric targets, were eliminated due to significant progress in modelling individual protein structures (Kryshtafovych *et al.,* 2023). Since this is the latest experiment of CASP, the finding had not been published at the time of writing.

1.8. Application of Contact Prediction Methods (quality estimation, refinement)

As mentioned, the most successful protein structure prediction pipelines include modelling of the tertiary structures (using template-based and/or template-free methods), evaluating these

models based on quality assessment scoring functions, and then finally refining them to fix any errors in conformation, thereby obtaining higher accuracy of 3D models that are closer to the native structures (Adiyaman and McGuffin, 2019). Many predictors have also adopted state-of-the-art methods of contact prediction and combined them into their servers at these different stages in order to boost the performance of their prediction pipelines.

1.8.1 Estimation of Model Accuracy (EMA) or Model Quality Assessment (QA)

Protein structure prediction methods may produce many dozens or even hundreds of alternative models that vary in their accuracy, both at the local or per-residue level and, overall, at the global level. To detect these errors and to select the optimal model from among alternatives, QA methods have been employed to provide estimates of the model accuracy (Olechnovič and Venclovas, 2017) based on scoring local and global accuracy. QA methods have been developed by integrating various protein features, for example, predicted secondary structure and solvent accessibility (Maghrabi and McGuffin, 2017; Olechnovič and Venclovas, 2017). Residue-residue contact predictions are an important additional feature that has played a key role in the enhancement of recently developed model quality assessment programs (MQAPs).

Protein contact predictions have been used for scoring the local and global accuracy in various model quality estimation servers. These servers can be classified into single-model methods and consensus-model methods depending on their inputs. Single-model methods only consider models individually and have been designed for evaluating local and global accuracy based purely on features of the input model, whereas consensus-model methods make multiple structural comparisons of all models for a given target in order to produce global and local scores and to select those of optimal quality (Studer, Biasini and Schwede, 2014; Uziela and Wallner, 2016; Won *et al.*, 2019). One of the top single model methods in CASP11 was QAcon. This method was developed by adding residue contact information with different protein features (Cao *et al.*, 2017). Contact scores were calculated by executing the PSICOV and

DNcon approaches (see section Approaches of Contact Prediction Methods) and have been used as an input with 11 feature scores to predict the global quality of a model (Jones *et al.*, 2012; Cao *et al.*, 2017). Based on QAcon results, Cao *et al.* (2017) determined that contact prediction can have an impact on the performance of model quality assessment, and this impact depends on the accuracy of that prediction. In terms of local model quality prediction accuracy, ProQ2 was one of the best methods based on the results of CASP12 (Kryshtafovych *et al.*, 2016). The process of ProQ2 can be described as inferring protein model properties from its sequence and structure and then combine these feature scores by using a machine learning method called support vector machines (SVM) for eventually predicting the final score of model accuracy (Ray, Lindahl and Wallner, 2012). Structural features of a model in ProQ2 included atom-atom contact, residue-residue contact and secondary structure. Residue-residue contacts have been reweighted with other features for predicting the local quality. Ray *et al.* (2012) point out that one reason for the performance improvement of ProQ2 could be attributed to the profile weighting of residue contacts and surface area features, which helped to increase the accuracy of predicted local quality. Therefore, CASP12 assessors have been recommended the users for using ProQ2 if they are interested in the local accuracy of a model. Additionally, ProQ2 has been ranked as one of the top-performing quality model assessment in terms of the accuracy of global prediction, which can be calculated by computing the average of local features scores on the length of protein sequence (Ray, Lindahl and Wallner, 2012; Kryshtafovych *et al.*, 2016). It is clear that much of the improvement of QA methods can be attributed to predicting residue contacts accurately, as this provides valuable information that is useful in identifying the errors in protein structure models.

1.8.2 Refinement of Models

Refinement of 3D models is a vital part in most successful protein structure prediction pipelines. The main purpose of the refinement method is to fix any errors that have resulted

from the modelling process, adding further value to the model by bringing it closer to the native structure. In general, refinement servers include two stages: sampling for generating alternative 3D models and scoring for assessing the accuracy of these models. In the sampling stage, refinement approaches can be categorised into fully automated server-based programs and non-server-based programs (Adiyaman and McGuffin, 2019). Methods that rely upon automated servers and use the knowledge of protein structures have had some success at improving parts in the starting models, according to the results of the refinement category in early CASP experiments (MacCallum *et al.*, 2009; MacCallum *et al.*, 2011; Read *et al.*, 2019).

Success in the refinement of 3D models of protein structures is reliant on an accurate energy function and a sufficient conformational search (Park *et al.*, 2019). However, due to the large search space, the refined models generated by refinement methods can often deviate greatly from the initial structures, and there is a large chance that they can result in lower quality models (Adiyaman and McGuffin, 2019; Read *et al.*, 2019). Although refinement could improve homology models with low resolution, even with unrestrained large-scale searches for the lowest energy states, it is clear that refinement of closer to native models with higher resolution can be achieved by restraining the conformational search space (Jagielska, Wroblewska and Skolnick, 2008; Park *et al.*, 2019). Conformational searches can be restrained by using structural information from the starting models as input for refinement methods (Park *et al.*, 2019). These restraints are used as parameters that help to reduce the deviation between the starting and native models.

Most state-of-the-art refinement methods have been improved by combining Molecular Dynamics (MD) simulation algorithms with physics-based force fields. Although these methods have performed well, they were often inefficient due to the lack of restraints for limiting and guiding conformational searches (Adiyaman and McGuffin, 2019). Recently, the utilisation of restraints in refining starting models has led to improvements in refinement

performance, but this success depends on the appropriateness of the restraints used. Strong restraints can restrict the refinement, but the weak restraints improve the refinement process of a model (Feig, 2017; Adiyaman and McGuffin, 2019). Various types of restraints have been derived from different sources of data (Adiyaman and McGuffin, 2019). For example, Zhang *et al.* (2011) have used a distance map derived from high-resolution starting models as restraints to optimise the energy funnel for MD simulations. However, restraints might be more effective when specific parts of models need to be refined, so the guidance of refinement towards fixing the local errors within models instead of the whole models could also be helpful in improving performance.

Improving locally inaccurate regions in 3D models still represents a challenge for refinement methods because of the difficulty in determining these regions (Park *et al.,* 2019). An alternative strategy is to instead rely on the predicted residue contacts interaction for determining restraints. Information derived from residue-residue contact prediction can be used as restraints for guiding in refinement methods to enhance 3D models locally. For example, the GREMLIN tool has been used for restraining the search space based on co-evolution information derived from residue contacts prediction (Park *et al.,* 2019). McGuffin's research group has also investigated the use of contact-based restraints, which have been incorporated into the latest version of ReFOLD method (Adiyaman and McGuffin, 2021), so the accuracy of the predicted contact data that McGuffin's research group will rely upon will be a high priority.

1.9 Approaches of Contact Prediction Methods

Contact prediction accuracy should be sufficient to capture the correct contacts that could be used to bring 3D models as close as possible to the native structures of proteins. Therefore, increasing the accuracy of predicted residue–residue contacts became a core challenge in the field of structural bioinformatics. Many approaches and algorithms have been used to extract accurate contact predictions between residue pairs in protein sequences. Using the evolutionary theory of protein folding, correlated mutation-based methods were developed based on the hypothesis that residue pairs in protein sequences are more likely to have correlated mutations to maintain the stability of protein structures. In other words, if one residue is mutated, then the corresponding interacting residue(s) will also be mutated in a co-evolutionary process to stabilise the protein structure, and these residues could be identified in MSAs (Wu and Zhang, 2008). Therefore, to extract co-evolutionary information, MSA methods are used to identify homologous proteins using various rapid algorithms (Jing *et al.*, 2019; Yang *et al.*, 2020; Pearce and Zhang, 2021b).

The evolutionary theory of protein folding suggests that proteins tend to conserve their structures and function over the evolution period, including homologous proteins, even when their amino acid sequences display variability. The protein structural conservation restricts the variability in homologous sequences. In other words, the changes in the sequences have to maintain the overarching structure and function. Therefore, different amino acid residues in the sequence are forced to coevolve (Morcos *et al.*, 2014). The biological meaning of co-evolution is when two or more molecules affect each other's evolution to maintain the functionality of proteins. Certain protein characteristics, such as 3D structures or catalytic sites, can remain consistent throughout evolution (Thompson *et al.*, 2011; De Juan, Pazos and Valencia, 2013).

Various computational approaches were developed to examine the co-evolutionary features of proteins. The aim of computational tools for the co-evolution of amino acid residues is the identification of residue pairs that could affect their evolution, which could help to predict the functional or structural interaction between protein residues. These methods were designed to analyse the evolutionary modifications in the proteins. Hence, evolution-related methods could help to identify the co-evolution patterns, the repeating changes between residues in a single protein (De Juan, Pazos and Valencia, 2013).

The computational methods of protein co-evolution were designed based on the covarion model. This model recognised the amino acid residues in a protein with interdependent changes over evolution. These approaches relied on the MSA strategy of homologous proteins to identify the correlated mutations between their residues (Thompson *et al.*, 2011; De Juan, Pazos and Valencia, 2013). MSA strategy was used to examine the protein sequence within the context of the total family, which can help to determine the crucial attributes that define large-scale protein functions. These attributes could include 3D structures or catalytic sites that have remained the same (conserved) throughout evolution (Thompson *et al.*, 2011). Various bioinformatic tools, such as HHblits, PSI-Blast, and Jackmmer, were used to generate MSAs (Adhikari and Cheng, 2016).

Such correlated mutations that were derived from MSA determine the associated changes between residues within a protein, which could indicate close residues such as those in direct contact or those that collaborate in catalytic or binding sites. Thus, the co-evolutionary modification could lead to folding the protein correctly to maintain the stability or functionality of the protein in the face of evolutionary pressures (De Juan, Pazos and Valencia, 2013).

1.9.1 Statistical Algorithms for Correlation-based Methods

Earlier contact prediction studies started exploiting the benefit of correlated mutation methods to predict the contact patterns between residues within a single chain. These studies attempted to interpret the mutation correlation features derived from MSAs using statistical independence-based approaches such as correlation coefficients and mutual information (Jing *et al.*, 2019). These methods assumed that residue pairs are statistically separated from neighbouring residue pairs (Horner, Pirovano and Pesole, 2007; Marks, Hopf and Sander, 2012; Jing *et al.*, 2019). They only considered the specific pair when calculating mutation information, ignoring the effects of other residues (Jing *et al.*, 2019).

The correlation coefficient-based algorithms aimed to detect pairs of positions or two columns in a MSA with dependent amino acid frequencies or showed similar patterns of amino acid substitutions. These methods used substitution matrices to calculate the frequency of residues at the positions across different sequences and compute the linear correlation of residue pairs to assess how similar they were. Such methods could capture residue pairs with close proximity within the protein's 3D structure, proposing that they may be in physical contact (Göbel *et al.*, 1994; Olmea and Valencia, 1997; De Juan, Pazos and Valencia, 2013; Jing *et al.*, 2019).

The mutual information-based methods focused on computing the distribution of each residue in multiple sequences for a specific position. The main principle was to assess the extent of the mutual dependence between two positions by quantifying the occurrence or absence of an amino acid in a particular position (Gomes *et al.*, 2012; De Juan, Pazos and Valencia, 2013). The accuracy of statistical independence-based methods was modest due to indirect correlation issues. This could lead to misleading information in a covariance analysis of correlated mutation data, resulting in incorrect contact prediction. To address this issue, global statistical algorithms were employed to remove the noisy data in co-evolutionary information (Jing *et al.*, 2019).

Global statistical approaches, such as direct coupling analysis (DCA), were introduced to solve the transitive interaction problem (Jing *et al.,* 2019). DCA-based methods achieved a breakthrough in the contact prediction field as they could distinguish between the direct and indirect correlation between residues, which helps to improve the accuracy of contact prediction (Zhang *et al.,* 2021). These methods used various statistical inference algorithms to analyse co-evolutionary data into direct and indirect correlations between pairs of residues. An example of DCA-based methods is mfDCA (Morcos *et al.,* 2011), which was developed by combining mean-field approximation of DCA with covariance analysis of co-evolutionary data. This method demonstrated its ability to detect strong correlations between distant residue pairs in a more significant number of domain sequences (Morcos *et al.,* 2011; Morcos *et al.,* 2014).

Another global statistical method used sparse inverse covariance estimation, a graphical inference technique. Jones *et al.* (2012) used this technique to develop the PSICOV method. In PSICOV, sparse inverse covariance estimation eliminated indirect correlations by eliminating their values and keeping the direct correlation values. This method, thus, renders the correlation matrix sparser and more straightforward to understand, allowing for the more accurate identification of residues that coevolve (Jones *et al.,* 2012; Jing *et al.,* 2019). A noteworthy study demonstrated that a maximum entropy model deduced meaningful co-evolutionary signals from random correlations. This statistical approach was used to calculate "couplings" between residue pairs on protein sequences by analyzing patterns of similarity (homologues) across various proteins. These couplings represent the strength of co-evolution between residue pairs; if two residues have a strong coupling, they are likely close to each other in the 3D shape of the protein. Notably, the strength of these inferred couplings was established to be an exceptional predictor of the proximity of residues in folded protein structures. When the pairs of residues with the highest coupling scores were examined, they were accurately and evenly

defined in the 3D protein fold (Marks *et al.*, 2011).

Correlated mutation-based methods have demonstrated their benefits in capturing long-range interactions, analysing sizeable MSAs, and achieving high accuracy and precision in contact prediction (Weigt *et al.*, 2009; Marks *et al.*, 2011; Morcos *et al.*, 2011; Jones *et al.*, 2012). However, the statistical approaches were insufficient to identify contact information between residues because of their inability to extract a precise mutation correlation between pairs of protein residues. In addition, traditional methods for contact prediction have been dependent on the existence of homologous sequences in protein databases, and the accuracy relied on the number of aligned sequences (the alignment depth) (He *et al.*, 2017; Pearce and Zhang, 2021b; Zhang *et al.*, 2021). Therefore, many researchers have sought to exploit the advantages of machine learning to improve the accuracy of contact prediction methods.

1.9.2 Machine Learning Algorithms in Contact Prediction Methods

Machine learning approaches are computational algorithms that adapt a fitted model for detecting meaningful patterns within data. In the contact prediction field, they learn to identify contact networks among residues through their properties from protein sequences and structural data. Machine learning methods are trained from protein structures by creating contact maps based on known coordinates. Protein sequence features are then fed into algorithm models, such as support vector machines, neural networks, and random forests, which are trained to predict the contact maps (Figure 1.6). Many machine learning-based contact prediction methods are freely accessible; some have web interfaces, and others are provided as downloadable binaries and/or open-source code (Table 1.1). The output of the machine learning models typically consists of lists of scores (or p-values) for pairs of contacting residues, which inform users how likely each residue pair is in contact. These models can combine large sets of protein features and learn from them, which makes them less dependent on the depth of

MSAs and thereby reduces the prediction accuracy when fewer homologous sequences can be identified (Wu and Zhang, 2008; Xue, Faraggi and Zhou, 2009; He *et al.*, 2017; Greener *et al.*, 2022). In this chapter, we will refer to the accuracy of machine learning approaches to predict long-range contacts as being L/5 for template-free targets or those that follow free-modelling approaches (FM).

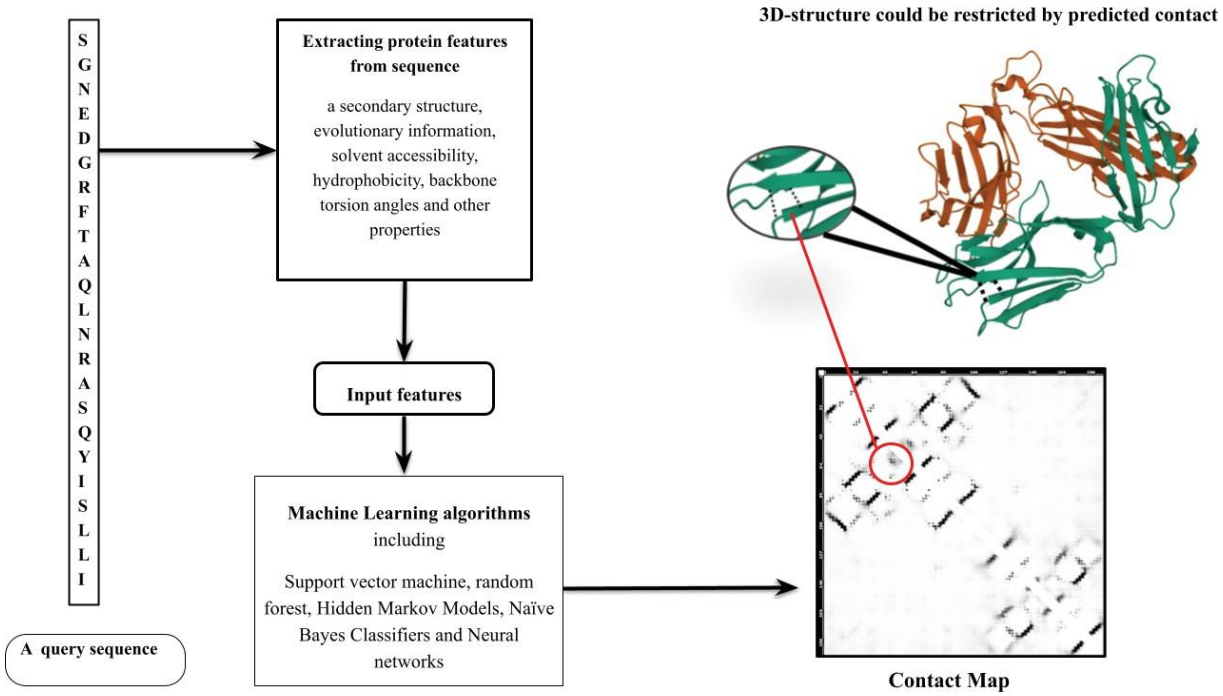


Figure 1.6. General schematic of contact prediction procedure. Starting with extracting protein features from a query sequence. These features are input data fed into machine learning algorithms to predict a contact between each residue pair. The output is a contact map of the query sequence, which can be used to aid 3D-structure prediction. The example query protein is TCR-017 ectodomain PDB ID: 7EA6. The contact map was predicted by RaptorX-Contact (Wang, Li, *et al.*, 2017; Wang, Sun, *et al.*, 2017; Wang, Sun and Xu, 2018; Xu, 2019; Xu and Wang, 2019; Xu, McPartlon and Li, 2021), and the 3D structure was visualised by Mol*Viewer (Sehnal *et al.*, 2021). The figure taken from (Alharbi and McGuffin, 2023).

Table 1.1. The available contact prediction methods based on machine learning algorithms.

Methods	Brief description	URL or web interface	Citation
DEEPCON	Deep learning-based method using covariance and sequences features as in DNCON2 and DeepCov and integrating these features into four models of fully residual convolutional neural networks with dropout layers and dilated convolution layers.	https://github.com/ba-lab/DEEPCON	(Adhikari, 2020)
DeepConPred2	The second version of DeepConPred is developed based on three models: the first and second models are deep belief networks, and the third model is a ResNet.	https://github.com/THU-gonglab/DeepConPred2	(Ding *et al.*, 2018)
SPOT-Contact	A deep learning-based method designed based on Recurrent neural networks with LSTM cells and input features predicted from SPIDER3, CMMPred and DCA.	https://sparks-lab.org/server/spot-contact/	(Hanson *et al.*, 2018)
SVMcon	The method was developed based on a support vector machine with many features.	https://multicom-toolbox.mu.hekademeia.org/SVMcon%201.0.html	(Cheng and Baldi, 2007)
DNCON2	The deep learning-based method improved by predicting protein features from PSIPRED, SCRATCH, CCMpred, FreeContact and PSICOV, which were fed into two-level CNN, where the first level had five CNNs and the second one has one CNN.	https://github.com/multicom-toolbox/DNCON2	(Adhikari, Hou and Cheng, 2018)
RaptorX-Contact	Deep learning-based method was developed by designing two ResNets models for integrating 1D and 2D protein features.	http://raptorx.uchicago.edu/ContactMap/	(Wang, Sun and Xu, 2018)
ResPRE	A method developed by integrating a precision matrix into fully residual convolutional neural	https://zhanggroup.org/ResPRE/ https://github.com/leeyang/ResPRE	(Li, Hu, *et al.*, 2019)

	networks.		
MapPred	Developed by combining two methods, DeepMSA and DeepMeta, into a dilated residual neural network model.	https://yanglab.nankai.edu.cn/MapPred/	(Wu, Peng, *et al.*, 2020)
DeepMetaPSICOV	Developed based on a deep, fully convolutional residual neural network with a set of features predicted from PSICOV, MetaPSICOV, PSICOV, CCMpred and FreeContact.	https://github.com/psipred/DeepMetaPSICOV	(Kandathil, Greener and Jones, 2019)
TripletRes	Deep learning-based method was developed by integrating three coevolutionary matrices into a residual neural network model.	https://zhanggroup.org/TripletRes/	(Li *et al.*, 2021a)
NeBcon	Developed by designing a naïve Bayes classifier (NBC) to combine eight contact prediction methods, then the NBC output with other features were fed into a neural network model.	https://zhanggroup.org/NeBcon/	(He *et al.*, 2017)
SVMSEQ	A machine learning-based method was developed to predict contact maps based on SVM software.	https://zhanggroup.org/SVMSEQ/	(Wu and Zhang, 2008)
DeepDist	Developed to predict real-value inter-residue distances based on four models of ResNet.	https://github.com/multicom-toolbox/deepdist	(Wu *et al.*, 2021)
DeepECA	Developed based on an end-to-end learning neural network to predict contact maps directly from MSAs.	https://github.com/tomiilab/DeepECA	(Fukuda and Tomii, 2020)

1.9.2.1 Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical model that estimates hidden events using observable events. HMMs comprise one category of machine learning algorithms that have been used broadly in structural bioinformatics. In the protein structure prediction field, HMMs have been applied for fold recognition pipelines and have been used to enhance performance since CASP2 (Björkholm *et al.*, 2009). FragHMMent is a HMM-based residue-residue contact prediction tool (Stecking and Schebesch, 2005). The HMMs have been applied to detect local protein neighbourhoods that include all inter-residue contacts at different ranges (short-, medium- and long-range) (Stecking and Schebesch, 2005). To this purpose, Björkholm *et al.* (2009) used local descriptors of protein structures to identify local neighbourhoods of amino acids. The local structural descriptors comprise all residues in the neighbourhood's area of desired residue pairs. These descriptors, in turn, were used to construct multiple backbone segments arranged close together. Hence, The HMMs were trained by combining sequence signals in structurally similar neighbourhoods, with two protein features derived from the secondary structure and evolutionary information to create a predicted contact map. It is worth mentioning that the identification of long-range contacts is particularly difficult for ab initio structure prediction. Interestingly, FragHMMent has proven to be particularly accurate for proteins with novel folds and is mostly fold-independent, and thus may be useful in this difficult application field (Stecking and Schebesch, 2005; Jing *et al.*, 2019).

1.9.2.2 Support Vector Machines

A Support Vector Machine (SVM) is a classification algorithm that maps high-dimensional input as vectors into nonlinear and linear models to solve binary classification problems (Zhao and Karypis, 2003; Cheng and Baldi, 2007). The performance of machine learning in contact

prediction depends on the feature sets and model designs used. Feature sets can be embedded into SVM models in order to classify residues that are either in contact or non-contact in protein structures, and more sufficient input features can be used to program such models to explore the contact patterns between residues (Horner, Pirovano and Pesole, 2007). As previously mentioned, contact prediction accuracy is often associated with the quality of the MSA analysis. Furthermore, it may be dependent on the secondary structure prediction accuracy and the frequency of β-sheets (Cheng and Baldi, 2007). A key advantage of SVMs is that they integrate linear and nonlinear methods: they can be used to design nonlinear models by representing input data nonlinearly into feature space while simultaneously classifying input dots in feature space utilising linear methods (Cheng and Baldi, 2007). Cheng and Baldi (2007) exploited this benefit when creating their contact prediction method SVMcon. they used an SVM model with a large set of protein features, including secondary structure, mutual information, solvent accessibility, and the global and local features of amino acid residues (Cheng and Baldi, 2007; Horner, Pirovano and Pesole, 2007). Another contact prediction method is SVMSEQ, which employs an SVM with two windows to predict protein residue contacts. The first window comprises local window features, including three protein features: position-specific scoring matrices (PSSMs), secondary structure predictions and solvent accessibility predictions. The second window comprises in-between segment feature sets involving sequence separations, which are the number of residues separating an interesting residue pair, the secondary structure content, the distribution of residues between residue pairs predicted to be in contact and the local properties of five residues distributed evenly in the middle of a desired residue pair (Wu and Zhang, 2008). SVM-based methods improved the accuracy of contact prediction for template-free and template-based modelling targets by approximately 25–40 % (Björkholm *et al.*, 2009). They have also been integrated with other methods into other server pipelines, such as the Yang-Server and Zhang_Contact server, which

were ranked as top-performing methods in CASP12 (Schaarschmidt *et al.*, 2018).

1.9.2.3 Random Forest Algorithms

Random Forest (RF) is a model constructed by merging several decision tree algorithms to obtain a final decision based on the most "votes". RFs are often used because they can solve a variety of problems at once, making them particularly suited to dealing with large, high-dimensional datasets and identifying noisy input information. They can also be used to build classification models rapidly (Li, Fang and Fang, 2011; Zheng *et al.*, 2012). A standard RF encompasses a set of classification models ("trees"), each of which creates a classifier and "votes" for one of the two classes (positive or negative) (Zheng *et al.*, 2012). Once it has been designed to consider predicting residue-residue contact as a classification problem, an RF model can be trained to identify residue pairs as being in contact (positive) or non-contacting (negative).

RF models have been used and incorporated with other algorithms to predict residue-residue contact maps. When using a sufficient dataset to derive protein properties, an RF model can extract accurate contact information from known protein structures. The first RF-based method for contact prediction was ProC_S3, developed by Li *et al.* (2011). The RF model they used constructed 500 classification trees for the training and prediction stages and was trained on a large dataset including 1,490 protein structures and feature sets, which considered "the average of [the] maximum accessible surface areas and isoelectric points of the amino acids in two local windows (four features) [and the] f-mean of the between segment (20 features and [the seven] features of the central residue of the segment" (Li, Fang and Fang, 2011, p. 3383). Since ProC_S3 was based on an RF algorithm, it acquired selected features which could determine the relevance of protein features to residue contacts (Li, Fang and Fang, 2011). To investigate the usefulness of this feature selection, the RF-based method TMhhcp was designed to predict contacts in alpha-helical transmembrane proteins based on all their features and the selected

feature set (Wang *et al.,* 2011). All features constructed from the evolutionary profiles of the residue pairs included the TM helix numbers, residue distance in the sequence, relative distance of two residues in between two helices, residue conservation scores and correlated mutation scores calculated by covariance algorithms, resulting in 408 feature vectors. From this construction, 10 feature subsets have been selected by using the correlation-based feature selection (CFS) (Wang *et al.,* 2011). The selected features experiment was conducted to identify a range of distinguishing features which have a greater individual capacity to predict the class (contact) but minimal inter-correlation (Wang *et al.,* 2011). Two models, named TMhhcp1 and TMhhcp2, were built based on training data with all the features, and two others, named TMhhcp_cfs1 and TMhhcp_cfs2, were built with selected features. In the latter two models, three protein features were found to produce particularly accurate contact predictions: the residue separation in the main sequence, the relative distance between two residues in helix-helix interaction and the correlated mutation score (Wang *et al.,* 2011). Other algorithms incorporating RF models included PhyCMAP, combining an RF model and an integer linear program, which predicts contact maps by integrating evolutionary and physical constraints (Wang and Xu, 2013; Zhang *et al.*, 2016). In general, RF-based methods have demonstrated reliable improvement with regard to the accuracy of contact prediction.

1.9.2.4 Naïve Bayes Classifiers

A Naïve Bayes Classifier (NBC) is a simple probability classifier based on the assumption that each feature value has an independent effect on a particular class. NBCs improve the accuracy of contact prediction in a complementary way for proteins which lack homologous sequences. NeBcon is a meta-server for contact prediction, which combines a Bayes classifier and a neural network to predict an accurate contact map by exploiting coevolutionary features and machine learning-based contact methods (He *et al.*, 2017; Peng, Zhou and Zhang, 2022). An NBC was

used in this server to compute the contact probability scores of eight contact prediction methods; it was also given a set of posterior probability values for predicted contacts. Subsequently, the output of the NBC, along with six structural features extracted from the target protein sequence, was fed into a neural network model to predict the final contact map (He *et al.,* 2017). The improvement of the performance of NeBcon was attributed to the integration of the complementary coevolutionary information from eight methods into the NBC model and the structural features using neural networks (He *et al.,* 2017). He *et al.* (2017) demonstrated that the combination of machine learning-based methods with coevolution methods into NBC model improved the accuracy of contact prediction from hard targets, which tends to have low accuracy predicted contacts by coevolution methods.

1.9.2.5 Neural Networks

Neural Network (NN) is an artificial neural network composed of a number of computing units known as neurons. These units are linked together by connections, each of which has a weight attached to it (Hapudeniya, 2010). NNs have had a considerable impact on the advancement of machine learning and on the accuracy of contact prediction methods. One of the first applications of NNs to the problem of contact prediction was when Fariselli and Casadio (1999) used them to extract the relationship between contact maps and the chemical interaction between protein residues. The NN had a high level of adaptability through the combination of different input features such as secondary structure prediction, chemicophysical properties of residues and evolutionary features extracted from MSA in its first layer, leading to adequate learning to increase its prediction power (Fariselli and Casadio, 1999; Shackelford and Karplus, 2007). Shackelford and Karplus (2007; cited in Wu and Zhang, 2008) demonstrated that NNs could play a vital role in improving contact prediction accuracy by integrating several protein features and training on large data sets. This confirmed the observation by Fariselli *et al.*

(2001), who showed that the performance of NNs in predicting a contact map improved when the amount of input data was raised. In prior NN models, input features, including protein sequences and predicted secondary structures, mutational information from MSAs, and hydrophobicity scores, were investigated for their importance in improving contact prediction accuracy through the design of different NN models, which included different input data (Fariselli and Casadio, 1999; Fariselli *et al.*, 2001; Liu *et al.*, 2005; Shackelford and Karplus, 2007). Fariselli and Casadio (1999) had initially demonstrated that protein features can improve the accuracy of contact prediction if they are combined using NNs. They subsequently showed that evolutionary information from structure-sequence alignments can provide accurate predicted contacts for proteins with less than 170 residues, while the sequence context, which are five potential couplings for each residue into parallel and antiparallel pairings encoded into three-amino-acid window, plays a role in the accuracy of contact prediction for proteins with sizes more than 170 (Schneider, De Daruvar and Sander, 1997; Fariselli and Casadio, 1999). The accuracy of contact prediction is computed by dividing the correctly predicted contacts by the total predicted contacts. Each protein's accuracy is evaluated separately before being averaged throughout the whole protein dataset (Fariselli *et al.*, 2001). By integrating a variety of protein features, contact prediction accuracy achieved a more reliable value (21 % of average accuracy in CASP3) (Fariselli *et al.*, 2001), however, alternative NN models have since been developed to further improve the accuracy. For example, Xue *et al.* (2009) developed SPINE-2D by designing a deeper NN model with two hidden layers to extract information from residue solvent accessibility and backbone torsion angle features, resulting in increasing average contact accuracy at 26 % in CASP8. Although this NN improved upon previous neural network-based methods, its accuracy did not achieve a sufficient level to be used for confidently modelling tertiary structures. Therefore, researchers were encouraged to employ deeper NN models, including residual convolutional neural networks (ResNets), recurrent

neural networks, and end-to-end learning models, which will be discussed in the next section.

1.9.2.5.1 Deep Neural Networks

Deep neural networks are complex architectures of NNs designed to obtain extensive knowledge from high-level data. Deep models of NNs differ from shallow models in terms of architectural construction. While shallow architectures are constructed from two layers (input and output) and a small number of hidden layers, deep models are designed from "deep stacks" of classical NNs with a large number of hidden layers (Fariselli and Casadio, 1999; Xue, Faraggi and Zhou, 2009; Torrisi, Pollastri and Le, 2020). Deep learning-based methods often perform better when the depth of NN layers is increased, enabling them to extract accurate information from very large datasets with numerous input features (Jing *et al.*, 2019). Since 2008, deep neural networks have been employed in contact prediction methods and have led to improvements in their accuracy. NNcon was an early deep learning-based contact prediction method designed with a 2D recursive neural network for predicting tertiary and secondary structure contacts (β-sheet). In CASP8, NNcon was ranked as one of the top-performing methods (Tegge *et al.*, 2009; Jing *et al.*, 2019). Later, in 2012, Di Lena *et al.* (2012) designed "a 3D of stack of neural networks" that could extract contact information, where each stack consisted of three NN layers (one input, one hidden, and one output). This method improved the accuracy of contact prediction by nearly 30 % (from 28 % to 35 %), which indicated that deep NNs could learn more efficiently than shallow NN models (Di Lena, Nagata and Baldi, 2012; Jing *et al.*, 2019). Due to the rapid development in Graphics Processing Units (GPUs), researchers have been focusing on improving deep neural network models by exploiting the increasing GPU card capabilities, which have allowed efficient training on big data sets with complex NN architectures. Another approach was developed by Eickholt and Cheng (2012) who designed DNcon, which uses several deep models from restricted Boltzmann machines,

combined with the boosting ensemble method. Restricted Boltzmann machines are neural networks with two layers, visible and hidden, with symmetric weights connecting the nodes of both layers. Several restricted Boltzmann machines were combined to construct deep networks (DNs). Many layers were added to each DN in the boosted ensemble model, and DNs were then trained in a stepwise, semi-supervised manner (Eickholt and Cheng, 2012; Eickholt and Cheng, 2013). They attributed the improved prediction performance of DNcon to its use of a feature set with a deep architecture (Eickholt and Cheng, 2012; Eickholt and Cheng, 2013; Jing *et al.*, 2019). Many alternative methods have been developed, which employ various deep model architectures with different input data sets, and have shown improvements in contact prediction performance over successive CASP experiments (Wang, Sun and Xu, 2018; Ruiz-Serra *et al.*, 2021; Zhang *et al.*, 2021). A different approach has also been taken in recent years after MSA analysis tools were developed. These tools help to enhance alignment approaches for extracting correlated mutation features, which is what contact prediction methods have often depended on. The new feature extraction approach uses mutual information predicted from coevolution methods as input data for deep neural networks (Monastyrskyy *et al.*, 2016). The success of this approach was demonstrated when Jones *et al.* (2015; cited in Torrisi, Pollastri and Le, 2020) designed MetaPSICOV by integrating the PSICOV, FreeContact, and CCMpred methods with a two-stage neural network model. The CASP11 evaluations revealed that MetaPSICOV outperformed all contact prediction methods, and the accuracy of contact prediction exceeded 30 % (Monastyrskyy *et al.*, 2016). Researchers were inspired by MetaPSICOV and developed their own servers through the inclusion of coevolutionary features derived from MSAs and structural properties in a variety of deep neural networks. Consequently, a milestone was achieved when the accuracy of contact prediction in CASP12 reached 47 % (Schaarschmidt *et al.*, 2018).

1.9.2.5.1.1 Residual Convolutional Neural Networks

A convolutional neural network (CNN) is a deep neural network comprised of varied layers (or filters), in which each layer has neurons with larger local receptive fields than the previous one. The input matrix of the CNN model is divided into submatrices, and each submatrix is filtered by each local receptive field to encode the local map. The output consists of multiple local maps, and this operation is called convolution. Following that, the pooling operation, which consists of pooling submatrix values from the convolution output into single values, results in size minimisation. Eventually, the classification stage operates in the last layers of CNN and transforms the output probabilities to a range between 0 and 1, with the sum equal to 1 (Jisna and Jayaraj, 2021).

CNN models often have the problem of overfitting because each neuron in each layer is fully connected with all neurons of the previous layer, which ultimately reduces the accuracy of these models. To fix this problem, a skip connection is applied by designing two residual blocks between layers, creating a residual neural network (ResNet). ResNets have been used in protein contact predictions because of the consistent spatial regularity of amino acid residues on the protein sequence. A ResNet can apply the same local filters over all residue positions by requiring a limited number of weights to be adjusted in relation to the input layer and the next layer's dimensionality. This leads to improved computational implementation and output accuracy (Jisna and Jayaraj, 2021; Pakhrin *et al.,* 2021). Therefore, protein sequence alignments can be analysed by ResNets to predict contact maps with far higher accuracy above 40 % (Schaarschmidt *et al.,* 2018).

As with previous machine learning approaches, a ResNet needs a large set of features to extract accurate contact patterns between protein residues. However, designing the best architecture for certain features is the key to improving contact prediction accuracy and therefore method performance (Kuhlman and Bradley, 2019). Wang *et al.* (2018) developed RaptorX-Contact

with a deep model consisting of two ResNet modules. The first module was designed to be one-dimensional (1D) to learn from 1D protein features, including sequence profile, predicted secondary structure, and solvent accessibility. The second module was built in 2D representation to learn from 2D pairwise properties. To extract contact patterns, the output of the 1D module was converted to a 2D matrix and fed into a 2D module simultaneously with the pairwise features. In the last step, the probability values of contact prediction are computed by integrating the output of the second module into logistic regression (Wang, Sun and Xu, 2018; Pearce and Zhang, 2021a). This designed model has the ability to capture contact existence between residues from the complex protein features, increasing contact prediction accuracy (to 47 % in CASP12) substantially. RaptorX-Contact was independently benchmarked for the first time in CASP12 and was among the top-performing contact prediction tools (Wang, Sun and Xu, 2018; Xu, 2019).

ResNets have been employed in various contact prediction servers with various input protein features and architectures, and most are designed to analyse the MSAs resulting from searches of massive sequence databases. In CASP13, Kandathil *et al.* (2019) developed DeepMetaPSICOV, a method that improved the accuracy of protein contact prediction by combining multiple protein properties with a deep convolutional residual network and using MSAs from a large sequence dataset.

Another contact-map predictor, ResPRE, was developed by combining coevolution-derived precision matrices that improved the analysis of MSA using deep ResNets (Li, Hu, *et al.*, 2019; Zheng *et al.*, 2019; Jisna and Jayaraj, 2021). Along with other methods, ResPRE was integrated into the meta-predictor method called NeBcon (described in the previous section). In this predictor, the confidence values of predicted contacts from these methods were fed into an NBC. The output of NBC was integrated with different sequence data in 350 units of a hidden layer connected to NN to refine the contact prediction model (Zheng *et al.*, 2019). The

combination of ResNets with other machine learning methods and the use of large sequence datasets to derive the input MSAs has helped to greatly improve the predictive performance, so much so that the performance of deep learning-based methods raised the accuracy of contact prediction to 70 % in CASP13 (Shrestha *et al.*, 2019).

Deep residual convolutional networks achieved success with other aspects of residue-residue contact prediction and the development of alignment techniques. Recent studies demonstrated that residue-residue contacts predicted as binary classification provide restricted information. On the other hand, predicting the actual distance between residues produces more precise information, and ResNet models that are trained in the universal network of inter-residual distances allow for the capture of higher-order residue relationship (Li *et al.*, 2021a; Li *et al.*, 2021b; Ruiz-Serra *et al.*, 2021). Thus, the cutting-edge contact prediction methods now expand to predict distances in their pipelines. For example, DeepPotential was developed by modifying the deep ResNet by adding 10 residual blocks as 1D and 2D representations for predicting inter-residue contacts within different ranges of distances. Additionally, to predict all inter-residue interactions, another fully ResNet was fed by the outputs of 1D and 2D ResNet and trained by cross-entropy loss. The inter-residue distances considered in this method are side chain contact, backbone contact, torsional angle, and hydrogen-bond interaction at different distance thresholds, ranging from 2 to 10, 13, 16, and 20 Å (Li *et al.*, 2021b; Zheng *et al.*, 2021).

TripletRes is ranked as a top-performing method in the most recent CASP experiment (Ruiz-Serra *et al.*, 2021). In this method, Zhang's group first developed an alignment strategy to improve the quality of MSA inputs for coevolutionary information extraction. The strategy was to construct a deep MSA using several rounds of HHblits, then extract "covariance features (COV), precision matrix features (PRE), and a coupling parameter matrix approximated by pseudolikelihood maximization (PLM)". These features were combined into the ResNets model with four sets of residual blocks and trained by loss function examining a discrete map

of the distance information between each residue pair (Pakhrin *et al.*, 2021). Li *et al.* (2021a) demonstrated that one of the success factors of TripletRes was incorporating deep neural networks with the three sets of coevolutionary features, which enabled the capture of more accurate contacts. This indicates that ResNets has made an invaluable contribution to the accuracy of contact prediction methods, whether or not they were used in conjunction with the prediction of absolute distances.

1.9.2.5.1.2 Recurrent Neural Network

Recurrent neural network (RNN) is an advanced architecture of neural networks in which nodes are connected in a recurrent pattern to process sequential data (Graves, Fernández and Schmidhuber, 2007). RNNs have been employed to predict protein secondary structures and they have been designed to extract these structural features from MSA data. The RNN was used to predict coarse contact and orientation of secondary structures, while a further deep neural network architecture was then used to generate final, more refined contact predictions (Di Lena, Nagata and Baldi, 2012; KC, 2017; Jing *et al.*, 2019). For contact map prediction, two-dimensional, bidirectional, recurrent long short-term memory (2D-BRLSTM) networks have been employed with residual convolution neural networks for the SPOT-Contact method. The SPOT-Contact method was designed to combine 2D-RNNs with long short-term memory (LSTM) cells. LSTM cells can learn the complicated context of long-range contacts between residues for the whole protein sequence, while 2D-RNNs can generate an accurate model because of their capacity to identify misleading data in all input variables (Hanson *et al.*, 2018; Jisna and Jayaraj, 2021). SPOT-Contact is ranked as one of the top-performing methods in CASP13 according to the independent blind evaluation of contact prediction methods (Wu Peng, *et al.*, 2020). The improvement in mean accuracy of neural network-based contact prediction methods during the CASP experiments can be seen in Figure 1.7.

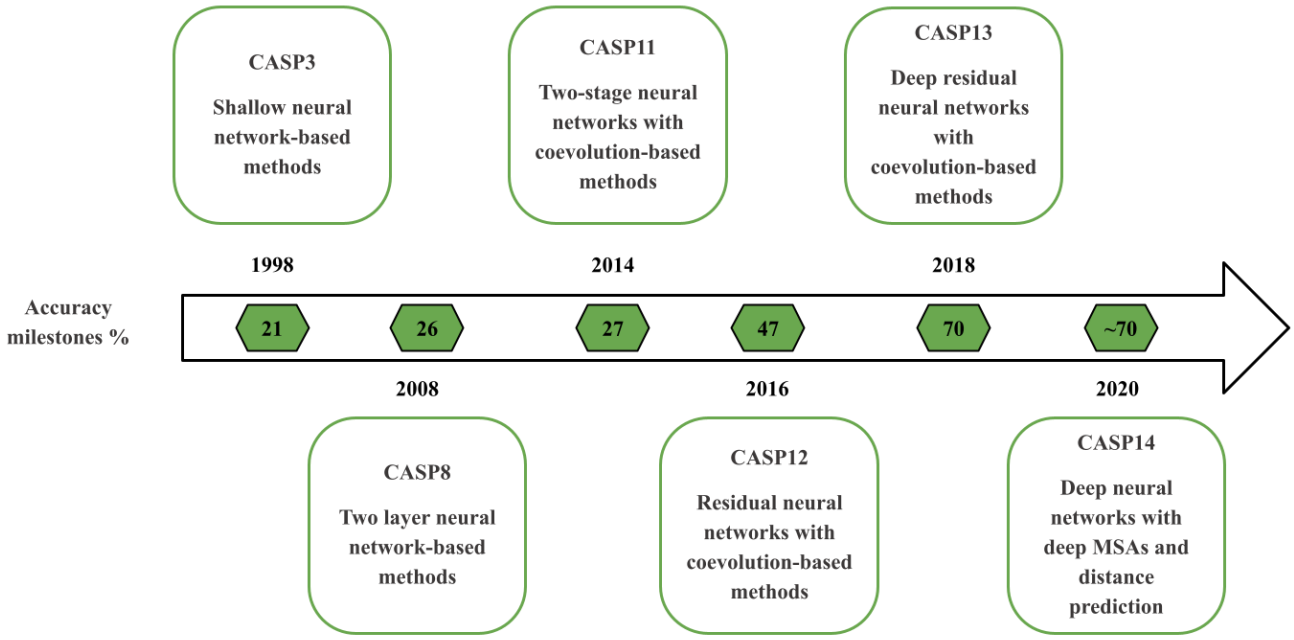


Figure 1.7. Timeline for the development of neural network-based methods and their average accuracy based on the CASP evaluation procedures. The accuracy of contact prediction is based on L/5 long-range contacts for FM targets.

1.9.2.5.1.3 End-to-end Learning Models

The previously discussed deep neural networks are built with multiple layers and successful at detecting coevolution-based features from MSAs, but they gain no information concerning the natural relationship between sequence and structure for proteins if no sequence homologs can be detected. To fix this issue, another class of methods was developed, which use end-to-end differentiable deep learning models based on an explanatory structure-to-sequence maps (AlQuraishi, 2019). In such methods, end-to-end differentiability indicates the ability to use a single approach to optimise a sophisticated multi-stage pipeline from input to output without relying explicitly on coevolutionary information, in which the whole prediction process is represented by a single deep neural network (AlQuraishi, 2019; Jisna and Jayaraj, 2021). Thus, end-to-end approaches have been employed to enhance contact prediction accuracy in the absence of deep MSAs. The method DeepECA was developed by Fukuda and Tomii to predict contact maps from both deep and shallow MSAs directly in a single neural network. With the availability of homologous sequences, correlated information can be extracted using a covariance matrix (COV), then the coevolution values from this matrix can be used as input for the deep neural network model. The model used was a 1×1 CNN using end-to-end learning to weight each sequence of an MSA, which helped to eliminate the noisy information from the abundant sequences. The weighting process in the end-to-end model was used to optimise the quality of the MSA analysis and provided the most relevant homologous sequences to the target protein sequence. This method showed an improvement in the contact prediction accuracy, even though predictions were made directly from an MSA alone without any other encoded features. In the case of shallow MSAs, the accuracy of contact prediction can be increased by adding other protein features with correlation information in the CNN model extended to a multi-task model (Karplus *et al.,* 1997; Vaz and Balaji, 2021). It is worth noting that employing an end-to-end model to improve the procedure of extracting pure and accurate mutation

information not only enhances the predictive power of contact predictions for tertiary structures, it may also be used for the currently unresolved problem of predicting protein-protein interactions (Laine *et al.*, 2021).

1.10 Research Objectives

Our research aims to explore the potential benefits of contact prediction in enhancing the accuracy of protein tertiary structure prediction. In this chapter, we provide a comprehensive literature review, discuss the advancements made in protein structure modelling, and emphasize the role of contact prediction in this process. We explain the concept of contact prediction, provide an overview of the latest contact prediction methods, and discuss their accuracy as evaluated by the CASP experiments. Additionally, we elaborate on the traditional and advanced approaches employed to improve the accuracy of contact prediction. Finally, we demonstrate the application of contact prediction methods in other relevant areas, such as model quality estimation and refinement.

1.10.1 Improvement of Deep Learning-based Contact Prediction Methods using Consensus Approaches

In chapter 2 we address our initial objective, which is to improve the accuracy of contact prediction further. To achieve this objective, we utilize the consensus approach, which has demonstrated promising potential in boosting the accuracy of protein structure prediction tools. Our study tests the benefits of the consensus approach to enhance the accuracy of contact prediction. We choose this method because it provides confident results, reduces errors in prediction data, and obtains accurate outcomes by combining the strengths of various methods. For our computational study, we select six top-performing methods in contact prediction based on the assessment in CASP13 and CAPS14. We use these six methods to design the consensus-based methods using the mean scores in two stages: consensus of two methods and consensus

of three. We then compare and assess the consensus-based contact prediction methods with individual methods. The evaluation findings in Chapter 2 highlight the advantages and disadvantages of consensus approaches in improving the accuracy of contact prediction methods.

1.10.2 Development of Consensus CDA scores for Model Quality Estimates

With the advancement in modelling methods addressing the single chain prediction problem, the focus has shifted to developing model QE methods that assess the local regions of high-quality models of tertiary protein structures. The study's objective in Chapter 3 is to test the usefulness of consensus contact prediction in improving the local assessment performance of ModFOLD9, a quality estimation method for tertiary structure models. The study is conducted in response to the developments in the protein structure prediction field, which has highlighted the difficulty in evaluating high-quality 3D models of tertiary structures. To integrate contact prediction into the scoring estimation system, we use a pure-single model quality estimation approach based on Contact Distance Agreement (CDA) scoring. We aim to derive six new CDA scores from six contact prediction methods and combine them using two versions of a Multilayer Perceptron (MLP) neural network to apply a new consensus approach. The two versions of the MLP are trained to learn from CDA-based contact prediction score inputs and predict two local quality scores: the S-score and the lDDT score. The MLP hyperparameters are fine-tuned to optimize performance. The approach is tested and trained on CASP14 data, and its performance is evaluated using correlation and ROC analysis.

1.10.3 Development of Consensus QA Methods for The ModFOLD9 Quality Estimation Server

The next objective is to investigate the integration of further scores from two types of quality scoring methods: pure-single and quasi-single model methods. These methods excel at estimating the local accuracy of 3D models, and leveraging their advantages in our ModFOLD9 server could lead to better local estimation performance. The study conducts a two-stage computational study. The first stage involves the consensus of six CDA scores integrating quality scores from pure-single methods, and the second stage involves combining the scores of the first stage with additional quality scores of quasi-single methods. Again, here we use two versions of the MLP to combine the scores and train them to predict either the S-score or the lDDT score. To optimise the MLP predictive performance, we implement fine-tuning in the two stages. We evaluate the consensus quality scores' performance in improving the accuracy of ModFOLD9's local quality assessment against established methods, using a similar evaluation as in Chapter 3.

1.10.4 Benchmarking of ModFOLD9 and ModFOLDdock Performance during the CASP15 Experiment and using the CAMEO Resource

Our final objective is to investigate how ModFOLD9 contributes to the predictive capabilities of our servers from parallel projects. Our approach is to utilize ModFOLD9 as the accuracy self-estimate server for the IntFOLD7 3D models submitted to the CASP15 experiment. In addition, we incorporate a similar consensus approach and integrated contact prediction data into our new ModFOLDdockS method for evaluating quaternary structure models in CASP15. To further evaluate the effectiveness of the ModFOLD9 enhancements, we also conduct extensive tests using the CAMEO resource. Chapter 5 presents a comprehensive analysis of the performance of the improved servers based on data obtained from these two independent blind tests (CASP and CAMEO). Through this evaluation, we can determine the performance

of our IntFOLD7 and ModFOLDdockS servers at CASP15, as well as the extent of the improvements achieved in CAMEO compared with previous versions of ModFOLD because of these enhancements.

Chapter 2 Improvement of Deep Learning-based Contact Prediction Methods Using Consensus Approaches

2.1 Introduction

The advances in contact prediction technologies have led to a significant increase in contact prediction accuracy. More specifically, the predictive performance of contact prediction methods has been improved in three major ways: MSA construction, input features, and a deep model of neural networks. MSA construction techniques have been developed along with the exponential growth in protein databases, allowing for the extraction of abundant coevolution information (Ovchinnikov *et al.*, 2017; Ovchinnikov *et al.*, 2018; Kandathil, Greener and Jones, 2019; Zheng *et al.*, 2019; Wen *et al.*, 2020; Wu, Peng, *et al.*, 2020; Zhang *et al.*, 2021); here protein structure and sequence properties are derived from known structures and used as input features (Adhikari and Cheng, 2016; Reza *et al.*, 2021; Zhang *et al.*, 2021). Deep neural network models are used to infer the contact distribution between protein residues from evolutionary-based data with enough input features; these data can assist deep neural networks in improved training (Jing *et al.*, 2019; Shrestha *et al.*, 2019; Zhang *et al.*, 2021). This advancement results in the production of highly accurate contact map matrices of the target proteins.

Contact prediction methods differ in terms of the advanced algorithms used in MSA analysis and the designed deep model of neural networks, as well as the use of distance and orientation prediction. Prediction methods have employed one or two out of three different types of statistical matrices to analyse MSA. These matrices are the precision matrix, the COV matrix and the pseudolikelihood maximisation of the Potts model (PLM) (Li, Zhang, *et al.*, 2019; Suh *et al.*, 2021). The first two matrices have been used to capture regional coevolutionary patterns between two residue positions (Li *et al.*, 2021a). To consider all evolutionary information, these metrics were combined with the PLM, which derives the global features of other residue positions (Li, Zhang, *et al.*, 2019; Li *et al.*, 2021a). Additionally, tools have been employed in

various contact prediction methods for constructing deep MSAs, such as DeepMSA and DeepAln, which have led to further improvement in their performance (Li, Zhang, *et al.*, 2019; Zheng *et al.*, 2019; Wu, Hou, *et al.*, 2020; Zhang *et al.*, 2020; Wu *et al.*, 2021). However, the depth and number of homologous sequences influence the quality of MSA (Guo *et al.*, 2021). Some studies have demonstrated that deep MSAs can produce inaccurate data that could reduce the accuracy of contact predictions (Kandathil, Greener and Jones, 2019; Guo *et al.*, 2021). In other words, deep MSA may include divergent sequences, which can issue misalignment or a loss of protein structure information or profile drift, causing sequence mismatches. These problems could render it difficult to reliably predict the evolutionary relationship between the residues of homologous proteins, which could reduce contact prediction accuracy (Kandathil, Greener and Jones, 2019). To avoid this, other contact prediction methods have considered using shallow MSAs in their servers (Fukuda and Tomii, 2020).

The use of deep neural networks has led to significant performance gains for contact prediction methods. Different deep model designs have been used, and the most common type is ResNets (Wang, Sun and Xu, 2018; Li, Zhang, *et al.*, 2019; Li *et al.*, 2020; Li *et al.*, 2021b; Yang *et al.*, 2020; Jayaraj, 2021; Suh *et al.*, 2021). However, the capability of deep models relies on the training function and input data (Suh *et al.*, 2021). In most contact prediction methods, a binary cross-function is used to train a deep model to classify the input data into contacting and non-contacting residues (Jones and Kandathil, 2018; Kandathil, Greener and Jones, 2019; Adhikari, 2020; Yang *et al.*, 2020; Wu *et al.*, 2021). In the top-performing methods, a new loss function helps to improve the predictive performance of deep learning models using discretised distance matrix prediction (Yang *et al.*, 2020; Li *et al.*, 2021a). Input data can include evolutionary information derived from MSA, protein profiles, secondary structure prediction, solvent accessibility, and other physicochemical properties (Hanson *et al.*, 2018; Fukuda and Tomii, 2020; Li *et al.*, 2020; Li *et al.*, 2021a; Yang *et al.*, 2020). Annotation features have been

considered as input features to derive contacts, including distance and orientation prediction (Yang *et al.*, 2020; Li *et al.*, 2021a; Li *et al.*, 2021b; Peng, Zhou and Zhang, 2022). As a result of this variation in methodology, the different contact algorithms have predicted distinct contact maps with varying degrees of accuracy for different targets (Shrestha *et al.*, 2019). Hence, combining the strengths of the top-performing methods using a consensus approach may aid in achieving optimal contact prediction accuracy.

2.1.1 Contact Prediction Methods

In this section, we will present deep learning-based contact prediction methods which ranked as top-performing methods based on the assessments of two CASP experiments (CASP13, CASP14) (Shrestha *et al.*, 2019; Ruiz-Serra *et al.*, 2021). These methods include DeepMetaPSICOV (Jones-UCL group), SPOT-Contact (ZHOU-Contact group) and NeBcon (Zhang_Contact group) from CASP13, and TripletRes, trRosetta (Yang_FM group) and DeepDist2 (MULTICOM group) from CASP14. The methods are publicly available as standalone programs and were therefore chosen for designing our consensus-based methods.

2.1.1.1 Deep Learning-based Contact Prediction Methods in CASP13

2.1.1.1.1 DeepMetaPSICOV (Jones-UCL group)

DeepMetaPSICOV (DMP) was developed by Kandathil *et al.* (2019). This was an improved approach that combined two methods: MetaPSICOV and DeepCov. DMP's concept is to exploit a large context from sequence features by employing deep, fully convolutional residual networks. This method generated MSAs from sequence databases during the prediction process, leading to increased precision in contact prediction. The DMP method was divided into two stages. The first stage contained 501 channels of input features derived from MSAs to form covariance matrices (441 channels from DeepCov and 58 channels from MetaPSICOV2);

in addition, there were two channels representing sequence separation and sequence bounds. The second stage was a model of a deep, fully convolutional 77-layer residual network. In the first layer, the input features were reduced to 64 channels, and the output in the last layer was the probability of predicting each residue pair to be in contact. The final mode was predicted from five models trained in different numbers of random seeds. During the prediction process, sequence alignments were trained by applying data augmentation strategies such as loop sampling and feature interpolation, which led to improving and generalising the DMP model. This step enhanced the performance of the DMP server to predict accurate residue-residue contacts in the 3D protein structure (Kandathil, Greener and Jones, 2019).

2.1.1.1.2. SPOT-Contact (ZHOU-Contact)

SPOT-Contact was a novel method designed by the ZHOU-Contact group using two ultra-deep neural networks with sets of two input features: one set contained the sequence-based features and the other evolutionary coupling-based features (Hanson *et al.,* 2018). The neural networks include ResNets and two-dimensional bidirectional recurrent long-term short-term memory (LSTM) networks. The latter were formed through a combination of 2D-RNNs which can predict an accurate model because of their ability to distinguish misrepresented data in all input dimensions, and LSTM cells, which were capable of assembling the complex relationship context of nonlocal residues for the whole protein sequence (Hanson *et al.*, 2018).

The architecture of the SPOT-Contact was a collection of six models, where the base model comprises four components. The first component was data preparation, which was a concatenating sequence featuring an in-depth way to transform them from one-dimensional into a two-dimensional image. The second was a ResNet, which is a residual convolutional neural network that predicts the contact map from the entire protein by combining evolution coupling information with sequence features. The third was 2D-BRLSTM, formed of the

bottleneck layer and LSTM layers that contain 200 cell blocks, peaking at 800 inputs on the following layer. The last component, which was fully connected (FC), comprises 400 nodes but excludes the final layer, which consists of a single neuron with sigmoid activation. In this final layer, the output was converted into probabilities of contact prediction for each residue pair (Hanson *et al.,* 2018). In SPOT-contact, the five models include a base model, a base without bottleneck, a base without FC, 2D-BRLSTM before ResNet in the base model, and the 2D-BRLSTM model only. The input features were derived from several programs. These features included one-dimensional sequence features, which were evolutionary profiles, probabilities of predicted structure, and seven physicochemical properties for each residue. Additional features included three outputs of pairwise features, which were a contact map from CCMpred and mutual information and direct coupling from DCA methods (Hanson *et al.,* 2018).

2.1.1.1.3 NeBcon (Zhang_Contact)

NeBcon was designed by the Zhang_Contact group. It was one of three modules integrated into Zhang-Server and QUARK pipelines to predict protein contact maps before using these maps for constructing models of FM targets, leading to increased accuracy in CASP12. In CASP13, NeBcon was improved by combining nine contact prediction methods including ResPRE (Li, Hu, *et al.,* 2019), DNCON2 (Adhikari, Hou and Cheng, 2018), GREMLIN (Kamisetty, Ovchinnikov and Baker, 2013), CCMpred (Seemayer, Gruber and Söding, 2014), DeepContact (Liu *et al.,* 2018), FreeContact (Kaján *et al.,* 2014), DeepPLM, DeepCov (Jones and Kandathil, 2018), and MetaPSICOV2 (Buchan and Jones, 2018). ResPRE employs deep residual neural networks that incorporate evolutionary precision matrices for predicting contact maps. DeepPLM used the same deep learning model of ResPRE with various features derived from CCMpred (Zheng *et al.,* 2019).

After predicting contacts from these methods, their confidence scores were added into an NBC, producing the final probabilities. These scores were merged with various sequence features in 350 units of a hidden layer linked with a NN for purifying the contact prediction model, which raised its accuracy by 50% on top L/5 long-range contacts (Zheng *et al.*, 2019). This improvement increased the ability of NeBcon to predict accurate contact information, leading to its ranking as one of the top-performing contact prediction methods in the CASP13 round.

2.1.1.1.4 Contact Prediction Methods Performance in the CASP13 round

Remarkable success has been achieved from these methods in CASP13 for the category of contact prediction. The advancement of contact prediction accuracy could be attributed to deep learning models coupled with coevolutionary features derived from MSA methods (Wu, Peng, *et al.*, 2020). Deep convolutional neural networks (CNN) have been employed in all these methods in different ways. The CNN model of DMP was complex, with more than 70 layers and additional data augmentation techniques; the Zhou-Contact deep learning model was deeper than the DMP model, which was a combination of CNN with two RNNs. Zhang-Contact has its own model, but it also integrated predicted contacts produced by various deep neural network models with six contact methods. Deep learning networks can learn from coevolutionary features with other protein features for predicting accurate contact maps.

On the other hand, contact prediction methods rely on deep sequence alignment strategies that have been used to infer residue contacts. DeepCov and DCA methods have been used to extract information on a mutation from MSA in DeepMetaPSICOV and Zhou-Contact, respectively. Nevertheless, the quality of alignments could affect the accuracy of contact prediction. During the process of generating MSAs, DeepCov and DCA methods produce coevolutionary information that could not be optimised due to the noise usually created by the indirect correlation between residue sequences, which led to difficulty in distinguishing the correct

correlation information (Fukuda and Tomii, 2020). However, MSA could be generated with optimised evolutionary information; that optimisation can be done with other methods that have improved the feature design of MSA methods (Fukuda and Tomii, 2020; Wu, Peng, *et al.*, 2020). For example, NeBcon uses a ResNet, which obtained an MSA from DeepMSA. This method outperformed DeepCov regarding the quality of alignments and in generating sufficient mutual information (Wu, Peng, *et al.*, 2020). Therefore, combining Zhang-Contact with other contact prediction methods could improve the extraction of mutual information, achieving more accurate contact prediction.

2.1.1.2 Deep learning-based Contact Prediction Methods in the CASP14 round

2.1.1.2.1 TripletRes

TripletRes was a deep learning-based method developed by integrating three matrices of MSAs to calculate coevolutionary information through the PLM, a precision matrix, and a COV matrix into a residual neural network model (Li *et al.*, 2021a). In TripletRes, MSAs were constructed using HHblits with three iterations. The DeepMSA pipeline was used to generate MSAs for testing proteins, where HHblits also created the first MSA, followed by numerous iterations. Jackhmmer and hmmsearch were used to generate an MSA in cases where the number of effective sequences was less than 128. To extract information from the MSAs, three matrices were applied to derive three sets of features. The first was a COV to analyse the marginal reliance between distinct sequential coevolutionary positions of residues. COV recognises correlated marginal distributions between variables, such as 'transitional correlations' (Li *et al.*, 2021a). The second was a PRE matrix that based on the mean-field approximation of the Potts model (Li *et al.*, 2021a). The last matrix was a PLM, which was employed to estimate the likelihood of a sequence for the Potts model. These features were fed into a fully convolutional neural network with residual blocks (Li *et al.*, 2021a).

The deep model employed in TripletRes was ResNet with feedforward neural networks. The residual blocks were four sets wherein three sets were connected to input layers fed with evolutionary feature extraction. The latter three sets have 24 basic blocks that can transform each input feature into a feature map of 64 channels. These feature maps were then concatenated along their channels, and another ResNet model was implemented with 24 blocks to extract the combined information from all of these maps (Li *et al.*, 2021a). To compute the probability of each residue pair, a softmax function was activated at the final layer. The probability values were then used to estimate the contact between a residue pair into ten bins of distance ranging from 5 to 15 Å, with one bin reflecting distances of less than 5Å and another indicating more than 15Å (Li *et al.*, 2021a). To train the entire set of deep ResNets, the maximum likelihood of the prediction was set by defining the loss function as 'the sum of the negative log-likelihood over all the residue pairs in the training protein' (Li *et al.*, 2021a, p. 15).

2.1.1.2.2 trRosetta (Yang_FM)

The trRosetta method was a deep learning-based method designed to predict residue orientations and distances (Yang *et al.*, 2020). In contrast to TripletRes, five MSA protocols were used in trRosetta to generate MSAs for each target. The first four were constructed separately using HHblits at four distinct *e*-value cutoffs: $1e^{-40}$, $1e^{-10}$, $1e^{-3}$, 1, while the fifth protocol was conducted using multiple rounds of iterative HHblits searches with progressively relaxed *e*-value cut-offs ($1e^{-80}$, $1e^{-70}$, .., $1e^{-8}$, $1e^{-6}$ and $1e^{-4}$). If the depth of protein sequences that were obtained from these protocols was not adequate, then another alignment was generated by searching in *hmmsearch* (version 3.1b2) against the metagenome sequence database. To prevent the generation of excessively deep MSAs, the search was halted after collecting 2,000 sequences with 75 % coverage or 5,000 sequences with 50 % coverage at a 90 % sequence identity threshold (Yang *et al.*, 2020).

The NN in trRosetta was a residual neural network. The first layer was designed as L×L×526 input features and used two-dimensional (2D) convolution to predict 'a distance histogram (d coordinate) and three angle histograms (ω, θ and φ coordinates)' at the same time (Yang *et al.*, 2020, p. 1502). In the first layer, 2D convolution with a size one filter reduced the number of inputs to 64. After that, a stack of 61 residual blocks was then added. On this stack, the number of dilations was performed 1, 2, 4, 8, and 16 times. At the last block, the network is split into four separate channels for each histogram, where each channel consists of 2D convolution. Following this, the output layer was activated by applying softmax activation. Because of the symmetric mapping for d and ω coordinates, symmetry in the NN was implemented before d and ω channels by inserting 'transposed and untransposed feature maps' from the preceding layer (Yang *et al.*, 2020, p. 1502). Except for the first and last convolution operations, all convolution operations employed sixty-four 3x3 filters, as well as exponential linear unit (ELU) activation functions, which were used through a deep neural model (Yang *et al.*, 2020).

2.1.1.2.3 DeepDist2 (MULTICOM group)

DeepDist2 is a deep learning-based method that has been designed to predict the distance between protein residues. For MSA generation, three tools were used to search homologous sequences against protein sequence databases: DeepAln, DeepMSA and HHblits. Three sets of coevolution-based metrics were employed, which included a COV matrix, a PRE matrix, and PLM, with other sequence features. To generate MSAs for each target, the DeepMSA and DeepAln methods were used to search in various sequence databases, and then different techniques were used to integrate the search results. When DeepAln and DeepMSA created MSAs with less than ten sequences, MSAs were generated by HHblits searching against the Big Fantastic Database (BFD) (referred to as HHlitbe_BSD). The sequence features included the coevolution contact values measured by CCMpred and the Shannon entropy sum, mean contact potential, normalised mutual information, and mutual information computed by DNCON2 (Guo *et al.,* 2021).

The coevolutionary features that were analysed by COV, PLM and PRE metrics and other features were fed into the first layer of deep models, which was an instance normalisation (IN) layer. The next two layers were convolutional and Maxout. After these, the residual block begins with the RCIN block, which consisted of three normalisation layers and an activation function (ReLU). The row normalisation (RN) layer, column normalisation (CN) layer and IN layer were the three normalisation layers of RCIN. The outputs of these layers were combined and fed into a ReLU activation function. In order, a convolutional layer, another RCIN block, and three convolutional layers followed. After that, there were other RCIN blocks, which were followed by a convolutional layer. Finally, there was the squeeze-and-excitation block (SE), which is a popular channel-wise attention method in computer vision; here, it represented the attention mechanism. This block involved two sections: a squeeze operation and an excitation

operation. The first part could collect information from all the channels, whereas the second combined two fully connected layers with the ReLU activation function to boost the impact of relevant features. Using SE, the network recalibrated the feature channels so that it could allocate more attention to those features that were more important. The output was the probability distribution of distances between residues, computed using a softmax activation to divide inter-residue distances into several intervals (Guo *et al.,* 2021).

2.1.1.2.4 Contact Prediction Methods Performance in CASP14

Three cutting-edge methods in contact prediction have been advanced in terms of their input features, MSA construction, and the training process of deep learning models. Because of its unique MSA analysis approach for predicting the distance between residues and employing distance to train deep models, TripletRes has been shown to have the best performance (Li *et al.,* 2021b). Consequently, the precision of TripletRes reached 64 % on FM targets when considering L/5 long-range contacts in CASP14 (as described in the Methods section) (Ruiz-Serra *et al.,* 2021). The performance of trRosetta in CASP14 was improved because of distance and orientation prediction, as well as the MSA selection procedure (Yang *et al.,* 2020). A distance prediction from MSA features with other features has been shown to enhance DeepDist2 predictive performance (Guo *et al.,* 2021).

In this round of CASP experiments, AF2 was designed based on contact prediction principles, which rely on constructing MSA and distance representations to predict accurate 3D models of protein structures from single sequences. However, this remarkable achievement for AF2 relied on the employment of an end-to-end neural network model to learn from co-evolutionary data and distance maps (Jumper *et al.,* 2021a; Saldaño *et al.,* 2022; Yang *et al.,* 2023). Here, the quality of MSA analysis and the training phase of deep neural networks were both critical aspects in improving contact prediction accuracy. These aspects have received attention in

TripletRes and were effectively enhanced (Li *et al.,* 2021a). trRosetta and DeepDist2, on the other hand, were restricted by the low quality of MSA. In other words, analysing MSAs by using a PRE matrix has produced the local features of two residue positions from coevolution-based data, ignoring other residue positions that provide global features (Yang *et al.,* 2020; Li *et al.,* 2021a). However, training deep models of trRosetta by subsampling and selective MSAs enhanced its learning ability, improving contact prediction accuracy (Yang *et al.,* 2020). In DeepDist2, a lot of false sequences were shown to be generated due to its MSA protocol, decreasing its quality (Guo *et al.,* 2021). The combination of distance maps as predicted from four protein feature sets in DeepDist2 helped deep models to extract precise distance information, hence improving the precision of distance prediction (Guo *et al.,* 2021). Since these methods considered different aspects of contact prediction, integrating their relative strengths could further enhance contact prediction accuracy.

2.1.2 Consensus Prediction

Consensus predictions are made using a combination of several different methods, and the various output scores are combined in some way (e.g., the average score or a weighted average) in order to produce a final prediction. The advantages of consensus methods lie in using the combined strengths of many methods to achieve better performance (Wei, Thompson and Floudas, 2012). In other words, for different targets, their best models may be produced by different methods, so if we are able to combine these top-performing methods optimally, then it is more likely the final consensus predictions will be of a higher accuracy overall than could be achieved for any individual method (Lundström *et al.,* 2008).

Consensus methods have been used in a wide range of applications in different stages of protein structure prediction pipelines. Pcons was a neural-network-based consensus predictor that combined six-fold recognition servers using their prediction scores to select the best models.

Lundström *et al.* (2008) confirmed that the consensus prediction using Pcons improved the performance of the overall fold recognition protocol. Aside from consensus fold recognition, CONCORD was a new mixed integer linear optimisation (MILP)-based consensus method designed to optimise secondary structure prediction. This method is based on a combination of the predicted information from seven individual methods by using the MILP model to predict a high accuracy of the secondary structure model (Wei, Thompson and Floudas, 2012). In addition, consensus approaches have been part of MQAPs for many years for estimating the model quality, leading to significant progress in this category. For example, the Cheng group has benchmarked two consensus-based methods, MULTICOM_CLUSTER and MULTICOM-CONSTRUCT, which incorporated nine single model methods with three consensus methods for producing accuracy scores for each model. Further to this, the Cheng group has also applied contact prediction and machine learning approaches to predict the global accuracy of individual models (Cheng *et al.,* 2019). Each consensus method has been evaluated by the CASP assessors and has often been ranked among the top-performing methods in their categories.

In the category of residue-residue contact prediction, there have been a few consensus-based servers that have been developed over the years. Yang and Chen (2011) developed LRcon based on logistic regression for obtaining a consensus contact map prediction. LRcon is a sequence-based protein contact map prediction method and has been constructed on the prediction results derived from contact map predictors evaluated in CASP9. LRcon made a consensus prediction by using the probability of predictors to form feature vectors, which fed into logistic regression and generated models. These models were evaluated in a machine learning framework through independent datasets. The LRcon performance showed significant improvement in prediction accuracy, principally due to the application of the consensus method using the logistic regression algorithm (Yang and Chen, 2011). Another earlier consensus-based contact prediction is MetaPSICOV (the forerunner of DeepMetaPSICOV, which integrated three

coevolution-based methods with a classical neural network learning-based method (PSICOV)). This method generated coevolutionary scores of residue-residue contacts and then used these scores as inputs to a NN to integrate them and produce the final contact prediction scores (Jones *et al.*, 2015; Wu, Hou, *et al.*, 2020). Although predicting residue pair contacts in each of the individual methods was of low accuracy, the consensus approaches combine the strengths of many individual scores to improve the accuracy of contact prediction. This means that employing consensus strategies can further increase the performance of state-of-the-art methods through method integration. Some predictors have developed their servers by employing a consensus approach for obtaining so-called "meta-server" predictions. For instance, the DeepMetaPSICOV server was a consensus-based method that has combined two methods: MetaPSICOV and DeepCov (Fukuda and Tomii, 2020). DeepCov was a covariance-based method that was able to predict sequence covariance features from sequence alignments and used them as input for CNNs (Jones and Kandathil, 2018; Li, Hu, *et al.*, 2019; Li, Zhang, *et al.*, 2019; Fukuda and Tomii, 2020). In the DeepMetaPSICOV program, DeepCov covariance features combined with MetaPSICOV inputs and were then fed into deep neural networks, leading to a further increase in the accuracy of contact predictions between residue pairs (Kandathil, Greener and Jones, 2019).

A recent consensus method for predicting inter-residue contacts using MILP was developed called COMTOP. This method used seven selected residue–residue contact prediction methods, including CCMpred, EVfold, DeepCov, NNcon, PconsC4, plmDCA, and PSICOV (Tegge *et al.*, 2009; Marks *et al.*, 2011; Jones *et al.*, 2012; Ekeberg *et al.*, 2013; Seemayer, Gruber and Söding, 2014; Jones and Kandathil, 2018; Michel, Menéndez Hurtado and Elofsson, 2019; Reza *et al.*, 2021). These methods differ in their input data and algorithm approaches (Reza *et al.*, 2021). Reza *et al.* (2021) demonstrated that COMTOP can considerably enhance the performance of individual techniques. In a recent study, the ensemble of three deep learning-

based contact prediction methods was designed by Billings *et al.* (2021), who aimed to investigate the benefit of the ensemble learning approach in the contact prediction field. The deep learning-based methods that were used to build the ensemble model were ProSPr, trRosetta and Alphafold1 (Senior *et al.*, 2020; Yang *et al.*, 2020; Billings, Morris and Della Corte, 2021). The predictors showed that contact prediction methods based on deep learning are often complementary and that a variety of outputs can be useful in forming ensembles that outperform single methods (Billings, Morris and Della Corte, 2021; Stern *et al.*, 2021).

2.2 Aims and Objectives

The state-of-the-art contact prediction methods have seen considerable improvements in accuracy, which can be attributed to various approaches that integrate coevolutionary features, distance and orientation distributions with deep neural networks. However, the contact prediction accuracy in CASP14 did not notably improve over the 70 % precision which was achieved in the previous CASP round (Ruiz-Serra *et al.*, 2021). Working in this context, the current research aims to enhance the accuracy of residue-residue contact prediction. To produce meaningful improvement, we have sought to develop a consensus method through the integration of top-performing contact prediction methods based on the CASP13 and CASP14 assessments. We first computed the contact scores for each set of residues in each target (domain and full chain) to obtain sets of predicted scores for each method. We then initially combined these scores using simple approaches (e.g., the mean scores) and used them to calculate the final scores. Hence, the objectives of this approach are to combine the outputs from the best contact prediction methods, determine if the predicted contacts of these methods are in agreement or could be complementary to each other and if any further improvements in accuracy could be gained from a simple consensus approach.

2.3 Methods

2.3.1 Data Collection

CASP 13 and CASP 14 datasets were used to conduct the experiment on long-range contacts. Protein targets in these datasets were split into domains. In the CASP13 dataset, 43 domains were classified into 12 template-based modelling/free modelling (TBM/FM) and 31 free modelling (FM) domains. Targets ranged in size from 44 to 431 residues (Shrestha *et al.*, 2019). The targets in the CASP14 dataset include 45 domains. They were classified into 8 TBM hard, 15 TBM/FM, and 22 FM domains. The lengths of the targets varied between 57 and 464 amino acid residues (Li *et al.*, 2021b). In addition, full chains of the targets were chosen for this experiment, which included 35 CASP13 and 36 CASP14 targets. The datasets were collected from the CASP website (https://predictioncenter.org/download_area/).

2.3.2 Contact Definition

We adopted the CASP definition of residue contact based on the Euclidian distance between their carbon atoms, where if the distance of two Cβ atoms for two residues (Cα in the case of glycine) is less than or equal to 8 Å, they are considered in contact; otherwise, they are non-contacting (Monastyrskyy *et al.*, 2011; Monastyrskyy *et al.*, 2014; Schaarschmidt *et al.*, 2018; Shrestha *et al.*, 2019; Ruiz-Serra *et al.*, 2021). Predicted contacts in this defined area can be assigned by computing their probability scores. As such, their probability values (p-values) should range between 1 and 0 (Monastyrskyy *et al.*, 2011; Monastyrskyy *et al.*, 2014; Schaarschmidt *et al.*, 2018; Shrestha *et al.*, 2019; Ruiz-Serra *et al.*, 2021). In our study, we classified prediction data as binary based on probability values into contacting and non-contacting residue pairs, where if the p-value is above 0, the residue pair has been predicted to be in contact with a certain probability.

Predicted contacts were divided into three categories based on sequence separation (i.e. the

number of residues separates between pairs of residues along protein sequence) into short-range ($6 \leq |i-j| \leq 11$), medium–range ($12 \leq |i–j| \leq 23$) and long-range ($|i–j| \geq 24$), where i and j are positions of residue pairs with predicted contacts (Monastyrskyy *et al.*, 2014; Monastyrskyy *et al.*, 2016; Schaarschmidt *et al.*, 2018; Shrestha *et al.*, 2019; Ruiz-Serra *et al.*, 2021). We have concentrated here on long-range contacts for a consensus approach. Long-range contacts have helped us with information on how to constrain the 3D modelling processes for protein structures (Ezkurdia *et al.*, 2009; Monastyrskyy *et al.*, 2014; Monastyrskyy *et al.*, 2016).

This category was divided into subsets based on the length of the target sequence (L) with the highest probability values. These sets include reduced lists (Top10, L/5, L/2, L) and the full list (FL) (Monastyrskyy *et al.*, 2014; Schaarschmidt *et al.*, 2018; Shrestha *et al.*, 2019; Ruiz-Serra *et al.*, 2021). The top 10 set represents the first ten predicted contacts of the residue pairs assigned with the highest probability. L/5 and L/2 subsets reflect predicted contacts for 20 % and 50 % of domain length, respectively. L set contains the predicted contacts of all residues pair within a domain length, whereas the FL set includes all contact prediction datasets. For any residue pairs that were not predicted to be either in contact or non-contacting, we assigned their probability values as zero (Monastyrskyy *et al.*, 2014; Monastyrskyy *et al.*, 2016; Ruiz-Serra *et al.*, 2021).

2.3.3 Consensus Method Design

The concept of the consensus method is to combine the strengths of individual methods to enhance the accuracy of residue–residue contact prediction. To achieve this, the average algorithm was applied to compute the mean of the prediction scores for individual methods in two ways. One of these is to combine two of three methods; the other is to calculate the mean of probability for three methods (Figure 2.1). The output of these consensus methods was then

compared with the output of the individual methods to determine the best consensus method.

The consensus approach implemented by averaging the probabilities of residue pairs in each target for all three methods was called consensus 3 (Cons3). Other consensus approaches are designed by computing the mean of prediction scores for two of three methods, producing three approaches: consensus A (ConsA), consensus B (ConsB), and consensus C (ConsC). Each consensus approach generated two distinct sets of consensus predictions, one from CASP13 and one from CASP14 datasets. Specifically, the first ConsA combined the prediction scores of two CASP13 methods, ZHOU-Contact and DMP, from the CASP13 data, while the second ConsA combined the prediction scores of two CASP14 methods, TripletRes and trRosetta, from the CASP14 data. Similarly, the first ConsB combined the prediction scores of ZHOU-Contact and Zhang_Contact from CASP13, and the second ConsB combined the prediction scores of TripletRes and DeepDist2 from CASP14. For ConsC, the first ConsC combined the prediction scores of DMP and Zhang_Contact from CASP13 and the second ConsC combined the prediction scores of trRosetta and DeepDist2 from CASP14. The same application was extended to Cons3, where the first integration used all three individual methods from CASP13, and the second combined the three methods from CASP14. In summary, the consensus approaches resulted in eight combinations from two datasets by integrating predictions from six different individual methods (Figure 2.1) (see Consensus Code written in Python3 in Appendix 1).

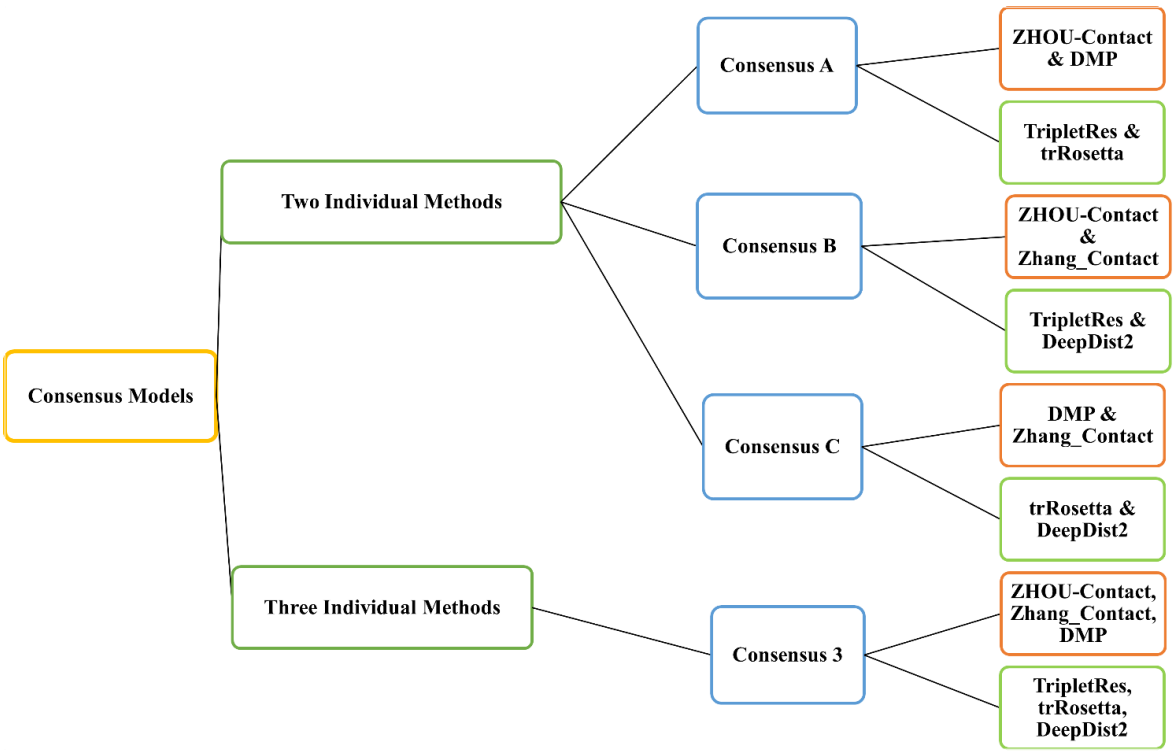


Figure 2.1. The different consensus approaches. The paradigm consists of distinct combinations of two or three individual methods, where each arrow that originates from the consensus approach boxes represents a unique combination. The combination of two methods includes three approaches: consensus A (ConsA), consensus B (ConsB), and consensus C (ConsC). ConsA combines ZHOU-Contact with DMP from CASP13 and TripletRes with trRosetta from CASP14. ConsB integrates ZHOU-Contact with Zhang_Contact from CASP13 and TripletRes with DeepDist2 from CASP14. ConsC combines DMP with Zhang_Contact from CASP13 and trRosetta with DeepDist2 from CASP14. The consensus of three methods (Cons3) integrates all three individual methods from either CASP13 or CASP14.

2.3.4 Evaluation Measures

Three approaches were chosen for evaluating consensus methods coming from the CASP13 assessment of protein contact prediction. These measures include precision, recall, and f1 scores (Shrestha *et al.,* 2019).

Precision is the fraction of correctly predicted contact related to all contacts in the prediction data, which reflects the quality of the prediction data, while recall is the percentage of true predicted contact with respect to all contacts in the target structure.

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

$$Recall = \frac{TP}{Nc}$$

Where *TP* and *FP* are true positive and false positive, indicating the number of correct and incorrect contacts in the prediction data, respectively, and *Nc* is the number of all contacts of the target domain. The f1 measure is the harmonic mean of precision and recall, which acquires the features of both and is more suitable for the full list of the prediction dataset.

$$f1\ score = 2.\frac{Precision\ .Recall}{Precision + Recall}$$

To further evaluate consensus methods, we performed precision-recall (PR) curve analysis, which is used for ranking contact prediction methods based on probability values and for assessing their ability to predict residue contacts correctly by computing area under the curve (AUC_PR) scores, which have been used to indicate the accuracy of methods in recent CASP experiments (Monastyrskyy *et al.*, 2014; Monastyrskyy *et al.,* 2016; Schaarschmidt *et al.,* 2018; Shrestha *et al.*, 2019). In using binary classification on an imbalanced dataset (i.e. the fraction of predicted contacts is lower than that of non-contacts), the best measure to evaluate

the accuracy of our method is the precision-recall curve (PR_curve) (Goadrich, Oliphant and Shavlik, 2004; Bunescu *et al.*, 2005; Kok and Domingos, 2005; Monastyrskyy *et al.*, 2014). This analysis resembles the Receiver Operating Characteristic (ROC) curve, but it is plotted in (recall precision) axes (Fawcett and Flach, 2005; Monastyrskyy *et al.*, 2014). Given skewed data, PR curves may be a more insightful tool than ROC curves, which are often too positive in such situations (Davis and Goadrich, 2006; He and Garcia, 2009; Monastyrskyy *et al.*, 2014). Precision is useful for explaining how a consensus method is good at correctly predicting residue contacts, whereas recall detects how successful a consensus prediction is in predicting true contacts (Tharwat, 2021)

The random performance of the PR curves is influenced by the class distribution in the dataset. Since the AUC for a random classifier in the ROC curve is constant at 0.5, regardless of the class distribution, the AUC_PR of random changes with the class distribution. In a balanced class distribution, where the number of positive instances equals the number of negative instances, the AUC of a random classifier in the PR curve would be 0.5. This means the random classifier performs no better than chance in correctly predicting positive instances. However, in an imbalanced class distribution, where the ratio of positive to negative instances is different, the AUC of a random classifier in the PR curve is equal to the baseline, which is calculated as P / (P + N), where P represents the number of positive instances and N represents the number of negative instances. For example, in a dataset with a 1:10 ratio of positives to negatives, the AUC of the random classifier would be 0.09 (see Figure 2.2) (Saito and Rehmsmeier, 2015). In our study, the PR curves of contact prediction methods have different AUC_PR values for random classifiers due to the imbalanced distribution of contact predictions. The AUC_PR for the methods was calculated in two different ways. The first approach involved calculating the average AUC for all AUC targets for each method. The second approach involved calculating the AUC for each method after pooling all the contact scores for all targets.

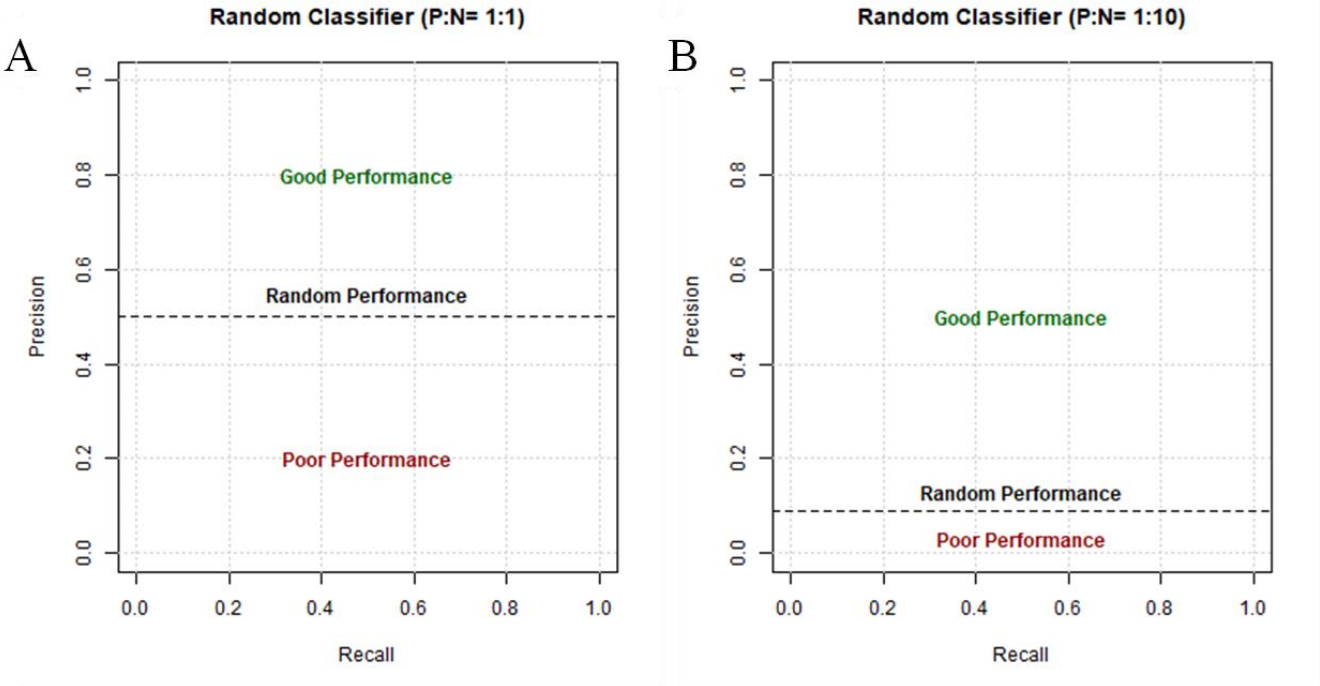


Figure 2.2. The random classifier performance in Precision-recall curve analysis. AUC of random classifier changes based on the ratio of positive prediction in the dataset. A) AUC of random classifier at 0.5 when the ratios of positive (P) and the negative (N) are equal (P:N = 1:1). B) AUC of random classifier at 0.09 when the ratio of positive and the negative are different (P:N = 1:10). Adapted from https://classeval.wordpress.com/.

2.3.5 ConEVA Tool

To calculate the evaluation measures following the CASP evaluation system, the developer of ConEVA adopted the contact definition of CASP experiments and all contact ranges (short-, medium, and long-range). The prediction data were divided into subsets: top-5, L/10, L/5, L/2, L, and top-2 L. The length (L) is defined as the length of the native chain if provided because the native chain could be shorter than the query protein. Otherwise, it is the length of the sequence for which contacts are predicted. The evaluation measures considered in this tool were Precision, coverage, Xd, and mean false-positive error. As the ConEVA web server was not working during the analysis, we downloaded and used the standalone version tool in our research. The tool's input was the prediction data stored in the RR format of the CASP experiment and the PDB file of the experimental protein structure, which can be used to compute native contacts (Adhikari *et al.,* 2016).

The ConEVA output was summarised in two sections. The first section represented the input filename, sequence length, number of native contacts, matching protein sequence with prediction, and sequence separation of long-range contacts. The second section showed the contact numbers and evaluation scores for prediction and native data on all data subsets. For our evaluation, we considered precision scores on three sets (L/5, L/2 and L) for comparison with the CASP experiment's latest evaluation method.

The precision scores were statistically tested to assess significant differences between the consensus-based methods and the individual methods using a paired Wilcoxon test. The paired Wilcoxon test is a statistical test that helps to determine if there is a significant difference between two related groups. This test does not require the data to have a normal distribution. It is an alternative to the paired t-test, which requires a normal distribution of data differences (Miller and Miller, 2010). The null hypothesis (H0) assumes that there is no difference between the paired values, while the alternative hypothesis (H1) suggests that there is a significant

difference. If the calculated p-value is less than 0.05, the null hypothesis is rejected, which indicates a significant difference between the paired groups. In the context of contact structure prediction, the paired Wilcoxon test was used to compare the accuracy of consensus prediction scores against individual prediction scores for the same target.

2.4. Results

The performance of the consensus-based methods was assessed based on the target classification. Protein targets were divided into their domains, which were defined according to CASP's assessment process. Domains were classified into TBM and FM, depending on the availability of structure templates. Our study focused on FM targets as they are the most challenging in protein prediction fields due to the absence of adequate templates and the lack of protein sequence similarities in MSAs. Furthermore, the consensus methods were assessed for the full chain and all domains, regardless of their classification, to investigate whether contact prediction accuracy would be improved for the full chain and domains by consensus approaches. The assessment process was conducted in order to answer the question, "To what extent do consensus methods improve contact prediction accuracy?"

The study findings have been analysed based on three assessment metrics: precision, f1-score and AUC_PR score. The results from each consensus approach were compared with those obtained from each of the individual component deep learning-based contact prediction methods. Here, the findings will be discussed based on the L/5 long-range contacts, following the procedure reported by the CASP assessors (Monastyrskyy *et al.*, 2016; Shrestha *et al.*, 2019; Ruiz-Serra *et al.*, 2021). In addition, evaluation results on L long-range contacts will be presented, which was also suggested by the CASP assessors. The L/5 long-range contacts represent contacts between residue pairs that can be used to reconstruct a 3D model of protein structures. The L long-range set includes the entire list of contacts along the protein sequence,

which assists in evaluating the predicted contacts of the target structure rather than considering all contact predictions that could be longer than the length of the target, resulting in unreasonable evaluation scores (Chen and Li, 2010; Shrestha *et al.*, 2019; Ruiz-Serra *et al.*, 2021).

2.4.1 Consensus-based Method Performance on FM Domains

2.4.1.1 CASP14 FM Domains

The performance of methods was calculated firstly using our own code (The consensus code for CASP14 is in Appendix 1 and is freely available at https://github.com/Shuaa82/Consensus-code) and secondly using the ConEVA tool. Table 2.1 shows the mean precision scores for the contact prediction methods of 22 FM domains from CASP14. Overall, the consensus methods outperformed individual methods based on their mean precision scores. On L/5 long-range contacts, two consensus methods attained the highest mean precision score among all contact prediction methods. The mean precision scores of ConsA and ConsB were higher than 65 %, whereas the mean precision scores of individual methods were lower than 64 %. The mean precision of ConsA was higher than the mean precision score of TripletRes (63.76 %) as well as higher than the average precision score of trRosetta (53.88 %) by ~ 11 %. ConsB achieved a higher mean precision score than TripletRes (63.76 %) and DeepDist2 (54.38 %). It should be mentioned that TripletRes performed similarly to top-performing human-server methods in CASP14 on L/5 long-range contacts (Ruiz-Serra *et al.*, 2021), whereas ConsA and ConsB outperformed them. The mean precision scores of the other two consensus methods (ConsC and Cons3) were comparable to those of DeepDist2 and TripletRes. In addition, the mean precision score of trRosetta was lower than the mean precision scores of ConsC and Cons3; the mean precision score of DeepDist2 was lower than that of Cons3. On L long-range contacts, mean precision scores of individual methods were lower than 40 %, whereas ConsA and Cons3

achieved scores higher than 40 %. In addition, ConsB and ConsC achieved scores comparable with those of TripletRes and trRosetta. However, their mean precision values were higher than those of trRosetta and DeepDist2 by ~2 %-5 %.

Table 2.1. Mean Precision Scores of individual methods compared with those of consensus methods on 22 FM domains of CASP14. The scores were measured for long-range on contact subset lists: Top10, L/5, L/2, L, FL, where L represents the sequence length. Top10 set includes ten amino acid residue pairs with the highest probability value of contacts. The L/5 set contains contact scores of residue pairs within 20 % of the sequence, whereas the L/2 set has predicted scores of contacts for residue pairs within 50 % of sequences. L set contains all predicted scores of residue pairs within sequence length. FL is a full contact prediction dataset.

Method	Top10	L/5	L/2	L	FL
TripletRes (G010)	71.82	63.76	53.32	39.64	1.98
trRosetta (G377)	61.36	53.88	44.42	33.41	**2.21**
DeepDist2 (G420)	63.18	54.38	41.62	31.42	2.02
ConsA (G010 & G377)	**76.19**	65.47	**58.56**	**43.59**	1.94
ConsB (G010 & G420)	66.82	**65.75**	52.68	39.07	1.95
ConsC (G377 & G420)	62.73	54.21	45.16	33.92	1.95
Cons3 (G010 & G377 & G420)	68.18	63.03	52.78	40.35	1.93

The contact prediction methods' performance was also evaluated using the ConEVA tool to compute precision scores as an alternative evaluation. In Table 2.2, the mean precision scores for the individual and consensus methods on 22 FM domains are shown for three sets of long-range contacts. The mean precision score of ConsA was the highest score among all contact prediction methods (individual and consensus) on all three long-range contact sets. For the L/5 long-range set, ConsA and ConsB achieved mean precision scores of more than 65 %, which was higher than the scores of TripletRes (63.80 %), trRosetta (53.8 %), and DeepDist2 (54.40 %). The mean precision score of ConsC (54.01 %) was comparable to that of DeepDist2 and higher than that of trRosetta. Additionally, combining three methods in Cons3 achieved a comparable mean precision score (63.06 %) to that of TripletRes and a higher score than the

other individual methods. On the L long-range contacts, ConsA and Cons3 achieved higher mean precision scores (42.02 % and 40.33 %, respectively) than TripletRes (39.66 %), trRosetta (33.43 %), and DeepDist2 (31.44 %). Moreover, the average precision scores of ConsB (39.14 %) and ConsC (33.96 %) were comparable to those of TripletRes (39.66 %) and trRosetta (33.43 %). However, the mean precision scores of these consensus methods were higher than that of DeepDist2 (31.44 %) by ~2 %-8 %. Thus, the findings indicate that consensus methods improved the mean precision of contact prediction, which demonstrates that combining methods could increase the accuracy of contact prediction and predictive performance.

Table 2.2. Mean Precision Scores of individual methods compared with those of consensus methods on 22 FM domains of CASP14 using the ConEVA tool. The scores were measured for long-range on contact subset lists: L/5, L/2, L, where L represents the sequence length. Top10 set includes ten amino acid residue pairs with the highest probability value of contacts. L/5 set contains contact scores of residue pairs within 20 % of the sequence, whereas the L/2 set has predicted scores of contacts for residue pairs within 50 % of sequences. L set contains all predicted scores of residue pairs within sequence length. FL is a full contact prediction dataset.

Methods	L/5	L/2	L
TripletRes (G010)	63.80	53.35	39.66
trRosetta (G377)	53.83	44.38	33.43
DeepDist2 (G420)	54.40	41.63	31.44
ConsA (G010 & G377)	**65.82**	**56.34**	**42.02**
ConsB (G010 & G420)	65.73	52.72	39.14
ConsC (G377 & G420)	54.01	45.17	33.96
Cons3 (G010 & G377 & G420)	63.06	52.76	40.33

.

The results for ConEVA show statistically significant differences between the precision scores of the consensus-based methods and those of the individual methods according to p-values in a paired Wilcoxon test (see Table 2.3). On L/5 long-range contacts, the differences between the mean precision scores of consensus methods (ConsA and Cons3, $p < 0.05$) and the mean precision score of trRosetta are statistically significant; in addition, the mean precision score of ConsB is significantly different from the mean precision score of DeepDist2. On the other hand, the mean precision scores of the consensus methods (ConsA, ConsB and Cons3) are not significantly different from the mean precision score of TripletRes ($p > 0.05$). Furthermore, the mean precision score of ConsC is not significantly different from the mean precision scores of trRosetta and DeepDist2. These results suggest that integrating TripletRes with other individual methods in consensus methods has a significant impact on their performance, leading to improved predictive accuracy. Similar observations were made for L long-range contacts, apart from ConsC, which has a mean precision score with a significant difference from that of trRosetta. This shows that the predicted number of contacts might affect the performance of prediction accuracy in consensus approaches.

Table 2.3. P-values of mean precision for L/5, L/2, and L long-range contact prediction of CASP14 target domains (FM). L/5 set contains contact scores of residue pairs within 20 % of the sequence, whereas the L/2 set has predicted scores of contacts for residue pairs within 50 % of sequences. L set contains all predicted scores of residue pairs within sequence length.

Method	ConsA	ConsB	ConsC	Cons3
	L/5 long-range contacts			
TripletRes	0.866	0.128	0.998	0.699
trRosetta	**0.001**	**0.003**	0.276	**0.010**
DeepDist2	0.109	**0.002**	0.773	**0.019**
	L/2 long-range contacts			
TripletRes	0.684	0.720	0.977	0.862
trRosetta	**0.000**	**0.001**	**0.050**	**0.002**
DeepDist2	**0.001**	**0.002**	0.149	**0.001**
	L long-range contacts			
TripletRes	0.448	0.700	0.970	0.552
trRosetta	**0.001**	**0.007**	**0.044**	**0.002**
DeepDist2	**0.006**	**0.008**	0.205	**0.002**

Precision scoring neglects native contacts in its calculations, which can reflect the difficulty of targets. Therefore, the f1_score was considered in our analysis to reflect how accurate consensus-based methods are when predicting difficult targets. In other words, predicting a small number of residue contacts in some targets demonstrates how difficult predicting these targets can be, as shown by the f1_score because it considers all true contacts of the targets when calculating the recall (Shrestha *et al.*, 2019).

As the f1_score inherits the properties of precision and recall in giving a reliable evaluation, we analysed our results by computing the mean f1_scores of the consensus methods to be compared with those of the individual methods on the FM domains (see Table 2.4). Overall, ConsA and ConsB outperformed the individual methods, whereas ConsC reached a similar level to those individual methods' performance on L/5 long-range contacts. The higher mean f1 scores of ConsA and ConsB could be attributed to the obvious effect of TripletRes on their performance, as combined with the other two individual methods in these consensus methods led to enhance accuracy in contact prediction. This can be seen with the L set, where the mean f1_score of ConsA (42.93 %) was the highest score among both the individual and consensus methods, indicating that ConsA effectively exploited the advantages of TripletRes and trRosetta, leading to it acquire a capability to explore native contacts. On the other hand, the mean f1_score of ConsB was slightly lower than that of TripletRes. This reduction could be related to DeepDist2 performance, the mean f1_score of which was the lowest among all contact prediction methods. However, the combination of DeepDist2 and trRosetta in ConsC resulted in a slightly higher value of f1_score than their individual scores. In Cons3, the mean f1_scores on L/5 and L were higher than those of DeepDist2 and trRosetta and comparable to the mean f1_score of TripletRes. Despite this, Ruiz-Serra *et al.* (2021) demonstrated that FM targets in CASP14 were more difficult than those in CASP13, which might affect the predictive performance of contact prediction methods. Our findings suggest the effectiveness of consensus methods compared to individual methods' performances. Subsequently, consensus prediction could play a valuable role in advancing contact prediction accuracy.

Table 2.4. Mean f1_score scores of individual methods compared with consensus methods on 22 FM domains of CASP14. The scores were measured for long-range on contact subset lists; Top10, L/5, L/2, L, FL, where L represents the sequence length. Top10 set includes 10 amino acid residue pairs with the highest probability value of contacts. L/5 set contains contact scores of residue pairs within 20 % of the sequence, whereas the L/2 set has predicted scores of contacts for residue pairs within 50 % of sequences. L set contains all predicted scores of residue pairs within sequence length. FL is a full contact prediction dataset.

Method	Top10	L/5	L/2	L	FL
TripletRes (G010)	8.74	21.88	35.76	39.28	3.86
trRosetta (G377)	6.48	17.46	28.24	31.83	**4.29**
DeepDist2 (G420)	7.19	17.90	26.71	29.97	3.94
ConsA (G010 & G377)	**9.27**	22.25	**38.93**	**42.93**	3.77
ConsB (G010 & G420)	7.84	**22.66**	35.23	38.67	3.81
ConsC (G377 & G420)	7.00	17.81	28.98	32.47	3.79
Cons3 (G010 & G377 & G420)	8.06	21.56	35.11	39.79	3.76

Increasing predicted contact numbers could render evaluation unreliable, as the number of contacts is lower than the non-contact number, therefore producing imbalanced data for binary classification. To overcome this problem, the best assessment measure for imbalanced data is PR curve analysis. PR analysis was performed to see whether contact prediction methods could accurately assign high confidence levels to predicted contacts. The area under the PR curve (AUC_PR) was calculated and used as ranking scores for contact prediction methods. It is worth noting that the AUC_PR scores for the best-performing methods in the CASP contact prediction assessments were below 0.5 (Monastyrskyy *et al.*, 2016; Shrestha *et al.*, 2019). Keeping this in mind, we have found that the AUC scores of our contact prediction methods are consistent with the AUC_PR scores observed in CASP experiments.

In Figure 2.3, the average of AUC_PR scores of individual and consensus methods have been

represented for FL long-range contacts for CASP14 FM domains. As shown in the Figure 2.3, ConsA outperformed the individual methods as well as the other consensus methods based on its average AUC_PR score (0.42). Moreover, Cons3 and ConsB achieved the average AUC_PR scores similar to the average AUC_PR score of TripletRes (0.41), which was the best individual method. Additionally, the average AUC_PR scores of Cons3 and ConsB were higher than the average AUC_PR scores of the other individual methods. These three consensus methods can more accurately predict contacts than the individual methods, which might explain these ratings. AUC_PR score of ConsC (0.32), which was consistent with prior assessment scores, indicates that the combination of trRosetta and DeepDist2 failed to accomplish the purpose of the consensus method, with trRosetta achieving a higher average AUC_PR score (0.34) (see Table 2.5).

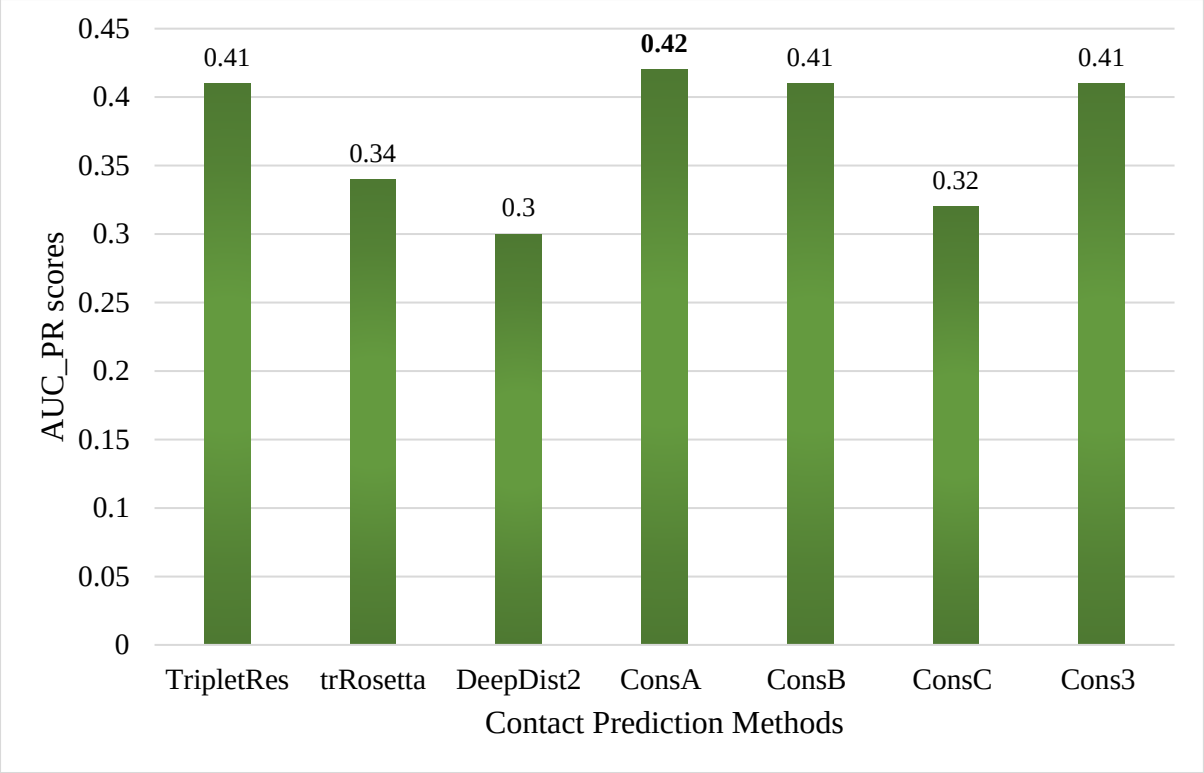


Figure 2.3. A comparison of consensus methods and individual methods on FL long-range contact set based on AUC_PR score of Precision-Recall curve analysis for 22 CASP14 FM domains.

The performance evaluation of individual methods and consensus methods for predicting FL long-range contacts for CASP14 FM domains based on AUC_PR scores are represented in Table 2.5. ConsA's performance is at a similar level as TripletRes, the top individual method, as both achieved an AUC_PR score of 0.47. This indicates that ConsA, as a consensus method, is more accurate in predicting contacts than other individual methods. Cons3 and ConsB demonstrated equivalent performance to TripletRes, as their AUC_PR scores were 0.46. Additionally, the AUC_PR scores of Cons3 and ConsB were higher than those of the other individual methods. Furthermore, ConsC had an AUC_PR score of 0.39, indicating that the combination of trRosetta and DeepDist2 achieved a slight improvement in the accuracy of contact prediction. Notably, trRosetta achieved a comparable AUC_PR score (0.38) to the consensus method, ConsC. This suggests that the consensus methods are more accurate in contact prediction compared to the individual methods, apart from TripletRes.

Table 2.5. AUC_PR scores of individual methods compared with consensus methods on 22 FM domains of CASP14. The scores were measured for long-range on contact subset FL, which is a full contact prediction dataset. The AUC_PR scores were calculated in two different ways. AUC_PR represent the scores of the prediction methods based on the contact prediction for all 22 FM targets. Average AUC_PR represent the scores of prediction methods based on the AUC of all targets for each method. AUC_PR of the random classifier is for each PR curve analysis of each method.

CAPS14 methods	AUC_PR	AUC_PR of a random classifier	Average AUC_PR
TripletRes (G010)	**0.47**	0.01	0.41
trRosetta (G377)	0.38	0.02	0.34
DeepDist2 (G420)	0.33	0.01	0.31
ConsA (G010 & G377)	**0.47**	0.01	**0.42**
ConsB (G010 & G420)	0.46	0.01	0.41
ConsC (G377 & G420)	0.39	0.01	0.32
Cons3 (G010 & G377 & G420)	0.46	0.01	0.41

2.4.1.2 CASP13 FM Domains

A similar evaluation was conducted on the consensus-based methods' performance on CASP13 data (the consensus code for CASP13 data is in Appendix 1 and is freely available at https://github.com/Shuaa82/Consensus-code). Overall, the consensus-based methods showed the best performance based on mean precision scores on 31 FM domains (see Table S.1, Table S.2 in Appendices 2 and 3). On L/5 long-range contacts, the mean precision scores of ConsA and ConsC reached 64.83 %, which is higher than the mean precision scores of the individual methods (Zhang_Contact=57.38 %, ZHOU-Contact = 58.90 %, DMP = 60.80 %). Moreover, the highest mean precision score belonged to Cons3 (65.98 %), which combined all individual methods. For L long-range contacts, three consensus-based methods (ConsB, ConsC and Cons3) achieved more than 40 % mean precision compared to the mean precision of the individual methods, which was less than 40 %. In agreement with these results, the mean precision scores computed by the ConEVA tool demonstrated that consensus-based methods outperformed the individual methods. Precision score means of consensus-based methods were higher than ~65 % on L/5 long-range contacts and higher than 40 % on L long-range contacts, which are statistically different from the mean precision scores of the individual methods according to the p-values shown in Table S.3 in Appendix 4.

Based on f1_score and the average of AUC_PR values of Precision-Recall analysis, the consensus approaches performed better than the individual methods. According to their mean f1_scores, the combination of three individual methods in Cons3 achieved a higher score on L/5 and L long-range contacts than any of the individual methods, followed by ConsC and ConsB (see Table S.4 and Table S.5 in Appendices 5 and 7). These consensus methods attained the best scores for average AUC_PR (0.41 for Cons3, 0.40 for ConsB and ConsC), as shown in Figure S.1 (see Appendix 6). In addition, the consensus methods outperformed the individual methods and performed comparably well to DMP based on AUC_PR scores (see Table S.5 in

Appendix 7). These findings, together with precision scores, suggest that consensus-based techniques may improve predictive accuracy depending on the contact prediction methods combined.

2.4.2 Consensus-based Method Performance on Full Chain versus Domains

Following the CASP-like assessment process, the evaluation measures were computed on the target domains to produce a precise analysis of the performance of the contact prediction methods. One of the lessons learned from the CASP14 experiment is that accuracy is improved when the prediction is conducted on full chains of proteins rather than domains, as all predicted contacts among and within domains of each target are considered. To investigate this assumption, we test our consensus approaches on full chains and domains of CASP13 and CASP14 targets, regardless of their classification, and evaluated their performance in comparison with individual methods.

The predictive accuracy differed on full chain and domains of CASP14 and CASP13 according to mean precision scores obtained with the ConEVA tool. For the CASP14 data, the consensus-based methods demonstrated successful performance for full chain and domains on L/5 long-range contacts, as shown in Figure 2.4. ConsA (77.67 %) achieved the highest mean precision scores on domains and Cons3 (77.12 %) on full chains. On L long-range, mean precision scores were relatively better on domains for most contact prediction methods. However, ConsA was the most accurate method for domains (52.5 %) as well as for full chains (51.33 %) (Figure 2.5). Overall, the predicted contacts for the domain dataset were more accurate than those for the full chain dataset of CASP14. Conversely, predicted contacts for full chains were significantly higher than those for domains using the CASP13 data. The mean precision scores of consensus methods on the full chain dataset were higher than 75 %, whereas those on the domain dataset were less than 74 % on L/5 long-range contacts. Interestingly, ConsB achieved

the highest score (79.02 % of mean precision) for full chains, which was higher by ~9 % than its mean precision score for domains. Regarding L long-range contacts, predicted domain contacts were lower by ~2 %-3 % of mean precision compared to those of full chains for consensus methods except for ConsC, which performed similarly on both (~ 47 %) (see Figures S.2, S.3 in Appendices 8 and 9). The difference between mean precision scores for full chains and domains from CASP13 versus those from CASP14 might be related to the differentiation between deep learning-based methods with respect to approaches and algorithms employed in them. Nevertheless, in summary, the consensus-based methods outscored the individual methods on both full chains and domains for both the CASP13 and CASP14 targets.

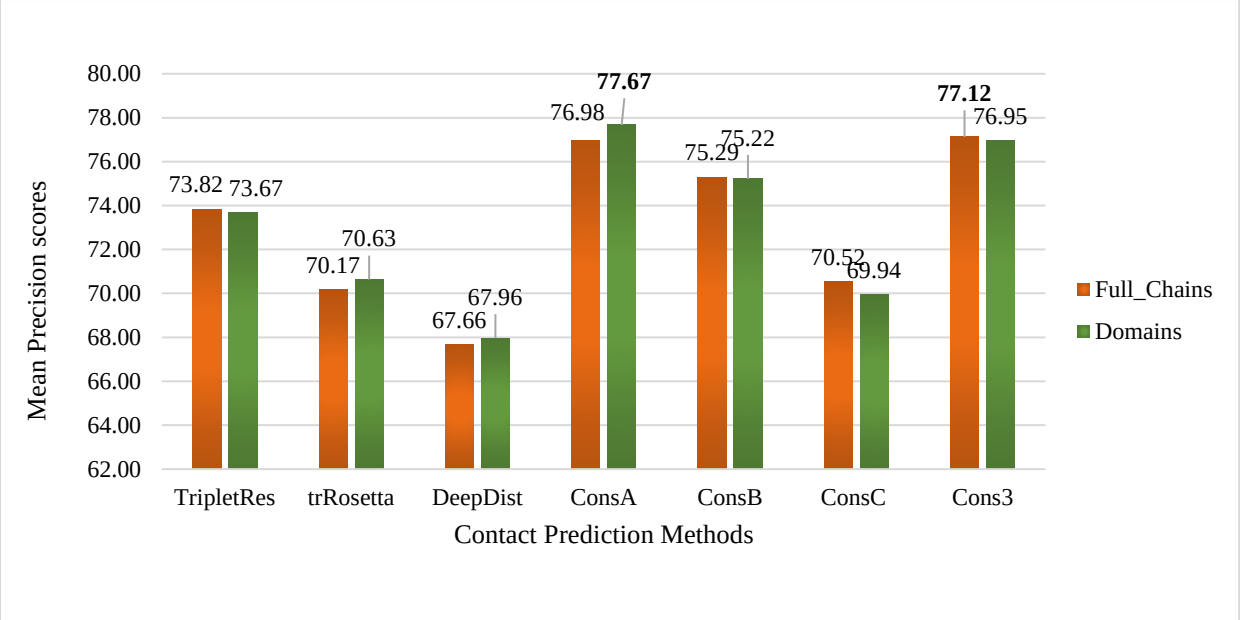


Figure 2.4. Mean precision scores of predicted contacts for domains and full chains of CASP14 targets on L/5 long-range contacts for 36 full chains and 43 domains-ConEVA tool.

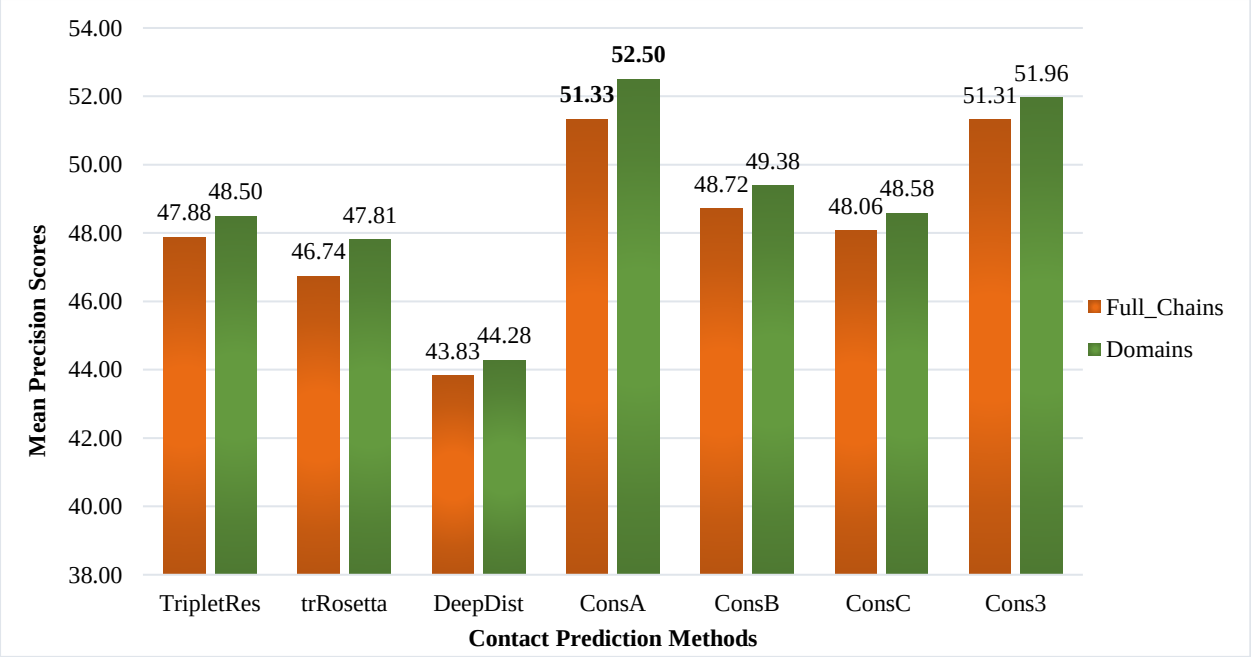


Figure 2.5. Mean precision scores of predicted contacts for domains and full chains of CASP14 targets on L long-range contacts for 36 full chains and 43 domains-ConEVA tool.

2.5 Discussion

Based on the findings from the evaluation, consensus-based methods have improved the predictive performance over individual methods, leading to an increase in contact prediction accuracy. The performance of the consensus methods can be attributed to the varying design of the component individual servers in three aspects: MSA construction, distance and orientation prediction, and a deep model of neural networks. In this section, we address the reasons for the enhanced performance of consensus methods on CASP14 and CASP13.

The highest accuracy of contact prediction went to ConsA, into which two CASP14 top-ranked methods were integrated among all consensus methods. This indicates that TripletRes and trRosetta were complementary. In TripletRes, a combination of three evolutionary matrices extracted precise evolutionary features from a deep MSA, and these matrices were the main factor that led to a considerable improvement in its performance in CASP14 (Li *et al.*, 2021a). Another improvement was added in TripletRes when 'discretised distance' information was used as a loss function to train the deep neural model (Li *et al.*, 2021a). In trRosetta, MSA selection was used in the deep model neural network to select precise MSA features among different MSA protocols, which are generally conducted because deep MSA might not have a good quality for some targets (Kandathil, Greener and Jones, 2019; Yang *et al.*, 2020). This step advanced trRosetta's prediction accuracy (Yang *et al.*, 2020; Du, Peng and Yang, 2022). Furthermore, orientation prediction contributed to the improvement of trRosetta. Both methods were designed to predict the distance map of protein sequences, which involves precise information that could support predicting accurate contact networks among protein residues (Yang *et al.*, 2020; Li *et al.*, 2021a). Combining prediction contact data of these methods in the ConsA approach leverages the strengths of their performance, leading to further improvement in the accuracy of contact prediction.

ConsB and Cons3 achieved a relatively high level of contact prediction accuracy compared

with ConsA. In ConsB, the quality of the MSA generation of DeepDist2 was less than that of TripletRes. In DeepDist2, the developers combined DeepMSA and DeepAln with HHlitbe_BSD to generate MSAs; they demonstrated that MSAs have noisy information due to most of the sequences being non-homologous, which creates a great deal of false-positive data that reduces the accuracy of contact prediction (Guo *et al.,* 2021). In TripletRes, the power of its performance came from the ensemble co-evolutionary matrix, which led to the high quality of MSA analysis. Combining these two methods in ConsB might have led to overcoming the weakness of DeepDist2 with the strength of TripletRes by reducing the false positive in prediction data, improving contact prediction accuracy. The effect of TripletRes's performance can be seen in Cons3. TripletRes's design distinguishes it from the other methods (trRosetta and DeepDist2) regarding the ensemble of statistical models of coevolution data and discrete distance function of deep learning training. These advanced stages improved TripletRes's performance over individual methods. Hence, the consensus of TripletRes with DeepDist2 and trRosetta in Con3 increased the accuracy of contact prediction.

ConsC was the least accurate consensus method. This may be related to MSA procedures found in the individual methods. The main way to improve the predictive performance of contact prediction is by using data derived from coevolution (Ruiz-Serra *et al.,* 2021). The MSA procedures of trRosetta produced evolutionary local features that did not consider the effect of universal features, and the DeepDist2 MSA construction yielded non-homologous sequences that delivered inaccurate information. This might be why ConsC was unable to outperform individual methods; the prediction data of the individual methods was less accurate because of the low quality of their MSA. However, ConsC achieved a comparable level of prediction accuracy as these individual methods.

It should be noted that the developers of DMP have demonstrated that the deep length of MSA could reduce the precision of contact prediction because of mismatching sequences in the alignment process. This mismatched sequence problem can produce incorrect coevolution information that leads to a reduction in the quality of MSA and the accuracy of contact prediction (Kandathil, Greener and Jones, 2019). However, it is believed that the success of contact prediction methods in CASP13 is due to models composed of deep neural networks. DMP substantially improved contact prediction accuracy, which can be attributed to its advanced deep neural network models. Training five models of NNs with adding strategies for data augmentation was the reason for the improvement in the contact prediction performance in DMP (Kandathil, Greener and Jones, 2019). In SPOT-Contact, the ensemble of NN models added a substantial improvement to the capability of deep model learning to extract interaction patterns between protein residues in a 3D model of protein structures (Hanson *et al.,* 2018). The improved performance of NeBcon was because of an ensemble of nine deep learning-based approaches, which utilise variance deep neural network models (Zheng *et al.,* 2019). However, each NN model in each method could predict a slightly different pattern because of the differences in their designs, ‘parameter initialisations’, data input and other variables. To boost predictive accuracy, ensemble averaging tends to take advantage of these patterns’ complementarity (Ding *et al.,* 2018). This could be suggesting the increasing accuracy of contact prediction of the four consensus methods over individual methods in CASP13.

2.6 Conclusion

Consensus-based contact prediction methods have been developed to improve contact prediction accuracy by integrating the top-ranked methods in the contact prediction field in two recent CASP rounds. The purpose of the consensus methods was to reach an accuracy of contact prediction beyond ~70 % by exploiting the advances in deep learning-based contact prediction methods. In recent methods, enhancements were applied to the most important stages of contact prediction, including MSA construction, deep models of neural networks, and employment of distance and orientation maps. Although the contact prediction accuracy did not reach more than 70 % for the harder FM targets, consensus-based methods succeeded in leveraging these advancements for further improvements to contact prediction accuracy.

The performance of consensus-based methods for contact prediction was determined by the success of combining the strengths of deep learning-based methods for prediction. By combining the outputs from TripletRes and trRosetta, we observed improvement in the accuracy of the predicted contacts (ConsA) by 3.2 % (from 63.80 to 65.82 %) according to mean precision on L/5 long-range contacts for FM domains in CASP14. This improvement indicates that TripletRes and trRosetta predicted contact maps for FM targets with varying degrees of accuracy; ConsA was likely able to combine these to obtain optimal contact maps with high accuracy. It is crucial to note that this increase may not be statistically significant ($p > 0.05$). In CASP13, integrating three deep learning-based methods in Cons3 brought the mean precision of predicted contacts by 10.5 % (from 61.17 % to 67.96 %) on L/5 long-range contacts for FM domains. This improvement is statistically significant with a p-value less than 0.05. This means that the individual methods were complementary to each other, and the consensus method thus exploited this to increase the contact prediction accuracy.

The inability of other consensus methods to increase contact prediction accuracy could be related to the target difficulty of FM domains and the choice of using deep learning-based

methods to design consensus methods. In a recent CASP14 report, Ruiz-Serra *et al.* (2021) demonstrated that FM targets in CASP14 posed more challenges than those in CASP13. This could potentially affect the predictive performance of individual prediction methods and, hence, consensus methods. In addition, choosing two deep learning-based methods to build consensus methods (seen with ConsC in CASP14) could lead to a reduction in contact prediction accuracy. Such a reduction would result from the merging of inaccurate data. Therefore, if deep learning-based methods are chosen improperly, the consensus of their prediction may result in the opposite of what was initially intended.

Consensus-based methods successfully improved predicted contact accuracy for both full chains targets and their domains. Based on mean precision scores, the accuracy of predicting contacts for domains was slightly higher than that of full chains in CASP14, whereas it was lower than that of their full chains in CASP13. This may indicate that various approaches and algorithms employed in deep learning-based methods produce different contact maps for full chains and domains with varying degrees of accuracy. The purpose of analysing contact prediction on both full chains and domains was to examine whether the accuracy of contact prediction would be better improved with full chains than with domains. However, the prediction accuracy of consensus methods reached ~77 % on both the full chains and their domains in CASP14 on L/5 long-range contacts.

Our new consensus-based contact prediction approaches can be used to complement the cutting-edge modelling methods, for example, utilising them to estimate the accuracy of 3D models at the local level. With that in mind, in the next chapter, we aimed to merge deep learning-based contact prediction into our ModFOLD9 pipelines to enhance its ability to estimate the 3D model quality. We describe how we used the CDA score to assess the local and global quality of 3D models based on a combination of contact scores.

Chapter 3 Development of Consensus Contact Distance Agreement Scores for Local Model Quality Estimates

3.1 Background

The interaction of protein residues is an important aspect of predicting protein structure models. Many protein prediction tools employ residue-residue contact maps to further improve their predictive performance and enhance their predicted 3D models of protein structures (Wu, Szilagyi and Zhang, 2011; Buchan and Jones, 2017; Zheng *et al.*, 2019; Yang *et al.*, 2020; Mortuza *et al.*, 2021; Quignot *et al.*, 2021). Recently, QA methods have integrated contact prediction tools into their in-house programmes, enhancing their predictive accuracy (Cao *et al.*, 2017; Maghrabi and McGuffin, 2017; Cheng *et al.*, 2019; Jing and Xu, 2020; McGuffin *et al.*, 2021; Ye *et al.*, 2021).

3.1.1 Model Quality Assessment

Protein structure prediction methods will often generate multiple alternative models with varying accuracy depending on the target. Highly accurate 3D models are necessary for application in biomedical studies. Therefore, researchers must be able to accurately select the best quality models from among the alternatives (Uziela and Wallner, 2016). Therefore, estimating the quality of predicted 3D models, prior to availability of experimental data, is one of the critical stages of protein structure prediction pipelines.

Estimation of Model Accuracy (EMA) or Quality Assessment (QA) methods are designed to detect errors in 3D models, which can then be avoided or fixed in order to increase the quality of the models (Won *et al.*, 2019). There are two kinds of errors: local and global errors. Estimating local errors, which denotes investigating how each residue in the 3D model deviates from the corresponding residue in a native structure, could help improve the accuracy of the local regions of the 3D models. On the other hand, global error estimation allows us to rank the many alternative models and then select the best models of the target protein.

QA methods are classified into single-model, quasi-single model, and clustering methods. Single-model methods predict the quality based on a single model input, whereas clustering-based methods rank the best models for a target by assessing multiple models (Won *et al.*, 2019). In quasi-single model methods, a set of reference structures are modelled from target sequence, then compared to a single model to evaluate its quality accuracy (Maghrabi and McGuffin, 2017; Cheng *et al.*, 2019; Maghrabi, 2019; Chen and Siu, 2020; McGuffin *et al.*, 2021).

A recent achievement in modelling methods has been the prediction of 3D models with 90 per cent accuracy by AF2 (Jumper *et al.*, 2021a; Pereira *et al.*, 2021). However, AF2 can have limitations in predicting local regions for some dynamic structures, which means that there remains a pressing need to be able to independently estimate the quality of local areas of 3D models, especially in dynamic regions (Fowler and Williamson, 2022; Yang *et al.*, 2023). Additionally, many other state-of-the-art publicly available modelling servers, still produce models that are inaccurate at the local level. Hence, the remaining local errors in high-quality models has increased the need for high-performance QA methods (Kwon *et al.*, 2021; McGuffin *et al.*, 2021). Furthermore, as it is a challenge for QA methods to distinguish between very high-quality 3D models, their estimation ability must be improved (Kwon *et al.*, 2021).

3.1.2 Application of Contact Prediction Methods for Model Quality Estimates

The quality of local regions and the overall model can be assessed by identifying per-residue errors in 3D models. For that purpose, QA methods were developed to include deep learning-based contact prediction methods, which improved their prediction performance in terms of assessing the local and global quality accuracy of 3D models (Cheng *et al.*, 2019; Chen and Siu, 2020; Jing and Xu, 2020; Chen *et al.*, 2021; McGuffin *et al.*, 2021; Liu *et al.*, 2022).

Single-model methods estimate the accuracy of protein models quality based on local features, such as secondary structure prediction, solvent accessibility and residue-residue contact patterns, which are extracted from a single model. For example, ProQ2 used SVM to integrate the protein features that were derived from the model (Ray, Lindahl and Wallner, 2012; Uziela and Wallner, 2016). The structural features were contact patterns between residues and surface accessibility, whereas sequential features were extracted from predicted secondary structures and sequence profiles (Uziela and Wallner, 2016).

In addition, contact prediction methods have been integrated with clustering-based model quality assessment methods. Cheng *et al.* (2019) developed three consensus-based quality assessment methods for CASP13. The development used a deep neural network model and ensemble approach to combine the predicted contact scores with a large set of other scores to estimate the global accuracy.

In earlier versions of ModFOLD, contact prediction was integrated using a pure-single model method called the CDA score. This method aims to determine the agreement between the predicted contacts computed by deep learning-based contact prediction methods and the contact scores of 3D models calculated via the Euclidean distance algorithm (Maghrabi, 2019). In the sixth and seventh versions of ModFOLD, the contact prediction method MetaPSICOV (Jones *et al.*, 2015) was used to produce the predicted contact scores to measure CDA score, which was combined with other pure- and quasi-single scores into a NN, which led to a substantial improvement in its performance in both local and global accuracy according to CASP12, CASP13 and CAMEO assessment (Maghrabi and McGuffin, 2017; Maghrabi and McGuffin, 2020; Elofsson *et al.*, 2018; Cheng *et al.*, 2019; Chen and Siu, 2020). In the eighth version, the ModFOLD server was developed with the combination of two new CDA scores derived from the top-ranked deep learning-based contact prediction methods along with the previous CDA score (McGuffin *et al.*, 2021). In total, the development of the eighth version of

ModFOLD combined 13 scores, nine pure-single model methods and four quasi-single model methods, aiming to increase performance accuracy. The pure-single methods included ProQ2, ProQ2D, ProQ3D, ProQ4, VoroMQA, the Secondary Structure Agreement (SSA) score, and three CDA scores, which were produced from three deep learning-based contact prediction methods. The two new pure-single methods were the CDA_SC and CDA_DMP scores derived from the SPOT-Contact and DeepMetaPSICOV contact prediction methods (McGuffin *et al.*, 2021). These CDA scores, along with the combination of other pure- and quasi-single model scores, were fed into the NN model to predict the final QA score of each model (McGuffin *et al.*, 2021). The CASP14 and CAMEO assessments ranked ModFOLD8 at the top in terms of the estimation of quality assessment (Kwon *et al.*, 2021; McGuffin *et al.*, 2021).

3.1.3 Development of Consensus CDA scores from Contact Prediction Methods

The CDA method added value to the predictive performance of the ModFOLD servers. This pure single-model method was designed to detect missing contacts in a 3D model of a protein structure. This detection can be observed when comparing the contact scores of targets produced from deep learning-based contact prediction with the contacts in 3D models that are calculated by applying the Euclidean distance algorithm. By employing contact scores in estimating procedures, the accuracy of the predicted local errors of models can be enhanced (Maghrabi and McGuffin, 2017; Maghrabi, 2019; McGuffin *et al.*, 2021).

Different deep learning-based methods for contact prediction will produce different contact scores for a given protein target. This might be related to the different approaches used by the multiple deep neural network models, depending on the developer's aims. Leveraging the benefits of these different methods could strengthen the accuracy of contact prediction for use in model quality estimates. This can be accomplished by combining the contact scores from various deep learning-based methods using a consensus approach. For local model quality

assessment, CDA scores can be measured using multiple alternative contact prediction methods, and then merged using a NN to produce a consensus CDA score for a single 3D protein model. Using the CDA scores as inputs, the NN can be trained to learn the output quality scores of the models by estimating the accuracy of the local regions. In the ModFOLD servers, two local observed model quality scoring measures were used for training and benchmarking the local model quality predictions: the superposition-based score (S-score) and local Distance Difference Test (lDDT) (Elofsson *et al.*, 2018; Cheng *et al.*, 2019; McGuffin *et al.*, 2021). These observed local model quality scoring methods have been used in the quality assessment category of the CASP experiments to evaluate the performance of the predicted local model quality scoring methods.

3.1.4 Description of the Observed Local Model Quality scores used for Training and Benchmarking the ModFOLD Method

3.1.4.1 The Superposition-based score (S-score)

The local quality of each 3D model was assessed by analysing the similarities between the model and the reference structure at the residue level. The comparison was conducted using a pairwise superposition to compute the Template Modelling score (TM-score), which represents the global similarity between the two structures (Zhang and Skolnick, 2004; Maghrabi, 2019). The superposition evaluation at a local (per-residue) level was performed using the S-score. Following superposition, the S-score reflects how close equivalent residues are in the predicted and observed structures. The S-score was used in model quality evaluation in various methods (Fischer, 2003; Wallner, 2006; Wallner and Elofsson, 2007; Larsson *et al.*, 2009; McGuffin, 2009; McGuffin, Buenavista and Roche, 2013; Maghrabi and McGuffin, 2017; Maghrabi, 2019; Jing and Xu, 2020; McGuffin *et al.*, 2021). In the recent ModFOLD versions, the S-score was used as a target function for training the NNs. The S-Score is computed for each residue in each model for a target by using the formula:

$$S_i = \frac{1}{1 + (\frac{d_i}{d_0})^2}$$

Where S_i is S-score for residue (i) in a model, d_i is the distance between residue (i) in the model aligned with the equivalent residue in the reference (observed) structure according to the TM score superposition, and d_0 is the distance cut-off set at 3.8 Å. If the di > d0, S-score will be 0, indicating no similarity between the two structures for residue (i) (McGuffin, 2009; McGuffin, Buenavista and Roche, 2013; Maghrabi, 2019).

The average S-score was calculated for each residue in the model by the summation of S-score of a residue (i) in all models of a target. The average S-score was defined as S_r and its formula is:

$$S_r = \frac{1}{N-1} \sum_{a \in A} S_{ia}$$

Where N represents the number of models of a target, A is the alignment set whose size is $N-1$, and S_{ia} is the S-score for a residue (i) in the model (a) (McGuffin, 2009; McGuffin, Buenavista and Roche, 2013; Maghrabi, 2019). The S-scores are local structural similarity scores which are dependent on the superposition of the predicted 3D model with the observed structure. This dependency can impact the robustness of the score as it could be highly sensitive to relative domain positions e.g., where there may be dynamic movement between domains with flexible linkers (Kryshtafovych, Monastyrskyy and Fidelis, 2014; Olechnovič and Venclovas, 2014a; Mulnaes and Gohlke, 2018; Maghrabi, 2019). To avoid this drawback, other superposition free target functions, such as Contact Area Difference (CAD) (Olechnovič, Kulberkytė and Venclovas, 2013) and Lddt (Mariani *et al.*, 2013), were built to be independent from superposition influence. The McGuffin group noticed this issue and addressed it by also considering the lDDT score as the target function for local assessment along with S-score (Maghrabi, 2019; McGuffin *et al.*, 2021).

3.1.4.2 The Local Distance Difference Test (lDDT) score

lDDT is a local superposition-free score that assesses the local geometry of 3D models. The lDDT score compares the distances between atoms in predicted 3D models to the equivalents in reference structures instead of aligning them. Thus, it is arguably more robust and less affected by the different relative structural orientations of the independent folds within a chain, such as domain movements (Kryshtafovych, Monastyrskyy and Fidelis, 2014; Olechnovič and Venclovas, 2014a; Mulnaes and Gohlke, 2018). Moreover, all residue interactions on the backbone and side-chain are considered in lDDT, which reflects the accurate assessment of all local regions of 3D models (Mariani *et al.*, 2013; Huang *et al.*, 2014).

lDDT has become an official evaluation measure for assessment of the quality 3D models in both CASP and in the Continuous Automated Model EvaluatiOn (CAMEO) project (Huang *et al.*, 2014; Kinch *et al.*, 2016; Li *et al.*, 2016; Haas *et al.*, 2018; Adiyaman and McGuffin, 2019). The purpose of the lDDT score is to gauge the accuracy of local areas in 3D models compared to those in reference protein structures (Mariani *et al.*, 2013). Additionally, it also assesses the accuracy of the protein's packing core in 3D models (Mariani *et al.*, 2013; Huang *et al.*, 2014). The lDDT score is computed by quantifying the variations between the distance maps of residue interactions in the model and the distance maps of equivalent interactions in reference structures at specific thresholds (Huang *et al.*, 2014; Kryshtafovych, Monastyrskyy and Fidelis, 2014; Olechnovič and Venclovas, 2014a; Studer, Biasini and Schwede, 2014; Cao *et al.*, 2016; Kim and Kihara, 2016; Li *et al.*, 2016; Modi and Dunbrack, 2016; Haas *et al.*, 2018; Waterhouse *et al.*, 2018). Like the S-score, a higher lDDT score implies a higher local prediction accuracy in 3D models, which shows how closely the local geometry matches native geometry.

3.1.5 An Overview of The Neural Network (NN) trained using The Input CDA scores

Machine learning approaches are advanced algorithms applied in protein structure prediction tools and they have made a notable contribution over the years in enhancing their predictive power (Chen and Siu, 2020; Greener *et al.*, 2022). Artificial neural networks (ANNs) are machine learning methods that can be used to process and identify patterns in large amounts of data, and they have been applied to address problems in many fields of research. Leveraging the benefits of the advances in NNs over the years has led to successive improvements in the performance of protein prediction methods.

The design of ANNs was inspired by the biological neural network of the human nervous system (Abiodun *et al.*, 2018; Maghrabi, 2019). A human neural network consists of a vast number of interconnected neuronal cells (neurons). Each neuron collects input signals from other neurons through a dendritic tree branch. Neurons are a composite of the cell body and axon. To address input signals, biological functions are processed in the cell body and then output signals are passed through the axon to other neurons. The interconnection networks between neurons enable higher functions such as intelligence, recognition, and classification. After studying how biological neural networks function, researchers tried to mimic the structures of human brain network to create an artificial network with high functionality (Dongare, Kharde and Kachare, 2012; Maghrabi, 2019).

In an imitation of human neural networks, artificial neural network architectures consist of nodes (neurons) connected through weighted networks. For example, a simple three-layer ANN architecture consists of these nodes arranged in layers. The first layer, known as the input layer, is where nodes receive data from input sources. The computational procedure is carried out in the second layer, which is known as the hidden layer. The output layer, which is the final layer, produces the output results (Dongare, Kharde and Kachare, 2012).

ANNs do not precisely solve problems in a pure mathematical sense. However, they have data

processing features that can estimate the solution to a specific situation (Dongare, Kharde and Kachare, 2012). ANNs can predict the connection between input data features to give an estimated result. Therefore, ANN algorithms have been applied in various fields, such as weather forecasting and health clinical studies (Abiodun *et al.,* 2018). In structural bioinformatics, different ANN methods have been designed to achieve high performance and predict accurate models of protein structure (Pakhrin *et al.,* 2021). NN methods have been integrated into 3D protein model QE methods in order to improve their accuracy.

A simple neural network is the multi-layer perceptron (MLP). The MLP architecture is composed of multiple layers connected in a feedforward direction. The layers are classified into input, output, and several hidden layers, where each layer consists of an array of nodes (Manaswi, 2018; Maghrabi, 2019; Chatterjee, Saha and Mukherjee, 2022; Singh and Ranjan, 2022; Yang and Ma, 2022). Standard MLPs are typically trained by applying the backpropagation algorithm to improve learning and reduce the output errors. During training phase, the MPL learns by comparing its predicted output with the desired value of the target function, e.g., the observed score. Using this comparison, when errors are detected, the weights are adjusted to obtain an accurate output that is closer to the actual value (Manaswi, 2018; Rana *et al.,* 2018; Maghrabi, 2019; Singh and Ranjan, 2022). In addition, the hyperparameters of the MLP can be tuned to optimise the learning process. The hyperparameters that could be changed include the number of neurons in hidden layers, the learning rate, and the number of iterations required for learning the MLP (Maghrabi, 2019).

3.2 Aims and Objectives

The primary goal of this chapter is to examine the potential of using a consensus of CDA scores for enhancing ModFOLD9 local model quality estimation performance. A consensus CDA score was generated to improve the accuracy of 3D models by detecting local errors. The five

new alternative CDA scores were developed, which measured the agreement between the predicted contact scores from five different contact prediction methods and the contacts in each 3D model, were used in addition to the original CDA score. The deep learning-based contact prediction methods were chosen based on the assessment of CASP13 and CAPS14. We selected the top available methods, which included DeepMetaPSICOV (Kandathil, Greener and Jones, 2019), SPOT-Contact (Hanson *et al.,* 2018), trRosetta2 (Anishchenko *et al.,* 2021), TripletRes (Li *et al.,* 2021a) and DeepDist (Wu *et al.,* 2021). To integrate the CDA scores, two versions of the MLP (multilayer perceptron) were applied to output two local quality scores. The aim of the first version was to learn from the input CDA scores to produce the S-score as the target function, while the aim of the second version was to predict the lDDT score as the target function. The other objective was to optimise ModFOLD9 performance by tuning the MLP's parameterisation. A set of MLP hyperparameters including the number of hidden neurons, the learning rate, the error rate and iteration number, were all tuned during training.

3.3 Methods

3.3.1 Data Set

The raw data were protein target models generated from modelling methods used in the CASP14 experiment. The data involved 70 targets, 27775 models and 3087364 residues. The CASP14 models were divided into three sets for the training and testing stages using a three-fold cross-validation method. Thus, each test set contained models for targets that were not in the other two sets, which were used for training. Hence, all data were considered in the training process leading to three different sets of NN weights, which could be used for testing all models (Maghrabi, 2019).

3.3.2 Consensus CDA Score

The CDA score is a pure-single model method designed to leverage residue-residue contact prediction to improve the local quality estimation of 3D models. This score has been used in previous ModFOLD versions. In the current version, five CDA scores were developed using the five top-performing deep learning-based contact prediction methods in CASP13 and CASP14. These scores were combined with the original CDA pure as inputs to a NN that was trained to learn two local quality scores (S-scores and lDDT scores) (Figure 3.1). The deep learning-based contact methods used for each new CDA score included DeepDist (Wu *et al.,* 2021), TripletRes (Li *et al.,* 2021a), trRosetta2 (Anishchenko *et al.,* 2021), SPOT-Contact (Hanson *et al.,* 2018) and DeepMetaPSICOV (Kandathil, Greener and Jones, 2019). The original CDA score used the MetaPSICOV method (Maghrabi and McGuffin, 2017). The CDA was a measure of the agreement between the contact scores predicted from the target sequence and the contacts in each model calculated using the Euclidean distance. If the distance between the C-alpha atoms of residue pairs in the model was less than 8 Angstroms (8Å), then the two residues were defined to be in contact; otherwise, they were non-contacting. To compute the

CDA score for each residue, the model's contact scores for a residue were compared with the predicted contact scores according to each contact prediction method. In other words, if residue (i) was in contact with residues (j) and (k) from a model, and the p-value from the contact prediction method indicated that residue pair (ij) and (ik) were in contact, the CDA score for residue (i) was computed using the following formula:

$$\boldsymbol{C = \sum p/numC}$$

Where $\boldsymbol{p}$ represents the probability (p-value) that the two residues were predicted to be in contact based on the contact prediction algorithms, and $\boldsymbol{numC}$ is the total number of contacts for the model's residue (i), from which a p-value is obtained. The global CDA score could be obtained by adding the CDA values for all residues and dividing the total by the target sequence's length (L) as $CDA = \sum C/L$ (Maghrabi, 2019). The CDA scores computed using each contact prediction method were assigned according to the name of the method (see Table 3.1). Integrating the five CDA scores with the original CDA score into a NN produced a consensus CDA score for ModFOLD9.

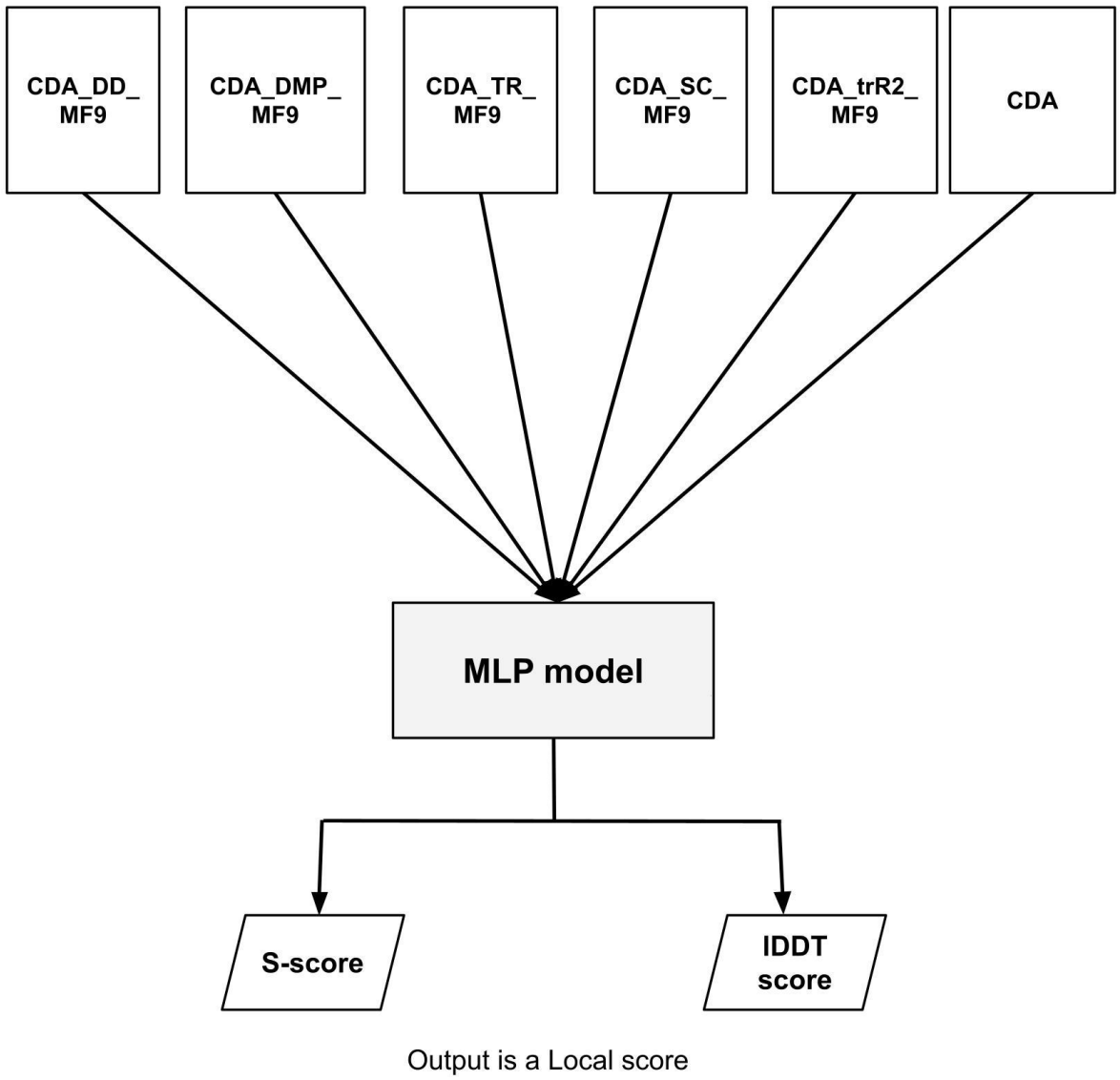


Figure 3.1. A simplified flowchart illustrating the consensus Contact Distance Agreement (CDA) approach to improve the local model quality estimates by ModFOLD9. Six CDA scores were measured according to six deep learning-based contact prediction methods. Each CDA score was assigned based on the method's name (Table 3.1). The scores were fed into an MLP to predict per-residue score; S-score or IDDT score.

Table 3.1. The CDA score names assigned according to their contact prediction methods for use with ModFOLD9.

CDA score	Contact prediction methods
CDA_DD_MF9	DeepDist
CDA_DMP_MF9	DeepMetaPSICOV
CDA_TR_MF9	TripletRes
CDA_SC_MF9	SPOT-Contact
CDA_trR2_MF9	trRosetta2

3.3.3 Neural Network Architecture

The architecture of MLP was similar in construction to that applied in the eighth version of ModFOLD. It was composed of three basic layers; the input, output and hidden layers (Maghrabi, 2019). The MLP was implemented using the RSNNS package in R. The input data were the six CDA scores measured according to the six deep learning-based contact prediction methods. As in ModFOLD versions 6 to 8, an input sliding window size of 5 was used centred on each residue score with zeros used for padding the end residues. Then the residue scores from the six CDA methods were taken, leading to 30 inputs (6x5) for each residue. In the training stage, two MLP versions were implemented where the first MLP was trained to learn the S-score and the second MLP was trained to learn lDDT score. Therefore, the output was a single scoring value for each version; either the S score or the lDDT score. To achieve the optimal performance, the hyperparameters, including the learning rate, the number of neurons in the hidden layer, the number of iterations, and the error rate, were changed during the tuning process. The values of the hyper-parameters that were selected to train the MLP are shown in

Table 3.2. To control for the effect of tuning each hyperparameter, one hyperparameter was changed during each implementation while the other hyperparameters were kept fixed. Initially, 15 hidden neurons were implemented and then altered to produce S-score. Other hyperparameters were also set to specific values, with the learning rate at 0.01, the error rate at 0.01, and the iteration number at 3 as defaults. MLP hyperparameters were set at 15 hidden neurons, 0.1 learning rate, 0.1 error rate, and 4 iterations for the lDDT score. The selection of initial hyperparameter values for the MLP was based on recommendations from previous studies and empirical evidence. We set the number of hidden neurons and learning rate using values commonly suggested in the literature (Sheela and Deepa, 2013; Wang *et al.,* 2018; Niu *et al.,* 2021). For the initial number of hidden neurons, we chose half of the maximum number of neurons. Similarly, we selected an initial learning rate of 0.1, based on empirical evidence suggesting that the learning rate for deep learning models typically falls within the range of 0.01 to 0.1 (Wang *et al.,* 2018; Niu *et al.,* 2021). We set the initial error rate to 0.1, as it falls within the effective range of 0.01 to 0.1 identified by previous research (Hansen and Salamon, 1990). Regarding the initial value of iteration, no research study advised a specific starting point. Therefore, the initial value of iteration was chosen randomly. The training runs were repeated up to 3 times for each combination of parameters, and the NN weights were saved for the highest-performing runs. This work was completed in collaboration with Megan Hird, an undergraduate student, and some of the data shown here was also presented in her final year project. Megan conducted the analysis of fine-tuning MLP hyperparameters for predicting the lDDT score, and the results of her analysis have been presented in the Results and Discussion section.

Table 3.2. Value ranges of hyperparameters that were applied during MLP training process for the consensus CDA approach for ModFOLD9.

Hyperparameter	Value range
Neuron number	8, 9, 10, 11, 12, 15, 20
Learning rate	0.02, 0.03, 0.05, 0.06, 0.1, 0.15, 0.2
Error rate	0.01, 0.02, 0.05, 0.1
Iteration	2, 3, 4

3.3.4 Evaluation Measurements

The performance of the consensus CDA approach was evaluated by comparing the predicted output from the MLPs with the observed S-scores and lDDT scores. The performance was also benchmarked against another high performing single-model method, VoroMQA (Olechnovič and Venclovas, 2017).

The linear and non-linear correlations of predicted and observed residue scores were measured using the Pearson's R and Spearman's Rho values. This assessment measured method performance in terms of how strongly predicted scores correlate with the observed scores (Maghrabi, 2019). The Pearson's R correlation coefficient represents the strength of the linear relationship between predicted quality scores and observed scores, assuming that both scores are normally distributed and that the relationship between them is linear. In contrast, Spearman's (Rho) correlation coefficient is a non-parametric test that examines the non-linear relationship between predicted and observed scores without relying on any presumptions about the nature of the bivariate distribution. The correlation results provide insights into how accurately the protein models are assessed by the prediction method. The correlation coefficient ranges from -1 to 1, with a value close to 1 indicating a strong positive correlation between the predicted quality scores and the observed quality scores. On the other hand, a coefficient close

to -1 shows a strong negative correlation, which means that there is an inverse relationship between the predicted and observed scores. If the coefficient is around 0, it suggests no correlation between the prediction quality scores and the observed quality scores, indicating that they are not related to each other (Bolboaca and Jantschi, 2006; Maghrabi, 2019; Chen *et al.*, 2021).

Additionally, ROC analysis was conducted in this study. This evaluation method is a common practice in previous CASP experiments to evaluate different methods' ability to identify the most accurate models (Kryshtafovych, Fidelis and Tramontano, 2011; Kryshtafovych *et al.*, 2014, 2016; Kryshtafovych, Monastyrskyy and Fidelis, 2016; Elofsson *et al.*, 2018; Cheng *et al.*, 2019; Kwon *et al.*, 2021). To identify accurate or inaccurate residue predictions, the AUC measures the area under the ROC curve, which graphs the true positive rate (TPR) versus the false positive rate (FPR) at various thresholds (Won *et al.*, 2019). In this investigation, the ratio of true positives (correctly identified the low-quality residues) to false positives (incorrectly identified the high-quality residues as low-quality) was plotted. The AUC scores were calculated at the lDDT threshold of 0.6 to determine the method's performance in distinguishing between low and high-quality residues. A residue was considered low-quality if the lDDT was below 0.6. The AUC ranges from 0 to 1, which signifies the prediction performance. A value close to 1 indicates excellent discrimination between low and high-quality residues, with a high TPR and a low FPR.

The study also considered ROC analysis with a low FPR of less than 0.1. Restricting the FPR to 0.1 and calculating the AUC within this range helps evaluate the effectiveness of the prediction method in distinguishing low- from high-quality residues while reducing the number of false positives. The standard ROC plots in this study have consistent scales, where the x- and y-axes range from 0 to 1, whereas the zoomed-in versions (ROC plots at FPR <= 0.1)

employed a smaller scale. This smaller scale has been applied in a previous version of ModFOLD (Maghrabi and McGuffin, 2017). The smaller scale provides a narrower view to emphasize the methods' performance in a low FPR region, allowing for a detailed comparison of the methods' performance.

3.4 Results and Discussion

3.4.1 The Hyperparameter Tuning Process

Since the aim of our study is to eventually improve the local model quality estimates in ModFOLD9 by integrating various contact prediction methods, the hyperparameter tuning process was conducted to determine the optimal performance for the MLP neural network. The performance of different hyperparameters during were compared according to the evaluation scores. To achieve this, other hyperparameters were fixed to change the number of hidden neurons during the implementation. Once the optimal number was achieved, the next hyperparameter was altered. Two correlation measures were computed, Pearson's R Correlation Coefficient and Spearman's Rho Rank Correlation, to assess the hyperparameter's performance, in addition, ROC analysis was considered to evaluate the effect on performance from tuning the MLP hyperparameters.

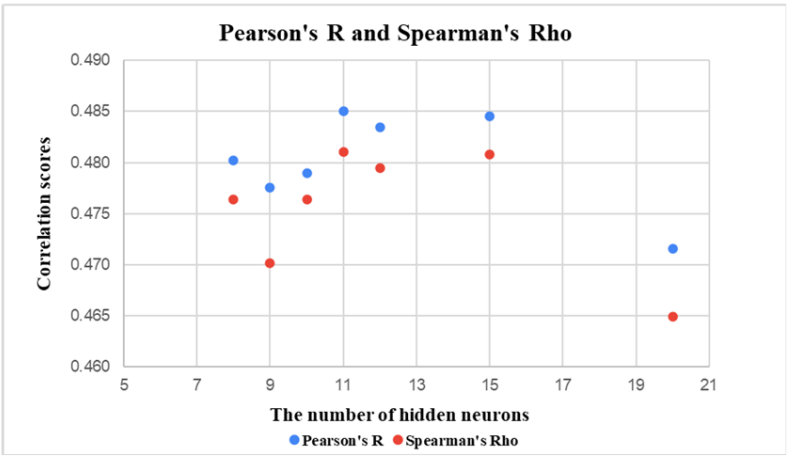
3.4.1.1 The Number of Neurons in Hidden Layers

We focused firstly on the optimal number of hidden neurons while keeping all other hyperparameters fixed. This is because the number of hidden neurons could determine the MLP's capacity for learning from input data (CDA scores) and result in the highest assessment scores. The initial number of hidden neurons was 15 while other hyperparameters were fixed at 0.01 for the learning rate, 0.01 for the error rate, and 3 for the iteration number. According to the S-score, in Figure 3.2A, the correlation scores varied with different numbers of neurons in the hidden layer. Correlations decreased when the number of neurons was <11, and the

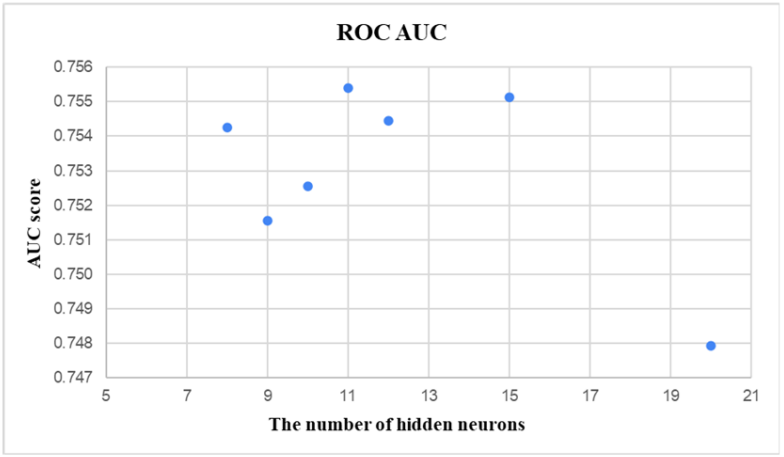
predicted and observed scores were more strongly correlated when neurons increased beyond 11. However, increasing the number of neurons too far adversely effected the MLP performance as we can see in the same figure. The correlations were weaker with 20 hidden neurons, which may have reduced the ability to generalise on the contact prediction data in the fixed number of cycles. Hence, the 11 neurons in the hidden layer were considered to be the optimal number to achieve the high performance according to Pearson's R and Spearman's Rho correlation scores (0.485, 0.481). The ROC AUC scores show similar trend with varying hidden neurons (see Figure 3.2B). The highest AUC scores were achieved when the MLP set at 11 neurons, which supports the results obtained with correlation scores. However, in contrast, at FPR<=0.1, the ROC AUC scores reached a higher value when the number of hidden neurons was increased to 20 (see Figure 3.2C). Although the ROC AUC FPR<=0.1 score at 11 neurons was slightly less than at 20, the other evaluation scores were improved at the same number. On balance, this may suggest that the optimal number of neurons in the hidden layer for these data should be 11.

The optimal number of hidden neurons was also determined in order to achieve the best performance according to the lDDT observed scores. Initially, 15 hidden neurons were implemented, followed by increases and decreases, while 0.1 learning rate, 0.1 error rate, and 4 iterations were the rest of the hyperparameters. Figure 3.3A shows how the MLPs performance on lDDT scores changed with different numbers of hidden neurons. The best performance was achieved when the number of hidden neurons was 15, according to the highest correlation scores. In Figure 3.3B, the best-performing configuration for the MLP was achieved when it had 15 hidden neurons, achieving the highest ROC AUC score. At FPR <=0.1, the ROC AUC scores reached a slightly higher value with 10 hidden neurons (Figure 3.3C). However overall, the optimal number of hidden neurons for producing the best lDDT scores is 15.

A



B



C

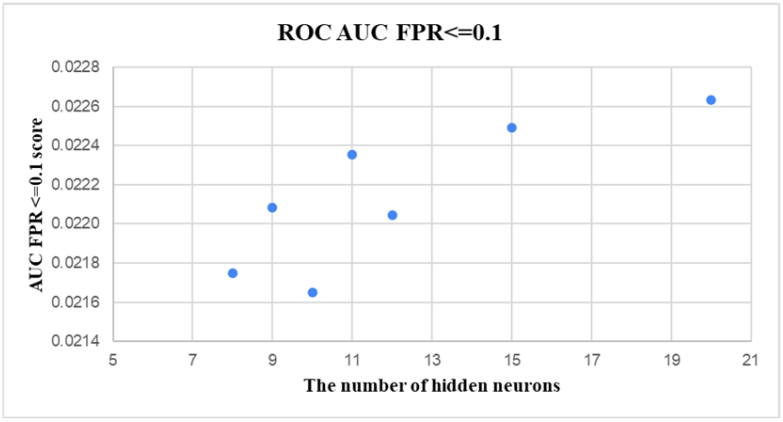


Figure 3.2. The effect of tuning the number of hidden neurons on the consensus CDA MLP performance according to the S-score. (A) Pearson's R and Spearman's Rho correlation scores versus the number of hidden neurons. (B) ROC AUC scores versus the number of hidden neurons. (C) ROC AUC FPR<=0.1 scores versus the number of hidden neurons.

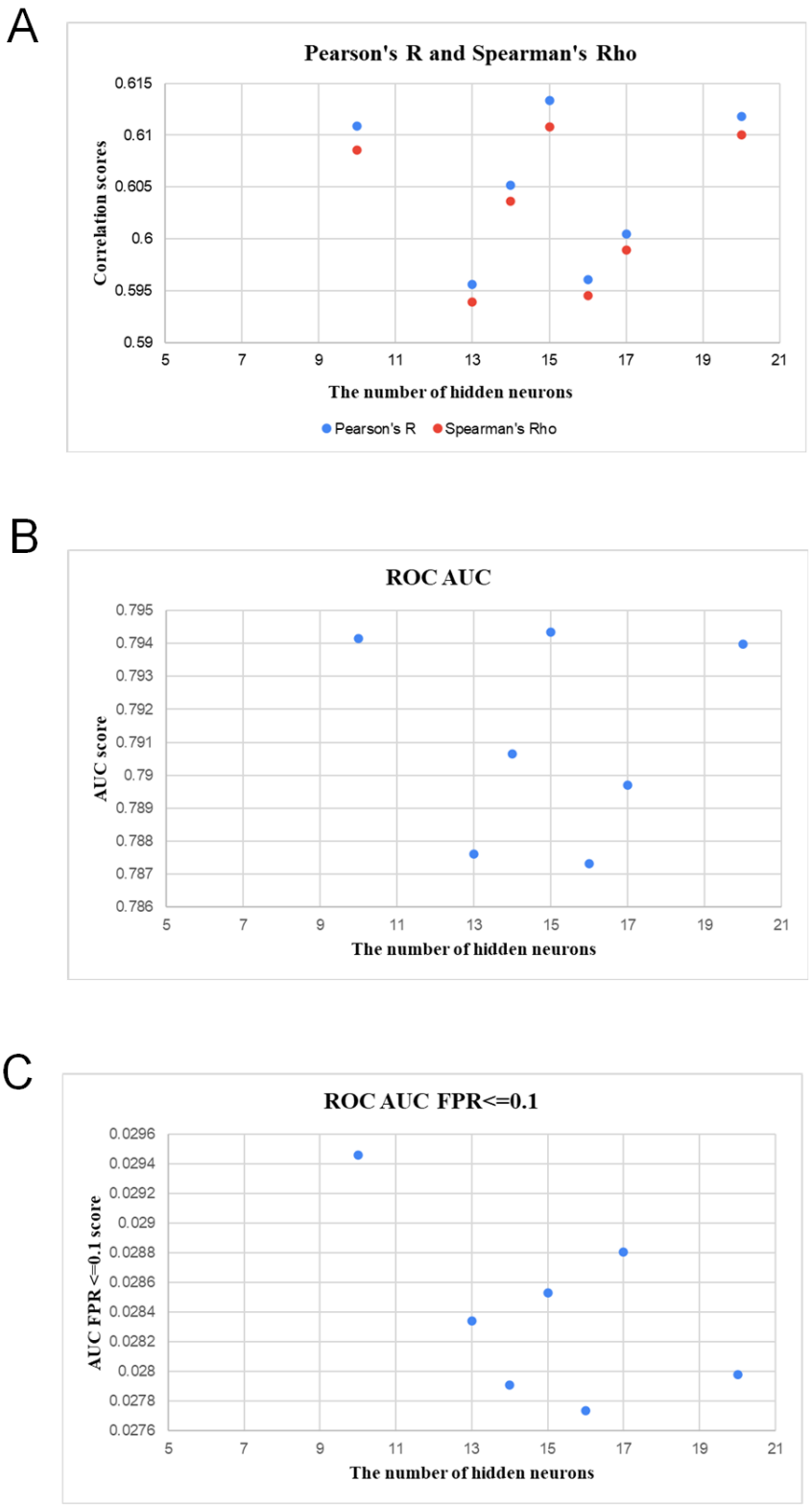


Figure 3.3. The effect of tuning the number of hidden neurons on the consensus CDA MLP performance according to lDDT score. (A) Pearson's R and Spearman's Rho correlation scores versus the number of hidden neurons. (B) ROC AUC scores versus the number of hidden neurons. (C) ROC AUC FPR<=0.1 scores versus the number of hidden neurons.

3.4.1.2 The Learning Rate

The learning rate is one of the vital parameters that has a great effect on MLP performance. Setting learning rate is one of the most challenging to obtain the optimal scores. In this study, a varied range of learning rate was tested to determine the best value for learning rate. For S-scores, we started with 0.1 and then increased and decreased to assess the differences MLP performance according to the evaluation scores. The results show that the best value for learning rate was 0.05 whereas the worst value was 0.15 for correlation scores (see Figure 3.4A). The similar results were shown for the ROC AUC scores, whereas for ROC AUC FPR<=0.1 it was highest at 0.03 (see Figures 3.4B and 3.4C). In terms of the lDDT scores, tuning the learning rate resulted in a different optimal value. In Figures 3.5A and 3.5B, the correlation scores and ROC AUC scores reached the highest values when learning rate was set at 0.07. For the ROC AUC FPR<=0.1, the best learning rate was 0.06 (Figure 3.4C). Based on these finding, overall, the optimal learning rate was chosen as 0.07.

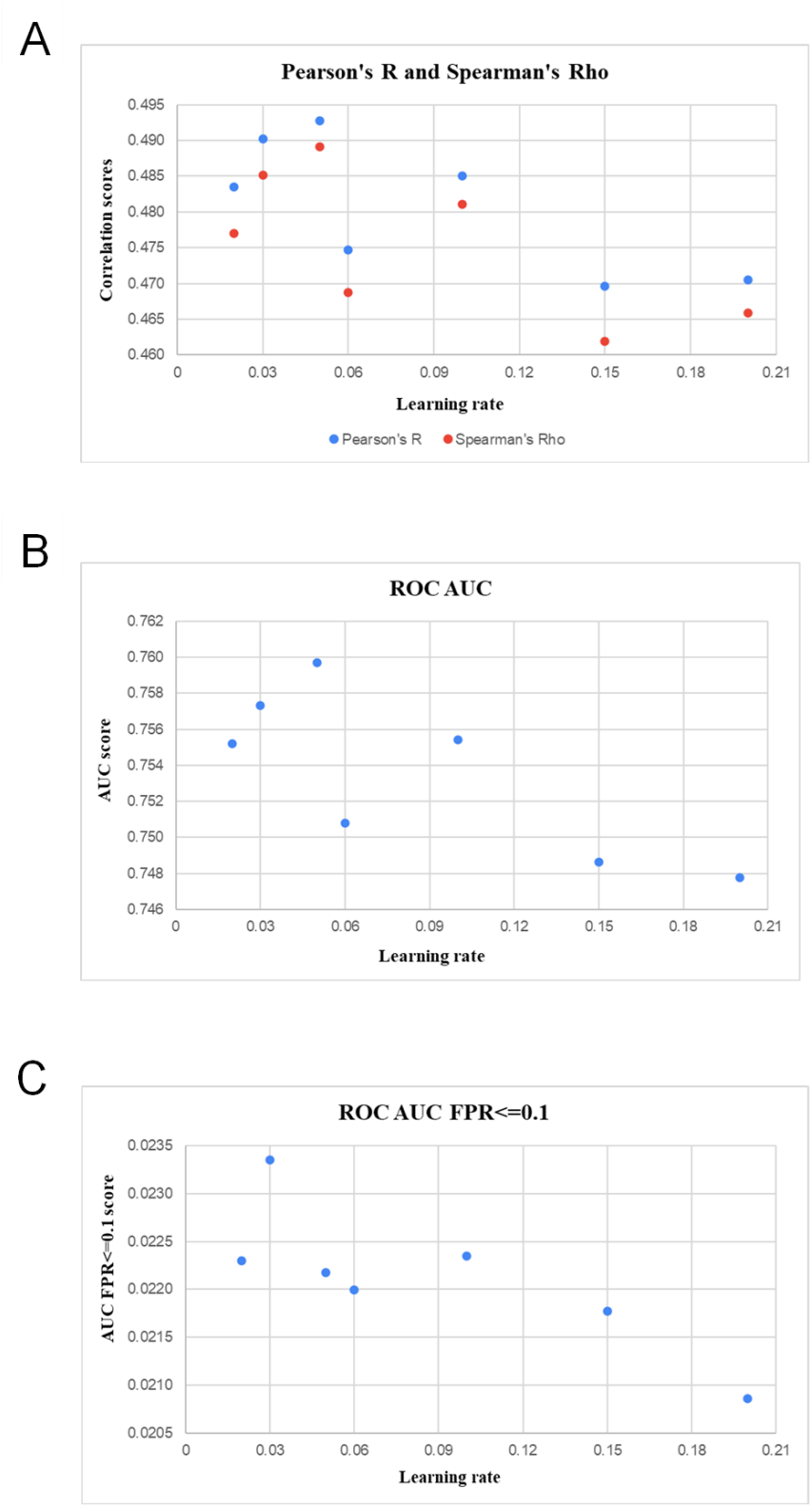


Figure 3.4. The effect of tuning the learning rate on the consensus CDA MLP performance according to S-score. (A) Pearson's R and Spearman's Rho correlation scores versus the learning rate. (B) ROC AUC scores versus the learning rate. (C) ROC AUC FPR<=0.1 scores versus the learning rate.

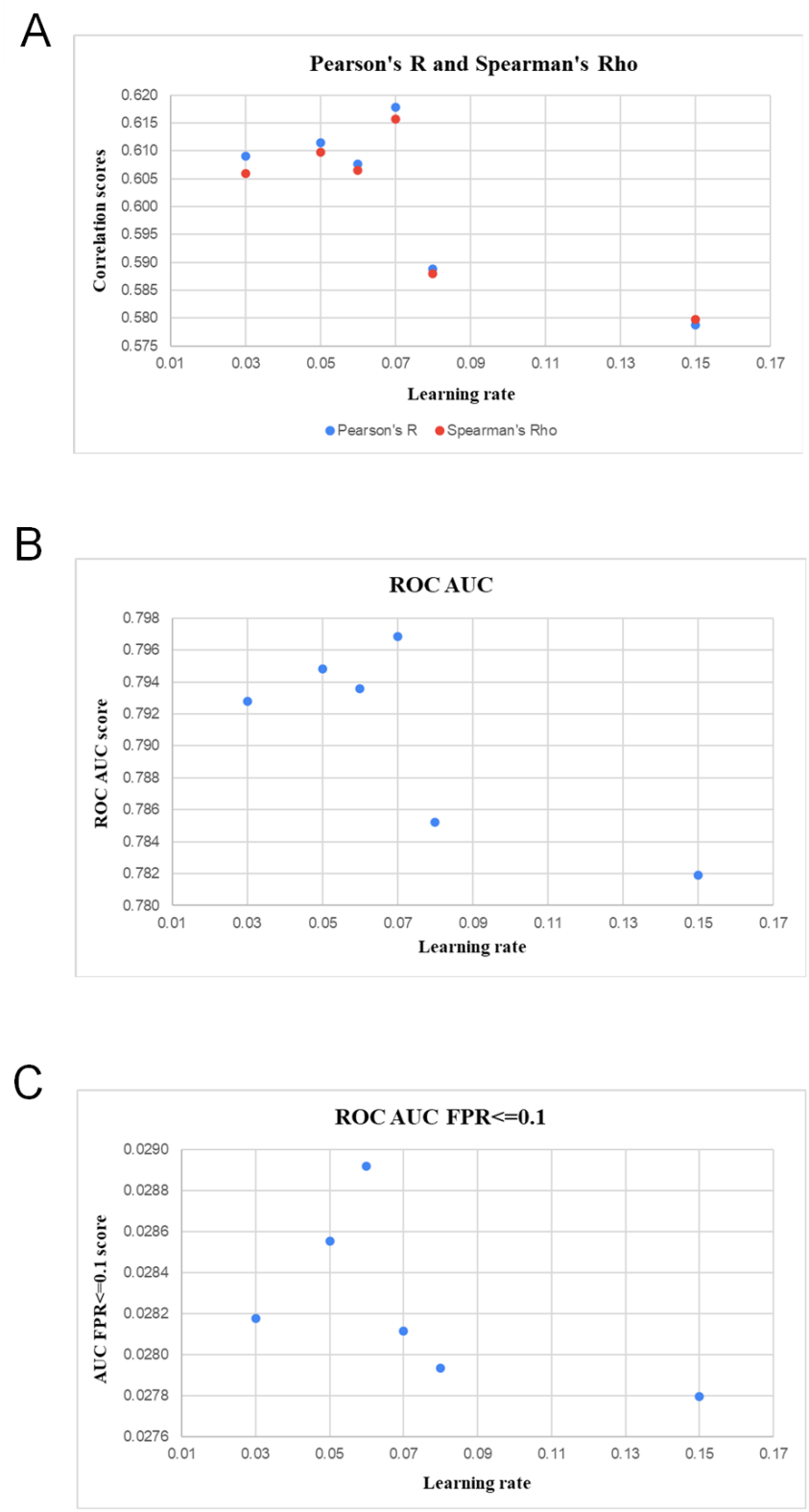


Figure 3.5. The effect of tuning the learning rate on the consensus CDA MLP performance according to lDDT score. (A) Pearson's R and Spearman's Rho correlation scores versus the learning rate. (B) ROC AUC scores versus the learning rate. (C) ROC AUC FPR<=0.1 scores versus the learning rate.

3.4.1.3 Fine Tuning of The Error Rate and Number of Iterations

After setting the optimal values of neuron number and learning rate at 11 and 0.05 for the S-score, the error and iteration values were fine-tuned. Table 3.3 shows the evaluation scores for the consensus CDA MLP according to the S-score with varying error rates and iterations. Overall, the data show that the MLP learned better with 0.01 error with three iterations based on the evaluation scores. However, according to the ROC FPR<=0.1 data a slightly higher score (0.023) was achieved with error rate of 0.05, but with the same number of iterations. This means that showing the MLP the dataset 3 times was sufficient to achieve the optimal accuracy and that any further iterations may result in overfitting. Overall, the optimal hyper-parameters for predicting S-score with the consensus CDA MLP were found to be: 11 hidden neurons, a 0.05 learning rate, a 0.01 error rate and three iterations.

Table 3.3. The effect of tuning the error rate and iterations on the consensus CDA MLP performance according to the S-score. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC and ROC AUC FPR<=0.1 scores. The hidden neuron numbers and learning rate were set at 11 and 0.05, respectively. Error values and iteration numbers were adjusted, and their evaluation scores were measured individually.

Hidden Neuron number	Learning Rate	Error	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
11	0.05	0.01	3	**0.4928**	**0.4892**	**0.7597**	0.0222
11	0.05	0.01	2	0.4859	0.4803	0.7559	0.0227
11	0.05	0.01	4	0.4754	0.4693	0.7502	0.0229
11	0.05	0.05	3	0.4907	0.4855	0.7577	**0.0231**
11	0.05	0.05	2	0.4818	0.4759	0.7538	0.0222
11	0.05	0.05	4	0.4854	0.4814	0.7547	0.0230
11	0.05	0.1	3	0.4916	0.4883	0.7580	0.0228
11	0.05	0.1	2	0.4846	0.4788	0.7551	0.0229
11	0.05	0.1	4	0.4790	0.4733	0.7526	0.0221

For the lDDT score, in Table 3.4, the performance was highest with a 0.05 error rate according to all measures, except for the AUC score of ROC FPR<=0.1, which was highest with a 0.15 error rate. However, 0.05 was chosen as the optimal error value, as the difference was ~0.0005 between the two error rates. In Table 3.5, it can be seen that the best scores of evaluation matrices were achieved at 4 iterations, except for the AUC score of ROC FPR<=0.1 (3 iterations). Therefore, the optimal hyper-parameters for the consensus CDA MLP to predict the lDDT score were chosen to be 15 hidden neurons at a 0.07 learning rate with a 0.05 error rate and 4 iterations.

Table 3.4. The effect of tuning error rate on the consensus CDA MLP performance according to the lDDT score. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC and ROC AUC FPR<=0.1 scores. The hidden neuron numbers and learning rate were set at 15 and 0.07, respectively. Error values were adjusted, and their evaluation scores were measured individually.

Hidden Neuron number	Learning Rate	error	iteration	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
15	0.07	0.1	4	0.5961	0.5950	0.7890	0.0292
15	0.07	0.15	4	0.6095	0.6076	0.7945	**0.0294**
15	0.07	0.05	4	**0.6125**	**0.6116**	**0.7956**	0.0289

Table 3.5. The effect of tuning the iteration on the consensus CDA MLP performance according to the lDDT score. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC and ROC AUC FPR<=0.1 scores. The hyper-parameters were set to 15 neuron numbers, a 0.07 learning rate, and a 0.05 error rate. Iteration numbers were adjusted, and their evaluation scores were measured individually.

Hidden Neuron number	Learning Rate	error	iteration	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
15	0.07	0.05	4	0.6125	**0.6116**	**0.7956**	**0.0289**
15	0.07	0.05	5	0.5993	0.5983	0.7878	0.0266
15	0.07	0.05	5	0.5977	0.5980	0.7898	0.0284
15	0.07	0.05	3	0.6097	0.6077	0.7939	**0.0289**
15	0.07	0.05	3	**0.6128**	0.6108	0.7940	0.0275

3.4.1.4 The effect of tuning CDA MLP hyperparameters on the performance of ModFOLD9

ModFOLD9's predictive performance relies on carefully tuning the MLP's hyperparameters during training. By optimising four specific hyperparameters through experimentation and validation, the MLP's performance was significantly improved. This, in turn, led to an enhancement in ModFOLD9's predictive assessment. The number of hidden neurons in a MLP directly affects its predictive capabilities and accuracy. The MLP with 11 hidden neurons better predicted the S-score, whereas, with 15 hidden neurons, the MLP could predict the lDDT score better. These observations suggest that the chosen values (11 and 15 neurons) improved the MLP's capacity to recognise complex patterns in data, resulting in improved S-score and lDDT score predictions. However, increasing the number of hidden neurons does not always prove beneficial. Overfitting, a common problem in machine learning, occurs when a model captures the noise and the underlying patterns in the training data. An overfitted model performs well on training data but poorly on unseen or validation data because it memorises the training data instead of generalising it (Awad and Khanna, 2015; Ying, 2019; Chasiotis, Nadi and Filios, 2021; Zhao *et al.*, 2023). This is exemplified by an MLP with 20 hidden neurons performing poorly when predicting S-scores, indicating potential overfitting. Hence, while increasing the hidden neurons of the model can enhance its predictive power to an extent, caution should be exercised to avoid overfitting.

The learning rate controls how much the network weights are adjusted during the learning process. Appropriate adjustment of the weights enables the MLP to converge towards an optimal solution, providing a better fit to the data (Zubair *et al.*, 2014; Awad and Khanna, 2015; Mukhtorov *et al.*, 2023). However, it is important to note that the optimal learning rate could vary based on the specific task and data, and finding the right learning rate often requires careful tuning and experimentation. In the case of predicting the S-score, a learning rate of 0.05 yielded optimal performance. However, for predicting the lDDT score with the same input

combination, a learning rate of 0.07 was preferable. These results highlight the importance of selectively adjusting learning rates depending on the specific task. Different MLP training scenarios may require vary learning rates to maximise the MLP's potential to learn effectively from the data and converge to a point where it efficiently predicts the S-score or lDDT score. Hence, choosing an appropriate learning rate is essential for improved MLP performance.

The error rate is the difference between predicted and observed output (Zubair *et al.*, 2014; Awad and Khanna, 2015; Elansari, Ouanan and Bourray, 2023). During training, the goal is to minimise this difference for better predictive accuracy. Gradually reducing the error rate can lead to higher accuracy, but choosing the right rate requires careful tuning. For instance, the MLP's performance improved when the error rate decreased from 0.1 to 0.05 while predicting the IDDT. Similarly, gradually reducing the error rate from 0.1 to 0.01 produced the best MLP performance for the S-score prediction, according to the evaluation results. This indicates that adjusting the error rate carefully can result in greater predictive accuracy. However, like learning rates, selecting the appropriate error rate requires precise tuning and experimentation because the optimal error rate may vary depending on the specific task and data.

MLP's performance heavily depends on the number of iterations during training. Each iteration involves a complete pass through the training set, during which the MLP adjusts its weights to enhance its learning capability. An MLP cycles through an entire dataset based on the number of iterations. Using too few iterations leads to underfitting, while too many iterations cause overfitting (Bengio, 2012). For the ModFOLD9 MLP, 3-4 iterations were optimal. Fewer or more iterations than this range resulted in the MLP learning noise instead of underlying patterns, resulting in less accurate predictions of the quality scores. Hence, adjusting the iteration values contributed to further improvement in the predictive learning of MLP during the experiments.

By adjusting these hyperparameters, the MLP could learn complex patterns in the data without overfitting. By using cross-validation, we could assess the MLP's performance on new data and demonstrate the reliability and effectiveness of the hyperparameter-adjusted MLP in the ModFOLD9. Hence, cross-validation and optimised hyperparameters improved MLP's learning and predictive ability, enhancing ModFOLD9's predictive assessment.

3.4.2 Evaluating MLP Performance lDDT and S-score Performance

In our study we are aiming to enhance the local quality estimation of ModFOLD9 by integrating CDA scores, as the correct contacts are an important aspect of 3D model quality. Local errors in 3D models were detected through estimating the distance between each residue in a model and its corresponding residue in native structure. These predicted scores were then compared with the observed S-score and lDDT scores for each residue in the model. The cross-validated consensus CDA MLPs were then compared with the component individual scoring methods in terms of their correlations and ROC performance based on the observed quality scores.

Tables 3.6 and 3.7 show the comparison between individual CDA methods, the consensus CDA MLP method (Consensus_CDA_ONLY_MF9) and the single-model method, VoroMQA, which is a leading single-model quality estimation method (Olechnovič and Venclovas, 2017). By comparing the Consensus_CDA_ONLY_MF9 method with individual methods, it is clear that the consensus CDA score achieved higher performance than the individual CDA scores according to both the S-score and lDDT score. Importantly, the Consensus_CDA_ONLY_MF9 method outperforms the benchmark VoroMQA method, which indicates the clear added value gained from combining all CDA scores, which can be used to enhance the local assessment of 3D models for ModFOLD9 (see subsequent chapters).

Table 3.6. Cross-validation performance benchmark of the consensus CDA MLP method (Consensus_CDA_ONLY_MF9) versus its component CDA methods and the single-model method (VoroMQA) using CASP14 data according to S-score. The evaluation measures are Pearson's R, Spearman's Rho, Receiver-Operating Characteristic Area Under Curve (ROC AUC), and ROC AUC with a False Positive Rate less than 0.1 (AUC FPR <=0.1). The table sorted by Pearson's R values.

Methods	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
CDA_DD_MF9	0.1497	0.1396	0.5825	0.0063
CDA_DMP_MF9	0.2159	0.2086	0.6143	0.0045
CDA_TR_MF9	0.2435	0.2408	0.6358	0.0165
CDA	0.3163	0.2905	0.6501	0.0185
CDA_SC_MF9	0.3622	0.3622	0.6921	0.0125
CDA_trR2_MF9	0.3987	0.4743	0.6959	0.0197
VoroMQA	0.4243	0.4215	0.7267	0.0178
Consensus_CDA_ONLY_MF9	**0.4928**	**0.4892**	**0.7597**	**0.0222**

Table 3.7. Cross-validation performance benchmark of the consensus CDA MLP method (Consensus_CDA_ONLY_MF9) versus its component CDA methods and single-model method (VoroMQA) using CASP14 data according to lDDT score. The evaluation measures are Pearson's R, Spearman's Rho, Receiver-Operating Characteristic Area Under Curve (ROC AUC), and ROC AUC with False Positive Rate less than 0.1 (AUC FPR <0=.1). The table sorted by Pearson's R values.

Methods	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
CDA_DD_MF9	0.1886	0.2069	0.6270	0.0071
CDA_DMP_MF9	0.3021	0.2970	0.6439	0.0085
CDA	0.3432	0.3140	0.6719	0.0177
CDA_TR_MF9	0.3616	0.3786	0.6968	0.0200
CDA_trR2_MF9	0.4590	0.4109	0.6978	0.0203
CDA_SC_MF9	0.4796	0.4814	0.7331	0.0248
VoroMQA	0.4950	0.4963	0.7315	0.0191
Consensus_CDA_ONLY_MF9	**0.6134**	**0.6108**	**0.7944**	**0.0285**

The density scatter plots were utilized to visually illustrate the correlation between Consensus_CDA_ONLY_MF9 and its respective component methods (Appendix 10-11). From the plots, the predicted S-scores produced by the Consensus_CDA_ONLY_MF9 and individual CDA scores have no strong association with the observed S-scores. Although density plots display S-scores' distribution, they do not illustrate the correlation between observed and predicted scores. This is because S-scores tend to have low or high predicted scores, leading to skewness in the data. As a result, density plots of S-score are not very useful in reflecting the relationship between predicted and observed scores. In contrast, there was a slightly correlation showed between lDDT scores of ModFOLD9 and the observed scores.

The Consensus_CDA_ONLY_MF9 method also outperformed the individual scoring methods and VoroMQA based on ROC analysis as shown in Figures 3.6 and 3.7. Based on Figure 3.6, Consensus_CDA_ONLY_MF9 achieved the highest ROC AUC scores (AUC = 0.760, AUC FPR <= 0.1 = 0.022) compared to individual methods and VoroMQA. This indicates that combining CDA scores remarkably improved ModFOLD9's local assessment accuracy, leading to enhanced S-score predictions. Similarly, in predicting the lDDT score, Consensus_CDA_ONLY_MF9 demonstrated the highest ROC AUC scores (AUC = 0.794, AUC FPR <= 0.1 = 0.029), outperforming individual methods and VoroMQA. This suggests that consensus CDA scores enhanced the accuracy of local assessment for predicting the lDDT score.

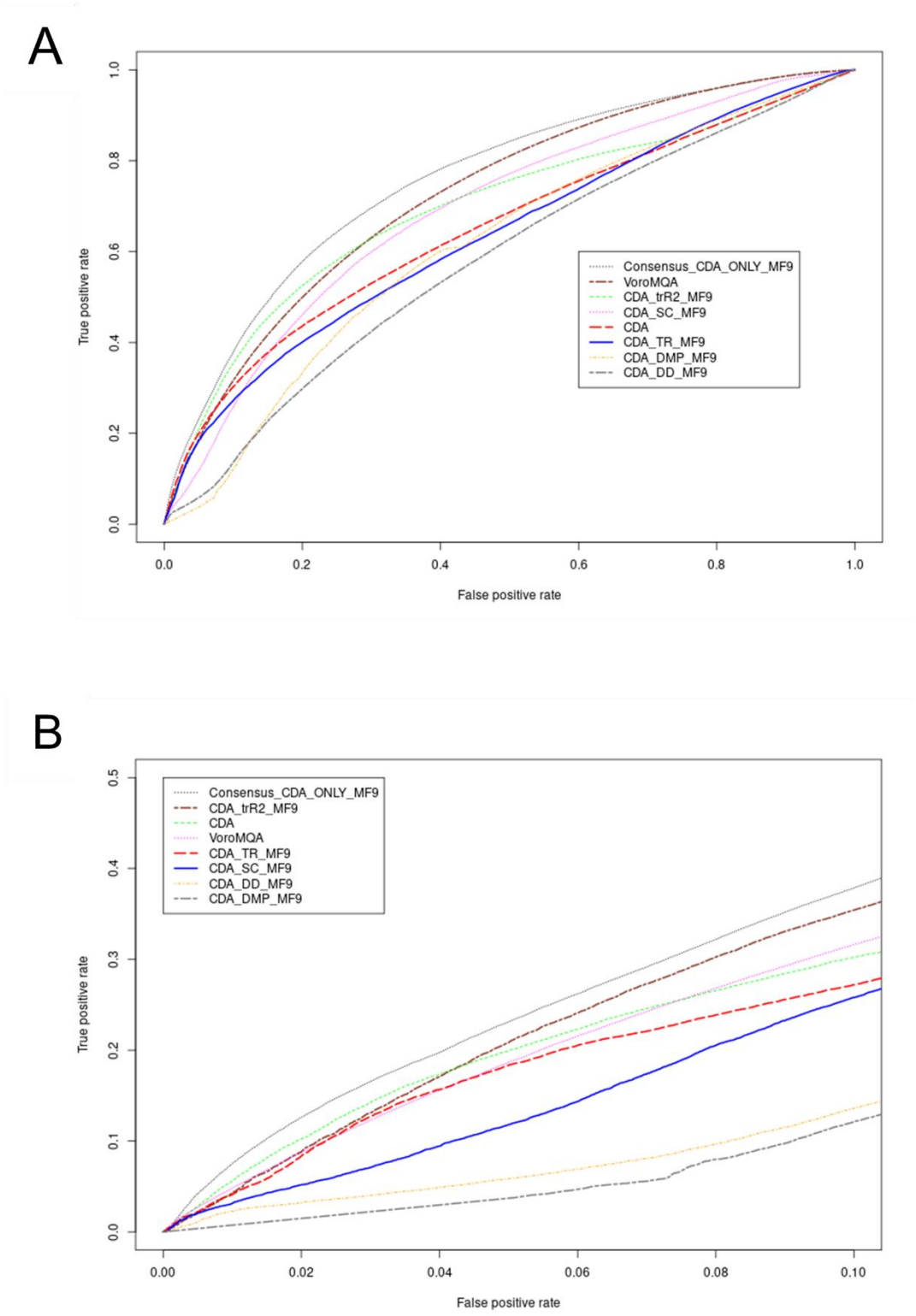


Figure 3.6. ROC curves for the Consensus_CDA_ONLY_MF9 ModFOLD9 against its component methods and VoroMQA method according to S-scores. A) Line graphs of ROC analysis for all methods. B) Line graphs with a condition of false positive rate less than 0.1.

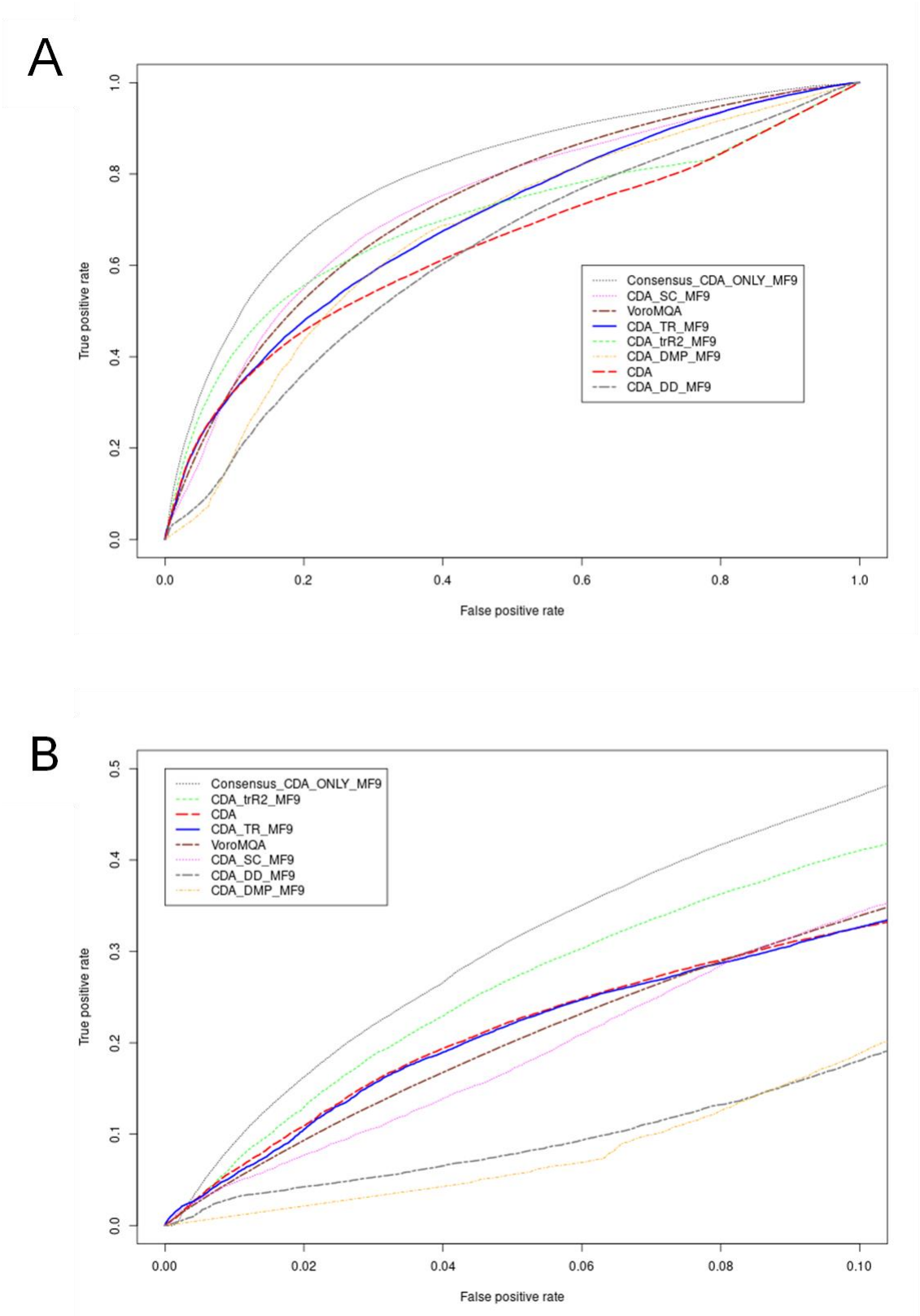


Figure 3.7. ROC curves for the Consensus_CDA_ONLY_MF9 ModFOLD9 against its component methods and VoroMQA method according to lDDT score. A) Line graphs of ROC analysis for all methods. B) Line graphs with condition of false positive rate less than 0.1.

3.5 Conclusion

Estimating the accuracy of 3D models of protein structures is a vital aspect of computational methods for protein structure prediction, as it crucial to know whether or not you can be confident in the prediction. As the estimation of quality accuracy is now a key stage in all protein structure prediction pipelines, many developers have focused on the enhancement of estimation performance by exploiting the advances in contact prediction methods. In our study, we examined the usefulness of consensus-based contact prediction methods for improving the local model quality estimates for integration with ModFOLD9.

The consensus CDA MLP scores (Consensus_CDA_ONLY_MF9) were a substantial improvement compared with the individual scores and importantly the approach also outperformed a leading pure-single model method, VoroMQA, which we used here as a useful benchmark. These results suggest that a consensus of deep learning-based contact methods has potential to boost the estimation accuracy of ModFOLD9. However, further enhancements to the estimation accuracy of ModFOLD9 could be achieved by exploiting the benefits of quality scores produced by both pure-single model methods and quasi-single model methods. In the next chapter, we investigate integrating these new CDA scores with additional new and existing scores and we benchmark the ModFOLD9 development further.

Chapter 4 Development of Consensus QA Methods for the ModFOLD9 Quality Estimation Server

4.1 Introduction

Recent achievements in modelling methods have led to considerable interest in using QA methods to provide independent evaluations. Model quality assessment, which involves estimating the reliability of a protein model, is required so that protein models may be used confidently for biomedical applications (Kwon *et al.*, 2021; Liu, Zhao and Zhang, 2023). Recently, the accuracy of the predicted models has improved significantly owing to due to the advent of modelling methods such as AF2 (Jumper *et al.*, 2021b) and RoseTTAFold (Baek *et al.*, 2021). As such, QA methods may have more difficulty estimating local errors in high-accuracy models as they become more challenging to discriminate. Thus, further enhancement is required to improve the predictive performance of QA methods (Kwon *et al.*, 2021; McGuffin *et al.*, 2021; Liu, Zhao and Zhang, 2023; Zhang, Xia and Shen, 2023).

Consensus approaches have played a significant role in improving protein prediction servers in various aspects. Consensus-based methods are meta-servers of individual methods designed to leverage their strength to boost predictive performance accuracy (Wei, Thompson and Floudas, 2012; Yan and Kurgan, 2015; Reza *et al.*, 2021; Alharbi and McGuffin, 2023). The employment of consensus approaches for QA servers helps to improve their estimates of local errors in 3D models. ModFOLD is a quality assessment server that has been updated continuously using the consensus approach. Previous versions of ModFOLD were enhanced by the addition of various scoring methods as inputs to the MLP neural network, including pure- and quasi-single model methods (Maghrabi and McGuffin, 2017; Cheng *et al.*, 2019; Maghrabi, 2019; McGuffin *et al.*, 2021). With the ninth version of ModFOLD, we aimed to improve the local assessment by producing a consensus CDA score based on the combination of individual CDA scores according to six deep learning-based methods. The consensus CDA score achieved a reliable improvement in the local assessment predictive. Therefore, we were encouraged to add other scoring methods to the consensus CDA score to further investigate the

effectiveness of consensus approaches in improving quality assessment performance.

4.1.1 Integration of The Consensus CDA Methods with Other Leading Established Methods

The primary purpose of local model quality assessment is to detect how much and where the predicted protein model deviates from the native structure with the aim of estimating the accuracy on a per-residue basis. The performance of QE methods have been enhanced by exploiting various protein features and combining scores for these features as inputs to different machine learning approaches (Maghrabi, 2019; Chen and Siu, 2020; McGuffin *et al.*, 2021; Liu, Zhao and Zhang, 2023; Zhang, Xia and Shen, 2023). In this chapter, both pure-single and quasi-single model methods were considered as input scores to leverage their benefit along with the consensus of CDA scores, with the aim of further improving ModFOLD9's local assessment performance. Pure-single model methods can detect local region deviations as they aim to evaluate the single model based on its features. These features include protein sequence and structural properties, indicating the spatial arrangement of protein residues and their distance distribution in a model (Uziela *et al.*, 2016; Uziela *et al.*, 2017; Olechnovič and Venclovas, 2017; Zhang, Xia and Shen, 2023).

Quasi-single model methods were considered as the best alternative to clustering methods in terms of overcoming the latter's limitations. Although clustering-based methods perform well when multiple models for a protein target are available, their performance may be poor when fewer models are available. Nevertheless, quasi-single model methods, such as the ModFOLD approaches pioneered by the McGuffin group, have produced reliable assessments, even when few models are available for each protein target (Kryshtafovych *et al.*, 2014; Cheng *et al.*, 2019). These methods evaluate a single model based on its inherent characteristics in comparison to reference 3D models generated using structure prediction pipelines (Maghrabi and McGuffin, 2017; Maghrabi, 2019; McGuffin *et al.*, 2021; McGuffin *et al.*, 2023).

4.1.2 The Combination of Consensus CDA Scores with Other Pure-Single Model Methods

4.1.2.1 Secondary Structure Agreement (SSA) Score

SSA was a straightforward local quality score that relied on a comparison of each residue's predicted secondary structure based on PSIPRED (Buchan *et al.*, 2013) with that residue's secondary structure state in the model based on the Dictionary of Secondary Structures of Proteins (DSSP) (Kabsch and Sander, 1983). To compute the agreement, the following formula was applied:

$$SSA = p_{CHE}$$

where P_{CHE} is the probability value (p-value) of each residue's predicted secondary structure from PSIPRED for eight DSSP states. The eight states were reduced to three using the standard scheme: coil (C), helix (H), and strand (E), while other states (H, I, G, E, B, S, T, -) were classified as coil (C) (Maghrabi and McGuffin, 2017; Cheng *et al.*, 2019; Maghrabi, 2019; McGuffin *et al.*, 201; McGuffin *et al.*, 2021).

4.1.2.2 The ProQ methods

ProQ family of methods were designed by the Elofsson group to estimate model assessment using a single model. Several different versions of the ProQ method have been developed over the years using the same basic rationale. The general strategy consisted of combining and comparing various protein features based on sequence and structure with machine learning algorithms to predict local errors in 3D models (detailed below). ProQ versions have achieved an excellent assessment performance and have ranked among the leading pure-single model methods for the estimation of quality model accuracy according to CASP experiments (Kryshtafovych, Fidelis and Tramontano, 2011; Kryshtafovych *et al.*, 2014; Elofsson *et al.*, 2018; Cheng *et al.*, 2019; Won *et al.*, 2019). The four major versions of the ProQ method,

ProQ2 (Ray, Lindahl and Wallner, 2012), ProQ2D, ProQ3D (Uziela *et al.*, 2017) and ProQ4, will be integrated with ModFOLD9.

4.1.2.2.1 ProQ2

ProQ2 was a machine learning-based method that applied a SVM to combine scores for assessing 3D models. ProQ2 incorporated both sequence and structural features of the protein target as input for the SVM. The sequence-based properties included predicted secondary structures, predicted surface exposure and conservation, calculated from MSAs. Structural features based on the observed structure include atom–atom interactions, residue–residue contacts and secondary structures. The SVM was trained by performing a linear kernel function, which can capture the linear relationship between residues in a protein model. The output is a local quality score for each residue in a target. To predict the global score, the local scores for target residues were summed and normalised by dividing by the target sequence length (Ray, Lindahl and Wallner, 2012).

ProQ2 was ranked as the top-performing in CASP9 in the quality estimation category (Kryshtafovych, Fidelis and Tramontano, 2011). Ray et. al. (2012) demonstrated that ProQ2's improved accuracy in predicting the quality of 3D models relied on the global features, meaning that protein features were predicted for the entire model. In other words, predicting local quality for a model from the global features perspective may improve the accuracy of the 3D model whereas prediction from the local perspective does not necessarily reflect the whole agreement between the predicted and actual features (Ray, Lindahl and Wallner, 2012). Other features that contributed to ProQ2's improvement were the contact properties between residues and surface area features. These features were re-weighted according to MSAs to improve the predictive accuracy of ProQ2 (Ray, Lindahl and Wallner, 2012). The prediction of each residue's position and the capture of conservation information also contributed to a slight

improvement (Ray, Lindahl and Wallner, 2012).

4.1.2.2.2 ProQ2D and ProQ3D

ProQ2D and ProQ3D were deep learning-based approaches for the estimation of 3D model quality. They represented the updated versions of ProQ2 and ProQ3 (Uziela *et al.*, 2017). ProQ3 was the developed version of ProQ2 using Rosetta full-atom and coarse-grained energy function, as inputs into the SVM along with ProQ2's single model input structural properties (Uziela *et al.*, 2016; Chen and Siu, 2020). In ProQ3D, the input features were similar to that for the ProQ3 input, but it was different in that it employed a deep neural network (MLP) rather than the SVM (Uziela *et al.*, 2017; Hiranuma *et al.*, 2021; Liu *et al.*, 2022; Liu, Zhao and Zhang, 2023). The MLP consisted of one input layer, two hidden layers, and one output layer. The hidden layers had different numbers of neurons: 600 in the first layer and 200 the second. The activation function was a rectified linear unit (ReLU), which can capture the nonlinearity relationship between protein residues in a model (Glorot, Bordes and Bengio, 2011; Uziela *et al.*, 2017; Chen and Siu, 2020). ProQ3D improved significantly with respect to estimating the accuracy of 3D models and was ranked as a top-performing method in CASP13 (Cheng *et al.*, 2019).

4.1.2.2.3 ProQ4

ProQ4's design was different to that of its predecessors with respect to its input features and NN architecture. The input was the predicted structural features of the 3D model using an MSA. From the MSA, two statistics were extracted: self-information and partial entropy. These statistics were used to improve the prediction of the proteins' structural features because they provide additional information about the conservation and variability of amino acids at different positions in the sequence. The structural features were dihedral angles, relative surface area, secondary structure, and hydrogen bonds. All inputs were used to train a deep neural network to predict the quality of protein models (Hurtado, Uziela and Elofsson, 2018).

The NN architecture was a Siamese network with two sub-neural networks wherein the two were identical. In each network, one target model was fed into a size-1 convolution. The convolution's output was visualised in a 64-dimensional space and processed by four ResNet modules. The model's output was combined with alignment features and transmitted via four more ResNet modules. The alignment features were predicted from the protein sequence, including 3- and 6-state secondary structure, surface accessibility and the dihedral angles. The prediction in ProQ4 was performed using the comparative method. To achieve this, the symmetrised perceptron, SortNet, was applied. SortNet was composed of two parallel hidden layers, each of which had 512 neurons per amino acid and included batch normalisation and dropout. This method was used to rank the best target models in addition to predicting their quality. The application of this approach enhanced ProQ4's ranking ability, achieving state-of-the-art performance in protein model quality assessment in CASP13 (Hurtado, Uziela and Elofsson, 2018; Cheng *et al.*, 2019; Chen *et al.*, 2023; Zhang, Xia and Shen, 2023).

4.1.2.3 VoroMQA

VoroMQA was a quality estimation method that relied on a statistical potential of atom interaction frequencies in a protein structure. This method conceptualised protein structures as intersecting spheres of heavy atoms. VoroMQA's essential feature was its ability to extract the interaction between these atoms using an algorithmic method called Voronoi tessellation. Use of this algorithm allowed the determination of contact areas and calculation of the interaction between them. In other words, Voronoi tessellation splits a space into different areas or cells according to established principles. Each cell generated through this procedure for VoroMQA depicts the area of a single atom in a 3D space. VoroMQA can precisely quantify the spatial distribution and interactions of these atoms by tessellating the area that the protein filled and using the heavy atoms as reference points. Obtaining interatomic contact areas with this tessellation yielded valuable information about atom interactions within a specific protein

structure, thus facilitating a more sophisticated and detailed evaluation of protein model quality (Olechnovič and Venclovas, 2017; Hurtado, Uziela and Elofsson, 2018; Liu, Zhao and Zhang, 2023).

4.1.2.4 DeepAccNet

DeepAccNet is a deep learning-based method developed by Baker's research group (Hiranuma *et al.*, 2021). This method was developed primarily to improve the accuracy of protein models by guiding refinement in the Rosetta method. The method used 3D and 2D convolution to assess the local atomic environment of a protein model and determine its global context using the per-residue accuracy and residue–residue distance signed errors (Hiranuma *et al.*, 2021; Guo *et al.*, 2022; Zhang, Xia and Shen, 2023).

The input features include 1D features for each amino acid, 2D features of amino acid residue pairs and 3D features of amino acid distribution within 3D the protein model's space. The one-dimensional features involved the physical and chemical properties of each residue in a protein, backbone angles, Rosetta intra-residue energy terms, and secondary structures. The two-dimensional features were distances between residues, orientations, and Rosetta-based energy terms as well as predictions from trRosetta and embeddings for ProtBert-BFD100; a machine learning model for protein sequences (Elnaggar *et al.*, 2022). The 3D features represented the local atomic coordinates of the amino acids in the 3D model (Hiranuma *et al.*, 2021; Guo *et al.*, 2022; Zhang, Xia and Shen, 2023).

The NNs' architecture comprised distinct dimensional CNNs fitted for each input feature. The first section was a series of 3D convolution layers fed by 'voxelized atomic coordinates' for each residue (Hiranuma *et al.*, 2021). These convolution layers have the same parameters for all residues, allowing them to identify common patterns on a universal scale. The tensor output from each of these layers is then 'flattened' into a 1D vector, converting the complex 3D data

into a more straightforward 1D format so that it might be combined with other 1D features. The network's second section performed a series of 2D convolution operations on the concatenated feature vectors. Input to this section consisted of two matrices: one for 1D features and one for 2D features. In this section, a 1D-feature matrix was tiled along the first and second axes of a 2D-feature matrix, and the resulting matrices were then concatenated to create a new feature matrix. The matrix's third axis represented a combination of 1D and 2D features for each pair of residues. This network formation allowed the method to analyse and extract important features from amino acid sequences (1D) and their interactions (2D) at the same time. Following the formation of the feature matrix, a residual network was split into two arms, each comprising four residual blocks. The purpose of this network was to predict Cβ distance errors and filter critical residue pairs in protein structure refinement (Hiranuma *et al.,* 2021).

The prediction was generated by three variants of DeepAccNet, each of which differs with respect to its 2D input features. The DeepAccNet-MSA variant used information from MSAs, particularly inter-residue distance predictions provided by the trRosetta network. The DeepAccNet-Bert used sequence-embedded data from the ProtBert-BFD100 model (Bert). The third variant, referred to as 'DeepAccNet-Standard', excluded both MSA and Bert embeddings as features (Hiranuma *et al.,* 2021).

4.1.3 The Combination with Quasi-Single Model Methods

The quasi-single model methods evaluated single model quality by comparing each model with a set of reference models generated using tertiary structure modelling approaches. These methods differ with respect to the algorithms and features that they used for model evaluation. Four quasi-single model methods have been integrated into the previous versions of the ModFOLD server: ResQ (Yang, Wang and Zhang, 2016), Disorder B-factor Agreement (DBA), ModFOLD5_single and ModFOLDclustQ (Maghrabi, 2019).

4.1.3.1 ModFOLD5_single, ModFOLDclustQ_single and DBA

Three alternative quasi-single model methods have been developed by the McGuffin group and integrated into the previous versions of ModFOLD (versions 6-8) enhancing the predictive accuracy of model quality estimates (Kryshtafovych *et al.*, 2016; Maghrabi and McGuffin, 2017; Cheng *et al.*, 2019; Maghrabi, 2019; McGuffin *et al.*, 2019; McGuffin *et al.*, 2021). The local quality scores of ModFOLD5_single were computed using the quasi-single model algorithm to evaluate single models with ModFOLDclust2 (McGuffin and Roche, 2010) using reference models generated using the IntFOLD, a structure prediction pipeline developed by the McGuffin group (Maghrabi, 2019; McGuffin *et al.*, 2019; McGuffin *et al.*, 2021; McGuffin *et al.*, 2023). For the ModFOLDclustQ_single scores, the local scores of single models were calculated in comparison with the reference IntFOLD set by employing the local Q-score algorithm (Ben-David *et al.*, 2009; McGuffin and Roche, 2010; Maghrabi and McGuffin, 2017; Maghrabi, 2019; McGuffin *et al.*, 2021). The Q-score is a metric implemented in ModFOLDclustQ method to assess the structural similarity between two protein models based on the spatial distances between the residues in the two structures. This score was produced from the Q measure formulated by the Wolynes group (Eastwood *et al.*, 2001; Ben-David *et al.*, 2009; Maghrabi, 2019). The DBA scores measure the degree of agreement between the predicted per-residue errors in a 3D protein model according to ModFOLDclust_single and the disordered residues in a protein sequence as predicted by DISOPRED3 (Jones and Cozzetto, 2015; Maghrabi and McGuffin, 2017; Maghrabi, 2019; McGuffin *et al.*, 2021).

4.1.3.2 ResQ

The ResQ method was developed to assess the residue-specific quality and its associated B-factor profile in a unified manner. ResQ used information generated during the simulation processes of modelling predictors. The modelling servers predicted 3D models of a particular

protein target using various TBM algorithms and parameters, that led to alternative predicted conformations. The variations of model conformations have valuable information that can be used to predict the quality of a 3D model. In addition, the target coverage of each template-based model was considered in ResQ. Using this feature allowed the identification and use of a known secondary structure from a database of existing structures, that matched the sequence of the protein being modelled. These intermediate features, the coverage of template based modelling and conformational variations, were combined with sequence and structural information derived from homologous proteins in ResQ to predict local residue accuracy and the B-factor profile, enhancing the accuracy and reliability of 3D protein structure prediction (Yang, Wang and Zhang, 2016). ResQ was integrated into the seventh and eight versions of ModFOLD method (Maghrabi, 2019; McGuffin *et al.*, 2019; McGuffin *et al.*, 2021). To calculate the ResQ score, each model was compared to alternative models of the same protein predicted by LOMETS (Wu and Zhang, 2007).

4.2 Aim and Objectives

In this chapter, our aim is to further improve on the local model quality assessment accuracy of ModFOLD9 in two stages. The first stage involved integrating the consensus CDA score with quality scores derived from pure-single model methods. The pure-single model methods considered in this study are SSA, ProQ2, ProQ2D, ProQ3D, ProQ4_MF9, VoroMQA, SSA, DeepAccNet, DeepAccNet_Bert and DeepAccNet_MSA. The quality scores were derived from these methods and then fed into two different MLP neural networks each trained to predict one of the two observed local model quality scores; the S-score and lDDT score. The second stage was to combine the consensus CDA score and pure-single scores with quality scores computed from quasi-model methods into NNs using a similar procedure. These quasi-single model methods included ModFOLD5_single, ModFOLD5clustQ_single, DBA and ResQ.

Again, the observed local scores (S-score and lDDT score) were the target functions for the two MLPs. The local score predictions were assessed by analysing their correlation with the observed scores and evaluating their performance via ROC analysis. To determine the improvement in local assessment accuracy for ModFOLD9, the predicted quality scores were compared with those of each of the component methods. A similar optimisation procedure was conducted (described in Chapter 3) in which the hyper-parameters of the MLP model were tuned during the training phase. The hyper-parameters included the following: the number of hidden neurons, learning rate, error rate and iterations.

4.3 Methods

4.3.1 The Consensus Algorithm for Predicting Local Model Quality

The consensus approach was applied to combine the quality scores predicted from pure-single and quasi-single model methods with the consensus CDA score. This consensus approach was conducted into two stages. The first stage involved testing the consensus of six CDA scores with nine pure-single model quality scoring methods. These methods included the SSA score (Maghrabi and McGuffin, 2017; Maghrabi and McGuffin, 2020; Maghrabi, 2019; McGuffin *et al.*, 2021), ProQ2 (Ray, Lindahl and Wallner, 2012), ProQ2D (Uziela *et al.*, 2017), ProQ3D (Uziela *et al.*, 2017), VoroMQA (Olechnovič and Venclovas, 2017), ProQ4 (Cheng *et al.*, 2019), DeepAccNet_Bert, DeepAccNet and DeepAccNet_MSA (Hiranuma *et al.*, 2021). Each method was used to predict the per-residue quality score for the interested model; these scores were then integrated along with the six CDA scores into two versions of the MLP neural network (Figure 4.1). The MLP architecture was similar to that of the MLP described in Chapter 3, in which a sliding window size of 5 was used for input, with zeros padding out the end residues. Using the residue scores from the first combination approach, 75 inputs (15×5) were generated for each residue. Two MLP versions were trained to learn each of the observed local scores: S-score or lDDT score.

The second stage combined the quality scores of the four quasi-single methods with the six CDA scores and the eight pure-single model scores for a single model. The quasi-single model methods considered in this experiment were ResQ, DBA, ModFOLD5_single and ModFOLDclustQ_single. For the consensus implementation, we applied two versions of the same MLP neural network architecture to predict S-score and lDDT score (Figure 4.2). This time the input data consisted of 19 quality scores per residue with a sliding window size of 5.

The MLP's hyperparameters were fine-tuned using a procedure similar to that presented in Chapter 3. In the first stage approach, the initial hyperparameters to predict the S-score were set as follows: 28 hidden neurons, learning rate of 0.1, error rate of 0.01 and three iterations. For the lDDT score, the default values were set to 35 hidden neurons, learning rate of 0.1, error rate of 0.01 and four iterations. After that, the number of hidden neurons was adjusted while the other hyperparameters were fixed during the training process. In the second stage approach, the default settings for MLP hyperparameters were as follows: 58 hidden neurons, learning rate of 0.1, error rate of 0.01 and three iterations to learn the S-score, whereas for learning the lDDT score, the number of hidden neurons was 48, learning rate and error rate set to 0.1 and four iterations were set (see Table 4.1). The determination of initial values for hyperparameters were based on precedents established in prior studies as well as empirical evidence, as mentioned in Chapter 3.

As part of the training process, the number of hidden neurons was modified, and all other hyperparameters remained unchanged. The best runs were achieved by training all parameter combinations up to three times and saving the NN weights. This work was completed in collaboration with Megan Hird, an undergraduate student, and some of the data shown here was also presented in her final year project. Megan conducted the analysis of fine-tuning MLP hyperparameters for predicting the lDDT score, and the results of her analysis have been presented in the Results and Discussion section.

Table 4.1. Default settings of hyperparameters for each MLP version during training to predict the S-score and lDDT from each consensus approach. The tuned hyper-parameters were the number of hidden neurons, learning rate, error rate and iterations.

Hyper-parameter	**First combination**		**Second combination**	
	S-score	lDDT	S-score	lDDT
The number of hidden neurons	28	35	58	48
Learning rate	0.1	0.1	0.1	0.1
Error rate	0.01	0.01	0.01	0.1
Iterations	3	4	3	4

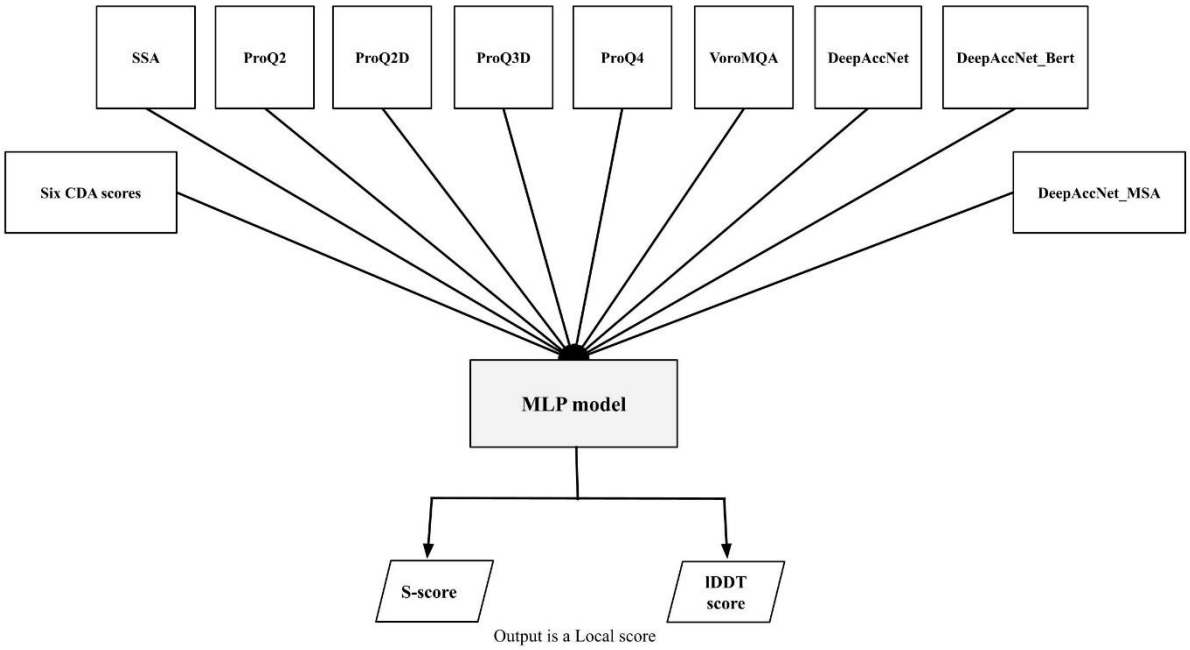


Figure 4.1. A simplified flowchart shows how the consensus algorithm was applied in the first combination stage of pure-single quality scores with CDA scores to improve the accuracy of local model quality estimates by ModFOLD9. The quality scores were computed according to pure-single model methods: SSA, ProQ2, ProQ2D, ProQ3D, ProQ4_MF9, VoroMQA, SSA, DeepAccNet, DeepAccNet_Bert, and DeepAccNet_MSA. The pure-single scores and six CDA scores were fed into an MLP to predict per-residue score; S-score or IDDT score.

.

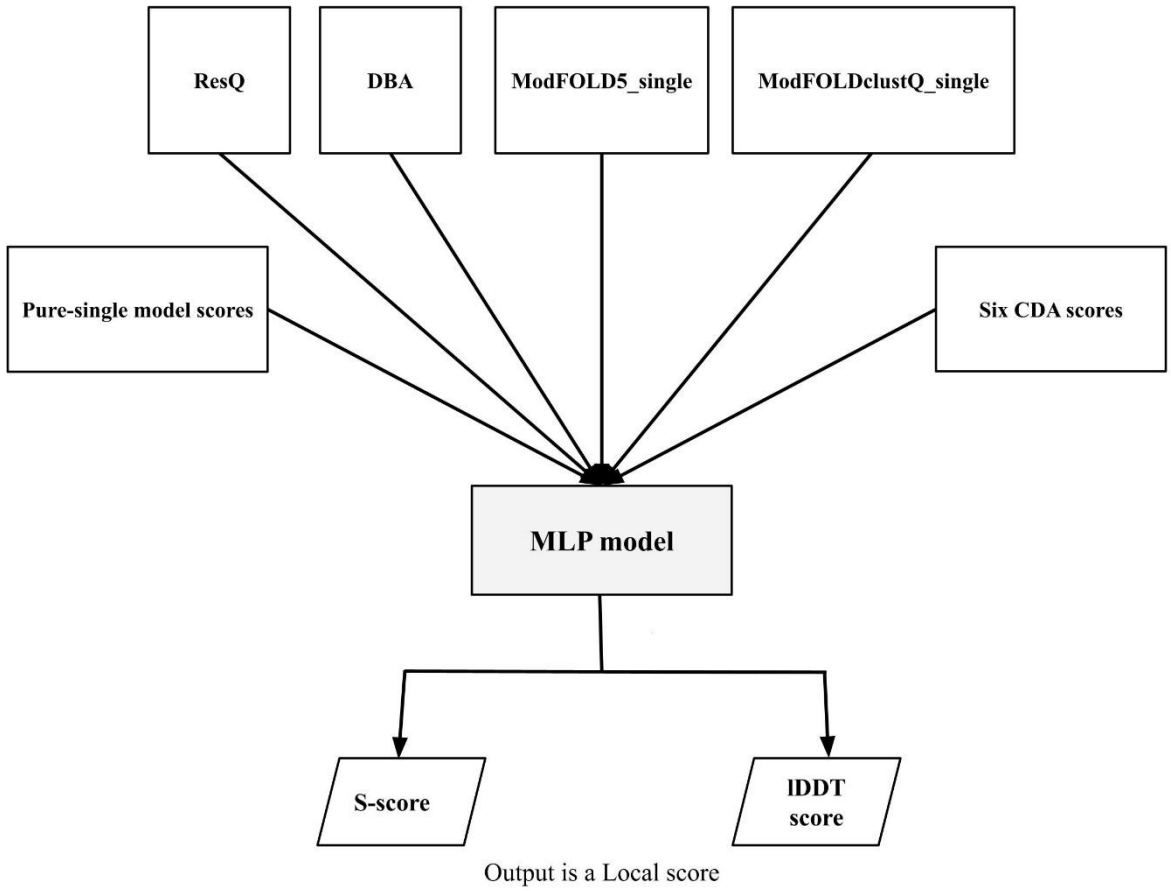


Figure 4.2. A simplified flowchart shows how the consensus algorithm was applied in the second combination stage of quasi-single quality scores with pure-single scores and CDA scores to improve the accuracy of local model quality estimates by ModFOLD9. The quality scores were computed according to quasi-single model methods: ResQ, DBA, ModFOLD5_single and ModFOLDclustQ_single. The MLP neural network was fed with quasi-single scores, pure-single scores, and six CDA scores to predict per-residue score; S-score or IDDT score.

4.3.2 Training and Testing Data and Evaluation Measurements

The CASP14 data set for protein structure prediction was used as training and testing data for MLP neural network following the cross-validation procedure similar to that outlined in Chapter 3. ModFOLD9's performance was evaluated with similar measurements: Pearson's R correlation coefficient, Spearman (Rho) correlation coefficient and ROC analysis. These assessment methods assessed the correlation between the predicted and the observed quality scores. The local quality scores, S-score and lDDT score, were target functions. Lastly, the performance of ModFOLD9 was compared against the individual pure-single model methods and quasi-single model methods according to the local quality scores.

4.4. Results and Discussion

Analysis of the evaluation scores allows us to determine the performance of ModFOLD9 in local model quality assessments. The hyperparameters of the MLP were tuned to achieve the optimal performance. We trained the MLP multiple times, each time adjusting its hyperparameters and the evaluation metrics were analysed until optimal performance was reached. Subsequently, we compared the predicted quality scores of ModFOLD9 with those of the established methods according to the relationship to the observed S-scores and lDDT scores.

In the first section we analyse the behaviour of the predicted quality scores during the process of optimising the MLP hyperparameters. The results are provided in two subsections. The first subsection is on training MLP neural networks to learn the S-score and lDDT from the integration of CDA scores and pure-single model scores, while the second subsection is on training MLP models to learn the S-score and lDDT from the integration of CDA scores and pure-single model scores with the quasi-single model scores. The last section demonstrates

how the two consensus methods improved the ModFOLD9 local assessment performance based on the S-score and lDDT score.

4.4.1. Parameterisation of The NN model

4.4.1.1. The Consensus of CDA Scores with Pure-Single Model Methods

In a similar manner to the MLP optimisation in Chapter 3, our initial focus was on determining the optimal number of hidden neurons, while maintaining fixed values for all other hyperparameters. For predicting the S-score, we started with 28 hidden neurons and kept the learning rate at 0.1, the error rate at 0.01, and the iteration number at 3 as fixed hyperparameters. Figure 4.3A shows the varying correlation scores for predicting the S-score based on the number of neurons in the hidden layer. The analysis reveals that increasing the number of neurons to 38 resulted in the highest correlation scores (Pearson's R =0.625 and Spearman's Rho= 0.628), which improved the MLP's ability to accurately predict the S-scores. We observed that exceeding 46 neurons negatively affected the performance of the MLP, which indicates that the best performance was achieved at 38 neurons in the hidden layer. This finding was supported by the ROC AUC score, which also peaked (0.825) at 38 neurons (Figure 4.3B). Additionally, we observed that increasing the number of hidden neurons to 38 (as shown in Figure 4.3C) resulted in higher ROC AUC scores of up to 0.032 at FPR<=0.1. Based on all the evaluation metrics, we identified 38 as the optimal number of hidden neurons to achieve high MLP performance.

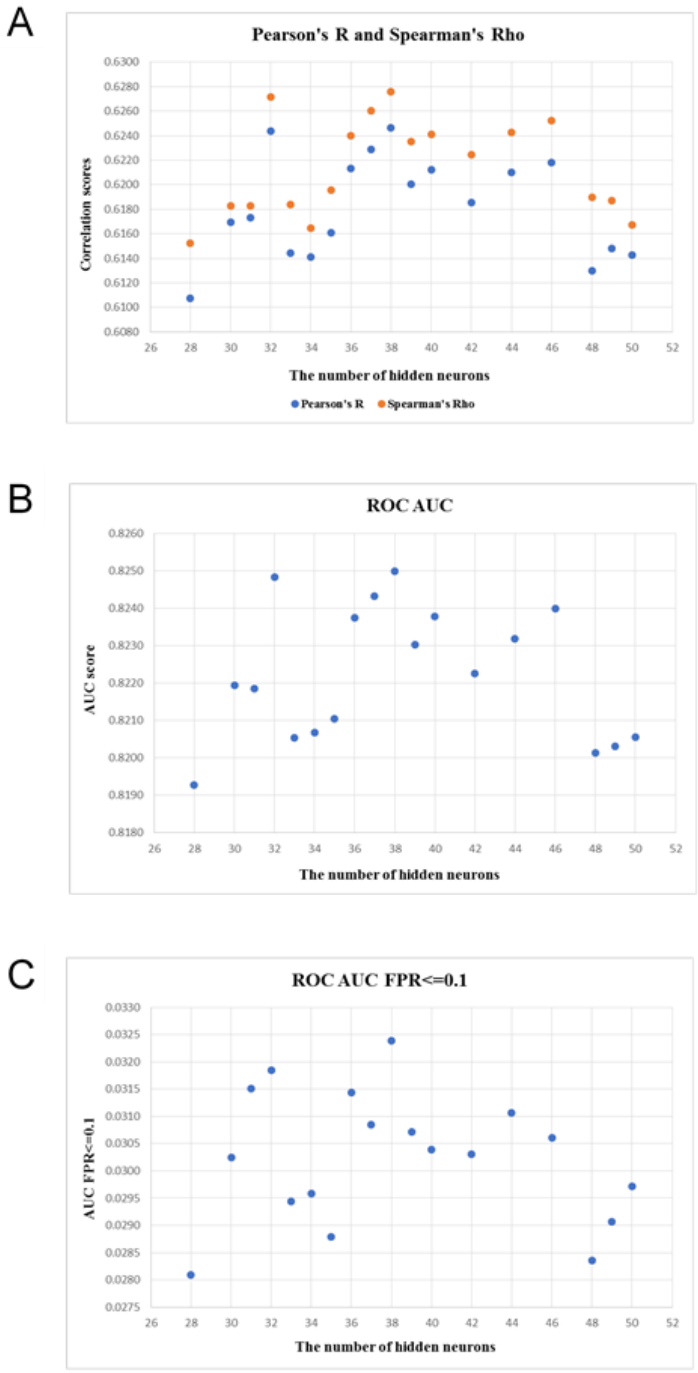


Figure 4.3. The effect of tuning the number of hidden neurons on the MLP's performance according to the S-score with the consensus of CDA scores and pure-single model scores. (A) Pearson's R and Spearman's Rho correlation scores versus the number of hidden neurons. (B) ROC AUC scores versus the number of hidden neurons. (C) ROC AUC FPR<=0.1 scores versus the number of hidden neurons.

The second hyperparameter was the learning rate. We tested different learning rates to identify the best one. We started with a rate of 0.1 before trying higher and lower values to assess the effect on the MLP's performance. Based on the findings presented in Figure 4.4A, it appears that a learning rate of 0.001 yielded the highest correlation scores. The same trend was observed for ROC AUC scores and ROC AUC FPR<=0.1 scores, as illustrated in Figures 4.4B and 4.4C. However, we observed a decrease in MLP performance when we attempted to modify the third parameter, the error rate, as indicated in Table 4.2. To address this issue, we opted to retrain using the second-best value of the learning rate (0.1) and the same range of error rate values. The findings suggest that the best results were achieved when implementing a 0.01 learning rate and a 0.1 error rate. Once the optimal neuron number, learning rate, and error rate were respectively set to 38, 0.01, and 0.1 for the S-score, the iteration values were adjusted. Table 4.3 presents the evaluation scores for the consensus of the CDA with pure-single scores based on the S-score for the three values of iterations. The evaluation scores reflect that the MLP learned better with three iterations. In total, 38 hidden neurons, a learning rate of 0.01, an error rate of 0.1, and three iterations were selected as the optimal hyperparameters for predicting the S-score.

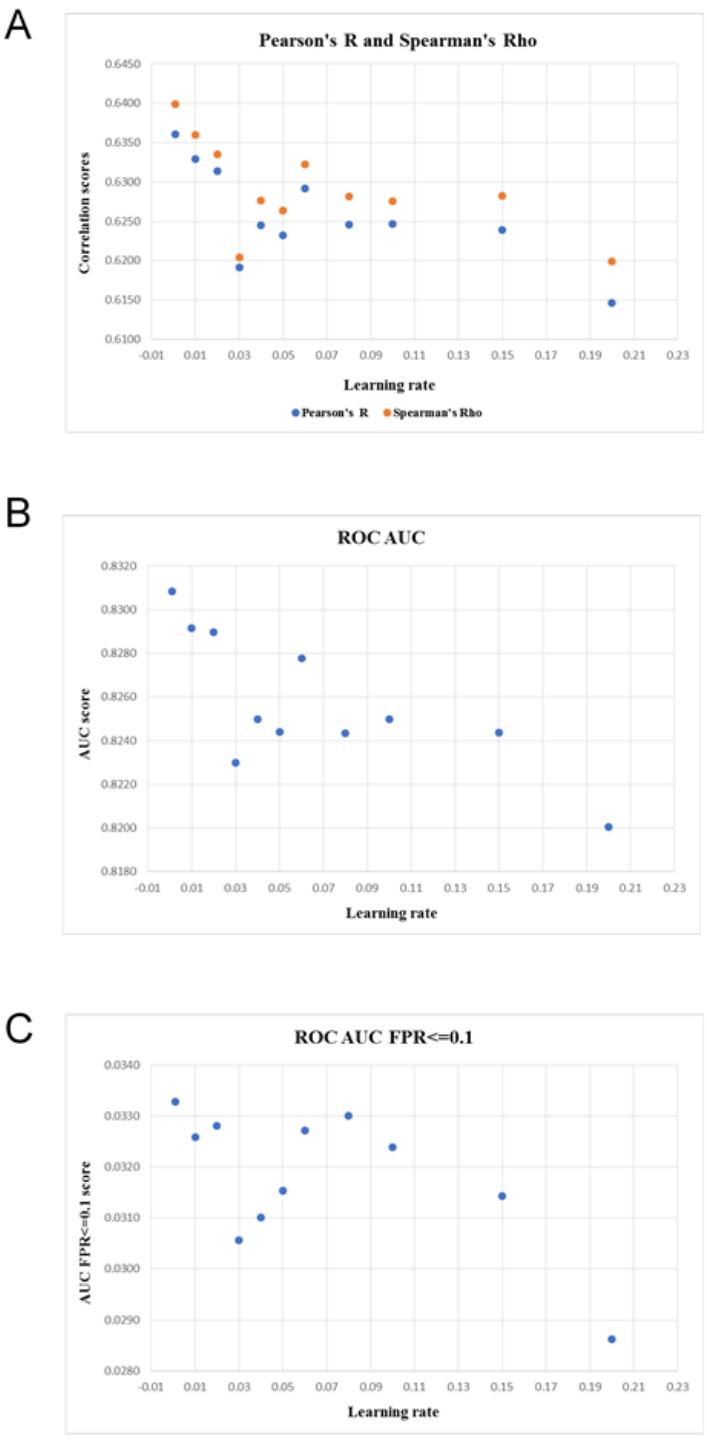


Figure 4.4. The effect of tuning the learning rate on the MLP's performance according to S-score with the consensus of CDA scores and pure-single model scores. (A) Pearson's R and Spearman's Rho correlation scores versus the learning rate. (B) ROC AUC scores versus the learning rate. (C) ROC AUC FPR<=0.1 scores versus the learning rate.

Table 4.2. The effect of tuning the error rate with the two highest learning rate values on the MLP's performance according to the S-score with the consensus of CDA scores and pure-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The hidden neuron numbers and iteration were 38 and 3, respectively. The error values were adjusted with two learning rates (0.01 and 0.001), and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

The number of hidden neurons	Learning Rate	Error rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
38	0.001	0.01	3	0.6361	0.6399	0.8308	0.0333
38	0.001	0.02	3	0.6364	0.6400	0.8310	0.0335
38	0.001	0.05	3	0.6365	0.6399	0.8312	0.0337
38	0.001	0.1	3	0.6368	0.6402	0.8313	0.0338
38	0.001	0.07	3	0.6357	0.6392	0.8308	0.0337
38	0.001	0.09	3	0.6362	0.6400	0.8309	0.0337
38	0.001	0.08	3	0.6362	0.6399	0.8310	0.0337
38	0.01	0.01	3	0.6329	0.6359	0.8291	0.0326
38	0.01	0.02	3	0.6288	0.6308	0.8274	0.0319
38	0.01	0.05	3	0.6345	0.6369	0.8303	0.0333
38	0.01	0.1	3	**0.6390**	**0.6420**	**0.8320**	**0.0346**
38	0.01	0.07	3	0.6366	0.6390	0.8311	0.0339
38	0.01	0.09	3	0.6286	0.6308	0.8275	0.0323
38	0.01	0.08	3	0.6370	0.6397	0.8312	0.0341

Table 4.3. The effect of tuning iterations on the MLP's performance according to the S-score with the consensus of CDA scores and pure-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The hidden neuron numbers, learning rate and error rate were 38, 0.01, and 0.1, respectively. The iteration values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

number of hidden neurons	Learning Rate	Error rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
38	0.01	0.1	3	**0.6390**	**0.6420**	**0.8320**	**0.0346**
38	0.01	0.1	2	0.6376	0.6413	0.8312	0.0335
38	0.01	0.1	4	0.6244	0.6278	0.8257	0.0310

The MLP optimisation for predicting the lDDT score also involved varying hyperparameters but with different values. We first conducted the adjustment for the number of hidden neurons for optimal performance. The implementation started with 35 hidden neurons. This number was later increased and decreased while maintaining a 0.1 learning rate, a 0.01 error rate, and 4 iterations as the remaining hyperparameters. Figure 4.5A depicts the changes in the MLP's lDDT score performance changed with varying numbers of hidden neurons. The highest correlation scores were obtained when the number of hidden neurons ranged from 44 to 47, which signified the best performance. According to the ROC AUC scores in Figure 4.5B, the optimal range of hidden neurons was between 44 and 47, as the MLP achieved the best AUC score with these values. At FPR <=0.1, the ROC AUC score was highest with 38 hidden neurons, as shown in Figure 4.5C. To determine the ideal number of hidden neurons, we retrained the MLP by changing the learning rate to 0.05 for the top four hidden neuron numbers, as seen in Table 4.4. After comparing the evaluation scores with those of the previous training process, which had a learning rate of 0.1, we found that the highest scores were achieved when the number of hidden neurons was 45 with a learning rate of 0.05. Thus, we concluded that 45 was an optimal number of neurons. After making this determination, we modified the learning rate to test different values and achieve the best evaluation scores. However, as Table 4.5 illustrates, a learning rate of 0.05 remained the optimal value based on the evaluation scores of the lDDT score.

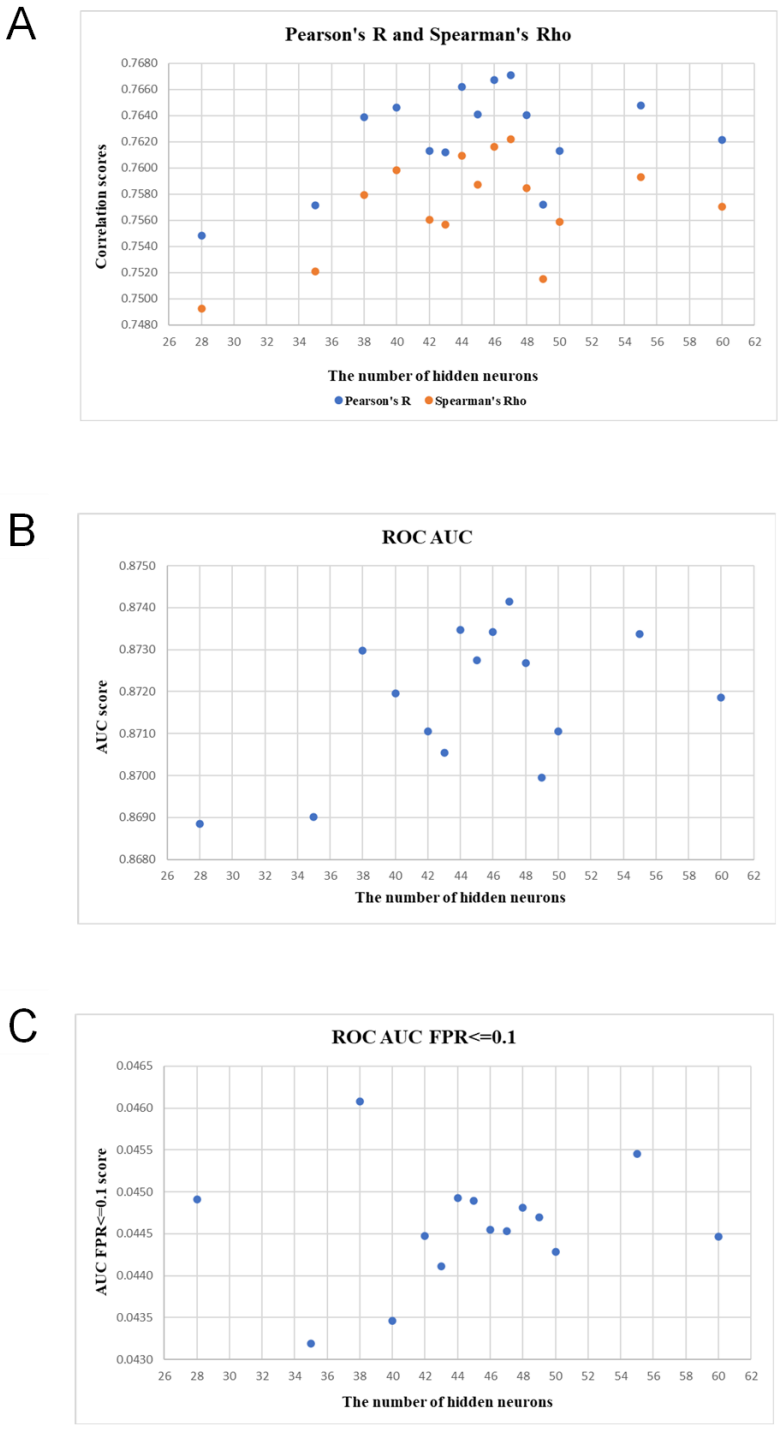


Figure 4.5. The effect of tuning the number of hidden neurons on the MLP's performance according to the lDDT score with the consensus of CDA scores and pure-single model scores. (A) Pearson's R and Spearman's Rho correlation scores versus the number of hidden neurons. (B) ROC AUC scores versus the number of hidden neurons. (C) ROC AUC FPR<=0.1 scores versus the number of hidden neurons.

Table 4.4. The effect of tuning the two learning rate values with optimal numbers of neurons on the MLP's performance according to the lDDT score with the consensus of CDA scores and pure-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The error rate and iteration were 0.1 and 4, respectively. The number of hidden neurons was adjusted with a learning rate of 0.05, and the evaluation scores were measured individually. The findings from this process were compared to the previous results of the numbers of hidden neurons with a learning rate of 0.1 to determine the best neuron number. The bolded scores denote the highest evaluation scores.

The number of hidden Neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
44	0.05	0.1	4	0.7689	0.7640	0.8747	0.0452
45	0.05	0.1	4	**0.7694**	**0.7694**	**0.8748**	**0.0449**
46	0.05	0.1	4	0.7663	0.7609	0.8740	0.0458
47	0.05	0.1	4	0.7683	0.7628	0.8747	0.0456
44	0.1	0.1	4	0.7662	0.7609	0.8735	0.0449
45	0.1	0.1	4	0.7641	0.7587	0.8727	0.0449
46	0.1	0.1	4	0.7667	0.7616	0.8734	0.0445
47	0.1	0.1	4	0.7696	0.7640	0.8749	0.0454

Table 4.5. The effect of tuning the learning rate on the MLP's performance according to the lDDT score with the consensus of CDA scores and pure-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The hidden neuron numbers, error rate and iterations were 45, 0.1, and 4, respectively. The learning rate values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

The number of hidden Neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
45	0.04	0.1	4	0.7618	0.7561	0.8723	0.0460
45	0.06	0.1	4	0.7662	0.7606	0.8736	0.0457
45	0.15	0.1	4	0.7667	0.7623	0.8742	0.0443
45	0.1	0.1	4	0.7641	0.7587	0.8727	0.0449
45	0.05	0.1	4	**0.7694**	**0.7694**	**0.8748**	**0.0449**

The optimal values for neuron number and learning rate were set to 45 and 0.05, respectively for the lDDT score. The adjustment of error and iteration values was carried out individually. Tables 4.6 and 4.7 present the evaluation scores for the lDDT score based on error rate values and iterations. Based on the evaluation scores, the data indicate that the MLP performed better with a 0.07 error rate and three iterations. However, according to the ROC AUC FPR<=0.1 result, a slightly higher score was achieved with four iterations. This finding suggests that with these settings, the MLP only needed to be shown the dataset three to four times to achieve optimal accuracy and any further iterations may result in overfitting. Based on this analysis, we determined that the optimal hyperparameters for the best MLP prediction performance according to the lDDT score were as follows: 45 hidden neurons, a learning rate of 0.05, an error rate of 0.07, and 4 iterations.

Table 4.6. The effect of tuning the error rate on the MLP's performance according to the lDDT score with the consensus of CDA scores and pure-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The hidden neuron numbers, learning rate and iteration were 45, 0.05 and 4, respectively. The error rate values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

The number of hidden neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
45	0.05	0.04	4	0.7648	0.7595	0.8736	0.0454
45	0.05	0.05	4	0.7656	0.7595	0.8726	0.0454
45	0.05	0.06	4	0.7709	0.7651	0.8751	0.0455
45	0.05	0.07	4	**0.7712**	**0.7658**	**0.8758**	**0.0461**
45	0.05	0.08	4	0.7674	0.7615	0.8734	0.0454
45	0.05	0.15	4	0.7609	0.7563	0.8719	0.0446
45	0.05	0.2	4	0.7549	0.7509	0.8701	0.0450

Table 4.7. The effect of tuning the iteration value on the MLP's performance according to the lDDT score with the consensus of CDA scores and pure-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The hidden neuron number, learning rate and error rate were 45, 0.05, and 0.07, respectively. The iteration values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

The number of hidden neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
45	0.05	0.07	7	0.7639	0.7577	0.8721	0.0455
45	0.05	0.07	2	0.7644	0.7591	0.8735	0.0460
45	0.05	0.07	5	0.7677	0.7622	0.8744	0.0455
45	0.05	0.07	6	0.7712	0.7654	0.8752	0.0454
45	0.05	0.07	4	0.7712	0.7658	0.8758	**0.0461**
45	0.05	0.07	3	**0.7716**	**0.7669**	**0.8760**	0.0453

4.7.1.2. The Consensus of CDA Scores, Pure- and Quasi-Single Model Methods

Initially, we began with 58 hidden neurons, with fixed hyperparameters: a learning rate of 0.1, an error rate 0.01, and 3 iterations. Figure 4.6A displays the fluctuation in correlation scores with the changes in the number of hidden neurons. The correlation scores were the highest when 44 neurons were applied in the hidden layer (Pearson's R = 0.699, Spearman's Rho = 0.706). The ROC AUC score also peaked (0.862) when the number of neurons was set to 44 (Figure 4.6B). In contrast, when we reduced the number of hidden neurons to 38, the ROC AUC FPR<=0.1 reached 0.043 (Figure 4.6C). While this score decreased marginally to 0.042 at 44 hidden neurons, there were observed improvements in other scores. Overall, the results suggest that 44 is an optimal number of neurons to maximise the performance of the MLP.

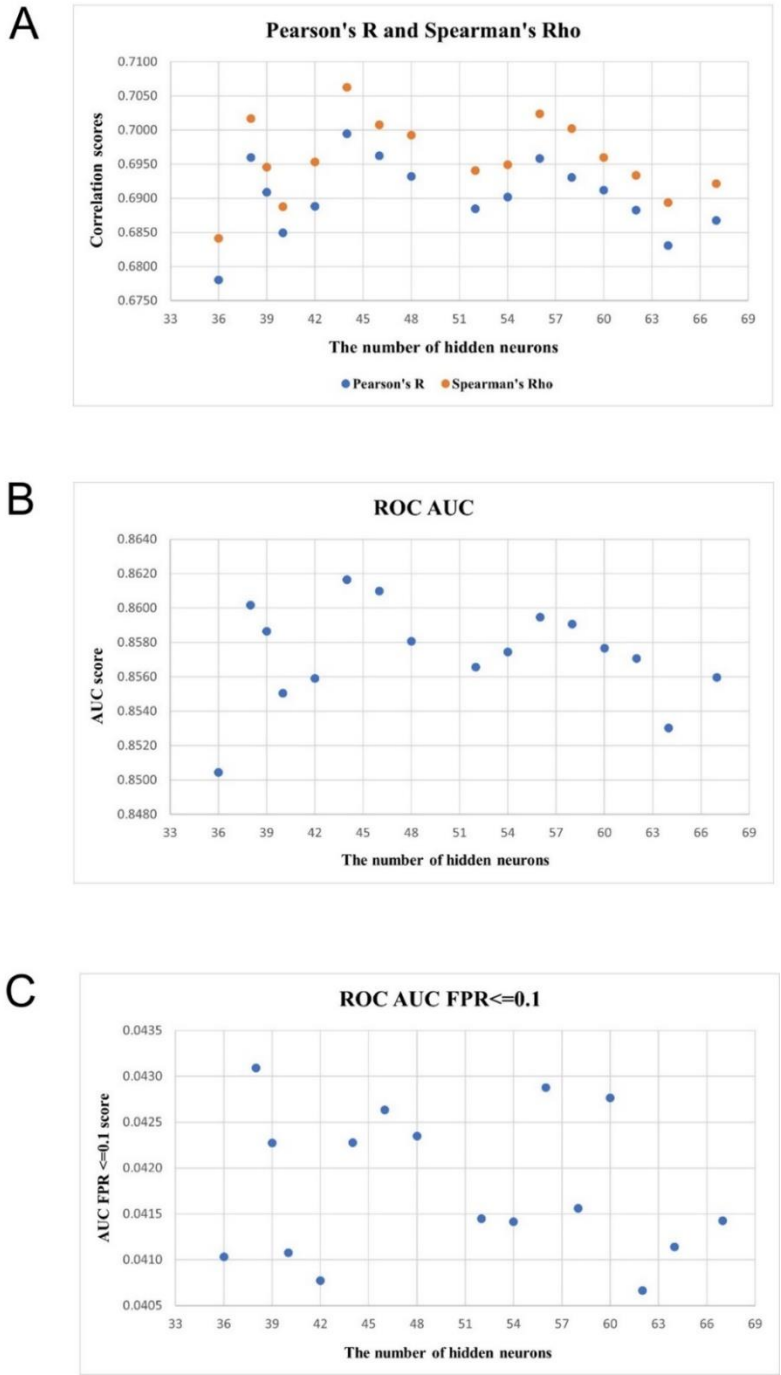


Figure 4.6. The effect of tuning the number of hidden neurons on the MLP's performance according to the S-score with the consensus of CDA scores, pure and quasi-single model scores. (A) Pearson's R and Spearman's Rho correlation scores versus the number of hidden neurons. (B) ROC AUC scores versus the number of hidden neurons. (C) ROC AUC FPR<=0.1 scores versus the number of hidden neurons.

For the learning rate adjustment, we started with a value of 0.1, which we subsequently increased and decreased. From an analysis of Table 4.8, we established that a learning rate of 0.003 yielded the highest scores for both correlation matrices and the ROC AUC analysis. In contrast, a learning rate of 0.001 produced a slightly higher ROC AUC FPR<=0.1 score than that produced by 0.003. The difference between the two scores at these two learning rate values was 0.0001, which suggests that when lowering the learning rate from 0.003 to 0.001 did not considerably boost MLP performance. Therefore, a learning rate of 0.003 appeared to be optimal as the evaluation scores improved with this rate.

After selecting 44 and 0.003 as the optimal hyperparameters of neuron number and learning rate, respectively, we adjusted the error rate, while setting the number of iterations to 3. The evaluation scores for S-score based on error rate values were illustrated in Figure 4.7. The results shows that the error rate of 0.04 resulted in the best performance for accurately predicting the S-score according to all evaluation metrics. However, as seen in Table 4.9, changes to iteration values produced higher evaluation scores at 2 and 3 iterations when adjusting iteration values. During the previous server upgrade process, we observed that using three iterations in the MLP's training process resulted in the best overall performance. Based on these findings, we determined that 44 hidden neurons, a learning rate of 0.003, an error rate of 0.04, and 3 iterations were optimal hyperparameters for the MLP's to predict the S-score.

Table 4.8. The effect of tuning the learning rate value on MLP's performance according to the S-score with the consensus of CDA scores, pure and quasi-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores, and ROC AUC FPR<=0.1 scores. The hidden neuron numbers, error rate and iteration were 44, 0.01, and 3, respectively. The learning rate values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

The number of hidden neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
44	0.001	0.01	3	0.7062	0.7134	0.8656	**0.0448**
44	0.003	0.01	3	**0.7068**	**0.7136**	**0.8658**	0.0447
44	0.005	0.01	3	0.7048	0.7119	0.8649	0.0437
44	0.008	0.01	3	0.7019	0.7072	0.8639	0.0432
44	0.01	0.01	3	0.6993	0.7039	0.8629	0.0437
44	0.02	0.01	3	0.6976	0.7018	0.8621	0.0436
44	0.06	0.01	3	0.6835	0.6892	0.8542	0.0404
44	0.1	0.01	3	0.6995	0.7062	0.8616	0.0423
44	0.15	0.01	3	0.6892	0.6962	0.8568	0.0424
44	0.2	0.01	3	0.6873	0.6932	0.8551	0.0412

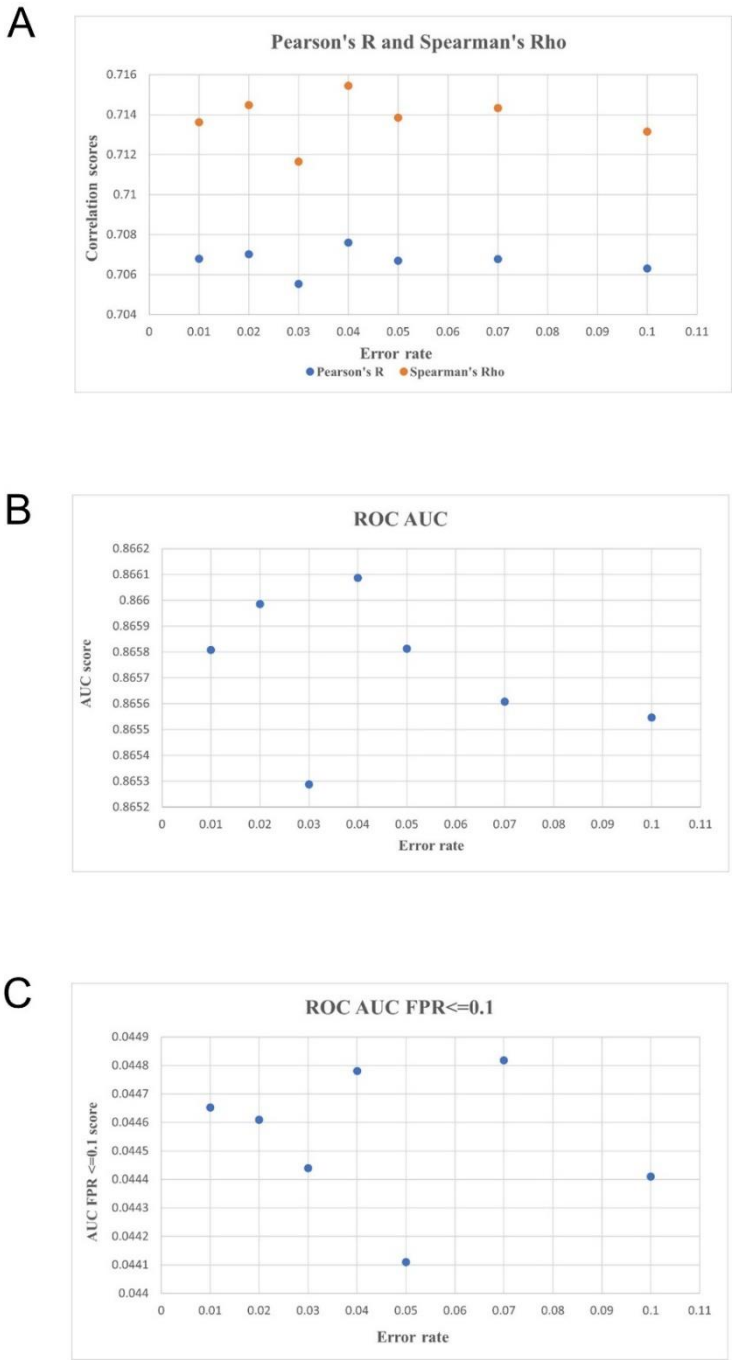


Figure 4.7. The effect of tuning the error rate on the MLP's performance according to the S-score with the consensus of CDA scores, pure and quasi-single model scores. (A) Pearson's R and Spearman's Rho correlation scores versus the error rate. (B) ROC AUC scores versus the error rate. (C) ROC AUC FPR<=0.1 scores versus the error rate.

Table 4.9. The effect of tuning the iteration value on the MLP's performance according to the S-score with the consensus of CDA scores, pure and quasi-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The hidden neuron numbers, learning rate and error rate were 44, 0.003, and 0.04, respectively. Iteration values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

The number of hidden neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
44	0.003	0.04	2	**0.7079**	0.7150	**0.8664**	0.0447
44	0.003	0.04	3	0.7076	**0.7154**	0.8661	**0.0448**
44	0.003	0.04	4	0.7054	0.7116	0.8652	0.0441
44	0.003	0.04	6	0.7032	0.7089	0.8643	0.0435

Through various evaluation metrics, we also optimised the MLP for predicting the lDDT score. We first adjusted the number of hidden neurons for optimal performance. We started with 48 hidden neurons while maintaining a 0.1 learning rate and error rate and 4 iterations as the remaining hyperparameters. Figure 4.8A and Figure 4.8B depict the changes in the MLP's lDDT score performance with varying numbers of hidden neurons according to the correlation scores and ROC AUC score. The optimal number of hidden neurons was 40, which produced the highest score and achieved the best performance of MLP. At FPR <=0.1, the ROC AUC score was highest with 35 hidden neurons, as shown in Figure 4.8C. To determine the optimal number of hidden neurons, we retrained the MLP by changing the learning rate values for the best two hidden neuron numbers, as seen in Table 4.10. We found that the two highest evaluation scores were produced with the learning rate values of 0.1 and 0.11 with 35 and 40 neurons. We determined that the best values for these two parameters were 35 hidden neurons and a learning rate of 0.1 while modifying the error rate to 0.05 (see Table 4.11). Table 4.12 provides the results from adjusting the iteration values, which reveal that four iterations were optimal for achieving the best MLP performance. Based on this analysis, the optimal hyper-parameters for the MLP to predict lDDT score were 35 hidden neurons, a learning rate of 0.1, an error rate of 0.05, and 4 iterations.

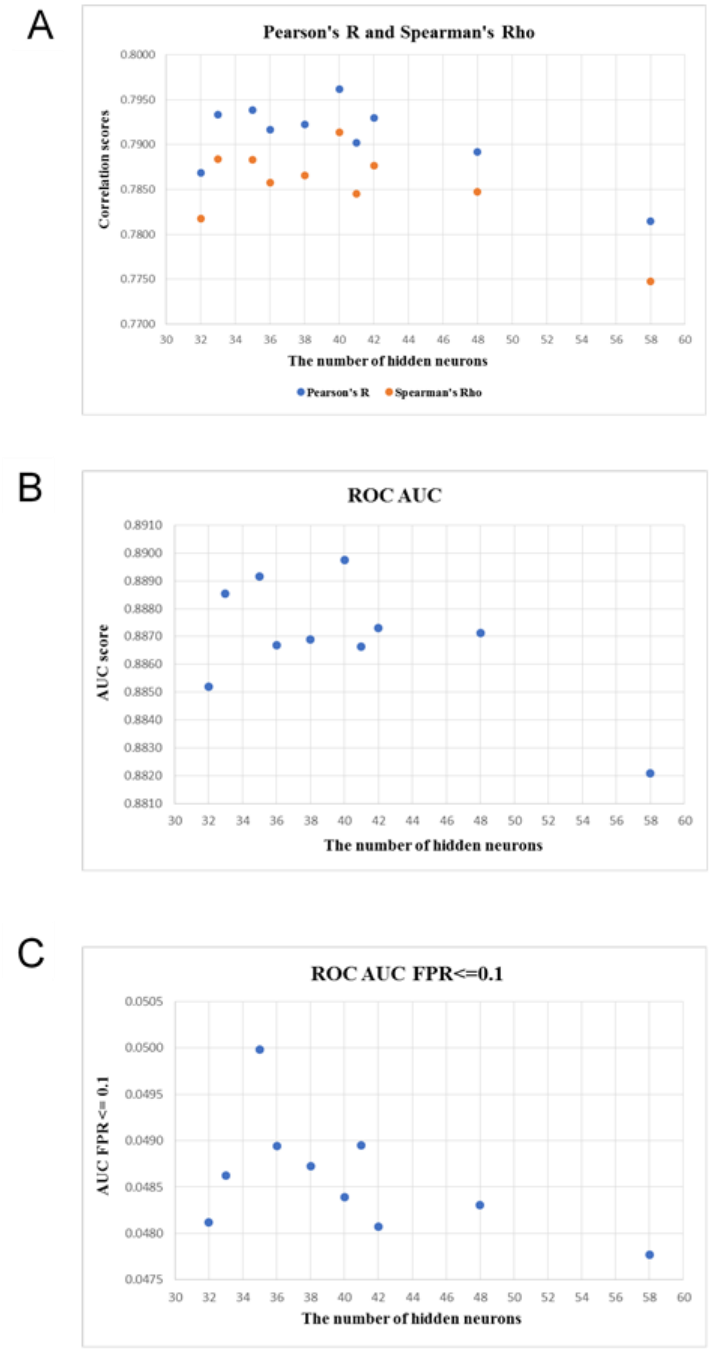


Figure 4.8. The effect of tuning the number of hidden neurons on the MLP's performance according to the lDDT score with the consensus of CDA scores, pure and quasi-single model scores. (A) Pearson's R and Spearman's Rho correlation scores versus the number of hidden neurons. (B) ROC AUC scores versus the number of hidden neurons. (C) ROC AUC FPR<=0.1 scores versus the number of hidden neurons.

Table 4.10. The effect of tuning the learning rate with the two best numbers of hidden neurons on the MLP's performance according to the lDDT score with the consensus of CDA scores, pure and quasi-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores, and ROC AUC FPR<=0.1 scores. The hidden neuron numbers were 35 and 40, and error rate and iterations were 0.1 and 4, respectively. The learning rate values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

The number of hidden neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
35	0.1	0.1	4	0.7939	0.7883	0.8892	**0.0500**
35	0.05	0.1	4	0.7870	0.7818	0.8855	0.0487
35	0.09	0.1	4	0.7840	0.7776	0.8840	0.0493
35	0.11	0.1	4	0.7955	0.7899	0.8887	0.0490
35	0.12	0.1	4	0.7850	0.7781	0.8838	0.0490
35	0.13	0.1	4	0.7882	0.7832	0.8866	0.0490
35	0.14	0.1	4	0.7776	0.7707	0.8805	0.0481
35	0.15	0.1	4	0.7917	0.7854	0.8870	0.0490
40	0.1	0.1	4	**0.7962**	**0.7914**	**0.8897**	0.0484
40	0.05	0.1	4	0.7862	0.7809	0.8843	0.0469
40	0.09	0.1	4	0.7862	0.7800	0.8845	0.0490
40	0.11	0.1	4	0.7851	0.7804	0.8839	0.0469
40	0.13	0.1	4	0.7894	0.7845	0.8866	0.0483
40	0.14	0.1	4	0.7905	0.7864	0.8875	0.0480
40	0.15	0.1	4	0.7912	0.7848	0.8873	0.0495
40	0.16	0.1	4	0.7710	0.7644	0.8775	0.0476

Table 4.11. The effect of tuning the error rate value with the two hidden neuron numbers and two learning rate values on the MLP's performance according to the lDDT score with the consensus of CDA scores, pure and quasi-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The hidden neuron numbers were 35 and 40, learning rate values were 0.1 and 0.11, and the number of iterations was 4. The error rate values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

The number of hidden Neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
35	0.1	0.2	4	0.7746	0.7718	0.8796	0.0448
35	0.1	0.15	4	0.7883	0.7834	0.8849	0.0478
35	0.1	0.01	4	0.7897	0.7842	0.8857	0.0483
35	0.1	0.05	4	**0.7968**	**0.7924**	**0.8908**	**0.0494**
35	0.11	0.05	4	0.7877	0.7824	0.8855	0.0482
35	0.11	0.01	4	0.7922	0.7863	0.8867	0.0484
35	0.11	0.15	4	0.7890	0.7843	0.8871	0.0481
40	0.1	0.2	4	0.7895	0.7839	0.8864	0.0488
40	0.1	0.15	4	0.7894	0.7853	0.8866	0.0478
40	0.1	0.05	4	0.7898	0.7845	0.8872	0.0490

Table 4.12. The effect of tuning the iteration value scores on the MLP's performance according to the lDDT score with the consensus of CDA scores, pure and quasi-single model scores. The evaluation measures were Pearson's R and Spearman's Rho correlation analyses, ROC AUC scores and ROC AUC FPR<=0.1 scores. The hidden neuron numbers, learning rate and error rate were 35, 0.1 and 0.05, respectively. The iteration values were adjusted, and their evaluation scores were measured individually. The bolded scores denote the highest evaluation scores.

Th number of hidden neurons	Learning Rate	Error Rate	Iterations	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR<=0.1
35	0.1	0.05	2	0.7827	0.7771	0.8828	0.0480
35	0.1	0.05	3	0.7902	0.7848	0.8867	0.0485
35	0.1	0.05	5	0.7949	0.7894	0.8875	0.0484
35	0.1	0.05	4	**0.7968**	**0.7924**	**0.8908**	**0.0494**

4.4.1.3 The Impact of Tuning MLP Hyperparameters on The Performance of ModFOLD9

Fine-tuning the hyperparameters of the MLP has improved the performance of ModFOLD9. Optimising the performance of the NN requires adjusting the hyperparameters and conducting a thorough validation process (Probst, Bischl and Boulesteix, 2018; Vabalas *et al.*, 2019). Therefore, finding the optimal set of hyperparameters is crucial to achieving the best possible performance of the MLP. This study has focused on tuning four hyperparameters during the MLP training phase, which has resulted in an improvement of the ModFOLD9 predictive assessment in both combination stages.

The number of hidden neurons in the MLP can impact its ability to predict quality scores accurately. Our study found that the optimal number of hidden neurons varied in both consensus stages. For instance, the MLP with 38 neurons was better at predicting the S-score when the input was the consensus of CDA and pure-single model scores, while the MLP with 44 neurons performed well in predicting the S-score when the input was the consensus of CDA scores and pure-single model scores with quasi-single model scores. This indicates that these hidden neuron values can enhance the MLP's ability to capture more aspects of the underlying data patterns, leading to better S-score predictions. However, more hidden neurons do not always result in better performance. Having too many neurons can result in overfitting, where the MLP memorises the training data and performs poorly on unseen data (Awad and Khanna, 2015; Chasiotis, Nadi and Filios, 2021; Zhao *et al.*, 2023). Our study found that an MLP with 46 neurons performed poorly predicting S-score on the consensus of CDA and pure-single model scores.

The learning rate plays a crucial role in adjusting the network weights during the learning process. By modifying the weights appropriately, the MLP neural network can converge to a more optimal solution that fits the data better (Zubair *et al.*, 2014; Awad and Khanna, 2015;

Mukhtorov *et al.*, 2023). Our study has established that different learning rates have a significant impact on MLP's performance. When predicting quality scores, the choice of learning rate varies depending on the combination of input quality scores. For instance, for the combination of CDA scores and pure-single scores, a learning rate of 0.01 was found to improve the performance of MLP in predicting the S-score, while a learning rate of 0.05 was optimal for predicting the lDDT score. However, for the combination of CDA scores, pure-single scores, and quasi-single scores, a learning rate of 0.003 achieved the highest evaluation scores for predicting the S-score. In contrast, a learning rate of 0.1 was the best value for accurately predicting the IDDT score. These findings suggest that choosing the optimal learning rate enables MLPs to learn more effectively from the data. Therefore, different combinations of input quality scores may require different learning rates. By choosing the right learning rate, MLP can learn more effectively and converge to a better prediction.

The error rate refers to the disparity between the predicted output of the MLP and the observed output (Zubair *et al.*, 2014; Awad and Khanna, 2015; Elansari, Ouanan and Bourray, 2023). Therefore, it is crucial to minimise error rates during the training phase to optimise MLP performance. In our experiment, we reduced the error rate for MLP from 0.1 to 0.07 while predicting IDDT from the consensus of CDA scores and pure-single model scores, resulting in an improved MLP performance according to evaluation results. However, it is essential to note that in some cases, a low error rate can lead to poor performance. This was observed in the second combination to predict the S-score, where the MLP training started with an error rate of 0.01, resulting in poor performance. When we increased the error rate from 0.01 to 0.04, MLP performed well based on the evaluation results. This suggests adjusting error rates can lead to higher predictive accuracy and improved performance.

The performance of MLP is heavily influenced by the number of iterations it undergoes during training. Each iteration involves a complete pass through the training set, during which the model adjusts its weights to enhance its learning capability. An MLP cycles through an entire dataset based on the number of iterations. Using too few iterations might result in the MLP underfitting the data because it cannot learn enough from it. Alternatively, if the number of iterations is too high, the MLP may begin memorising the training data, resulting in overfitting (Bengio, 2012). In the case of ModFOLD9 MLP, the optimal iteration range was between 3 and 4. Fewer or more iterations than this range resulted in the MLP learning noise instead of underlying patterns, resulting in less accurate predictions of the quality scores. Therefore, adjusting the iteration range contributed to further improvement in the predictive learning of MLP during the experiments.

This study highlighted the significant impact of fine-tuning the hyperparameters of MLP for practical training and avoiding overfitting. Adjusting hyperparameters, which were learning rate, hidden neurons, error rate, and iterations, could help the MLP learn complex patterns without overfitting. Furthermore, cross-validation validated the MLP's performance on unseen data, ensuring the reliability of ModFOLD9.

4.4.2 Evaluating ModFOLD9 Performance

The new combined approaches have achieved a substantial improvement in terms of local assessment accuracy, boosting ModFOLD9 performance. Our findings were presented based on evaluation according to S-score and lDDT score for the two variants of the method. The integration of CDA scores with pure-single scores was referred to ModFOLD9_pure. ModFOLD9_pure has been assessed by comparing its correlation scores with those of the pure-single model and CDA methods. The second consensus, ModFOLD9_quasi, has been assessed by comparing its correlation scores with those of every component method. The five top-performing established methods were also compared with ModFOLD9 variants using ROC analysis. The evaluation process was aimed at assessing whether the consensus methods performed better when quality scores from different scoring methods were similar.

4.4.2.1 Evaluating The Performance of ModFOLD9_pure

The performance of ModFOLD9_pure was enhanced as a result of utilising the consensus algorithm, which integrated diverse scores to achieve the best results, effectively improving the predictive power. As is evident from Figure 4.9, the S-score of ModFOLD9_pure obtained the highest Pearson's R correlation score (0.639) with the observed S-score when compared to the Pearson's R correlation scores of the S-scores predicted from CDA and pure-single model methods individually. A similar trend was observed in Spearman's Rho correlation analysis (0.642 for ModFOLD9_pure), indicating that ModFOLD9_pure outperformed the established methods. From Figure 4.10A, the ROC AUC score of ModFOLD9_pure (0.832) was the highest score, indicating that improvement gains were achieved with the consensus of quality scores. Furthermore, ModFOLD9_pure outperformed the five top-performing pure-single model methods based on AUC scores of ROC FPR <= 0.1 (Figure 4.10B). Thus, the combination of CDA scores with pure-single model scores improved the local assessment accuracy of ModFOLD9_pure according to the S-score.

The enhancement of ModFOLD9_pure's performance was also observed based on the prediction of the lDDT score. As shown in Figure 4.11, ModFOLD9_pure output scores achieved the highest correlation (Pearson's R = 0.771, Spearman's Rho = 0.766) with the observed lDDT score in comparison to all component methods. A similar pattern was observed for the ROC AUC analysis, where the ROC AUC score of ModFOLD9_pure (0.876) was the highest. In addition, ModFOLD9_pure achieved a comparable ROC AUC FPR <0.1 score to DeepAccNet_MSA (Figure 4.12B). The analysis shows that ModFOLD9_pure had improved its local assessment accuracy for predicting the lDDT, indicating the consensus quality scores enhanced the accuracy.

To visualise the distribution of the local quality scores, S-score and lDDT density plots were generated. These plots were not as useful for showing the relationship between predicted versus observed S-scores (Figure S.6 in Appendix 12), due to non-linear nature of the S-score, as was mentioned in the previous chapter. Conversely, a strong linear correlation was evident between the predicted lDDT score of ModFOLD9_Pure and the observed lDDT score as illustrated in Figure 4.13.

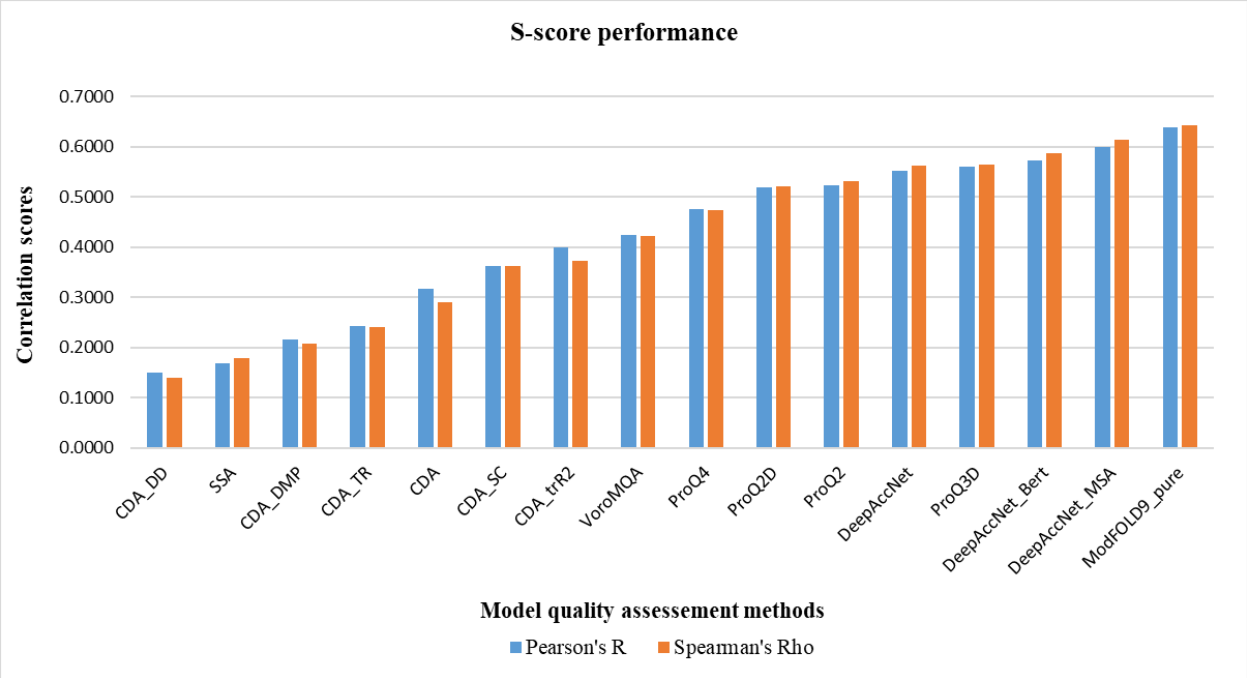


Figure 4.9. Correlations with the S-scores for ModFOLD9_pure and established component methods. The strong positive correlations are closer to 1 and low correlations are closer to 0. The correlation coefficients used were Pearson's R and Spearman's Rho. The established methods include CDA scores derived from contact prediction methods and pure-single model methods. The CDA scores were CDA_DD, CDA_DMP, CDA_SC, CDA_TR, CDA_trR2, and CDA. The pure-single model scores were SSA, ProQ2, ProQ2D, ProQ3D, ProQ4, VoroMQA, DeepAccNet, DeepAccNet_Bert and DeepAccNet_MSA. The scores sorted by Pearson's R values.

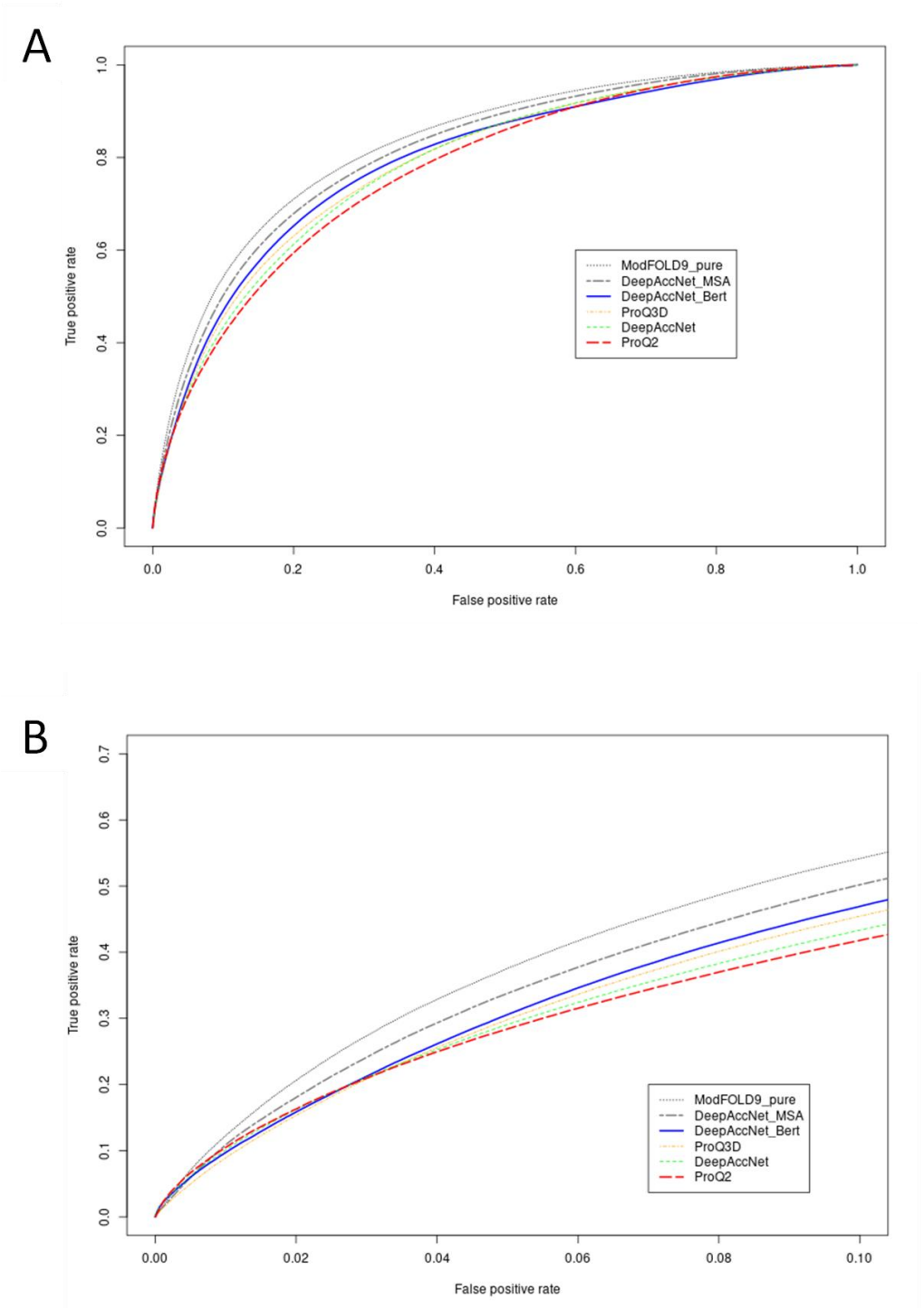


Figure 4.10. ROC curves for ModFOLD9_pure against the top five component methods according to S-score. The top five methods were DeepAccNet_MSA, DeepAccNet_Bert, ProQ3D, DeepAccNet and ProQ2. A) Line graphs of ROC analysis. B) Line graphs with a condition of false positive rate less than 0.1.

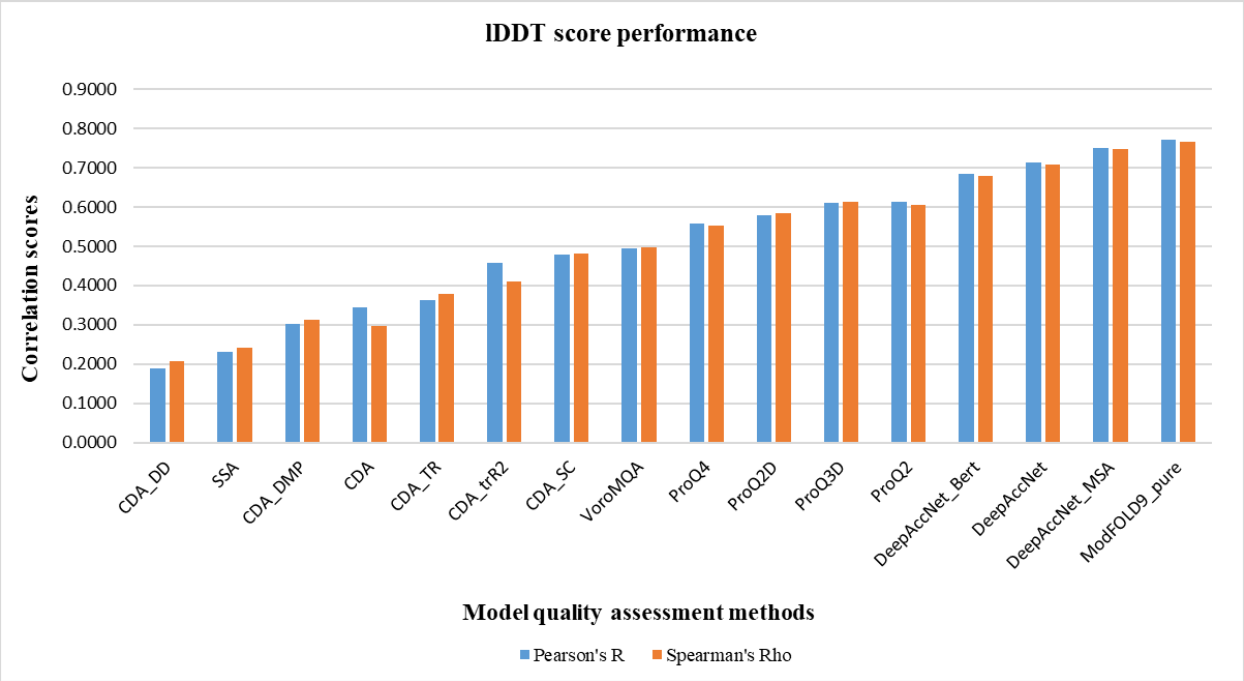


Figure 4.11. Correlations with the lDDT score for ModFOLD9_pure and established component methods. The strong positive correlations are closer to 1 and low correlations are closet to 0. The correlation coefficients used were Pearson's R and Spearman's Rho. The established methods include CDA scores derived from contact prediction methods and pure-single model methods. The CDA scores were CDA_DD, CDA_DMP, CDA_SC, CDA_TR, CDA_trR2, and CDA. The pure-single model scores were SSA, ProQ2, ProQ2D, ProQ3D, ProQ4, VoroMQA, DeepAccNet, DeepAccNet_Bert and DeepAccNet_MSA. The scores sorted by Pearson's R values.

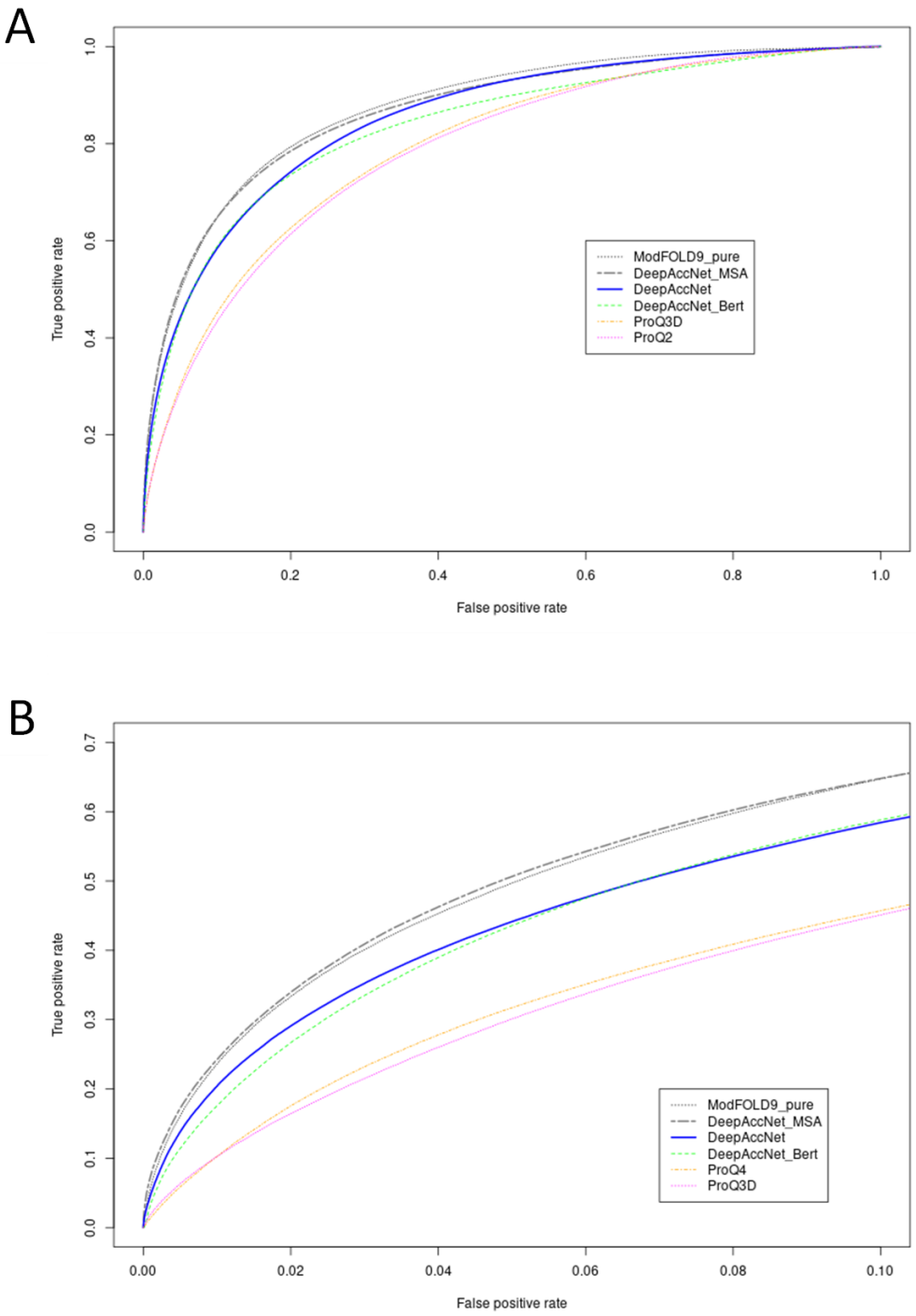


Figure 4.12. ROC curves for ModFOLD9_pure against the top five component methods according to lDDT score. The top five methods were DeepAccNet_MSA, DeepAccNet_Bert, DeepAccNet, ProQ4 and ProQ3D. A) Line graphs of ROC analysis. B) Line graphs with a condition of false positive rate less than 0.1.

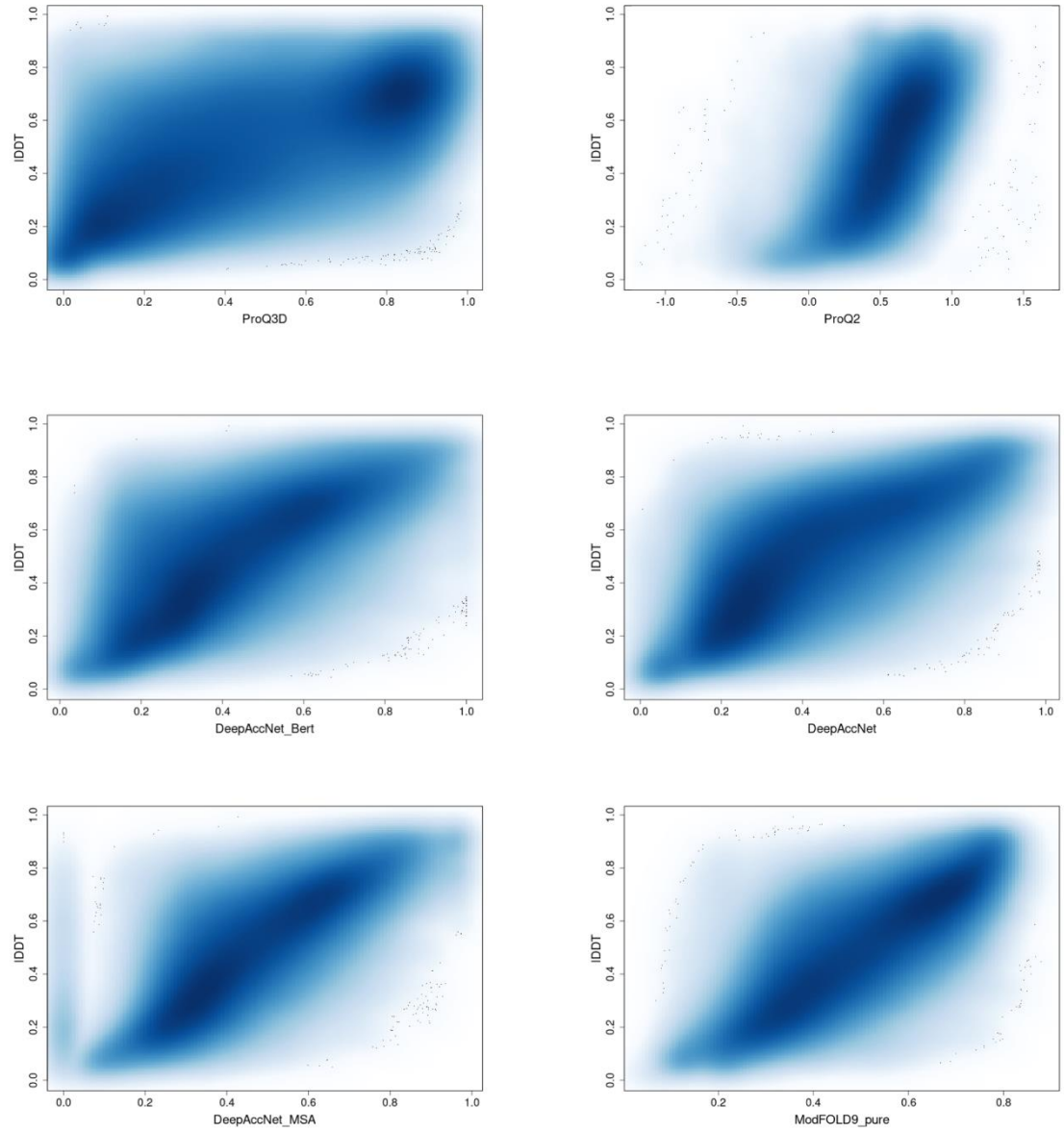


Figure 4.13. Density scatter plots show the relationship between ModFOLD9_pure and its five top component methods according to lDDT scores.

4.4.2.2 Evaluating The Performance of ModFOLD9_quasi

Utilising the consensus approach significantly improved the performance of ModFOLD9_quasi. As demonstrated in Figure 4.14, ModFOLD9_quasi's S-score obtained the highest correlation scores (Pearson's R = 0.708, Spearman's Rho = 0.715) with the observed S-score when compared to other component methods, indicating that ModFOLD9_quasi outperformed established methods. Figure 4.15 shows that the ROC AUC score of ModFOLD9_quasi compared to those of the quasi-single model methods. The highest AUC score went to ModFOLD9_quasi (0.866). Additionally, ModFOLD9_quasi outperformed quasi-single model methods based on AUC scores of ROC FPR <= 0.1 (refer to Figure 4.15B). Based on the prediction of the lDDT score, ModFOLD9_quasi performed even more efficiently. As shown in Figure 4.16, the correlation scores of ModFOLD9_quasi's lDDT score reached almost 0.8 (Pearson's R = 0.797, Spearman's Rho = 0.792). As compared to quasi-single model methods, ModFOLD9_quasi had the highest ROC AUC score (0.891) (Figure 4.17A). In addition, ModFOLD9_quasi had better local assessment accuracy based on ROC AUC FPR <0.1 scores in Figure 4.17B. According to the density plots, ModFOLD9_quasi's predicted lDDT score correlated strongly with the observed lDDT score compared to quasi-single model methods (Figures S.7 in Appendix 13 and Figure 4.18). Based on the analysis, combining quality scores from various QA methods significantly improved ModFOLD9_quasi's local assessment accuracy in predicting both the S-score and lDDT score**.**

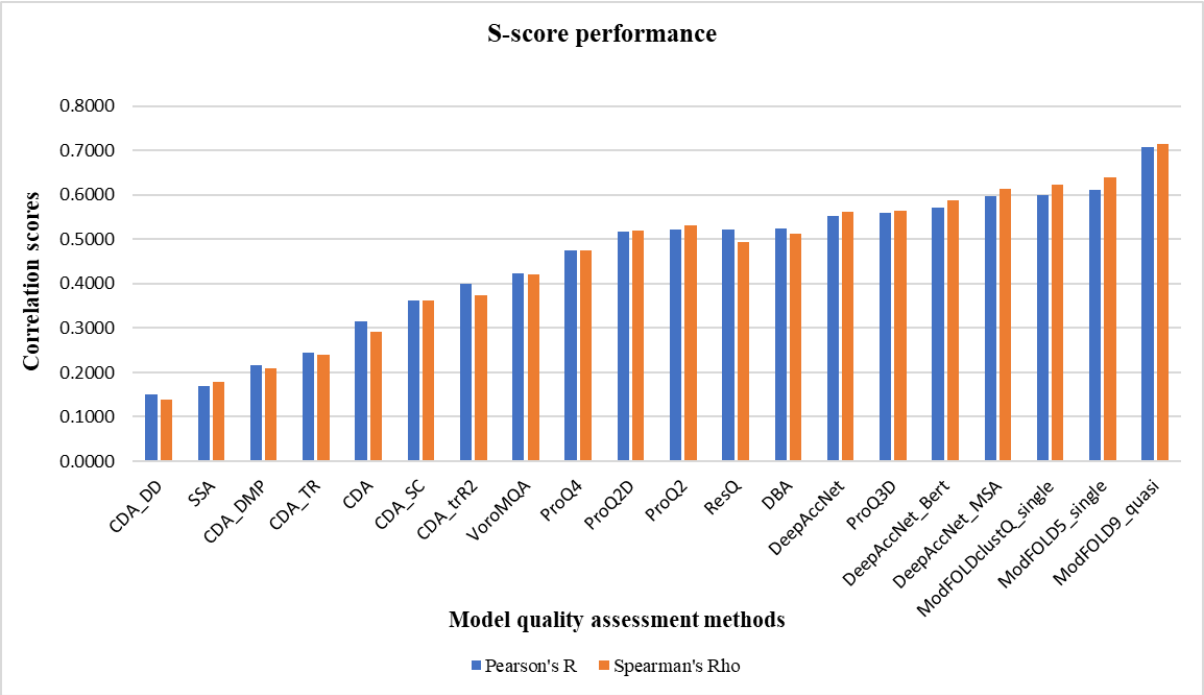


Figure 4.14. Correlations with the S-score for ModFOLD9_quasi and established component methods. The strong positive correlations are closer to 1 and low correlations are closer to 0. The correlation coefficients used were Pearson's R and Spearman's Rho. The established methods include CDA scores derived from contact prediction methods, pure-single model methods and quasi-single model methods. The CDA scores were CDA_DD, CDA_DMP, CDA_SC, CDA_TR, CDA_trR2, and CDA. The pure-single model scores were SSA, ProQ2, ProQ2D, ProQ3D, ProQ4, VoroMQA, DeepAccNet, DeepAccNet_Bert and DeepAccNet_MSA. The quasi-single model methods were ResQ, DBA, ModFOLDclustQ_single and ModFOLD5_single. The scores sorted by Pearson's R values.

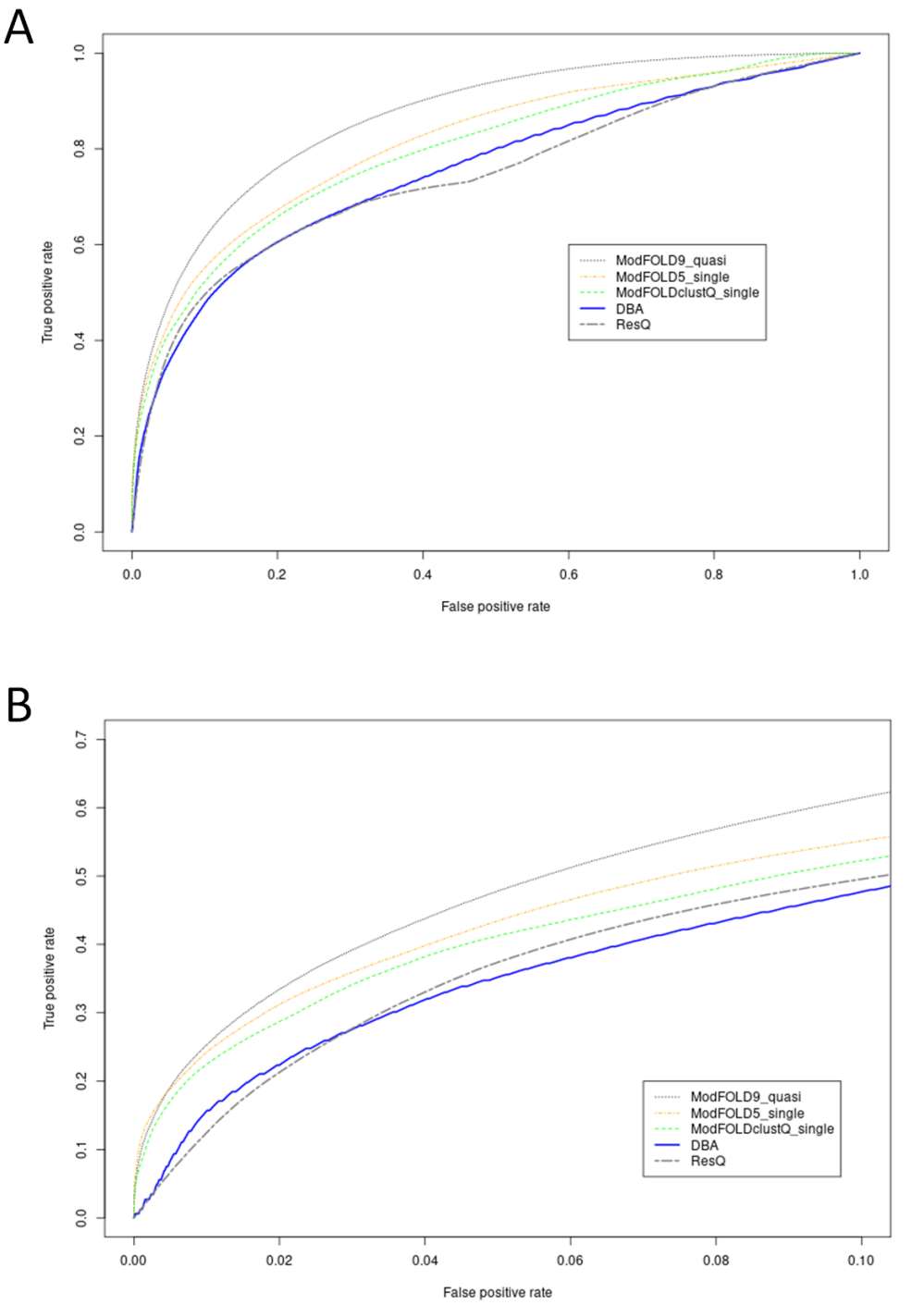


Figure 4.15. ROC curves for ModFOLD9_quasi against the quasi-single model methods according to S-score. The quasi-single model methods were ResQ, DBA, ModFOLD5_single and ModFOLDclustQ_single. A) Line graphs of ROC analysis. B) Line graphs with a condition of false positive rate less than 0.1.

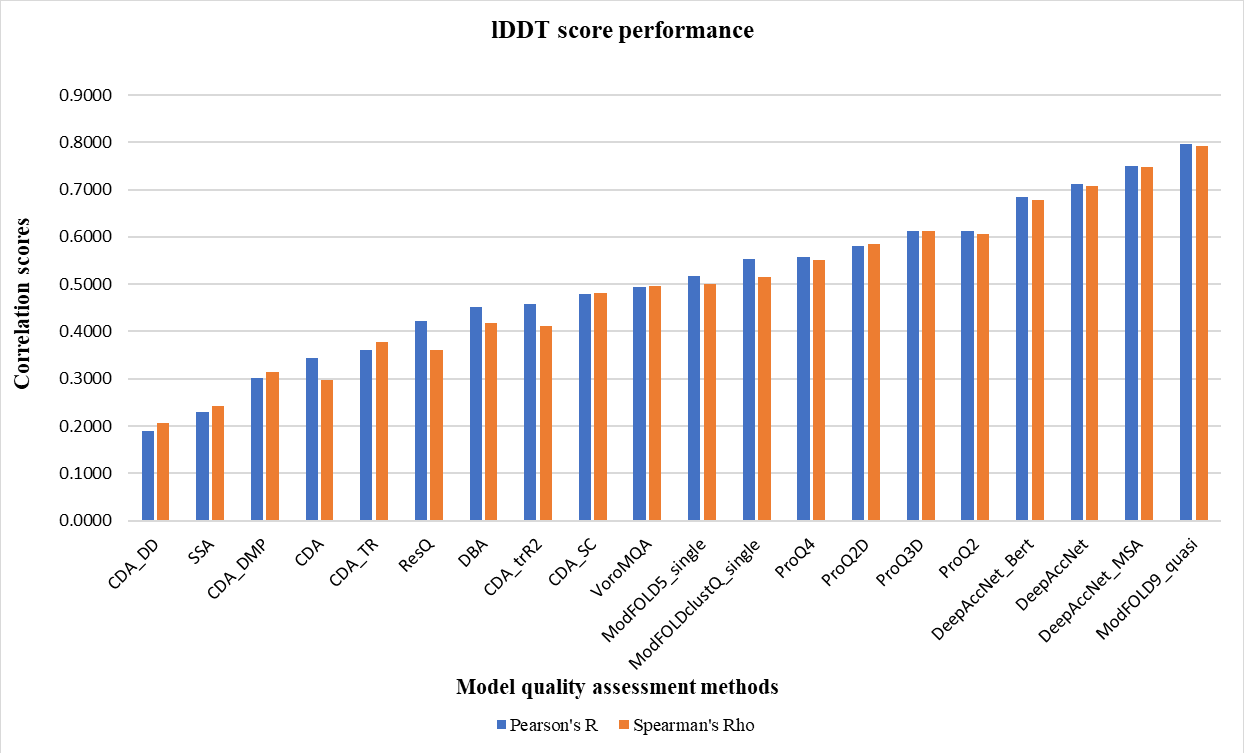


Figure 4.16. Correlations with the lDDT score for ModFOLD9_quasi against those for established component methods. The strong positive correlations are closer to 1 and low correlations are closer to 0. The correlation coefficients used were Pearson's R and Spearman's Rho. The established methods include CDA scores derived from contact prediction methods, pure-single model methods and quasi-single model methods. The CDA scores were CDA_DD, CDA_DMP, CDA_SC, CDA_TR, CDA_trR2, and CDA. The pure-single model scores were SSA, ProQ2, ProQ2D, ProQ3D, ProQ4, VoroMQA, DeepAccNet, DeepAccNet_Bert and DeepAccNet_MSA. The quasi-single model methods were ResQ, DBA, ModFOLDclustQ_single and ModFOLD5_single. The scores sorted by Pearson's R values.

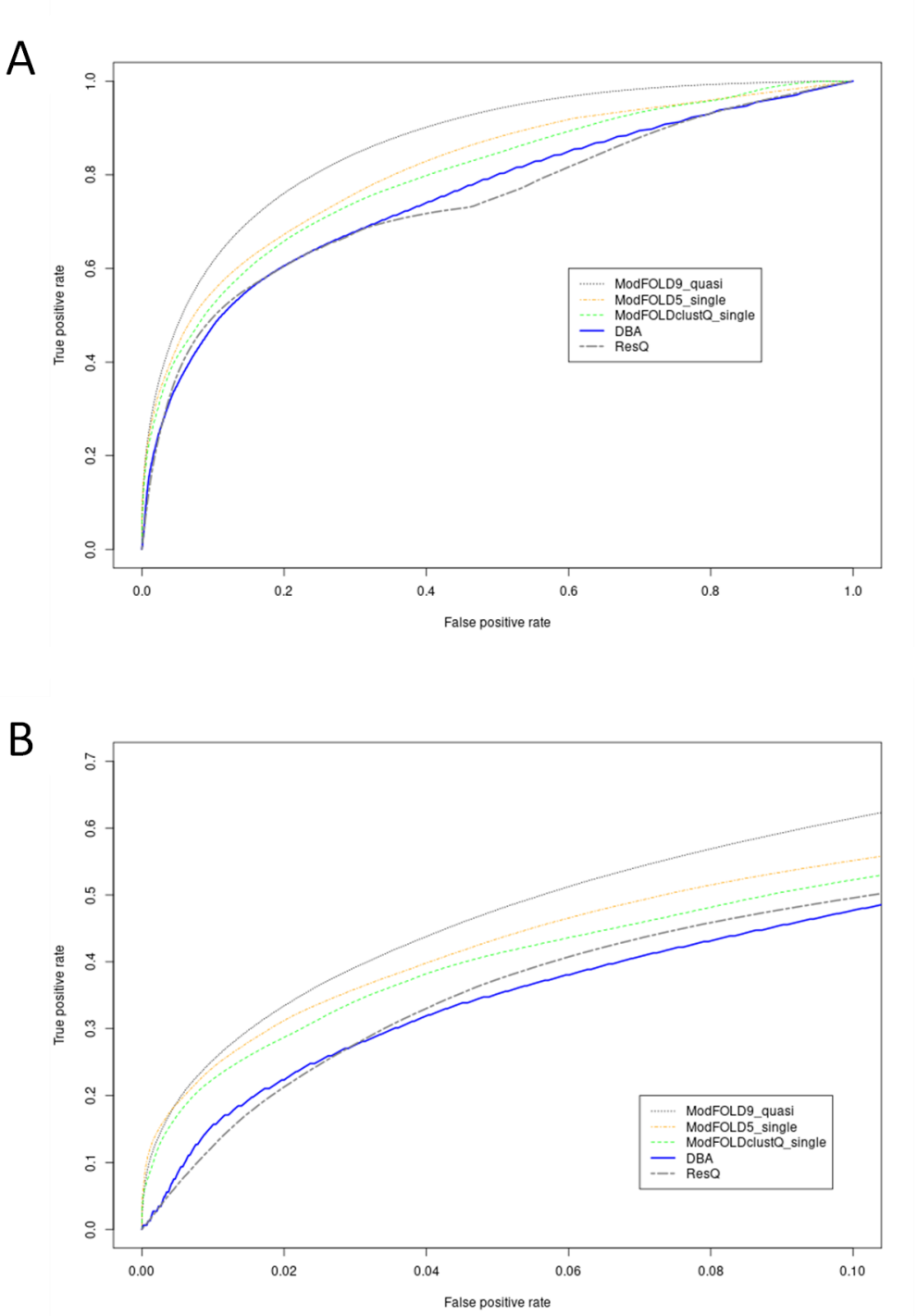


Figure 4.17. ROC curves for ModFOLD9_quasi against the quasi-single model methods according to lDDT score. The quasi-single model methods were ResQ, DBA, ModFOLD5_single and ModFOLDclustQ_single. A) Line graphs of ROC analysis. B) Line graphs with a condition of false positive rate less than 0.1.

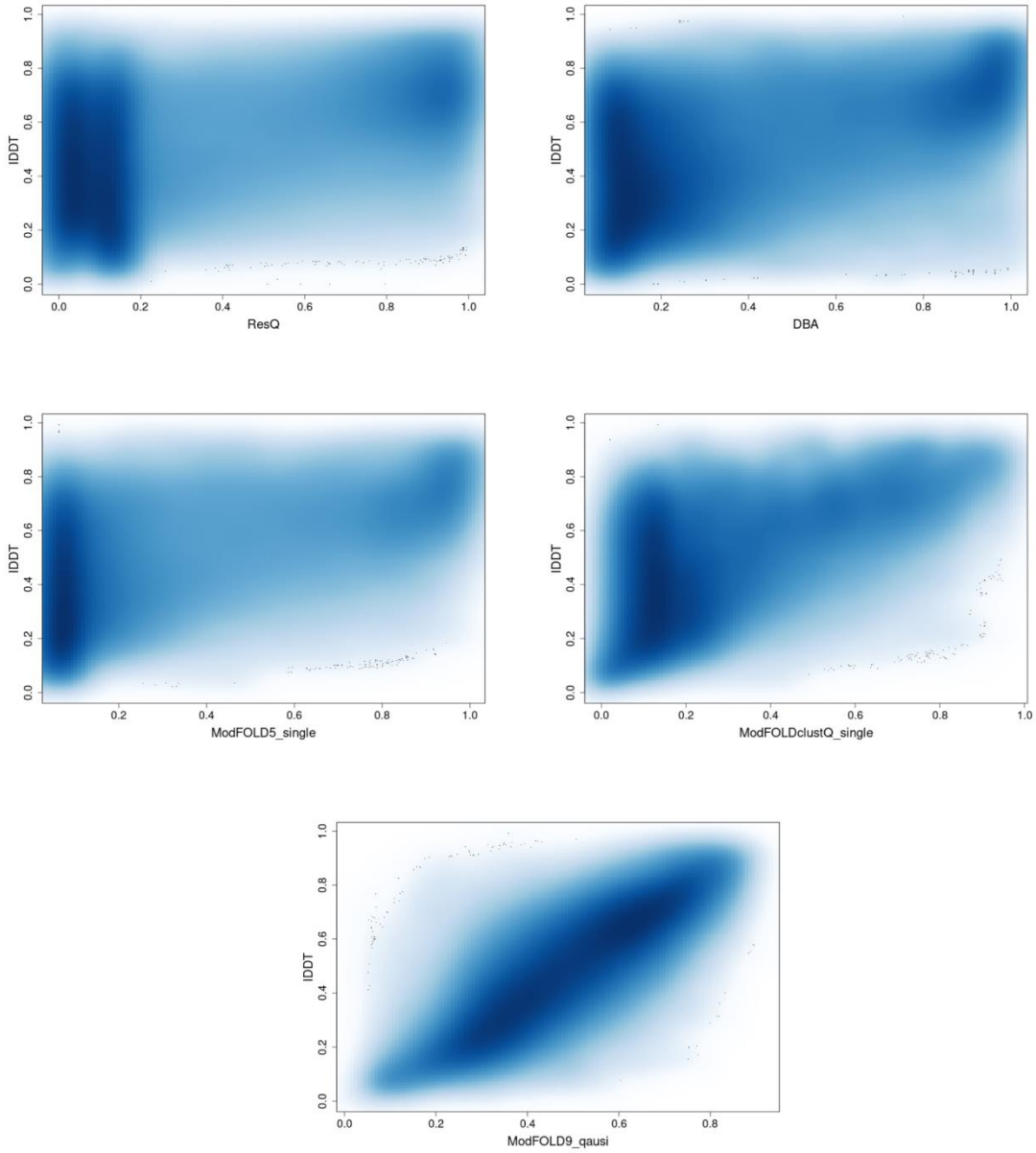


Figure 4.18. Density scatter plots show the relationship between ModFOLD9_quasi and the quasi-single model methods according to lDDT scores.

4.5 Conclusions

For the ModFOLD9 variants, we have implemented consensus approaches that integrate orthogonal data, such as pure-single and quasi-single model scores, using MLPs comprising numerous neurons to enhance the accuracy of local estimation of 3D model quality. These consensus approaches played a vital role in improving the local assessment accuracy of the ModFOLD9 variants as they produced the best local quality scores overall compared with the individual methods. The incorporation of pure-single and quasi-single scores has led to a notable improvement in the performance of ModFOLD9. By optimising the hyperparameters of the MLP, we achieved the highest evaluation scores resulting from the consensus of different quality scoring methods, improving the accuracy of MoFOLD9's assessment on a local scale according to the S-score and lDDT scores (Figure 4.19). All these strategies combined enable ModFOLD9 to provide the most reliable local prediction of 3D model quality out of all of the methods tested.

Previous versions of ModFOLD sever have employed similar strategies, and here we build upon this approach through the integration of a wider range of high performing component scores resulting in much improved predictive performance. The integration of CDA scores, which were derived from deep learning-based contact prediction methods also helped to enhance overall performance. These methods allowed for the accurate prediction of the distance between amino acid pairs in a protein, thereby reflecting the evaluation of local accuracy in the quality of 3D models. In the following chapter, we will explore further objective evaluation of ModFOLD9 compared to previous versions of the method and other state-of-the art approaches, through real-world independent blind tests provided by the CAMEO project. Additionally, we will assess the efficacy of ModFOLD9 scores in accurately estimating our IntFOLD7 server models that were submitted during the CASP15 experiment.

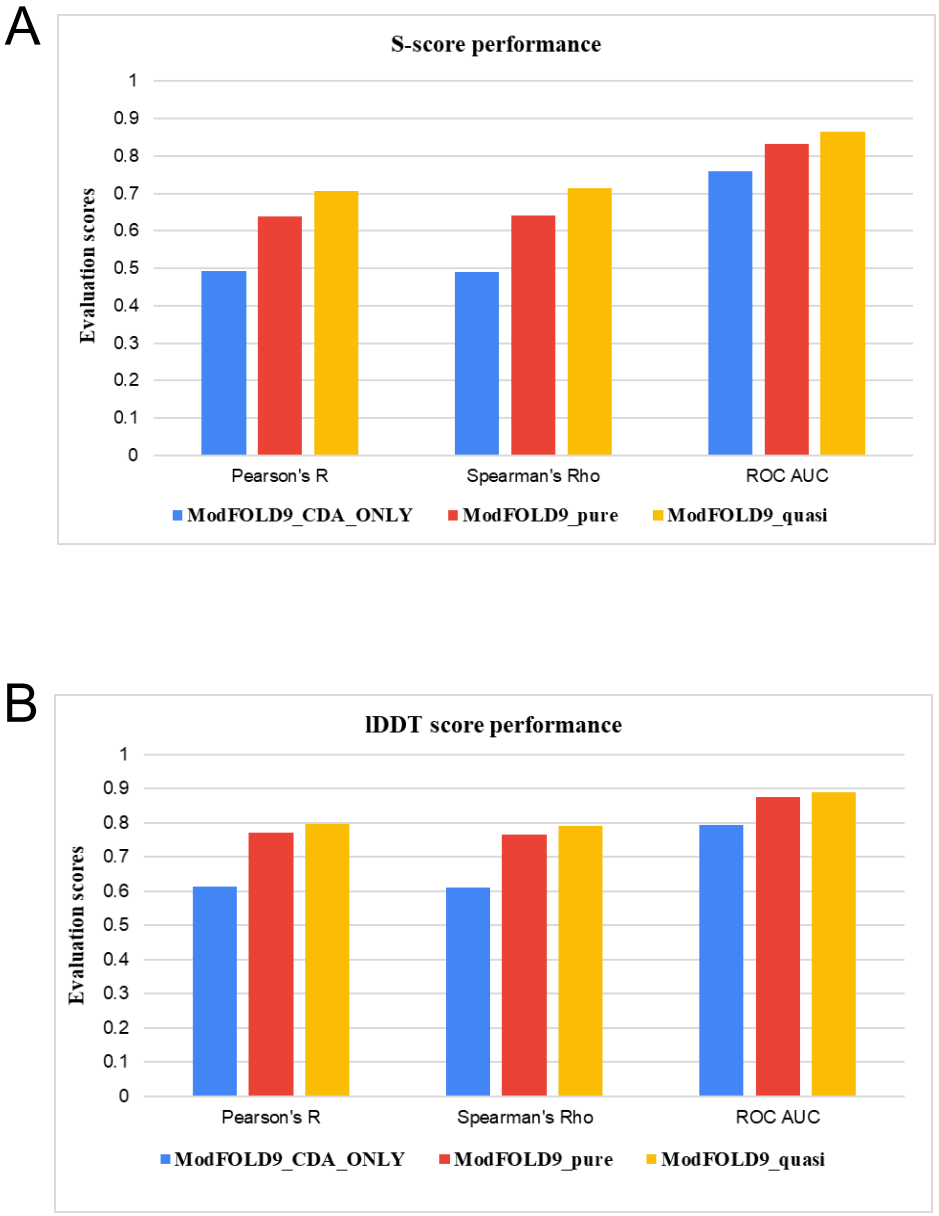


Figure 4.19. The performance of the MLP using different model quality input scores. The evaluation scores were computed for each protocol. A) Analysis of ModFOLD9 performance with different quality scores based on the S-score. B) Analysis of ModFOLD9 performance with different quality scores data based on lDDT. The evaluation measures are Pearson's R and Spearman's Rho correlation and ROC analysis. ModFOLD9_CDA_only refers to the consensus CDA score input. ModFOLD9_pure refers to the consensus CDA score with pure-single scores as input data. ModFOLD9_quasi refers to the combination of quasi-single and pure-single scores with a consensus CDA score.

Chapter 5 Benchmarking of ModFOLD9 and ModFOLDdock performance during the CASP15 experiment and using the CAMEO resource

5.1 Background

Independent benchmarking experiments allow us to objectively assess protein structure prediction methods to obtain data on their relative real-world performance. These tests help researchers in other fields, such as biomedical sciences and drug design discovery, to have critical insights into the application of appropriate computational tools. The CAMEO is a continuous experiment that offers resources to assess computational methods in different categories independently. The primary purpose of CAMEO is to test the validation of prediction algorithms of computational tools, ensuring their effectiveness and reliability. Furthermore, the CASP is a biennial experiment that provides the gold standard blind tests for protein prediction servers and standalone methods. The accuracy of prediction methodologies is assessed on unseen structure data, which will eventually be made public in the Protein Data Bank. The CASP community encourages developers to advance their prediction methods by evaluating the accuracy of their methods' performance in various categories (Kryshtafovych, Schwede, *et al.*, 2021; Robin *et al.*, 2021).

The application of contact prediction methods was apparent in various aspects of protein structure prediction pipelines. Contact maps were an essential step in the progress of many different approaches (Wu, Szilagyi and Zhang, 2011; Jumper *et al.*, 2021a; McGuffin *et al.*, 2021; Ye *et al.*, 2021; Roy *et al.*, 2023). For instance, in our study, contact prediction was applied as one of the deriving scoring methods, the CDA pure-single method, which was used in ModFOLD9 to estimate the accuracy of local regions in 3D models. ModFOLD9 is then used as a self-estimate scoring method for assessing the quality of the IntFOLD7 server models and our manual CASP15 predictions from the McGuffin group (McGuffin *et al.*, 2023). In addition, the CDA score was integrated into the ModFOLDdock scoring system. The ModFOLDdock server was designed by our group in order to estimate the quality of modelled

protein complexes (quaternary structures rather than tertiary structures) (Edmunds *et al.,* 2023; McGuffin *et al.,* 2023). The contribution of contact prediction for improving the protein prediction performance of our servers was evaluated in two real-test community-wide experiments: the CAMEO and the fifteenth experiment of Critical Assessment of protein Structure Prediction (CASP15).

5.1.1 The CAMEO Quality Estimation (QE) Category

Evaluating the accuracy of model quality is essential in protein structure prediction in order to produce the most reliable models. Quality estimates of 3D models are required to help us accurately distinguish between the high-quality and low-quality regions of models. The assessment of the relative performance of QE methods is an independent category in the CAMEO, which the prediction community uses to rank the methods. In this category, CAMEO aims to assess the predicted local quality performance of QE methods based on the predicted lDDT (plDDT) scores for these methods (Haas *et al.,* 2019). The 3D models that were used to evaluate the QE methods are collected from the modelling servers in the CAMEO 3D structure prediction category every week. Each of the QE methods then evaluates these 3D models, and the plDDT scores from each QE method are generated. Subsequently, the observed lDDT scores are then taken once the native structures are available, and then they are compared with the plDDT scores to assess per-residue model quality prediction accuracy for each method. The lDDT scores range from 0 to 100, where lDDT > 60 are defined as well-modelled regions, while lDDT < 60 are defined as poorly modelled regions at the local level (Haas *et al.,* 2019; Robin *et al.,* 2021).

ModFOLD9 has undergone continuous testing using the CAMEO resource over the past few years, since before the start of CASP15 and beyond. The results are published weekly on the CAMEO website based on the lDDT score using a cutoff of 60 to distinguish between high-

and low-quality residues in models. This evaluation resource assists us in examining ModFOLD9 assessment performance and identifying its strengths and weaknesses. In this regard, the CAMEO dataset was used to further analyse the performance of ModFOLD9 local model quality assessment.

5.1.2 IntFOLD7 Method

IntFOLD is a modelling server designed to predict protein tertiary structures and functions from given sequences, providing comprehensive 3D model predictive data to expert and non-expert researchers. The IntFOLD server offers freely accessible high-quality 3D model data, including quality estimates, domain and disorder predictions, models of protein-ligand interactions, as well as the option of further refinement to models (McGuffin and Roche, 2011; Roche *et al.*, 2011; Buenavista, Roche and McGuffin, 2012; McGuffin *et al.*, 2015; McGuffin *et al.*, 2019; McGuffin *et al.*, 2023). The IntFOLD7 server is the most advanced version of IntFOLD to date, which features substantial advancements in its component methods compared to the previous versions. Notably, two advanced 3D modelling servers were integrated into the IntFOLD7 modelling program: trRosetta2 (Anishchenko *et al.*, 2021) and LocalcolabFold (Mirdita *et al.*, 2022) (a community fork of the AlphaFold2 program). Importantly, the update also included enhanced model quality estimations by integrating ModFOLD9 (McGuffin *et al.*, 2023).

IntFOLD7 participated in two categories related to the CASP15 experiment: interdomain and regular modelling. IntFOLD7 demonstrated creditable performance in predicting 3D models of multidomain proteins, outperforming leading human predictor groups (McGuffin *et al.*, 2023). Furthermore, IntFOLD7 performed well when modelling regular proteins. As part of ModFOLD9's accuracy assessment, IntFOLD7's models were examined for further improvement. To explore how ModFOLD9 added to IntFOLD7’s accuracy self-estimates

(ASE) performance, here we present an analysis of the official CASP15 assessment data.

5.1.3 ModFOLDdock Method

In parallel to our ModFOLD9 server, which produces quality estimates for tertiary structure models, we have also developed the ModFOLDdock server, which produces quality estimates for quaternary structure models. Like ModFOLD9, ModFOLDdock uses a consensus approach to combine several quality scores from both single-model and clustering-based methods. The combination of various scores is integrated, which assesses the quality of modelled complexes in multiple aspects. Three variations of ModFOLDdock methods participated in CASP15: ModFOLDdock, ModFOLDdockR, and ModFOLDdockS. Each version estimated complex models with different goals focusing on different facets of the model quality estimation problem. Firstly, the ModFOLDdock variant was developed to generate predicted quality scores for models which would linearly correlate with the observed quality scores. In other words, the predicted and observed scores have a positive linear correlation, where the highest predictions reflect the more accurate the models. Secondly, the goal of the ModFOLDdockR variant is to produce predicted scores that allow us to rank the 3D models most accurately in the order of their observed quality. Finally, the ModFOLDdockS variant was developed as a quasi-single model method to evaluate models on an individual basis against a reference set of models predicted by the MultiFOLD method (Edmunds *et al.*, 2023; McGuffin *et al.*, 2023)

The consensus scoring system of ModFOLDdock variants included seven individual scoring methods, both single-model and clustering-based. The methods were: ModFOLDIA, DockQJury, QSscoreJury, QSscoreOfficialJury, lDDTOfficialJury, voronota-js-voromqa, and the CDA score. The ModFOLDIA score was developed to assess the interface accuracy using a clustering approach. The DockQJury score was based on clustering DockQ scores that assess the docking models' quality between proteins (Basu and Wallner, 2016). The QSscoreJury and

QSscoreOfficialJury scores were computed using the QS-scores clustering method (Biasini *et al.*, 2013; Bertoni *et al.*, 2017). The lDDTOfficialJury score was derived from clustering lDDT scores (Mariani *et al.,* 2013). The voronota-js-voromqa score was a single-model method scoring the surface area of the interface, producing the Voronoi tessellation score (Olechnovič and Venclovas, 2014b). Finally, the CDA score was produced based on deep learning-based contact predictions (Maghrabi and McGuffin, 2017; McGuffin *et al.*, 2018). The range of all scores was scaled between 0 and 1, where the higher scores indicate the high accuracy of models (Edmunds *et al.*, 2023; McGuffin *et al.*, 2023)

The seven scoring methods were used as input scores in various combinations for all three versions of ModFOLDdock. The target scores for the CASP15 QA category for the evaluation of modelled complexes included the overall analogous "fold" score (the global quality of the entire complex), the overall interface quality score (the quality of the interacting residues as a whole) and the per-residue interface confidence scores. The global fold accuracy score of ModFOLDdock was the consensus of the DockQJury and lDDTOfficialJury scores, and the interface accuracy score was the combination of the DockQJury and QSscoreOfficialJury scores. The per-residue interface confidence scores were produced using ModFOLDIA alone. In ModFOLDdockR, the global scores were the mean of three scores: QSscoreJury, lDDTOfficialJury, and voronota-js-voromqa for the fold accuracy, and the mean of the DockQJury, QSscoreOfficialJury and voronota-js-voromqa scores for the interface accuracy. For per-residue interface confidence scores, in this case, the ModFOLDIA and local scores of voronota-js-voromqa were averaged for each residue in the model. Finally, the fold accuracy score of ModFOLDdockS was the mean of the DockQJury and lDDTOfficialJury scores, whereas the interface score was computed by averaging the DockQJury and QSscoreOfficialJury scores. The per-residue interface confidence score of ModFOLDdockS

was the mean of the ModFOLDIA, voronota-js-voromqa and CDA scores (Edmunds *et al.*, 2023). The CASP15 official assessment for the ModFOLDdockS server can be used to gauge how the CDA scores contributed towards the performance of quality estimation for modelled protein complexes.

5.2 Aims and Objectives

The main aim of this chapter is to examine the contribution of contact prediction in improving the predictive capabilities of IntFOLD7 via ModFOLD9 and in boosting the quality estimates of the ModFOLDdockS server based on real-world tests using the CASP15 and CAMEO data. The first objective is to evaluate IntFOLD7's self-estimation accuracy based on the CASP15 data for regular targets. To accomplish this, ModFOLD9's plDDT scores, which measured the local quality of IntFOLD7's models, were collected from this dataset for further analysis. IntFOLD7's self-estimation accuracy was then compared with other modelling methods' accuracy using the same CASP15 data. An additional comparison was performed based on the CASP15 official assessment of IntFOLD7 according to the global lDDT and ASE scores. The second objective is to assess ModFOLD9's performance in estimating the local accuracy of models from other modelling methods. ModFOLD9 was used to estimate the models predicted by three different groups during the CASP15 experiment. After that, a similar analysis was conducted on ModFOLD9's plDDT scores for these group's models. The third objective is to examine ModFOLD9's local performance based on lDDT scores using the CAMEO common subset across varying time frames. This exercise was done in three ways. The first was to rank ModFOLD9's performance against independent server variants of its component methods. The second was to assess ModFOLD9's performance against the previous versions of ModFOLD. The third was to compare the ModFOLD9 local assessment with other top-ranked model quality methods. Lastly, for ModFOLDdockS, we aimed to analyse the accuracy of the per-

residue interface confidence scores according to the observed local lDDT and CAD scores presented in the CASP15 official evaluation of QE methods.

5.3 Methods

5.3.1 Data Collection

CAMEO data is generated every week to benchmark the QA methods, so ModFOLD9 was tested on CAMEO's independent real-world data every week for >1 year. In our study, ModFOLD9's predicted models were downloaded from the CAMEO website (https://www.cameo3d.org/quality-estimation/) over different time frames: one month, three months, six months and one year. We focused on evaluating the lDDT scores of models assessed by ModFOLD9 to gauge its local assessment performance compared with other QA methods. To perform a fair comparison, the CAMEO data was pre-processed to generate a common subset, in which all compared methods had run on the same number of targets (see Table 5.1). Common subset analysis is unavailable for the QE category on the CAMEO website, so the data were downloaded and analysed in-house using bespoke Python3 and R scripts (the common subset code written in Python3 in Appendix 14, where the ROC analysis was conducted using R in Appendix15). Following this, the lDDT scores for all models predicted by each method were compiled into one file. Then, the ROC analysis was carried out to derive the ROC AUC and ROC AUC FPR <= 0.1 scores of lDDT scores for each method. The comparison between QA methods on ROC AUC and ROC AUC FPR <= 0.1 scores was conducted, ranking their local assessment accuracy using the lDDT local score cutoff at 60.

ModFOLD9 performance was compared in three different ways on CAMEO common subset data. The first comparison was conducted to rank the ModFOLD9 local assessment accuracy against the established servers of its component methods. The second comparison was to

investigate how much ModFOLD9 improved by comparing it with its preceding versions. The last comparison was assessing the ModFOLD9 performances with three top-quality assessment methods. All comparisons were conducted based on the ROC AUC and ROC AUC FPR <= 0.1 scores using the lDDT score cutoff at 60.

Table 5.1. Common subsets from the CAMEO dataset over different time frames. Three comparisons were performed on the common subsets. The first comparison was to compare the local quality assessment of ModFOLD9 to that of the independent server variants of its component methods. The second comparison was to compare ModFOLD9's local assessment performance against ModFOLD's previous versions. The third comparison was ModFOLD9's performance in local assessment with the top-ranked quality assessment methods.

	Common subset data		
Time frame	**First comparison**	**Second comparison**	**Third comparison**
One month	3376 models	830 models	1980 models
Three months	10152 models	4370 models	4110 models
Six months	21376 models	10360 models	10740 models
One year	12856 models	7380 models	6850 models

In the tertiary structure prediction category, CASP15 data was collected for 68 regular targets to assess ModFOLD9 performance through IntFOLD7. The predicted models and native structures of regular targets were obtained from the website (https://predictioncenter.org/casp15/index.cgi)—the analysis was conducted for two purposes. The first was to assess the self-estimation predictive performance of IntFOLD7 and compare it with other top-performing modelling methods. Here, the analysis was done on 68 targets and 3352 models for ten modelling methods, including IntFOLD7. It should be noted that, despite

the fact that the methods were supposed to predict five models for each target, the number of models was lower than 3400 (68*5*10 = 3400). This is because some methods analysed in this study predicted fewer than five models for some targets. The second purpose of the CASP15 analysis was to assess the ModFOLD9 performance in estimating the quality of models from other groups. For this purpose, the analysis was conducted on 26 targets and 494 models for three groups: Elofsson group (af2-standard), Baker group (BAKER-SERVER) and Colabfold group (LocalcolabFold) (Detailed descriptions of the three groups' servers can be found in the abstracts of CASP15 https://predictioncenter.org/casp15/doc/CASP15_Abstracts.pdf).

The official assessment results of CASP15 targets have been collected from https://predictioncenter.org/casp15/index.cgi to show the performance of IntFOLD7 and ModFOLDdock methods on their two categories: regular targets modelling and the estimates of model accuracy (EMA), respectively. For IntFOLD7, the results were for 130 domains of 43 regular targets. For ModFOLDdockS, the CASP15 official assessment results were for the individual residue confidence scores based on the 40 multimeric targets.

5.3.2 CASP15 Assessment Metrics

The tertiary structure prediction methods were evaluated according to different measures that assess their performance according to many aspects. These measures included the Global Distance Test - Total Score (GDT_TS), the Global Distance Test - High Accuracy (GDT_HA), the local Difference Distance Test (lDDT) and the Accuracy of Self-Estimate (ASE) (Pereira *et al.*, 2021). The two latest measures were considered in our study. The local Difference Distance Test score was designed to compute the difference between the relative per-residue positions in predicted models and the corresponding relative per-residue positions in the reference structures (See Chapter 3).

The accuracy of the self-estimate (ASE) score measured how accurate the modelling servers

are at estimating the error of each residue in their own models. The main aim of this measure was to assess the predictive quality estimation performance of each modelling server (Kryshtafovych, Monastyrskyy and Fidelis, 2016; Pereira *et al.*, 2021). The ASE evaluation metric was first used in the assessors' formula by the CASP12 assessors to emphasise the importance of accuracy self-estimates by modelling servers. In previous CASP experiments, predictors were asked to estimate the distances between their models' predicted residues and the corresponding residues in native structures according to a structural superposition. The S-function formula was used to compute ASE as follows:

$$S(d) = \frac{1}{1 + (\frac{d}{d0})^2}$$

Where d was normalised in the range 0 and 1. The S score averaged for the model and renormalised in range 0 and 100 by the formula:

$$ASE = 100\ x(1 - \frac{1}{N}\sum_{i=1}^{N}|S(e_i)\ -\ S(d_i\ )|)$$

Where d_i is the actual distance, e_i is the predicted error, and d_0 is a scaling factor set to 5. Thus, higher ASE scores reflect better accuracy in self-estimates (Pereira *et al.*, 2021). In the recent CASP15 experiment, the assessors adapted the ASE formula to include the plDDT and lDDT scores as follows:

$$ASE\ =\ 100\ -\ Mean(|plDDT_i\ -\ lDDT_i|)$$

Where plDDT is local per-residues error estimation from modelling servers. Both plDDT and lDDT scores range between 0 and 100.

The CASP15 evaluation matrices for measuring the accuracy of interface residue confidence scores for modelled protein complexes were the PatchDockQ, PatchQS, lDDT and CAD

scores. Here, we presented the official assessment on lDDT and CAD as these scores are contact-based measurements. These scores scored the atom positions within specific model regions to determine their differences with the experimental structures (Kamisetty, Ovchinnikov and Baker, 2013; Mariani *et al.*, 2013; Kryshtafovych *et al.*, 2023). CAD score measured the differences between contact surface areas of the predicted residues in a model with their corresponding residue in the native experimental structure (Pereira *et al.*, 2021)

5.4 Results and Discussion

5.4.1 Independent Benchmarking of Local Quality Estimations for ModFOLD9 with CAMEO Data

ModFOLD9 and ModFOLD9_pure were evaluated for their local quality estimation performance based on per-residue lDDT scores with CAMEO common subset data. The ROC AUC and ROC AUC FPR <= 0.1 scores were calculated, based on the lDDT score < 60 cutoff, for ModFOLD9 and ModFOLD9_pure and their performance was compared with that of other QA methods. Three comparisons were conducted. We first compared ModFOLD9 (MF9) and ModFOLD9_pure (MF9_pure) against the QA servers that are components built into the MF9 and MF9_pure pipelines. A second comparison was made between MF9 and MF9_pure against the three previous versions of ModFOLD9. Finally, the third comparison was with the top QA methods. Each comparison of the ModFOLD9 performance was made over four different time frames (one month, three months, six months, and one year) to examine improvements made over time with an increasing data set.

ModFOLD9 and ModFOLD9_pure performed best against the component methods over all four periods, reaching ROC AUC score above 0.9 and ROC AUC FPR<= 0.1 score around 0.07, as shown in Figures 5.1, 5.2 and Table S.6 in appendix 16. ModFOLD9 showed

significant improvement in local assessment accuracy when compared to the previous versions. When comparing ModFOLD9 with ModFOLD8, the ROC AUC score of ModFOLD9 is ~5 % higher, which indicates that local predictive assessment accuracy has been improved (Figure 5.3, Table S.7 in appendix 17). At FPR <= 0.1, ROC AUC for ModFOLD9 was the highest among all ModFOLD versions (Figure 5.4 and Table S.7 in appendix 17). This indicates that the integration of consensus of CDA and other scores in ModFOLD9 had improved its ability to estimate local 3D model regions. As part of the MF9 upgrade, three deep-learning-based contact prediction methods have been added to exploit the benefits of distance and contact prediction methods in model quality assessment. Additionally, according to our CAMEO common data set analysis, ModFOLD9 was ranked as one of the leading individual methods in the world for assessing the local quality of 3D models (Figures 5.5, 5.6 and Table S.8 in appendix 18). ModFOLD9 achieved second place among QA methods for one month. In contrast, it achieved third and fourth positions during the other periods.

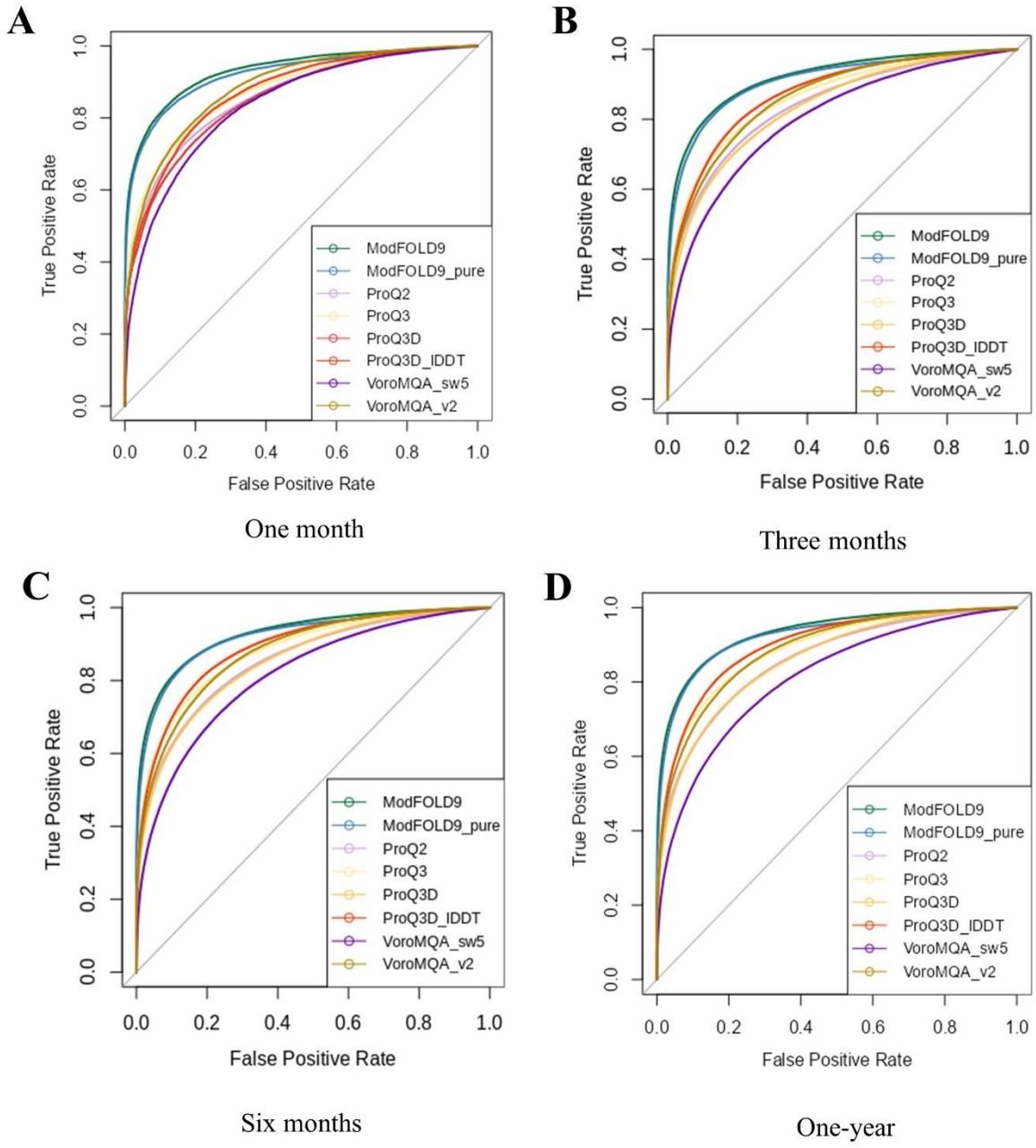


Figure 5.1. ROC curves compare the local assessment accuracy for ModFOLD9 performance against independent servers based on its component methods based on ROC AUC scores (lDDT cutoff < 60) on common subset CAMEO data. Based on A) One month, B) Three months, C) Six months, and D) One year data.

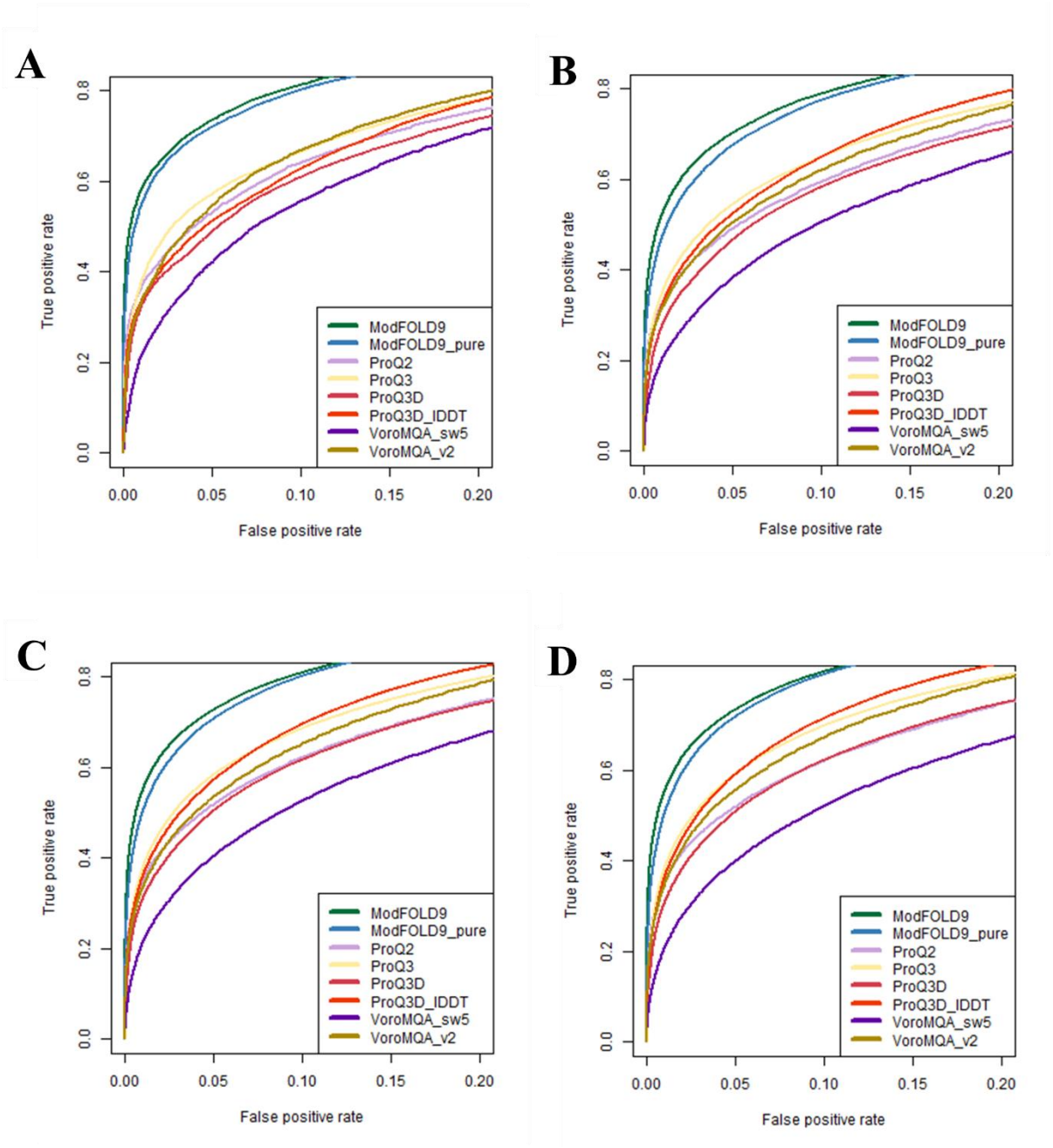


Figure 5.2. ROC curves at False Positive Rate <= 0.1 compare the local assessment accuracy for ModFOLD9 performance against independent servers based on its component methods based on ROC AUC scores (lDDT cutoff < 60) on common subset CAMEO data. Based on A) One month, B) Three months, C) Six months, and D) One year data.

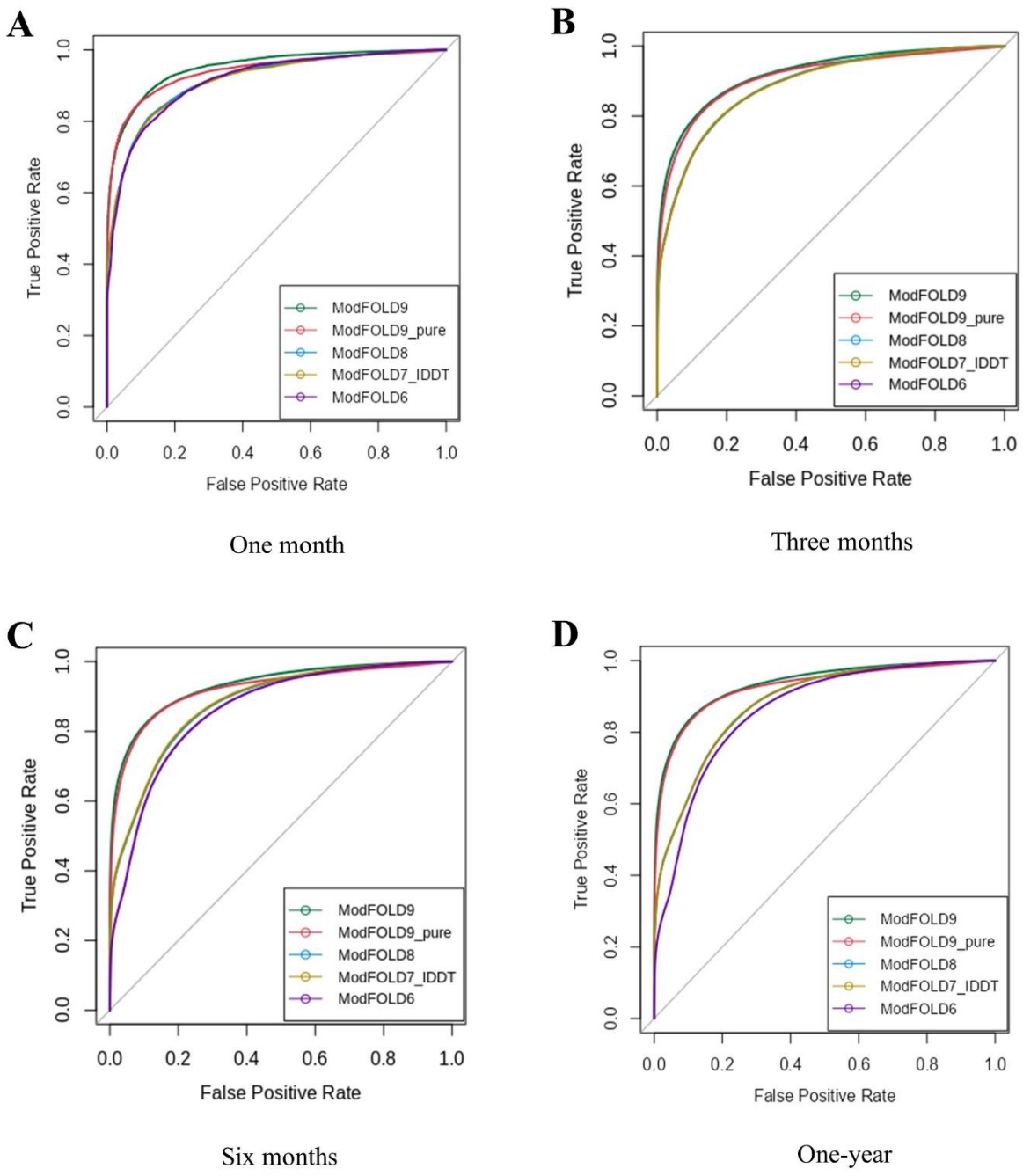


Figure 5.3. ROC curves compare the local assessment accuracy for ModFOLD9 performance against its previous versions based on ROC AUC scores (lDDT cutoff < 60) on common subset CAMEO data. Based on A) One month, B) Three months, C) Six months, and D) One year data.

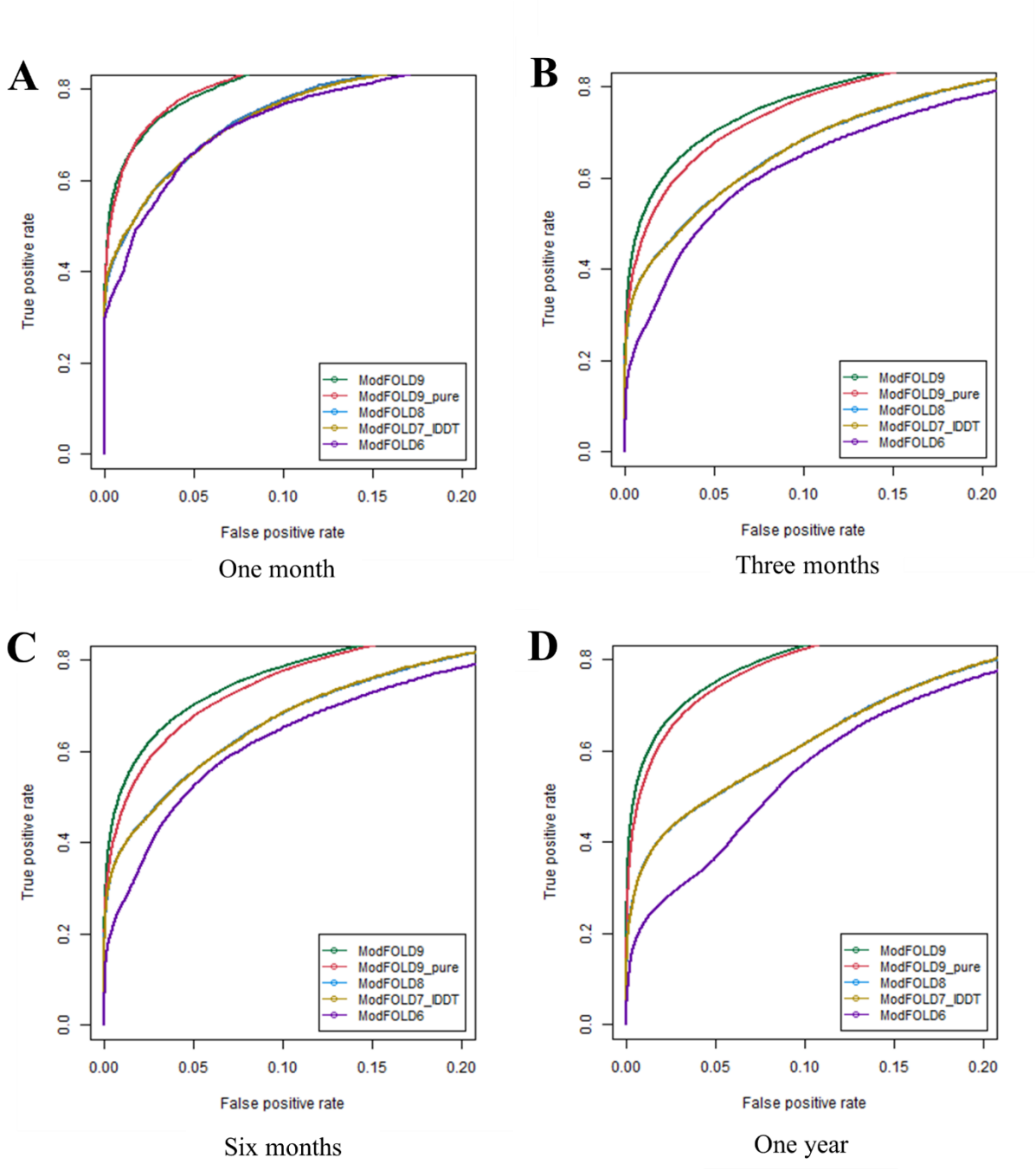


Figure 5.4. ROC curves at False Positive rate <= 0.1 compare the local assessment accuracy for ModFOLD9 performance against its previous versions based on ROC AUC FPR <= 0.1 scores (lDDT cutoff < 60) on common subset CAMEO data. Based on A) One month, B) Three months, C) Six months, and D) One year data.

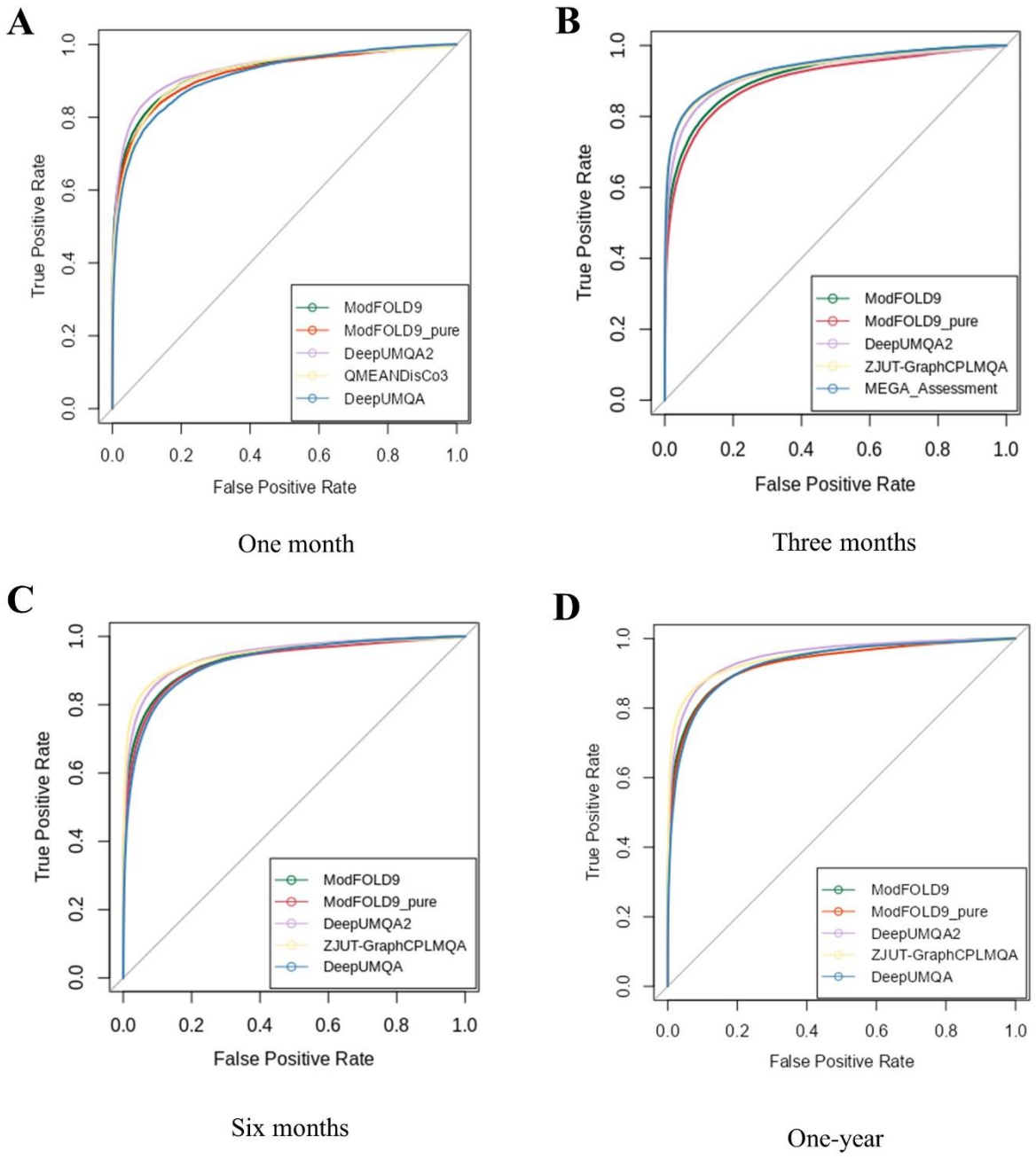


Figure 5.5. ROC curves represent a comparison of the local assessment accuracy for five leading quality assessment methods based on ROC AUC score (lDDT cutoff < 60) on common subset CAMEO data. Based on A) One month, B) Three months, C) Six months, and D) One year data.

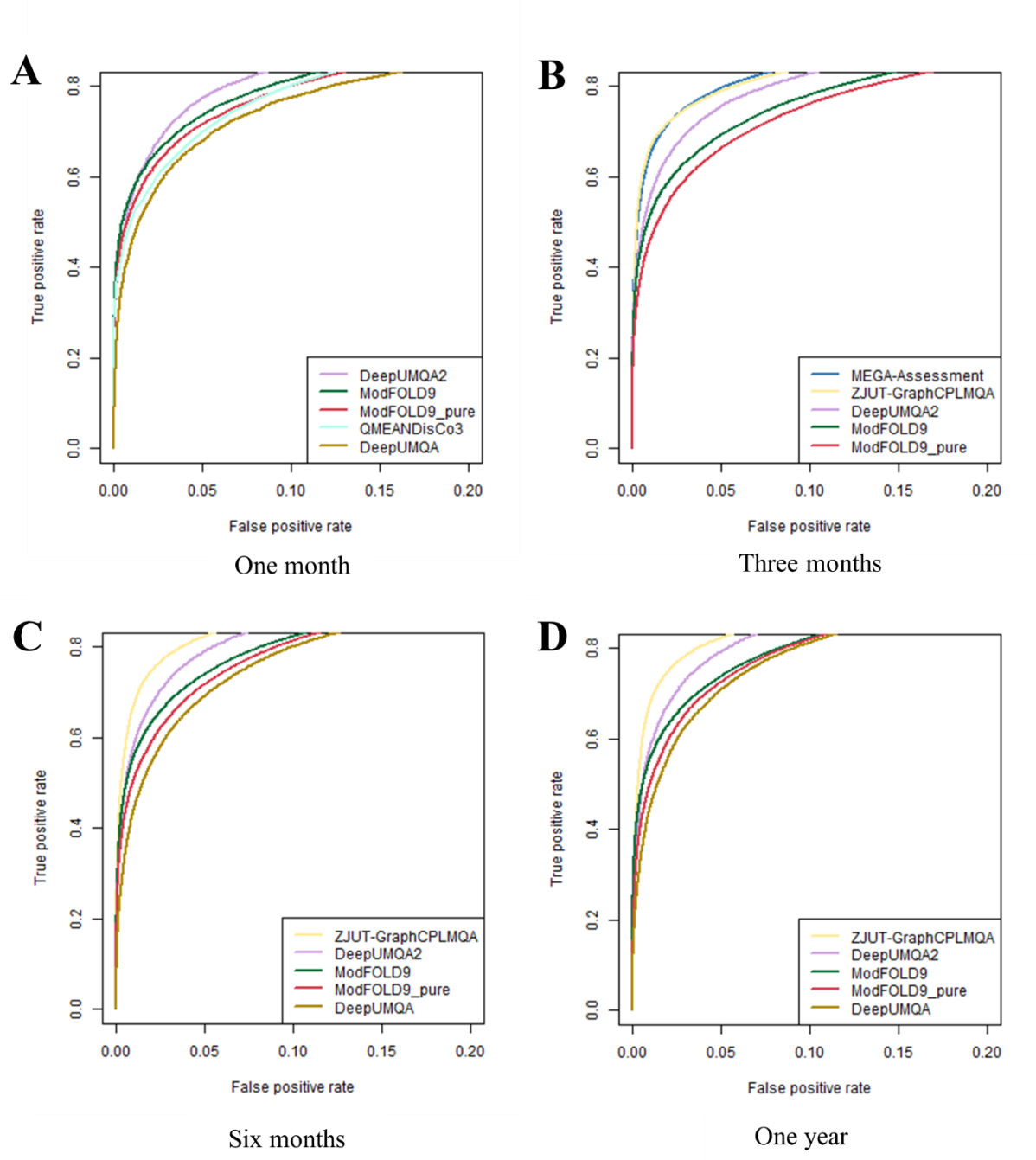


Figure 5.6. ROC curves at False Positive rate <= 0.1 represent a comparison of the local assessment accuracy for five leading quality assessment methods based on ROC AUC FPR <= 0.1 score (lDDT cutoff < 60) on common subset CAMEO data. Based on A) One month, B) Three months, C) Six months, and D) One year data.

5.4.2 Independent Benchmarking of IntFOLD7 and ModFOLDdockS with CASP15 Data

5.4.2.1 IntFOLD7 Self-Estimation Prediction Performance

The accuracy self-estimation scores for IntFOLD7 models are calculated using ModFOLD9. Thus, the ModFOLD9 local quality estimation performance can be measured in part from the official assessment results of CASP15 for IntFOLD7. The global lDDT and ASE scoring methods focused on evaluating the 3D modelling accuracy of a server and its own model quality estimates, respectively. Table 5.2 shows the CASP15 official assessment in the regular modelling category for ten modelling methods. IntFOLD7 performed relatively well based on average global lDDT scores. In the context of the average score of ASE for the first model and all models, IntFOLD7 demonstrated competitive performance compared to the other methods. These findings indicate that QA methods ModFOLD9 positively contribute to enhancing the predictive performance of IntFOLD7.

A detailed analysis has been conducted on the per-residue scores for IntFOLD7 models to assess how ModFOLD9 performed on CASP15 at the local quality estimation level. The prediction data for regular targets (tertiary structures) were assessed according to the plDDT scores, as these scores represent the per-residue quality estimates for each 3D model.

The evaluation analysis encompassed both ROC analysis and correlation analysis. The plDDT scores from each server were then compared with the observed lDDT scores of models compared to the native structures. We also evaluated the nine alternative modelling methods, of which four were the top-performing server groups based on the CASP15 z-score. Then, we compared IntFOLD7's self-estimation performance with each modelling server's. Initially, we compared it with all nine modelling methods and subsequently, for clarity, narrowed the comparison down to just the top four server groups.

Table 5.2. The official assessment results for 68 CASP15 regular targets and 3352 models from ten modelling servers. The scores are the averages of lDDT scores and ASE scores for ten modelling servers. The suffix "All" stands for all models, and "Model1" stands for just the first model for each target predicted by the modelling servers. The table is sorted by the average of ASE for the first model. The table adapted from https://predictioncenter.org/casp15/results.cgi?view=tables&target=T1104-D1&model=1&groups_id= .

Group Name	Average Global lDDT_All	Average ASE_All	Average Global lDDT_Model1	Average ASE_Model1
UM-TBM	**0.81**	87.19	0.80	**87.46**
DFolding-server	0.79	87.95	0.79	87.41
NBIS-AF2-standard	0.79	**88.78**	0.78	87.21
NBIS-AF2_multimer	0.79	87.20	0.77	87.11
IntFOLD7	0.76	86.71	0.74	86.78
RaptorX	0.80	88.11	0.79	86.58
MULTICOM_refine	**0.81**	86.38	**0.81**	85.99
ManiFold-E	0.78	82.74	0.79	84.68
BAKER-SERVER	0.77	86.70	0.74	84.45
Yang-Server	0.79	85.39	0.79	84.37

ROC analysis reveals how accurate the modelling servers are in estimating the quality of their models, particularly at a local level. In other words, it answered the question about to what extent these methods are able to distinguish between correct and incorrect local regions of their own models. The results showed that all modelling servers perform well in predicting their local models' accuracy, as suggested by their ROC AUC and ROC AUC FPR <= 0.1 scores (see Figures 5.7A and 5.7B). NBIS-AF2-standard was ranked as the best server in self-estimation (ROC AUC= 0.94, ROC AUC (FPR<= 0.1) = 0.06), indicating its strength in local estimation. Despite this, Figures 5.7C and 5.7D show that IntFOLD7 scored higher (ROC AUC = 0.88, ROC AUC (FPR<= 0.1) = 0.039) than three other top-performing servers: MULTICOM_refine, Yang-Server, and ManiFold-E, suggesting that it may be better at estimating its own errors than these other servers (Table 5.3). Pearson's R and Spearman's Rho correlation analysis shows agreement with ROC analysis, where IntFOLD7 plDDT scores have

a high correlation with observed lDDT scores (Table 5.3). Furthermore, the scatter plots of IntFOLD7 self-estimate scores versus the observed accuracy scores show a strong linear relationship (Figure 5.8).

The variation in performance raises the question of how confident the very best modelling servers are in assessing the local accuracy of their models. Some servers could generate high-accuracy models globally but may struggle to evaluate their relative accuracy at the local level. To improve in areas where other servers may have limitations, IntFOLD7 utilises its own leading estimation server, ModFOLD9, to enhance the self-estimation performance. ModFOLD9 integrates six deep learning-based methods, which focus on predicting contact and distance distribution, which may assist us in selecting 3D models with more accurate contacts.

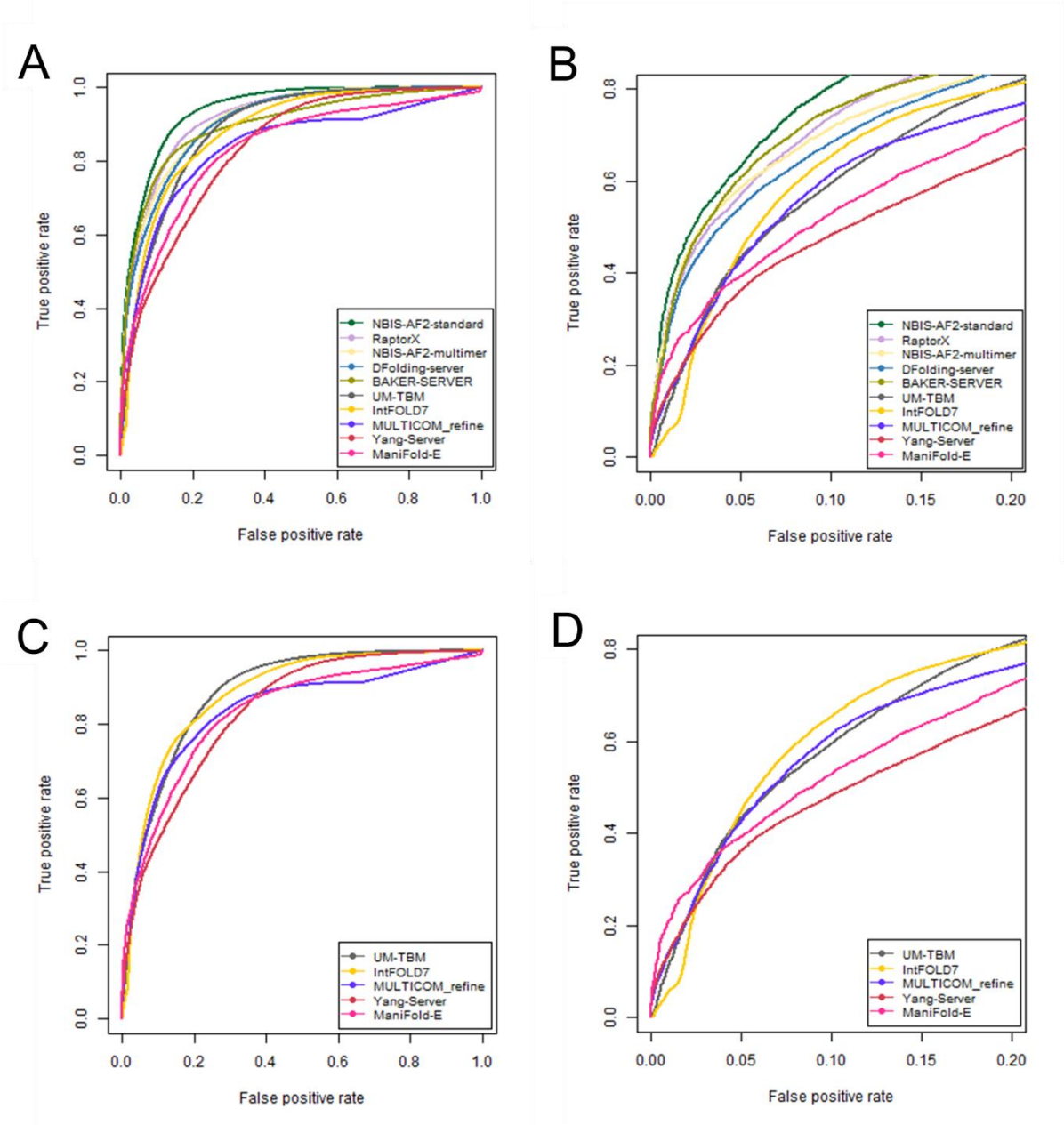


Figure 5.7. ROC curves compare the self-estimation accuracy for ten modelling methods on CASP15 regular targets based on the ROC AUC score (lDDT cutoff < 60). A) ROC AUC for ten modelling methods. B) ROC AUC FPR <= 0.1 for ten modelling methods. C) ROC AUC for IntFOLD7 against the top four performing modelling servers. D) ROC AUC FPR <= 0.1 for IntFOLD7 against the top four performing modelling servers.

Table 5.3. Correlation analysis for modelling methods on CASP15 data. Pearson's R and Spearman's Rho correlation coefficients measure the relationship between the predicted models and native structures based on plDDT scores. The table is sorted by Pearson's R values. The top four modelling servers in terms of model quality are in bold.

Group name	Pearson's R	Spearman's Rho	ROC AUC	ROC AUC FPR <=0.1
NBIS-AF2-standard	**0.87**	**0.78**	**0.94**	**0.060**
RaptorX	0.81	0.75	0.92	0.054
NBIS-AF2-multimer	0.77	0.70	0.91	0.050
DFolding-server	0.76	0.72	0.91	0.053
BAKER-SERVER	0.75	0.76	0.90	0.056
UM-TBM	0.75	0.66	0.89	0.038
IntFOLD7	0.71	0.58	0.88	0.039
MULTICOM_refine	0.61	0.54	0.84	0.033
Yang-Server	0.55	0.66	0.84	0.039
ManiFold-E	0.54	0.57	0.83	0.037

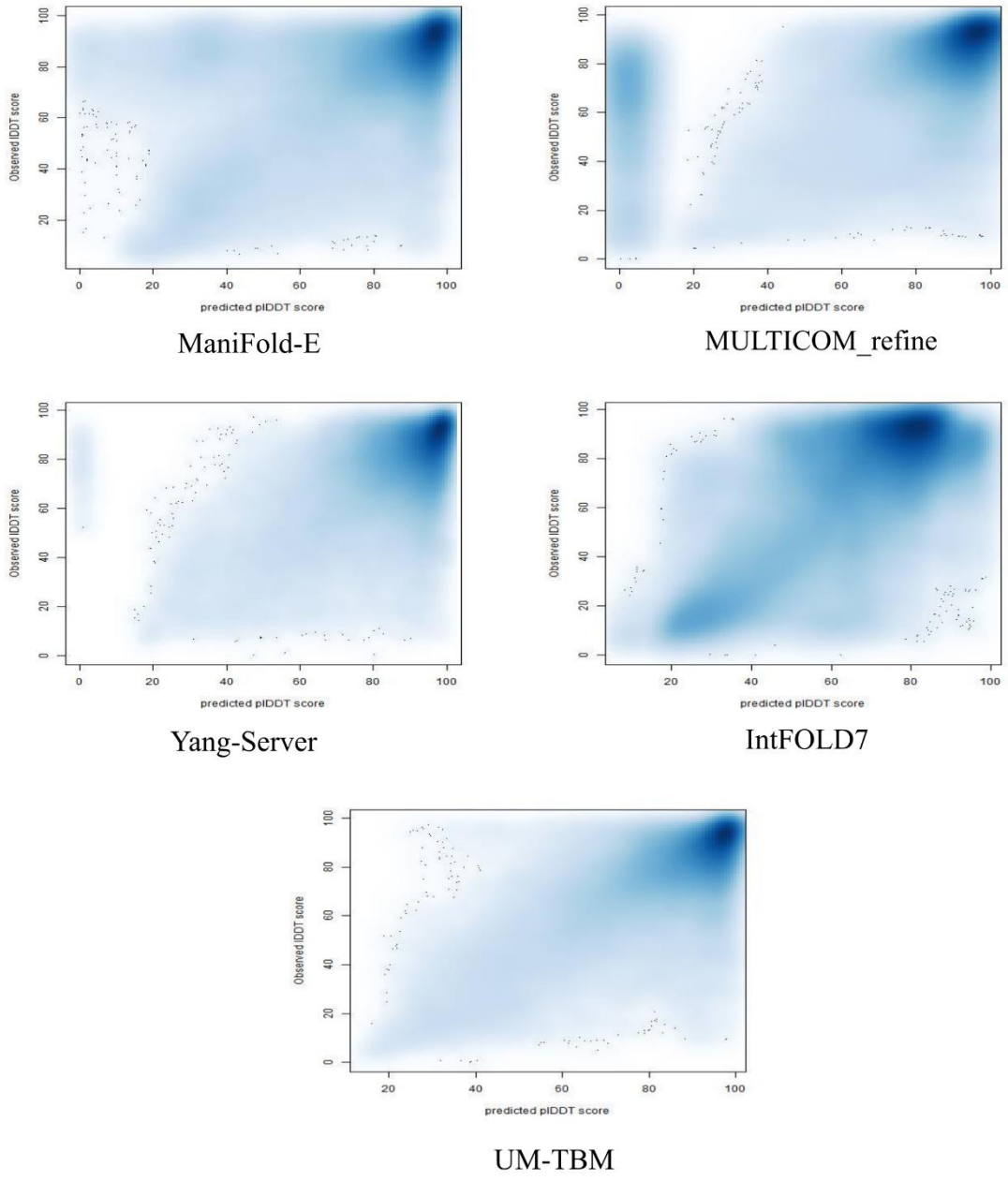


Figure 5.8. Density scatter plots show the relationship between the plDDT and lDDT scores for IntFOLD7 and the top four modelling servers regarding model quality at CASP15.

5.4.2.2 ModFOLD9 Performance on Models from Other Groups

ModFOLD9 is an independent quality assessment method which is designed to assess the quality of tertiary structure models produced by *any* 3D modelling method or server. During the CASP15 experiment, three alternative groups made their models publicly available to the community so we could evaluate their quality using ModFOLD9 before each target expired. These groups were the Elofsson group (af2-standard), the Baker group (BAKER-SERVER) and the Colabfold group (LocalcolabFold). Table 5.4 presents the evaluation results for three groups models on 26 CASP15 tertiary structure targets. The data show that the ModFOLD9 plDDT scores strongly correlate with the observed lDDT scores when models are evaluated from each different group. This demonstrates the ability of ModFOLD9 to accurately estimate the quality of models regardless of their source. In other words, ModFOLD9 should be able to provide a fair comparison of models from different sources without being biased by the modelling approach used. Additionally, ModFOLD9 uses different scoring methods, which could cover different aspects of models, which leads to a more orthogonally comprehensive assessment. Of course, one of the modelling aspects considered in ModFOLD9 is the assessment of the contact prediction between the model's residues using consensus CDA scores derived from a set of deep-learning contact prediction methods, which enhanced its local assessment performance.

Table 5.4. The evaluation analysis for ModFOLD9 local quality assessment of CASP15 models for three modelling groups. Pearson's R and Spearman's Rho correlation coefficients measure the relationship between the predicted models and native structures based on plDDT scores. The table is sorted by Pearson's R values.

Group	Pearson's R	Spearman's Rho	ROC AUR	ROC AUC FPR =<0.1
LocalcolabFold	0.86	0.63	0.95	0.055
BAKER-SERVER	0.85	0.66	0.95	0.055
af2-standard	0.84	0.64	0.94	0.054

5.4.2.3 ModFOLDdockS Prediction Performance

The ModFOLDdockS server used a consensus of various scoring methods to assess the quality of multimeric protein models, one of which is the CDA score derived from contact prediction. By incorporating the predicted inter-residue contact accuracy assessment into the model using the CDA score, the system can better evaluate the quality of the interacting residues within protein complexes. Therefore, the inclusion of contact prediction can aid in estimating the reliability and accuracy of multimeric models.

To demonstrate how contact prediction helps to improve the predictive assessment of ModFOLDdockS, the official assessment results of CASP15 data on individual residue confidence scores of QA methods are presented in Tables 5.5 and 5.6. The tables show the correlation and ROC analysis for two local scores, CAD and lDDT. The results demonstrated the accuracy of the per-residue assessment of the ModFOLDdockS server, where it ranked as the second top method based on the correlation and ROC AUC of CAD. In agreement with these results, evaluating the lDDT score revealed that ModFOLDdockS ranked as the third top-performing method among other QA methods.

Table 5.5. The official CASP15 assessment results of quality estimation methods for modelled protein complexes. The evaluation metrics are Pearson's R (Pears_LDDT) and Spearman's Rho (Spear_LDDT) correlations and ROC (AUC_LDDT) analysis for LDDT scores. The table is derived from https://predictioncenter.org/casp15/qa_local.cgi.

Group	GR#	Pears_LDDT	Spear_LDDT	AUC_LDDT
GuijunLab-RocketX	89	0.564	0.535	0.755
ModFOLDdockR	266	0.476	0.433	0.681
ModFOLDdockS	83	0.455	0.416	0.674
VoroIF	121	0.333	0.339	0.664
Venclovas	494	0.332	0.338	0.664
FoldEver	245	0.277	0.279	0.625
ModFOLDdock	41	0.243	0.227	0.584
APOLLO	168	0.192	0.213	0.565
Manifold	248	0.180	0.176	0.542
MULTICOM_deep	158	0.091	0.094	0.538
DLA-Ranker	101	0.100	0.112	0.529
MASS	468	0.151	0.172	0.527
LAW	426	0.169	0.168	0.525

Table 5.6. The official CASP15 assessment results of quality estimation methods for modelled protein complexes. The evaluation metrics are Pearson's R (Pears_CAD) and Spearman's Rho (Spear_CAD) correlations and ROC (AUC_CAD) analysis for CAD scores. The table is derived from https://predictioncenter.org/casp15/qa_local.cgi.

Group	GR#	Pears_CAD	Spear_CAD	AUC_CAD
GuijunLab-RocketX	89	0.505	0.456	0.714
ModFOLDdockS	83	0.420	0.379	0.660
ModFOLDdockR	266	0.411	0.369	0.651
VoroIF	121	0.272	0.271	0.619
Venclovas	494	0.271	0.271	0.619
FoldEver	245	0.217	0.194	0.583
ModFOLDdock	41	0.209	0.200	0.572
APOLLO	168	0.156	0.159	0.549
MULTICOM_deep	158	0.082	0.091	0.534
Manifold	248	0.153	0.149	0.529
DLA-Ranker	101	0.093	0.095	0.526
MASS	468	0.141	0.152	0.521
LAW	426	0.143	0.133	0.513

5.5 Conclusion

Contact prediction has had an impact on the prediction performance of our servers in different aspects. It helped to enhance the accuracy of the ModFOLD9 local quality estimations for tertiary structure models from any source, the self-estimation performance of IntFOLD7 and the performance of the ModFOLDdockS interface residue quality estimates for quaternary structure models. We tested our servers using the two "gold standard" independent blind testing experiments, CAMEO and CASP15. The CAMEO results demonstrated the improved accuracy of the ModFOLD9 local quality assessment. Using the CASP15 data, we evaluated how accurately ModFOLD9 estimated the prediction performance of IntFOLD7 and other groups. We also presented the CASP15 official assessment in two categories: regular modelling and EMA (Estimation of Model Accuracy) to demonstrate the performance of IntFOLD7 and ModFOLDdockS, respectively. Thus, it is essential to note that the contact prediction-based methods developed in the previous chapters are integral components of our prediction tools, contributing to three key areas: local model quality estimates for tertiary structure predictions, accuracy self-estimates for modelling tertiary structures and model quality estimation for predicted protein complexes.

Chapter 6 Synopsis of Thesis and Future Work

6.1 Synopsis of thesis

6.1.1 Consensus-based Approaches to Improving Deep Learning-based Contact Prediction Methods

The consensus approach showed promising potential for boosting the accuracy of protein structure prediction tools. Our study tested the benefits of the consensus approach for improving contact prediction accuracy. We chose the consensus approach because it provides confident results, reduces errors in prediction data, and obtains accurate outcomes by combining the strengths of various methods. Such approaches work well because there are often many ways of being wrong but fewer ways of being correct, so by combining methods, we are more likely to find the correct solution. As a simple example, suppose one tool predicted an incorrect contact between two residues while the other tools did not predict that they were in contact. In that case, the consensus prediction gives the majority agreement of results between the tools, decreasing the overall FPR. As such, consensus-based approaches can help enhance prediction accuracy beyond individual deep learning-based contact predictions.

Deep learning-based contact prediction methods use different approaches, often leading to variation in the contact map predictions for each protein target. To increase our confidence in contact prediction, in Chapter 2, we designed two consensus approaches simply using the mean scores to combine the top-performing contact prediction methods in CASP13 and CASP14. The first approach was to average the contact scores predicted by two of the three contact prediction methods. This approach produced three consensus methods: Consensus A, B, and C. The second approach was to compute the mean score of three contact prediction methods, producing the Cons3 method. The predictions from the consensus-based methods were compared with the individual methods using CASP13 evaluation metrics: precision, recall, f1-score, and PR curve analysis. Additionally, the ConEVA tool was used to further analyse the consensus-based methods.

The contact prediction accuracy of consensus-based methods varied depending on the individual methods combined. For instance, when TripletRes (Li *et al.*, 2021a) and trRosetta (Yang *et al.*, 2020) were combined in the ConsA approach, the accuracy of L/5 long-range contacts for FM domains increased by 3.2 % of the mean precision. Furthermore, a significant improvement was achieved by combining the three CASP13 top-performing methods in Cons3, increasing the accuracy of L/5 long-range contacts for FM domains by 10.5 % of the mean precision. The consensus prediction accuracy also achieved 77 % on both full chains and their domains in CASP14. However, other consensus-based contact prediction methods, such as Consensus C, have highlighted a drawback of using the simple mean score in consensus-based contact prediction. The Consensus C approach resulted in a reduction in accuracy when the predictions of two individual methods, which were trRosetta and DeepDist2 (Guo *et al.*, 2021), were combined. This issue may concern selecting similar deep learning-based contact prediction methods for consensus. In other words, if the individual methods predicted similarly inaccurate contact data, combining their predictions may conflate the false positives rather than reduce them. Thus, choosing accurate and orthogonal methods is essential when considering consensus approaches. Overall, our simple consensus approach has been shown to enhance contact prediction accuracy, which we tested using two different datasets (CASP13 and CASP14) with eight combinations of 6 alternative individual methods.

6.1.2 Developing a consensus of Contact Distance Agreement (CDA) Scores for Estimating Local Model Quality

Following the recent advances in protein tertiary structure prediction, Chapter 3 focused on the application of consensus deep learning-based contact prediction methods for the improvement of local model quality assessment. With the emergence of methods, such as AphaFold2, which can predict 3D models that are much closer to the native structures, the identification of the local errors in such high-accuracy models represented a new challenge for QE methods. The earlier versions of ModFOLD have previously integrated contact prediction methods, boosting the performance of local assessment. The ModFOLD6 (Maghrabi and McGuffin, 2017; Elofsson *et al.,* 2018), ModFOLD7 (Cheng *et al.,* 2019; Maghrabi and McGuffin, 2020) and ModFOLD8 (McGuffin *et al.,* 2021) methods were among the top-performing quality estimation methods in CASP12, CASP13 and CASP14 respectively.

In this study, we investigated the use of the six deep learning-based contact prediction methods: TripletRes (Li *et al.,* 2021a), trRosetta2 (Anishchenko *et al.,* 2021) and DeepDist (Wu *et al.,* 2021), DeepMetaPSICOV (Kandathil, Greener and Jones, 2019), SPOT-Contact (Hanson *et al.,* 2018), and MetaPSICOV (Jones *et al.,* 2015), to develop a consensus CDA score for each residue in a model. These methods formed the basis for six individual CDA scores to predict local quality scores. Unlike the previous study, we employed an MLP neural network to combine the individual CDA scores in this chapter. Using the MLP neural network allows us to establish the optimal weightings for combining the multiple scores in order to improve predictive performance. Two versions of the MLP were designed to combine the six CDA scores, each trained to predict one of two target functions, the S-score and the lDDT scores. The tuning of MLP hyper-parameters was conducted in this study in order to optimise ModFOLD9 performance. The MLP versions were trained and initially cross-validated using the CASP14 dataset. To gauge the effectiveness of the Consensus CDA approach, we conducted both correlation and ROC analysis to evaluate the performance of local model quality

assessment against each of the component CDA methods and VoroMQA, a leading pure-single model quality assessment method.

The consensus approach increased the local assessment accuracy compared to individual scoring methods. The correlation scores showed that the relationship between the predicted lDDT and the observed lDDT scores reached above 0.60. Additionally, ROC analysis indicated the consensus approach as having the best performance in terms of local model quality estimates according to the lDDT ROC AUC score (~0.80), outperforming the individual methods and VoroMQA. These findings demonstrate the improvement in local assessment performance gained by integrating deep learning-based contact prediction methods. It is worth mentioning that the fine-tuning MLP enhanced the performance in predicting the local quality scores. Hence, the MLP is an essential step in implementing the consensus algorithm to incorporate contact prediction information and raise the accuracy of the local quality assessment for integration with ModFOLD9.

6.1.3 Developing Consensus Quality Assessment Methods for ModFOLD9

Developing the consensus CDA score improved the local assessment accuracy for integration with ModFOLD9. This success highlighted the usefulness of consensus algorithms in enhancing the model quality estimation accuracy. Here, the study focused on further development in the local quality estimation of ModFOLD9 by integrating quality scores from additional alternative methods. These methods employed different approaches to score specific aspects of the model. Hence, using the quality methods individually could favour different aspects of 3D model quality and lead to skewed performance or biased predicted scores. Combining scores in ModFOLD9 allows for a more orthogonal assessment of features of the 3D model, making for a more balanced, consistent, and comprehensive score.

In Chapter 4, we introduced two types of quality scoring methods to develop our consensus scores: the pure-single and quasi-single model quality scoring methods. Our experiment was performed in two stages. The first stage was to combine the six CDA scores with quality scores derived from nine pure-single model methods, generating a total of 15 pure-single model quality scores. The second stage was to merge the 15 pure single model quality scores with four quasi-single model scores. In each step, again, we used two versions of MLP to combine the scores and trained them to predict either the S-score or the lDDT score. To optimise the MLP predictive performance, we implemented fine-tuning in the two stages. We assessed the consensus quality scores' performance in improving the accuracy of ModFOLD9 local quality assessment against the established methods, using a similar evaluation to the one used in Chapter 3.

The consensus of different quality scoring methods boosted the accuracy of ModFOLD9's local model quality assessment. In comparison with the consensus CDA approach result for lDDT score (correlation scores = 0.61), the finding shows that the improvement in the first stage of the combination increased the correlation scores by approximately 16 %, bringing it above 0.76. In contrast, the second combination stage increased them by at least 19 % to 0.80. In addition, the ROC analysis revealed that the accuracy of local assessment based on lDDT score in two stages was improved by more than 8 % and 9 % (ROC AUC for ModFOLD9_pure = 0.876, ROC AUC score for ModFOLD9_quasi = 0.891) compared with the previous Consensus CDA approach (lDDT ROC AUC score = 0.794). Again, optimising the MLP hyper-parameters enhanced the training, increasing the predictive performance when using the mode of extensive input data. ModFOLD9 outperformed all individual methods based on evaluation scores, underlining the effectiveness of the consensus approaches in increasing predictive assessment accuracy.

6.1.4 ModFOLD9 and ModFOLDdock Performance Benchmarking during the CASP15 Experiment and using the CAMEO Resource

The consensus algorithm and contact prediction data were exploited to improve our servers' predictive capabilities, which were benchmarked using the CAMEO resource and during the CASP15 experiment. In Chapter 5, we analysed the improved servers' performance using the data obtained from these two independent blind tests. We conducted the evaluation analyses using these datasets to identify the extent of the improvement gained due to these enhancements. Specifically, we observed enhanced performance in our servers, including IntFOLD7 server accuracy self-estimates, the ModFOLD9 local model quality assessment, and our new ModFOLDdockS method for interface residue accuracy scoring in multimeric models.

The CAMEO data findings revealed that the local assessment accuracy for ModFOLD9 outperformed those for established individual methods as well as the previous versions of ModFOLD. Furthermore, ModFOLD9 ranked as one of the leading QA methods overall according to lDDT score.

ModFOLD9 was used to generate the accuracy self-estimates for the IntFOLD7 models in CASP15. Analysing the official scores for IntFOLD7's models showed that ModFOLD9 successfully predicted the local errors in regular targets. Furthermore, the CASP15 official evaluation demonstrated that InFOLD7 achieved high performance in predicting the errors in its models for interdomain targets. Such findings affirmed the positive contribution of ModFOLD9 in enhancing the predictive accuracy of IntFOLD7. In addition, ModFOLD9 accurately assessed other groups' models for the CASP15 target, assuring it can be trusted as an independent quality estimation method for use with models produced by any state-of-the-art pipeline. Finally, the ModFOLDdockS method, which integrates the CDA score, achieved the highest global lDDT and CAD scores overall based on the official CASP15 assessment.

6.2 Conclusions

The objective of this study was to investigate the use of consensus contact prediction for enhancing protein structure prediction tools and improving the accuracy of 3D models. Initially, the study aimed to enhance the accuracy of deep learning-based contact prediction methods through consensus approaches in Chapter 2. We achieved improved predictive accuracy of contact prediction using the simple mean score consensus approach. However, we demonstrated that this simple approach was sub-optimal and could lead to decreased accuracy in some instances. Therefore, we applied a more advanced algorithm using neural networks to implement an optimal weighting of consensus contact prediction scores in the form of CDA scores, which could be used in our ModFOLD9 model quality estimation method. This more advanced consensus approach improved the local model quality assessment accuracy, as described in Chapter 3. To improve the accuracy further, we explored the usefulness of consensus approaches using neural networks by adding various model scoring quality methods along with the consensus of CDA scores. This approach increased the local assessment accuracy, leading to higher performance for ModFOLD9, as outlined in Chapter 4. The success of ModFOLD9 encouraged us to use it for our IntFOLD7 model accuracy self-estimates in CASP15. Furthermore, we applied a CDA score based on contact prediction in our ModFOLDdockS scoring pipeline to estimate the accuracy of interface residues in modelled protein complexes. These advancements were independently blind tested using CASP15 and CAMEO data, which are the gold-standard benchmarks of the field (see Chapter 5).

6.3 Future Directions

While discussing the effectiveness of contact prediction and consensus algorithms in enhancing the predictive performance of our servers, our study has highlighted the significant role of the consensus approach using neural networks (multiple-layer perceptron) in ModFOLD9, which integrates independent scoring methods for assessing the quality of 3D models of proteins. Our results show that neural networks can capture the complex relationship of different independent model quality assessment measures, leading to a more reliable overall model quality score. Following the success of the NN approach used in ModFOLD9, we plan to apply a similar multiple-layer perceptron in the related context of assessing the quality of interface residues in quaternary structure models rather than the simple mean score approach currently used in the ModFOLDdock variants. We expect this approach will lead to improved prediction performance.

The development of accurate contact prediction methods has impacted the predictive performance of protein structure prediction techniques, as shown in our study. Future research should be conducted to gauge the impact of protein contact prediction in enhancing protein *function* prediction methods, such as protein ligand-binding site modelling methods. One of our projects, FunFOLD, was designed based on the hypothesis that if two proteins have a similar structure, they will have similar functions and binding sites. Based on this premise, FunFOLD is a structural-based method intended to predict the ligand-binding site in a 3D model. The prediction process in FunFOLD starts by measuring the similarity between the 3D model and template structures with known binding sites using the TMalign method. If the model and templates have a similar structure according to TMalign, then FunFOLD will predict the same binding sites in the model based on those of the templates (Roche, Tetchner and McGuffin, 2011; Roche, Buenavista and McGuffin, 2013). Integrating binding site contact

prediction data could help FunFOLD to gain a more detailed view of the spatial arrangement of ligand binding residues in a 3D model, assisting in both model selection and the comparison with equivalent sites in known structures. Therefore, binding site contact information may help to boost the predictive accuracy of future versions of FunFOLD.

References

References

Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A. and Arshad, H. (2018) 'State-of-the-art in artificial neural network applications: A survey', *Heliyon*, 4(11), p. e00938. Available at: https://doi.org/10.1016/j.heliyon.2018.e00938.

Adhikari, B. (2020) 'DEEPCON: protein contact prediction using dilated convolutional neural networks with dropout', *Bioinformatics*, 36(2), pp. 470–477. Available at: https://doi.org/10.1093/bioinformatics/btz593.

Adhikari, B. and Cheng, J. (2016) 'Protein Residue Contacts and Prediction Methods', in O. Carugo and F. Eisenhaber (eds) *Data Mining Techniques for the Life Sciences. Methods in Molecular Biology*. New York, NY: Springer New York, pp. 463–476. Available at: https://doi.org/10.1007/978-1-4939-3572-7_24.

Adhikari, B., Hou, J. and Cheng, J. (2018) 'DNCON2: improved protein contact prediction using two-level deep convolutional neural networks', *Bioinformatics*, 34(9), pp. 1466–1472. Available at: https://doi.org/10.1093/bioinformatics/btx781.

Adhikari, B., Nowotny, J., Bhattacharya, D., Hou, J. and Cheng, J. (2016) 'ConEVA: a toolbox for comprehensive assessment of protein contacts', *BMC Bioinformatics*, 17, p. 517. Available at: https://doi.org/10.1186/s12859-016-1404-z.

Adiyaman, R. and McGuffin, L.J. (2019) 'Methods for the Refinement of Protein Structure 3D Models', *International Journal of Molecular Sciences*, 20(9), p. 2301. Available at: https://doi.org/10.3390/ijms20092301.

Adiyaman, R. and McGuffin, L.J. (2021) 'ReFOLD3: refinement of 3D protein models with gradual restraints based on predicted local quality and residue contacts', *Nucleic Acids Research*, 49(W1), pp. W589–W596. Available at: https://doi.org/10.1093/nar/gkab300.

Akdel, M., Pires, D.E.V., Pardo, E.P., Jänes, J., Zalevsky, A.O., Mészáros, B., Bryant, P., Good, L.L., Laskowski, R.A., Pozzati, G., Shenoy, A., Zhu, W., Kundrotas, P., Serra, V.R., Rodrigues, C.H.M., Dunham, A.S., Burke, D., Borkakoti, N., Velankar, S., Frost, A., Basquin, J., Lindorff-Larsen, K., Bateman, A., Kajava, A.V., Valencia, A., Ovchinnikov, S., Durairaj, J., Ascher, D.B., Thornton, J.M., Davey, N.E., Stein, A., Elofsson, A., Croll, T.I. and Beltrao, P. (2022) 'A structural biology community assessment of AlphaFold2 applications', *Nature Structural & Molecular Biology*, 29(11), pp. 1056–1067. Available at: https://doi.org/10.1038/s41594-022-00849-w.

Alharbi, S.M.A. and McGuffin, L.J. (2023) 'Machine Learning Methods for Predicting Protein Contacts', in L. Kurgan, *Machine Learning in Bioinformatics of Protein Sequences*. WORLD SCIENTIFIC, pp. 155–181. Available at: https://doi.org/10.1142/9789811258589_0006.

AlQuraishi, M. (2019) 'End-to-End Differentiable Learning of Protein Structure', *Cell Systems*, 8(4), pp. 292-301.e3. Available at: https://doi.org/10.1016/j.cels.2019.03.006.

Anderegg, M.A., Gyimesi, G., Ho, T.M., Hediger, M.A. and Fuster, D.G. (2022) 'The Less

Well-Known Little Brothers: The SLC9B/NHA Sodium Proton Exchanger Subfamily—Structure, Function, Regulation and Potential Drug-Target Approaches', *Frontiers in Physiology*, 13, p. 898508. Available at: https://doi.org/10.3389/fphys.2022.898508.

Anishchenko, I., Baek, M., Park, H., Hiranuma, N., Kim, D.E., Dauparas, J., Mansoor, S., Humphreys, I.R. and Baker, D. (2021) 'Protein tertiary structure prediction and refinement using deep learning and Rosetta in CASP14', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1722–1733. Available at: https://doi.org/10.1002/prot.26194.

Awad, M. and Khanna, R. (2015) *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkeley, CA: Apress. Available at: https://doi.org/10.1007/978-1-4302-5990-9.

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G.R., Wang, J., Cong, Q., Kinch, L.N., Schaeffer, R.D., Millán, C., Park, H., Adams, C., Glassman, C.R., DeGiovanni, A., Pereira, J.H., Rodrigues, A.V., Van Dijk, A.A., Ebrecht, A.C., Opperman, D.J., Sagmeister, T., Buhlheller, C., Pavkov-Keller, T., Rathinaswamy, M.K., Dalwadi, U., Yip, C.K., Burke, J.E., Garcia, K.C., Grishin, N.V., Adams, P.D., Read, R.J. and Baker, D. (2021) 'Accurate prediction of protein structures and interactions using a three-track neural network', *Science*, 373(6557), pp. 871–876. Available at: https://doi.org/10.1126/science.abj8754.

Basu, S. and Wallner, B. (2016) 'DockQ: A Quality Measure for Protein-Protein Docking Models', *PLOS ONE*, 11(8), p. e0161879. Available at: https://doi.org/10.1371/journal.pone.0161879.

Ben-David, M., Noivirt-Brik, O., Paz, A., Prilusky, J., Sussman, J.L. and Levy, Y. (2009) 'Assessment of CASP8 structure predictions for template free targets', *Proteins: Structure, Function, and Bioinformatics*, 77(S9), pp. 50–65. Available at: https://doi.org/10.1002/prot.22591.

Bengio, Y. (2012) 'Practical Recommendations for Gradient-Based Training of Deep Architectures', in G. Montavon, G.B. Orr, and K.-R. Müller (eds) *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 437–478. Available at: https://doi.org/10.1007/978-3-642-35289-8_26.

Benkert, P., Biasini, M. and Schwede, T. (2011) 'Toward the estimation of the absolute quality of individual protein structure models', *Bioinformatics*, 27(3), pp. 343–350. Available at: https://doi.org/10.1093/bioinformatics/btq662.

Bertoline, L.M.F., Lima, A.N., Krieger, J.E. and Teixeira, S.K. (2023) 'Before and after AlphaFold2: An overview of protein structure prediction', *Frontiers in Bioinformatics*, 3, p. 1120370. Available at: https://doi.org/10.3389/fbinf.2023.1120370.

Bertoni, M., Kiefer, F., Biasini, M., Bordoli, L. and Schwede, T. (2017) 'Modeling protein quaternary structure of homo- and hetero-oligomers beyond binary interactions by homology', *Scientific Reports*, 7(1), p. 10480. Available at: https://doi.org/10.1038/s41598-017-09654-8.

Bhojwani, H.R. and Joshi, U.J. (2022) 'Homology Modelling, Docking-based Virtual Screening, ADME Properties, and Molecular Dynamics Simulation for Identification of

Probable Type II Inhibitors of AXL Kinase', *Letters in Drug Design & Discovery*, 19(3), pp. 214–241. Available at: https://doi.org/10.2174/1570180818666211004102043.

Biasini, M., Schmidt, T., Bienert, S., Mariani, V., Studer, G., Haas, J., Johner, N., Schenk, A.D., Philippsen, A. and Schwede, T. (2013) 'OpenStructure : an integrated software framework for computational structural biology', *Acta Crystallographica Section D Biological Crystallography*, 69(5), pp. 701–709. Available at: https://doi.org/10.1107/S0907444913007051.

Billings, W.M., Morris, C.J. and Della Corte, D. (2021) 'The whole is greater than its parts: ensembling improves protein contact prediction', *Scientific Reports*, 11(1), p. 8039. Available at: https://doi.org/10.1038/s41598-021-87524-0.

Björkholm, P., Daniluk, P., Kryshtafovych, A., Fidelis, K., Andersson, R. and Hvidsten, T.R. (2009) 'Using multi-data hidden Markov models trained on local neighborhoods of protein structure to predict residue–residue contacts', *Bioinformatics*, 25(10), pp. 1264–1270. Available at: https://doi.org/10.1093/bioinformatics/btp149.

Bolboaca, S.-D. and Jantschi, L. (2006) 'Pearson versus Spearman, Kendall's Tau Correlation Analysis on Structure-Activity Relationships of Biologic Active Compounds', *Leonardo Journal of Sciences*, 5(9), pp. 179–200.

Bonetta, R. and Valentino, G. (2020) 'Machine learning techniques for protein function prediction', *Proteins: Structure, Function, and Bioinformatics*, 88(3), pp. 397–413. Available at: https://doi.org/10.1002/prot.25832.

Bonneau, R., Tsai, J., Ruczinski, I., Chivian, D., Rohl, C., Strauss, C.E.M. and Baker, D. (2001) 'Rosetta in CASP4: Progress in ab initio protein structure prediction', *Proteins: Structure, Function, and Genetics*, 45(S5), pp. 119–126. Available at: https://doi.org/10.1002/prot.1170.

Breda, A., Valadares, N.F., de Souza, O.N. and Garratt, R.C. (2008) 'Protein Structure, Modelling and Applications', in A. Gruber, A. Durham, and C. Huynh (eds) *Bioinformatics in Tropical Disease Research: A Practical and Case-Study Approach [Internet]*. Bethesda (MD): National Center for Biotechnology Information (US). Available at: https://www.ncbi.nlm.nih.gov/books/NBK6824/.

Buchan, D.W.A. and Jones, D.T. (2017) 'EigenTHREADER: analogous protein fold recognition by efficient contact map threading', *Bioinformatics*, 33(17), pp. 2684–2690. Available at: https://doi.org/10.1093/bioinformatics/btx217.

Buchan, D.W.A. and Jones, D.T. (2018) 'Improved protein contact predictions with the MetaPSICOV2 server in CASP12', *Proteins: Structure, Function, and Bioinformatics*, 86, pp. 78–83. Available at: https://doi.org/10.1002/prot.25379.

Buchan, D.W.A., Minneci, F., Nugent, T.C.O., Bryson, K. and Jones, D.T. (2013) 'Scalable web services for the PSIPRED Protein Analysis Workbench', *Nucleic Acids Research*, 41(W1), pp. W349–W357. Available at: https://doi.org/10.1093/nar/gkt381.

Buenavista, M.T., Roche, D.B. and McGuffin, L.J. (2012) 'Improvement of 3D protein models using multiple templates guided by single-template model quality assessment', *Bioinformatics*, 28(14), pp. 1851–1857. Available at:

https://doi.org/10.1093/bioinformatics/bts292.

Bunescu, R., Ge, R., Kate, R.J., Marcotte, E.M., Mooney, R.J., Ramani, A.K. and Wong, Y.W. (2005) 'Comparative experiments on learning information extractors for proteins and their interactions', *Artificial Intelligence in Medicine*, 33(2), pp. 139–155. Available at: https://doi.org/10.1016/j.artmed.2004.07.016.

Cao, R., Adhikari, B., Bhattacharya, D., Sun, M., Hou, J. and Cheng, J. (2017) 'QAcon: single model quality assessment using protein structural and contact information with machine learning techniques', *Bioinformatics*, 33(4), pp. 586–588. Available at: https://doi.org/10.1093/bioinformatics/btw694.

Cao, R., Bhattacharya, D., Adhikari, B., Li, J. and Cheng, J. (2016) 'Massive integration of diverse protein quality assessment methods to improve template based modeling in CASP11', *Proteins: Structure, Function and Bioinformatics*, 84(S1), pp. 247–259. Available at: https://doi.org/10.1002/prot.24924.

Chasiotis, V., Nadi, F. and Filios, A. (2021) 'Evaluation of multilayer perceptron neural networks and adaptive neuro-fuzzy inference systems for the mass transfer modeling of Echium amoenum Fisch. & C. A. Mey', *Journal of the Science of Food and Agriculture*, 101(15), pp. 6514–6524. Available at: https://doi.org/10.1002/jsfa.11323.

Chatterjee, A., Saha, J. and Mukherjee, J. (2022) 'Clustering with multi-layered perceptron', *Pattern Recognition Letters*, 155, pp. 92–99. Available at: https://doi.org/10.1016/j.patrec.2022.02.009.

Chen, C., Chen, X., Morehead, A., Wu, T. and Cheng, J. (2023) '3D-equivariant graph neural networks for protein model quality assessment', *Bioinformatics*, 39(1), p. btad030. Available at: https://doi.org/10.1093/bioinformatics/btad030.

Chen, J. and Siu, S.W.I. (2020) 'Machine Learning Approaches for Quality Assessment of Protein Structures', *Biomolecules*, 10(4), p. 626. Available at: https://doi.org/10.3390/biom10040626.

Chen, P. and Li, J. (2010) 'Prediction of protein long-range contacts using an ensemble of genetic algorithm classifiers with sequence profile centers', *BMC Structural Biology*, 10(SUPPL. 1), pp. 1–13. Available at: https://doi.org/10.1186/1472-6807-10-S1-S2.

Chen, X., Liu, J., Guo, Z., Wu, T., Hou, J. and Cheng, J. (2021) 'Protein model accuracy estimation empowered by deep learning and inter-residue distance prediction in CASP14', *Scientific Reports*, 11(1), p. 10943. Available at: https://doi.org/10.1038/s41598-021-90303-6.

Cheng, J. and Baldi, P. (2007) 'Improved residue contact prediction using support vector machines and a large feature set', *BMC Bioinformatics*, 8(1), p. 113. Available at: https://doi.org/10.1186/1471-2105-8-113.

Cheng, J., Choe, M., Elofsson, A., Han, K., Hou, J., Maghrabi, A.H.A., McGuffin, L.J., Menéndez-Hurtado, D., Olechnovič, K., Schwede, T., Studer, G., Uziela, K., Venclovas, Č. and Wallner, B. (2019) 'Estimation of model accuracy in CASP13', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1361–1377. Available at: https://doi.org/10.1002/prot.25767.

Covell, D.G. and Jernigan, R.L. (1990) ‘Conformations of folded proteins in restricted spaces’, *Biochemistry*, 29(13), pp. 3287–3294. Available at: https://doi.org/10.1021/bi00465a020.

Davis, J. and Goadrich, M. (2006) ‘The relationship between Precision-Recall and ROC curves’, in *Proceedings of the 23rd international conference on Machine learning - ICML ’06*. *the 23rd international conference*, Pittsburgh, Pennsylvania: ACM Press, pp. 233–240. Available at: https://doi.org/10.1145/1143844.1143874.

De Juan, D., Pazos, F. and Valencia, A. (2013) ‘Emerging methods in protein co-evolution’, *Nature Reviews Genetics*, 14(4), pp. 249–261. Available at: https://doi.org/10.1038/nrg3414.

Dhingra, S., Sowdhamini, R., Cadet, F. and Offmann, B. (2020) ‘A glance into the evolution of template-free protein structure prediction methodologies’, *Biochimie*, 175, pp. 85–92. Available at: https://doi.org/10.1016/j.biochi.2020.04.026.

Di Lena, P., Nagata, K. and Baldi, P. (2012) ‘Deep architectures for protein contact map prediction’, *Bioinformatics*, 28(19), pp. 2449–2457. Available at: https://doi.org/10.1093/bioinformatics/bts475.

Ding, W., Mao, W., Shao, D., Zhang, W. and Gong, H. (2018) ‘DeepConPred2: An Improved Method for the Prediction of Protein Residue Contacts’, *Computational and Structural Biotechnology Journal*, 16, pp. 503–510. Available at: https://doi.org/10.1016/j.csbj.2018.10.009.

Dongare, A.D., Kharde, R.R. and Kachare, A.D. (2012) ‘Introduction to Artificial Neural Network’, *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1), pp. 189–193.

Du, Z., Peng, Z. and Yang, J. (2022) ‘Toward the assessment of predicted inter-residue distance’, *Bioinformatics*, 38(4), pp. 962–969. Available at: https://doi.org/10.1093/bioinformatics/btab781.

Duarte, J.M., Sathyapriya, R., Stehr, H., Filippis, I. and Lappe, M. (2010) ‘Optimal contact definition for reconstruction of Contact Maps’, *BMC Bioinformatics*, 11(1), p. 283. Available at: https://doi.org/10.1186/1471-2105-11-283.

Eastwood, M.P., Hardin, C., Luthey-Schulten, Z. and Wolynes, P.G. (2001) ‘Evaluating protein structure-prediction schemes using energy landscape theory’, *IBM Journal of Research and Development*, 45(3.4), pp. 475–497. Available at: https://doi.org/10.1147/rd.453.0475.

Edmunds, N.S., Alharbi, S.M.A., Genc, A.G., Adiyaman, R. and McGuffin, L.J. (2023) ‘Estimation of model accuracy in CASP15 using the ModFOLDdock server’, *Proteins: Structure, Function, and Bioinformatics*, 91(12), pp. 1871–1878. Available at: https://doi.org/10.1002/prot.26532.

Eickholt, J. and Cheng, J. (2012) ‘Predicting protein residue–residue contacts using deep networks and boosting’, *Bioinformatics*, 28(23), pp. 3066–3072. Available at: https://doi.org/10.1093/bioinformatics/bts598.

Eickholt, J. and Cheng, J. (2013) ‘A study and benchmark of DNcon: a method for protein

residue-residue contact prediction using deep networks', *BMC Bioinformatics*, 14(S14), p. S12. Available at: https://doi.org/10.1186/1471-2105-14-S14-S12.

Eisenberg, D. and McLachlan, A.D. (1986) 'Solvation energy in protein folding and binding', *Nature*, 319(6050), pp. 199–203. Available at: https://doi.org/10.1038/319199a0.

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M. and Aurell, E. (2013) 'Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models', *Physical Review E*, 87(1), p. 012707. Available at: https://doi.org/10.1103/PhysRevE.87.012707.

Elansari, T., Ouanan, M. and Bourray, H. (2023) 'Modeling of Multilayer Perceptron Neural Network Hyperparameter Optimization and Training', Preprint. Available at: https://doi.org/10.21203/rs.3.rs-2570112/v1.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D. and Rost, B. (2022) 'ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10), pp. 7112–7127. Available at: https://doi.org/10.1109/TPAMI.2021.3095381.

Elofsson, A. (2023) 'Progress at protein structure prediction, as seen in CASP15', *Current Opinion in Structural Biology*, 80, p. 102594. Available at: https://doi.org/10.1016/j.sbi.2023.102594.

Elofsson, A., Joo, K., Keasar, C., Lee, J., Maghrabi, A.H.A., Manavalan, B., McGuffin, L.J., Ménendez Hurtado, D., Mirabello, C., Pilstål, R., Sidi, T., Uziela, K. and Wallner, B. (2018) 'Methods for estimation of model accuracy in CASP12', *Proteins: Structure, Function, and Bioinformatics*, 86(S1), pp. 361–373. Available at: https://doi.org/10.1002/prot.25395.

El-Rashidy, N., Abdelrazik, S., Abuhmed, T., Amer, E., Ali, F., Hu, J.-W. and El-Sappagh, S. (2021) 'Comprehensive Survey of Using Machine Learning in the COVID-19 Pandemic', *Diagnostics*, 11(7), p. 1155. Available at: https://doi.org/10.3390/diagnostics11071155.

Emerson, I.A. and Amala, A. (2017) 'Protein contact maps: A binary depiction of protein 3D structures', *Physica A: Statistical Mechanics and its Applications*, 465, pp. 782–791. Available at: https://doi.org/10.1016/j.physa.2016.08.033.

Ezkurdia, I., Graña, O., Izarzugaza, J.M.G. and Tress, M.L. (2009) 'Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8', *Proteins: Structure, Function, and Bioinformatics*, 77(S9), pp. 196–209. Available at: https://doi.org/10.1002/prot.22554.

Fang, C., Jia, Y., Hu, L., Lu, Y. and Wang, H. (2020) 'IMPContact: An Interhelical Residue Contact Prediction Method', *BioMed Research International*, 2020, pp. 1–10. Available at: https://doi.org/10.1155/2020/4569037.

Farhadi, T. (2018) 'Advances in Protein Tertiary Structure Prediction', *Biomedical and Biotechnology Research Journal (BBRJ)*, 2(1), pp. 20–25. Available at: https://doi.org/10.4103/bbrj.bbrj_94_17.

Fariselli, P. and Casadio, R. (1999) 'A neural network based predictor of residue contacts in proteins', *Protein Engineering, Design and Selection*, 12(1), pp. 15–21. Available at:

https://doi.org/10.1093/protein/12.1.15.

Fariselli, P., Olmea, O., Valencia, A. and Casadio, R. (2001) 'Prediction of contact maps with neural networks and correlated mutations', *Protein Engineering, Design and Selection*, 14(11), pp. 835–843. Available at: https://doi.org/10.1093/protein/14.11.835.

Fathi, S., Sakhteman, A. and Solhjoo, A. (2023) 'Unveiling Attributes of Human 15-Lipoxygenase-1 as a Potential Candidate for Prostate Cancer Drug Development Using *in Silico* Approaches', *Journal of Computational Biophysics and Chemistry*, 22(01), pp. 99–111. Available at: https://doi.org/10.1142/S2737416523500060.

Fawcett, T. and Flach, P.A. (2005) 'A Response to Webb and Ting's On the Application of ROC Analysis to Predict Classification Performance Under Varying Class Distributions', *Machine Learning*, 58(1), pp. 33–38. Available at: https://doi.org/10.1007/s10994-005-5256-4.

Feig, M. (2017) 'Computational protein structure refinement: almost there, yet still so far to go', *WIREs Computational Molecular Science*, 7(3), p. e1307. Available at: https://doi.org/10.1002/wcms.1307.

Fischer, D. (2003) '3D-SHOTGUN: A novel, cooperative, fold-recognition meta-predictor', *Proteins: Structure, Function, and Genetics*, 51(3), pp. 434–441. Available at: https://doi.org/10.1002/prot.10357.

Fowler, N.J. and Williamson, M.P. (2022) 'The accuracy of protein structures in solution determined by AlphaFold and NMR', *Structure*, 30(7), pp. 925-933.e2. Available at: https://doi.org/10.1016/j.str.2022.04.005.

Fukuda, H. and Tomii, K. (2020) 'DeepECA: An end-to-end learning framework for protein contact prediction from a multiple sequence alignment', *BMC Bioinformatics*, 21(1), pp. 2–4. Available at: https://doi.org/10.1186/s12859-019-3190-x.

Gilson, M.K., Sharp, K.A. and Honig, B.H. (1988) 'Calculating the electrostatic potential of molecules in solution: Method and error assessment', *Journal of Computational Chemistry*, 9(4), pp. 327–335. Available at: https://doi.org/10.1002/jcc.540090407.

Glorot, X., Bordes, A. and Bengio, Y. (2011) 'Deep Sparse Rectifier Neural Networks', in G. Gordon, D. Dunson, and M. Dudík (eds) *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale, FL, USA: PMLR, pp. 315–323. Available at: https://proceedings.mlr.press/v15/glorot11a.html.

Goadrich, M., Oliphant, L. and Shavlik, J. (2004) 'Learning Ensembles of First-Order Clauses for Recall-Precision Curves: A Case Study in Biomedical Information Extraction', in R. Camacho, R. King, and A. Srinivasan (eds) *Inductive Logic Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 98–115. Available at: https://doi.org/10.1007/978-3-540-30109-7_11.

Göbel, U., Sander, C., Schneider, R. and Valencia, A. (1994) 'Correlated mutations and residue contacts in proteins', *Proteins: Structure, Function, and Bioinformatics*, 18(4), pp. 309–317. Available at: https://doi.org/10.1002/prot.340180402.

Gomes, M., Hamer, R., Reinert, G. and Deane, C.M. (2012) 'Mutual information and variants

for protein domain-domain contact prediction', *BMC Research Notes*, 5(1), p. 472. Available at: https://doi.org/10.1186/1756-0500-5-472.

Graves, A., Fernández, S. and Schmidhuber, J. (2007) 'Multi-dimensional Recurrent Neural Networks', in J.M. de Sá, L.A. Alexandre, W. Duch, and D. Mandic (eds) *Artificial Neural Networks – ICANN 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 549–558. Available at: https://doi.org/10.1007/978-3-540-74690-4_56.

Greener, J.G., Kandathil, S.M., Moffat, L. and Jones, D.T. (2022) 'A guide to machine learning for biologists', *Nature Reviews Molecular Cell Biology*, 23(1), pp. 40–55. Available at: https://doi.org/10.1038/s41580-021-00407-0.

Guo, S.-S., Liu, J., Zhou, X.-G. and Zhang, G.-J. (2022) 'DeepUMQA: ultrafast shape recognition-based protein model quality assessment using deep learning', *Bioinformatics*, 38(7), pp. 1895–1903. Available at: https://doi.org/10.1093/bioinformatics/btac056.

Guo, Z., Wu, T., Liu, J., Hou, J. and Cheng, J. (2021) 'Improving deep learning-based protein distance prediction in CASP14', *Bioinformatics*, 37(19), pp. 3190–3196. Available at: https://doi.org/10.1093/bioinformatics/btab355.

Haas, J., Barbato, A., Behringer, D., Studer, G., Roth, S., Bertoni, M., Mostaguir, K., Gumienny, R. and Schwede, T. (2018) 'Continuous Automated Model EvaluatiOn (CAMEO) complementing the critical assessment of structure prediction in CASP12', *Proteins: Structure, Function, and Bioinformatics*, 86(S1), pp. 387–398. Available at: https://doi.org/10.1002/prot.25431.

Haas, J., Gumienny, R., Barbato, A., Ackermann, F., Tauriello, G., Bertoni, M., Studer, G., Smolinski, A. and Schwede, T. (2019) 'Introducing "best single template" models as reference baseline for the Continuous Automated Model Evaluation (CAMEO)', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1378–1387. Available at: https://doi.org/10.1002/prot.25815.

Hagler, A.T., Huler, E. and Lifson, S. (1974) 'Energy functions for peptides and proteins. I. Derivation of a consistent force field including the hydrogen bond from amide crystals', *Journal of the American Chemical Society*, 96(17), pp. 5319–5327. Available at: https://doi.org/10.1021/ja00824a004.

Hagler, A.T. and Lifson, S. (1974) 'Energy functions for peptides and proteins. II. Amide hydrogen bond and calculation of amide crystal properties', *Journal of the American Chemical Society*, 96(17), pp. 5327–5335. Available at: https://doi.org/10.1021/ja00824a005.

Hansen, L.K. and Salamon, P. (1990) 'Neural network ensembles', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), pp. 993–1001. Available at: https://doi.org/10.1109/34.58871.

Hanson, J., Paliwal, K., Litfin, T., Yang, Y. and Zhou, Y. (2018) 'Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks', *Bioinformatics*, 34(23), pp. 4039–4045. Available at: https://doi.org/10.1093/bioinformatics/bty481.

Hapudeniya, M. (2010) 'Artificial Neural Networks in Bioinformatics', *Sri Lanka Journal of Bio-Medical Informatics*, 1(2), pp. 104–111. Available at:

https://doi.org/10.4038/sljbmi.v1i2.1719.

He, B., Mortuza, S.M., Wang, Y., Shen, H.-B. and Zhang, Y. (2017) 'NeBcon: protein contact map prediction using neural network training coupled with naïve Bayes classifiers', *Bioinformatics*, 33(15), pp. 2296–2306. Available at: https://doi.org/10.1093/bioinformatics/btx164.

He, H. and Garcia, E.A. (2009) 'Learning from Imbalanced Data', *IEEE Transactions on Knowledge and Data Engineering*, 21(9), pp. 1263–1284. Available at: https://doi.org/10.1109/TKDE.2008.239.

Heo, L. and Feig, M. (2020) 'Modeling of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) Proteins by Machine Learning and Physics-Based Refinement', *bioRxiv*, Preprint. Available at: https://doi.org/10.1101/2020.03.25.008904.

Hiranuma, N., Park, H., Baek, M., Anishchenko, I., Dauparas, J. and Baker, D. (2021) 'Improved protein structure refinement guided by deep learning based accuracy estimation', *Nature Communications*, 12(1), p. 1340. Available at: https://doi.org/10.1038/s41467-021-21511-x.

Ho, B.K. and Dill, K.A. (2006) 'Folding Very Short Peptides Using Molecular Dynamics', *PLoS Computational Biology*, 2(4), p. e27. Available at: https://doi.org/10.1371/journal.pcbi.0020027.

Horner, D.S., Pirovano, W. and Pesole, G. (2007) 'Correlated substitution analysis and the prediction of amino acid structural contacts', *Briefings in Bioinformatics*, 9(1), pp. 46–56. Available at: https://doi.org/10.1093/bib/bbm052.

Hou, J., Wu, T., Cao, R. and Cheng, J. (2019) 'Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1165–1178. Available at: https://doi.org/10.1002/prot.25697.

Huang, B., Kong, L., Wang, C., Ju, F., Zhang, Q., Zhu, J., Gong, T., Zhang, H., Yu, C., Zheng, W.-M. and Bu, D. (2023) 'Protein Structure Prediction: Challenges, Advances, and the Shift of Research Paradigms', *Genomics, Proteomics & Bioinformatics*, 21(5), pp. 913–925. Available at: https://doi.org/10.1016/j.gpb.2022.11.014.

Huang, Y.J., Mao, B., Aramini, J.M. and Montelione, G.T. (2014) 'Assessment of template-based protein structure predictions in CASP10', *Proteins: Structure, Function and Bioinformatics*, 82, pp. 43–56. Available at: https://doi.org/10.1002/prot.24488.

Hurtado, D.M., Uziela, K. and Elofsson, A. (2018) 'Deep transfer learning in the assessment of the quality of protein models', *arXiv: Biomolecules*, preprint. Available at: https://doi.org/10.48550/ARXIV.1804.06281.

Jagielska, A., Wroblewska, L. and Skolnick, J. (2008) 'Protein model refinement using an optimized physics-based all-atom force field', *Proceedings of the National Academy of Sciences*, 105(24), pp. 8268–8273. Available at: https://doi.org/10.1073/pnas.0800054105.

Jayaraj, V.A.J.P.B. (2021) 'Protein Structure Prediction : Conventional and Deep Learning Perspectives', *The Protein Journal*, 40(4), pp. 522–544. Available at: https://doi.org/10.1007/s10930-021-10003-y.

Jernigan, R.L. and Bahar, I. (1996) ‘Structure-derived potentials and protein simulations’, *Current Opinion in Structural Biology*, 6(2), pp. 195–209. Available at: https://doi.org/10.1016/S0959-440X(96)80075-3.

Jing, X., Dong, Qimin, Lu, R. and Dong, Qiwen (2019) ‘Protein Inter-Residue Contacts Prediction: Methods, Performances and Applications’, *Current Bioinformatics*, 14(3), pp. 178–189. Available at: https://doi.org/10.2174/1574893613666181109130430.

Jing, X. and Xu, J. (2020) ‘Improved protein model quality assessment by integrating sequential and pairwise features using deep learning’, *Bioinformatics*, 36(22–23), pp. 5361–5367. Available at: https://doi.org/10.1093/bioinformatics/btaa1037.

Jisna, V.A. and Jayaraj, P.B. (2021) ‘Protein Structure Prediction: Conventional and Deep Learning Perspectives’, *The Protein Journal*, 40(4), pp. 522–544. Available at: https://doi.org/10.1007/s10930-021-10003-y.

Jones, D.T. (2001) ‘Predicting novel protein folds by using FRAGFOLD’, *Proteins: Structure, Function, and Bioinformatics*, 45(S5), pp. 127–132. Available at: https://doi.org/10.1002/prot.1171.

Jones, D.T., Buchan, D.W.A., Cozzetto, D. and Pontil, M. (2012) ‘PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments’, *Bioinformatics*, 28(2), pp. 184–190. Available at: https://doi.org/10.1093/bioinformatics/btr638.

Jones, D.T. and Cozzetto, D. (2015) ‘DISOPRED3: precise disordered region predictions with annotated protein-binding activity’, *Bioinformatics*, 31(6), pp. 857–863. Available at: https://doi.org/10.1093/bioinformatics/btu744.

Jones, D.T. and Kandathil, S.M. (2018) ‘High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features’, *Bioinformatics*, 34(19), pp. 3308–3315. Available at: https://doi.org/10.1093/bioinformatics/bty341.

Jones, D.T., Singh, T., Kosciolek, T. and Tetchner, S. (2015) ‘MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins’, *Bioinformatics*, 31(7), pp. 999–1006. Available at: https://doi.org/10.1093/bioinformatics/btu791.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P. and Hassabis, D. (2021a) ‘Highly accurate protein structure prediction with AlphaFold’, *Nature*, 596(7873), pp. 583–589. Available at: https://doi.org/10.1038/s41586-021-03819-2.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M.,

Berghammer, T., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P. and Hassabis, D. (2021b) 'Applying and improving AlphaFold at CASP14', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1711–1721. Available at: https://doi.org/10.1002/prot.26257.

Kabsch, W. and Sander, C. (1983) 'Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features', *Biopolymers*, 22(12), pp. 2577–2637. Available at: https://doi.org/10.1002/bip.360221211.

Kaján, L., Hopf, T.A., Kalaš, M., Marks, D.S. and Rost, B. (2014) 'FreeContact: fast and free software for protein contact prediction from residue co-evolution', *BMC Bioinformatics*, 15(1), p. 85. Available at: https://doi.org/10.1186/1471-2105-15-85.

Kamisetty, H., Ovchinnikov, S. and Baker, D. (2013) 'Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era', *Proceedings of the National Academy of Sciences*, 110(39), pp. 15674–15679. Available at: https://doi.org/10.1073/pnas.1314045110.

Kandathil, S.M., Greener, J.G. and Jones, D.T. (2019) 'Prediction of interresidue contacts with DeepMetaPSICOV in CASP13', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1092–1099. Available at: https://doi.org/10.1002/prot.25779.

Karplus, K., Sjölander, K., Barrett, C., Cline, M., Haussler, D., Hughey, R., Holm, L. and Sander, C. (1997) 'Predicting protein structure using hidden Markov models', *Proteins: Structure, Function, and Genetics*, 29(S1), pp. 134–139. Available at: https://doi.org/10.1002/(SICI)1097-0134(1997)1+<134::AID-PROT18>3.0.CO;2-P.

KC, D.B. (2017) 'Recent advances in sequence-based protein structure prediction', *Briefings in Bioinformatics*, 18(6), pp. 1021–1032. Available at: https://doi.org/10.1093/bib/bbw070.

Kessel, A. and Ben-Tal, N. (2018) *Introduction to Proteins: Structure, Function, and Motion*. SECOND EDITION. Abingdon, UK: Chapman and Hall/CRC. Available at: https://doi.org/10.1201/9781315113876.

Kim, H. and Kihara, D. (2016) 'Protein structure prediction using residue- and fragment-environment potentials in CASP11', *Proteins: Structure, Function and Bioinformatics*, 84, pp. 105–117. Available at: https://doi.org/10.1002/prot.24920.

Kinch, L.N., Li, W., Monastyrskyy, B., Kryshtafovych, A. and Grishin, N.V. (2016) 'Assessment of CASP11 contact-assisted predictions', *Proteins: Structure, Function and Bioinformatics*, 84, pp. 164–180. Available at: https://doi.org/10.1002/prot.25020.

Kmiecik, S., Gront, D., Kolinski, M., Wieteska, L., Dawid, A.E. and Kolinski, A. (2016) 'Coarse-Grained Protein Models and Their Applications', *Chemical Reviews*, 116(14), pp. 7898–7936. Available at: https://doi.org/10.1021/acs.chemrev.6b00163.

Kok, S. and Domingos, P. (2005) 'Learning the structure of Markov logic networks', in *Proceedings of the 22nd international conference on Machine learning - ICML '05*. *the 22nd international conference*, Bonn, Germany: ACM Press, pp. 441–448. Available at: https://doi.org/10.1145/1102351.1102407.

Konopka, B.M., Ciombor, M., Kurczynska, M. and Kotulska, M. (2014) 'Automated

Procedure for Contact-Map-Based Protein Structure Reconstruction', *The Journal of Membrane Biology*, 247(5), pp. 409–420. Available at: https://doi.org/10.1007/s00232-014-9648-x.

Kryshtafovych, A., Antczak, M., Szachniuk, M., Zok, T., Kretsch, R.C., Rangan, R., Pham, P., Das, R., Robin, X., Studer, G., Durairaj, J., Eberhardt, J., Sweeney, A., Topf, M., Schwede, T., Fidelis, K. and Moult, J. (2023) 'New prediction categories in CASP15', *Proteins: Structure, Function, and Bioinformatics*, 91(12), pp. 1550–1557. Available at: https://doi.org/10.1002/prot.26515.

Kryshtafovych, A., Barbato, A., Fidelis, K., Monastyrskyy, B., Schwede, T. and Tramontano, A. (2014) 'Assessment of the assessment: Evaluation of the model quality estimates in CASP10', *Proteins: Structure, Function, and Bioinformatics*, 82(S2), pp. 112–126. Available at: https://doi.org/10.1002/prot.24347.

Kryshtafovych, A., Barbato, A., Monastyrskyy, B., Fidelis, K., Schwede, T. and Tramontano, A. (2016) 'Methods of model accuracy estimation can help selecting the best models from decoy sets: Assessment of model accuracy estimations in CASP11', *Proteins: Structure, Function, and Bioinformatics*, 84(S1), pp. 349–369. Available at: https://doi.org/10.1002/prot.24919.

Kryshtafovych, A., Fidelis, K. and Tramontano, A. (2011) 'Evaluation of model quality predictions in CASP9', *Proteins: Structure, Function, and Bioinformatics*, 79(S10), pp. 91–106. Available at: https://doi.org/10.1002/prot.23180.

Kryshtafovych, A., Monastyrskyy, B. and Fidelis, K. (2014) 'CASP prediction center infrastructure and evaluation measures in CASP10 and CASP ROLL', *Proteins: Structure, Function, and Bioinformatics*, 82(S2), pp. 7–13. Available at: https://doi.org/10.1002/prot.24399.

Kryshtafovych, A., Monastyrskyy, B. and Fidelis, K. (2016) 'CASP11 statistics and the prediction center evaluation system', *Proteins: Structure, Function, and Bioinformatics*, 84(S1), pp. 15–19. Available at: https://doi.org/10.1002/prot.25005.

Kryshtafovych, A., Moult, J., Billings, W.M., Della Corte, D., Fidelis, K., Kwon, S., Olechnovič, K., Seok, C., Venclovas, Č., Won, J., and CASP-COVID participants (2021) 'Modeling SARS-CoV-2 proteins in the CASP-commons experiment', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1987–1996. Available at: https://doi.org/10.1002/prot.26231.

Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. and Moult, J. (2019) 'Critical assessment of methods of protein structure prediction (CASP)—Round XIII', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1011–1020. Available at: https://doi.org/10.1002/prot.25823.

Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. and Moult, J. (2021) 'Critical assessment of methods of protein structure prediction (CASP)—Round XIV', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1607–1617. Available at: https://doi.org/10.1002/prot.26237.

Kuhlman, B. and Bradley, P. (2019) 'Advances in protein structure prediction and design',

Nature Reviews Molecular Cell Biology, 20(11), pp. 681–697. Available at: https://doi.org/10.1038/s41580-019-0163-x.

Kwon, S., Won, J., Kryshtafovych, A. and Seok, C. (2021) 'Assessment of protein model structure accuracy estimation in CASP14: Old and new challenges', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1940–1948. Available at: https://doi.org/10.1002/prot.26192.

Laine, E., Eismann, S., Elofsson, A. and Grudinin, S. (2021) 'Protein sequence-to-structure learning: Is this the end(-to-end revolution)?', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1770–1786. Available at: https://doi.org/10.1002/prot.26235.

Larsson, P., Skwark, M.J., Wallner, B. and Elofsson, A. (2009) 'Assessment of global and local model quality in CASP8 using Pcons and ProQ', *Proteins: Structure, Function, and Bioinformatics*, 77(S9), pp. 167–172. Available at: https://doi.org/10.1002/prot.22476.

Latek, D. and Kolinski, A. (2008) 'Contact prediction in protein modeling: Scoring, folding and refinement of coarse-grained models', *BMC Structural Biology*, 8(1), p. 36. Available at: https://doi.org/10.1186/1472-6807-8-36.

Lee, D., Xiong, D., Wierbowski, S., Li, L., Liang, S. and Yu, H. (2022) 'Deep learning methods for 3D structural proteome and interactome modeling', *Current Opinion in Structural Biology*, 73, p. 102329. Available at: https://doi.org/10.1016/j.sbi.2022.102329.

Lesk, A.M. (1997) 'CASP2: Report on *ab initio* predictions', *Proteins: Structure, Function, and Genetics*, 29(S1), pp. 151–166. Available at: https://doi.org/10.1002/(SICI)1097-0134(1997)1+<151::AID-PROT20>3.0.CO;2-M.

Levitt, M. (1976) 'A simplified representation of protein conformations for rapid simulation of protein folding', *Journal of Molecular Biology*, 104(1), pp. 59–107. Available at: https://doi.org/10.1016/0022-2836(76)90004-8.

Li, J. and Xu, J. (2021) 'Study of real-valued distance prediction for protein structure prediction with deep learning', *Bioinformatics*, 37(19), pp. 3197–3203. Available at: https://doi.org/10.1093/bioinformatics/btab333.

Li, W., Dustin Schaeffer, R., Otwinowski, Z. and Grishin, N.V. (2016) 'Estimation of uncertainties in the global distance test (GDT_TS) for CASP models', *PLoS ONE*, 11(5), pp. 1–16. Available at: https://doi.org/10.1371/journal.pone.0154786.

Li, Y., Fang, Y. and Fang, J. (2011) 'Predicting residue–residue contacts using random forest models', *Bioinformatics*, 27(24), pp. 3379–3384. Available at: https://doi.org/10.1093/bioinformatics/btr579.

Li, Y., Hu, J., Zhang, C., Yu, D.-J. and Zhang, Y. (2019) 'ResPRE: high-accuracy protein contact prediction by coupling precision matrix with deep residual neural networks', *Bioinformatics*, 35(22), pp. 4647–4655. Available at: https://doi.org/10.1093/bioinformatics/btz291.

Li, Y., Zhang, C., Bell, E.W., Yu, D. and Zhang, Y. (2019) 'Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1082–1091.

Available at: https://doi.org/10.1002/prot.25798.

Li, Y., Zhang, C., Bell, E.W., Zheng, W., Zhou, X., Yu, D.-J. and Zhang, Y. (2021a) 'Deducing high-accuracy protein contact-maps from a triplet of coevolutionary matrices through deep residual convolutional networks', *PLOS Computational Biology*, 17(3), p. e1008865. Available at: https://doi.org/10.1371/journal.pcbi.1008865.

Li, Y., Zhang, C., Zheng, W., Zhou, X., Bell, E.W., Yu, D. and Zhang, Y. (2021b) 'Protein inter-residue contact and distance prediction by coupling complementary coevolution features with deep residual networks in CASP14', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1911–1921. Available at: https://doi.org/10.1002/prot.26211.

Li, Z., Lin, Y., Elofsson, A. and Yao, Y. (2020) 'Protein Contact Map Prediction Based on ResNet and DenseNet', *BioMed Research International*, 2020, p. 7584968. Available at: https://doi.org/10.1155/2020/7584968.

Liang, T., Jiang, C., Yuan, J., Othman, Y., Xie, X.-Q. and Feng, Z. (2022) 'Differential performance of RoseTTAFold in antibody modeling', *Briefings in Bioinformatics*, 23(5), p. bbac152. Available at: https://doi.org/10.1093/bib/bbac152.

Liu, G., Zhu, Y., Zhou, W., Huang, Y., Zhou, C. and Wang, R. (2005) 'A study on protein residue contacts prediction by recurrent neural network', *Journal of Bionic Engineering*, 2(3), pp. 157–160. Available at: https://doi.org/10.1007/BF03399492.

Liu, J., He, G.-X., Zhao, K.-L. and Zhang, G.-J. (2022) 'De novo protein structure prediction by incremental inter-residue geometries prediction and model quality assessment using deep learning', *bioRxiv*, preprint. Available at: https://doi.org/10.1101/2022.01.11.475831.

Liu, J., Zhao, K. and Zhang, G. (2023) 'Improved model quality assessment using sequence and structural information by enhanced deep neural networks', *Briefings in Bioinformatics*, 24(1), p. bbac507. Available at: https://doi.org/10.1093/bib/bbac507.

Liu, Y., Palmedo, P., Ye, Q., Berger, B. and Peng, J. (2018) 'Enhancing Evolutionary Couplings with Deep Convolutional Neural Networks', *Cell Systems*, 6(1), pp. 65-74.e3. Available at: https://doi.org/10.1016/j.cels.2017.11.014.

Lundström, J., Rychlewski, L., Bujnicki, J. and Elofsson, A. (2008) 'Pcons: A neural-network-based consensus predictor that improves fold recognition', *Protein Science*, 10(11), pp. 2354–2362. Available at: https://doi.org/10.1110/ps.08501.

Ma, W., Zhang, S., Li, Z., Jiang, M., Wang, S., Lu, W., Bi, X., Jiang, H., Zhang, H. and Wei, Z. (2022) 'Enhancing Protein Function Prediction Performance by Utilizing AlphaFold-Predicted Protein Structures', *Journal of Chemical Information and Modeling*, 62(17), pp. 4008–4017. Available at: https://doi.org/10.1021/acs.jcim.2c00885.

MacCallum, J.L., Hua, L., Schnieders, M.J., Pande, V.S., Jacobson, M.P. and Dill, K.A. (2009) 'Assessment of the protein-structure refinement category in CASP8', *Proteins: Structure, Function, and Bioinformatics*, 77(S9), pp. 66–80. Available at: https://doi.org/10.1002/prot.22538.

MacCallum, J.L., Pérez, A., Schnieders, M.J., Hua, L., Jacobson, M.P. and Dill, K.A. (2011) 'Assessment of protein structure refinement in CASP9', *Proteins: Structure, Function, and*

Bioinformatics, 79(S10), pp. 74–90. Available at: https://doi.org/10.1002/prot.23131.

Maghrabi, A.H.A. (2019) *Improvements to methods for the quality assessment of three-dimensional models of proteins*. Doctoral thesis. University of Reading.

Maghrabi, A.H.A. and McGuffin, L.J. (2017) 'ModFOLD6: an accurate web server for the global and local quality estimation of 3D protein models', *Nucleic Acids Research*, 45(W1), pp. W416–W421. Available at: https://doi.org/10.1093/nar/gkx332.

Maghrabi, A.H.A. and McGuffin, L.J. (2020) 'Estimating the Quality of 3D Protein Models Using the ModFOLD7 Server', in D. Kihara (ed.) *Protein Structure Prediction.* New York, NY: Springer US, pp. 69–81. Available at: https://doi.org/10.1007/978-1-0716-0708-4_4.

Manaswi, N.K. (2018) 'Multilayer Perceptron', in N.K. Manaswi (ed.) *Deep Learning with Applications Using Python : Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras*. Berkeley, CA: Apress, pp. 45–56. Available at: https://doi.org/10.1007/978-1-4842-3516-4_3.

Marcu, Ș.-B., Tăbîrcă, S. and Tangney, M. (2022) 'An Overview of Alphafold's Breakthrough', *Frontiers in Artificial Intelligence*, 5, p. 875587. Available at: https://doi.org/10.3389/frai.2022.875587.

Mariani, V., Biasini, M., Barbato, A. and Schwede, T. (2013) 'lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests', *Bioinformatics*, 29(21), pp. 2722–2728. Available at: https://doi.org/10.1093/bioinformatics/btt473.

Marks, D.S., Colwell, L.J., Sheridan, R., Hopf, T.A., Pagnani, A., Zecchina, R. and Sander, C. (2011) 'Protein 3D Structure Computed from Evolutionary Sequence Variation', *PLoS ONE*, 6(12), p. e28766. Available at: https://doi.org/10.1371/journal.pone.0028766.

Marks, D.S., Hopf, T.A. and Sander, C. (2012) 'Protein structure prediction from sequence variation', *Nature Biotechnology*, 30(11), pp. 1072–1080. Available at: https://doi.org/10.1038/nbt.2419.

McGuffin, L.J. (2008) 'The ModFOLD server for the quality assessment of protein structural models', *Bioinformatics*, 24(4), pp. 586–587. Available at: https://doi.org/10.1093/bioinformatics/btn014.

McGuffin, L.J. (2009) 'Prediction of global and local model quality in CASP8 using the ModFOLD server', *Proteins: Structure, Function, and Bioinformatics*, 77(S9), pp. 185–190. Available at: https://doi.org/10.1002/prot.22491.

McGuffin, L.J., Adiyaman, R., Maghrabi, A.H.A., Shuid, A.N., Brackenridge, D.A., Nealon, J.O. and Philomina, L.S. (2019) 'IntFOLD: an integrated web resource for high performance protein structure and function prediction', *Nucleic Acids Research*, 47(W1), pp. W408–W413. Available at: https://doi.org/10.1093/nar/gkz322.

McGuffin, L.J., Aldowsari, F.M.F., Alharbi, S.M.A. and Adiyaman, R. (2021) 'ModFOLD8: accurate global and local quality estimates for 3D protein models', *Nucleic Acids Research*, 49(W1), pp. W425–W430. Available at: https://doi.org/10.1093/nar/gkab321.

McGuffin, L.J., Atkins, J.D., Salehe, B.R., Shuid, A.N. and Roche, D.B. (2015) 'IntFOLD: an integrated server for modelling protein structures and functions from amino acid sequences', *Nucleic Acids Research*, 43(W1), pp. W169–W173. Available at: https://doi.org/10.1093/nar/gkv236.

McGuffin, L.J., Buenavista, M.T. and Roche, D.B. (2013) 'The ModFOLD4 server for the quality assessment of 3D protein models', *Nucleic Acids Research*, 41(W1), pp. W368–W372. Available at: https://doi.org/10.1093/nar/gkt294.

McGuffin, L.J., Edmunds, N.S., Genc, A.G., Alharbi, S.M.A., Salehe, B.R. and Adiyaman, R. (2023) 'Prediction of protein structures, functions and interactions using the IntFOLD7, MultiFOLD and ModFOLDdock servers', *Nucleic Acids Research*, 51(W1), pp. W274–W280. Available at: https://doi.org/10.1093/nar/gkad297.

McGuffin, L.J. and Roche, D.B. (2010) 'Rapid model quality assessment for protein structure predictions using the comparison of multiple models without structural alignments', *Bioinformatics*, 26(2), pp. 182–188. Available at: https://doi.org/10.1093/bioinformatics/btp629.

McGuffin, L.J. and Roche, D.B. (2011) 'Automated tertiary structure prediction with accurate local model quality assessment using the Intfold-TS method', *Proteins: Structure, Function, and Bioinformatics*, 79(S10), pp. 137–146. Available at: https://doi.org/10.1002/prot.23120.

McGuffin, L.J., Shuid, A.N., Kempster, R., Maghrabi, A.H.A., Nealon, J.O., Salehe, B.R., Atkins, J.D. and Roche, D.B. (2018) 'Accurate template-based modeling in CASP12 using the IntFOLD4-TS, ModFOLD6, and ReFOLD methods', *Proteins: Structure, Function, and Bioinformatics*, 86(S1), pp. 335–344. Available at: https://doi.org/10.1002/prot.25360.

McMurry, J.E., Hoeger, C.A., Peterson, V.E. and Ballantine, D.S. (2013) 'Amino Acids and Proteins', in J.E. McMurry (ed.) *Fundamentals of General, Organic, and Biological Chemistry*. 7th edn. Pearson Education UK, pp. 596–636.

Melo, F. and Feytmans, E. (1998) 'Assessing protein structures with a non-local atomic interaction energy', *Journal of Molecular Biology*, 277(5), pp. 1141–1152. Available at: https://doi.org/10.1006/jmbi.1998.1665.

Michel, M., Menéndez Hurtado, D. and Elofsson, A. (2019) 'PconsC4: Fast, accurate and hassle-free contact predictions', *Bioinformatics*, 35(15), pp. 2677–2679. Available at: https://doi.org/10.1093/bioinformatics/bty1036.

Miller, J.N. and Miller, J.C. (2010) *Statistics and chemometrics for analytical chemistry*. 6th ed. Harlow: Prentice Hall/Pearson.

Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S. and Steinegger, M. (2022) 'ColabFold: making protein folding accessible to all', *Nature Methods*, 19(6), pp. 679–682. Available at: https://doi.org/10.1038/s41592-022-01488-1.

Modi, V. and Dunbrack, R.L. (2016) 'Assessment of refinement of template-based models in CASP11', *Proteins: Structure, Function and Bioinformatics*, 84, pp. 260–281. Available at: https://doi.org/10.1002/prot.25048.

Monastyrskyy, B., D'Andrea, D., Fidelis, K., Tramontano, A. and Kryshtafovych, A. (2014) 'Evaluation of residue–residue contact prediction in CASP10', *Proteins: Structure, Function, and Bioinformatics*, 82(S2), pp. 138–153. Available at: https://doi.org/10.1002/prot.24340.

Monastyrskyy, B., D'Andrea, D., Fidelis, K., Tramontano, A. and Kryshtafovych, A. (2016) 'New encouraging developments in contact prediction: Assessment of the CASP11 results', *Proteins: Structure, Function, and Bioinformatics*, 84(S1), pp. 131–144. Available at: https://doi.org/10.1002/prot.24943.

Monastyrskyy, B., Fidelis, K., Tramontano, A. and Kryshtafovych, A. (2011) 'Evaluation of residue–residue contact predictions in CASP9', *Proteins: Structure, Function, and Bioinformatics*, 79(S10), pp. 119–125. Available at: https://doi.org/10.1002/prot.23160.

Morcos, F., Hwa, T., Onuchic, J.N. and Weigt, M. (2014) 'Direct Coupling Analysis for Protein Contact Prediction', in D. Kihara (ed.) *Protein Structure Prediction*. New York, NY: Springer, pp. 55–70. Available at: https://doi.org/10.1007/978-1-4939-0366-5_5.

Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D.S., Sander, C., Zecchina, R., Onuchic, J.N., Hwa, T. and Weigt, M. (2011) 'Direct-coupling analysis of residue coevolution captures native contacts across many protein families', *Proceedings of the National Academy of Sciences*, 108(49), pp. E1293–E1301. Available at: https://doi.org/10.1073/pnas.1111471108.

Mortuza, S.M., Zheng, W., Zhang, C., Li, Y., Pearce, R. and Zhang, Y. (2021) 'Improving fragment-based ab initio protein structure assembly using low-accuracy contact-map predictions', *Nature Communications*, 12(1), p. 5011. Available at: https://doi.org/10.1038/s41467-021-25316-w.

Moult, J., Pedersen, J.T., Judson, R. and Fidelis, K. (1995) 'A large-scale experiment to assess protein structure prediction methods', *Proteins: Structure, Function, and Bioinformatics*, 23(3), pp. ii–iv. Available at: https://doi.org/10.1002/prot.340230303.

Mukhtorov, D., Rakhmonova, M., Muksimova, S. and Cho, Y.-I. (2023) 'Endoscopic Image Classification Based on Explainable Deep Learning', *Sensors*, 23(6), p. 3176. Available at: https://doi.org/10.3390/s23063176.

Mulnaes, D. and Gohlke, H. (2018) 'TopScore: Using Deep Neural Networks and Large Diverse Data Sets for Accurate Protein Model Quality Assessment', *Journal of Chemical Theory and Computation*, 14(11), pp. 6117–6126. Available at: https://doi.org/10.1021/acs.jctc.8b00690.

Nahirñak, V., Almasia, N.I., Hopp, H.E. and Vazquez-Rovere, C. (2012) 'Snakin/GASA proteins: Involvement in hormone crosstalk and redox homeostasis', *Plant Signaling & Behavior*, 7(8), pp. 1004–1008. Available at: https://doi.org/10.4161/psb.20813.

Niu, M., Wu, J., Zou, Q., Liu, Z. and Xu, L. (2021) 'rBPDL:Predicting RNA-Binding Proteins Using Deep Learning', *IEEE Journal of Biomedical and Health Informatics*, 25(9), pp. 3668–3676. Available at: https://doi.org/10.1109/JBHI.2021.3069259.

Olechnovič, K., Kulberkytė, E. and Venclovas, Č. (2013) 'CAD-score: A new contact area difference-based function for evaluation of protein structural models', *Proteins: Structure, Function, and Bioinformatics*, 81(1), pp. 149–162. Available at:

https://doi.org/10.1002/prot.24172.

Olechnovič, K. and Venclovas, Č. (2014a) ‘The CAD-score web server: Contact area-based comparison of structures and interfaces of proteins, nucleic acids and their complexes’, *Nucleic Acids Research*, 42(W1), pp. 259–263. Available at: https://doi.org/10.1093/nar/gku294.

Olechnovič, K. and Venclovas, Č. (2014b) ‘Voronota: A fast and reliable tool for computing the vertices of the Voronoi diagram of atomic balls’, *Journal of Computational Chemistry*, 35(8), pp. 672–681. Available at: https://doi.org/10.1002/jcc.23538.

Olechnovič, K. and Venclovas, Č. (2017) ‘VoroMQA: Assessment of protein structure quality using interatomic contact areas’, *Proteins: Structure, Function, and Bioinformatics*, 85(6), pp. 1131–1145. Available at: https://doi.org/10.1002/prot.25278.

Olmea, O. and Valencia, A. (1997) ‘Improving contact predictions by the combination of correlated mutations and other sources of sequence information’, *Folding and Design*, 2, pp. S25–S32. Available at: https://doi.org/10.1016/S1359-0278(97)00060-6.

Onufriev, A., Bashford, D. and Case, D.A. (2004) ‘Exploring protein native states and large-scale conformational changes with a modified generalized born model’, *Proteins: Structure, Function, and Bioinformatics*, 55(2), pp. 383–394. Available at: https://doi.org/10.1002/prot.20033.

Onufriev, A., Case, D.A. and Bashford, D. (2002) ‘Effective Born radii in the generalized Born approximation: The importance of being perfect’, *Journal of Computational Chemistry*, 23(14), pp. 1297–1304. Available at: https://doi.org/10.1002/jcc.10126.

Orengo, C.A., Bray, J.E., Hubbard, T., LoConte, L. and Sillitoe, I. (1999) ‘Analysis and assessment of ab initio three-dimensional prediction, secondary structure, and contacts prediction’, *Proteins: Structure, Function, and Genetics*, 37(S3), pp. 149–170. Available at: https://doi.org/10.1002/(SICI)1097-0134(1999)37:3+<149::AID-PROT20>3.0.CO;2-H.

Ovchinnikov, S., Park, H., Kim, D.E., DiMaio, F. and Baker, D. (2018) ‘Protein structure prediction using Rosetta in CASP12’, *Proteins: Structure, Function, and Bioinformatics*, 86(Suppl 1), pp. 113–121. Available at: https://doi.org/10.1002/prot.25390.

Ovchinnikov, S., Park, H., Varghese, N., Huang, P.-S., Pavlopoulos, G.A., Kim, D.E., Kamisetty, H., Kyrpides, N.C. and Baker, D. (2017) ‘Protein structure determination using metagenome sequence data’, *Science*, 355(6322), pp. 294–298. Available at: https://doi.org/10.1126/science.aah4043.

Pak, M.A., Markhieva, K.A., Novikova, M.S., Petrov, D.S., Vorobyev, I.S., Maksimova, E.S., Kondrashov, F.A. and Ivankov, D.N. (2023) ‘Using AlphaFold to predict the impact of single mutations on protein stability and function’, *PLOS ONE*, 18(3), p. e0282689. Available at: https://doi.org/10.1371/journal.pone.0282689.

Pakhrin, S.C., Shrestha, B., Adhikari, B., Kc, D.B. and Gelly, J.-C. (2021) ‘Deep Learning-Based Advances in Protein Structure Prediction’, *International Journal of Molecular Sciences*, 22(11), p. 5553. Available at: https://doi.org/10.3390/ijms22115553.

Park, H., Lee, G.R., Kim, D.E., Anishchenko, I., Cong, Q. and Baker, D. (2019) ‘High-

accuracy refinement using Rosetta in CASP13', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1276–1282. Available at: https://doi.org/10.1002/prot.25784.

Pearce, R. and Zhang, Y. (2021a) 'Deep learning techniques have significantly impacted protein structure prediction and protein design', *Current Opinion in Structural Biology*, 68, pp. 194–207. Available at: https://doi.org/10.1016/j.sbi.2021.01.007.

Pearce, R. and Zhang, Y. (2021b) 'Toward the solution of the protein structure prediction problem', *Journal of Biological Chemistry*, 297(1), p. 100870. Available at: https://doi.org/10.1016/j.jbc.2021.100870.

Peng, C.-X., Zhou, X.-G. and Zhang, G.-J. (2022) 'De novo Protein Structure Prediction by Coupling Contact With Distance Profile', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(1), pp. 395–406. Available at: https://doi.org/10.1109/TCBB.2020.3000758.

Pereira, J., Simpkin, A.J., Hartmann, M.D., Rigden, D.J., Keegan, R.M. and Lupas, A.N. (2021) 'High-accuracy protein structure prediction in CASP14', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1687–1699. Available at: https://doi.org/10.1002/prot.26171.

Probst, P., Bischl, B. and Boulesteix, A.-L. (2018) 'Tunability: Importance of Hyperparameters of Machine Learning Algorithms', *Journal of Machine Learning Research*, 20(53), pp. 1--32. Available at: https://doi.org/10.48550/ARXIV.1802.09596.

Puton, T., Kozlowski, L., Tuszynska, I., Rother, K. and Bujnicki, J.M. (2012) 'Computational methods for prediction of protein–RNA interactions', *Journal of Structural Biology*, 179(3), pp. 261–268. Available at: https://doi.org/10.1016/j.jsb.2011.10.001.

Quignot, C., Granger, P., Chacón, P., Guerois, R. and Andreani, J. (2021) 'Atomic-level evolutionary information improves protein–protein interface scoring', *Bioinformatics*, 37(19), pp. 3175–3181. Available at: https://doi.org/10.1093/bioinformatics/btab254.

Rana, A., Singh Rawat, A., Bijalwan, A. and Bahuguna, H. (2018) 'Application of Multi Layer (Perceptron) Artificial Neural Network in the Diagnosis System: A Systematic Review', in *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE). 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)*, San Salvador: IEEE, pp. 1–6. Available at: https://doi.org/10.1109/RICE.2018.8509069.

Rangwala, H. and Karypis, G. (2010) 'Introduction to Protein Structure Prediction', in H. Rangwala and G. Karypis (eds) *Introduction to Protein Structure Prediction*. 1st edn. Hoboken, New Jersey: John Wiley & Sons, Inc., pp. 1–13. Available at: https://doi.org/10.1002/9780470882207.ch1.

Ray, A., Lindahl, E. and Wallner, B. (2012) 'Improved model quality assessment using ProQ2', *BMC Bioinformatics*, 13(1), p. 224. Available at: https://doi.org/10.1186/1471-2105-13-224.

Read, R.J., Sammito, M.D., Kryshtafovych, A. and Croll, T.I. (2019) 'Evaluation of model refinement in CASP13', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1249–1262. Available at: https://doi.org/10.1002/prot.25794.

Reza, Md., Zhang, H., Hossain, Md., Jin, L., Feng, S. and Wei, Y. (2021) 'COMTOP: Protein Residue–Residue Contact Prediction through Mixed Integer Linear Optimization', *Membranes*, 11(7), p. 503. Available at: https://doi.org/10.3390/membranes11070503.

Robin, X., Haas, J., Gumienny, R., Smolinski, A., Tauriello, G. and Schwede, T. (2021) 'Continuous Automated Model EvaluatiOn (CAMEO)—Perspectives on the future of fully automated evaluation of structure prediction methods', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1977–1986. Available at: https://doi.org/10.1002/prot.26213.

Roche, D.B., Buenavista, M.T. and McGuffin, L.J. (2013) 'The FunFOLD2 server for the prediction of protein–ligand interactions', *Nucleic Acids Research*, 41(W1), pp. W303–W307. Available at: https://doi.org/10.1093/nar/gkt498.

Roche, D.B., Buenavista, M.T. and McGuffin, L.J. (2014) 'Assessing the Quality of Modelled 3D Protein Structures Using the ModFOLD Server', in D. Kihara (ed.) *Protein Structure Prediction*. New York, NY: Springer New York, pp. 83–103. Available at: https://doi.org/10.1007/978-1-4939-0366-5_7.

Roche, D.B., Buenavista, M.T., Tetchner, S.J. and McGuffin, L.J. (2011) 'The IntFOLD server: an integrated web resource for protein fold recognition, 3D model quality assessment, intrinsic disorder prediction, domain prediction and ligand binding site prediction', *Nucleic Acids Research*, 39(suppl_2), pp. W171–W176. Available at: https://doi.org/10.1093/nar/gkr184.

Roche, D.B., Tetchner, S.J. and McGuffin, L.J. (2011) 'FunFOLD: an improved automated method for the prediction of ligand binding residues using 3D models of proteins', *BMC Bioinformatics*, 12(1), p. 160. Available at: https://doi.org/10.1186/1471-2105-12-160.

Roy, R.S., Liu, J., Giri, N., Guo, Z. and Cheng, J. (2023) 'Combining pairwise structural similarity and deep learning interface contact prediction to estimate protein complex model accuracy in CASP15', *Proteins: Structure, Function, and Bioinformatics*, 91(12), pp. 1889–1902. Available at: https://doi.org/10.1002/prot.26542.

Ruiz-Serra, V., Pontes, C., Milanetti, E., Kryshtafovych, A., Lepore, R. and Valencia, A. (2021) 'Assessing the accuracy of contact and distance predictions in CASP14', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1888–1900. Available at: https://doi.org/10.1002/prot.26248.

Saito, T. and Rehmsmeier, M. (2015) 'The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets', *PLOS ONE*, 10(3), p. e0118432. Available at: https://doi.org/10.1371/journal.pone.0118432.

Saldaño, T., Escobedo, N., Marchetti, J., Zea, D.J., Mac Donagh, J., Velez Rueda, A.J., Gonik, E., García Melani, A., Novomisky Nechcoff, J., Salas, M.N., Peters, T., Demitroff, N., Fernandez Alberti, S., Palopoli, N., Fornasari, M.S. and Parisi, G. (2022) 'Impact of protein conformational diversity on AlphaFold predictions', *Bioinformatics*, 38(10), pp. 2742–2748. Available at: https://doi.org/10.1093/bioinformatics/btac202.

Sathiyamani, B., Daniel, E.A., Ansar, S., Esakialraj, B.H., Hassan, S., Revanasiddappa, P.D., Keshavamurthy, A., Roy, S., Vetrivel, U. and Hanna, L.E. (2023) 'Structural analysis and molecular dynamics simulation studies of HIV-1 antisense protein predict its potential role in

HIV replication and pathogenesis', *Frontiers in Microbiology*, 14, p. 1152206. Available at: https://doi.org/10.3389/fmicb.2023.1152206.

Schaarschmidt, J., Monastyrskyy, B., Kryshtafovych, A. and Bonvin, A.M.J.J. (2018) 'Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age', *Proteins: Structure, Function, and Bioinformatics*, 86(S1), pp. 51–66. Available at: https://doi.org/10.1002/prot.25407.

Schneider, R., De Daruvar, A. and Sander, C. (1997) 'The HSSP database of protein structure-sequence alignments', *Nucleic Acids Research*, 25(1), pp. 226–230. Available at: https://doi.org/10.1093/nar/25.1.226.

Seemayer, S., Gruber, M. and Söding, J. (2014) 'CCMpred - Fast and precise prediction of protein residue-residue contacts from correlated mutations', *Bioinformatics*, 30(21), pp. 3128–3130. Available at: https://doi.org/10.1093/bioinformatics/btu500.

Sehnal, D., Bittrich, S., Deshpande, M., Svobodová, R., Berka, K., Bazgier, V., Velankar, S., Burley, S.K., Koča, J. and Rose, A.S. (2021) 'Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures', *Nucleic Acids Research*, 49(W1), pp. W431–W437. Available at: https://doi.org/10.1093/nar/gkab314.

Senior, A.W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A.W.R., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D.T., Silver, D., Kavukcuoglu, K. and Hassabis, D. (2020) 'Improved protein structure prediction using potentials from deep learning', *Nature*, 577(7792), pp. 706–710. Available at: https://doi.org/10.1038/s41586-019-1923-7.

Shackelford, G. and Karplus, K. (2007) 'Contact prediction using mutual information and neural nets', *Proteins: Structure, Function, and Bioinformatics*, 69(S8), pp. 159–164. Available at: https://doi.org/10.1002/prot.21791.

Shao, Y. and Bystroff, C. (2003) 'Predicting interresidue contacts using templates and pathways', *Proteins: Structure, Function, and Genetics*, 53(S6), pp. 497–502. Available at: https://doi.org/10.1002/prot.10539.

Sheela, K.G. and Deepa, S.N. (2013) 'Review on Methods to Fix Number of Hidden Neurons in Neural Networks', *Mathematical Problems in Engineering*, 2013, pp. 1–11. Available at: https://doi.org/10.1155/2013/425740.

Shen, M. and Sali, A. (2006) 'Statistical potential for assessment and prediction of protein structures', *Protein Science*, 15(11), pp. 2507–2524. Available at: https://doi.org/10.1110/ps.062416606.

Shrestha, R., Fajardo, E., Gil, N., Fidelis, K., Kryshtafovych, A., Monastyrskyy, B. and Fiser, A. (2019) 'Assessing the accuracy of contact predictions in CASP13', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1058–1068. Available at: https://doi.org/10.1002/prot.25819.

Simpkin, A.J., Mesdaghi, S., Sánchez Rodríguez, F., Elliott, L., Murphy, D.L., Kryshtafovych, A., Keegan, R.M. and Rigden, D.J. (2023) 'Tertiary structure assessment at CASP15', *Proteins: Structure, Function, and Bioinformatics*, 91(12), pp. 1616–1635. Available at: https://doi.org/10.1002/prot.26593.

Singh, A.K. and Ranjan, R. (2022) ‘Multi-Layer Perceptron Based Spectrum Prediction in Cognitive Radio Network’, *Wireless Personal Communications*, 123(4), pp. 3539–3553. Available at: https://doi.org/10.1007/s11277-021-09302-5.

Stecking, R. and Schebesch, K.B. (2005) ‘Informative Patterns for Credit Scoring: Support Vector Machines Preselect Data Subsets for Linear Discriminant Analysis’, in C. Weihs and W. Gaul (eds) *Classification — the Ubiquitous Challenge*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 450–457. Available at: https://doi.org/10.1007/3-540-28084-7_52.

Stern, J., Hedelius, B., Fisher, O., Billings, W.M. and Della Corte, D. (2021) ‘Evaluation of Deep Neural Network ProSPr for Accurate Protein Distance Predictions on CASP14 Targets’, *International Journal of Molecular Sciences*, 22(23), p. 12835. Available at: https://doi.org/10.3390/ijms222312835.

Stollar, E.J. and Smith, D.P. (2020) ‘Uncovering protein structure’, *Essays in Biochemistry*, 64(4), pp. 649–680. Available at: https://doi.org/10.1042/EBC20190042.

Studer, G., Biasini, M. and Schwede, T. (2014) ‘Assessing the local structural quality of transmembrane protein models using statistical potentials (QMEANBrane)’, *Bioinformatics*, 30(17), pp. i505–i511. Available at: https://doi.org/10.1093/bioinformatics/btu457.

Suh, D., Lee, J.W., Choi, S. and Lee, Y. (2021) ‘Recent Applications of Deep Learning Methods on Evolution- and Contact-Based Protein Structure Prediction’, *International Journal of Molecular Sciences*, 22(11), p. 6032. Available at: https://doi.org/10.3390/ijms22116032.

Tegge, A.N., Wang, Z., Eickholt, J. and Cheng, J. (2009) ‘NNcon: improved protein contact map prediction using 2D-recursive neural networks’, *Nucleic Acids Research*, 37(suppl_2), pp. W515–W518. Available at: https://doi.org/10.1093/nar/gkp305.

Tharwat, A. (2021) ‘Classification assessment methods’, *Applied Computing and Informatics*, 17(1), pp. 168–192. Available at: https://doi.org/10.1016/j.aci.2018.08.003.

Thompson, J.D., Linard, B., Lecompte, O. and Poch, O. (2011) ‘A Comprehensive Benchmark Study of Multiple Sequence Alignment Methods: Current Challenges and Future Perspectives’, *PLoS ONE*, 6(3), p. e18093. Available at: https://doi.org/10.1371/journal.pone.0018093.

Torrisi, M., Pollastri, G. and Le, Q. (2020) ‘Deep learning methods in protein structure prediction’, *Computational and Structural Biotechnology Journal*, 18, pp. 1301–1310. Available at: https://doi.org/10.1016/j.csbj.2019.12.011.

Tress, M.L. and Valencia, A. (2010) ‘Predicted residue–residue contacts can help the scoring of 3D models’, *Proteins: Structure, Function, and Bioinformatics*, 78(8), pp. 1980–1991. Available at: https://doi.org/10.1002/prot.22714.

Uziela, K., Menéndez Hurtado, D., Shu, N., Wallner, B. and Elofsson, A. (2017) ‘ProQ3D: improved model quality assessments using deep learning’, *Bioinformatics*, 33(10), pp. 1578–1580. Available at: https://doi.org/10.1093/bioinformatics/btw819.

Uziela, K., Shu, N., Wallner, B. and Elofsson, A. (2016) ‘ProQ3: Improved model quality assessments using Rosetta energy terms’, *Scientific Reports*, 6(1), p. 33509. Available at:

https://doi.org/10.1038/srep33509.

Uziela, K. and Wallner, B. (2016) 'ProQ2: estimation of model accuracy implemented in Rosetta', *Bioinformatics*, 32(9), pp. 1411–1413. Available at: https://doi.org/10.1093/bioinformatics/btv767.

Vabalas, A., Gowen, E., Poliakoff, E. and Casson, A.J. (2019) 'Machine learning algorithm validation with a limited sample size', *PLOS ONE*, 14(11), p. e0224365. Available at: https://doi.org/10.1371/journal.pone.0224365.

Vaz, J.M. and Balaji, S. (2021) 'Convolutional neural networks (CNNs): concepts and applications in pharmacogenomics', *Molecular Diversity*, 25(3), pp. 1569–1584. Available at: https://doi.org/10.1007/s11030-021-10225-3.

Wallner, B. (2006) 'Identification of correct regions in protein models using structural, alignment, and consensus information', *Protein Science*, 15(4), pp. 900–913. Available at: https://doi.org/10.1110/ps.051799606.

Wallner, B. and Elofsson, A. (2007) 'Prediction of global and local model quality in CASP7 using Pcons and ProQ', *Proteins: Structure, Function, and Bioinformatics*, 69(S8), pp. 184–193. Available at: https://doi.org/10.1002/prot.21774.

Wang, J., Cao, H., Zhang, J.Z.H. and Qi, Y. (2018) 'Computational Protein Design with Deep Learning Neural Networks', *Scientific Reports*, 8(1), p. 6349. Available at: https://doi.org/10.1038/s41598-018-24760-x.

Wang, S., Li, Z., Yu, Y. and Xu, J. (2017) 'Folding Membrane Proteins by Deep Transfer Learning', *Cell Systems*, 5(3), pp. 202-211.e3. Available at: https://doi.org/10.1016/j.cels.2017.09.001.

Wang, S., Sun, S., Li, Z., Zhang, R. and Xu, J. (2017) 'Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model', *PLOS Computational Biology*, 13(1), p. e1005324. Available at: https://doi.org/10.1371/journal.pcbi.1005324.

Wang, S., Sun, S. and Xu, J. (2018) 'Analysis of deep learning methods for blind protein contact prediction in CASP12', *Proteins: Structure, Function, and Bioinformatics*, 86(S1), pp. 67–77. Available at: https://doi.org/10.1002/prot.25377.

Wang, X.-F., Chen, Z., Wang, C., Yan, R.-X., Zhang, Z. and Song, J. (2011) 'Predicting Residue-Residue Contacts and Helix-Helix Interactions in Transmembrane Proteins Using an Integrative Feature-Based Random Forest Approach', *PLoS ONE*, 6(10), p. e26767. Available at: https://doi.org/10.1371/journal.pone.0026767.

Wang, Z., Shen, W., Kotler, D.P., Heshka, S., Wielopolski, L., Aloia, J.F., Nelson, M.E., Pierson, R.N. and Heymsfield, S.B. (2003) 'Total body protein: a new cellular level mass and distribution prediction model', *The American Journal of Clinical Nutrition*, 78(5), pp. 979–984. Available at: https://doi.org/10.1093/ajcn/78.5.979.

Wang, Z. and Xu, J. (2013) 'Predicting protein contact map using evolutionary and physical constraints by integer programming', *Bioinformatics*, 29(13), pp. i266–i273. Available at: https://doi.org/10.1093/bioinformatics/btt211.

Waterhouse, A., Bertoni, M., Bienert, S., Studer, G., Tauriello, G., Gumienny, R., Heer, F.T., De Beer, T.A.P., Rempfer, C., Bordoli, L., Lepore, R. and Schwede, T. (2018) 'SWISS-MODEL: Homology modelling of protein structures and complexes', *Nucleic Acids Research*, 46(W1), pp. W296–W303. Available at: https://doi.org/10.1093/nar/gky427.

Wei, Y., Thompson, J. and Floudas, C.A. (2012) 'CONCORD: a consensus method for protein secondary structure prediction via mixed integer linear optimization', *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 468(2139), pp. 831–850. Available at: https://doi.org/10.1098/rspa.2011.0514.

Weigt, M., White, R.A., Szurmant, H., Hoch, J.A. and Hwa, T. (2009) 'Identification of direct residue contacts in protein–protein interaction by message passing', *Proceedings of the National Academy of Sciences*, 106(1), pp. 67–72. Available at: https://doi.org/10.1073/pnas.0805923106.

Wen, B., Zeng, W.-F., Liao, Y., Shi, Z., Savage, S.R., Jiang, W. and Zhang, B. (2020) 'Deep Learning in Proteomics', *PROTEOMICS*, 20(21–22), p. 1900335. Available at: https://doi.org/10.1002/pmic.201900335.

Wodak, S.J., Vajda, S., Lensink, M.F., Kozakov, D. and Bates, P.A. (2023) 'Critical Assessment of Methods for Predicting the 3D Structure of Proteins and Protein Complexes', *Annual Review of Biophysics*, 52(1), pp. 183–206. Available at: https://doi.org/10.1146/annurev-biophys-102622-084607.

Won, J., Baek, M., Monastyrskyy, B., Kryshtafovych, A. and Seok, C. (2019) 'Assessment of protein model structure accuracy estimation in CASP13: Challenges in the era of deep learning', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1351–1360. Available at: https://doi.org/10.1002/prot.25804.

Wu, Q., Peng, Z., Anishchenko, I., Cong, Q., Baker, D. and Yang, J. (2020) 'Protein contact prediction using metagenome sequence data and residual neural networks', *Bioinformatics*, 36(1), pp. 41–48. Available at: https://doi.org/10.1093/bioinformatics/btz477.

Wu, S., Szilagyi, A. and Zhang, Y. (2011) 'Improving Protein Structure Prediction Using Multiple Sequence-Based Contact Predictions', *Structure*, 19(8), pp. 1182–1191. Available at: https://doi.org/10.1016/j.str.2011.05.004.

Wu, S. and Zhang, Y. (2007) 'LOMETS: A local meta-threading-server for protein structure prediction', *Nucleic Acids Research*, 35(10), pp. 3375–3382. Available at: https://doi.org/10.1093/nar/gkm251.

Wu, S. and Zhang, Y. (2008) 'A comprehensive assessment of sequence-based and template-based methods for protein contact prediction', *Bioinformatics*, 24(7), pp. 924–931. Available at: https://doi.org/10.1093/bioinformatics/btn069.

Wu, T., Guo, Z., Hou, J. and Cheng, J. (2021) 'DeepDist: real-value inter-residue distance prediction with deep residual convolutional network', *BMC Bioinformatics*, 22(1), p. 30. Available at: https://doi.org/10.1186/s12859-021-03960-9.

Wu, T., Hou, J., Adhikari, B. and Cheng, J. (2020) 'Analysis of several key factors influencing deep learning-based inter-residue contact prediction', *Bioinformatics*, 36(4), pp. 1091–1098. Available at: https://doi.org/10.1093/bioinformatics/btz679.

Xu, J. (2019) ‘Distance-based protein folding powered by deep learning’, *Proceedings of the National Academy of Sciences*, 116(34), pp. 16856–16865. Available at: https://doi.org/10.1073/pnas.1821309116.

Xu, J., McPartlon, M. and Li, J. (2021) ‘Improved protein structure prediction by deep learning irrespective of co-evolution information’, *Nature Machine Intelligence*, 3(7), pp. 601–609. Available at: https://doi.org/10.1038/s42256-021-00348-5.

Xu, J. and Wang, S. (2019) ‘Analysis of distance-based protein structure prediction by deep learning in CASP13’, *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1069–1081. Available at: https://doi.org/10.1002/prot.25810.

Xue, B., Faraggi, E. and Zhou, Y. (2009) ‘Predicting residue–residue contact maps by a two-layer, integrated neural-network method’, *Proteins: Structure, Function, and Bioinformatics*, 76(1), pp. 176–183. Available at: https://doi.org/10.1002/prot.22329.

Yan, J. and Kurgan, L. (2015) ‘Consensus-Based Prediction of RNA and DNA Binding Residues from Protein Sequences’, in M. Kryszkiewicz, S. Bandyopadhyay, H. Rybinski, and S.K. Pal (eds) *Pattern Recognition and Machine Intelligence*. Cham: Springer International Publishing, pp. 501–511. Available at: https://doi.org/10.1007/978-3-319-19941-2_48.

Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S. and Baker, D. (2020) ‘Improved protein structure prediction using predicted interresidue orientations’, *Proceedings of the National Academy of Sciences*, 117(3), pp. 1496–1503. Available at: https://doi.org/10.1073/pnas.1914677117.

Yang, J., Wang, Y. and Zhang, Y. (2016) ‘ResQ: An approach to unified estimation of B-factor and residue-specific error in protein structure prediction’, *Journal of Molecular Biology*, 428(4), pp. 693–701. Available at: https://doi.org/10.1016/j.jmb.2015.09.024.

Yang, J.-Y. and Chen, X. (2011) ‘A Consensus Approach to Predicting Protein Contact Map via Logistic Regression’, in J. Chen, J. Wang, and A. Zelikovsky (eds) *Bioinformatics Research and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 136–147. Available at: https://doi.org/10.1007/978-3-642-21260-4_16.

Yang, M. and Ma, J. (2022) ‘Machine Learning Methods for Exploring Sequence Determinants of 3D Genome Organization’, *Journal of Molecular Biology*, 434(15), p. 167666. Available at: https://doi.org/10.1016/j.jmb.2022.167666.

Yang, Z., Zeng, X., Zhao, Y. and Chen, R. (2023) ‘AlphaFold2 and its applications in the fields of biology and medicine’, *Signal Transduction and Targeted Therapy*, 8(1), p. 115. Available at: https://doi.org/10.1038/s41392-023-01381-z.

Ye, L., Wu, P., Peng, Z., Gao, J., Liu, J. and Yang, J. (2021) ‘Improved estimation of model quality using predicted inter-residue distance’, *Bioinformatics*, 37(21), pp. 3752–3759. Available at: https://doi.org/10.1093/bioinformatics/btab632.

Ying, X. (2019) ‘An Overview of Overfitting and its Solutions’, *Journal of Physics: Conference Series*, 1168, p. 022022. Available at: https://doi.org/10.1088/1742-6596/1168/2/022022.

Yuan, C., Chen, H. and Kihara, D. (2012) ‘Effective inter-residue contact definitions for

accurate protein fold recognition', *BMC Bioinformatics*, 13(1), p. 292. Available at: https://doi.org/10.1186/1471-2105-13-292.

Zahiri, J., Emamjomeh, A., Bagheri, S., Ivazeh, A., Mahdevar, G., Sepasi Tehrani, H., Mirzaie, M., Fakheri, B.A. and Mohammad-Noori, M. (2020) 'Protein complex prediction: A survey', *Genomics*, 112(1), pp. 174–183. Available at: https://doi.org/10.1016/j.ygeno.2019.01.011.

Zhang, C., Zheng, W., Mortuza, S.M., Li, Y. and Zhang, Y. (2020) 'DeepMSA: Constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins', *Bioinformatics*, 36(7), pp. 2105–2112. Available at: https://doi.org/10.1093/bioinformatics/btz863.

Zhang, H., Huang, Q., Bei, Z., Wei, Y. and Floudas, C.A. (2016) 'COMSAT: Residue contact prediction of transmembrane proteins based on support vector machines and mixed integer linear programming', *Proteins: Structure, Function, and Bioinformatics*, 84(3), pp. 332–348. Available at: https://doi.org/10.1002/prot.24979.

Zhang, Huiling, Bei, Z., Xi, W., Hao, M., Ju, Z., Saravanan, K.M., Zhang, Haiping, Guo, N. and Wei, Y. (2021) 'Evaluation of residue-residue contact prediction methods: From retrospective to prospective', *PLOS Computational Biology*, 17(5), p. e1009027. Available at: https://doi.org/10.1371/journal.pcbi.1009027.

Zhang, J., Liang, Y. and Zhang, Y. (2011) 'Atomic-Level Protein Structure Refinement Using Fragment-Guided Molecular Dynamics Conformation Sampling', *Structure*, 19(12), pp. 1784–1795. Available at: https://doi.org/10.1016/j.str.2011.09.022.

Zhang, P., Xia, C. and Shen, H.-B. (2023) 'High-accuracy protein model quality assessment using attention graph neural networks', *Briefings in Bioinformatics*, 24(2), p. bbac614. Available at: https://doi.org/10.1093/bib/bbac614.

Zhang, Y. (2008) 'Progress and challenges in protein structure prediction', *Current Opinion in Structural Biology*, 18(3), pp. 342–348. Available at: https://doi.org/10.1016/j.sbi.2008.02.004.

Zhang, Y. (2009a) 'I-TASSER: Fully automated protein structure prediction in CASP8', *Proteins: Structure, Function, and Bioinformatics*, 77(S9), pp. 100–113. Available at: https://doi.org/10.1002/prot.22588.

Zhang, Y. (2009b) 'Protein structure prediction: when is it useful?', *Current Opinion in Structural Biology*, 19(2), pp. 145–155. Available at: https://doi.org/10.1016/j.sbi.2009.02.005.

Zhang, Y. and Skolnick, J. (2004) 'Scoring function for automated assessment of protein structure template quality', *Proteins: Structure, Function, and Bioinformatics*, 57(4), pp. 702–710. Available at: https://doi.org/10.1002/prot.20264.

Zhao, Y. and Karypis, G. (2003) 'Prediction of contact maps using support vector machines', in *Third IEEE Symposium on Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on BioInformatics and BioEngineering. BIBE 2003*, Bethesda, MD, USA: IEEE Comput. Soc, pp. 26–33. Available at: https://doi.org/10.1109/BIBE.2003.1188926.

Zhao, Y., Liu, R., Liu, Z., Liu, L., Wang, J. and Liu, W. (2023) 'A Review of Macroscopic Carbon Emission Prediction Model Based on Machine Learning', *Sustainability*, 15(8), p. 6876. Available at: https://doi.org/10.3390/su15086876.

Zheng, C., Wang, M., Takemoto, K., Akutsu, T., Zhang, Z. and Song, J. (2012) 'An Integrative Computational Framework Based on a Two-Step Random Forest Algorithm Improves Prediction of Zinc-Binding Sites in Proteins', *PLoS ONE*, 7(11), p. e49716. Available at: https://doi.org/10.1371/journal.pone.0049716.

Zheng, W., Li, Y., Zhang, C., Pearce, R., Mortuza, S.M. and Zhang, Y. (2019) 'Deep-learning contact-map guided protein structure prediction in CASP13', *Proteins: Structure, Function, and Bioinformatics*, 87(12), pp. 1149–1164. Available at: https://doi.org/10.1002/prot.25792.

Zheng, W., Li, Y., Zhang, C., Zhou, X., Pearce, R., Bell, E.W., Huang, X. and Zhang, Y. (2021) 'Protein structure prediction using deep learning distance and hydrogen-bonding restraints in CASP14', *Proteins: Structure, Function, and Bioinformatics*, 89(12), pp. 1734–1751. Available at: https://doi.org/10.1002/prot.26193.

Zubair, M., Ponniah, G., Yang, Y.J. and Choi, K.H. (2014) 'Web Tension regulation of multispan roll-to-roll system using integrated active dancer and load cells for printed electronics applications', *Chinese Journal of Mechanical Engineering*, 27(2), pp. 229–239. Available at: https://doi.org/10.3901/CJME.2014.02.229.

Appendices

Appendix 1

Consensus Code

```
#!/usr/bin/env python3

import sys
import numpy as np
import pandas as pd
from itertools import chain, repeat
import json
import math
import recordlinkage
import os.path



#Read native protein data (long type)
def NativeD(filename1):
    with open(filename1) as f:
        df = json.load(f)
    chainer = chain.from_iterable
    NATIVE = pd.DataFrame({'R1': list(chainer(repeat(k, len(v)) for k, v in
df.items())), 'R2' : list (chainer(df.values()))})
    NATIVE['R1'] = NATIVE['R1'].astype(int)
    NATIVE['R2']= NATIVE['R2'].astype(int)
    NATIVE['true_class']= (NATIVE['R2'] != 0).astype(int)
    #save residue pairs of native data in new dataframe for evaluation
process:
    NL=NATIVE[['R1', 'R2']]

    #comput the number of native contacts for evaluation process:
    Nc= []
    for index, row in NATIVE.iterrows():
        if row['R1'] > row['R2']:
            if (row['R1']-row['R2']) >= 24:
                Nc.append(row ['R2'])
            else:
                pass
        else:
            pass

    return (NATIVE, NL, Nc)


#recall native data

Native_data = NativeD(sys.argv[1])

native = Native_data[0] #native data
RR_native = Native_data[1] # residue pairs
Nc_N = len(Native_data[2]) # the number of native contacts
print(Nc_N)
```

```
# A function to read and extract prediction data of individual methods:

def subset(filename):
    with open(filename, 'r') as f:
        df1 = json.load(f)
        for k in df1.values():# Extracting data and save as dataframe:
            key = [k]
            for j in key:
                values=[j]
                S =list(values[0].values())
        L_10 = S[0] # top10 set
        L_5= S[1]   # L/5 set
        L_2= S[2]   # L/2 set
        FL_0 = S[3] # FL set
        L_0 = S[4]  # L set

        # save the subsets as dataframes:

        L10 =pd.DataFrame(L_10, columns=['R1', 'R2', 'P'], dtype = int) #
top 10 of predicted contact
        L5 = pd.DataFrame(L_5, columns=['R1', 'R2', 'P'], dtype= int)   #
L/5 set
        L2 = pd.DataFrame(L_2, columns=['R1', 'R2', 'P'], dtype = int)  #
L/2 set
        L = pd.DataFrame(L_0, columns=['R1', 'R2', 'P'], dtype= int)    # L
set
        FL = pd.DataFrame(FL_0, columns=['R1', 'R2', 'P'], dtype = int) #
Full list

        # convert all numbers of residues pairs from string to integer:

        L10['R2']= L10['R2'].astype(int)
        L10['R1'] = L10['R1'].astype(int)
        L10['P'] = L10['P'].astype(float)
        L5['R2']= L5['R2'].astype(int)
        L5['R1'] = L5['R1'].astype(int)
        L5['P']= L5['P'].astype(float)
        L2['R2']= L2['R2'].astype(int)
        L2['R1'] = L2['R1'].astype(int)
        L2['P']= L2['P'].astype(float)
        L['R2']= L['R2'].astype(int)
        L['R1'] = L['R1'].astype(int)
        L['P']= L['P'].astype(float)
        FL['R2']= FL['R2'].astype(int)
        FL['R1'] = FL['R1'].astype(int)
        FL['P'] = FL['P'].astype(float)

        return (L10, L5, L2, L, FL)



# building a function for classifying consensus data into subsets (top10,
L/5, L/2, L, FL):
def sets(length, data, p_value):
```

```
    for x in range(length):
        if x == 10:
            Top10=(data.nlargest(x, p_value))
            Top10= Top10.astype(str).values.tolist()
            L10 =pd.DataFrame(Top10, columns=['R1', 'R2', 'ConsP'], dtype =
int) # top 10 of predicted contact
            L10['R1']= L10['R1'].astype(int)
            L10['R2'] = L10['R2'].astype(int)
            L10['ConsP']= L10['ConsP'].astype(float)

        elif x == math.ceil(length/5):
            L_5= (data.nlargest(x, p_value))
            L_5 = L_5.astype(str).values.tolist()
            L5 = pd.DataFrame(L_5, columns=['R1', 'R2', 'ConsP'], dtype=
int)   # L/5 set
            L5['R1']= L5['R1'].astype(int)
            L5['R2'] = L5['R2'].astype(int)
            L5['ConsP']= L5['ConsP'].astype(float)


        elif x == math.ceil(length/2):
            L_2 = (data.nlargest(x, p_value))
            L_2 =L_2.astype(str).values.tolist()
            L2 = pd.DataFrame(L_2, columns=['R1', 'R2', 'ConsP'], dtype =
int)  # L/2 set
            L2['R1']= L2['R1'].astype(int)
            L2['R2'] = L2['R2'].astype(int)
            L2['ConsP']= L2['ConsP'].astype(float)

        else:
            L_1 = (data.nlargest(length, p_value))
            L_1 = L_1.astype(str).values.tolist()
            L = pd.DataFrame(L_1, columns=['R1', 'R2', 'ConsP'], dtype=
int)    # L set
            L['R1']= L['R1'].astype(int)
            L['R2'] = L['R2'].astype(int)
            L['ConsP']= L['ConsP'].astype(float)



            FL_0=data.astype(str).values.tolist()
            FL = pd.DataFrame(FL_0, columns=['R1', 'R2', 'ConsP'], dtype =
int) # Full list
            FL['R1']= FL['R1'].astype(int)
            FL['R2'] = FL['R2'].astype(int)
            FL['ConsP'] = FL['ConsP'].astype(float)
            FL = FL.drop_duplicates(subset=['R1', 'R2'], keep ='first')
    return (L10, L5, L2, L, FL)

#Building function that computing scores of evaluation measurements:
def Scores(experD, predD, class1, class2, v):
    # experD is the native contact data
    # predD is the prediction contact data
    # class1 is the positive cases in prediction data (contacts)
    # class 2 is the true cases in native data
    # v is the number of all contacts in  native data
    #save the native and prediction dataframe with multiindex (R1, R2):
```

```
    native =pd.MultiIndex.from_frame(experD, names=('R1', 'R2'))
    pred =pd.MultiIndex.from_frame(predD, names=('R1', 'R2'))

    # compute confusion matrix using recordlinkage package:

    Conf= recordlinkage.confusion_matrix(native, pred)

    # count TP and FP from confusion matrix
    TP= Conf[0][0]
    FP= Conf[1][0]

    #count TN and FN according to the true cases of native and against to
positive and negative cases of prediction data, where class1 = 0 represent
negative cases (non-contact) and class1 = 1 represent positive cases
(contact):
    TN = np.sum(np.logical_and(class1 == 0, class2 == 0))
    FN = np.sum(np.logical_and(class1 == 1, class2 == 0))


    # calculate precision, recall and f1_score:
    if TP != 0:
        Precision =recordlinkage.precision(native, pred)
        Recall = (TP/v)
        f1_score= 2* Precision * Recall/(Precision + Recall)

    else:
        Precision =0
        Recall = 0
        f1_score = 0

    # save the scores as list:
    scores=[TP, FP, FN, TN, Precision*100, Recall*100, f1_score*100]
    return scores



# build a function to apply consensus method:

#Consensus two methods:

def Cons2(d, d1):
    #first: merge the input data into one dataframe:
    # d, d1 represent the input data:

    data= pd.merge(d, d1, on=['R1', 'R2'], how='outer')

   #padding p_value into zero and covert its type to float:
    data['P_x'].fillna(0, inplace = True)
    data['P_y'].fillna(0, inplace = True)
    data['P_x'] = data['P_x'].astype(float)
    data['P_y']= data['P_y'].astype(float)

   #Second: Consensus prediction:
    #calculating mean of probabilities for each residue pairs from two data
    data['ConsP'] = data[['P_x', 'P_y']].mean(axis=1)
```

```
    data['ConsP'].fillna(0, inplace = True)
    pd.options.display.float_format = '{:.3f}'.format # display consensus
p_value as 3 digits

    #convert type of residue pairs into intger
    data['R1'] = data['R1'].astype(int)
    data['R2']= data['R2'].astype(int)

    #save consensus data into a new dataframe
    ConsD= data[['R1', 'R2', 'ConsP']]
    return ConsD



#Consensus three methods:

def Cons3(d, d1, d2):

    #first merge the input data into one datafram:
    # d, d1, d2 represent the input data:

    dfs=[d, d1, d2]
    df =pd.merge(dfs[0], dfs[1], on=['R1','R2'], how='outer')

    for d in dfs[2:]:
        data=pd.merge(df, d, on=['R1','R2'], how='outer')


    # padding p_value into zero and covert its type to float:
    data['P_x'].fillna(0, inplace = True)
    data['P_y'].fillna(0, inplace = True)
    data['P'].fillna(0, inplace = True)
    data['P_x'] = data['P_x'].astype(float)
    data['P_y']= data['P_y'].astype(float)
    data['P']= data['P'].astype(float)


    #Consensus prediction:
    # calculating mean of probabilities for each residue pairs from three
data
    data['ConsP'] = data[['P_x', 'P_y', 'P']].mean(axis=1)

    #padding consensus p_value into zero and convert its type to float
    data['ConsP'].fillna(0, inplace = True)
    pd.options.display.float_format = '{:.3f}'.format # display consensus
p_value as 3 digits


    # convert type of residue pairs into intger
    data['R1'] = data['R1'].astype(int)
    data['R2']= data['R2'].astype(int)

    #save consensus data into a new dataframe
    ConsD= data[['R1', 'R2', 'ConsP']]
    return ConsD
```

```
def main(argv):
    # reading input data and save them as dataframes:
    input1= subset(sys.argv[2])
    input2 = subset(sys.argv[3])
    input3= subset(sys.argv[4])


    # Appling consensus method :
    Cons2A = Cons2(input1[4], input2[4])
    Cons2B = Cons2(input1[4], input3[4])
    Cons2C = Cons2(input2[4],input3[4])
    Cons3M = Cons3(input1[4], input2[4], input3[4])


    # Categrizing consensus data into subsets(Top10, L/5, L/2, L, FL):
    # This step was repeated for each consensus method:
    # the fifth argument in command line will be the sequence length:

    sets1 = sets(int(sys.argv[5]), Cons3M, 'ConsP')

    # Top10

    Pred_top10 = sets1[0]
    top10= Pred_top10[['R1', 'R2']]

    # L/5 set
    Pred_L5= sets1[1]
    L5= Pred_L5[['R1', 'R2']]

    # L/2 set:
    Pred_L2 =sets1[2]
    L2 = Pred_L2[['R1', 'R2']]


    # L set:
    Pred_L = sets1[3]
    L = Pred_L[['R1', 'R2']]

    #FL set:
    Pred_FL = sets1[4]
    FL= Pred_FL[['R1', 'R2']]


    #save prediction data as rr format file for ConEva tool:

    #Cons2A
    Cons2A['D_min'] ='0'
    Cons2A['D_max']= '8'
    Cons2A=Cons2A[['R1', 'R2', 'D_min', 'D_max', 'ConsP']]


    dirA= 'file path'
    fileA='{}.rr'.format(str(sys.argv[6]))
    file_path_A=os.path.join(dirA, fileA)

    with open(file_path_A, 'w') as fa:
        fa.write(Cons2A.to_string(header = False, index= False))
```

```
    #Cons2B
    Cons2B['D_min'] ='0'
    Cons2B['D_max']= '8'
    Cons2B=Cons2B[['R1', 'R2', 'D_min', 'D_max', 'ConsP']]

    dirB= 'file path'
    fileB='{}.rr'.format(str(sys.argv[6]))
    file_path_B=os.path.join(dirB, fileB)

    with open(file_path_B, 'w') as fb:
        fb.write(Cons2B.to_string(header = False, index= False))

    #Cons2C:

    Cons2C['D_min'] ='0'
    Cons2C['D_max']= '8'
    Cons2C=Cons2C[['R1', 'R2', 'D_min', 'D_max', 'ConsP']]

    dirC= 'file path'
    fileC='{}.rr'.format(str(sys.argv[6]))
    file_path_C=os.path.join(dirC, fileC)

    with open(file_path_C, 'w') as fc:
        fc.write(Cons2C.to_string(header = False, index= False))


    #Cons3:
    Cons3M['D_min'] ='0'
    Cons3M['D_max']= '8'
    Cons3M=Cons3M[['R1', 'R2', 'D_min', 'D_max', 'ConsP']]

    dir3M= 'file path'
    file3M='{}.rr'.format(str(sys.argv[6]))
    file_path_3M=os.path.join(dir3M, file3M)

    with open(file_path_3M, 'w') as f3m:
        f3m.write(Cons3M.to_string(header = False, index= False))



 # Third step:
    # Evaluation measures (Precision, Recall, F1_measure):
    # A- merge native contact data with predicted data based on the residue
pairs of both data for compraison:


    # native contact with top 10 set of predicted contact data:
    F0= pd.merge(native, Pred_top10, on= ['R1', 'R2'], how='right' )


    F0['R2'].fillna(0, inplace = True)
    F0['R2'] = F0['R2'].astype(int)
    F0['ConsP'].fillna(0, inplace = True)
```

```
    F0['ConsP'] = F0['ConsP'].astype(float)
    F0['true_class'].fillna(0, inplace = True)
    F0['true_class'] = F0['true_class'].astype(int)


    #native contact with L/5 set of predicted contact data:
    F1 = pd.merge(native, Pred_L5, on=['R1', 'R2'], how='right')

    F1['R2'].fillna(0, inplace = True)
    F1['R2'] = F1['R2'].astype(int)
    F1['ConsP'].fillna(0, inplace = True)
    F1['ConsP'] = F1['ConsP'].astype(float)
    F1['true_class'].fillna(0, inplace = True)
    F1['true_class'] = F1['true_class'].astype(int)


    # save consensus prediction data in another file for Precision-Recall
curve analysis
    F_1 =F1[['ConsP', 'true_class']]
    dirB= 'file path'
    fileB='{}.csv'.format(str(sys.argv[3]))
    file_path_B= os.path.join(dirB, fileB)
    F_1.to_csv(file_path_B, index=False)


    # native contact with L/2 set of predicted contact data:

    F2 = pd.merge(native, Pred_L2, on=['R1', 'R2'],  how='right')


    F2['R2'].fillna(0, inplace = True)
    F2['R2'] = F2['R2'].astype(int)
    F2['ConsP'].fillna(0, inplace = True)
    F2['ConsP'] = F2['ConsP'].astype(float)
    F2['true_class'].fillna(0, inplace = True)
    F2['true_class'] = F2['true_class'].astype(int)


      #native contact with L set:
    F3 = pd.merge(native, Pred_L, on= ['R1', 'R2'], how = 'right')


    F3['R2'].fillna(0, inplace = True)
    F3['R2'] = F3['R2'].astype(int)
    F3['ConsP'].fillna(0, inplace = True)
    F3['ConsP'] = F3['ConsP'].astype(float)
    F3['true_class'].fillna(0, inplace = True)
    F3['true_class'] = F3['true_class'].astype(int)


    # native contact with FL set:
    F4 = pd.merge(native, Pred_FL, on= ['R1', 'R2'], how = 'right')


    F4['R2'].fillna(0, inplace = True)
    F4['R2'] = F4['R2'].astype(int)
    F4['ConsP'].fillna(0, inplace = True)
```

```
    F4['ConsP'] = F4['ConsP'].astype(float)
    F4['true_class'].fillna(0, inplace = True)
    F4['true_class'] = F4['true_class'].astype(int)


     # save consensus prediction data in another file for Precsion-Recall
curve analysis
    F_4 = F4[['ConsP', 'true_class']]

    dirB= 'file path'
    fileB='{}.csv'.format(str(sys.argv[3]))
    file_path_B=os.path.join(dirB, fileB)
    F_4.to_csv(file_path_B, index=False)


     # Classification consensus prediction data into contact and noncontact
at p-value >0:

    F0['Cor'] = (F0['ConsP'] > 0).astype(int)
    F1['Cor'] = (F1['ConsP'] > 0).astype(int)
    F2['Cor'] = (F2['ConsP'] > 0).astype(int)
    F3['Cor'] = (F3['ConsP'] > 0).astype(int)
    F4['Cor'] = (F4['ConsP'] > 0).astype(int)


     # C- Evaluation process:
       #Make confusion matrix for each subsets of predicted contact data to
calculate tp, tn, fp, fn values:
       # Computing Precision, Recall and F1_measure:
       # depending on CASP assessores evaluation of contact prediction
method:
       # Precision = TP of subset / len of predicted contact set:
       # Recall = TP of subset / len of native contact data:
       # f1_score = 2 * Precision* Recall/ (Precision + Recall):

    # top 10 set:
    scores_top10= Scores(RR_native, top10, F0.Cor, F0.true_class, Nc_N)
    df_top10 =pd.DataFrame([scores_top10], columns=['TP','FP', 'FN', 'TN',
'Precision', 'Recall', 'f1_score'], index =['Top10'], dtype=int)
    df_top10.index.name = 'set'

    dir_top10='file path'
    file_top10='{}.csv'.format(str(sys.argv[6]))
    file_top10_path=os.path.join(dir_top10, file_top10)
    df_top10.to_csv(file_top10_path)


    #L/5 set:
    scores_L5= Scores(RR_native, L5, F1.Cor, F1.true_class, Nc_N)
    df_L5 =pd.DataFrame([scores_L5], columns=['TP','FP', 'FN', 'TN',
'Precision', 'Recall', 'f1_score'], index=['L/5'], dtype=int)
    df_L5.index.name ='set'


    dir_L5='file path'
    file_L5='{}.csv'.format(str(sys.argv[6]))
    file_L5_path=os.path.join(dir_L5, file_L5)
```

```
    df_L5.to_csv(file_L5_path)

    # L/2 set:
    scores_L2= Scores(RR_native, L2, F2.Cor, F2.true_class, Nc_N)
    df_L2 =pd.DataFrame([scores_L2], columns=['TP','FP', 'FN', 'TN',
'Precision', 'Recall', 'f1_score'], index = ['L/2'], dtype=int)
    df_L2.index.name = 'set'

    dir_L2='file path'
    file_L2='{}.csv'.format(str(sys.argv[6]))
    file_L2_path=os.path.join(dir_L2, file_L2)
    df_L2.to_csv(file_L2_path, index=False)


    # L set:
    scores_L= Scores(RR_native, L, F3.Cor, F3.true_class, Nc_N)
    df_L =pd.DataFrame([scores_L], columns=['TP','FP', 'FN', 'TN',
'Precision', 'Recall', 'f1_score'], index= ['L'], dtype=int)
    df_L.index.name = 'set'

    dir_L='file path'
    file_L='{}.csv'.format(str(sys.argv[6]))
    file_L_path=os.path.join(dir_L, file_L)
    df_L.to_csv(file_L_path, index=False)

    #FL set:
    scores_FL = Scores(RR_native, FL, F4.Cor, F4.true_class, Nc_N)
    df_FL =pd.DataFrame([scores_FL], columns=['TP','FP', 'FN', 'TN',
'Precision', 'Recall', 'f1_score'], index=['FL'], dtype=int)
    df_FL.index.name = 'set'

    dir_FL='file path'
    file_FL='{}.csv'.format(str(sys.argv[6]))
    file_FL_path=os.path.join(dir_FL, file_FL)
    df_FL.to_csv(file_FL_path, index=False)

    #save the scores evaluation into dataframe:
    df =pd.DataFrame([scores_top10, scores_L5, scores_L2, scores_L,
scores_FL], columns=['TP','FP', 'FN', 'TN', 'Precision', 'Recall',
'f1_score'], index=['Top10', 'L/5', 'L/2', 'L', 'FL'],  dtype=int)
    df.index.name = 'sets'


    dir_B='file path'
    file_B='{}.csv'.format(str(sys.argv[6]))
    file_B_path=os.path.join(dir_B, file_B)
    df.to_csv(file_B_path, index=False)
    print(df)


if __name__ =='__main__':
    main(sys.argv)
```

Appendix 2

Table S.1. Mean Precision Scores of individual methods compared with consensus methods on 31 FM domains of CASP13. The scores were measured for long-range on contact subset lists; Top10, L/5, L/2, L, FL, where L represents the sequence length. Top10 set includes 10 amino acid residue pairs that have the highest probability values of contacts. L/5 set contains contact scores of residue pairs within 20 % of the sequence, whereas the L/2 set has predicted scores of contacts for residue pairs within 50 % of sequences. L set contains all predicted scores of residue pairs within sequence length. FL is a full contact prediction dataset.

Method	Top10	L/5	L/2	L	FL
Zhang_Contact (G036)	63.23	57.38	48.87	38.81	2.31
ZHOU-Contact (G189)	65.48	58.90	48.29	37.52	2.41
DMP (G491)	68.71	60.80	47.67	37.18	**7.35**
ConsA (G189 & G491)	**72.26**	64.83	51.69	39.72	2.36
ConsB (G189 & G036)	70.65	63.18	52.60	41.34	2.27
ConsC (G491 & G036)	69.35	64.83	52.89	41.52	2.27
Cons3 (All)	70.97	**65.98**	**55.12**	**42.59**	2.27

Appendix 3

Table S.2. Mean precision scores of individual methods compared with consensus methods on 31 FM domains of CASP13 using ConEVA. The scores were measured for long-range on contact subset lists: L/5, L/2, L, where L represents the sequence length. Top10 set includes 10 amino acid residue pairs that have the highest probability values of contacts. L/5 set contains contact scores of residue pairs within 20 % of the sequence, whereas the L/2 set has predicted scores of contacts for residue pairs within 50 % of sequences. L set contains all predicted scores of residue pairs within sequence length. FL is a full contact prediction dataset.

Method	L/5	L/2	L
Zhang_Contact (G036)	57.17	48.82	38.66
ZHOU-Contact (G189)	58.53	48.13	37.37
DMP (G491)	61.17	47.61	37.05
ConsA (G189 & G491)	66.54	53.05	40.64
ConsB (G189 & G036)	64.97	53.94	42.34
ConsC (G491 & G036)	67.61	54.96	42.75
Cons3 (All)	**67.96**	**56.66**	**43.61**

Appendix 4

Table S.3. P-values of mean precision for L/5, L/2, and L long-range contact prediction of CASP13 target domains (FM). L/5 set contains contact scores of residue pairs within 20 % of the sequence, whereas the L/2 set has predicted scores of contacts for residue pairs within 50 % of sequences. L set contains all predicted scores of residue pairs within sequence length.

Method	ConsA	ConsB	ConsC	Cons3
	L/5 long-range contacts			
Zhang_Contact	**0.011**	**0.004**	**0.001**	**0.001**
Zhou-Contact	**0.008**	**0.013**	**0.007**	**0.001**
DMP	**0.038**	0.369	**0.040**	**0.052**
	L/2 long-range contacts			
Zhang_Contact	**0.026**	**0.001**	**0.000**	**0.000**
Zhou-Contact	**0.013**	**0.011**	**0.009**	**0.000**
DMP	**0.005**	**0.051**	**0.005**	**0.000**
	L long-range contacts			
Zhang_Contact	0.064	**0.000**	**0.000**	**0.000**
Zhou-Contact	**0.001**	**0.005**	**0.003**	**0.000**
DMP	**0.003**	**0.006**	**0.000**	**0.000**

Appendix 5

Table S.4. Mean f1_score Scores of individual methods compared with consensus methods on 31 FM domains of CASP13. The scores were measured for long-range on contact subset lists; Top10, L/5, L/2, L, FL, where L represents the sequence length. Top10 set includes 10 amino acid residue pairs that have the highest probability values of contacts. L/5 set contains contact scores of residue pairs within 20 % of the sequence, whereas the L/2 set has predicted scores of contacts for residue pairs within 50 % of sequences. L set contains all predicted scores of residue pairs within sequence length. FL is a full contact prediction dataset.

Method	Top10	L/5	L/2	L	FL
Zhang_Contact (G036)	8.52	18.46	29.94	35.39	4.47
ZHOU-Contact (G189)	8.38	18.78	29.38	33.92	4.67
DMP (G491)	8.79	18.88	28.70	33.54	**12.44**
ConsA (G189 & G491)	**9.71**	20.80	31.83	36.24	4.58
ConsB (G189 & G036)	9.52	20.08	32.76	38.02	4.39
ConsC (G491 & G036)	9.39	21.12	33.05	38.30	4.39
Cons3 (All)	9.55	**21.23**	**34.47**	**39.38**	4.39

Appendix 6

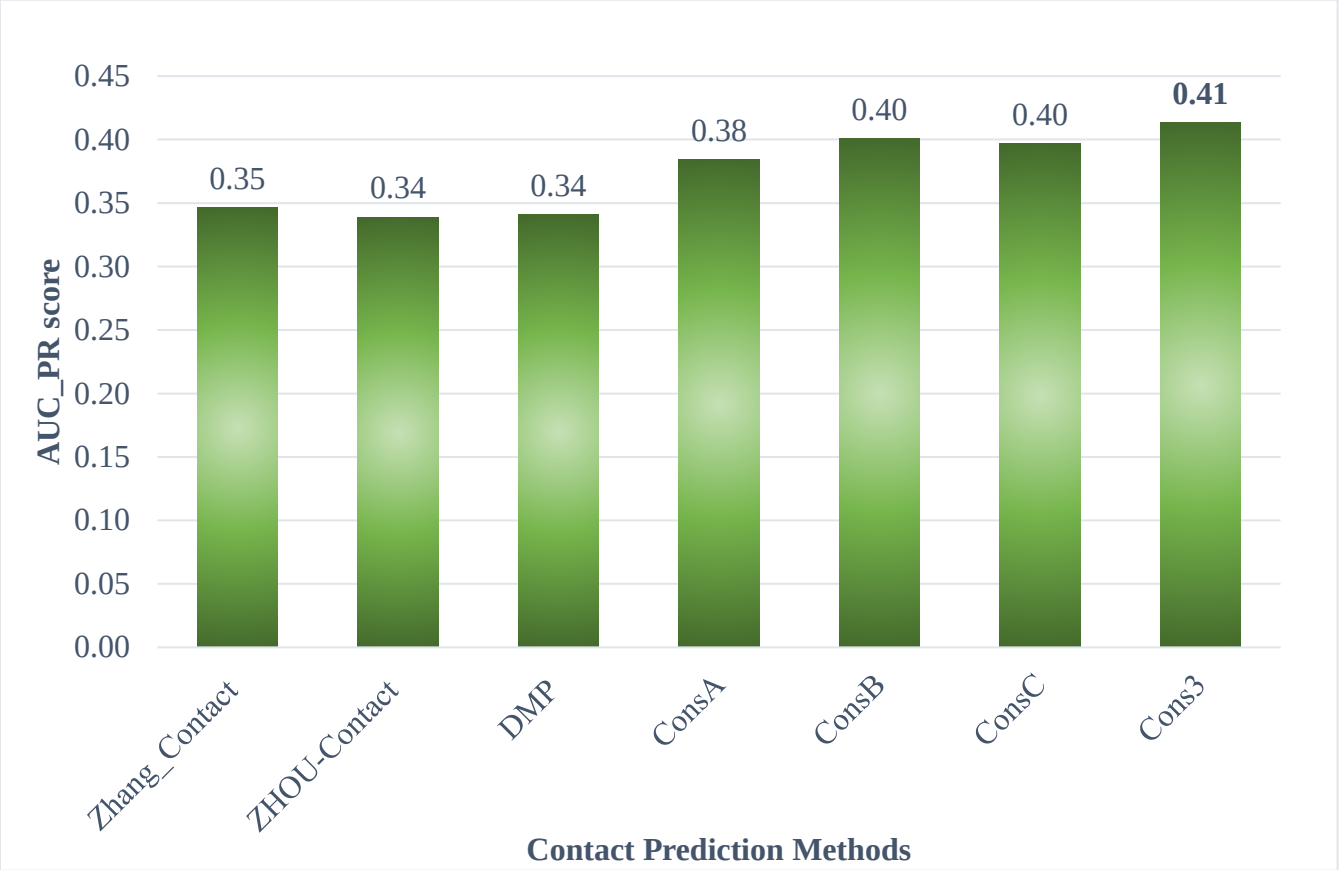


Figure S.1. A comparison of consensus and individual methods on FL long-range contact sets based on AUC_PR score of Precision-Recall curve analysis for CASP13 on 31FM domains.

Appendix 7

Table S.5. AUC_PR scores of individual methods compared with consensus methods on 31 FM domains of CASP13. The scores were measured for long-range on contact subset FL, which is a full contact prediction dataset. The AUC_PR scores were calculated in two different ways. AUC_PR represent the scores of the prediction methods based on the contact prediction for all 31 FM targets. Average AUC_PR represent the scores of prediction methods based on the AUC of all targets for each method. AUC_PR of the random classifier is for each PR curve analysis of each method.

CASP13 methods	AUC_PR	AUC_PR of a random classifier	Average AUC_PR
Zhang_Contact (G036)	0.37	0.02	0.35
ZHOU-Contact (G189)	0.43	0.02	0.34
DMP (G491)	0.50	0.05	0.34
ConsA (G189 & G491)	0.48	0.02	0.38
ConsB (G189 & G036)	0.47	0.02	0.40
ConsC (G491 & G036)	0.46	0.02	0.40
Cons3 (All)	0.49	0.02	0.41

Appendix 8

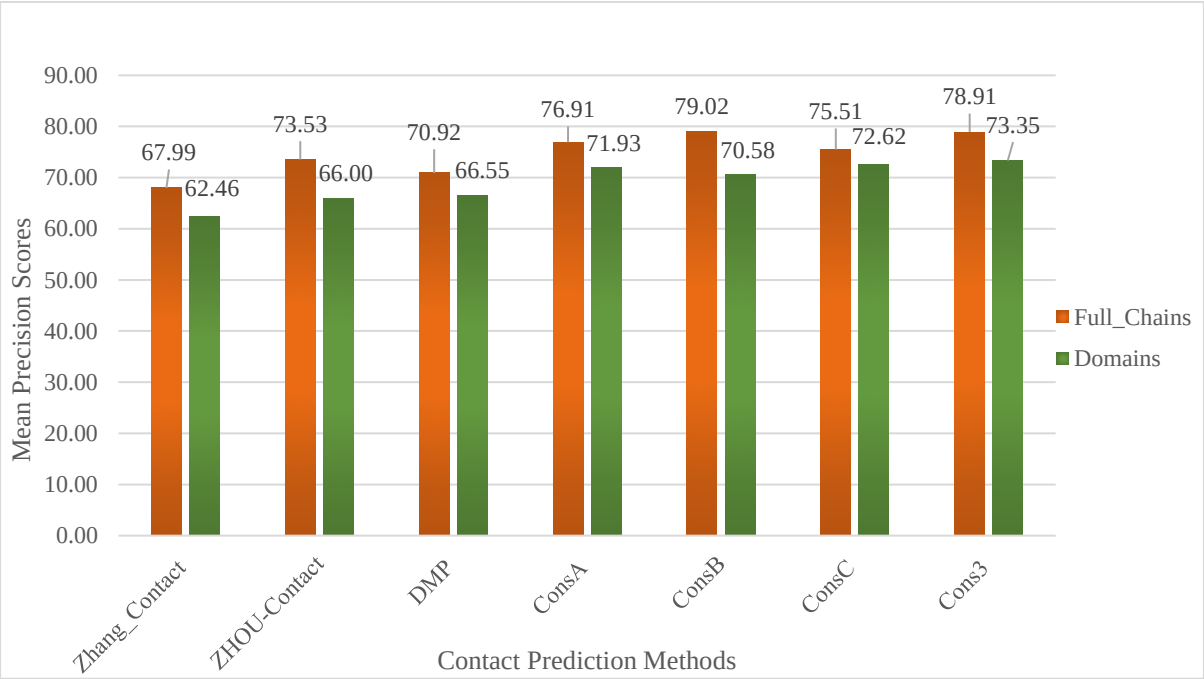


Figure S.2. Mean precision scores of predicted contacts for domains and full chains of CASP13 targets on L/5 long-range contacts for 35 full chains & 43 domains-ConEVA tool.

Appendix 9

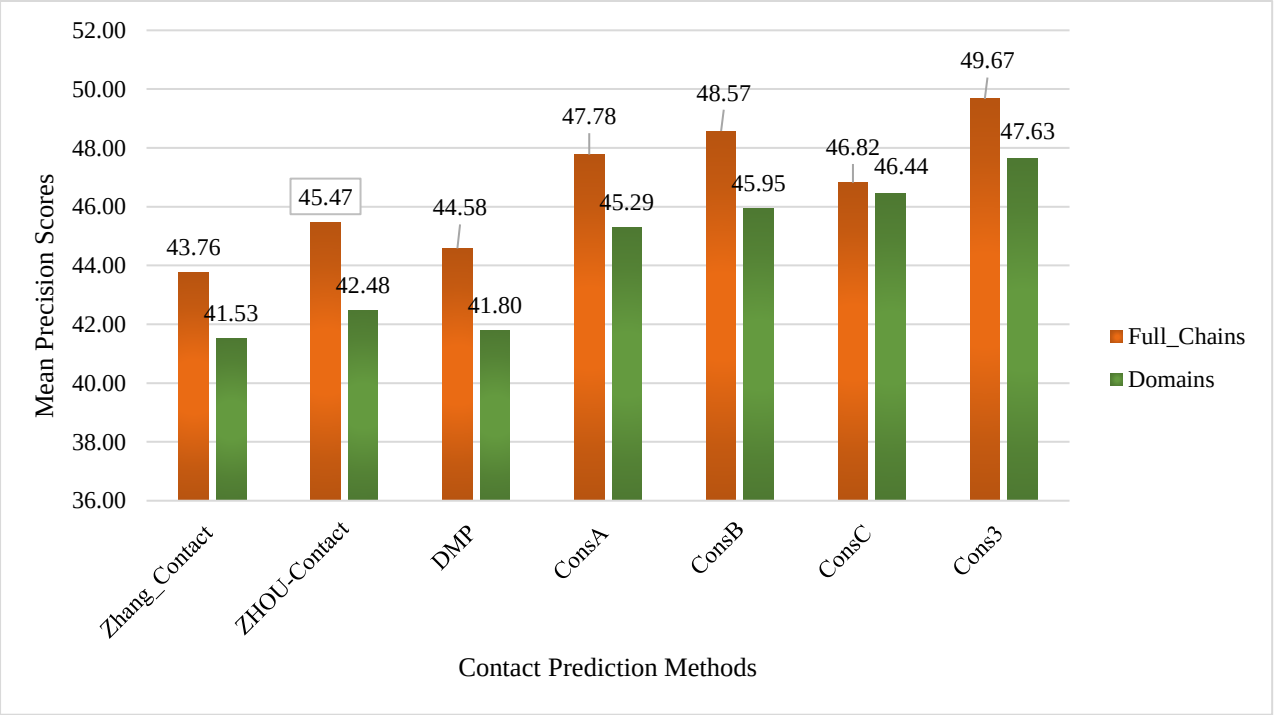


Figure S.3. Mean precision scores of predicted contacts for domains and full chains of CASP13 targets on L long-range contacts for 35 full chains & 43 domains- ConEVA tool.

Appendix 10

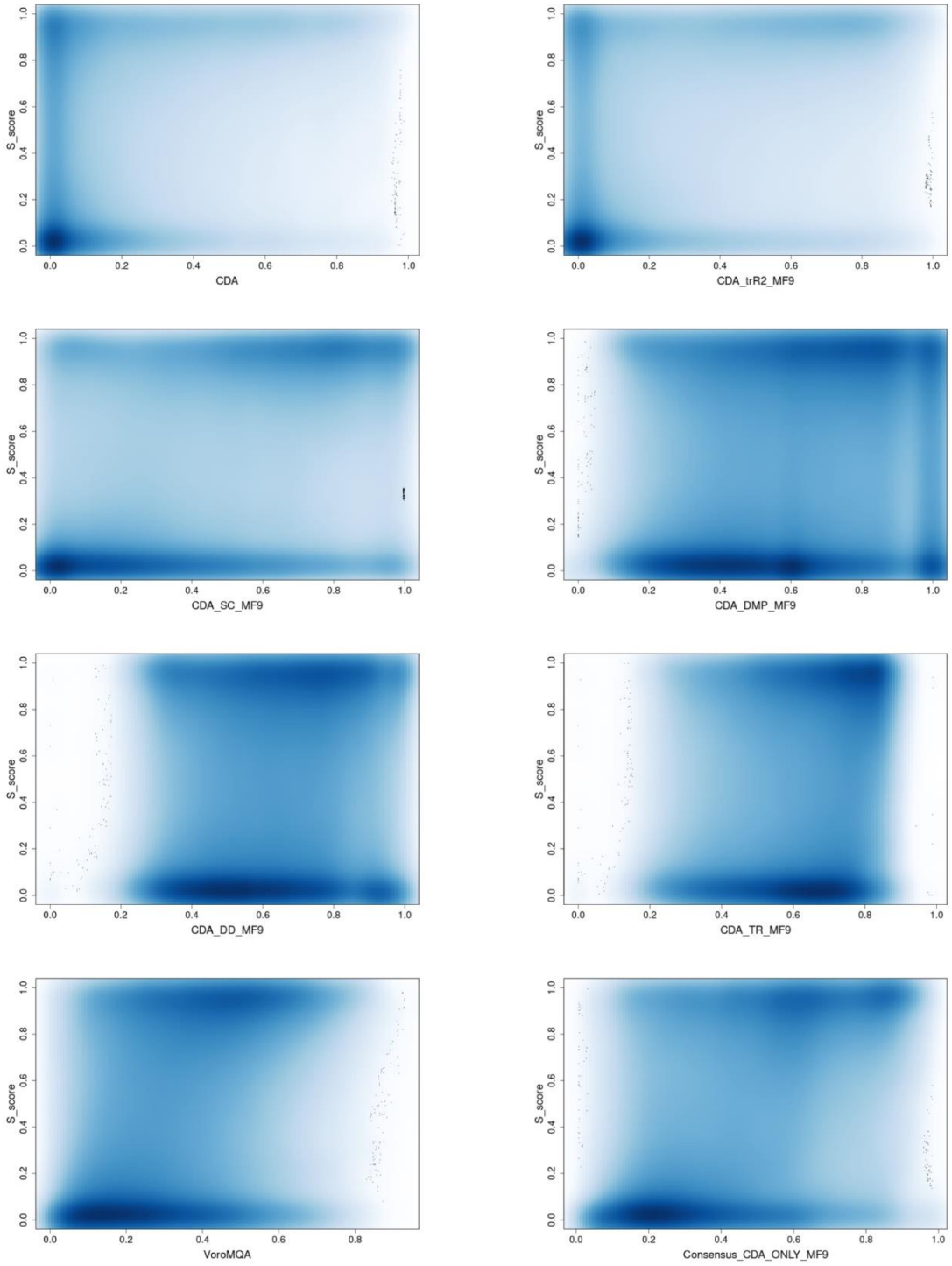


Figure S.4. Density scatter plots show the relationship between ModFOLD9 and its component methods according to S-scores.

Appendix 11

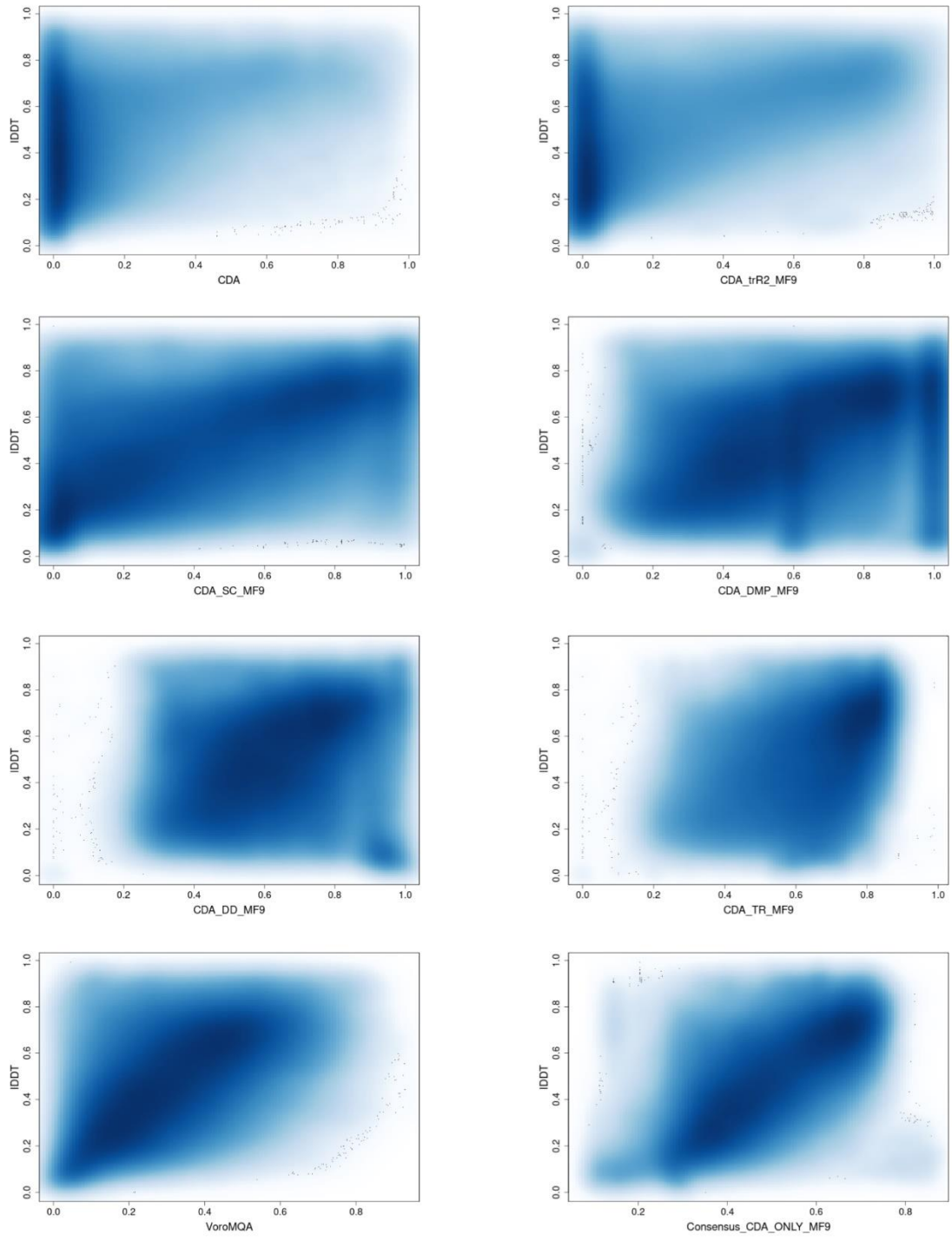


Figure S.5. Density scatter plots show the relationship between ModFOLD9 and its component methods according to lDDT scores.

Appendix 12

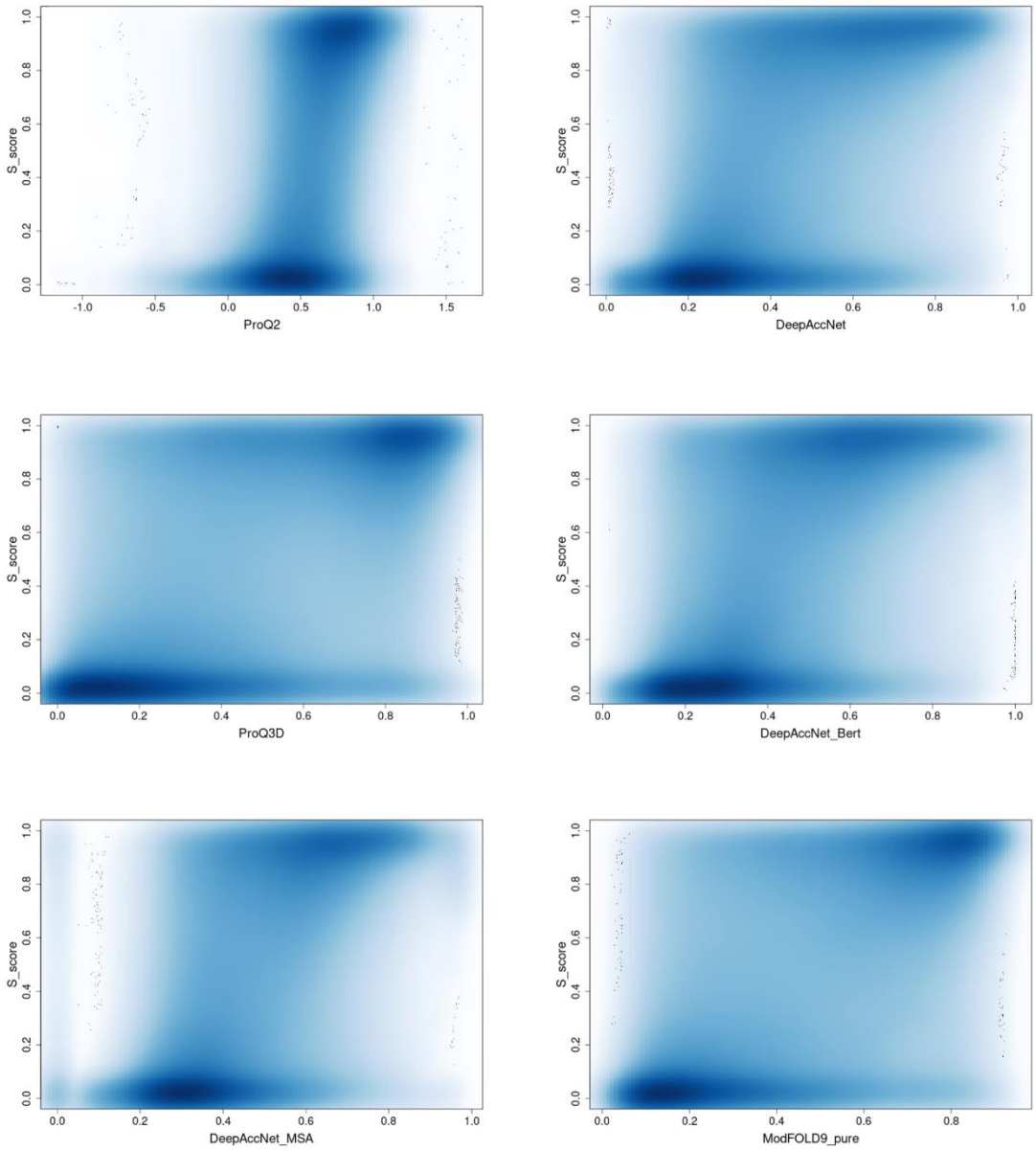


Figure S.6. Density scatter plots show the relationship between ModFOLD9_pure and its five top component methods according to S-scores.

Appendix 13

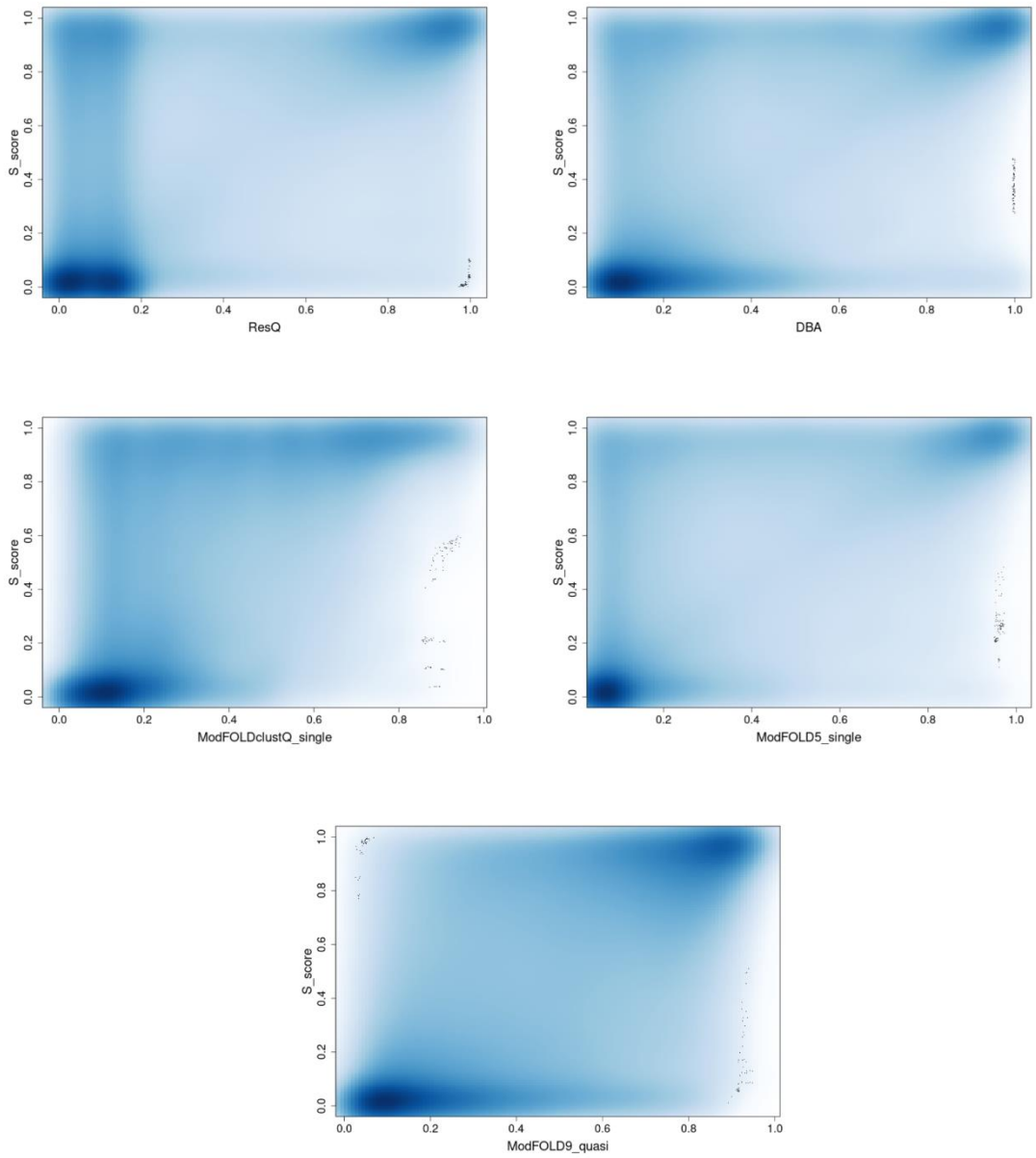


Figure S.7. Density scatter plots show the relationship between ModFOLD9_quasi and the quasi-single model methods according to S-scores.

Appendix 14

The common subset code for the CAMEO dataset

```
#!/usr/bin/env python3
import os
import shutil
import sys
from glob import glob
import pandas as pd
import numpy as np
import json


#First, import all predicted models of targets from CAMEO
dataset. Append number to file names as all files have the
same name (qalddt):

for number, filename in
enumerate(glob('/home/ky820206/Desktop/CAMEO_QA/6_months/raw_d
ata/{}'.format(sys.argv[1]))):# sys.argv[1] is the folder name
written in the commond line
   try:
       os.rename(filename, "qalddt_{0}".format(number))
   except OSError as e:
       print('Something happend:', e)


#Ref:https://stackoverflow.com/questions/12336594/trying-to-
rename-files-with-glob-and-os-modules

#####

# Second: Rename files with target ID after extracting it from
file content

files = glob.glob('qalddt_*')
for file in files:
    with open(file) as f:
        d = json.load(f)
        for k, v in d.items():
            if k == "unique_id":
                df = d[k]
                if df[7:10] == '100' or df[7:10] == '106':
```

```
                    new_name = df[0:12]
                    print(new_name)
                    os.rename(file, new_name)
                else:
                    new_name = df[0:11]
                    os.rename(file, new_name)

#####

# Third: The code will read raw data containing all targets ID
and save it as a dataframe in a CSV file.

listdirs=
os.listdir('/home/ky820206/Desktop/CAMEO_QA/6_months/quality_e
stimation/{}'.format(sys.argv[1]))
targets =[]
for item in listdirs:
    if "_1" in item:
        targets.append(item)

df = pd.DataFrame(np.array([targets]).T)
df.columns = ['Target']
print(df)
df.to_csv('{}.csv'.format(str(sys.argv[1])), index=False)

#####

#Fourth: Repeat step three with each quality estimation method
file to obtain their predicted models of the targets.


dirListing =
os.listdir('/home/ky820206/Desktop/CAMEO_QA/6_months/Component
_methods/data/{}'.format(sys.argv[2]))

editFiles = []
for item in dirListing:
    if "_1" in item:
        editFiles.append(item)
#
print(editFiles)
f = pd.DataFrame(np.array([editFiles]).T)
f.columns = [str(sys.argv[2])]
print(f)
f.to_csv('{}.csv'.format(str(sys.argv[2])))

#####

#Fifth: read all the file have the whole list of targets ID
and the predicted models
df1 = pd.read_csv(sys.argv[1])
```

```
df2 = pd.read_csv(sys.argv[2])
df3 = pd.read_csv(sys.argv[3])
df4 = pd.read_csv(sys.argv[4])
df5 = pd.read_csv(sys.argv[5])
df6 = pd.read_csv(sys.argv[6])
df7 = pd.read_csv(sys.argv[7])
df8 = pd.read_csv(sys.argv[8])
df9 = pd.read_csv(sys.argv[9])

#####

#Sixth: merge the targets with the predicted models of quality
estimation methods in one dataframe based on the target IDs:

merged_1 = pd.merge(df1, df2, how='left', left_on='Target',
right_on='ProQ2')
merged_2 = pd.merge(merged_1, df3, how='left',
left_on='Target', right_on='ProQ3')
merged_3 = pd.merge(merged_2, df4, how='left',
left_on='Target', right_on='ProQ3D')
merged_4 = pd.merge(merged_3, df5, how='left',
left_on='Target', right_on='ProQ3D_LDDT')
merged_5 = pd.merge(merged_4, df6, how='left',
left_on='Target', right_on='VoroMQA_sw5')
merged_6 = pd.merge(merged_5, df7, how='left',
left_on='Target', right_on='VoroMQA_v2')
merged_7 = pd.merge(merged_6, df8, how='left',
left_on='Target', right_on='ModFOLD9')
merged_8 = pd.merge(merged_7, df9, how='left',
left_on='Target', right_on='ModFOLD9_pure')

dataset = merged_4.drop(['Unnamed: 0_y', 'Unnamed: 0_x',
'Unnamed: 0'], axis=1)
data = dataset.dropna()
data.columns = ['Target', 'ProQ2', 'ProQ3', 'ProQ3D',
'ProQ3D_LDDT', 'VoroMQA_sw5', 'VoroMQA_v2', 'ModFOLD9',
'ModFOLD9_pure']
##
data.set_index('Target', inplace=True)
print(data)
data.to_csv('commonsubset.csv', index=True)

#save the common target IDs in one file for the next step:

target = list(data['ModFOLD9'])

#save targets in txt file
with open('targetlist.txt', 'w') as h:
    for x in target:
        h.write(str(x)+'\n')
```

```
#####

#In the last step, we remove the uncommon targets which could
not be predicted by other methods based on the common subset
data frame.

#We implemented this step after commenting on the previous
stpes.

with open('targetlist.txt', 'r') as m:
     lines = set((line.rstrip('\n') for line in
m.readlines()))
#     print(lines)
#


for root, dirs, files in
os.walk('/home/ky820206/Desktop/CAMEO_QA/Component_methods/{}'
.format(sys.argv[1])):
    for name in files:
        path = os.path.join(root, name)
        if os.path.isfile(path):
            if name not in (lines):
                print(name)
                os.remove(path)
            else:
                pass
```

Appendix 15

The R code of ROC analysis on CAMEO common subset data

```
library('RColorBrewer')
library('ROCR')


MF9 <- read.csv('ModFOLD9.csv')
MF9_pure <- read.csv('ModFOLD9_pure.csv')
ProQ2<- read.csv('ProQ2.csv')
ProQ3 <- read.csv('ProQ3.csv')
ProQ3D <- read.csv('ProQ3D.csv')
ProQ3D_lDDT <- read.csv('ProQ3D_LDDT.csv')
VoroMQA_sw5 <- read.csv('VoroMQA_sw5.csv')
VoroMQA_v2 <- read.csv('VoroMQA_v2.csv')


#ROC AUC and ROC AUC FPR <= 0.1 calculations:

MF9_auc <- performance(prediction(MF9$pred, MF9$X0),
'auc')@y.values[[1]]
MF9_auc_0_1 <- performance(prediction(MF9$pred, MF9$X0),
'auc', fpr.stop=0.1)@y.values[[1]]

MF9_pure_auc <- performance(prediction(MF9_pure$pred,
MF9_pure$X0), 'auc')@y.values[[1]]
MF9_pure_auc_0_1 <- performance(prediction(MF9_pure$pred,
MF9_pure$X0), 'auc', fpr.stop=0.1)@y.values[[1]]

ProQ2_auc <- performance(prediction(ProQ2$pred, ProQ2$X0),
'auc')@y.values[[1]]
ProQ2_auc_0_1 <- performance(prediction(ProQ2$pred, ProQ2$X0),
'auc', fpr.stop=0.1)@y.values[[1]]

ProQ3_auc <- performance(prediction(ProQ3$pred, ProQ3$X0),
'auc')@y.values[[1]]
ProQ3_auc_0_1 <- performance(prediction(ProQ3$pred, ProQ3$X0),
'auc', fpr.stop=0.1)@y.values[[1]]

ProQ3D_auc <- performance(prediction(ProQ3D$pred, ProQ3D$X0),
'auc')@y.values[[1]]
ProQ3D_auc_0_1 <- performance(prediction(ProQ3D$pred,
ProQ3D$X0), 'auc', fpr.stop=0.1)@y.values[[1]]

ProQ3D_lDDT_auc <- performance(prediction(ProQ3D_lDDT$pred,
ProQ3D_lDDT$X0), 'auc')@y.values[[1]]
ProQ3D_lDDT_auc_0_1 <-
performance(prediction(ProQ3D_lDDT$pred, ProQ3D_lDDT$X0),
'auc', fpr.stop=0.1)@y.values[[1]]
```

```
VoroMQA_sw5_auc <- performance(prediction(VoroMQA_sw5$pred,
VoroMQA_sw5$X0), 'auc')@y.values[[1]]
VoroMQA_sw5_auc_0_1 <-
performance(prediction(VoroMQA_sw5$pred, VoroMQA_sw5$X0),
'auc', fpr.stop=0.1)@y.values[[1]]


VoroMQA_v2_auc <- performance(prediction(VoroMQA_v2$pred,
VoroMQA_v2$X0), 'auc')@y.values[[1]]
VoroMQA_v2_auc_0_1 <- performance(prediction(VoroMQA_v2$pred,
VoroMQA_v2$X0), 'auc', fpr.stop=0.1)@y.values[[1]]
print(ProQ2_auc_0_1)


#ROC curves plot
png ('ROC_curves_Componenet_Methods_CAMEO_6_months.png', width
= 400, height=450)
plot(performance(prediction(MF9$pred, MF9$X0), 'tpr', 'fpr'),
col='#00703c', lwd=2)
plot(performance(prediction(MF9_pure$pred, MF9_pure$X0),
'tpr', 'fpr'), col='#377eb8', lwd=2, add=TRUE)
plot(performance(prediction(ProQ2$pred, ProQ2$X0), 'tpr',
'fpr'), col='#cda4de', lwd=2, add=TRUE)
plot(performance(prediction(ProQ3$pred, ProQ3$X0), 'tpr',
'fpr'), col='#ffec9e', lwd=2, add=TRUE)
plot(performance(prediction(ProQ3D$pred, ProQ3D$X0), 'tpr',
'fpr'), col='#d53e4f', lwd=2, add=TRUE)
plot(performance(prediction(ProQ3D_lDDT$pred, ProQ3D_lDDT$X0),
'tpr', 'fpr'), col='#f03900', lwd=2, add=TRUE)
plot(performance(prediction(VoroMQA_sw5$pred, VoroMQA_sw5$X0),
'tpr', 'fpr'), col='#6600a6', lwd=2, add=TRUE)
plot(performance(prediction(VoroMQA_v2$pred, VoroMQA_v2$X0),
'tpr', 'fpr'), col='#ad8b00', lwd=2, add=TRUE)
legend('bottomright', legend = c('ModFOLD9', 'ModFOLD9_pure',
'ProQ2', 'ProQ3', 'ProQ3D', 'ProQ3D_lDDT', 'VoroMQA_sw5',
'VoroMQA_v2'), col=c('#00703c', '#377eb8', '#cda4de',
'#ffec9e', '#d53e4f', '#f03900', '#6600a6', '#ad8b00' ),
lwd=4)

dev.off()

#ROC curves (FPR <= 0.1) plot
png
('ROC_curves_FPR_zoomed_Componenet_Methods_CAMEO_6_months.png'
, width = 400, height=450)
plot(performance(prediction(MF9$pred, MF9$X0), 'tpr', 'fpr'),
col='#00703c', lwd=2, xlim = c(0, 0.2), ylim = c(0, 0.8))
plot(performance(prediction(MF9_pure$pred, MF9_pure$X0),
'tpr', 'fpr'), col='#377eb8', lwd=2, add=TRUE)
```

```
plot(performance(prediction(ProQ2$pred, ProQ2$X0), 'tpr',
'fpr'), col='#cda4de', lwd=2, add=TRUE)
plot(performance(prediction(ProQ3$pred, ProQ3$X0), 'tpr',
'fpr'), col='#ffec9e', lwd=2, add=TRUE)
plot(performance(prediction(ProQ3D$pred, ProQ3D$X0), 'tpr',
'fpr'), col='#d53e4f', lwd=2, add=TRUE)
plot(performance(prediction(ProQ3D_lDDT$pred, ProQ3D_lDDT$X0),
'tpr', 'fpr'), col='#f03900', lwd=2, add=TRUE)
plot(performance(prediction(VoroMQA_sw5$pred, VoroMQA_sw5$X0),
'tpr', 'fpr'), col='#6600a6', lwd=2, add=TRUE)
plot(performance(prediction(VoroMQA_v2$pred, VoroMQA_v2$X0),
'tpr', 'fpr'), col='#ad8b00', lwd=2, add=TRUE)
legend('bottomright', legend = c('ModFOLD9', 'ModFOLD9_pure',
'ProQ2', 'ProQ3', 'ProQ3D', 'ProQ3D_lDDT', 'VoroMQA_sw5',
'VoroMQA_v2'), col=c('#00703c', '#377eb8', '#cda4de',
'#ffec9e', '#d53e4f', '#f03900', '#6600a6', '#ad8b00' ),
lwd=4)

dev.off()
```

Appendix 16

Table S.6. ROC AUC scores of the local assessment accuracy of ModFOLD9 performance and independent server based on its component quality methods. These scores are based on ROC AUC and ROC AUC FPR <= 0.1 analysis (with lDDT cutoff < 60) on common subset CAMEO data over four periods: one month, three months, six months, and one year. The bold scores refer to the highest AUC scores.

Time	Server	ROC AUC	ROC AUC FPR<=0.1	Time	Server	ROC AUC	ROC AUC FPR<=0.1
One month	ProQ2	0.860	0.051	Three months	ProQ2	0.842	0.047
	VoroMQA_sw5	0.838	0.039		VoroMQA_sw5	0.802	0.036
	VoroMQA_v2	0.885	0.051		VoroMQA_v2	0.865	0.048
	ProQ3lDDT	0.871	0.049		ProQ3lDDT	0.874	0.050
	ProQ3	0.874	0.054		ProQ3	0.863	0.051
	ProQ3D	0.852	0.047		ProQ3D	0.836	0.044
	ModFOLD9	**0.931**	**0.071**		ModFOLD9	**0.921**	**0.067**
	ModFOLD9_pure	0.921	0.069		ModFOLD9_pure	0.911	0.064
	Server	**ROC AUC**	**ROC AUC FPR<=0.1**		**Server**	**ROC AUC**	**ROC AUC FPR<=0.1**
Six months	ProQ2	0.855	0.049	One year	ProQ2	0.858	0.050
	VoroMQA_sw5	0.814	0.038		VoroMQA_sw5	0.811	0.038
	VoroMQA_v2	0.879	0.051		VoroMQA_v2	0.886	0.053
	ProQ3lDDT	0.891	0.054		ProQ3D_lDDT	0.899	0.056
	ProQ3	0.881	0.055		ProQ3	0.888	0.056
	ProQ3D	0.852	0.047		ProQ3D	0.858	0.048
	ModFOLD9	**0.929**	**0.070**		ModFOLD9	**0.933**	**0.070**
	ModFOLD9_pure	0.921	0.067		ModFOLD9_pure	0.924	0.068

Appendix 17

Table S.7. ROC AUC scores of the local assessment accuracy of ModFOLD9 performance along with its previous versions. These scores are based on ROC AUC and ROC AUC FPR <= 0.1 analysis (with lDDT cutoff < 60) on common subset CAMEO data over four periods: one month, three months, six months, and one year. The bold scores refer to the highest AUC scores.

Time	Server	ROC AUC	ROC AUC FPR<=0.1	Time	Server	ROC AUC	ROC AUC FPR<=0.1
One month	ModFOLD6	0.914	0.062	Three months	ModFOLD6	0.873	0.048
	ModFOLD7_lDDT	0.914	0.064		ModFOLD7_lDDT	0.887	0.054
	ModFOLD8	0.916	0.064		ModFOLD8	0.888	0.054
	ModFOLD9	**0.951**	**0.076**		ModFOLD9	**0.921**	**0.067**
	ModFOLD9_pure	0.940	**0.076**		ModFOLD9_pure	0.912	0.064
	Server	**ROC AUC**	**ROC AUC FPR<=0.1**		**Server**	**ROC AUC**	**ROC AUC FPR<=0.1**
Six months	ModFOLD6	0.860	0.039	One year	ModFOLD6	0.861	0.037
	ModFOLD7_lDDT	0.880	0.048		ModFOLD7_lDDT	0.882	0.049
	ModFOLD8	0.878	0.048		ModFOLD8	0.882	0.048
	ModFOLD9	**0.931**	**0.071**		ModFOLD9	**0.936**	**0.072**
	ModFOLD9_pure	0.921	0.068		ModFOLD9_pure	0.927	0.070

Appendix 18

Table S.8. ROC AUC scores of the local assessment accuracy of five leading quality assessment methods. These scores are based on ROC AUC and ROC AUC FPR <= 0.1 analysis (with lDDT cutoff < 60) on common subset CAMEO data over four periods: one month, three months, six months, and one year. The bold scores refer to the highest AUC scores.

Time	Server	ROC AUC	ROC AUC FPR<=0.1	Time	Server	ROC AUC	ROC AUC FPR<=0.1
One month	ModFOLD9	0.927	0.071	Three months	ZJUT-GraphCPLMQA	0.934	**0.076**
	QMEANDisCo3	0.924	0.067		DeepUMQA2	0.926	0.072
	DeepUMQA	0.913	0.064		ModFOLD9	0.918	0.066
	DeepUMQA2	**0.933**	**0.073**		ModFOLD9_pure	0.903	0.063
	ModFOLD9_pure	0.919	0.068		MEGA-Assessment	**0.938**	**0.076**
Time	**Server**	**ROC AUC**	**ROC AUC FPR<=0.1**	**Time**	**Server**	**ROC AUC**	**ROC AUC FPR<=0.1**
Six months	ZJUT-GraphCPLMQA	**0.947**	**0.079**	One year	ZJUT-GraphCPLMQA	**0.947**	**0.079**
	DeepUMQA2	0.945	0.075		DeepUMQA2	0.949	0.075
	ModFOLD9	0.935	0.071		ModFOLD9	0.935	0.071
	ModFOLD9_pure	0.926	0.068		ModFOLD9_pure	0.927	0.069
	DeepUMQA	0.925	0.065		DeepUMQA	0.929	0.066