Application of coevolution-based methods and deep learning for structure prediction of protein complexes

Nikita Desai

Bioinformatics Group
Department of Computer Science
University College London

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

August 2022
I, Nikita Desai, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.
Abstract

The three-dimensional structures of proteins play a critical role in determining their biological functions and interactions. Experimental determination of protein and protein complex structures can be expensive and difficult. Computational prediction of protein and protein complex structures has therefore been an open challenge for decades. Recent advances in computational structure prediction techniques have resulted in increasingly accurate protein structure predictions. These techniques include methods that leverage information about coevolving residues to predict residue interactions and that apply deep learning techniques to enable better prediction of residue contacts and protein structures. Prior to the work outlined in this thesis, coevolution-based methods and deep learning had been shown to improve the prediction of single protein domains or single protein chains.

Most proteins in living organisms do not function on their own but interact with other proteins either through transient interactions or by forming stable protein complexes. Knowledge of protein complex structures can be useful for biological and disease research, drug discovery and protein engineering. Unfortunately, a large number of protein complexes do not have experimental structures or close homolog structures that can be used as templates. In this thesis, methods previously developed and applied to the de novo prediction of single protein domains or protein monomer chains were modified and leveraged for the prediction of protein heterodimer and homodimer complexes. A number of coevolution-based tools and deep learning methods are explored for the purpose of predicting inter-chain and intra-chain residue contacts in protein dimers. These contacts are combined with existing protein docking methods to explore the prediction of homodimers and heterodimers.

Overall, the work in this thesis demonstrates the promise of leveraging coevolution and deep-learning for the prediction of protein complexes, shows improvements in protein complex prediction tasks achieved using coevolution based methods and deep learning methods, and demonstrates remaining challenges in protein complex prediction.
Impact Statement

Most of the biological processes essential to life involve proteins or protein complexes. Protein complexes are made up of two or more interacting protein chains and constitute the majority of cellular and biological processes. Accurate structure prediction of protein monomers and the interfaces between interacting monomers can be critical to understanding the biological function of complexes. Accurate interface characterization can therefore be important for biological and disease research, drug discovery, and protein engineering.

Experimental determination of protein complexes can be expensive, time-consuming, and difficult. Computational complex structure prediction can therefore be very useful, either eliminating the need for expensive and difficult experiments or adding information to existing experimental data. In this thesis, computational prediction of protein dimer complexes, which are made up of two interacting protein subunits, is explored. Two computational techniques for protein complex interface prediction are explored: coevolution-based methods and deep-learning based methods.

Within academia, the work in this thesis contributes to overall knowledge in the area of structural bioinformatics by demonstrating the benefits and limitations of leveraging deep learning and coevolution for interface prediction. The work in this thesis demonstrates that improved structure prediction of protein subunits greatly contributes to the overall accuracy of protein docking, i.e., predicting the interfaces of interacting protein subunits. The work in this thesis also outlines further research areas to improve overall dimer structure prediction. A number of tools were developed to perform dimer interface prediction. These tools can be used to predict interacting residues in protein complexes and can also be compared to other tools developed for that purpose. Overall, improved protein complex structure prediction can contribute to other fields of biological research where knowledge of protein complexes and interfaces can contribute to increased knowledge of complex function, understanding of mechanisms of action of protein complexes, and understanding of how mutations that cause changes on the surface of protein monomers may impact protein interfaces.
Outside of academia, improved protein complex prediction can have important implications in commercial areas such as drug discovery or development and protein engineering. In the field of drug discovery, understanding protein interfaces can be useful for designing molecules to interrupt particular protein-protein interactions, or to compensate for changes in protein interfaces caused by mutations to the proteins. Several existing companies use structural data of proteins and protein complexes to design molecules that may have specific biological effects. Better protein interface prediction can aid such design processes by improving the structural data available to make drug-binding predictions. In the area of protein engineering, accurate prediction of protein interfaces using amino acid sequences can be very useful for engineering protein complexes with particular functions, or changing the interfaces between proteins i.e., making the binding of two proteins in a complex stronger or weaker to accomplish a particular biological function.

Overall, the work in this thesis contributes inside and outside academia by contributing to overall knowledge of how to better predict protein complexes, and what improvements remain to overall protein complex prediction.
Acknowledgements

First and foremost, I would like to thank my thesis supervisor, David Jones, for giving me the opportunity to do my PhD in his lab, and for his guidance, help and support over the past 5 years. The knowledge I received in his lab was invaluable and my time in the Jones lab has made me a better and more rigorous scientist. I would also like to thank my secondary supervisor Christine Orengo and internal examiner Kevin Bryson for their useful feedback during the course of my thesis.

I would also like to thank the members of the Bioinformatics group at UCL, past and present, who helped me through the last 5 years with useful knowledge, coding help, and great conversations. I would like to thank former and present post-docs in the group, Dan Buchan, Joe Greener, Andy Lau, and Shaun Kandathil who helped me with my coding problems and who shared their extensive knowledge of proteins, coding, and general trivia with me. I’d also like to thank my fellow PhD students, Michael Jones and Lewis Moffat for their help, support, and jokes. My time in the Jones lab provided me with lots of knowledge and as well as great conversations and many moments of laughter. I am very grateful to have been part of such a great lab.

I am also grateful to the Francis Crick Institute and University College London for providing stimulating academic environments and resources, as well as giving me the opportunity to meet many wonderful fellow academics. I would also like to thank the European Research Council for the funding that made this PhD work possible.

Finally, I would like to thank my friends and family who gave me so much emotional and mental support as well as lots of moments of levity and adventure over the last five years. I would like to thank my family for always believing in me and giving me endless support. I would like to thank the B&B crew™ Michael, Hannah J., and Trent for the endless laughter, support, and adventures. I’d also like to thank Rachel S., Rachel W., Hilda, Fran, and Hannah C. for supporting me, sharing pastries, and tea, making me laugh, and for helping me push through challenging moments. I am grateful to them all.
UCL Research Paper Declaration Form: referencing the doctoral candidate’s own published work(s)

Please use this form to declare if parts of your thesis are already available in another format, e.g. if data, text, or figures:

- have been uploaded to a preprint server;
- are in submission to a peer-reviewed publication;
- have been published in a peer-reviewed publication, e.g. journal, textbook.

This form should be completed as many times as necessary. For instance, if you have seven thesis chapters, two of which containing material that has already been published, you would complete this form twice.

<table>
<thead>
<tr>
<th>1. For a research manuscript that has already been published (if not yet published, please skip to section 2):</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Where was the work published? (e.g. journal name)</td>
</tr>
<tr>
<td>b) Who published the work? (e.g. Elsevier/Oxford University Press):</td>
</tr>
<tr>
<td>c) When was the work published?</td>
</tr>
<tr>
<td>d) Was the work subject to academic peer review? Please select.</td>
</tr>
<tr>
<td>e) Have you retained the copyright for the work? Please select.</td>
</tr>
</tbody>
</table>

[If no, please seek permission from the relevant publisher and check the box next to the below statement]:

☐ I acknowledge permission of the publisher named under 1b to include in this thesis portions of the publication named as included in 1a.

<table>
<thead>
<tr>
<th>2. For a research manuscript prepared for publication but that has not yet been published (if already published, please skip to section 3):</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Has the manuscript been uploaded to a preprint server? (e.g. medRxiv): Yes If yes, which server? <a href="https://arxiv.org/">https://arxiv.org/</a></td>
</tr>
<tr>
<td>b) Where is the work intended to be published? (e.g. names of journals that you are planning to submit to) Click or tap here to enter text.</td>
</tr>
<tr>
<td>c) List the manuscript's authors in the intended authorship order:</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>d) Stage of publication</td>
</tr>
<tr>
<td>3. For multi-authored work, please give a statement of contribution covering all authors (if single-author, please skip to section 4):</td>
</tr>
<tr>
<td>I, Nikita Desai, contributed to the section of the paper pertaining to the prediction of protein complexes for the minimal genome</td>
</tr>
<tr>
<td>4. In which chapter(s) of your thesis can this material be found?</td>
</tr>
<tr>
<td>5. e-Signatures confirming that the information above is accurate (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work):</td>
</tr>
<tr>
<td>Candidate:</td>
</tr>
<tr>
<td>Nikita Desai</td>
</tr>
<tr>
<td>Supervisor/ Senior Author (where appropriate):</td>
</tr>
<tr>
<td>David Jones</td>
</tr>
</tbody>
</table>
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPRI</td>
<td>Critical Assessment of Predicted Interactions</td>
</tr>
<tr>
<td>CASP</td>
<td>Critical Assessment of protein Structure Prediction</td>
</tr>
<tr>
<td>CBM</td>
<td>Coevolution-based methods</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional neural networks</td>
</tr>
<tr>
<td>CPORT</td>
<td>Consensus Prediction Of interface Residues in Transient complexes</td>
</tr>
<tr>
<td>DCA</td>
<td>Direct coupling analysis</td>
</tr>
<tr>
<td>DL</td>
<td>Deep learning</td>
</tr>
<tr>
<td>DMP</td>
<td>DeepMetaPSICOV</td>
</tr>
<tr>
<td>EC</td>
<td>Evolutionary couplings</td>
</tr>
<tr>
<td>ECOD</td>
<td>Evolutionary Classification of protein Domains</td>
</tr>
<tr>
<td>ESM</td>
<td>Evolutionary Scaling Model</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier transform</td>
</tr>
<tr>
<td>GRU</td>
<td>Gated recurrent unit</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov models</td>
</tr>
<tr>
<td>HMMER</td>
<td>HMM building tool that performs biological sequence analysis using profile hidden Markov models</td>
</tr>
<tr>
<td>IPA</td>
<td>Iterative pairing algorithm</td>
</tr>
<tr>
<td>LM</td>
<td>Language model</td>
</tr>
<tr>
<td>MCC</td>
<td>Matthews correlation coefficient</td>
</tr>
<tr>
<td>MI</td>
<td>Mutual information</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MSA</td>
<td>Multiple sequence alignment</td>
</tr>
<tr>
<td>NMR</td>
<td>Nuclear magnetic resonance</td>
</tr>
<tr>
<td>PAE</td>
<td>predicted aligned error</td>
</tr>
<tr>
<td>PDB</td>
<td>Protein Data Bank</td>
</tr>
<tr>
<td>PPI</td>
<td>Protein-protein interaction</td>
</tr>
<tr>
<td>PSICOV</td>
<td>Protein Sparse Inverse COVariance</td>
</tr>
<tr>
<td>ReLu</td>
<td>Rectified linear unit</td>
</tr>
<tr>
<td>ResNet</td>
<td>Residual neural network</td>
</tr>
<tr>
<td>RMSD</td>
<td>Root mean square deviation</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SASA</td>
<td>Solvent-accessible surface area</td>
</tr>
<tr>
<td>SOTA</td>
<td>State of the art</td>
</tr>
<tr>
<td>TBM</td>
<td>Template based modelling</td>
</tr>
</tbody>
</table>
# Table of Contents

1.1. Introduction: In Silico Structure Prediction of Protein Dimer Complexes ............................ 20
   1.1.1 Challenges in Protein Complex Structure Prediction .................................................... 21
   1.1.2 Motivation: Protein Dimer Complex Prediction using CBMs and Deep Learning .................. 26

1.2 Literature Review .................................................................................................................. 30
   1.2.1 Factors Contributing to the Improved Protein Contact Prediction ..................................... 30
      1.2.1.1 Impact of Increased Size of Sequence Databases ...................................................... 32
      1.2.1.2 Improved Sequence Searching Algorithms for Multiple Sequence Alignment .................. 32
      1.2.1.3 Application of Covariation for Contact Prediction: Global Statistical Methods ................ 33
      1.2.1.4 Meta-Predictors and Deep Learning for Improved Monomer Structure Prediction .......... 40
   1.2.2 In silico methods for prediction of homodimer complex structures .................................. 45
      1.2.2.1 Non-Coevolution-Based Methods for Homodimer Prediction .................................. 48
      1.2.2.2 Coevolution-Based Methods for Homodimer Prediction ......................................... 51
   1.2.3 In silico methods for prediction of heterodimer complex structures .................................. 55
      1.2.3.1 Pairing of Multiple Sequence Alignments for Heterodimer Complex Prediction .............. 57
      1.2.3.2 CBMs for heterodimer complex contact prediction .................................................. 63
      1.2.3.3 Complex docking for heterodimer complex contact prediction .................................... 66
      1.2.3.4 Machine learning and deep learning concepts for complex contact prediction ............... 69

1.3 Thesis Overview .................................................................................................................... 70

2. Homodimer Complex Contact Classification ........................................................................... 72
   2.1 Introduction: Homodimer Contact Classification ................................................................. 72

   2.2 Curating Datasets for Homodimer Prediction ................................................................. 74
      2.2.1 Homodimer Dataset 1.0: Curation of a Non-redundant Set of Homodimers ....................... 74
         2.2.1.1 Obtaining PDBs and Homodimer Complex Curation ............................................... 75
         2.2.1.2 Dataset Filtering by Sequence Redundancy and Alignment Depth ............................... 76
         2.2.1.3 Dataset Splitting: Training, Test, and Validation Set Creation using ECODAnalyse .......... 77
         2.2.1.4 Contact Map Generation ......................................................................................... 79
      2.2.2 Homodimer Dataset 2.0: Curation of a Newer and Less Stringent Set of Homodimers ........ 80
         2.2.2.1 Obtaining Homodimer PDBs .................................................................................. 81
         2.2.2.2 Dataset Filtering by Sequence Redundancy and Alignment Depth ............................... 81
         2.2.2.3 Dataset Splitting: Training, Test, and Validation Set Creation using ECODclust ............ 82

   2.3 Coevolution and Deep Learning for Homodimer Contact Classification .......................... 84
      2.3.1 Deep Learning Model 1: CNNContactClassify Model for Classification of Known Homodimer Contacts 84
4.3 Conclusions .................................................................................................................. 161

5. Benchmarking of in silico heterodimer complex prediction methods ........ 165

5.1 Introduction and motivation: heterodimer complex prediction ................... 165

5.2 Benchmark Set for Heterodimer Prediction Benchmarking ......................... 171

5.3 Multiple Sequence Alignment Pairing for Heterodimer Complexes .......... 173

5.3.1 Methodology ........................................................................................................... 175

5.3.1.1 MSA Pairing Methods ....................................................................................... 175

5.3.1.2 Direct Coupling Analysis Methods for Evaluation of Paired Alignments .......... 178

5.3.2 Results ..................................................................................................................... 180

5.4 Inter-Protein Contact Prediction for Heterodimer Complexes ................. 182

5.4.1 Methodology ........................................................................................................... 184

5.4.1.1 Heterodimer MSA Pairing ............................................................................... 184

5.4.1.2 Contact Maps for Benchmark Heterodimer Complexes ................................. 184

5.4.1.3 Inter-Protein Contact Prediction for Benchmark Heterodimer Complexes ...... 186

5.4.2 Results ..................................................................................................................... 188

5.5. Heterodimer Complex Docking and Ranking ................................................. 190

5.5.1 Methodology ........................................................................................................... 192

5.5.1.1 Docking of Heterodimer Complexes Using Crystal Structures ................. 192

5.5.1.2 Predicted DMPFold2 Inter-Chain Contacts for Decoy Ranking .................... 193

5.5.1.2 Docking of Crystal Structures and DMPfold generated structures ............. 194

5.5.2 Results ..................................................................................................................... 195

5.5.2.1 HADDOCK Docking: CPORT vs. DMPfold 2 contacts .................................... 195

5.5.2.2 Heterodimer Docking of Monomer Native Structures ................................. 196

5.5.2.3 Comparison of Docked Crystal Structures and DMPfold Structures .......... 197

5.5.2.4 Decoy Ranking with ZRank, Real Contacts and Predicted Contacts .......... 198

5.6 Conclusions: Heterodimer Docking and Ranking ............................................. 211

6. Heterodimer Prediction with GRUs and Language Models ......................... 214

6.1 Introduction and Motivation .................................................................................... 214

6.2 Methodology: Interface Contact Prediction and Heterodimer Complex Prediction ............................................................................................................................................... 217

6.2.1 Dataset Curation for Heterodimer Complex Prediction ................................. 217

6.2.2 Multiple Sequence Alignments for Training of biGRU models ..................... 219
6.2.3 Bidirectional GRU and Residual Neural Network for Contact Prediction ................................................ 220
6.2.4 Language Model Sequence Embeddings .......................................................................................... 222
6.2.5 Sequence pairing for LM + biGRU + ResNet model ........................................................................... 224
6.2.6 Model training of variations of the biGRU + ResNet model .......................................................... 226

6.3 Results: BiGRU + ResNet Models for Inter-Chain Contact Prediction .................. 226

6.4 Conclusion: Deep Learning Methods for Heterodimer Complex Prediction....... 229

7. AlphaFold2 Evaluation for Complex Docking ................................................................. 231

7.1 Methodology: AlphaFold2 for Heterodimer Complex Prediction ......................... 232

7.2 Results ................................................................................................................................................. 233
  7.2.1. AlphaFold2 Heterodimer Structure Prediction ............................................................................. 233

7.3 Conclusion: AlphaFold2 Complex Docking ................................................................. 245

8. General conclusions and future outlook ................................................................. 247

8.2 Limitations and Future Directions ...................................................................................... 252

8.2 Final Conclusions ......................................................................................................................... 255

Appendices .............................................................................................................................................. 257

A1. List of Abbreviations ..................................................................................................................... Error! Bookmark not defined.

A2. Deep learning concepts .................................................................................................................. 257

A3. Contribution Statement .................................................................................................................. 259

Software availability ............................................................................................................................. 260

Bibliography ......................................................................................................................................... 261
List of Figures

Figure 1.1: Annual growth of Protein Data Bank  
Figure 1.2: Growth of the UniProtKB/TrEMBL database in terms of available sequences  
Figure 1.3: X-Ray Crystallographic structure of asparagine synthetase mutant C51A from Escherichia coli str. K-12 (PDB ID: 11AS)  
Figure 1.4: X-Ray Crystallographic structure of Mms2/Ubc13 Ubiquitin Conjugating Enzyme Complex from Escherichia coli (PDB ID: 1JAT)  
Figure 1.5: CASP Structure Evaluation  
Figure 1.6: MSA co-evolution signals for the prediction of physically interacting residue pairs  
Figure 1.7: Covariation of directly and indirectly interacting residues  
Figure 1.8: Residual Blocks and skip connections  
Figure 1.9: SOD1 PPI and Homodimer Structure  
Figure 1.10: Covariation signals in homodimer chains  
Figure 1.11: PDB structures for dimeric ErbB3 receptor tyrosine kinase transmembrane domain  
Figure 1.12: Solvent-excluded surface of 11AS (asparagine synthetase mutant C51A)  
Figure 1.13: MSA pairing for inter-chain contact prediction using coevolution signals  
Figure 1.14: UniProt IDs for gene distance calculation  
Figure 2.1: HD1.0 Curation Pipeline  
Figure 2.2: Intra-chain and inter-chain contact maps for Asparagine synthetase mutant C51 (PDB ID 11AS)  
Figure 2.3: HD2.0 Curation Pipeline: Pipeline used for generation of training, test, and validation sets for HD2.0  
Figure 2.4: Architecture of the CNNContactClassify  
Figure 2.5: Homodimer Deep Learning Inputs  
Figure 2.6: CNNContactClassify training progress
Figure 2.7: Ranked MCC of CNNContactClassify on HD1.0 test set
Figure 2.8: PDB structures of homodimers from the homodimer test set
Figure 2.9: Homodimer HADDOCK docking
Figure 2.10: DMPContactClassify architecture
Figure 2.11: DMPContactClassify training
Figure 2.12: DMPContactClassify training using different losses
Figure 2.13: DMPContactClassify1.0 vs. DMPContactClassify2.0
Figure 3.1: CNNContactPredict Architecture
Figure 3.2: Inter-chain vs Intra-chain MCC
Figure 3.3: DMPContactPredict Architecture (DMP_CP)
Figure 3.4: DMP_CP vs. CNNContactPredict
Figure 3.5: Homodimer Complexes with Correct Class I Contacts
Figure 3.6: DMPfold + DMP_CP Method
Figure 3.7: Homodimer Complexes with Correct Class I Contacts DMPfold + DMP_CP
Figure 3.8: DMPfold RMSD and predicted TM-score versus DMPfold + DMP_CP inter-chain contact prediction precision
Figure 4.1: 1A1U_AC Docked Complexes
Figure 4.2: 1IS7_QR Docked Complexes
Figure 4.3: 1A2Z_CD Docked Complexes
Figure 4.4: 1NI8_AB Docked Complexes
Figure 5.1 Examples of Heterodimer Complexes
Figure 5.2: The MSA pairing problem
Figure 5.3: Examples of heterodimer complexes in the Heterodimer Benchmark Set
Figure 5.4: Inter-chain contact prediction Precision
Figure 5.5: 2P5T_AB Decoy Ranking
Figure 5.6: 2P5T_AB Top Ranked Decoy
Figure 5.7: 1FS0_EG Decoy Ranking
Figure 5.8: 1FS0_EG Top Ranked Decoys
Figure 5.9: 3RKO_FG Decoy Ranking
Figure 5.10: 3RKO_FG Top Ranked Decoy
Figure 6.1: HetDataset Generation
Figure 6.2: BiGRU + ResNet model
Figure 6.3: LM + BiGRU + ResNet model architecture
Figure 6.4: Paired LM + BiGRU + ResNet model architecture
Figure 7.1: 7D9W_AB AlphaFold2 models
Figure 7.2: 6W2L_AB AlphaFold2 models
Figure 7.3: 6L4P_AB AlphaFold2 models
Figure 7.4: 7LRQ_AB AlphaFold2 models
Figure 7.5: 6LE5_AB AlphaFold2 models
Figure 7.6: 7CSL_AC AlphaFold2 models
Figure 7.7: 7D5B_AD AlphaFold2 models
Figure 7.8: 7KOZ_CD AlphaFold2 models
Figure 7.9: 7D5P_AC AlphaFold2 models
Figure 7.10: 5R1D_AB AlphaFold2 models
List of Tables

Table 1.1: Top 10 performing contact prediction servers from the CASP13 evaluation
Table 2.1: Pisces Filtering Criteria
Table 2.2: Input map features for CNNContactClassify
Table 2.3: Performance metrics of CNNContactClassify from epoch with best validation MCC
Table 2.4: Confusion Matrix CNNContactClassify epoch with best validation MCC
Table 2.5: dilation rates for DMPContactClassify
Table 2.6: DMPContactClassify Best Epoch Performance Metrics
Table 2.7 DMPContactClassify Loss Function Comparison
Table 2.8: DMPContactClassify Test Set Performance
Table 2.9: Binomial test of proportions for MCC loss vs. BCE loss
Table 3.1: CNNContactClassify Training Schemes
Table 3.2: CNNContactPredict Model Training Schemes
Table 3.3: CNNContactPredict Training Performance
Table 3.4: CNNContactPredict Ranked Contact Performance
Table 3.5: Statistical significance of training strategy improvement (binomial test)
Table 3.6: DMP_CP Training Schemes
Table 3.7: DMPContactPredict
Table 3.8: Training schemes for DMPfold-DMP_CP
Table 3.9: Training schemes for DMPfold-DMP_CP
Table 3.10: Native monomer structures and DMPfold structures for 10 PDBs from test set
Table 4.1: DockQ scores and docking decoy quality
Table 4.2: DockQ Scoring Examples
Table 4.3: Overall ZDock generated decoy lists
Table 4.4: Overall ZDock generated decoy lists
Table 5.1: MSA Pairing Impact on DCA Contact Prediction (overlapping complexes)
Table 5.2: Paired t-test p-values for the MSA pairing methods
Table 5.3: Inter-chain contact prediction on benchmarking dataset
Table 5.4: HADDOCK Docking – CPfort restraints vs. DMPfold2 restraints
Table 5.5: Heterodimer docking results for five docking protocols
Table 5.6: ZDock Decoy Ranking using ZRank vs. DMPfold2 predicted inter-chain contacts
Table 5.7: Zdock Decoy Ranking using real vs. DMPfold2 predicted inter-chain contacts
Table 6.1: Training, Validation and Test Metrics
Table 6.2: Precision of top ranked contacts for HetData test set
Table 6.3: Table 0-3: Binomial Test of Proportions of paired vs. unpaired models
Table 7.1: AlphaFold2 structure prediction on AlphaFold2 Heterodimer benchmark set
1. Introduction and Literature Review

1.1. Introduction: *In Silico* Structure Prediction of Protein Dimer Complexes

Proteins have specific amino acid sequences, and the physicochemical properties of amino acids and their sequence determine their three-dimensional structures and structural dynamics. The finely tuned three-dimensional structures of proteins and the biophysical and biochemical properties of these structures enable diverse functions, interactions, and biological mechanisms that occur in all living organisms. Interactions with other macromolecules or molecules can also affect the three-dimensional structure of proteins, including interactions with other proteins, ligands, peptides, sugars, or cofactors (Alberts, Johnson and Lewis, 2002; Zhao, Cao and Zhang, 2020). The structural characterization of proteins and their complexes they in is often essential to the detailed elucidation of their properties, function, and mechanisms of action (Orengo, Todd and Thornton, 1999; Dill *et al*., 2008; Rigden, 2009; Kuhlman and Bradley, 2019).

In the last two decades, there has been an exponential increase in the number of available protein sequences due to large scale sequencing efforts and technological advances in sequencing methods (Apweiler, 2004; Breuza *et al*., 2016). Although experimental techniques such as X-ray crystallography, nuclear magnetic resonance (NMR), and cryogenic electron microscopy (cryo-EM) can be used to characterize protein structures, these techniques are often time consuming, expensive, experimentally challenging, and sometimes experimentally infeasible (Dill *et al*., 2008; Kuhlman and Bradley, 2019). This is especially true in the case of protein complexes, where the determination of the correct structure and biological assembly using experimental methods can often more complicated than for protein monomers.
1.1.1 Challenges in Protein Complex Structure Prediction

The increasing availability of protein sequences has far out-paced the experimental characterization of protein structures. Figure 1.1 and 1.2 show the increase in sequence and structure databases, respectively, in the last 2 decades.

![Growth of the PDB](image)

**Figure 1.1: Annual growth of Protein Data Bank**: Total number of protein structures in the PDB from 1999 to 2019. Data obtained from: https://www.wwpdb.org/stats/deposition.
Figure 1.2: Growth of the UniProtKB/TrEMBL database in terms of available sequences:
Total number of available sequences in UniProt database from 1997 to 2018. Chart obtained from https://www.ebi.ac.uk/uniprot/TrEMBLstats.

Although 2015 saw a significant reduction in the overall size of the UniProtKB database, this was due to the reduction of redundancy through the clustering of protein sequences with greater than 90% sequence similarity (Bursteinas et al., 2016). More manual curation of sequences was carried out in 2015 to improve the quality of sequence annotations and remove incorrect annotations (Bursteinas et al., 2016). This means that although the number of sequences decreased in 2015, the overall quality of the sequence information in the database increased significantly.

As knowledge of protein structures can add important biological context over sequence alone, it is desirable to have information about both the sequence and the structure of protein coding regions. Due to the difficulty and cost of experimental structure determination, accurate in silico prediction of protein structures has been an area of interest to the field of structural biology for several years.

Most cellular processes in living organisms are carried out by protein complexes, which are made up of one or more interacting protein chains (Gorka et al., 2019). Characterizing the structures of protein complexes can often be important for understanding biological functions, mechanisms, and pathways the proteins may be involved in and how genetic variations may affect their function (Orengo, Todd and Thornton, 1999; Rigden, 2009).
Accurate *in silico* prediction of the tertiary and quaternary structures of proteins and their complexes has been an open challenge for decades as it has important implications for biological research, disease research, drug discovery, and protein engineering (Garnier, 1990; Janin *et al*., 2003; Dill *et al*., 2008; Dill and MacCallum, 2012).

Although several methods have been developed and established to do template-based modelling using closely related or homologous templates for protein structure prediction, these methods depend on the availability of close enough sequence and structure homologs with available crystal structures to perform accurate structure prediction (Jones, Taylor and Thornton, 1992; Šali and Blundell, 1993; Fiser, 2010; Wang, Eickholt and Cheng, 2010; Ko, Park and Seok, 2012). The applicability of these methods for protein structure prediction is therefore limited by the availability of structures of close homologs in protein structure databases (Adhikari *et al*., 2017). This is especially true for protein complexes, where crystal structures or templates are often required for constituent subunits, and often the whole complex, to perform accurate complex structure prediction by template-based modelling (Janin *et al*., 2003; Lensink and Wodak, 2010; Dapkūnas *et al*., 2020).

The Critical Assessment of Predicted Interactions (CAPRI) is community-wide initiative which was established in 2001 and focuses on the assessment of computational methods for modelling of protein complexes and assemblies (Janin *et al*., 2003; Janin, 2005). A number of cases from the 7th CAPRI evaluation meeting in 2019 demonstrated that, even in cases where reasonable templates for protein complexes could be found, correct prediction of the protein complexes structures was a non-trivial problem (Dapkūnas *et al*., 2020). It has been demonstrated that even highly homologous complexes can still have differences in their corresponding interfaces (Dapkūnas *et al*., 2020). The performance of computational methods of structure prediction falls even further when close homologs for the constituent monomers or complex cannot be found (Dapkūnas *et al*., 2020). For these reasons, accurate *de novo* or *ab initio* prediction of protein complex structures where close or accurate homologs are not available in protein structure databases is an important challenge to improve *in silico* characterization of protein complexes and protein-protein interactions.
The work detailed in this PhD thesis therefore explores a number of *de novo* prediction methods, specifically coevolution-based and deep-learning methods, for the prediction of protein complex structures. This thesis also details the development and evaluation of a number of new deep-learning methods aimed at improving *de novo* protein complex structure prediction, specifically dimer complex prediction. The overall goal of this PhD was to explore the ability of current state-of-the-art structure prediction tools to perform accurate and reliable prediction of dimer structures where close templates are not available for the constituent protein chains and/or the overall protein complex. This challenge involved accurate prediction of the tertiary structures of constituent protein chains as well as accurate prediction of the orientation of interacting protein chains. The structure prediction of heterodimers and homodimers are explored separately as each category presents different challenges for coevolution-based and deep-learning protein structure prediction methods. Figures 1.3 and 1.4 illustrate examples of homodimer and heterodimer protein complexes, respectively, and interfaces between interacting chains.

**Figure 1.3:** X-Ray Crystallographic structure of asparagine synthetase mutant C51A from *Escherichia coli* str. K-12 (PDB ID: 11AS): Homodimer complex structure of asparagine synthetase enzyme. Protein complex chains A (light blue) and B (green) have identical sequence and structure. Image generated using ChimeraX (Pettersen et al., 2021)
A large challenge in predicting protein structure from amino acid sequence alone is the accurate prediction of residue-residue contacts within the protein that can be used for the prediction of protein tertiary and quaternary structures (Hopf et al., 2012; Ekeberg et al., 2013; Coucke et al., 2016). Accurate protein residue-residue contact and/or distance prediction is often an essential step to the accurate prediction of structure from amino acid sequence and has been an evaluation category in the Critical Assessment of protein Structure Prediction (CASP) competition since 1996 (Monastyrskyy et al., 2016). In the CASP evaluations, residue pairs are defined as being in contact if their Cβ atoms (Ca in the case of glycine) are 8Å or less apart for residues that are at least five residues apart in amino acid sequence (Monastyrskyy et al., 2016). Accurate residue contact predictions can be used to place residue-residue distance constraints that make overall structure prediction more accurate and tractable (Monastyrskyy et al., 2016; Adhikari et al., 2017; Schaarschmidt et al., 2018).
1.1.2 Motivation and Thesis Outline: Protein Dimer Complex Prediction using Coevolution Based Methods and Deep Learning

The years preceding the start of the work outlined in this thesis (2017) saw consistent improvement in the accurate de novo prediction of protein structure from amino acid sequences that could be attributed to a number of factors. These factors included: (1) the rapidly increasing availability of genomic sequences due to advances in next generation sequencing (Apweiler, 2004; Breuza et al., 2016), (2) improved multiple sequence alignment generation methods (Camacho et al., 2009; Eddy, 2009; Johnson, Eddy and Portugaly, 2010; Remmert et al., 2012; Steinegger and Söding, 2017), (3) the use of improved direct coupling analysis (DCA) methods that better differentiated covariation signals between directly and indirectly interacting protein residues (Weigt et al., 2009; Hopf et al., 2012; Jones et al., 2012; Sulkowska et al., 2012; Kamisetty, Ovchinnikov and Baker, 2013; Ekeberg, Hartonen and Aurell, 2014) and (4) the application of deep learning methods for protein contact prediction (Zhou, Wang and Xu, 2017; Schaarschmidt et al., 2018; Kandathil, Greener and Jones, 2019; Senior et al., 2019).

These advances in protein structure prediction were evident through benchmarking experiments such as the Critical Assessment of protein Structure Prediction (CASP) challenge, the biggest standardized blind evaluation of protein structure prediction methods. Figure 1.5 demonstrates the improvement in protein structure prediction methods between 2014 and 2018.
A major factor that contributed to the improvement in protein structure prediction came from improvements in the prediction of protein residue contacts. Many of these methods used coevolution-based methods and in some cases meta-predictors, to make protein contact predictions. Methods from the Jones lab that integrated covariation based prediction as well as meta-predictors and neural networks were amongst the top performing tools for residue contact prediction in CASP11 and CASP12 (Monastyrskyy et al., 2016; Schaarschmidt et al., 2018). A comprehensive review of coevolution-based methods for the prediction of protein contacts as well as an overview of other meta-predictors and deep learning methods that contributed to the increased accuracy of protein structure prediction is provided in section 1.2 of this thesis. All these coevolution-based methods, meta-predictors, deep-learning based methods however, had been applied to the structure prediction of protein monomers and not to the prediction of protein complexes.

During the 7th CAPRI evaluation (rounds 38-46), which took place in 2019, very few methods used for protein complex prediction used coevolution-based information or deep-learning for docking of homodimers or heterodimers (Wodak, Velankar and
Sternberg, 2020). One of the previous CAPRI rounds, made up of 25 targets, was a prediction experiment run jointly between CAPRI and CASP, and the results of that round demonstrated the importance of integrating information such as inter-chain contact predictions for successful protein assembly prediction (Lensink et al., 2016). Prior to the start of the PhD work outlined in this document (2017), a number of papers had explored the application of coevolution-based methods for the prediction of protein-protein contacts and complex interfaces (Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014; dos Santos et al., 2015; Uguzzoni et al., 2017). The identification of interacting residues in protein interfaces can be very useful for reducing the search space for docking simulations, and in some cases, as few as 4 correct inter-protein contacts can be enough for successful generation of docking models, given a low enough false positive rate in the predicted contact list (dos Santos et al., 2015; Coucke et al., 2016; Yu et al., 2016). The number of papers published exploring the application of CBMs for protein complex contact prediction, however, was far fewer than those published for protein monomer contact prediction. This discrepancy was due to a number of challenges in the application of CBM to protein dimer complexes.

In the case of protein homodimers, identical MSAs mean covariation-based methods struggle to differentiate between residue contacts on the same chain and on different chains of homodimer complexes i.e., intra-protein contacts and inter-protein contacts (dos Santos et al., 2015; Uguzzoni et al., 2017; Zhou, Wang and Xu, 2017). This problem could potentially be addressed using methods that are able to analyse patterns in the overall residue pair map, such as convolutional layers or deep-learning. An overview of published studies exploring the application of CBMs to homodimer contact prediction as well as the limitations of those studies is outlined in section 1.2.2. At the time the work outlined in this thesis was started, no meta-predictors or deep-learning methods had been published for the prediction of homodimer interface contacts.

Work outlined in chapters 2 to 4 of this thesis therefore combined predictions from evolutionary statistical methods, secondary structure predictions, and solvent accessibility predictions with deep-learning for the prediction of homodimer complex contacts. The hypothesis was that using patterns of coevolution and other predicted features such as secondary structures and solvent accessibility across residue pair maps,
it would be possible for deep-learning models to (1) predict both inter-chain and intra-chain protein contacts and (2) differentiate between these two types of contacts.

In the case of protein heterodimers, the main limitation of CBMs is the fact that it is necessary to generate paired multiple sequence alignments (MSAs) where biologically interacting paralogs and homologs of constituent protein chains are correctly paired (Weigt et al., 2009; Hopf et al., 2014; Uguzzoni et al., 2017; Zhou, Wang and Xu, 2017). It is often difficult to generate large, accurate paired MSAs for interacting proteins because matching interacting homologs is often a non-trivial problem in cases where multiple paralogs exist in species, and where multiple sequence alignments have different depths for interacting protein chains (Hopf et al., 2014; Bitbol et al., 2016; Wang et al., 2016; Taylor, 2017; Uguzzoni et al., 2017). A discussion of challenges in coevolution-based heterodimer complex contact prediction is covered in section 1.2.3.

Chapter 5 of this thesis therefore outlines a benchmarking of previously published MSA pairing tools and compares the ability of CBMs, meta-prediction and deep-learning methods to perform accurate prediction of heterodimer interface contacts using paired MSAs. Work in Chapter 5 also compared a number of previously developed complex docking tools for the docking of heterodimer complexes and explored how the precision of interface contact prediction impacts heterodimer complex docking. Experimental work outlined in Chapter 6 explored the use of multiple sequence embedding methods for the prediction of heterodimer interface contacts and explored whether it was possible to circumvent MSA pairing to get accurate inter-chain heterodimer contacts.

Overall, the work outlined in this thesis attempts to address the challenges of de novo dimer complex prediction by combining CBMs with meta-predictors and deep-learning methods that had shown improvements in the successful prediction of protein monomer contacts. The goal of this work was to determine whether these methods could improve complex contact prediction, and whether improvements in contact prediction could in turn improve overall de novo prediction of protein dimer structures.
1.2 Literature Review

The following literature review covers papers, articles, and textbook chapters published before the work outlined in this thesis was started (October 2017) as well as relevant research work published while the thesis work outlined was being carried out. The goal of this literature review was to determine the state of the art in coevolution-based methods, deep-learning, and de novo protein dimer structure prediction, and to determine where existing tools and benchmarks could be built on to improve the prediction of protein complexes. The following literature review covers an assessment of published methods and research, including limitations of published papers, and an overview of questions that were explored through the work outlined in this thesis.

1.2.1 Factors Contributing to the Improved Protein Contact Prediction

Before 2012, protein residue contact prediction showed few detectable improvements for protein structure prediction in CASP10 (Moult et al., 2014). This was because of two factors: first, only one group involved in CASP10 tried to use contact predictions for protein structure modelling, and second, deep and well-balanced multiple sequence alignments that were needed to make protein contact predictions were not available for most CASP10 targets (Moult et al., 2014). Before 2010, true-positive contact prediction of interacting residues in proteins was low for most available tools, with a maximum average precision less than 20% for the top-L/5 long-range contacts, where L was the length of the protein (Moult et al., 2014).

Since 2011, however, several protein contact prediction methods have been published and released, with an increasing effort made to incorporate predicted contacts in protein structure prediction. Contact-assisted modelling and long-range contact prediction were both categories added in the CASP11 (2014) experiment (Monastyrskyy et al., 2016). In CASP11, 29 groups participated in the intra-molecular contact prediction category, which assessed the prediction of long-range contacts (sequence separation >23 residues). At least eight groups used evolutionary coupling techniques to perform contact prediction. The best performing group, CONSIP2 (Jones, UCL), used the MetaPSICOV method
(Jones et al., 2015) to predict long-range contacts and achieved a precision of 27% for L/5 predicted contacts for protein domains that could not be modelled by homology modelling tools, i.e., free modelling targets (Monastyrskyy et al., 2016).

Assessments of methods published since 2011 showed significant improvements in the accuracy and precision of long-range contact prediction (Jones et al., 2012, 2015; Kamisetty, Ovchinnikov and Baker, 2013; Ekeberg, Hartonen and Aurell, 2014; Hopf et al., 2014; Kaján et al., 2014; Seemayer, Gruber and Söding, 2014; Zhou, Wang and Xu, 2017). The CASP experiment that occurred prior to the beginning of the PhD work described in this thesis, CASP12, demonstrated the ability of several tools to accurately predict long-range protein contacts, and showed significant improvement in the modelling of protein structures due to the incorporation of more precise protein contact predictions and better differentiation between directly and indirectly residue interactions (Ovchinnikov et al., 2018; Schaarschmidt et al., 2018, p. 12).

There are a number of reasons for the increase in precision of long-range contact prediction since 2011. First, is the large increase in the number and quality amino acid sequences available in sequence databases due to large-scale sequencing efforts (Apweiler, 2004; Suzek et al., 2015; Breuza et al., 2016). Second, is the significant improvement in the performance of sequence alignments methods used to build multiple sequence alignments, or MSAs (Eddy, 2009; Johnson, Eddy and Portugaly, 2010; Remmert et al., 2012; Steinegger and Söding, 2017). Third, a number of global evolutionary statistical methods, or coevolution-based methods (CBMs), that could disentangle directly and indirectly interacting contacts have been developed to perform improved contact prediction in protein monomers (Hopf et al., 2012; Jones et al., 2012; Sulikowska et al., 2012; Kamisetty, Ovchinnikov and Baker, 2013; Ekeberg, Hartonen and Aurell, 2014; Seemayer, Gruber and Söding, 2014; Coucke et al., 2016). Finally, several of these CBMs have been combined with meta-predictors, shallow neural networks, and deep learning for improved contact prediction (Jones et al., 2015; Wang et al., 2016; Zhou, Wang and Xu, 2017; Schaarschmidt et al., 2018; Kandathil, Greener and Jones, 2019). In recent years, improved residue contact and distance predictions have been increasingly used to perform better de novo structure prediction using methods such as CNS or Rosetta (Yang et al., 2020).
1.2.1.1 Impact of Increased Size of Sequence Databases

In the 2 decades prior to the start of the work outlined in this thesis the number of available protein sequences had increased exponentially, as shown in figure 1.2. This was due to large-scale sequencing efforts and improved sequencing technologies (The UniProt Consortium, 2015; Breuza et al., 2016). This large increase in the size of available sequences meant that deeper and more diverse multiple sequence alignments could be built for protein targets. The ability to build deeper alignments has been shown to directly impact the ability to obtain accurate protein contact predictions using evolutionary statistical methods (Jones et al., 2015; Monastyrskyy et al., 2016, p. 11; Schaarschmidt et al., 2018, p. 12).

1.2.1.2 Improved Sequence Searching Algorithms for Multiple Sequence Alignment

The ability of CBMs to accurately identify potential residue contacts is dependent on their ability to correctly identify co-evolving residues. These methods therefore need deep and accurate multiple sequence alignments (MSAs) to be able to accurately pick up on coevolution signals between interacting residues. Previous studies (Jones et al., 2012; Hopf et al., 2014; Seemayer, Gruber and Söding, 2014) have shown that deeper MSAs often correlated with better performance of CBMs for identification of residue-residue contacts. Generally, a minimum MSA depth of between 20 and 50 sequences was needed to perform contact prediction (Jones et al., 2012; Hopf et al., 2014; Kaján et al., 2014). Accurate MSA generation is therefore essential to the accuracy of protein contact prediction using CBMs.

For this reason, another factor that contributed to the increase in the quality of protein contact predictions was the development of better and faster sequence alignment methods (Eddy, 2009; Johnson, Eddy and Portugaly, 2010; Remmert et al., 2012). HMMER 3, released in 2009, implemented probabilistic inference using profile hidden Markov models (HMMs) to perform local sequence alignment (Eddy, 2009). The method employed a new heuristic acceleration algorithm and log-odds likelihood scores summed over alignment certainty rather than over a single optimal alignment, and also computed
posterior distributions over an ensemble of possible alignments, thereby producing probabilities for each aligned residue (Eddy, 2009). The main benefit of HMMER was the fact it was able to perform a more powerful probabilistic sequence alignment at speeds comparable to established BLAST and PSI-BLAST sequence searching tools (Eddy, 2009). HMMER was soon followed by other powerful homology search tools that used probabilistic inference methods. An iterative profile-HMM search method, JackHMMER, sped up and improved the HMMER method using a series of database filtering steps, called HMMERHEAD, that were applied prior to scoring algorithms to reduce overall search times (Johnson, Eddy and Portugaly, 2010). These filtering steps resulted in a six-to twenty-fold decrease in search time with minimal loss to sequence search sensitivity (Johnson, Eddy and Portugaly, 2010). JackHMMER greatly improved on previously existing sequence search methods based on BLAST in terms of speed and sensitivity.

Another method, HHblits (Remmert et al., 2012), worked by representing both query and database sequences using a 219-letter extended alphabet representation of sequence profiles. This allowed the comparison of sequence profiles by matching alphabet representations using fast and standard dynamic programming techniques (Remmert et al., 2012). The HHblits method sped up sensitive sequence searching compared to PSI-BLAST, and resulted in 50-100% higher sensitivity and more accurate alignments (Remmert et al., 2012).

The introduction of these faster and more sensitive sequence searching methods improved the overall quality of multiple sequence alignments that could be used for contact prediction using evolutionary statistical methods.

1.2.1.3 Application of Covariation for Contact Prediction: Global Statistical Methods

Another factor that contributed to the dramatic increase of contact prediction precision was the development of global evolutionary statistical methods, or coevolution-based methods (CBMs), such as direct coupling analysis (DCA) and sparse inverse covariance prediction (Marks et al., 2011; Hopf et al., 2012; Jones et al., 2012; Sulkowska et al.,
At the time the work in this thesis was started, a number of studies had explored the prediction of tertiary protein contacts in protein domains or monomer chains using CBMs (Hopf et al., 2012; Jones et al., 2012; Ekeberg et al., 2013; Kamisetty, Ovchinnikov and Baker, 2013; Coucke et al., 2016). CBMs operate on the hypothesis that important and functionally relevant protein contacts – i.e., residue pairs which are important to the structure and thereby the function of proteins - will be maintained across homologs and paralogs of those proteins (Hopf et al., 2012, 2014; Jones et al., 2012; Ekeberg et al., 2013; Kamisetty, Ovchinnikov and Baker, 2013; Kaján et al., 2014; Coucke et al., 2016; Monastyrskyy et al., 2016). This hypothesis implies that mutations occurring in one functionally important residue in a residue-residue contact are likely to have compensatory changes in the corresponding residue, and that these compensatory changes can be captured in multiple sequence alignments (MSAs). CBMs therefore attempt to identify interacting residues through covariation signals in the protein MSAs (Miyazawa, 2018). At the time the work in this thesis was started, a number of CBMs had been developed that greatly improved the ability of these methods to deconvolute directly and indirectly interacting protein residues. These CBMs had been demonstrated to be effective at improving protein contact prediction single domains and protein monomers over older mutual information based approaches (Weigt et al., 2009; Hopf et al., 2012; Jones et al., 2012; Sułkowska et al., 2012; Kamisetty, Ovchinnikov and Baker, 2013; Ekeberg, Hartonen and Aurell, 2014).
**Figure 1.6: MSA co-evolution signals for the prediction of physically interacting residue pairs:** Evolutionary statistical models use information about covarying residues to make predictions of residue contacts. Contact map on the right shows likely contacts predicted from an MSA, with predicted contacts in yellow and non-contacts in blue.

**Mutual Information**

The first covariation-based methods for the prediction of residue contacts in proteins relied on the use of mutual information to identify covarying residues. Mutual information (MI) methods work by measuring the statistical codependence of two amino acid positions (Korber et al., 1993; Lapedes et al., 1999; Little and Chen, 2009). Multiple sequence alignments (MSAs) are used to estimate the distribution of amino acid frequencies at each position along the length of the protein sequence being studied (Korber et al., 1993; Lapedes et al., 1999; Weigt et al., 2009). MI methods work by quantifying the amount of knowledge or information that can be removed from one amino acid site by the knowledge of which amino acid is at the corresponding site being studied.

For an MSA, if \( p_i \) is a vector of frequencies of 20 amino acids plus gaps at position \( i \), then entropy at position \( i \), \( H_i \), is given by:

\[
H_i = - \sum_{\text{aa} \in \{A, C, D, \ldots, Y\}} p_i(\text{aa}) \log_2(p_i(\text{aa})),
\]

\[\forall \text{aa} \in \{A, C, D, \ldots, Y\} \ni p_i(\text{aa}) \neq 0\] (Eq. 1.1)

\( H_i \) has a minimum value of 0 when all the amino acids at position \( i \) have the same value, and a maximum value when all 20 amino acids (and gaps) are equally distributed at
position $i$. The entropy at position $j$, $H_j$, is defined in a similar manner. If $p_{ij}$ is the 21x21 matrix with the joint distribution of each amino acid pair at $i$ and $j$, the joint entropy $H_{ij}$ for $p_i$ and $p_j$ is the entropy of the joint distribution $p_{ij}$ and is given by:

$$H_{ij} = - \sum p_{i,j}(aa_i, aa_j) \log_2 \left( p_i(aa_i, aa_j) \right)$$

$$\forall aa_i, aa_j \in \{A, C, D, ..., Y\} \ni p_i(aa_i, aa_j) \neq 0 \quad \text{(Eq. 1.2)}$$

The mutual information of $p_i$ and $p_j$, $MI_{ij}$, is therefore a quantification of the dependency between two positions and is given by:

$$MI_{ij} = H_i + H_j - H_{ij} \quad \text{(Eq. 1.3)}$$

To determine the significance of covariation signals, the mutual information of every pair of amino acid positions across the protein length is evaluated. Korber et al. introduced the first application of MI to sequence alignments for the identification of sequence alignments in viral peptides (Korber et al., 1993), a method which was extended to larger protein sequences by Lapedes et al. (Lapedes et al., 1999).

MI based methods, however, have a limited ability to differentiate between directly and indirectly covarying residues: i.e., residues likely to be in contact and residues coupled indirectly by indirect coupling or phylogenetic effects, as illustrated in Figure 1.7. Work by Burger and van Nimwegen previously showed that co-evolving contacts commonly formed covarying chains that ‘percolate through protein structures’, and result in indirect statistical dependencies between non-interacting residue pairs (Burger and van Nimwegen, 2010). This problem, called the chaining problem (Lapedes et al., 1999), means that MI is likely to pick up covariation signals between all residues involved in such covarying chains, thereby leading to a large number of false-positives for protein contact prediction (Lapedes, Giraud and Jarzynski, 2012; Coucke et al., 2016). This fact limits the ability of mutual-information based methods to accurately predict only physically interacting residues from protein MSAs.
Figure 1.7: Covariation of directly and indirectly interacting residues: Demonstration of the chaining effect of directly and indirectly coevolving residues. Mutual information methods might show covariation between non-physically interacting residues. Here the protein chain is represented by the orange line and physically interacting residues are connected with black dashed lines. Blue dashed lines connect non-physically interacting residues that may covary.

**Global Statistical Methods and Maximum Entropy Approaches**

To overcome the challenges raised by the use of mutual information for contact prediction, a number of global statistical methods have been developed that can better differentiate between covariation signals of directly and indirectly interacting residues (Weigt et al., 2009; Marks et al., 2011; Morcos et al., 2011; Jones et al., 2012; Sulkowska et al., 2012; Ekeberg et al., 2013; Kamisetty, Ovchinnikov and Baker, 2013; Skwark, Abdel-Rehim and Elofsson, 2013; Seemayer, Gruber and Söding, 2014). Many of these methods involve direct coupling analysis (DCA). DCA methods are based on the maximum entropy model for the distribution of amino acid frequencies in MSAs, which are based on the underlying idea of looking for the simplest statistical model of protein sequences capable of reproducing empirically observed correlations, or the minimal set of contacts that can explain covariation signals in MSAs (Lapedes, Giraud and Jarzynski, 2012).

Contact prediction using DCA methods and sparse inverse covariance largely outperformed mutual information and raw correlations for contact prediction, but depending on the method used, deconvolution of directly and indirectly interacting residues could be very computationally expensive (Miyazawa, 2018). For this reason, a number of global statistical evolutionary methods, or coevolution-based methods, were
developed that attempted to improve the contact prediction precision and computational efficiency for the deconvolution of directly and indirectly interacting residues.

One of the earliest CBMs adapted a previously developed Bayesian network model (Burger and van Nimwegen, 2008) to disentangle directly and indirectly interacting residue pairs and allowed residue pairs with weak statistical dependencies to be detected and removed from contact prediction lists (Burger and van Nimwegen, 2010). This was done by exploiting the fact directly interacting residue pairs are likely to have chains of indirect dependencies induced by the direct interactions of true residue contacts (Burger and van Nimwegen, 2010). This logic and Bayesian pairwise posterior probabilities were used to differentiate directly and indirectly interacting residue pairs by finding residue pairs that induced a common set of indirect dependencies (Burger and van Nimwegen, 2010). This method was demonstrated to successfully remove indirect dependencies while also being able to predict contacts with weaker statistical dependencies (Burger and van Nimwegen, 2010). This Bayesian network method, however, was very computationally expensive and could not efficiently be extended to long protein sequences (Burger and van Nimwegen, 2010).

Another maximum-entropy DCA approach, developed by Lapedes et al. used conditional mutual information (CMI) to score residue-residue contacts and then used a Monte-Carlo algorithm to determine the simplest probabilistic model able to account for the co-variation network observed through CMI (Lapedes, Giraud and Jarzynski, 2012). Although the method was one of the first with the ability to disentangle direct from indirect contacts, the process of calculating marginal probabilities by Monte Carlo simulations was also very computationally intensive and at the time could only be applied to very short protein sequences, i.e., less than 50 residues (Lapedes, Giraud and Jarzynski, 2012).

A cross-entropy method developed in 2009 used a message passing algorithm (mpDCA) to reduce the computational load of calculating directly coupled pairs (Weigt et al., 2009). The method was still quite slow when applied in large-scale analysis across multiple protein families. In 2011, Morcos et al. published a computationally efficient implementation of DCA, called mean-field DCA (mfDCA) that was based on mean-field approximation of DCA (Morcos et al., 2011). This mfDCA was shown to be $10^3$ to $10^4$ times faster than mpDCA and could therefore be used to rapidly analyse many long protein
sequences (Morcos et al., 2011). When evaluated on a set of 131 domain families, the method resulted in significantly better performance in identifying true positive protein contacts than the Bayesian network approach and mutual information (Morcos et al., 2011). The method was also able to capture interdomain residue contacts (Morcos et al., 2011).

Another method, from Marks et al., also used mfDCA and demonstrated a similar improvement in true positive contact prediction observed by Morcos et al (Marks et al., 2011). The mfDCA method, called EVfold, demonstrated an improved ability to use identified positive contacts for protein structure prediction (Marks et al., 2011). The EVfold method used a maximum entropy model to infer couplings but used pseudocount instead of regularization terms to calculate the inverse covariance matrix, which reduced the computational requirements of the method (Marks et al., 2011; Miyazawa, 2018). EVfold was used to translate residue couples into distance constraints that could be used to generate 3D structures that could be followed by energy minimization and molecular dynamics methods to generate protein structures (Marks et al., 2011). Overall, the EVfold method was adapted as part of the EVCouplings server (Hopf et al., 2018) and has been successfully applied to the prediction of protein structures and protein complex structures. mfDCA has also been implemented as part of the FreeContact (Kaján et al., 2014) contact prediction tool.

PSICOV (Jones et al., 2012), a method developed by Jones et al, addressed the computational expense of previously developed global statistical evolutionary methods by using sparse inverse covariance estimation for contact prediction (Jones et al., 2012). The method used a graphical lasso technique to estimate the inverse of the covariance matrix to determine the minimal number of contacts that could explain observed covariation signals (Jones et al., 2012). The method showed greatly increased accuracy in contact prediction over mutual information and was also more computationally tractable than most DCA techniques (Jones et al., 2012). Rank-50 results for residue pairs with sequence separation >23 showed that PSICOV was able to predict over 50% of true contacts compared to normalized mutual information approaches (Jones et al., 2012; Miyazawa, 2018).

Finally, a pseudolikelihood approach (plmDCA), called GREMLIN, used Gaussian prior probabilities of site pairing for contact prediction, and showed increased protein contact
prediction accuracy for slightly shallower MSAs than other DCA methods (Kamisetty, Ovchinnikov and Baker, 2013). The method integrated sequence covariation and structural context information, which allowed more accurate contact predictions from fewer homologous sequences, since previous DCA methods required very large MSAs as a requirement for effective contact prediction (Kamisetty, Ovchinnikov and Baker, 2013). CCM-pred, a faster, optimized implementation of the pseudolikelihood DCA method implemented in GREMLIN showed greatly reduced runtimes and improved prediction efficiency (Seemayer, Gruber and Söding, 2014).

All of these DCA and global statistical evolutionary methods showed mean precision values that greatly out-performed normalized mutual approaches for contact prediction, reduced the number of false positives of such MI approaches and contributed to the overall improvement of contact prediction precisions using global evolutionary statistical methods. PSICOV, EVfold, GREMLIN, and CCM-pred also implemented methods that reduced computational complexity and processing time of protein contact prediction, successfully decreased the number of false positive contact predictions, and improved overall contact precision.

1.2.1.4 Meta-Predictors, Deep Learning and Structure Modules for Improved Monomer Structure Prediction

After the development of global statistical methods to improve contact prediction precision, the next big improvements in contact prediction accuracy came from a new set of methods that used meta-predictors and deep-learning to improve the accuracy and precision of contact prediction.

Meta-Predictors: Combined Predictor Outputs for Improved Monomer Contact Prediction

Meta-predictors combine information from various prediction tools to make computational predictions. These predictions can include mutual information predictions, outputs from CBM or DCA methods, secondary structure predictions, and solvent accessibility predictions as input features into models for contact prediction. Most meta-predictors
developed prior to 2014 used shallow neural networks trained for contact prediction using a combination of input features.

The best performing method for residue contacts prediction in CASP11 was MetaPSICOV (Jones et al., 2015) which was trained on MSAs built using HHblits run on the March 2013 release of UniProt20 with an E-value threshold of 10^{-3} (Jones et al., 2015). MetaPSICOV combined predictions from PSICOV, CCMPred (Seemayer, Gruber and Söding, 2014), and FreeContact (Kaján et al., 2014) with solvent accessibility predictions from SOLVPRED, shannon entropy of alignment columns, global amino acid compositions, and log sequence length (Jones et al., 2015). These input features were fed into two stages of classic feed-forward neural networks with 55 hidden units and a single output unit. MetaPSICOV mean precision for top-L predicted long range contacts showed significantly better performance than PSICOV (60% higher), CCMPred (40% higher), mfDCA (57% higher), and consensus only methods (37% higher).

Overall, most methods for contact prediction in CASP11 combined predictions from evolutionary statistical methods with other prediction features to achieve higher contact prediction precisions than DCA methods on their own (Monastyrskyy et al., 2016).

**Neural Networks and Deep Learning for Improved Monomer Contact and Structure Prediction**

CASP12 (2016) saw an increasing use of deep learning methods and meta predictors for protein contact prediction including MetaPSICOV, SPOT-contact, PconsC, RaptorX-contact, iFold, and EPSILON-CP (Schaarschmidt et al., 2018, p. 12).

The best performing method for contact prediction at CASP12, RaptorX, was an ultra-deep learning method that incorporated evolutionary couplings from CCMPred with sequence conservation information fed into two dimensional residual neural networks (ResNets) to perform contact prediction. RaptorX had the best CASP12 performance in terms of the F1 scores for L/2 predicted contacts (Schaarschmidt et al., 2018). The F1 score measures the accuracy of a model and symmetrically represents both precision and recall in a single metric. The equation for F1 score is shown in equation 1.4.
\[ F1 \text{ score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]  

(Eq. 1.4)

Compared to traditional convolutional neural networks (CNNs), ResNets are able to train much deeper neural networks through the use of ‘skip connection’ identity mapping and residual blocks. This identity mapping adds the output from the preceding layer in a residual block to the output of the residual block (Figure 1.8) allowing the efficient training of very deep neural networks.

![Figure 1.8: Residual Blocks and skip connections](image)

**Figure 1.8: Residual Blocks and skip connections:** Residual blocks are made up of two weight layers, usually with an activation function between them (i.e. ReLu). Skip connections involve adding the input to a residual block to the output of the residual block.

An updated version of MetaPSICOV, MetaPSICOV2, was the second-best performing method at CASP12. The third best performing method, EPSILON-CP (referred to as iFold in the CASP experiments), also used a deep-learning residual neural network model trained on combined DCA inputs to obtain protein contact predictions (Schaarschmidt *et al.*, 2018). iFold combined evolutionary, physiochemical, and sequence-based information into a ResNet with 4 hidden layers to predict protein contacts (Stahl, Schneider and Brock, 2017). The iFold method performed quite well on the contact prediction task, ranking just below RaptorX and MetaPSICOV2 in CASP12 (Schaarschmidt *et al.*, 2018). Another method, PconsC2, used a meta predictor that
incorporated PSICOV and plmDCA outputs and used random forest classifiers to make contact predictions. PconsC2 also performed quite well in CASP12, although not as well as RaptorX, MetaPSICOV2, or iFold (Skwark et al., 2014). Overall, meta-predictors, machine learning (i.e., random forests), and deep-learning methods that used convolutional neural networks (CNNs) or residual neural networks (ResNets) greatly improved the overall precision of predicted contacts in the CASP12 assessment.

**CASP13: Coming of Age of Deep Learning and Monomer Structure Prediction**

In 2018, while the work described in this thesis was in progress, the CASP13 Assessment took place. The CASP13 evaluation saw a significant increase in the accuracy of residue contact and distance predictions, as well as overall structure predictions (Shrestha et al., 2019). The AlphaFold method that was first announced at CASP13 showed the best performance in terms of de novo protein structure prediction and did not explicitly predict residue contacts or distances for the purpose of structure prediction. Instead the method used a distance potential to predict coordinates of protein fragments that could be used to generate overall protein structures (Senior et al., 2019).

In terms of protein contact prediction, the best performing contact prediction method evaluated at CASP13 was the TripletRes method (Li et al., 2021) which combined prediction outputs from PSICOV inverse covariance estimation from the MSA (Jones et al., 2012), mean-field approximation of Potts model (Morcos et al., 2011), and the matrix of the inverse Potts model approximated using pseudo-likelihood approximation (Skwark, Abdel-Rehim and Elofsson, 2013). These covariance matrices were combined with predicted secondary structure (Jones, 1999) into four separate stacked deep residual networks (ResNets) that were trained simultaneously in an end-to-end fashion (Li et al., 2021). The resulting method ranked 4th in F1 contact prediction performance in CASP13.

DeepMetaPSICOV (DMP) was another method that did quite well in CASP13 F1 contact prediction. DMP combined the 58-channel MetaPSICOV inputs with 441-channel covariance matrices used in a previously developed method (DeepCov) which captured pairwise covariance information for all 20 amino acids plus gaps for all residue pairs in a protein (Jones and Kandathil, 2018; Kandathil, Greener and Jones, 2019). These inputs were fed into a deep, fully convolutional residual neural network with 18 residual blocks
that had varying dilation rates. An explanation of convolutional layers and dilated convolutions can be found in Appendix 2 of this thesis. DMP also used additional data augmentations such as loop sampling and flipping input tensors and contact maps, to improve overall contact prediction. Overall, DMP output more precise contact predictions than MetaPSICOV.

Table 1-1, obtained from the CASP website shows the top performing contact prediction methods. All the top performing methods (TripletRes, RaptorX, DMP, and Zhang.Contact) used residual neural networks trained on the outputs of multiple DCA predictors in addition to other predictions such as secondary structure and solvent accessibility predictions to make protein contact predictions.

<table>
<thead>
<tr>
<th>#</th>
<th>GR code</th>
<th>GR name</th>
<th>Domains Count</th>
<th>SUM Zscore (≥-2.0)</th>
<th>Rank SUM Zscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>032</td>
<td>TripletRes</td>
<td>31</td>
<td>33.0683</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>498</td>
<td>RaptorX-Contact</td>
<td>31</td>
<td>32.9798</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>323</td>
<td>TripletRes_AT</td>
<td>30</td>
<td>24.0527</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>180</td>
<td>ResTriplet</td>
<td>31</td>
<td>25.5417</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>491</td>
<td>DMP</td>
<td>31</td>
<td>21.1262</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>036</td>
<td>Zhang.Contact</td>
<td>31</td>
<td>17.4296</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>189</td>
<td>ZHOU-Contact</td>
<td>31</td>
<td>17.1937</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>125</td>
<td>ResTriplet_AT</td>
<td>30</td>
<td>14.8173</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>164</td>
<td>Yang-Server</td>
<td>31</td>
<td>12.3500</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 1-1: Top 10 performing contact prediction servers from the CASP13 evaluation: The top performing methods (RaptorX-Contact, TripletRes, DMP, ZHOU-Contact and Yang-Server) all used residual neural networks for contact prediction. Results obtained from https://predictioncenter.org/casp13.

The results of CASP13 (2018) demonstrated the usefulness of residual neural networks trained on combined outputs of evolutionary statistical methods, secondary structure predictions, and solvent accessibility predictions for the prediction of intra-chain protein contacts. Overall, the CASP13 assessments and papers exploring the application of deep-learning methods such as CNNs and ResNets trained on covariance matrices, CBM predictions, secondary structure predictions and solvent accessibility predictions showed a significant increase in accuracy and precision of protein residue-residue contact and distance predictions in protein domains and monomer chains.
Prior to the start of this PhD thesis, however, no papers had been published exploring the application of these deep-learning methods and meta-predictors to the prediction of protein complex residue contacts. For this reason, one of the main themes of the work done over the course of this PhD was exploring the application of coevolution-based methods and deep-learning for dimer contact prediction.

1.2.2 In silico methods for prediction of homodimer complex structures

Protein homo-oligomers, or self-associated proteins, are made up of multiple units of the same protein chain. Homo-oligomers constitute many biologically relevant complexes that perform essential functions including enzyme catalysis, cell signalling, and cell regulation (Ispolatov, 2005; Wang and Barth, 2015). The simplest class of homo-oligomers are homodimers, which are made up of two identical interacting protein chains. Homodimers make up the most prevalent group of homomeric complexes in the human proteome (Ispolatov, 2005) and previous studies have demonstrated homo-dimeric proteins can have up to twice as many interaction partners in eukaryotic protein-protein interaction (PPI) networks as non-self-interacting proteins (Ispolatov, 2005). This may be attributed to a number of factors. First, self-associated proteins have been shown to provide improved stability and control over active sites in proteins (Ispolatov, 2005; Singh and Jois, 2018). Second, self-association of proteins in proteomes can help minimize genome size (Singh and Jois, 2018). Finally, structurally similar proteins on average have higher interaction propensities than non-similar proteins (Ispolatov, 2005).

Figure 1.9 demonstrates an example of a local PPI network for SOD1, a homodimeric superoxide dismutase [Cu-Zn] that is a key antioxidant enzyme which destroys radicals normally produced in cells and has been associated with various human diseases, including cancer (Lin et al., 2013). The PPI shows physical interactions of SOD1 with 6 other complexes that are also homo-oligomers: SOD3, CCS, KARS and PARK7 which are all homodimers, and SOD2 and CAT which are homo 4-mers. This is one example of a local PPI which demonstrates the importance of homodimer and homo-oligomers in essential biological PPI networks.
**Figure 1.9: SOD1 PPI and Homodimer Structure:** a) Local PPI of SOD1, derived from StringDB [https://string-db.org/]. Lines between proteins indicate proteins are part of a physical complex, only interactions with medium or higher confidence (string score >0.4) are included. b) Homodimer PDB structure for SOD1 obtained from RCSB (PDB ID: 6FN8). Structure obtained using ChimeraX (Pettersen et al., 2021)

Characterizing the three-dimensional structures of homodimer complexes is often a useful and important step to understanding their biological function, interfaces, interactions, and mechanisms at a structural level (dos Santos et al., 2015; Wang and Barth, 2015; Uguzzoni et al., 2017).

As is the case with contact prediction in protein monomers, covariation-based methods have been shown to be useful for the identification of inter-chain and intra-chain protein residue contacts in homodimers (dos Santos et al., 2015; Wang and Barth, 2015; Uguzzoni et al., 2017). One of the main challenges in applying covariation-based methods to homodimer complexes, demonstrated in Figure 1.10, is the fact that the multiple sequence alignments (MSA) of the chains are identical and differentiation between inter-protein and intra-protein residue contacts is non-trivial without knowledge of the structure of constituent monomer subunits (dos Santos et al., 2015; Wang and Barth, 2015; Uguzzoni et al., 2017). This is because the MSAs are identical and therefore residue changes in identical MSA columns are also identical.
Due to homodimer protein chains having the same MSAs, covariation signals within protein chains and between protein chains are difficult to differentiate. In the case of parallel helical homo-oligomers, differentiating between inter- and intra-monomer contacts becomes even more difficult as both inter-protein and intra-protein contacts can be encoded by fairly short-range contacts, i.e., between residues close in the sequence (Wang and Barth, 2015). In other homodimers, certain residue-residue contacts can act as both intra- and inter- protein contacts - up to 32.3% of inter-protein contacts in some cases, can be both inter-chain and intra-chain contacts (Uguzzoni et al., 2017).

At the time the work outlined in this thesis was started, the differentiation of inter- and intra-protein contacts in homodimer complex structures without available monomer structure data was an open problem. As of 2017, the differentiation of intra-protein and inter-protein contacts remained unsolved for the cases where there were no crystal structures or close enough templates for the constituent monomers (Uguzzoni et al., 2017).

A number of template-based methods and a few covariation-based methods (summarized below) for homodimer prediction had been published shortly before the
thesis work in this PhD was started (Polyansky et al., 2014; dos Santos et al., 2015; Wang and Barth, 2015; Baek et al., 2017; Uguzzoni et al., 2017). None of the methods, however, had been able to successfully make contact and structural predictions of homodimer complexes using covariation information from sequence without using known monomer contacts. Sections 1.2.2.1 and 1.2.2.2 below summarize existing tools for homodimer contact and structure prediction and the limitations of each.

1.2.2.1 Non-Coevolution-Based Methods for Homodimer Prediction

One tool for homo-oligomer structure prediction was the Galaxy protein modelling suite of tools developed in the Seok lab (Lee et al., 2017). These tools, which are available through the Galaxy webserver (Baek et al., 2017), perform template-based modelling and/or ab initio docking depending on the availability of oligomer templates. The server can take either a monomer sequence or monomer structure as input, as well as information about the oligomeric state of the target, if available (Baek et al., 2017). To successfully predict the oligomer structure for a target protein, the Galaxy pipeline first determines the oligomeric state of the target (Lee et al., 2017), which is either derived from input information, or determined using a similarity-based method to detect protein homologs with at least 70% sequence identity and with known oligomeric states (Lee et al., 2017). Once the oligomeric states are predicted, another tool, GalaxyGemini (Lee et al., 2013), is used to detect homo-oligomer templates from the protein monomer structure or monomer sequence. A template-based modeller, GalaxyTBM (Ko, Park and Seok, 2012), is then used to predict the homo-oligomer structure using templates selected on the basis of sequence and tertiary or quaternary structure similarity.

When the Galaxy modelling suite was applied to 25 CAPRI targets from CAPRI round 30, an acceptable quality of prediction (by CAPRI standards) was obtained for about 15 targets, for which proper oligomer or monomer targets were obtained (Lee et al., 2017). In addition, for the most recent CAPRI targets (rounds 38-46), the GALAXY method was able to generate acceptable to medium quality models for about half of the easy homodimer targets, and 1 of the difficult targets (Park et al., 2020).

The authors concluded that overall, the model prediction quality depended greatly on the quality of the templates that were selected for that purpose (Lee et al., 2017). In addition,
in most cases, a medium-quality model could be generated only in cases where oligomer templates were detected similar enough to the target proteins with a TM-score greater than 0.6 (Lee et al., 2017). TM-scores (template modelling scores) measure the topological similarity of protein structures. Protein chains with TM-scores greater than 0.5 can be assumed to be in the same protein fold (Zhang and Skolnick, 2004). Useful oligomer structures could therefore only be detected in cases where templates could be found that met this threshold (Lee et al., 2017). The applicability of GalaxyHomomer is therefore limited to cases where templates could be found that were close enough to the input monomer. For these reasons, the method is not able to perform de novo homodimer structure prediction where no crystal structures or close templates exist for the constituent protein chains or the complex.

A second non-covariation-based method for homodimer prediction is PREDDIMER (Polyansky et al., 2014). The method was designed specifically for the prediction of transmembrane helical protein dimers (Polyansky, Volynsky and Efremov, 2012). It should be noted that several homodimer structure prediction studies focus on the specific case of transmembrane proteins because in the case of those homodimers, it is usually possible to identify lipid-facing residues, which reduces the likelihood of those residues being involved in inter-chain residue contacts (Polyansky, Volynsky and Efremov, 2012). In addition, several transmembrane proteins have sections that can be approximated as idealized helices, as shown in Figure 1.11, and this makes the structural prediction of those segments easier (Polyansky, Volynsky and Efremov, 2012).
Figure 1.11: PDB structures for dimeric ErbB3 receptor tyrosine kinase transmembrane domain: Solution NMR conformers for ErbB3 from species Homo Sapiens (PDB ID: 2L9U) obtained from the RCSB webserver (https://www.rcsb.org/structure/2L9U). Image generated using PyMol (Schrödinger, LLC, 2015).

To perform transmembrane dimer modelling, PREDDIMER took the amino acid sequence as an input and then built the transmembrane helix in an ideal three-dimensional conformation. The surface hydrophobicity was then mapped onto each of the transmembrane helices based on the molecular hydrophobicity potential (Polyansky, Volynsky and Efremov, 2012). The surfaces were projected on cylinders and overlaps between the helices for each subunit were scored using a sliding window (Polyansky, Volynsky and Efremov, 2012). The scoring function of the overlaps was used to pick the best dimer structures. For the cases where NMR structures were available, the NMR conformation of the dimer was found in the top three solutions from PREDDIMER, usually the first ranked (Polyansky et al., 2014). Although PREDDIMER showed very good prediction results, the method was limited to the prediction of transmembrane helical-like shaped homodimers. In addition, the paper demonstrated a correlation between the
availability of close templates for TM homodimers being modelled and the quality of the predicted TM models (Polyansky, Volynsky and Efremov, 2012).

Apart from these methods, a few docking approaches had previously been tested for the prediction of homodimer complexes (Dominguez, Boelens and Bonvin, 2003; Wiehe et al., 2007; Kozakov et al., 2017; Peterson et al., 2018). These docking methods can be used for both homodimer and heterodimer docking and are summarized in section 1.2.3.3 below.

1.2.2.2 Coevolution-Based Methods for Homodimer Prediction

In 2015, Wang et al. published a covariation-based method for the prediction of self-associated transmembrane helix homodimers (Wang and Barth, 2015). The paper described RosettaMembrane EFDOCK-TM, a tool for the prediction of homodimer transmembrane protein helical (TMH) structures using covariation information. This was accomplished by combining covariation information from MSAs with predictions for transmembrane helix binding surfaces predicted from the sequence (Wang and Barth, 2015). This was because the authors believed that the fraction of residues forming inter-chain short-range interactions would be enriched in the binding surface between the helices (Wang and Barth, 2015).

To obtain predictions, the authors created a pipeline that first used EVfold (Hopf et al., 2012) to predict all coevolving residue contact predictions along the transmembrane region. Then another method, LIPS (Adamian and Liang, 2006), was used to predict the TMH surface facing lipids, and all contacts along the predicted TMH surface were removed. The residue pairs with the strongest covariation signals were selected as pairwise distance constraints for folding and docking simulations (Wang and Barth, 2015). Folding and docking simulations were carried out using EFDOCK-TM, an iterative protocol for folding and docking of TMH oligomers that randomly selected sets of constraints and then enumerated all possible conformations with all possible combinations of constraints (Wang and Barth, 2015). One of the key findings of the paper was that filtering for the lipid-facing surfaces increased the rate of true inter-chain contact prediction from 16% to 74% (Wang and Barth, 2015). This finding is important as it
indicates knowledge of environmental conditions of predicted homodimers may be useful for increasing the quality of homodimer predictions.

Although the RosettaMembrane EFDOCK-TM method used coevolution-based information to predict homodimers, the method was limited to transmembrane proteins, relying on the identification of lipid-facing residues to correctly identify homodimer inter-chain chain contacts. The method therefore cannot be easily generalized to other non-TM homodimers.

Another paper (dos Santos et al., 2015) developed a covariation-based method to predict the association of homodimers. This was accomplished using covarying couplings extracted using a mean field implementation of direct coupling analysis (mfDCA) (Morcos et al., 2011). Once coevolutionary signals were obtained, predicted monomeric DCA contacts were filtered in two steps. First, Solvent Accessible Surface Area (SASA) for each residue in a contact was calculated as residues generally need to be significantly exposed to the surface of a dimer (Figure 1.12) in order to participate in protein-protein interactions (dos Santos et al., 2015)

![Figure 1.12](image)

**Figure 1.12:** Solvent-excluded surface of 11AS (asparagine synthetase mutant C51A): solvent-excluded surface of monomer subunit (left) and homodimer complex (right). Inter-protein interactions can only occur in solvent-accessible surfaces of the protein subunits
All predicted pairs where at least one residue had a SASA larger than 50% were removed. Next, the crystal structures or close templates of the monomer were used to filter out the top 100 predicted directly interacting pairs from mfDCA (dos Santos et al., 2015). The remaining contacts were then classified as inter-chain contacts, and were fed into a coarse-grained structure based method (Lammert, Schug and Onuchic, 2009) for complex structure prediction (dos Santos et al., 2015). The paper demonstrated that for cases with rich sequence information and where known monomer contact maps are available, it was possible to predict the homodimer interface contacts, and that these contacts could be used to build overall homodimer structures using molecular dynamics. The method relied on having crystal structures of the constituent monomers, however, and therefore could not perform fully de novo structure prediction of homodimers.

Another study, published in 2017, also attempted to apply covariation methods to the prediction of inter-protein contacts in homo-dimer structures (Uguzzoni et al., 2017). To accomplish this, the authors used a pseudo-likelihood maximization method for direct coupling analysis (plmDCA) (Ekeberg et al., 2013). Once all residue pairs in contact were predicted, the intra-protein contacts were filtered out by removing the predicted contacts compatible with the known monomer structures. In the situations where there was no experimental protein structure available for the monomer, homology modelling was used to try to model monomeric contacts. This meant that a close template was required in the cases where experimental structures were not available (Uguzzoni et al., 2017). The study also limited the prediction of inter-protein contacts largely to proteins with large interaction interfaces because, as was demonstrated by the results in the paper, larger interfaces (contact density > 0.3) tended to have much higher positive predictive values (PPV) for contacts than the covariation signals for medium (0.1 < contact density < 0.3) or small (contact density < 0.1) interfaces. Once the inter-protein contacts were predicted, docking simulations were run using HADDOCK (Dominguez, Boelens and Bonvin, 2003). To test the applicability of the method for prediction of inter- versus intra-protein contacts, the authors applied the pipeline to a domain family of response regulators in bacterial two-component signal transduction and demonstrated that it was possible to obtain reliable predictions of homo-dimeric structures with root mean square deviation (RMSD) < 1.1Å to native structures for these RR systems (Uguzzoni et al., 2017).
Although the method by Uguzzoni et al. demonstrated the usefulness of DCA-predicted inter-chain contacts for homodimer docking using HADDOCK, the method was still limited by the fact a crystal structure or close template was required for the monomer subunits of the RR homodimers being predicted. It was also found that up to 32.3% of all intra-protein contacts were also inter-protein contacts (Uguzzoni et al., 2017). This meant that filtering out the intra-protein contacts could lead to the loss of inter-protein contacts that may have been important for docking simulations. In addition, the paper focused on prediction of homodimers with large protein-protein interfaces (contact density > 0.3). Finally, docking predictions using HADDOCK were only done on response regulator proteins in bacteria, and therefore it was not possible to extrapolate the performance of this method on non-RR or eukaryotic targets.

The paper made an interesting observation, however, that intra-protein contacts that tended to be closer together in the monomer structure (<5Å) were less likely to also be inter-protein contacts as they were less likely to break intra-protein residue contacts to make room to interact in an interprotein manner (Uguzzoni et al., 2017). This could provide a potential filter: intra-chain residues predicted to be within 5Å of one another.

Overall, Uguzzoni et al., stated that as of the publication date of the paper (March 2017), the problem of differentiation between inter- and intra- protein contacts using covariation signals without monomer crystal structures or close monomer templates remained an unsolved problem.

Chapter 2 of this thesis therefore focused on determining whether it was possible to train deep-learning models to differentiate between inter-chain and intra-chain homodimer contacts using CNN or ResNet deep learning architectures. Work in chapter 3 explored the ability of deep learning methods to make accurate inter-chain and intra-chain contact predictions, agnostic of the homodimer type being predicted. Overall, work in chapters 2-4 of this thesis explored the application of covariation, meta-prediction, and deep-learning for the prediction of intra-chain and inter-chain contacts of homodimers.
1.2.3 *In silico* methods for prediction of heterodimer complex structures

The second half of the work outlined in this thesis focused on the prediction of the structures of heterodimer complexes. Heterodimers are the simplest hetero-oligomers and are made up of two protein subunits with different sequence and structure. Similar to homodimers, heterodimer complexes are essential to several cellular processes including enzyme pathways, cell signalling, DNA replication, and cell regulation. As is the case for homodimers, accurate prediction of quaternary structure of heterodimer complexes can be very useful for understanding the function and physiological mechanism of action of heterodimers.

One of the primary challenges in protein complex prediction that has been demonstrated through multiple CAPRI challenges is the accurate prediction of docking poses of heterodimer complexes, which involves predicting the correct orientation of heterodimer chains to one another (Janin, 2005; Lensink and Wodak, 2010; Wodak and Janin, 2017). Accurate prediction of inter-chain contacts can be very useful for improved heterodimer docking as predicted contacts can be used to inform the docking process in the form of input residues to tools such as HADDOCK (Dominguez, Boelens and Bonvin, 2003) or to rank docking decoys produced by tools such as ZDock (Wiehe *et al.*, 2007) or ClusPro (Kozakov *et al.*, 2017).

Prior to October 2017 a number of studies applied CBMs to the problem of heterodimer complex contact prediction (Weigt *et al.*, 2009; Hopf *et al.*, 2014; Ovchinnikov, Kamisetty and Baker, 2014; Feinauer *et al.*, 2016). Most of these methods were specifically interested in exploring the ability of CBMs to predict inter-chain heterodimer contacts.

An important step to using CBMs for contact prediction of heterodimer complexes, however, is obtaining paired multiple sequence alignments that maximize the number of interacting homologs (Hopf *et al.*, 2012; Ovchinnikov, Kamisetty and Baker, 2014; Bitbol *et al.*, 2016; Feinauer *et al.*, 2016; Taylor, 2017). This is to ensure that coevolution signals across heterodimer interfaces are maintained in the paired multiple sequence alignment, as correct sequence pairings would maintain the evolutionary history of coevolving protein
pairs (Bitbol et al., 2016; Taylor, 2017). Generation of accurate paired MSAs is essential to the application of CBMs for accurate prediction of heterodimer interface contacts (Figure 1.13). The first challenge that needs to be addressed before applying CBMs to heterodimer contact prediction, therefore, is the accurate pairing of multiple sequence alignments.

**Figure 1.13:** MSA pairing for inter-chain contact prediction using coevolution signals: MSAs of heterodimer proteins often have different numbers of homologs and/or paralogs. Here lines with the same colour indicate sequences from the same species and dashed arrows indicate physically interacting homologs. Columns in purple and green indicate residues in contact across the heterodimer interface.
1.2.3.1 Pairing of Multiple Sequence Alignments for Heterodimer Complex Prediction

The goal of MSA pairing for CBM contact prediction is to correctly pair interacting homologs in the MSAs of the complex chains. Since 2008, a number of tools had been developed to address the challenge of MSA pairing. Three of these methods (Burger and van Nimwegen, 2008; Procaccini et al., 2011; Bitbol et al., 2016) used covariation based techniques with simultaneous contact prediction to optimize sequence pairing in joint MSAs. In all methods used, the homologs in the two MSAs are grouped by species first and can only be paired with homologs within the same species (paralogs).

The first of these methods, developed by Burger and Van Nimwegen (Burger and van Nimwegen, 2008, 2010), involved the use of a Bayesian network method to predict interaction partners from multiple sequence alignments (Burger and van Nimwegen, 2008). To do this, potential interaction positions were first identified by determining the dependence of two positions, \( i \) and \( j \), in the MSAs (Burger and van Nimwegen, 2008). This was accomplished by looking at the ratio of likelihoods of the joint and independent models for the column (the probability of positions \( i \) and \( j \) being dependent, divided by the probability of \( i \) and \( j \) being independent, given the distributions of amino acid frequencies in columns \( i \) and \( j \)):

\[
R_{ij} = \frac{P(D_{ij})}{P(D_i)P(D_j)} \tag{Eq. 1.4}
\]

This measure is very similar to the mutual information of the two columns. To determine the most likely pairings of protein sequences in the joint alignments, multiple different joint alignments, \( a \), and their associated directed dependence trees \( T \), were used to determine the probabilities of seeing particular amino acid distributions in particular columns (Eq. 1.5).

\[
P(D|a, T) = P(D_r) \prod_{i \in r} P(D_i | D_{\pi(i)}, a, T) \tag{Eq. 1.5}
\]
Directed dependence trees are Bayesian graphical models or that encode the directed dependencies of variables in a model. Variables are represented on the nodes of the graph while edges indicate directed dependence of variables to other variables in the model.

Posterior distributions, \( P(a|D) \) were sampled over all possible joint alignments using a Markov chain Monte-Carlo (MCMC) sampling and the fraction of sampled assignments where two proteins, \( m \) and \( m' \) were interaction partners was tracked (Burger and van Nimwegen, 2008). Pairs with highest summed probabilities were assumed to interact with each other (Burger and van Nimwegen, 2008).

The method was tested on sequences extracted for a large collection of histidine kinases and response regulators (HK-RR) from NCBI using profiles from Pfam. When applied to a database of two-component bacterial systems with TCS protein sequences extracted from 399 bacterial genomes in NCBI, the prediction method had about 95% true predictions at 60% sensitivity, but when sensitivity increased to 75%, the fraction of true positive predictions was about 80% (Burger and van Nimwegen, 2008). This represented a significantly high accuracy for interaction partner prediction. Code for the method was unfortunately not available for benchmarking in the thesis work outlined in chapter 5.

A second, similar method was published a few years later (Procaccini et al., 2011). The method used a variation of message-passing DCA developed for protein-protein residue contact prediction (Weigt et al., 2009) to predict interacting protein partners from joint MSAs. This was done by first using a DCA method to measure the likelihood of particular columns in the MSA interacting, given an alignment with both known and unknown interaction partners (Procaccini et al., 2011). Due to computational restrictions, the 70 MSA columns with most correlated position pairs were selected for the statistical model used for partner pairing.

To determine the likelihood of a particular joint alignment, a likelihood score measured the log-likelihood of two sequences being paired given particular residues being in contact, and this likelihood score was then compared to a null model where the sequences were not paired (Procaccini et al., 2011). The joint alignment that maximized the difference of the likelihood score of the interacting model over the null was assumed
to be the most likely combination of pairs in the MSAs (Procaccini et al., 2011). Compared to methods based on mutual information alone, or sequence similarity, the method appeared to perform better, with an AUC of 91%, compared to 71% for the Burger and van Nimwegen method. This was important as it seemed to indicate that phylogeny or sequence similarity alone may not always lead to the best pairings of interaction partners. Unfortunately, the code for this method was also not available for benchmarking.

A method published in 2016, DCA-IPA (Bitbol et al., 2016) used an iterative approach to build more accurate joint multiple sequence alignments, with no prior knowledge of interacting partners. This was accomplished by starting with randomly paired alignments in a joint MSA and predicting the interacting residues using a pairwise maximum entropy model (Bitbol et al., 2016). Interaction energies between sequences were then calculated by summing up the inter-protein couplings assigned by the models (Bitbol et al., 2016). The interaction pairs with the highest interaction energies were added iteratively to the multiple sequence alignment, and the iterations were repeated until all sequences were paired. This was done by predicting the number of interacting residues in an MSA and pairing interacting partners that had the largest sum of interacting residues (Bitbol et al., 2016).

Without a priori knowledge of training pairs, HK-RR within the same species were randomly paired and these 5,064 random pairs were used to train the initial model (Bitbol et al., 2016). After a number of iterations, a TP fraction of 0.84 was obtained. For the larger, complete dataset of 23,424 HK-RR pairs from 2,102 species, the iterative algorithm yielded a true positive fraction of 0.93 with no prior knowledge of protein pairings (Bitbol et al., 2016). The model took at least 150 iterations to reach an acceptable threshold of contact prediction accuracy. One of the limitations of this approach is that it assumes a one-to-one pairing of the homologs in the joint MSAs (Bitbol et al., 2016).

A more recent version of the Bitbol IPA method, called MI-IPA was published in 2018. MI-IPA used the mutual information (MI) instead of DCA of paired MSAs as the objective function to be maximized (Bitbol, 2018). MI-IPA worked by pairing paralogous proteins from interacting protein families by approximately maximizing the mutual information of the paired alignment of the two proteins (Bitbol, 2018). MI-IPA used an iterative method to add sequences to the paired alignment, while attempting to maximize the pointwise
mutual information (PMI) of the interacting proteins (Bitbol, 2018). To do this, MI-IPA used a pairing score based on PMI (Bitbol, 2018).

Given a protein sequence A from MSA 1, and another protein sequence B from MSA 2, each consisting of 20 amino acids and gaps, MI-IPA calculated a pairwise score based on a concatenated sequence M of length L. The pointwise mutual information of any pair of residues $(\alpha, \beta)$ at a pair of sites $(i, j)$ is defined by the pairwise mutual formula shown in equation 3.1.

$$PMI_{ij}(\alpha, \beta) = \log \left[ \frac{p_{ij}(\alpha, \beta)}{p_i(\alpha)p_j(\beta)} \right] \approx \log \left[ \frac{f_{ij}(\alpha, \beta)}{f_i(\alpha)f_j(\beta)} \right]$$ (Eq. 3.1)

Here $p_i(\alpha)$ represents the probability of a particular residue $\alpha$ being in a particular position, $i$, and $p_{ij}(\alpha, \beta)$ represents the probability of the residue pair $\alpha, \beta$ being found in the sequence positions $i$ and $j$. When the number of sequences is much higher than 1, these can be approximated as the frequency ($f$) of the residue $\alpha$ at position, $i$, and the frequency of residues $\alpha, \beta$ at positions $i$ and $j$. The point mutual information for a set of concatenated sequences could therefore be calculated by obtaining the single-site frequencies and paired frequencies of the residues in the concatenated paired sequence alignment being analysed.

Averaging the PMI over all possible residue pairs in the concatenated alignment gave the mutual information between sites $i$ and $j$ ($MI_{ij}$). To get the pairing score of a particular concatenated sequence, the pointwise mutual information was summed for each pair of residues over each pair of positions across the length of the concatenated sequence. This score was then normalized by dividing by the depth of the concatenated paired sequence alignment to get a final normalized score, $S_x$.

The first iteration used a random pairing of sequences in the MSAs, with no assumptions made of interacting pairs. Next, sequence pairs are assigned within each species by maximizing the sum of $S_{AB}$ scores in the species, assuming one-to-one specific interactions, thereby maximizing $S_x$. A confidence score was given to each predicted pair by using the difference between the optimal assignment of pairs in the species, and the best alternative assignment that does not involve the pair being predicted. This process
involved maximizing the sum of the scores of all pairs in a species to make a one-to-one pairing. This resulted in a matrix of scores of all possible pairs, where one element had to be selected per line and column in a way that maximized the sum of the elements. The Hungarian algorithm (Kuhn, 2010), also known as the Munkres algorithm, was used to solve this problem exactly and efficiently (Bitbol, 2018). The protein pairs with the highest confidence scores for each species were then added to the concatenated alignment (CA), which was then used to calculate mutual information scores for the next iteration. This process was continued until all pairs of proteins were scored and added to the CA. MI-IPA showed improved performance over DCA-IPA, yielding deeper and more accurately paired alignments (Bitbol, 2018). The code for MI-IPA was available as a MATLAB code.

An algorithm published by Taylor (Taylor, 2017) used a phylogeny-based approach to pair protein sequences. The approach involved selecting particular “outgroups”, which were unique pairs of interacting proteins in species that had only one copy of each of the proteins of interest (singletons). Once this outgroup was selected, the phylogenetic distance of all the other sequences to the outgroup was determined, and this phylogenetic distance was then used to pair the protein sequence (Taylor, 2017). This was repeated about 10 times with different outgroups, and the protein pairings with the most consensus across multiple pairings were selected at the “correct” pairings (Taylor, 2017). Two different methods were used to pair sequences: the first a topology-based method that paired sequences based on the matching of topologies of the phylogenetic trees while the second was a distance-based method that paired sequences based on their sequence identity distance from the outgroups. During testing, it was found that the topology-based method tended to have a higher accuracy (Taylor, 2017). To test the accuracy of the method, databases were built by extracting sequences in PFAM that had three specified domains, splitting up the sequences by domain, and then applying the pairwise sequence matching algorithm to determine if correct pairings could be obtained (Taylor, 2017). Between the six cases analysed in the paper, the distance-based method had an accuracy between 39.7-95.8%, while the topology-based method had an accuracy between 53.1%-91.9% (Taylor, 2017). One of the main limitations of this method was the fact testing was done on datasets where each sequence had exactly one interacting partner, which is usually not the case for heterodimers. Unfortunately, the method did not allow for the pairing of MSAs of unequal depth.
Finally, two methods, GREMLIN-stitch (Ovchinnikov, Kamisetty and Baker, 2014) and EVComplex pairing (Hopf et al., 2014) took advantage of the fact interacting bacterial sequence pairs are likely to have smaller genomic distances than non-interacting pairs.

The GREMLIN-stitch method exploited the fact that interacting genes are often found near each other in bacterial genomes (Ovchinnikov, Kamisetty and Baker, 2014). The method used the fact most UniProtKB/TrEMBL accession ids are assigned sequentially for bacterial genomes to calculate the alphanumeric difference in uniprot id(s) (Δgene) as a proxy for gene distances of bacterial genomes. Δgene is the approximate number of genes that are between a gene pair. Pairs with Δgene values greater than 20 were excluded from analysis. Below an example is used to demonstrate how the method worked.

Given two sequences as shown in Figure 1.14, the method first identified genes that were on the same contig using the contig ID. The gene distances were calculated by looking at the difference between the gene loc ID. For example, Δgene = |00I9-00Ia| = 1 tells us these genes are 1 unit apart (gene count is in base-62 (0-9,a-z,A-Z)). Finally, if two pairs had similar Δgene values, gene pairings with the same direction (F-F or R-R) were chosen over pairings with different directions (R-F).

![Figure 1.14: UniProt IDs for gene distance calculation](image)

> B3MJ72 07aQ_00I9_F (from alignment A)
> B3MJ73 07aQ_00Ia_F (from alignment B)

Figure 1.14: UniProt IDs for gene distance calculation: Example sequence IDs used for calculation of distance in gene units for the GREMLIN-stitch method

Similar to GREMLIN-stitch, EVComplex pairing focuses on pairing prokaryotic bacterial sequences, and assumes that proteins in close proximity on the genome (i.e., on the same operon) are likely to interact. The pairing works by extracting genomic locations of protein sequences in the MSAs, and pairing sequences following 2 rules: 1. The coding
sequence of the sequences being paired must be on the same genomic contig, and 2. each pair must be the closest together in terms of genomic distance compared to all the other possible sequence pairings (in the same species) for the sequences in that pair. The European Nucleotide Archive dataset (ENA) (Pakseresht et al., 2014) is used to determine the sequence genomic locations and genomic contigs. Paired sequences are concatenated and are filtered to exclude all pairs that have genomic distances greater than 10k nucleotides (Hopf et al., 2014). Finally, to reduce the number of duplicates in the paired MSAs, alignment members are clustered and reweighted if 80% or more residues are identical (Hopf et al., 2014).

Compared to the Bitbol and Taylor sequence pairing methods, GREMLIN-stitch and EVComplex pairing had the ability to deal with pairing MSAs with differing depths within each species. In addition, both methods had either code available (GREMLIN-stitch) or a prediction server (EVComplex) that could be used to get sequence pairings.

As of 2019, when the heterodimer prediction work described chapter 5 was started, no comprehensive benchmarking of pairing methods existed. For this reason, work in chapter 5 of this thesis explored the performance of three of the methods described here (MI-IPA, GREMLIN-stitch, and EVComplex) that had available code or webserver, and that could be used for MSA pairs of different depth as was the case for most heterodimers benchmarked.

1.2.3.2 CBMs for heterodimer complex contact prediction

As of 2018, a number of studies had been published on the application of various coevolution-based methods to the prediction of residue contacts between the chains of two different protein subunits. A number of these studies (Weigt et al., 2009; Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014; Feinauer et al., 2016) focused on predicting structures of interacting proteins with interacting partners located close in bacterial genomes. This allowed these methods to produce large joint multiple sequence alignments where interacting partners can be paired based on their genomic distance in the bacterial genomes.
A 2008 paper (Weigt et al., 2009) attempted to predict the inter-protein residue contacts between the histidine kinase domain of the sensor kinase and their corresponding response regulators by looking at a multiple sequence alignment of 2,546 homologous pairs of SKs and their partner RRss. Chromosomally adjacent SK and RR sequences were concatenated into joint sequences and included into a larger joint MSA (Weigt et al., 2009). After applying a mutual information measure, 60 pairs that were determined to be the most highly correlated were selected and a semi-heuristic message passing global statistical method was used to estimate the marginal distributions for single positions and pairs of positions in the alignments (Weigt et al., 2009) According to the study, about 62% of the top inter-protein residue contacts predicted by the method were found to be within 8Å or closer in about 13 of the SK-RR protein complex structures evaluated (Weigt et al., 2009).

Another method (Hopf et al., 2014) attempted to apply mean-field DCA to the prediction of protein contacts for a set of binary protein-protein interactions in *E.coli* as predicted by yeast-to-hybrid and literature curation as well as complex structures in PDB. The multiple sequence alignments were generated using jackHMMER (Johnson, Eddy and Portugaly, 2010) to search the UniProt 2014_02 database. Complexes with gene distances >20 were removed and genes were concatenated based on distance in the *E.coli* genome (Hopf et al., 2014). The study used pseudo-likelihood maximization DCA, plmDCA (Marks et al., 2011), to calculated coupling parameters. For 15 protein complexes studied in the paper that had known 3D complex structures, EVcomplex had a minimum true positive (TP) rate of 0.21 with an average TP of 0.68. Docking models created using the contacts predicted by EVcomplex had an average RMSD of 2.453Å with the worst model having an RMSD of 5.5Å and the best model having an RMSD of 1.1Å (Marks et al., 2011).

Another study (Ovchinnikov, Kamisetty and Baker, 2014) used a previously published pseudo-likelihood based approach (GREMLIN) to identify covarying residues across protein-protein interfaces in the 50S ribosomal unit and 28 additional bacterial protein complexes. The multiple sequence alignments for each of the subunits was built using HHblits. The joint multiple sequence alignments in the paper were built by pairing sequences that were close together on the prokaryotic genomes using the GREMLIN-stitch algorithm. GREMLIN was then used to predict direct residue contacts between the protein subunits, then a Rosetta ab initio protocol (Raman et al., 2009) was used to
perform docking and folding simulations to create model complex structures (Ovchinnikov, Kamisetty and Baker, 2014). To evaluate the prediction accuracy, protein residues in the experimental complex structures were considered to be in contact if they were less than 12Å apart (Ovchinnikov, Kamisetty and Baker, 2014). The evaluation of protein complex structure prediction was done on 18 complexes with available experimental structures. Based on the evaluation, the average fraction of native contacts predicted by GREMLIN was 0.81 (0.42 to 0.99) (Ovchinnikov, Kamisetty and Baker, 2014). In addition, the models that were built with Rosetta had an average interface $\text{C}x$ RMSD of 3.32Å, with a range between 0.4 and 16.5Å (Ovchinnikov, Kamisetty and Baker, 2014). This was an improvement on previously existing methods for heterodimer docking predictions, however still indicated room for improvement.

Finally, a fourth study (Feinauer et al., 2016) studied the protein residue interactions in bacterial ribosomes and the Trp operon. The paper used plmDCA to predict statistical couplings between residue pairs (Ekeberg et al., 2013). Once more the genomic distances between sequences were used to pair sequences for the joint multiple sequence alignment. To predict contacts, plmDCA was used on concatenated paired MSAs to predict inter- and intra-protein contacts simultaneously (Feinauer et al., 2016). Docking and folding simulations were not performed in the paper, which meant RMSD statistics were not available.

The limitations of most of these methods of inter-protein contact prediction in protein hetero-oligomers was the fact most of the methods had been applied only to bacterial systems where genomic distance could be used to predict the interacting partners in a paired MSA. In addition, at the time the work in this PhD thesis was started, no comprehensive benchmarking had been done to compare the performance of these methods in terms of their performance on heterodimer inter-chain contact prediction.

For this reason, Chapter 5 of this thesis performed a benchmarking of 3 of the DCA methods that had code available (GREMLIN-Complex, EVComplex, and PSICOV) for the prediction of general heterodimer inter-chain contacts and compared these methods to a consensus-based contact predictor and two deep-learning based contact predictors.
1.2.3.3 Complex docking for heterodimer complex contact prediction

An important step in dimer complex prediction is the docking of complex subunits. In the last 3 decades, a large number of proteins docking methods have been published and released for docking of homodimer and heterodimer complexes. Protein docking involves solving the orientation of protein chains in a predicted complex pose that is physically realistic: i.e., avoids major clashes in protein backbone chains and follows physical principles such as electrostatics and shape complementarity. Docking methods can either be rigid-body docking, where protein structures being docked do not undergo any conformational changes in the docking process, or flexible docking, where protein structures can undergo conformational changes while being docked.

The bi-annual CAPRI assessments evaluate some of the most commonly used docking methods for the docking of protein-protein complexes. During the 7th CAPRI evaluation (Wodak, Velankar and Sternberg, 2020), a number of methods were shown to effectively dock both homodimers and heterodimers. The best performing methods during the CAPRI evaluation included HADDOCK (Dominguez, Boelens and Bonvin, 2003; Bonvin et al., 2018), ZDock (Vreven et al., 2020), GalaxyTBM (Ko, Park and Seok, 2012), InterEvDock, SwarmDock (Torchala et al., 2020), PyDock (Pallara et al., 2017), ClusPro (Kozakov et al., 2017), and LZerD (Christoffer et al., 2020).

Each of these methods can take the sequences or structures of constituent proteins and, using either crystal structures or homology-modelled structures of monomers, output predicted docked models for heterodimers. Most of these docking methods (Dominguez, Boelens and Bonvin, 2003; Yu et al., 2016; Padhorny et al., 2020; Vreven et al., 2020) output a number of predicted docking decoys, and also have internal scoring functions that perform ranking of the docking decoys (Pallara et al., 2017; Vangone et al., 2017; Peterson et al., 2018; Vreven et al., 2020).

Some of these methods, including PyDock (Jiménez-García, Pons and Fernández-Recio, 2013), SwarmDock (Torchala et al., 2020), and ZDock (Pierce, Hourai and Weng, 2011) use *ab-initio* docking to dock predicted protein structures. Generally, *ab initio* docking methods use a similar approach to one another: one protein is fixed in space while the
other is rotated and translated in relation to the first protein (Dominguez, Boelens and Bonvin, 2003). Protein docking poses are evaluated based on interface parameters such as electrostatics, interface energetics, and/or shape complementarity.

Other methods such as HADDOCK (Dominguez, Boelens and Bonvin, 2003) and InterEvDock (Yu et al., 2016), use bioinformatics input information such as residue constraints from CPORT (de Vries and Bonvin, 2011) in the case of HADDOCK or inter-chain coevolution predicted contacts in the case of InterEvDock (Yu et al., 2016).

HADDOCK (High Ambiguity Driven protein-protein DOCKing) (Dominguez, Boelens and Bonvin, 2003; Bonvin et al., 2018) is a flexible, information driven docking approach that makes use of biochemical and/or biophysical interaction data, introduced as ambiguous interaction restraints (AIRs) to drive the docking process. HADDOCK uses a combination of energetics and shape complementarity to dock proteins and can make use of AIRs to help the docking process. In the 7th CAPRI evaluation, HADDOCK was applied with reasonable success to the prediction of heterodimer and homodimer interfaces, and the HADDOCK server was the top ranked tool in the docking servers category (Koukos et al., 2020; Wodak, Velankar and Sternberg, 2020). The limitation of HADDOCK, however, is the fact it needs to be provided at least a few interaction restraints that it uses for docking, and therefore these restraints need to be reasonably accurate for HADDOCK to generate decent docking results (Bonvin et al., 2018).

GalaxyTBM (Ko, Park and Seok, 2012) is a template-based docking tool that detects any available complex templates for protein complexes being docked, and uses the templates to extract chain orientations that are used to dock protein-protein complexes. Although the method has fairly good performance in cases where templates are available for the complex being docked, it cannot dock complexes where complex templates are not available in the PDB (Ko, Park and Seok, 2012).

ZDock is a fast Fourier transform (FFT) based rigid-body docking protocol which searches all possible binding modes for two proteins in translational and rotational space (Wiehe et al., 2007; Vreven et al., 2020). ZDock uses an energy-based scoring function which combines IFACE statistical potentials described by Vreven et al. (Vreven et al., 2020), shape complementarity, and electrostatics to evaluate each pose explored and outputs a
ranked list of the best scored docking poses (Vreven et al., 2020). Poses are then re-ranked using the Integrative Ranking of Protein-Protein Assemblies, IRaPPA (Moal et al., 2017). IRaPPA uses physicochemical descriptions and combines several metrics using ranking support vector machines (R-SVMs) (Moal et al., 2017). Previous evaluations (Moal et al., 2017; Vreven et al., 2020) have shown that the IFACE + IRaPPA method outperforms most other physics-based interface scoring methods for the ranking of docking decoys.

ClusPro (Padhony et al., 2020), and pyDock (Pallara et al., 2017) are also ab-initio rigid body docking methods that use FFT for docking. pyDock is built on an earlier version of the FTDock method (Moont, Gabb and Sternberg, 1999), and similar to ZDock, does rescoring of poses using the IRaPPA. PyDock predictions are obtained by providing the monomer crystal structures to the PyDockWeb server, which outputs the 10 top ranked decoys for each complex. ClusPro uses the FFT-based program PIPER, and the 1000 lowest energy structures generated are clustered using pairwise RMSD, with the centres of the largest clusters [rather than the lowest energy structures] used for decoy clustering.

InterEvDock (Yu et al., 2016) is a protein docking server that integrates evolutionary information into the docking process and uses the InterEvScore (Andreani, Faure and Guerois, 2013) potential to rank models generated using FroDock (Garzon et al., 2009). The InterEvScore (Andreani, Faure and Guerois, 2013) potential combines statistical potentials with evolutionary information for the monomers in the complex as well as atom-based statistical potential.

A recent benchmark (Vreven et al., 2015), compared 4 docking tools, HADDOCK (Dominguez, Boelens and Bonvin, 2003), PyDock (Pallara et al., 2017), SwarmDock (Torchala et al., 2020), and Zdock (Pierce, Hourai and Weng, 2011) on 55 heterodimer complexes in the Protein Docking Benchmark 5.0. The benchmark showed that, taking the top 10 predicted docked decoys output by each method, acceptable or better docked models could be generated for between 11 (PyDock) and 21 (SwarmDock) complexes. This number dropped much lower when only the top predicted model was considered with acceptable or better docked models generated for between 3 (ZDock) and 9 (SwarmDock) complexes.
Overall, most existing inter-chain docking methods predicted correct complex docking for less than 20% of heterodimer complexes. Only one existing docking method, InterEvDock, uses coevolution information to perform docking decoy ranking. None of the methods available prior to the start of the docking work done in this thesis used deep-learning contact or distance prediction to perform docking or ranking of protein-protein complexes.

For this reason, work in chapter 5 also explored the use of inter-protein contact predictions for ranking of heterodimer docking decoys.

1.2.3.4 Machine Learning and Deep Learning Concepts for Complex Contact Prediction

In this thesis, several deep learning models were developed and trained for the prediction of residue-residue contacts. There are several important concepts that had to be used carefully to ensure these models were trained and evaluated accurately for prediction of inter-chain protein contacts.

First, an important consideration for the training and evaluation of deep learning models is the creation of robust and non-redundant training, validation, and test sets. In deep learning, the training set is the set that is used to train the model, usually by backpropagating differences between the true values in a training set and predicted values in a training set through the model being trained. Some of the goals when creating training sets are to make them as big and diverse as possible, and to ensure the diversity of real complexes in terms of sequence and structure are captured in the training sets. The validation sets created and used throughout this thesis were used to measure the performance of the deep-learning model after each epoch.

Throughout this thesis, a training epoch consists of running the deep learning model (with backpropagation and model updates) over the whole training set, or over a certain random subset of the training set. The training and validation loss and performance metrics (i.e. MCC) are evaluated at the end of each epoch.
Validation sets were created in such a way as to ensure no overlap in terms of sequence or structure with the training or test sets. Validation sets were used to ensure that models were improving over training epochs, and that the deep learning models were not over-training on the training set. Over-training is often indicated by validation loss increasing while the training loss decreases over training epochs. Finally, the test sets are used to evaluate the performance of the trained deep-learning models on protein complexes that the model has not yet been trained or validated on. Ensuring no overlap between the training and test sets is essential for demonstrating that a deep learning model has truly learnt to generalize.

1.3 Thesis Overview

The goal of this thesis was to explore the application of coevolution-based methods and deep learning for the prediction of homodimer and heterodimer complexes. In this thesis, CBMs and deep-learning methods are used to predict tertiary structure and quaternary contacts to be used for docking of dimer complexes. Previous work had explored some limited cases where coevolution-based techniques could identify interface contacts in protein complexes (Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014; Wang and Barth, 2015) and had demonstrated that deep learning methods could be used to improve the prediction of protein monomer contacts (Schaarschmidt et al., 2018). Here, the application of deep-learning methods trained using combined inputs from CBM predictions was extended to the prediction of protein dimers.

Chapter 3 of this thesis explored the application of two deep learning architectures, based on convolutional neural networks (CNNs) and residual neural networks (ResNets) for the differentiation of homodimer inter-chain contacts and intra-chain contacts. This was done by training these models to classify known homodimer contacts and determining whether classification of the two contact types was possible.

Work in chapter 3 focused on the development and training of three deep learning architectures for the improved prediction of homodimer interfaces. The best performing deep learning technique was used to generate inter-chain contacts that were used to test
the ranking of homodimer docking decoys in chapter 4. In chapter 4, the effectiveness of using deep-learning predicted inter-chain contacts for ranking of homodimer docking decoys was compared to existing force-based methods for decoy ranking.

Work in chapter 5 benchmarked existing methods for the prediction of heterodimer complexes. In chapter 5, various existing MSA pairing methods, inter-chain contact prediction methods, and heterodimer docking protocols were explored to compare their performance on the problem of homodimer docking without complex structure templates. The limitations of existing methods for general heterodimer docking are explored and outlined.

Chapter 6 explored the application of newer deep learning approaches, including gated recurrent units (GRUs) and protein language models (LMs) for the prediction of inter-chain contacts in heterodimers. The methods were trained on a new heterodimer structure dataset and evaluated on a test set to determine whether these methods were able to effectively predict inter-chain protein contacts.

Finally, work in chapter 7 explored the application of a recently published AlphaFold2 method for the prediction of heterodimer complexes. The performance of the AF2 method for heterodimer complex prediction was studied to determine the improvement of the method over existing state of the art (SOTA) heterodimer predictors.

Overall, a large number of prediction tools are explored for the de novo prediction of protein dimer complexes. The performance and limitations of these methods was explored, and the usefulness of adding deep-learning to coevolution-based predictions for dimer interface prediction was demonstrated. This thesis work demonstrates some improvements from deep-learning for protein dimer prediction, but also demonstrates remaining challenges for the de novo prediction of protein complexes.
2. Homodimer Complex Contact Classification

2.1 Introduction: Homodimer Contact Classification

Protein homo-oligomers, or self-associated proteins, are a class of protein complexes made up of two or more subunits of the same protein. Solving the structures of homodimer complexes can be challenging even when structures of their constituent proteins are known (Janin, 2005; Kong et al., 2020).

*De novo* homodimer complex prediction involves making structural predictions for homodimer complexes that do not have available crystal structures or close templates. The aim of *de novo* structure prediction for homodimers is to accurately predict both the monomer subunits and the overall homodimer complex. For the most part, this means being able to make accurate predictions of the interface between interacting chains. This can be done by predicting pairs of amino acid residues that are in contact within monomer protein chains (intra-chain residue pairs) and between the protein chains (inter-chain residue pairs).

As described in Chapter 1, covariation based methods (CBMs), or evolutionary statistical methods, have been developed to identify covariation signals in protein multiple sequence alignments and can be leveraged to identify interacting residues from MSAs (Marks et al., 2011; Jones et al., 2012; Ekeberg et al., 2013; Seemayer, Gruber and Söding, 2014; dos Santos et al., 2015; Uguzzoni et al., 2017; Kandathil, Greener and Jones, 2019). At the time the work described in this chapter was started, a number of studies had demonstrated the existence of covariation signals between interacting residues of homodimer interfaces (dos Santos et al., 2015; Wang and Barth, 2015; Uguzzoni et al., 2017).

CBMs, however, were shown to have limited application for *de-novo* prediction of both monomer subunit and homodimer complex prediction when crystal structures or close template structures did not exist for monomer subunits (dos Santos et al., 2015; Wang and Barth, 2015; Uguzzoni et al., 2017). This was because both chains of homodimer
complexes had the same MSA, and therefore differentiating between inter-chain and intra-chain residue contacts without prior knowledge of contacts in the monomeric structure of the individual subunits was a non-trivial problem (dos Santos et al., 2015; Uguzzoni et al., 2017). Differentiating between the two contact types, however, is essential to using contacts to build homodimer complex models. In this chapter, deep-learning methods were explored for the differentiation of inter-chain and intra-chain residue contacts.

Prior to 2017, deep learning techniques had been shown to be quite effective in tackling challenging prediction problems (LeCun, Bengio and Hinton, 2015). Several advances in neural network training methodologies (Rumelhart, Hinton and Williams, 1986; G. E. Hinton et al., 2012; Russakovsky et al., 2014; Ioffe and Szegedy, 2015) as well as advances in computing hardware (Raina, Madhavan and Ng, 2009; Kim et al., 2017) had allowed training of much deeper neural network models than was previously possible (Steinkraus, Buck and Simard, 2005; Kim et al., 2017). This allowed models to learn far more abstract and complex patterns than existing machine learning methods. Deep learning techniques had been shown to be very effective for complex tasks such as image recognition and speech recognition (G. Hinton et al., 2012; Russakovsky et al., 2014).

A number of publications had demonstrated benefits of applying deep learning and meta-predictors for the prediction of protein contacts in protein monomers (Skwark, Abdel-Rehim and Elofsson, 2013; Ma et al., 2015; Wang et al., 2016; Jones and Kandathil, 2018). No published papers, however, had yet explored the application of deep learning methods for the prediction of homodimer contacts.

The work outlined in this chapter therefore explored the application of deep-learning methods previously developed in the Jones lab for the classification of homodimer contacts as inter-chain or intra-chain contacts. The inputs to these deep-learning methods combined predictions from covariation-based methods, secondary structure predictors, mutual information, and sequence profiles (Greener, Kandathil and Jones, 2019; Kandathil, Greener and Jones, 2019). The goal of the work outlined in this chapter was to determine whether convolutional neural networks (CNNs) or residual neural networks (ResNets) could differentiate between inter-chain and intra-chain contacts for known sets of contacts in homodimer complexes.
To do this, two homodimer datasets were curated for the training and testing of the deep-learning methods developed. The goal of dataset curation was to compile comprehensive, robust, and non-redundant sets of homodimer complexes for the training, testing, and validation of the deep learning methods explored in this chapter. Two datasets, called homodimer dataset 1.0 (HD1.0) and homodimer dataset 2.0 (HD2.0), were obtained and curated as described in section 2.2 of this thesis.

Once these datasets were obtained, a number of experiments were carried out to determine how well deep learning methods were able to classify known sets of homodimer contacts as either inter-chain or intra-chain contacts. These experiments are outlined in section 2.3 of this chapter. Overall, the work in this chapter showed the potential for improved homodimer contact classification as well as areas for improvement in the task of differentiating between inter-chain and intra-chain contacts.

2.2 Curating Datasets for Homodimer Prediction

Two homodimer datasets were curated for the training and testing of deep-learning models for homodimer complex contact prediction. The first of these datasets, hereafter referred to as homodimer dataset 1.0 (HD1.0), was curated using stringent filtering criteria on all structures available in the Protein Data Bank (PDB) before 27th October 2017. The methodology used to extract and curate of HD1.0 is outlined in section 2.2.1. The second dataset, homodimer dataset 2.0 (HD2.0), was curated with a less stringent filtering criteria and used all homodimer structures available in the PDB before 24th May 2019. This resulted in a larger number of homodimer structures available for training. The methodology used for the curation and processing of HD2.0 is outlined in section 2.2.2.

2.2.1 Homodimer Dataset 1.0: Curation of a Non-redundant Set of Homodimers

Dataset curation for the first homodimer dataset was done with the help of Erik Pfeiffenberger from the lab of Paul Bates’ at the Francis Crick Institute. The PDB extraction, curation and cleaning pipeline detailed here was used to create a non-
redundant set of homodimer complexes to be used for training, testing, and validating of our deep learning methods for homodimer contact classification.

The aim of the dataset curation process was to make the training set as large and comprehensive as possible while limiting the redundancy of the set. Redundancy was limited to avoid overtraining on families or folds over-represented in the Protein Data Bank (PDB). The training, test, and validation sets were split in a way that ensured no overlap between the training, test or validation sets in terms of topology or folds to ensure robust cross-validation of trained models.

2.2.1.1 Obtaining PDBs and Homodimer Complex Curation

A list of available homodimer PDB structures was obtained using the advanced search of the RCSB website (Berman, 2000). Entries in the PDB were filtered to obtain all available protein crystal structures which met the following criteria: a resolution less than 2.5Å, a 2-mer homomeric state, and a representative sequence identity of 30% (as defined by the RCSB website at the time). Additionally, only structures composed of protein chains were kept in the set. The search was performed on 27th October 2017. This search resulted in a list of 6,178 PDB identifiers that were further curated and processed to obtain HD1.0.

Once PDB IDs were obtained, the homodimer structures were checked and curated by Erik Pfeiffenberger. Erik checked the completeness of the chains of the structures, made sure sequences in the complexes matched, and checked the biological assembly files in RCSB to ensure the structures were not incorrectly labelled as homodimers. 1,612 PDB structures were removed due to mismatches in the homodimer chains, or due to one or both homodimer chains being incomplete. Another 122 structures were removed because the biological assembly files indicated the structures were not homodimers. 4,424 PDB homodimer structures remained after mismatches, incomplete structures, or incorrectly labelled structures were removed from the set.
2.2.1.2 Dataset Filtering by Sequence Redundancy and Alignment Depth

The remaining 4,424 PDB identifiers were then run through the PISCES server from the Dunbrack lab (Wang and Dunbrack, 2003). The PISCES server enables better culling of sets of protein sequences by sequence identity and structural quality criteria (Wang and Dunbrack, 2003). The PISCES method uses a hidden Markov model (HMM) from the Hhblits program to perform a multiple sequence alignment and can identify sequence overlap even in cases where there are very low sequence identities (i.e. 15-30%) (Wang and Dunbrack, 2003). The PISCES server therefore results in better and more robust culling than most servers which use BLAST (Wang and Dunbrack, 2003).

Passing the PDB IDs through PISCES resulted in a more robust culling of sequences by structure quality and mutual sequence identity. The structure list was filtered by chain to ensure sequences in the set had 25% sequence identity or less, were greater than 40 residues in length, and had a resolution less than 3.0Å. Structures obtained using methods other than X-ray crystallography were excluded, as well as chains that had only C alpha (Cα) atoms in their structures. The criteria that were used to cull the input PDBs are summarized in Table 2-1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence percentage identity</td>
<td>≤ 25%</td>
</tr>
<tr>
<td>Sequence chain length</td>
<td>40-10000</td>
</tr>
<tr>
<td>Resolution</td>
<td>≤ 3.0</td>
</tr>
<tr>
<td>R-factor value</td>
<td>≤ 0.3</td>
</tr>
<tr>
<td>Non-X-ray entries</td>
<td>Exclude</td>
</tr>
<tr>
<td>CA-only entries</td>
<td>Exclude</td>
</tr>
<tr>
<td>Cull PDB by</td>
<td>Chain</td>
</tr>
</tbody>
</table>

Table 2-1: Pisces Filtering Criteria: Filter specifications for PDB culling using the PISCES server. PDBs were culled by chain for mutual sequence identities < 25%, lengths between 40 and 10,000 residues, resolutions < 3Å and excluding all entries with only Cα coordinates
Once the structures were run through PISCES, 3,365 PDB IDs remained in the homodimer dataset.

Multiple sequence alignments (MSAs) were then generated for these 3,365 homodimers using the HHblits method for remote homology detection (Remmert et al., 2012). MSAs were generated by running HHblits on the UniProt20 version released in February 2016 (Breuza et al., 2016). HHblits was run with 3 iterations, an E-value threshold of 0.001, and a minimum coverage of the query sequence of 20% (Remmert et al., 2012). Previous studies had shown that low MSA depth could result in poor contact prediction (Marks et al., 2011; Hopf et al., 2014; Kandathil, Greener and Jones, 2019) so MSAs with effective sequence counts (Meff) less than 30 were filtered out.

2.2.1.3 Dataset Splitting: Training, Test, and Validation Set Creation using ECODAnalyse

Once the filtering steps were completed, 3,182 remaining PDB IDs were split into a training, test and validation set. It was important at this stage to ensure no overlap in protein fold (i.e., topology) between the training, test, and validation sets.

Sequence redundancy in the set had been minimized by ensuring less than 25% mutual sequence identity. It is possible, however, to have structural homologues even in cases where sequence identity is less than 25%. It was necessary to ensure that there was no overlap of fold or topology between the training, test, and validation sets.

To ensure no structural overlap between the training, test, and validation sets, the evolutionary classification of protein domain structures (ECOD) database was used to check for PDB homodimer chains in the same ECOD T (Topology) class (Cheng et al., 2014). Protein domains in ECOD are classified in a hierarchical manner based on their evolutionary relationships (Cheng et al., 2014). The T-groups of ECOD are made up of domains with similar homology and similar topology and T-groups therefore distinguish between domains with notable differences in topology (Cheng et al., 2014). In cases where structures have similar homologies but different topologies, domains are put in different T-groups (Cheng et al., 2014). It should be noted that both the ECOD and the
CATH Protein Structure Classification database were considered for the filtering of training and tests. ECOD was chosen because at the time of analysis, it had a greater overlap with the chains in our dataset.

An initial training, test and validation set was obtained by running a tool developed in the Jones lab, called ECODanalyse, to eliminate PDB chains from the training set that overlapped in any of the first three ECOD classes with the test or validation sets. Random selection of homodimers in the train, validation, and tests sets resulted in a significant number of examples removed from the sets by ECODanalyze.

As it was desirable to maximise the number of homodimers in the training set, a different tool, called ECODclust, was used to cluster homodimer structures in the same ECOD T-level. These clusters were then sorted by size from smallest to largest. The first 200 entries, all of which had a cluster size of 1, were selected to be the testing and validation set. This resulted in a non-redundant dataset consisting of 2,982 PDB structures in the training set, 100 PDB structures in the validation set, and 100 structures in the test set. The pipeline used to extract the training, test, and validation sets is shown in Figure 2.1.
This process of filtering for sequence identity and domain structure overlap ensured there was no sequence or structure overlap between the training, test, and validation sets of HD1.0. This meant that if a deep learning model was able to perform successfully on the test and validation sets, it has learnt to generalize and could make predictions for sequences and structures that it has not “seen” (i.e., been trained on) before.

2.2.1.4 Contact Map Generation

Once the training, test, and validation sets from RCSB had been created, the contact maps were extracted for each of the homodimer complexes. The contact maps were 1-channel matrices of dimension L x L where L is the length of the homodimer chain. The contact maps contained information about residues in contact in the complexes.
For each complex, two contact maps were extracted: one that contained the inter-chain contacts (residue contacts between the two chains of the homodimer) and a second that contained the intra-chain contacts (residue contacts within the monomer chains of the complex). Residue pairs were classified as being an intra-chain contact if the residue pair had Cβ atoms (Ca in the case of glycine) within 8Å of one another in the same monomer chain. Residue pairs were classified as being inter-chain contacts if the residue pair had Ca atoms within 10Å of one another between homodimer chains. Figure 2.2 shows an example of the intra-chain and inter-chain contact maps for Asparagine synthetase mutant C51 (structure shown in figure 1.1), which was one of the homodimer structures in the HD1.0 training set.

**Figure 2.2: Intra-chain and inter-chain contact maps for Asparagine synthetase mutant C51 (PDB ID 11AS):** a) Intra-chain contact maps for homodimer 11AS, chain A. X and y axes correspond to residue position of chain A. b) Inter-chain contact map for 11AS. X axis corresponds to length of chain A, y axis corresponds to residue position in chain B

### 2.2.2 Homodimer Dataset 2.0: Curation of a Newer and Less Stringent Set of Homodimers

One of the main considerations when it comes to training deep-learning methods is the size of the training set. In many cases larger training sets that maintains overall diversity
of training set can result in better deep learning models. For this reason, a second, less stringent curation method was used to create a larger homodimer training set.

The Homodimer Dataset 1.0 was created using very stringent sequence and structure redundancy criteria to ensure no folds were over-represented in the training set. The second dataset, Homodimer Dataset 2.0 (HD2.0), was created at a later date (24th May, 2019) with less stringent filtering criteria to reduce the number of homodimer complexes culled from the set. Sequence redundancy was still used to filter out redundant complexes to avoid significant over-representation of protein or domain families in the training set. The goal of compiling this second homodimer dataset was to explore the impact of using a larger, less stringently filtered training set on the ability of our deep learning methods to perform homodimer prediction tasks.

2.2.2.1 Obtaining Homodimer PDBs

A different protocol was used to create HD2.0 than the one used for the first dataset. An initial list of available homodimer structures was obtained from the RCSB website using the advanced search feature. Entries were filtered to obtain homodimer crystal structures with a resolution less than 3.5Å, a homo-oligomeric state and released before 24th May 2019. This search resulted in 43,917 homo-oligomer structures.

Homo-oligomers with more than two constituent subunits were included in the search, with only two of the chains from each homo-oligomer retained for further processing.

2.2.2.2 Dataset Filtering by Sequence Redundancy and Alignment Depth

Once the list of homodimer PDB IDs was obtained, a processing pipeline was used to process and clean the PDB files, and to check that they met certain filtering criteria. First the PDB files were processed using the MODELLER (Fiser, Do and Šali, 2000; Eswar et al., 2006) ’complete_pdb’ function to fill in any missing atoms using internal co-ordinates. Next, the PDB structures were checked to ensure they had 2 chains, chain lengths between 30 and 600, and sequence similarity greater than 99%. Filtering out structures that did not meet these criteria resulted in a list of 39,213 homo-oligomer complexes.
Once the list of homodimer PDBs was obtained from RCSB and filtered by our selection criteria, the mmseqs2 method (Steinegger and Söding, 2017, 2018) was used to cluster the resulting homodimer sequences by 40% sequence identity, and 50% sequence coverage. Representative structures were selected from each cluster, resulting in 7,824 structures.

After filtering the PDB list and obtaining 7,824 representative homodimer structures, MSAs were generated for the sequences of all the homodimer chains. This was done using HHblits against the 2018_08 version of the UniClust30. The alignments were generated by running HHblits with 3 iterations, an E-value threshold of 0.001, a minimum coverage of the query sequence of 20%, and a maximum pairwise sequence identity of 99%. All homodimer chains with an effective sequence count (Meff) less than 20 were filtered out. This resulted in a list of 6817 homodimers with MSAs with Meff > 20.

2.2.2.3 Dataset Splitting: Training, Test, and Validation Set Creation using ECODclust

After cleaning, clustering, and filtering, the homodimer structure list was split into training, test, and validation sets. The ECODclust method was used to cluster the homodimer structures in the dataset by ECOD T-class (Cheng et al., 2014). Clusters were then ordered in increasing size. A set of the top 200 clusters, all of which had only one structure in the cluster, was split evenly into the validation and test sets. The remaining 6817 structures were put in the training set. The pipeline used to create the training, test, and validation set for HD2.0 is shown in Figure 2.3.
Once the homodimer training, test, and validation sets had been created the contact maps were extracted for the homodimer complexes. Inter-chain and intra-chain contact maps were extracted using the same method outlined in section 2.2.1.4.
2.3 Coevolution and Deep Learning for Homodimer Complexes Contact Classification

As discussed in chapter 1, it is often challenging to differentiate between inter- and intra-chain contacts using coevolution-based methods in homodimer complexes. Differentiating between inter- and intra-chain contacts using co-evolution signals alone is difficult without information about monomer intra-chain contacts (also referred to as tertiary contacts).

The experimental work in this chapter explored the application of two deep learning architectures – convolutional neural networks (CNNs) and residual neural networks (ResNets) for the differentiation of known homodimer contacts. The purpose of this experiment was to determine if it was possible to differentiate between inter-chain and intra-chain residue pairs given a set of known crystal contacts. Although this is not of much use in the case of de-novo homodimer prediction since it relies on having a set of the known contacts for homodimers, it was important to establish that deep learning methods could indeed differentiate between the two contact types before trying to use such methods for de novo homodimer prediction.

2.3.1 Deep Learning Model 1: CNNContactClassify Model for Classification of Known Homodimer Contacts

The first experiment carried out as part of the work summarized in this chapter used a convolutional neural network (CNN) model for homodimer contact classification. The goal was to explore whether a CNN-based model that had previously been demonstrated to be capable of making intra-chain contact predictions (Jones and Kandathil, 2018) could be leveraged to classify known homodimer contacts. Specifically, a CNN-based method, called CNNContactClassify, was trained to classify known homodimer contacts (i.e., a list containing both inter-chain and intra-chain contacts in homodimer complex) as being either inter-chain or intra-chain. This method was trained, validated, and tested on the Homodimer Dataset 1.0 summarized in section 2.2.
2.3.1.1 Methodology: CNNContactClassify Model Architecture and Training

The CNN module used for the CNNContactClassify was very similar to the DeepCov method previously developed in the Jones lab (Jones and Kandathil, 2018). Convolutional neural networks had previously been demonstrated to be effective in several domains due to their ability to learn complex local spatial patterns (G. Hinton et al., 2012). Convolutional neural networks use convolutional layers to detect spatial patterns in inputs, and an explanation of convolutional layers can be found in appendix 2 of this thesis.

The CNNContactClassify model consisted of a fully convolutional neural network with 60 input channels, hereafter referred to as map inputs. These map inputs contained predictions from contact prediction methods and secondary structure predictions.

One of the challenges in training the contact classifier was the fact there was a very large class imbalance of intra-chain and inter-chain contacts in the dataset. Specifically, there were 68,663 inter-chain residue contacts and 1,614,44 intra-chain contacts in the training set. This meant that about 4.25% of all the contacts in our set were inter-protein contacts. For this reason, the contact classifier was trained on an equal number of intra- and inter-protein contacts at each epoch (i.e., over each run over the training set). This was done by creating crops of 66 x 66 residues around the residue contact being classified – thereby generating inputs with dimensions of 60x67x67. The model was then trained on a balanced set made up of an equal number of crops around the inter-protein contacts and crops around the intra-protein contacts.

CNNContactClassify Architecture

The CNN module of CNNContactClassify consisted of a 2D convolutional layer followed by a batch norm layer (Ioffe and Szegedy, 2015) and max pooling. This was followed by eight 2D convolutional blocks, a batch norm layer, 2 dense layers made up of 256 maxout units and 50% dropout. This was then followed by a fully connected output layer with 2 softmax units. CNNContactClassify was implemented using the Lasagne (Dieleman et
and Theano (The Theano Development Team et al., 2016) libraries in python. The architecture of CNNContactClassify is shown in Figure 2.4.

**Figure 2.4:** Architecture of the CNNContactClassify: model used for binary contact classification of inter-chain and intra-chain homodimer residue contacts. Numbers in parenthesis describe parameters for layers (kernel size for convolutional and max pooling layers, and maxout units for dense layers). Numbers on the right show the output dimensions of each layer.
**Input map generation**

The input features used to train CNNContactClassify were generated using the same input features as those used by DeepMetaPSICOV (Kandathil, Greener and Jones, 2019). These features are generated by combining information from a number of prediction tools. Contact predictions for all the homodimer targets were extracted with three contact prediction methods: PSICOV, CCMpred (Seemayer, Gruber and Söding, 2014), and FreeContact (Kaján et al., 2014). The secondary structure predictions for the targets were also obtained using PSIPRED (McGuffin, Bryson and Jones, 2000).

The DSSP (Hooft et al., 1996) tool was used for hydrogen bond estimation from atomic-level coordinates from PDB structures and were used to generate .tdb files. These .tdb files were used to get inter- and intra-protein contacts. Figure 2.5 shows the pipeline that was used to generate the input map features for the deep learning models. Mutual information and normalised mutual information, PSIBLAST sequence profile and the Shannon entropy of the MSA columns are also provided as input features.

![Image of the pipeline](image.png)

**Figure 2.5: Homodimer Deep Learning Inputs:** Input features used to generate input .map files used to train deep learning models

All these features were collated into input tensors of dimension 60 channels x L x L, where L was the residue length of each monomer subunit. Table 2-2 shows the information captured in each of the channels of the input tensors.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Residue feature (1) or residue pair feature (2)</th>
<th>Dimensionality</th>
<th>Channels in input tensor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PSIBLAST sequence profile</strong></td>
<td>1</td>
<td>21</td>
<td>42</td>
</tr>
<tr>
<td><strong>Mutual Information</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Normalised mutual information</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Mean contact potential</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>PSICOV contact scores</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>FreeContact (mfDCA) contact scores</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>CCMpred (plmDCA) contact scores</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>PSIPRED secondary structure</strong></td>
<td>1</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td><strong>Shannon entropy of alignment columns</strong></td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>SOLVPRED solvent accessibility</strong></td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Log (1 + sequence separation)</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Sequence bounds (channel of ones)</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td>60</td>
</tr>
</tbody>
</table>

Table 2-2: Input map features for CNNContactClassify: Residue features are striped horizontally and vertically. This converts them into 2-dimensional feature maps. This means these features occupy 2 channels each in the input tensor.
Model Training

The CNN architecture shown in Figure 2.4 was trained on HD1.0. The network was trained using a multiclass hinge loss function. Training was done on batch sizes of 64 contacts at a time. The network was trained for 400 epochs. The model with the best validation Matthew's correlation coefficient (MCC) saved for testing. The graph in Figure 2.6 shows the training process for the model over 400 training epochs.

![Training Progress - 60 channel input](image)

**Figure 2.6: CNNContactClassify training progress**: Training progress of CNN for classification of inter-protein vs. intra-protein contacts

The main metric used to measure the performance of the model on the training, validation, and test sets was the Matthews correlation coefficient (MCC). MCC values were chosen as opposed to F1 or accuracy scores. This is because although F1 and accuracy metrics are some of the most popularly used metrics in binary classification tasks, they can often show ‘dangerously overoptimistic inflated results’, especially in cases that involve imbalanced datasets (Chicco and Jurman, 2020). The Matthew’s correlation coefficient (MCC) function is a special case of the $\phi$ phi coefficient that measures differences between actual and predicted values in a 2x2 contingency table (Chicco and Jurman, 2020). MCC only produces a high score if good results are obtained for all four categories of a confusion matrix - true positives, true negatives, false positives, and false negatives.
(Chicco and Jurman, 2020). MCC also accounts for the sample size of a dataset being evaluated (Chicco and Jurman, 2020). The formula for MCC score is outlined in equation 2.1 below.

\[
MCC \text{ score} = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp) \times (tp + fn) \times (tn + fp) \times (tn + fn)}}
\]  
\( (Eq. \, 2.1) \)

Once the CNN model for contact prediction was trained, the model was run in evaluation mode on the proteins in the homodimer test set to analyse the accuracy of contact classification on the test set.

To explore how the CNNContactClassify classification of inter-chain contacts affected docking of homodimer complexes, the HADDOCK (Dominguez, Boelens and Bonvin, 2003) docking tool was used to dock a number of the homodimers with higher MCC scores (MCC > 0.75). To do this, the crystal structures of the monomer subunits were fed into HADDOCK along with the predicted lists of inter-chain residues. To dock the homodimers, the monomer subunits were separated and rotated randomly around the x, y, and z axes before being fed into HADDOCK. The TM-score of these generated complexes were then determined using the TM-score tool from the Zhang lab (Zhang and Skolnick, 2004). RMSDs were obtained using the cealign tool in PyMol (Schrödinger, LLC, 2015).

The TM-score of a protein is a measure of the topological similarity of protein structures to a native protein chain (Zhang and Skolnick, 2004). TM-score score is more sensitive to the global fold similarity than RMSD and introduces a length-dependent scale to normalize distance errors, making TM-scores for two chains independent of differences in length between structures being compared (Zhang and Skolnick, 2004). TM-scores with values about 0.5 can usually be assumed to be in the same SCOP/CATH fold (Zhang and Skolnick, 2004).
2.3.1.2 Results: CNNContactClassify Homodimer Contact Classification

Table 2-3 summarizes the results for the training epoch which had the best performance on the validation set. The model weights from the training epoch with the best performance was saved and used for prediction of the test set of homodimers.

<table>
<thead>
<tr>
<th>Epoch 48</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Training loss</td>
<td>0.2971</td>
</tr>
<tr>
<td>Validation loss</td>
<td>0.1825</td>
</tr>
<tr>
<td>Validation precision</td>
<td>0.326</td>
</tr>
<tr>
<td>Validation recall</td>
<td>0.501</td>
</tr>
<tr>
<td>False positive rate (FPR)</td>
<td>0.0512</td>
</tr>
<tr>
<td>Validation MCC</td>
<td>0.342</td>
</tr>
</tbody>
</table>

Table 2-3: Performance metrics of CNNContactClassify from epoch with best validation MCC: Prediction metrics for training epoch number 45, which produced the best performance on the validation set

The validation MCC obtained was 0.342.

**Prediction accuracy of CNNContactClassify for binary classification**

The confusion matrix for contact prediction using the trained model for the HD1.0 test set are shown in Table 2-4. The average MCC for the test set was 0.3739. This corresponded to an overall accuracy of 91.78% and a precision of 0.3145.

<table>
<thead>
<tr>
<th>N = 81,672</th>
<th>Actual: Inter-chain</th>
<th>Actual: Intra-chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted: Inter-chain</td>
<td>2,372</td>
<td>5,169</td>
</tr>
<tr>
<td>Predicted: Intra-chain</td>
<td>1,976</td>
<td>77,316</td>
</tr>
</tbody>
</table>

Table 2-4: Confusion Matrix CNNContactClassify epoch with best validation MCC: Confusion matrix for CNN inter- vs. intra-protein contact classification of test set
Next, the homodimers in the test set were ranked by the MCC values of the inter- and intra-chain contact classification for each homodimer complex. The contact classification MCC of each test set complex is shown in ascending order in Figure 2.7.

![Figure 2.7: Ranked MCC of CNNContactClassify on HD1.0 test set](image)

**Figure 2.7: Ranked MCC of CNNContactClassify on HD1.0 test set:** MCC values of inter- vs intra-chain contact classification of known contacts homodimers in the HD1.0 test set. Red line indicates MCC of 0.3

As can be seen from figure 2.7, some homodimer complexes were predicted far more accurately than others. These homodimers tended to have certain characteristics that made it easier to accurately predict their inter-protein contacts. First, for a number of these homodimers, including 3QFL, 1DK8, and 3AJW (shown in figure 2.8), the contact density of the interacting dimers was quite high, increasing the probability that a residue pair would be involved in an inter-chain interaction. This made it easier for the model to correctly identify interacting residues. Interface contact density is defined as the proportion of residues in the monomer unit of a dimer that are involved in the interaction interface. Previous work (Hopf et al., 2014; Feinauer et al., 2016; Quadir, Roy, Halfmann, et al., 2021) has demonstrated that large interaction interfaces and high contact densities
tend to contribute to higher accuracy of inter-chain contact prediction. Second, examination of the 6 homodimers with the highest MCC values showed that these homodimers tended to be made up mostly of well-structured alpha helix interfaces which may also have contributed to the ability of these models to correctly identify inter-chain contacts in those homodimers.

Out of 100 homodimer complexes in the test set, 55 had an MCC score around or above 0.3

**Homodimer subunits: Exploring correctly versus incorrectly predicted residues involved in inter-protein interfaces**

Figure 2.8 shows 10 structures of protein homodimers with decreasing MCCs, demonstrating the location of correctly (green) vs. incorrectly (red) predicted inter-chain residues, i.e., the true positives and false positive inter-chain contact predictions.

<table>
<thead>
<tr>
<th>a) 3QFL</th>
<th><img src="image" alt="Structure" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>MCC: 0.902</td>
<td>Precision: 0.96</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>b) 1DK8</th>
<th><img src="image" alt="Structure" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>MCC: 0.877</td>
<td>Precision: 0.791</td>
</tr>
<tr>
<td></td>
<td>3AJW</td>
</tr>
<tr>
<td>---</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>MCC: 0.804</td>
</tr>
</tbody>
</table>
**Figure 2.8: PDB structures of homodimers from the homodimer test set:** Structures of 10 homodimers from test set with decreasing MCC scores. Structures for the monomer subunits and homodimer complexes are shown on the left and right respectively. Residues coloured in green show true-positive inter-chain residue predictions, residues in yellow false-negative inter-chain residue predictions, and residues in red show false-positive inter-chain residue predictions.

Overall, low MCC values correspond to low precision values, and usually resulted from several false positive (red residue) predictions and fewer true positive (green) predictions.

Overall, homodimers with higher MCC scores tended to have larger interaction interfaces. In addition, it appears that in the homodimers with good contact classification, false positive inter-protein predictions tended to be spatially close or adjacent to the real inter-protein residues. This was not the case for the complexes with very poor prediction MCCs (i.e., 4BT7).
Finally, it was found that the quality of predictions is correlated to the rate of false positives: the models with poor MCCs and predictions tended to have very high rates of false-positive predictions compared to false negative predictions, which makes sense given the class imbalance in the data.

**HADDOCK Complex Docking: predicted inter-protein residues**

Figure 2.9 shows three examples of complexes from the test set generated using the HADDOCK webserver with the native homodimer complex structure. To perform docking, monomer subunits of the dimers were randomly rotated, then put into the HADDOCK 2 server with the list of the residues involved in the top 10 predicted inter-protein residue contacts as docking restraints. The structures in Figure 2.9 show predicted structures for the top predicted docked model from HADDOCK super-imposed with the native crystal structure. These results are for three homodimers with the highest precision.

In the case of 3AJW, it is apparent that even in cases where quite precise contacts were predicted by DMPContactClassify, HADDOCK got the wrong orientation of non-interacting coils. This could be due to the fact the corresponding protein dimer, FliJ has an antiparallel coiled-coil structure (Ibuki et al., 2011). Coiled-coils have been previously shown to be quite difficult to dock, in some cases needing very accurate homology models and/or template-based modelling for successful docking (de Vries et al., 2010).

1DK8 and 1QJH had quite accurately docked models as the top predicted model from HADDOCK. These docking results were limited to the cases where contact classification precision was greater than 0.75 and demonstrate the importance of precise predictions for inter-chain contacts.
Figure 2.9: Homodimer HADDOCK docking: (a)-(c): HADDOCK docking results for three test homodimer structures with high MCCs structures in blue show correct native orientations of the homodimer complexes, while structures in orange show the best docked structure from HADDOCK.

The results here demonstrate that a CNN deep learning method that uses covariation, secondary structure, solvent accessibility, and sequence profile features may be able to differentiate between inter- and intra-chain contacts to some level, although there is room for improvement. Some routes to improve the differentiation of contacts would be explore the effect of training on bigger, more robust datasets and different deep learning models.
Although the work done in this section was largely a proof of concept as it uses contact maps from the solved homodimer complexes to predict the inter-protein contacts, it was important to first establish that there it is possible to differentiate between inter-chain and intra-chain contacts. These results demonstrated that deep learning could be used to improve the prediction of homodimer complex contacts.

2.3.2 Deep Learning Model 2: DMPContactClassify - Residual Neural Networks for Homodimer Contact Classification

In this section, a different, deeper neural network than CNNContactClassify was evaluated for homodimer contact classification. This deep learning architecture, DMPContactClassify, was adapted from the DeepMetaPSICOV (DMP) method developed in the Jones lab (Kandathil, Greener and Jones, 2019). The DMP model uses a deep, fully convolutional residual neural network (ResNet) and was developed for the prediction of intra-chain single-domain contacts (Kandathil, Greener and Jones, 2019). The DMP method had been shown in previous papers to be quite effective for the task of monomer contact prediction, improving significantly on previous CNN-based models used in the Jones lab (Jones and Kandathil, 2018; Kandathil, Greener and Jones, 2019).

Two versions of the ResNet model, DMPContactClassify 1.0 and DMPContactClassify 2.0 were trained in this section. DMPContactClassify 1.0 had the same input features as the CNNContactClassify model described in section 2.3.1. The input the DMPContactClassify 1.0 was therefore a 60-channel input map of 67 x 67 residue crops around the residue pair being predicted. The second of these models, DMPContactClassify 2.0, had an identical architecture to DMPContactClassify 1.0, but with a different set of inputs. The inputs to DMPContactClassify 2.0 were a 501-channel input made up of the 60 map channels previously described in section 2.3.1 and an additional 441 channels which captured the raw pair covariance values for 20 amino acid types plus gaps for each pair of positions in a sequence alignment.

In this experiment, the DMP architecture was adapted for the task of contact classification of known contacts. To account for the class imbalance, the training process was once again done on an equal number of inter-protein and intra-protein contacts at each epoch (each run over the training data).
2.3.2.1 Training of DMPContactClassify for Homodimer Contact Prediction

Homodimer Dataset 1.0 and 2.0

In this section, the impact of using a bigger and less stringently filtered dataset on the ability of deep learning models to classify homodimer contacts was examined. Here two DMPContactClassify architectures were trained on HD1.0 and HD2. This was done to determine whether using the larger and less stringently filtered HD2 improved the performance of deep learning methods for homodimer complex contact classification.

Some of the homodimer pairs in the original HD2 dataset were removed because DMPContactClassify trains on pairwise map and covariation features, which require the exact same sequence for both homodimer chains. Although most homodimer complexes in HD2 had identical protein sequences, a small fraction had mismatches of one or two residues. Enforcing the requirement that both protein chains had the exact same length and sequence resulted in a list of 6388 homodimers in the training set, 100 homodimers in the validation set, and 100 homodimers in the test set.

DMPContactClassify 1.0 Model Architecture

In the DMPContactClassify 1.0 model, inputs were made up of the 60-channel inputs described in section 2.3.1.1, with the same 67x67 crops around the residue pairs. The ResNet model is very similar to the one previously used in the Jones lab to train the DeepMetaPSICOV (DMP) method and had a hidden dimension of 64. The DMP model is made up of an initial convolutional 2D max-out layer, followed by eighteen residual blocks (Kandathil, Greener and Jones, 2019). The DMPContactClassify model, however, had 14 residual blocks instead of the 18 blocks used in DMP.

In DMP, each residual block was made up of two successive 2D dilated convolutional layers with Rectified Linear Unit (ReLU) activation functions each followed by a 2D instance normalization layer. Each convolutional layer had 5x5 filters, with varying dilation rates and 64 channel outputs (Kandathil, Greener and Jones, 2019). An explanation of convolutional layers and dilated convolutions can be found in Appendix 2 of this thesis.
DMPContactClassify used identical residue blocks but had 32 output feature map sizes rather than 64. Although the sequence of layers in the ResNet blocks was identical, DMPContactClassify had 4 less residual blocks than DMP: DMP had a total of 18 residual layers, while DMPContactClassify has 14 residual blocks. This reduced the overall size of the DMPContactClassify model and allowed for faster training of the model. In addition, the original DMP method had dilation rates of 2, 4, 8, 16, 32, and 64 (Kandathil, Greener and Jones, 2019) while the dilation rates for DMPContactClassify were 2, 4, 8, 16 and 32 (Table 2-5). The dilation rates for each of the residual blocks is outlined in Table 2-5. A dilation of $d=1$ produces regular, non-dilated convolutions.

<table>
<thead>
<tr>
<th>Residual Block</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dilation rate $d$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>32</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2-5: dilation rates for DMPContactClassify: The dilation rate for each residual block in the DMPContactClassify method.

Finally, the output of the 14 residual blocks was fed into a final 2D convolutional layer and instance norm before going into two fully connected linear layers that output a single prediction for the probability of the residue being an inter-protein contact or an intra-protein contact. The architecture for the modified DMP architecture used for DMPContactClassify is shown in Figure 2.10.
**Figure 2.10: DMPContactClassify architecture:** The output of the network is a 1-dimensional probability of a contact being an inter-chain or intra-chain contact. Numbers in parenthesis describe parameters for layers (kernel size for convolutional and max pooling layers, and maxout units for dense layers). Numbers on the right show the dimensions the output of each layer. $d$ corresponds to the dilations used in each of the convolutional layers. DMPContactClassify 1.0 had only input maps as inputs, while DMPContactClassify 2.0 had contact maps and raw pair covariance information as inputs.
DMPContactClassify 2.0 Model Architecture

DMPContactClassify 2.0 had an identical architecture to DMPContactClassify 1.0, with the only difference being the input layer had 501 input channels rather than 60 input channels.

DMPContactClassify Training

Loss Functions

Both models explored here (DMPContactClassify 1.0 and DMPContactClassify 2.0) were trained with two different loss functions. This was done to determine how the choice of loss functions affected the training process of each of the models, and whether one of the loss functions resulted in a better trained model.

The first loss function used to train both DMPContactClassify models was the PyTorch binary cross entropy with logits loss function - BCELoss with logits - which combines a sigmoid layer and BCELoss in a single class. As explained in the PyTorch documentation, this is more numerically stable than using a plain sigmoid followed by a BCELoss (‘PyTorch Docs - BCELoss with logits’, no date).

The second loss function used to train the DMPContactClassify models was the MCC loss function. The MCC loss function is a modification of the MCC score. The formula for MCC loss function is shown in equation 2.2.

\[
MCC\ loss = 0.5 \times \left(1 - \frac{tp \times tn - fp \times fn}{1 + \sqrt{(tp + fp) \times (tp + fn) \times (tn + fp) \times (tn + fn)}}\right) \quad (Eq.\ 2.2)
\]

Model Training

The DMPContactClassify models were trained with an Adam optimizer (Kingma and Ba, 2014) which had a more regular training curve for this prediction task than stochastic gradient descent (SGD) (Amari, 1993; Ruder, 2016). The models were trained for multiple epochs, until the models converged. Models were saved at each epoch if the training loss
was less than previous epochs, or if the validation precision or MCC values were greater than previous epochs.

When training epochs were run over the entire HD2.0, model training took a significant amount of time: between ~13-18 hours per epoch for DMPContactClassify 1.0 with BCE with logits loss and ~90hrs per epoch for DMPContactClassify 2.0 with mcc loss. This is because the MCC loss took longer to calculate per training batch than the BCE logits loss. Due to the very long training times for the whole training set, each epoch in the training process was made up of 500 homodimer examples. This meant that the training set was covered over approximately 13 epochs.

2.3.2.2 Results: DMPContactClassify Prediction

Dataset Comparison: HD1.0 vs. HD2.0

To determine whether training DMPContactClassify on the larger but less stringently filtered HD2.0 improved the performance of the deep learning models over HD1.0, both DMPContactClassify 1.0 and DMPContactClassify 2.0 were trained using BCE loss with logits on both datasets. The training progress of the DMPContactClassify models using the two different datasets is shown in Figure 2.11.
Figure 2.11: DMPContactClassify training: Performance of each deep learning method over training epochs of the DMPContactClassify models. Each training epoch involved training on 500 homodimers. The blue line corresponds to the training loss evaluated after each epoch, the purple line is the MCC on the training set (training MCC), red is the validation loss after each epoch, and green is MCC on the validation set (validation MCC).

a. Training progress of DMPContactClassify 1.0 on HD1.0.
b. Training progress of DMPContactClassify 1.0 on HD2.0
c. Training progress of DMPContactClassify 2.0 on HD1.0
d. Training progress of DMPContactClassify 2.0 on HD2.0

Figure 2.11 shows the training metrics of each of the DMPContactClassify models trained on the two Homodimer Datasets (HD1.0 and HD2.0). The figures show the performance of each of the models over training epochs and demonstrates the decrease in training and validation losses (blue and red line respectively) and the corresponding increases in training and validation MCC (purple and green line respectively). As can be seen from
figure 2.11, while the models were able to achieve relatively high training MCCs over training epochs, the increases in validation MCCs were relatively modest, which may explain the modest performance of some of these models on the test sets as shown in table 2.8.

The results for the epoch with the highest validation MCC score are reported in Table 2-6.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Epoch</th>
<th>Training Loss</th>
<th>Validation precision</th>
<th>Validation MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMPContactClassify 1.0</td>
<td>HD1.0</td>
<td>209</td>
<td>0.4046</td>
<td>0.2042</td>
<td>0.2291</td>
</tr>
<tr>
<td>DMPContactClassify 1.0</td>
<td>HD2.0</td>
<td>335</td>
<td>0.2146</td>
<td>0.2216</td>
<td>0.2543</td>
</tr>
<tr>
<td>DMPContactClassify 2.0</td>
<td>HD1.0</td>
<td>431</td>
<td>0.1931</td>
<td>0.2595</td>
<td>0.2719</td>
</tr>
<tr>
<td>DMPContactClassify 2.0</td>
<td>HD2.0</td>
<td>368</td>
<td>0.1815</td>
<td>0.2742</td>
<td>0.3016</td>
</tr>
</tbody>
</table>

Table 2-6: DMPContactClassify Best Epoch Performance Metrics: Metrics for the best training epochs for DMPContactClassify 1.0 and 2.0 trained on HD1.0 versus HD2.0

A paired sample t-test with a confidence interval of 0.95 was used to compare the validation precision of DMPContactClassify methods trained on HD1.0 compared to HD2.0. The p-value for the paired t-test was 0.03942, indicating the improvement in validation precision when using HD2.0 instead of HD1.0. was statistically significant. For this reason, the rest of the models tested in this chapter were trained on HD2.0.

**DMPContactClassify Training: BCE loss vs. MCC loss**

Once it was established that the DMPContactClassify models performed better when trained on the HD2.0, the DMPContactClassify models were trained on HD2.0 with two different loss functions. The training progress of each of the methods with each of the loss functions is shown in Figure 2.12.
**Figure 2.12: DMPContactClassify training using different losses:** Performance of each deep learning method over training epochs of the DMPContactClassify models trained using two different loss functions (BCE logits loss and MCC loss). The blue line corresponds to the training loss after each epoch, the purple line is MCC on the training set (training MCC), the red line is the validation loss, and green line is the MCC on the validation set (validation MCC). Y-axis of the models corresponds to the loss type that was used to train each model, while the performance of each model is always measured in the MCC value on the training and validation sets.

**a.** Training progress of DMPContactClassify 1.0 with BCE logits loss  
**b.** Training progress of DMPContactClassify 1.0 with MCC loss  
**c.** Training progress of DMPContactClassify 2.0 with BCE logits loss  
**d.** Training progress of DMPContactClassify 2.0 with MCC loss
The results for the epoch with the highest validation MCC score are reported in Table 2-6.

<table>
<thead>
<tr>
<th>Model</th>
<th>Loss Function</th>
<th>Epoch</th>
<th>Training Loss</th>
<th>Validation precision</th>
<th>Validation MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMPContactClassify 1.0</td>
<td>BCE loss</td>
<td>335</td>
<td>0.2146</td>
<td>0.2216</td>
<td>0.2543</td>
</tr>
<tr>
<td>DMPContactClassify 1.0</td>
<td>MCC loss</td>
<td>611</td>
<td>0.2749</td>
<td>0.2307</td>
<td>0.2734</td>
</tr>
<tr>
<td>DMPContactClassify 2.0</td>
<td>BCE loss</td>
<td>368</td>
<td>0.1815</td>
<td>0.2742</td>
<td>0.3016</td>
</tr>
<tr>
<td>DMPContactClassify 2.0</td>
<td>MCC loss</td>
<td>517</td>
<td>0.2601</td>
<td>0.2942</td>
<td>0.3220</td>
</tr>
</tbody>
</table>

Table 2-7 DMPContactClassify Loss Function Comparison: Metrics for the best training epochs for DMPContactClassify 1.0 and 2.0 trained using BCE logit loss and MCC loss

Once training of each of the DMPContactClassify models was completed, each model was run on the homodimer test set, and the results of contact classification results for each model with each loss function are reported in Table 2-8.

<table>
<thead>
<tr>
<th>Model</th>
<th>Loss Function</th>
<th>Test Precision</th>
<th>Test Recall</th>
<th>Test FPR</th>
<th>Test MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMPContactClassify 1.0</td>
<td>BCE loss</td>
<td>0.1920</td>
<td>0.3883</td>
<td>0.1024</td>
<td>0.2218</td>
</tr>
<tr>
<td>DMPContactClassify 1.0</td>
<td>MCC loss</td>
<td>0.2254</td>
<td>0.2864</td>
<td>0.0626</td>
<td>0.2323</td>
</tr>
<tr>
<td>DMPContactClassify 2.0</td>
<td>BCE loss</td>
<td>0.2588</td>
<td>0.2662</td>
<td>0.0481</td>
<td>0.2654</td>
</tr>
<tr>
<td>DMPContactClassify 2.0</td>
<td>MCC loss</td>
<td>0.2766</td>
<td>0.2756</td>
<td>0.045</td>
<td>0.2815</td>
</tr>
</tbody>
</table>

Table 2-8: DMPContactClassify Test Set Performance: Performance of saved model from best training epoch for the two versions of DMPContactClassify and the two loss functions tested. Metrics reported include the precision, recall, test false positive rate (FPR) and test MCC
A binomial test of proportions was performed to compare the performance of DMPContactClassify using the two different loss functions (BCE loss and MCC loss). Another set of binomial tests of proportions were performed to compare the performance of DMPContactClassify1.0 and DMPContactClassify 2.0. Table 2.9 below shows the p-values of the binomial test of proportions comparing MCC vs BCE loss results and DMPContactClassify 1.0 vs 2.0.

<table>
<thead>
<tr>
<th>Binomial test P-value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DMPContactClassify 1.0: MCC loss vs. BCE loss</td>
<td>0.1551</td>
</tr>
<tr>
<td>DMPContactClassify 2.0: MCC loss vs. BCE loss</td>
<td>0.2799</td>
</tr>
<tr>
<td>BCE loss:</td>
<td></td>
</tr>
<tr>
<td>DMPContactClassify1.0 vs. DMPContactClassify2.0</td>
<td>0.0253</td>
</tr>
<tr>
<td>MCC loss:</td>
<td></td>
</tr>
<tr>
<td>DMPContactClassify1.0 vs. DMPContactClassify2.0</td>
<td>0.06080</td>
</tr>
</tbody>
</table>

**Table 2-9: Binomial test p-values: comparison of performance of MCC loss function vs. BCE loss function on HD2.0 test set.**

The results in Error! Reference source not found. demonstrated no statistical difference in the performance of DMPContactClassify models trained using BCE loss and MCC loss. The two loss functions therefore did not yield statistically different performances in terms of inter-chain contact precision. The table also demonstrates that when using the MCC loss function, there is a statistically significant difference in the performance of DMPContactClassify1.0 and DMPContactClassify2.0, while this is not the case when using the BCE loss function.

It is difficult to say based on the values alone whether the best DMPContactClassify method (precision: 0.2601, MCC: 0.3220) was better than the best CNNContactClassify method (precision: 0.3145, MCC: 0.339). Although the precision and MCC for CNNContactClassify seem a bit higher, it is difficult to tell whether this difference is statistically significant, as only one CNNContactClassify model was trained in this chapter.
2.4 Conclusions: Deep Learning for Homodimer Contact Classification

In chapter 2, two model architectures were explored for the problem of classifying homodimer contacts as either inter-chain or intra-chain contacts. The first model used a CNN model, called CNNContactClassify, trained on a high-quality non-redundant set of homodimers. The second model used a deep residual neural network (ResNet) model, called DMPContactClassify, trained on two different homodimer datasets.

At the time this experimental work was done, no methods existed that had been demonstrated to effectively differentiate between inter-chain and intra-chain contacts. The work in this section was one of the first studies we were aware of that tested the application of deep learning methods to large and comprehensive datasets of homodimer proteins. The results demonstrate that deep learning methods that use covariation, secondary structure, solvent accessibility, and sequence profile feature inputs may be able to differentiate between inter- and intra-chain contacts.

It is apparent from all the experiments done in this section that the DMPContactClassify 2.0 method (which had 501 input features, including all the map features and 441 features representing the raw covariances of residue pairs) performed better than DMPContactClassify 1.0 (which had only the 60 map input features) regardless of dataset and loss function used. This implies that the raw covariances add useful information for the classification of homodimer contacts as either inter-chain or intra-chain contacts.

Comparison of training results of DMPContactClassify on HD1.0 and HD2.0 seem to indicate that increasing the size and reducing the redundancy stringency increased the overall performance of the models trained on the larger dataset. Previous work from Jones et al. (Jones and Kandathil, 2018; Kandathil, Greener and Jones, 2019) used very stringent redundancy filters to obtain the training data used to train monomer contact prediction models. It is possible, however, that homodimer inter-chain contact prediction may benefit from having slightly more redundant but larger datasets because fewer non-redundant homodimer structures are available on PDB than non-redundant monomer structures. In addition, it is possible to have different homodimer interfaces even when
monomers may have the same folds or topology, if residues at the interfaces are divergent enough to result in different inter-chain contact patterns (i.e. different interfaces). Therefore, even though the same folds may have been represented more than once in the training set, it is possible the corresponding homodimers may have had slightly different interfaces. Finally, it is also possible that due to the large class imbalance between inter-chain and intra-chain residue pairs, the models simply benefitted from having more examples of inter-chain residue pairs to train on.

Two different loss functions were used to train the models in this section: the binary cross entropy with logits loss (BCE loss with logits) and a Matthews correlation coefficient loss (MCC loss) which was calculated using slight modifications of the MCC function. The results of the binomial test of proportions seem to indicate that there was no significant difference in the performance of models trained with each of these loss functions. Models trained using the MCC loss function, however, tended to converge faster than those trained with BCE logit loss. For this reason, deep learning models trained in chapter 3 were trained using MCC loss.

Finally, although the precision and MCC values of CNNContactClassify on the HD1.0 test set were slightly higher than the best DMPContactClassify model evaluated on HD2.0 test set, it was not possible to determine with statistical certainty whether one model was better than the other. Future work could train CNNContactClassify on HD2.0, with BCE and MCC loss to determine whether the difference between the CNN and DMP models is statistically significant on the task of contact classification.

The work done in this section was largely a proof of concept as it used sets of known contacts from crystal structures of homodimer complexes to perform contact classification. These contacts are not usually known beforehand when performing de novo homodimer complex prediction. It was important, however, to establish that it was indeed possible to differentiate between the two contact types before extending such methods to the prediction of homodimer contacts. These results demonstrated that deep learning could be used to improve the deconvolution of interface and monomer residue contacts.
3. Deep Learning for De Novo Homodimer Contact Prediction

3.1 Introduction: Homodimer Complex Contact Prediction

The problem of simultaneous prediction of inter- and intra-chain protein contacts was explored in this chapter. The ultimate goal of the work described in this chapter, was to explore the ability of deep-learning tools developed in the Jones lab to do de-novo prediction of homodimer contacts where no close structural homologs of the monomers or the complex exist in the Protein Data Bank (PDB). The work in this chapter explores deep learning methods for contact prediction of intra-chain and inter-chain residue pairs.

Work in chapter 2 explored the classification of known homodimer contacts as inter- or intra-chain residue contacts. This was an important proof of concept. In practice, however, fully de novo prediction of homodimer complexes requires being able to predict both intra-chain and inter-chain residue contacts without any prior knowledge and with an acceptable level of accuracy for further folding and/or docking of protein chains into complexes.

For this reason, the work outlined in this chapter focused on exploring modifications to the CNN and ResNet models used in chapter 2 for de novo prediction of both inter-chain and intra-chain contacts. Both the CNN-based method (CNNContactClassify) and the DMP-based methods (DMPContactClassify) were adapted to output 2 channel outputs, with the first channel containing the likelihood of a residue pair being an inter-chain contact and the second channel containing the likelihood of a residue pair being an intra-chain contact.

Based on the results in chapter 2, the DMP model with 501 channel inputs was tested here for multi-class contact prediction. All models were trained on HD2.0 which had
previously shown to result in better trained models. In addition, the MCC loss function was used because it tended to converge faster than BCE loss.

A few training strategies were explored to account for large imbalances between the three classes of residue pairs – i.e., inter-chain contacts, intra-chain contacts and non-contacts - which had ratios of 1:24:318 in the training set. Each of these residue pair types will hereafter be referred to as class I (inter-chain contacts), class II (intra-chain contacts) and class III (non-contact residue pairs). In this work, non-contact residue pairs are defined as all residue pairs with intra-chain Cβ distance (Cα for glycine) > 8Å and inter-chain Cβ distance > 10Å.

In all methods tested, the MCC loss function was used in the training process. In the case of the multi-class prediction tasks, MCC loss was calculated for both the class I and class II residue pairs. A few different weightings of the two MCC losses were tested for training the contact prediction methods.

### 3.2. Convolutional neural networks for multi-class prediction of homodimer contacts

The first deep learning model tested for the multi-class prediction of homodimer complex contacts was the CNNContactPredict method. This method used an architecture almost identical to CNNContactClassify, with the only difference being the output layer of the network, which had a two-channel prediction rather than one. The first channel contained predictions of whether the residue pair was involved in a class I (inter-chain) contact or not while the second channel contained predictions between 0 and 1 for whether the residue pair was involved in a class II (intra-chain) contact. The target prediction for each channel was 1 if the contact was involved in a contact of the particular class and a 0 if not.
3.2.1 Methodology

3.2.1.1 Model inputs and model architecture

As results in chapter 2 had shown better training results for the 501-channel input channels made up of the 60-channel map inputs and 441-channel raw covariance information, these were the inputs used to train the CNNContactPredict method.

The CNNContactPredict method was trained and tested on crops around the residue pairs of interest like the CNNContactClassify method described in section 2.3.1. The CNN models used to train CNNContactPredict were identical to the CNNContactClassify method. The sole difference between CNNContactPredict and CNNContactClassify was that the former had a 2-channel output. The first output dimension corresponded to the probability of a residue pair being a class I (inter-chain contact) pair and the second dimension contained the probability a residue pair being a class II (intra-chain contact) pair. The aim of CNNContactPredict was to perform simultaneous prediction of inter-chain and intra-chain contacts for homodimer complexes.

Figure 3.1 shows the difference between CNNContactPredict and CNNContactClassify, which is mainly the dimension of the output.
Figure 3.1: **CNNContactPredict Architecture**: Architecture of CNNContactPredict, which uses the same CNN architecture as CNNContactClassify. Differences between the two models are shown in purple text.

### 3.2.1.2 Model training

#### Training Strategies

Two training strategies were used to account for class imbalances between class I, class II, and class III residue pairs. Both strategies involved training and prediction on one residue pair at a time, with 67x67 crops around the residue pair of interest. As was done in the training of DMPContactClassify, crops of the protein contact maps were created by taking crops of 33 residues around the residue contact pair being classified (i.e. 33 residues on either side of the contact pair of interest in the 2D input map). This resulted in crops of 67x67, including the residue pair being classified. Since there were 60 channels in the input maps, this generated inputs with dimensions of 60x67x67. The model in Figure 3.1 was then trained on a balanced set made up of an equal number of crops around the inter-protein contacts and crops around the intra-protein contacts.

The first of strategy involved training on an equal number of positives and negative contact cases during each training epoch. Here, the positive set was made up of class I and class II residue pairs while the negative set was made up of non-contacts. CNNContactPredict was therefore trained on an equal number of positive contacts (either
inter- or intra-chain) and negative contacts (non-contacts) at each epoch. The second strategy involved training on an equal number of class I, class II, and class III contacts at each epoch.

The training strategies used to train the CNNContactPredict method are summarised in Table 3-1.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Details</th>
</tr>
</thead>
</table>
| 1        | - Single contact prediction  
           - Each epoch: trained on equal number of positive (class I + class II) examples and negative (class III) examples |
| 2        | - Single contact prediction  
           - Each epoch: trained on equal number of class I, class II and class III examples |

Table 3-1: CNNContactClassify Training Schemes: Training strategies used for the training and testing of the CNNContactClassify method

Loss Functions

The MCC loss function was used to train the CNNContactPredict models. MCC losses were calculated for both the inter-contact and intra-contact predictions. Different weightings of the loss functions were tested for the training process.

Weighted inter-chain MCC loss functions were calculated using equation 3.1

\[
MCC\ loss\ weighted\ inter = w_{\text{inter}} \ast MCCloss_{\text{inter}} + w_{\text{intra}} \ast MCCloss_{\text{intra}} \quad (Eq\ 2.4)
\]

Where \(w_{\text{inter}}\) was the weighting for inter-chain contact (class I) losses and \(w_{\text{intra}}\) was the weighting for intra-chain (class II) contact losses. \(MCCloss_{\text{inter}}\) and \(MCCloss_{\text{intra}}\) were the MCC losses for the inter-chain contacts and intra-chain contacts respectively.
In addition, the sum of the weightings had to be equal to one:

\[ w_{\text{inter}} + w_{\text{intra}} = 1 \quad (Eq \ 2.5) \]

The MCC loss weighted function could therefore be calculated as:

\[ MCC \ \text{loss weighted} = w_{\text{inter}} \times MCC_{\text{loss}_{\text{inter}}} + (1 - w_{\text{inter}}) \times MCC_{\text{loss}_{\text{intra}}} \quad (Eq \ 2.6) \]

**Model Training Schemes**

Four different weighting schemes were tested for both class imbalance training strategies. Given the two training strategies and 4 weighting schemes, 8 versions of the CNNContactPredict model were trained using the schemes shown in Table 3-2

<table>
<thead>
<tr>
<th>Model Training Scheme</th>
<th>Training strategy</th>
<th>Model weighting</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.95, 0.05</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.85, 0.15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.65, 0.35</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.5, 0.5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0.95, 0.05</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0.85, 0.15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0.65, 0.35</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0.5, 0.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 3-2: CNNContactPredict Model Training Schemes: Outline of training schemes used for the training and testing of CNNContactPredict method. Two input feature lists, two training schemes, and four model weighting schemes were trained in different combinations resulting in 16 training schemes in total.
Network training

The CNNContactPredict models were trained on the HD2.0 as outlined in section 2.2.2. The training set was made up of 7,317 homodimer examples. The validation and test sets were made up of 100 homodimers each.

For each training strategy, the network was trained until the network converged. For each epoch, if the model had a lower training loss, higher validation precision, or higher validation MCC value than previous epochs, the model was saved at that epoch. The Adam optimizer (Kingma and Ba, 2014) was used for training. Training was done on batches of 60 crops per batch. Similar to model training methods used in chapter 2, each training epoch involved training models on contacts from 500 randomly selected homodimers. As the training set contained 7,317 training examples, this meant that it took ~30 epochs to cover the training set. The 8 models outlined in table 2.12 were trained and evaluated for simultaneous prediction of class I and class II contacts.

Evaluation of CNNContactPredict on Test Set

Once the eight CNNContactPredict training and weighting schemes had been trained on the HD2.0 training set, the models were then run in evaluation mode over the test set from the same dataset. The precision, recall, and MCC of the contacts predicted for both inter-chain (class I) and intra-chain (class II) contact types were extracted and are reported in table 2.14 of the results section below.

3.2.1.3 Results

Once the CNNContactPredict deep learning architecture was trained using the 8 training schemes outlined above, the networks that had the best validation MCC score over the training epochs was saved for testing on the test set. The training, validation, and test results for the epoch with the lowest validation loss for each training scheme are outlined in table 3.3.
<table>
<thead>
<tr>
<th>Training scheme</th>
<th>Training strategy</th>
<th>Model weighting ((W_{\text{inner}}, W_{\text{outer}}))</th>
<th>Training loss</th>
<th>Validation loss</th>
<th>Validation class I (\text{MCC})</th>
<th>Validation class II (\text{MCC})</th>
<th>Test class I (\text{MCC})</th>
<th>Test class II (\text{MCC})</th>
<th>Test class I precision</th>
<th>Test class II precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.95, 0.05</td>
<td>0.2808</td>
<td>0.2938</td>
<td>0.0011</td>
<td>0.5562</td>
<td>-0.0002</td>
<td>0.5460</td>
<td>0</td>
<td>0.5949</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.85, 0.15</td>
<td>0.2708</td>
<td>0.2771</td>
<td>-0.0003</td>
<td>0.5673</td>
<td>0.0009</td>
<td>0.5222</td>
<td>0.0005</td>
<td>0.5783</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.65, 0.35</td>
<td>0.2805</td>
<td>0.2951</td>
<td>-0.0010</td>
<td>0.5801</td>
<td>-0.0004</td>
<td>0.5711</td>
<td>0</td>
<td>0.6743</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.5, 0.5</td>
<td>0.2817</td>
<td>0.2925</td>
<td>-0.0010</td>
<td>0.5795</td>
<td>-0.0013</td>
<td>0.4754</td>
<td>0</td>
<td>0.5019</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0.95, 0.05</td>
<td>0.3478</td>
<td>0.3801</td>
<td>0.0091</td>
<td>0.4308</td>
<td>0.0095</td>
<td>0.5711</td>
<td>0.0053</td>
<td>0.6710</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0.85, 0.15</td>
<td>0.3658</td>
<td>0.3672</td>
<td>0.0138</td>
<td>0.5086</td>
<td>0.0131</td>
<td>0.5104</td>
<td>0.0062</td>
<td>0.5781</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0.65, 0.35</td>
<td>0.3559</td>
<td>0.3710</td>
<td>0.0275</td>
<td>0.5802</td>
<td>0.0220</td>
<td>0.5985</td>
<td>0.0064</td>
<td>0.8895</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0.5, 0.5</td>
<td>0.3878</td>
<td>0.3774</td>
<td><strong>0.0343</strong></td>
<td><strong>0.5828</strong></td>
<td><strong>0.0318</strong></td>
<td>0.5728</td>
<td><strong>0.0077</strong></td>
<td>0.7636</td>
</tr>
</tbody>
</table>

Table 3.3: CNNContactPredict Training Scheme Performance: Training, test, and validation metrics for each trained CNNContactPredict model. Numbers in bold indicate the best performing model for a particular metric.
The overall accuracy, precision, and MCC for inter-chain contacts on the test sets were observed to be quite low. Most existing methods, however, do not usually measure overall accuracy or precision of inter-chain contact prediction, but instead measure the accuracy of the top N (usually less than 15) predicted contacts (Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014; dos Santos et al., 2015; Wang and Barth, 2015; Uguzzoni et al., 2017). It had been previously shown that in many cases, 3-5 reliable inter-chain contacts could be enough for reasonable docking of protein dimers using existing docking tools (Lensink and Wodak, 2010; Lensink et al., 2016). For these reasons, instead of looking only at the overall accuracy of the predicted contacts, the precision of the top 5, 10 and 15 predicted inter-chain contacts (i.e., the residue pairs predicted with the highest probability of being class I inter-chain contacts) was extracted. Accuracy of the top N inter-chain contacts is shown in Table 3-4.

<table>
<thead>
<tr>
<th>Model Training Details</th>
<th>Precision of top N ranked inter-chain contacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training scheme</td>
<td>Training strategy</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 3-4: CNNContactPredict Ranked Contact Performance:** Performance of models trained using different training schemes in terms of precision of top 5, top 10, and top 15 ranked inter-chain contacts.
It appeared from these results the models trained using an equal number of classes I, II and III residue pairs at each epoch (i.e., training schemes 4-8) had better performance than models trained on an equal number of positive contact (class I and class II) and non-contact (class III) residue pairs at each epoch (i.e., training schemes 1-4).

The results of binomial tests of proportions for training strategy 1 vs. training strategy 2 for the top 5, 10, and 15 ranked contacts is shown in Table 3.5.

<table>
<thead>
<tr>
<th>Training strategy 1 vs. Training strategy 2</th>
<th>Class I - Top 5 precision (p-value)</th>
<th>Class I – Top 10 precision (p-value)</th>
<th>Class I - Top 15 precision (p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model weighting (0.95, 0.05)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Model weighting (0.85, 0.15)</td>
<td>8.07E-05</td>
<td>7.16E-06</td>
<td>2.03E-08</td>
</tr>
<tr>
<td>Model weighting (0.65, 0.35)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Model weighting (0.5, 0.5)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table 3-5: Statistical significance of training strategy improvement: Binomial test of proportion p-values for differences in top N precision for class I contacts predicted using models trained using training strategy 1 vs. strategy 2.

Results in Table 3-5 show that the improvements of models trained with training strategy 2 over strategy 1 could only be shown for the second model weighting scheme. Since the other three weighting schemes had 0 contacts successfully predicted for training strategy 1, the p-value for the binomial tests of proportions for these comparisons was always 0. This means that training models on a balanced set of inter-chain contacts, intra-chain contacts, and non-contacts led to better prediction outcomes.

Another question of interest was whether better intra-chain MCCs corresponded to better inter-chain MCCs for trained models. To explore this, class I and class II training and validation MCCs were plotted to see if any correlations could be observed (Figure 3.2).
Figure 3.2: Inter-chain vs Intra-chain MCC: a. graph of inter-chain training MCCs vs intra-chain training MCCs b. graph of intra-chain validation MCCs vs. intra-chain validation MCCs

As can be seen from the graphs in Figure 3.2, no correlation could be observed between the inter-chain MCCs and intra-chain MCCs for the best saved models. This seems to indicate that the performance of the saved models of the two types of contacts were independent of each other in the CNNContactPredict models.

Overall, using more balanced training during training epochs and bigger input feature sets improved the performance of CNNContactPredict for the simultaneous prediction of inter-chain and intra-chain contacts, but inter-chain contact precisions were quite low and likely would not be useful for homodimer docking.
3.3. Residual neural networks for multi-class prediction of homodimer contacts

Experiments with the CNNContactPredict method did not result in very accurate results for inter-chain contact prediction.

The second deep learning model tested for the multi-class prediction of homodimer complex contacts was the DMPContactPredict. This method used an architecture identical to the DMPContactClassify method outlined in chapter 2. Again, the main difference of DMPContactPredict from the DMPContactClassify method was that the output layer of the deep residual neural network had a two-channel output prediction identical to the output channels described for CNNContactClassify (section 3.2.1.1).

3.2.1 Methodology

3.2.1.1 Model inputs

Work done in chapter 2 showed that using the 501-channel inputs consisting of map inputs in addition to raw covariances from DeepCov (Jones and Kandathil, 2018; Kandathil, Greener and Jones, 2019) resulted in better performance of trained models than using the 60-channel map inputs alone.

3.2.1.2 Model architecture

The training strategy used for the DMPContactPrediction method was identical to the second training strategy used to CNNContactPredict method. This training strategy was trained and tested on crops of 67x67 residues around the residue pairs of interest. Training was done on an equal number of class I, II and III residue pairs at each epoch.

DMPContactPredict used a ResNet architecture identical to the architecture used to train DMPContactClassify, with the only difference being the output layer which had 2
channels. Each output channel contains the probability of a contact being a class I or class II contact, respectively. Changes to the DMPContactClassify architecture made for DMPContactPredict are shown in Figure 3.3. DMPContactPredict is hereafter referred to as DMP_CP

![Diagram of DMPContactPredict Architecture](image)

**Figure 3.3: DMPContactPredict Architecture (DMP_CP):** Architecture of DMPContactPredict, which uses the same architecture as DMPContactClassify. Differences between the two models are shown in purple.

**Model Training**

**Loss Function**

The MCC loss function described in section 3.2 was used to train the DMP_CP methods. The same loss weighting strategies used to train models in section 3.2 were used to train DMP_CP models.

**Model Training Schemes**

Training strategy 2 (table 3.1) which balanced class I, class II and class III residue pairs for each training run was used to train DMPContactClassify. Four training schemes were explored in this section, using different weightings of the inter-chain and intra-chain MCC
All these methods were trained using a DMPContactPredict architecture with 501 input features made up of 60 map inputs and 441 channels containing raw covariances of residue pairs including gaps.

Since one input type, one training strategy, and four weighting methods were used to train various versions of the DMP_CP method, 4 different training schemes were used for training. The training schemes used to train and test the DMP_CP method are given in Table 3-6. All the models trained using the 501 channel inputs.

<table>
<thead>
<tr>
<th>Model Training Scheme</th>
<th>Training strategy</th>
<th>Model weighting ($w_{inter}, w_{intra}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.95, 0.05</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.85, 0.15</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.65, 0.35</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.5, 0.5</td>
</tr>
</tbody>
</table>

Table 3-6: DMP_CP Training Schemes: Training schemes used for training different DMP_CP models for multiclass contact prediction

Network training

DMP_CP was trained on HD2.0 using the four training schemes. The models were trained until the MCC loss converged. The Adam optimizer (Kingma and Ba, 2014) was used for training. Training was done on batches of 60 crops. Each training epoch consisted of training on examples from 500 homodimer complexes at a time.

Evaluation of DMP_CP on Test Set

Once the four DMP_CP models had been trained, the models were then run in evaluation mode over the homodimer chains in the test set. The precision, recall, and MCC of the contacts predicted for both inter-chain (class I) and intra-chain (class II) contact types were extracted. The precision of the top 5, 10 and 15 predicted inter-chain contacts (i.e.,
the residue pairs predicted with the highest probability of being class I inter-chain contacts) were calculated and are shown below.

3.2.1 Results

During training DMP_CP with the four training schemes outlined in table 3.6, the networks that had the best validation MCC score over the training epochs were saved for testing on the test set. The results of evaluation of the saved models on the training, validation, and test sets are outlined in Error! Reference source not found.. Error! Reference source not found. includes results for the class I precision of the top 5, 10, and 15 ranked inter-chain contacts for the DMP_CP models.
<table>
<thead>
<tr>
<th>Training strategy</th>
<th>Model weighting (W_{\text{inter}} ) ( W_{\text{inter}} )</th>
<th>Training loss</th>
<th>Validation loss</th>
<th>Validation class I MCC</th>
<th>Validation class II MCC</th>
<th>Test class I MCC</th>
<th>Test class II MCC</th>
<th>Test class I precision</th>
<th>Test class II precision</th>
<th>Top 5 ranked contact precision</th>
<th>Top 10 ranked contact precision</th>
<th>Top 15 ranked contact precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.95, 0.05</td>
<td>0.4503</td>
<td>0.5149</td>
<td>0.0433</td>
<td>0.4934</td>
<td>0.0359</td>
<td>0.4559</td>
<td>0.0100</td>
<td>0.4184</td>
<td>0.0240</td>
<td>0.0280</td>
<td>0.0207</td>
</tr>
<tr>
<td>2</td>
<td>0.85, 0.15</td>
<td>0.4212</td>
<td>0.5082</td>
<td>0.0401</td>
<td>0.5114</td>
<td>0.0331</td>
<td>0.4816</td>
<td>0.0086</td>
<td>0.4708</td>
<td>0.0180</td>
<td>0.0250</td>
<td>0.0167</td>
</tr>
<tr>
<td>2</td>
<td>0.65, 0.35</td>
<td>0.4061</td>
<td>0.3942</td>
<td>0.0427</td>
<td>0.4103</td>
<td>0.0397</td>
<td>0.4764</td>
<td>0.0102</td>
<td>0.4620</td>
<td>0.0220</td>
<td>0.0280</td>
<td>0.0207</td>
</tr>
<tr>
<td>2</td>
<td>0.5, 0.5</td>
<td>0.3720</td>
<td>0.3172</td>
<td><strong>0.0478</strong></td>
<td><strong>0.5741</strong></td>
<td><strong>0.0433</strong></td>
<td><strong>0.5171</strong></td>
<td><strong>0.0109</strong></td>
<td><strong>0.5323</strong></td>
<td><strong>0.0280</strong></td>
<td><strong>0.0310</strong></td>
<td><strong>0.0246</strong></td>
</tr>
</tbody>
</table>

**Table 3.7: DMPContactPredict Training Scheme Performance**: Training, test, and validation metrics for each trained DMPContactPredict model. Numbers in bold indicate the best performing model for a particular metric. Ranked contact predictions for top 5, 10, and 15 contacts are included here.
To determine whether DMP_CP was statistically better than CNNContactPredict in terms of precision of top 5, 10, and 15 predicted inter-chain contacts, a paired sample t-test with confidence interval of 0.95 was used to compare all DMP_CP models trained using training strategy 2 to all CNNContactPredict models trained using the same strategy. Results for this paired t-test are shown in Figure 3.4.

![Figure 3.4: DMP_CP vs. CNNContactPredict](image)

Figure 3.4: DMP_CP vs. CNNContactPredict: Box plots and t-test p-values for inter-chain contact precision of a) top 5 b) top 10 and c) top 15 predicted inter-chain contacts.

Results in Figure 3.4 show that while DMP_CP precision for the top 5 and top 10 predicted inter-chain contact were significantly better than those from CNNContactPredict, the difference was not statistically significant for the top 15 predicted inter-chain contacts. The increased precision of DMP_CP over CNNContactPredict makes sense as the ResNet model used was much deeper than the CNN model. In addition, residual neural networks have previously been demonstrated to have better performance for monomer contact prediction than CNNs (Wang et al., 2016; Kandathil, Greener and Jones, 2019).

The best performing DMP_CP model was evaluated on the HD2.0 test set. Figure 3.5 shows the number of homodimers in the test set which had above a certain number of true positive contacts in the top 5, 10, and 15 ranked inter-chain contacts.
Figure 3.5: Homodimer Complexes with Correct Class I Contacts: Number of homodimers in test set with at least 1, 2, 3, or 4 correct class I contacts in top 5, 10, 15 ranked lists for best trained DMP_CP method

It has been demonstrated in previous docking studies (Lensink and Wodak, 2010; Andreani, Faure and Guerois, 2013; Lensink et al., 2016; Yu et al., 2016) that having 2-5 accurate positive contacts with a low false-positive rate can contribute to improved docking of protein-protein dimer complexes.

The results in Figure 3.5 show that for top 5, and top 10 ranked inter-chain contacts for the best DMP_CP method, 4 and 7 homodimer complexes, respectively, had more than 2 correctly predicted contacts. For the top 15 predicted contacts, 9 complexes had more than 2 correctly predicted contacts. Unfortunately, this means only 4% of the 100 homodimers in the test sets have at least 3 correctly predicted inter-chain contacts in the top 10 ranked lists. This indicates DMP_CP was not good enough to reliably dock homodimers using existing rigid body docking tools.

When these homodimers were examined further, it was found that 3 out of these 4 homodimers were common between the top 10 and top 15 lists. These 3 homodimers all
had large inter-protein interfaces in the homodimers. The smallest interface out of these three homodimers had 15 inter-chain contacts between the homodimer subunits. The largest interface had 22 inter-chain contacts between the homodimer subunits.

The main objective for further experiments was therefore to determine if it was possible to increase the number of homodimer complexes with at least 2-5 accurate class I contacts in the top 5 or top 10 list of contacts. The method trained in section 3.3 therefore used monomer predictions from a published monomer structure prediction method, DMPfold (Greener, Kandathil and Jones, 2019)

3.3. Prediction of inter-protein contacts using monomers information predicted with DMPfold

The work done in sections 3.2 and 3.3 tested the ability of deep learning networks to learn how to predict inter-chain and intra-chain homodimer contacts simultaneously. Unfortunately, the methods evaluated in section 3.3 had very limited success when predicting inter-chain contacts.

The goal of this chapter (Chapter 3) was to explore the use of deep-learning methods to do completely *de-novo* prediction of homodimer structures where crystal structures were unavailable for both the monomer and complex structure. In this section, a monomer prediction method, DMPfold, was used to generate intra-chain contact predictions for homodimer subunits from the amino acid sequences of the monomer chains. These intra-chain contact predictions were fed as an additional input to the DMP_CP method trained in section 3.2. The aim of doing this was to see if adding monomer predictions from DMPfold, which had been shown to have decent performance in monomer structure prediction, improved the ability of DMP_CP to correctly predict homodimer inter-chain contacts.

Before the experiments described in this section were started (June 2019), a number of deep-learning methods for single-domain and single-chain structure prediction had been published, including DMPfold (Greener, Kandathil and Jones, 2019) from the Jones lab.
These methods showed significant improvement in prediction of the structure of single domains and single protein chains (AlQuraishi, 2019; Greener, Kandathil and Jones, 2019; Xu and Wang, 2019; Senior et al., 2020).

The DMPfold method used residual neural networks similar to those used in the DeepMetaPSICOV (Kandathil, Greener and Jones, 2019) to learn inter-atom distance bounds, main chain hydrogen bond networks, and torsion angles (Greener, Kandathil and Jones, 2019). One difference between DMPfold and DMP was that instead of predicting binary contacts between residue pairs, DMPfold output a probability distribution over 20 distance bins for inter-residue distances. The distance, hydrogen bond, and torsion angle predictions were then fed as constraints to CNS (Brunger, 2007) over 3 iterations (Greener, Kandathil and Jones, 2019). CNS outputs for each iteration were used to refine the distance and H-bond prediction. Once 3 iterations of this process were completed, a final clustering was performed to get the final structure predictions (Greener, Kandathil and Jones, 2019).

DMPfold matched DMP in terms of contact prediction for single chains, but provided the benefit of having more granular structure information, including distance, H-bond and torsion angle predictions as well as output structures (Greener, Kandathil and Jones, 2019; Kandathil, Greener and Jones, 2019). In addition, DMPfold was shown to have better overall performance for structure prediction in terms of TM-score for the top 5 generated models than CONFOLD2 and Rosetta, two previously existing structure prediction methods (Greener, Kandathil and Jones, 2019). The TM-score for a pair of crystal or predicted protein structures is a measure of the topological similarity of the protein structures (Zhang and Skolnick, 2004). The TM-score has two advantages compared to the root-mean-square deviation (RMSD) of pairs of proteins (Zhang and Skolnick, 2004). First, the TM-score is more sensitive to the global fold similarity of two proteins rather than local structural variations (Zhang and Skolnick, 2004). Second, TM-score uses a length-dependent scale to normalize distance errors, which makes it easier to compare structures with different lengths. Generally, pairs of protein structures with a TM-score greater than 0.5 are generally assumed to have the same fold in SCOP (Zhang and Skolnick, 2004).

The equation for TM-score is shown below in equation 2.7.
Where $L_N$ and $L_T$ are the length of the native structure and length of the aligned residues in the template structure, $d_i$ is the distance between aligned residues, and $d_o$ is a scale to normalize local differences (Zhang and Skolnick, 2004). TM-scores have values between 0 and 1, with a higher score corresponding to higher similarities between protein structures being compared.

It is important to note that although some of the methods, including AlphaFold (Senior et al., 2020) and RaptorX (Xu and Wang, 2019) had shown better performance on monomer structure prediction than DMPfold during CASP12, the code used to generate structures was not available for either of these methods when the work in this chapter was started. Additionally, although a webserver was available for the Raptor-X method, the server was quite slow for individual predictions. It would have taken weeks or months to generate predictions for the 7517 homodimers in the RCSB homodimer dataset 2.0. For this reason, although it would have been interesting to see how the improvements in monomer prediction of these methods affected the prediction of inter-chain contacts in homodimer complexes using deep learning, that exploration was unfeasible at the time.

In contrast, the code for DMPfold and the information about the DMPfold training data were both available. In this section (3.3), the best deep learning method explored in section 3.2, DMP_CP, was trained using DMPfold monomer predictions to get inter-chain contact predictions. This new combined method is called DMPfold + DMP_CP.

For the experiments done in this section, it was assumed that the intra-chain distance or contact predictions from DMPfold would be more accurate than intra-chain contact predictions made by DMP_CP. This would be a reasonable assumption if homodimer subunits did not undergo significant conformational changes in the dimer state, or where the bound conformation of the monomer was represented in the DMPfold training set.
3.3.1 Methodology

To train the DMPfold + DMP_CP models explored in this section, DMPfold outputs were combined with the 60-channel map inputs. To obtain the DMPfold outputs used as inputs for the models trained in this section, the DMPfold method was run on all the homodimers in HB2.0. DMPfold predictions were successfully generated for 7,179 homodimers: 6,985 in the training set, 98 in the validation set, and 96 in the test set.

Once DMPfold was run on the monomer chains that made up the homodimer complexes of the RCSB homodimer dataset 2.0, output files from DMPfold were saved and turned into input features for inter-chain contact prediction with DMPfold-DMP_CP. DMPfold outputs at least 1 predicted structure for each input sequence. For the analysis work done in this section, distance and contact maps were extracted for the top predicted structure from DMPfold (in the final_1.pdb file).

Overall, 2 input types with different feature dimensions could be fed into the DMPFold + DMP_CP method trained in this section. These input types include 1-channel distance and contact maps from the output PDB structure of DMPfold.

The distance map for each DMPfold prediction was obtained by extracting the Cβ atom distances (Cα in the case of glycine) for each residue pair in the predicted DMPfold structure. The binary contact map was generated by extracting all residue pairs with Cβ distances (Cα in the case of glycine) less than 8Å.

To ensure robust cross validation, it was important to ensure the validation and test sets used to evaluate DMPfold + DMP_CP had no overlap with the DMPfold training set. Homodimer chains that overlapped with the DMPfold training set were removed from the validation and test sets by using the ECOD-analyse method described in chapter 2. This left 92 homodimers in the validation set, and 84 homodimers in the test set.

3.3.1.2 Model Architecture
The model architecture for DMPfold + DMP_CP was identical to DMPContactPredict, with only two differences. First, the input tensor had one additional channel for either the distance or contact map of the DMPfold predicted monomer. The second difference was the output layer had only one output dimension, which corresponded to the likelihood of a residue pair being an inter-chain residue contact. Figure 3.6 shows the method that was used to generate inputs for the DMPfold + DMP_CP inter-chain prediction method.

![Diagram](image)

**Figure 3.6: DMPfold + DMP_CP Method:** The DMPfold + DMP_CP method combines the 60-channel .map inputs with residue pair distances extracted from DMPfold monomer structure predictions. Differences to the original DMPContactPredict model are shown in purple.

As in previous sections, inter-chain contacts were defined as residue pairs with less than 10Å distance between Cα atoms. Training and prediction of DMP_CP was done using crops of 67 by 67 residues, with the residue pair of interest at the centre of the crops.

### 3.3.1.2 Model Training

#### Loss Functions

A new loss function, margin ranking loss, was used to train the DMPfold + DMP_CP model. In Chapter 2, the two main functions that were explored for the training and
evaluation of the deep learning models were the binary cross entropy and MCC loss functions. Both these loss functions are used when trying to accurately differentiate between binary classes (i.e. inter-chain vs. intra-chain residue contacts). The work in this chapter, however, was aimed at determining whether it would be possible to identify inter-chain contacts by ranking them higher than residue pairs that were not involved in interface contacts. This meant that the goal was to accurately rank contacts by their likelihood of being involved in inter-chain interactions rather than simply giving a binary prediction of whether it was an inter-chain or intra-chain contact. For this reason, the margin ranking loss was used as it is ideal for such ranking problems. The margin ranking loss, or contrastive loss, is aimed at predicting distances between the likelihoods of two residue pairs being inter-chain contacts. Note here, the term ‘distance’ does not refer to spatial atomic distances, but to likelihood differences between two sets of residue pairs being involved in an inter-chain contact. The loss therefore is aimed at getting the network to rank positive inter-chain contacts higher than non-inter-chain contacts.

As described in the PyTorch documents, the margin ranking loss function for a mini-batch is calculated by the following equation:

\[
\text{loss}(x_1, x_2, y) = \max(0, -y \times (x_1 - x_2) + \text{margin}) \quad (Eq. 3.8)
\]

This equation gives the margin ranking loss between two pairs of residues (x1 and x2) given a predicted ranking distance between the two residues (y).

For positive pairs (i.e. pairs where both residue pairs (x1 and x2) are inter-chain contacts or both residue pairs are not inter-chain contacts), the loss is 0 when the predicted ranking distance of the contacts x1 and x2 is less than the margin and is greater than 0 when the predicted distance for the contacts is greater than the margin.

For negative pairs (i.e. where residue pair is an inter-chain contact and the other is not) the loss is 0 if the inter-chain contact is correctly ranked higher than the non-inter-chain contact by a ranking distance larger than value of the margin, and loss is greater than 0 if this is not the case.

For the models trained in this section, a margin of 1 was used for the margin ranking loss.
Model Training Schemes

Two input types were explored for the prediction of homodimer inter-chain contacts. These model inputs are summarized in Table 3-8.

<table>
<thead>
<tr>
<th>Model Training Scheme</th>
<th>Input type</th>
<th>Number of input features</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cβ- Cβ atom distance map</td>
<td>61</td>
<td>DMPfold output structure</td>
</tr>
<tr>
<td>2</td>
<td>Cβ- Cβ atom contact map</td>
<td>61</td>
<td>DMPfold output structure</td>
</tr>
</tbody>
</table>

Table 3-8: Training schemes for DMPfold-DMP_CP: Input types, feature dimensions, and loss functions used for training schemes explored for the training and evaluation of DMPContactPredict

Network training

The models were trained on the RCSB homodimer dataset 2. The Adam optimizer (Kingma and Ba, 2014) was used for training. Training was done on batches of 60 crops.

Each training epoch consisted of training on examples from 500 homodimer complexes at a time. As the training set contained 7,317 training examples, this meant that it took ~15 epochs to cover the training set.

Evaluation of DMPfold + DMP_CP on Test Set

Since all homodimers that had structural overlap with the training set had to be removed from the test set, the performance of DMPfold + DMP_CP was evaluated on 84 homodimers left in the test set.
3.3.2 Results

For both training schemes the networks that had the best validation MCC score over the training epochs were saved for evaluation on the test set. The training, validation, and test results for the best saved DMPfold + DMP_CP models are shown in Table 3-9.

<table>
<thead>
<tr>
<th>DMPfold Input Type</th>
<th>Model Performance on Training and Validation Sets (HD2.0)</th>
<th>Model Performance on Test Set (HD2.0)</th>
<th>Class I Contact Precision on Test Set - Top N Contacts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training loss</td>
<td>Validation loss</td>
<td>Validation class I MCC</td>
</tr>
<tr>
<td>Distance map</td>
<td>0.4251</td>
<td>0.6385</td>
<td>0.2116</td>
</tr>
<tr>
<td>Contact map</td>
<td>0.4428</td>
<td>0.6880</td>
<td>0.1742</td>
</tr>
</tbody>
</table>

Table 3-9: Training schemes for DMPfold-DMP_CP: Training, validation, and test metrics for the two versions of DMPfold + DMP_CP. Numbers in bold indicate the best performing model for a particular metric. Ranked contact predictions for top 5, 10, and 15 contacts are included here.

Top 5, 10, and 15 precision values for top ranked inter-chain contacts for DMPfold + DMP_CP were much higher overall than for DMP_CP alone (shown in table 3.7).

The number of homodimers in the test set for which a certain number of correctly predicted class I contacts were in the top 5, 10 and 15 ranked inter-chain contacts for DMPfold + DMP_CP are shown in Figure 3.7.
Figure 3.7: Homodimer Complexes with Correct Class I Contacts DMPfold + DMP_CP:
Number of homodimers in test set with at least 1, 2, 3, or 4 correct class I contacts in top 5, 10, 15 ranked lists for DMPfold + DMP_CP

Overall, the performance of DMPfold + DMP_CP was far better than previous versions of DMP_CP trained in section 3.3. The DMP_CP method seemed to benefit greatly from the addition of residue distance predictions from DMPfold predicted monomer structures.

To determine whether the quality of DMPfold structure predictions had an impact on the precision of contact precision of DMPfold + DMP_CP, the RMSD of the predicted DMPfold structures to native structures were extracted and plotted against the precision of class I contact predictions for the homodimer complexes in the test set. Figure 3.8 shows the correlation between the RMSD of the monomers and the precision of class I contact predictions.
As can be seen from Figure 3.8, some correlation can be observed between the quality of the structure produced by DMPfold and the precision of the contacts predicted by the DMPfold + DMP_CP method. This suggests that improved monomer structure prediction correlates to better inter-chain residue contact prediction.

Structures of 11 DMPfold generated structures from the test set are shown in Table 3-10. The RMSDs of these structures, along with the TM-scores and the DMP_CP class I contact prediction are also shown in Table 3-10. As can be seen from the structures and class I precision values in the table, visibly better monomer structure predictions were generally correlated to higher inter-chain contact precision. Images in table 3.10 were generated using PyMol (Schrödinger, LLC, 2015)
<table>
<thead>
<tr>
<th>PDB ID</th>
<th>RMSD TO NATIVE</th>
<th>PREDICTED TM-SCORE</th>
<th>CLASS I CONTACT PRECISION</th>
<th>STRUCTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2HJM_A</td>
<td>2.324</td>
<td>0.68</td>
<td>0.315</td>
<td>![Image](402x582 to 540x710)</td>
</tr>
<tr>
<td>1A2Z_C</td>
<td>2.3368</td>
<td>0.74</td>
<td>0.6215</td>
<td>![Image](361x443 to 578x568)</td>
</tr>
<tr>
<td>1QOJ_A</td>
<td>2.382</td>
<td>0.77</td>
<td>0.3527</td>
<td>![Image](402x299 to 554x408)</td>
</tr>
<tr>
<td>1A95_C</td>
<td>3.073</td>
<td>0.76</td>
<td>0.0067</td>
<td>![Image](402x122 to 548x230)</td>
</tr>
<tr>
<td>PDB ID</td>
<td>RMSD TO NATIVE</td>
<td>PREDICTED TM-SCORE</td>
<td>CLASS ICONTACT PRECISION</td>
<td>STRUCTURE</td>
</tr>
<tr>
<td>--------</td>
<td>----------------</td>
<td>--------------------</td>
<td>--------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>1IS7_Q</td>
<td>4.228</td>
<td>0.51</td>
<td>0.6358</td>
<td></td>
</tr>
<tr>
<td>1A1U_A</td>
<td>4.964</td>
<td>0.82</td>
<td>0.4444</td>
<td></td>
</tr>
<tr>
<td>1EZS_A</td>
<td>5.045</td>
<td>0.56</td>
<td>0.1538</td>
<td></td>
</tr>
<tr>
<td>1NI8_A</td>
<td>5.590</td>
<td>0.72</td>
<td>0.1523</td>
<td></td>
</tr>
<tr>
<td>PDB ID</td>
<td>RMSD TO NATIVE</td>
<td>PREDICTED TM-SCORE</td>
<td>CLASS I CONTACT PRECISION</td>
<td>STRUCTURE</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>1K1F_C</td>
<td>7.858</td>
<td>0.23</td>
<td>0.0083</td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
<tr>
<td>2I15_A</td>
<td>16.507</td>
<td>0.58</td>
<td>0</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>1YAU_Q</td>
<td>23.839</td>
<td>0.47</td>
<td>0</td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Table 3-10: Native monomer structures and DMPfold structures for 10 PDBs from test set:
Structures in pink are the native crystal structures for the monomers, while structures in yellow are the DMPfold generated monomer structures. Here predicted TM-score corresponds to the predicted TM-score for the monomer predicted by DMPfold.
3.4. Conclusion: Deep learning methods for \textit{de novo} prediction of homodimer contacts

The work in this chapter was focused on exploring the application of deep-learning methods for the \textit{de novo} prediction of homodimer complexes. At the time the work summarized in this chapter was carried out, no methods existed that were aimed at completely \textit{de novo} prediction of homodimer complexes without close structural homologs. A few servers did exist to perform template-based modelling of homodimer complexes, but these methods relied on being able to find close enough structural homologs (Lensink \textit{et al.}, 2016; Baek \textit{et al.}, 2017). In addition, a number of coevolution based-methods had previously been explored for the prediction of homodimer complexes (Wang and Barth, 2015; Uguzzoni \textit{et al.}, 2017; Xu and Wang, 2019), all methods published relied on having crystal structures for constituent monomers to remove intra-chain contacts from overall predicted contact lists to get inter-chain contacts.

A number of different methods were tested for the \textit{de novo} prediction of homodimer inter-chain and intra-chain contacts. Two deep learning architectures, CNNs and ResNets, were explored in sections 3.2 and 3.3 for simultaneous prediction of inter-chain and intra-chain contacts in homodimer complexes. The results showed that although the DMPContactClassify performed better than CNNContactClassify on simultaneous prediction of the two contact types, the precision of inter-chain contact predictions was not high enough to be used for overall prediction of the complex structure (i.e., for ranking of docking decoys). Overall, the deep learning methods trained on the HD2.0 were not powerful enough to reliably predict both monomer contacts and homodimer complex structures.

In section 3.4, a different method, DMPfold + DMP\_CP was trained and evaluated. This model was trained by adding DMPfold residue distance predictions for the monomer subunits to the 60-channel map inputs previously used to train contact classification models (Chapter 2). DMPfold + DMP\_CP was set up to only do prediction of class I contacts rather than simultaneous prediction of class I and class II contacts.
DMPfold intra-chain contact predictions were better than DMP_CP intra-chain contact predictions for a number of reasons. First, the DMPfold training set was made up of a larger cleaned dataset of monomer chains. Second, the iterations through the distance, H-bond, and torsion angle prediction and CNS model generation that are part of the DMPfold method, have been shown to significantly improve the quality of the predicted structure after each iteration (Greener et al., 2019). Using CNS to generate models from distance restraints also helps improve the overall structure predictions by ensuring the generated structures are physically sensible.

Previous studies (Wang and Barth, 2015; Lensink et al., 2016; Uguzzoni et al., 2017) have shown that using monomer crystal contacts to remove intra-chain contacts from predicted contact maps is necessary when doing inter-chain contact prediction using co-evolution based methods. Here, giving better contact or distance information (i.e. information that is closer to the contact or distance information of native structures) may help the DMPfold + DMP_CP model better differentiate between coevolution signals of inter-chain contacts and intra-chain contacts.

It is also possible improved information about the monomer structure helps the DMP_CP method better determine what regions are likely to have inter-chain contacts. For example, if the model is able to better differentiate buried versus surface residues of the monomers, it may reduce the likelihood of predicting inter-chain interacting residues in buried regions. This is conjecture, however, and further work would need to be done to determine what information DMP_CP is using to make better class I contact predictions. This is an area that could be explored by future research work.

There were a few cases where the TM-score was low or the RMSD to the native monomer structure was high but contact precisions above 0.4 were obtained. Overall, for at least one of these cases, this was because the interface between the interacting monomer units was fairly large (37-52 inter chain contacts for 3 of these homodimer complexes). It is possible, therefore, that better DMPfold + DMP_CP predictions may be obtained for homodimers with very large interfaces even when the predicted monomer structures are not correct.
Although the overall MCC (0.1789) and precision (0.1795) of DMPfold + DMP_CP was still not very high on the homodimer test set, it was significantly higher than the DMP_CP method explored in section 3.2. The results in figures 3.7 and 3.8 indicate that better monomer structure predictions resulted in better prediction of inter-chain contacts, although there is clearly much room for improvement for both better monomer structure predictions and inter-chain contact predictions.

Deep learning delivers a number of important benefits over previously used coevolution-based methods, by both improving intra-chain contact or monomer structure prediction and improving differentiation of inter-chain and intra-chain contacts. Further improvements could be obtained by training on larger datasets, using more powerful deep learning methods such as transformers, using better monomer structure predictions to predict inter-chain contacts or distances, and using iterative modelling of complexes. These are all areas that could be explored to improve de novo homodimer complex modelling. In addition, further work could be done to combine the monomer structure model of DMPfold with the dimer contact prediction of DMP_CP to create an end-to-end homodimer complex prediction method. It may also be possible to use metrics like the DMPfold TM-score and predicted interface size to predict the likelihood of a complex being correctly predicted by DMPfold + DMP_CP. This analysis was outside of the scope of this thesis but could be explored in future work.

Work in the next chapter, chapter 4, explored use of inter-chain contact predictions from DMPfold + DMP_CP for the docking of homodimer complexes.
4. Homodimer Docking with Predicted Inter-Chain Contacts

Work done in Chapter 3 explored the ability of 3 deep learning frameworks (CNNContactPredict, DMP_CP, and DMPfold + DMP_CP) to predict inter-chain contacts. In this chapter, inter-chain contact predictions from DMPfold + DMP_CP were used to rank docking decoys from ZDock (Pierce, Hourai and Weng, 2011). This was done on the DMPfold + DMP_CP test set that was evaluated in chapter 3. The goal of the work in this chapter was to determine whether it was possible to use inter-chain contact predictions from DMPfold + DMP_CP to correctly select docking decoys that were close to native complex structures from a large set of predicted docking poses. At the time this work was conducted, no published tools could perform the simultaneous de novo folding and docking of homodimer complexes. The goal of the work in this chapter was to therefore explore the ability of DMP-based methods to correctly fold the monomer chains of homodimer structures and predict inter-chain contacts that could be used to rank homodimer docking decoys to get near-native complex structures.

ZDock has previously been demonstrated in benchmarking studies and CAPRI evaluations to be one of the top 4 tools for rigid body docking (Vreven et al., 2013, 2020). ZDock is a rigid-body protein docking program that uses Fast Fourier Transform to sample protein complex orientations and generate a comprehensive list of possible protein complex structures based on different possible orientations of the monomers (Wiehe et al., 2007; Pierce, Hourai and Weng, 2011, 2011). This list of possible complex structures are called docking decoys and each docking decoy has the two monomers of a dimer being docked in a different orientation – i.e. with different interacting contact residues and a different interaction interface (Wiehe et al., 2007; Pierce, Hourai and Weng, 2011). Once possible orientations of monomers being docked (i.e. decoys) are generated, the decoys are then filtered and ranked by the likelihood of the predicted orientation being the correct one – i.e. the same as the protein complex in the crystal structure of the protein (Pierce, Hourai and Weng, 2011; Vreven et al., 2013). For this reason, the correct ranking of docking decoys is essential to being able to predict the correct structure of a protein complex.
Ranking of docking decoys can be done in a number of different ways, and in this chapter we explore the accuracy of ranking docked decoys in a number of different ways.

Here, ZDock was chosen as a rigid-body docking tool for a number of reasons. First, ZDock is open source and freely available for academic use. Second, ZDock is able to perform very fast generation of docked decoys and can produce up to 10,000 docked models in 1-5 minutes per complex (Pierce, Hourai and Weng, 2011). Third, ZDock comes with an internal decoy ranking score, IFACE + IRaPPA (Vreven, Hwang and Weng, 2011), which has previously been demonstrated to be one of the best performing ranking functions for ranking of docked decoys. IFACE uses a combination of statistical potential, shape complementarity, and electrostatics to score predicted interfaces and poses (Vreven, Hwang and Weng, 2011). IRaPPA (Integrative Ranking of Protein-Protein Assemblies) uses physicochemical descriptions and combines these descriptors using ranking support vector machines (Moal et al., 2017).

In this chapter, contacts predicted by DMPfold + DMP_CP were used to rank docked models generated using ZDock. The ranking of docked decoys using predicted these inter-chain contacts was compared to ranking by ZDock’s internal IFACE + IRaPPA scoring method. Both crystal structures and DMPfold generated structures were used for docking with ZDock to determine whether DMPfold structures can be used to generate near-native docked decoys for homodimer complexes.

4.1 Methodology: ZDock Docking and Ranking

4.1.1 Generating ZDock Docked Decoys

Docked decoys were generated for each of the 84 homodimer complexes evaluated in the test set with DMPfold + DMP_CP predictions in Chapter 3. For each complex, ZDock was run twice: once using the crystal structures of the constituent monomers rotated randomly around the x, y and z axes, and once using the DMPfold generated structures also randomly rotated. 2000 docked decoys were generated for each ZDock run. This meant for each complex, 4000 decoys were generated: 2000 decoys using crystal structures of the monomer subunits, and 2000 decoys using the DMPfold generated
structures. Due to some issues with ZDock docking, docked models were generated for only 78 of the test homodimers.

4.1.2 Docked Decoy Ranking Using ZRank, Real Contacts, and DMP_CP Predicted Contacts

Once docked decoys were generated using ZDock, each set of decoys was ranked using two different methods. First, the decoys were ranked using ZRank, an internal ranking tool of ZDock. This did not have to be done explicitly as ZDock by default output decoys in the order of their ZRank ranking.

Next, the list of real contacts from the native structure were used to rank the docked decoys. This was done by calculating the overlap of the list of residue contacts in the native structure, and the list of contacts in the docked model, as shown in equation 4.1.

\[
\text{Contact Overlap} = \frac{\text{number of overlapping correct contacts}}{\text{total interface contacts in decoy model}} \quad (\text{Eq.} \ 4.1)
\]

Finally, the decoys were ranked using the predicted contacts from DMP_CP. To do this, the top 15 predicted inter-chain contacts were extracted from DMPfold + DMP_CP. ZDock models were then ranked by extracting the contacts in the docked decoys and determining the overlap with the top 15 predicted contacts i.e., how many contacts in the top 15 predicted list were in the contact list of the docked decoy (Eq. 4.2).

The top 15 contacts were used instead of the top 10 contacts because when the top 10 contacts were used, there were often several decoys that had the same rank (i.e., the same number of decoys with the same overlap with the top 10 list).

If more than one decoy had the same overlap with the top 15 contacts, the list of residues involved in the contact was used to refine the ranking. In this case, the list of all residues involved in the interface of the decoy was compared the list of all residues predicted to be in contact from the top 15 contacts as shown in equation 2.9.
4.1.3 Evaluation of Docked Decoys Using DockQ Score

To evaluate the overall quality of the docked models, the DockQ (Basu and Wallner, 2016) score of the models was calculated. DockQ is a continuous model quality score for protein docking models that integrates three of the quality measures usually used in CAPRI evaluations to assess the quality of docking models: the fraction of native contacts in the model, $F_{\text{nat}}$, the ligand RMSD, LRMS, and the receptor-ligand interface RMSD, iRMS (Basu and Wallner, 2016). DockQ can be used to assess model quality and has been shown to closely reproduce CAPRI classifications of models into Incorrect, Acceptable, Medium, and High quality.

DockQ scores correspond to CAPRI docking decoy quality as shown in table 4.1.

<table>
<thead>
<tr>
<th>DockQ score</th>
<th>Decoy Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 0.23</td>
<td>Incorrect</td>
</tr>
<tr>
<td>0.23 – 0.49</td>
<td>Acceptable</td>
</tr>
<tr>
<td>0.49 – 0.80</td>
<td>Medium</td>
</tr>
<tr>
<td>0.80 – 1.0</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 4-1: DockQ scores and docking decoy quality: DockQ classifications of docking decoys based on DockQ score

The DockQ scores of all generated homodimer decoys were extracted and used to assess the number of docked models that had at least acceptable docking quality. Table 4-2 shows examples of docking decoys superimposed with native homodimer complexes for each DockQ category and shows the DockQ scores and full complex RMSDs for each docking decoy.
<table>
<thead>
<tr>
<th>PDB ID</th>
<th>High</th>
<th>Medium</th>
<th>Acceptable</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A1U_AC</td>
<td>RMSD: 0.351 DockQ: 0.946</td>
<td>RMSD: 1.29 DockQ: 0.771</td>
<td>RMSD: 3.564 DockQ: 0.385</td>
<td>RMSD: 7.491 DockQ: 0.229</td>
</tr>
<tr>
<td>1ISZ_QR</td>
<td>RMSD: 0.889 DockQ: 0.814</td>
<td>RMSD: 0.747 DockQ: 1.083</td>
<td>RMSD: 3.491 DockQ: 0.328</td>
<td>RMSD: 7.999 DockQ: 0.078</td>
</tr>
</tbody>
</table>
Table 4.2: DockQ Scoring Examples

Table 4.2: DockQ Scoring Examples: Incorrect, acceptable, medium, and high-quality docking decoys (light blue and green) superimposed with native homodimer complexes (purple).

<table>
<thead>
<tr>
<th>dockQ</th>
<th>RMSD</th>
<th>dockQ</th>
<th>RMSD</th>
<th>dockQ</th>
<th>RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A2Z</td>
<td>8.204</td>
<td>DockQ:</td>
<td>2.931</td>
<td>DockQ:</td>
<td>1.978</td>
</tr>
<tr>
<td></td>
<td>0.070</td>
<td></td>
<td>0.385</td>
<td></td>
<td>0.548</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not Available</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


4.1.4 Decoy ranking and evaluation

Once the ranked list for each of the ZDock docking runs was generated, the top 10 and top 1 ranked models from each of the lists was extracted for each docking run. For each complex, this meant 4 ranked lists were generated:

- Docked crystal structures ranked using ZRank
- Docked crystal structures ranked using predicted contacts from DMPfold + DMP_CP
- Docked DMPfold structures ranked using ZRank
- Docked DMPfold structures ranked using contacts from DMPfold + DMP_CP

DockQ scores of 2000 ZDock generated models from each list were calculated to determine the number of models with at least acceptable quality. These numbers are reported in Table 4-3.

<table>
<thead>
<tr>
<th>Monomer Structure Type</th>
<th>Number of docked models</th>
<th>Complexes with at least one acceptable model (/78)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal structures</td>
<td>2000</td>
<td>59 (75.6%)</td>
</tr>
<tr>
<td>DMPfold structures</td>
<td>2000</td>
<td>23 (29.5%)</td>
</tr>
</tbody>
</table>

Table 4-3: Overall ZDock generated decoy lists: Number of homodimer complexes in test set with at least one acceptable docked model in ZDock-generated lists of 2000 docked models
4.2 Results: Docking and Decoy Ranking

Before evaluating the ability of the ranking methods to rank acceptable docked models in the top 10 and top 1 predicted lists, it was important to first check the overall lists of 1000 docked decoys for each complex with each structure type. This was to determine how many of the complexes had at least one model with an acceptable model quality as determined by DockQ. The number of complexes with at least one acceptable model in the generated list of 2000 decoys for the docked crystal and DMPfold structures are shown in Table 4.3.

When crystal structures were used to generate docked models, 59 out of 78, or 76% of complexes had at least one docked model of acceptable DockQ quality in the full set of generated models. Unfortunately, this number fell by almost two thirds when DMPfold generated structures were used for docking, with only 23 of the 78 complexes (29.5%) having at least one acceptable model in the full set of generated models.

Next, the lists of ZDock generated models were ranked using three ranking methods (ZRank, real contacts, and DMPfold + DMP_CP contacts). The top 1 and top 10 models from each of these lists were extracted, and the number of complexes with at least one model in the top 1 or top 10 ranked list for each ranking method are shown in Table 4-4.
<table>
<thead>
<tr>
<th>Monomer Structure Type</th>
<th>Ranking Method</th>
<th>Number of Models Extracted</th>
<th>Complexes with at least one acceptable model (78)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal Structures</td>
<td>Real Contacts</td>
<td>Top 10</td>
<td>57 (73.1%)</td>
</tr>
<tr>
<td>Crystal Structures</td>
<td>Real Contacts</td>
<td>Top 1</td>
<td>55 (70.5%)</td>
</tr>
<tr>
<td>DMPfold Structures</td>
<td>Real Contacts</td>
<td>Top 10</td>
<td>20 (25.6%)</td>
</tr>
<tr>
<td>DMPfold Structures</td>
<td>Real Contacts</td>
<td>Top 1</td>
<td>17 (21.8%)</td>
</tr>
<tr>
<td>Crystal Structures</td>
<td>ZRank</td>
<td>Top 10</td>
<td>26 (33.3%)</td>
</tr>
<tr>
<td>Crystal Structures</td>
<td>ZRank</td>
<td>Top 1</td>
<td>21 (26.9%)</td>
</tr>
<tr>
<td>DMPfold Structures</td>
<td>ZRank</td>
<td>Top 10</td>
<td>8 (10.2%)</td>
</tr>
<tr>
<td>DMPfold Structures</td>
<td>ZRank</td>
<td>Top 1</td>
<td>4 (5.1%)</td>
</tr>
<tr>
<td>Crystal Structures</td>
<td>DMPfold + DMP_CP</td>
<td>Top 10</td>
<td>19 (24.3%)</td>
</tr>
<tr>
<td>Crystal Structures</td>
<td>DMPfold + DMP_CP</td>
<td>Top 1</td>
<td>14 (17.9%)</td>
</tr>
<tr>
<td>DMPfold Structures</td>
<td>DMPfold + DMP_CP</td>
<td>Top 10</td>
<td>5 (6.4%)</td>
</tr>
<tr>
<td>DMPfold Structures</td>
<td>DMPfold + DMP_CP</td>
<td>Top 1</td>
<td>0 (0.0%)</td>
</tr>
</tbody>
</table>

Table 4-4: Overall ZDock generated decoy lists: Number of homodimer complexes in test set with at least one acceptable docked model in ZDock-generated lists of 2000 docked models.

The results in table 4.4 show the number of complexes (out of 78) which had an acceptable model by CAPRI docking standards in either the top 10 ranked docking decoys or the top 1 docking decoy. Docking decoys were generated by done either the crystal structures of the monomer subunits, or the predicted monomer structures from DMPfold. Ranking of the top 10 or top 1 predicted docking decoys was done using three methods. The first method was using real contacts from the crystal structures. The second method was using the internal ZRank method of ZDock, which combines iFACE and IRaPPA scoring to rank decoys (Wiehe et al., 2007; Moal et al., 2017). Finally, predicted contacts from DMPfold + DMP_CP were used to rank the docking decoys by determining...
the decoys with the most overlap with the list of predicted contacts and using this overlap to rank the decoys.

Overall, for all decoy ranking methods, the number of complexes with acceptable or better quality were much higher for docked crystal structures than for docked DMPfold structures. In addition, when real contacts were used to rank crystal structures, over two thirds of crystal structures had at least one model of acceptable quality in the top 1 and top 10 ranked decoys. Conversely, when real contacts were used for ranking, about a fourth of docked DMPfold structures had acceptable models in the top 1 or top 10 ranked decoys.

Decoy ranking using the ZRank method performed better overall than ranking using the DMP_CP contacts.

In general, when docking crystal structures, DMPfold + DMP_CP ranking tended to be successful in ranking decoys in cases with higher contact precision. All complexes with successfully docked decoys at the top of the decoy ranking list had DMPfold + DMP_CP prediction precisions greater than 0.2667, with an average precision of 0.3143. This corresponds to between 4 and 7 correct contacts in the top 15 predicted contact lists.

ZRank and DMP_CP did not always perform well on the same complexes. This meant that the set complexes where ZRank successfully ranked acceptable models in the top 10 or top 1 ranks was not the same as the set of complexes DMPfold + DMP_CP was able to successfully predict rank.

Figures 4.1 – 4.4 show examples of the top ranked docked decoys for the homodimer complexes in the test set. The structures of the native monomer crystal structures and DMPfold models used for docking for these complexes can be seen in Table 3-10. The TM-scores of the DMPfold generated structures can also be seen in Table 3-10.

Figure 4.1 shows an example complex (1A1U_AC) where ZRank and DMPfold + DMP_CP ranking methods both successfully ranked medium or high-quality decoys at the top of the model ranking list for crystal structures. All images of protein structures were generated using PyMol (Schrödinger, LLC, 2015)
Figure 4.1: 1A1U_AC Docked Complexes: Native homodimer complex structure (salmon) superimposed with top ranked decoys (green and blue): a. Docked crystal structures ranked using ZRank b. Docked crystal structures ranked using DMPfold + DMP_CP top 15 contacts c. Docked DMPfold generated structures ranked using ZRank d. Docked DMPfold generated structures ranked using DMPfold + DMP_CP top 15 contacts

The best ranked decoy for both ranking methods for the DMPfold generated structures were much worse than the best ranked decoy for docked crystal structures. The DMPfold monomer structures used for docking had an RMSD of 4.964 and a top 15 inter-chain contact precision of 0.3333, which corresponded to 5 correctly predicted contacts. In this case, even though the complex had a higher-than-average inter-chain precision
compared to other homodimers in the test set, it seems the quality of the DMPfold monomer structure was not high enough to result in high-quality docking decoys from Zdock: none of the docked decoys generated by ZDock (2000 decoys) for this complex had an acceptable DockQ score.

Figure 4.2 shows another complex (1IS7_QR) where ZRank and DMPfold + DMP_CP were able to rank medium or higher quality docked models at the top of ranking lists for crystal structures.

**Figure 4.2: 1IS7_QR Docked Complexes:** Native homodimer complex structure (salmon) superimposed with top ranked decoys (green and blue): a. Docked crystal structures ranked using ZRank b. Docked crystal structures ranked using DMPfold + DMP_CP top 15 contacts c. Docked DMPfold generated structures ranked using ZRank d. Docked DMPfold generated structures ranked using DMPfold + DMP_CP top 15 contacts
Similar to 1A1U_AC, the docked decoys that generated by docking crystal structures were much better than the docked decoys generated using DMPfold structures. The RMSD of the DMPfold monomer to native was 4.228, and the DMPfold + DMP_CP inter-chain precision was 0.4, which corresponded to 6 correctly predicted contacts. Although the inter-chain contact prediction precision was high enough to correctly rank acceptable crystal structure docked decoys, the DMPfold structure wasn’t good enough to get acceptable docked decoys after ranking. There was only one decoy of acceptable quality in the docked decoy list generated by Zdock, but the decoy was not ranked in the top 10 ranked decoys.

Figure 4.3 shows the best ranked decoys for 1A2Z_CD, where the best ranked decoy using DMP_CP for both crystal and DMPfold generated structures were better than ZRank ranked decoys.
Figure 4.3: 1A2Z_CD Docked Complexes: Native homodimer complex structure superimposed (salmon) with top ranked decoys (green and blue): a. Docked crystal structures ranked using ZRank b. Docked crystal structures ranked using DMPfold + DMP_CP top 15 contacts c. Docked DMPfold generated structures ranked using ZRank d.Docked DMPfold generated structures ranked using DMPfold + DMP_CP top 15 contacts

For 1A2Z_CD, the qualities of the top ranked docking decoys for both the crystal structures and DMPfold generated structures were better than the top ranked decoys by ZRank. The RMSD for the DMPfold structure of 1A2Z_CD was 2.3368. This complex also had the highest top 15 precision of any of the complexes in the test set, with a precision of 0.4667 or 7 correct class I contacts in the top 15 predicted contact list. For 1A2Z_CD, it seems higher monomer structure quality may have contributed to better inter-chain contact predictions. The better quality also resulted in a better set of docking of decoys than 1A1U or 1IS7 – Zdock was able to produce 5 decoys of acceptable quality when
docking the DMPfold generated monomer structures (DockQ scores between 0.23 and 0.49).

Neither ZRank nor DMPfold + DMP_CP, however, ranked any of those decoys at the top of the ranked decoy lists. For DMPfold + DMP_CP ranking, 1 acceptable model was included in the top 10 ranked docked decoys, with a DockQ score of 0.241. For 1A2Z_CD, a decent DMPfold structure and good DMPfold + DMP_CP precision resulted in better docking decoys and decoy ranking, but even then, ranking with ZRank or DMP_CP interchain contacts did not result in an acceptable model for the top ranked decoy.

In contrast to 1NI8_AB, Figure 4.4 shows the results for a different complex, 1NI8_AB, where the top ZRank decoys were much better than the corresponding top DMPfold + DMP_CP predicted models.
Figure 4.4: 1NI8_AB Docked Complexes: Native homodimer complex structure superimposed (salmon) with top ranked decoys (green and blue): a. Docked crystal structures ranked using ZRank b. Docked crystal structures ranked using DMPfold + DMP_CP top 15 contacts c. Docked DMPfold generated structures ranked using ZRank d. Docked DMPfold generated structures ranked using DMPfold + DMP_CP top 15 contacts

For 1NI8_AB, the ZRank ranked top decoys for the docked crystal structures was much better than the DMP_CP ranked top decoy. The top decoys for the DMPfold docked structures were much worse than the crystal docked models. The RMSD for the DMPfold structures used for docking was 5.59, and the DMPfold + DMP_CP precision was 0.1333, which corresponded to 2 correctly predicted class I contacts. For this complex, while ZRank ranking of crystal decoys had high quality, the decoys created using the DMPfold structures and DMPfold + DMP_CP contacts were all incorrect. This result seems to indicate that a higher precision of inter-chain predictions is needed to outperform ZRank for ranking docking decoys.
4.3 Conclusions

The work in this chapter explored the use of homodimer inter-chain contact predictions from the DMPfold + DMP_CP method developed in Chapter 3 for the ranking of docking decoys. These homodimer decoys were generated using the ZDock protein docking method (Pierce, Hourai and Weng, 2011) that has previously been shown to be one of the fastest and best performing docking methods in CAPRI evaluations (Wiehe et al., 2007; Vreven et al., 2013). Performance of homodimer docked decoy ranking was measured for docking of crystal structures ranked using ZDock 3.0.2 (Pierce and Weng, 2007), which uses IFACE, which scores docked decoys based on statistical potential, shape complementarity and electrostatics of interfaces (Vreven, Hwang and Weng, 2011). This decoy ranking was compared to ranking of decoys using predicted inter-chain contacts from DMPfold and DMPfold + DMP_CP.

Docked decoys were made up of either two crystal monomer structures, or two DMPfold generated monomer structures. Both ranking methods performed worse at ranking acceptable models than using real contacts, which demonstrates room for improvement for homodimer interface prediction for both statistical potential methods like IFACE and contact-prediction based method like DMPfold + DMP_CP.

Results in table 4.4 demonstrate the fact that the ZRank ranking performed better overall in terms of the number of complexes with at least one acceptable model in the top 10 or top 1 ranked models across docked model types. Further examination of the docked models, however, shows that the precision of the top 15 class I contacts from DMPfold + DMP_CP had a significant impact on the ability of DMPfold + DMP_CP to rank acceptable models at the top of the ranked decoy lists. Indeed, in many cases where the docked model ranked first by DMP_CP was better than the docked model ranked first by ZRank, the precision of the top 15 DMP_CP predicted contacts was much higher (average: 0.3143) than the overall average precision of DMPfold + DMP_CP across all complexes (average: 0.1278).

Out of the 14 docked crystal structure decoys with a top-ranked model of acceptable or better-quality ranked using DMPfold + DMP_CP ranking, 6 models did not have an
acceptable top ranked model from ZRank. Out of these, 4 models did not have an acceptable model in the top 10 ranked models for ZRank. All 6 of the models that had better rank for DMP_CP ranking than ZRank had DMP_CP precision greater than 0.3333, or at least 5 correctly predicted contacts in the top 15 predicted class I contacts. The results in this section demonstrate the importance of high class I contact precision (0.3333 or greater) for better ranking of docked decoys.

The results in this chapter also demonstrate that correct docking of DMPfold generated structures is much more challenging than docking of crystal structures. Overall, better DMPfold structures tended to correspond to better sets of docked decoys. The impact of better DMPfold structures was two-fold: better structures often correlated to better DMPfold + DMP_CP contact prediction precision, and also tended to result in better sets of docked models from ZDock.

Unfortunately, the overall results in this section demonstrate the difficulty of performing completely de novo prediction of homodimer complexes where close structural templates do not exist. All cases where the RMSD of the DMPfold structure to native was higher than 4.5Å resulted in incorrect docking decoys for all 2000 decoys generated by ZDock. The complexes that had at least one acceptable decoy in the 2000 decoys generated by ZDock had an RMSD less than 4.5 Å to native protein structures, and the complexes where the DMPfold structure had an RMSD less than 2.5 Å to native had at least 4 acceptable models in the ZDock generated set.

These results demonstrate two areas where overall improvement could result in better de novo prediction of homodimer structures: improved prediction of monomer structures and improved prediction of inter-chain contacts for homodimer complexes. These are areas that can be explored in future work.

It should be noted that since the work in this chapter was done, a number of methods have been developed using similar techniques for the de novo prediction of homodimer complexes. Since CASP 14, two methods in particular, tr-Rosetta-homo (Baek, Anishchenko, et al., 2021) and AlphaFold-multimer (Evans et al., 2021) have been developed and have demonstrated significant improvements in homodimer complex
Both these methods combine deep learning with structural modelling and refinement steps to generate homodimer complexes.

In the case of tr-Rosetta-homo, contacts were generated using a 2D residual convolutional network on MSAs similar to DMP_CP (Baek, Anishchenko, et al., 2021). In the case of trRosetta, however, inter-chain contacts were defined as any residues with 12Å Cβ-Cβ distances, which was less stringent than the contact distance cut-offs used for the methods developed in this chapter. These contacts were combined with high-probability contacts generated by the GREMLIN coevolution based method (Baek, Anishchenko, et al., 2021). These contacts were filtered using contacts in the monomer structure, so the trRosetta-homo method did depend to some extent on knowing monomer contact predictions, although in cases where templates couldn’t be found for the monomers, predicted monomer contacts were used for filtering. To perform prediction of homo-oligomers where high-confidence inter-chain contact predictions could be made, a gradient-based fold and dock method was used to predict complex structures (Baek, Anishchenko, et al., 2021). This method employed a gradient-based energy minimization to sample structures. Although this method has advantages over the methods developed in this chapter in that it could theoretically perform flexible docking of homodimers, the method was only evaluated on 8 CASP 14 assembly complexes, and therefore rigorous benchmarking would be required to evaluate the inter-chain contact predictions of trRosetta-homo. This benchmarking is outside the scope of this thesis but would be an interesting area for future exploration.

In addition to trRosetta, CASP 14 also saw a significant development in protein structure prediction with the AlphaFold2 method. Since AlphaFold2 results were announced at CASP14, a modified version of AlphaFold2 (Senior et al., 2020; Jumper et al., 2021), called AlphaFold-multimer has been released and demonstrated significant improvements for oligomer prediction from previously existing methods (Evans et al., 2021). When evaluated on a set of 4,446 homology-reduced protein complexes from PDB, AlphaFold-multimer successfully predicted 72% of interfaces, and produced high-accuracy predictions for 36% of cases (Evans et al., 2021). Unfortunately, these results did not give a clear indication on the performance of AlphaFold-multimer on completely novel homo-oligomers as the set used for prediction was not robustly filtered against the
AlphaFold training set by both sequence and structural similarity. Additional benchmarking studies (Akdel et al., 2021; Yin et al., 2022) have also demonstrated great improvements in prediction of protein complexes. Similar to the AlphaFold-multimer paper, the cases that AF-multimer was tested on were not robustly filtered against the AF2 training set. One of these benchmarks (Yin et al., 2022) was performed on Protein Docking Benchmark 5.5 (Vreven et al., 2015), which was published in 2015 and is likely to have been included in the AF2 training set. Similarly, a large-scale community assessment of AlphaFold2 by Akdel et al (Akdel et al., 2021) showed improved performance of AlphaFold-multimer complex prediction, but was assessed on the DockGround 4.3 (Kundrotas et al., 2020) benchmark, which was frozen in 2018 and may have some overlap with the AlphaFold2 training set. Although AlphaFold-multimer seems to have improved upon previously existing methods for homodimer prediction, rigorous benchmarking might be needed to determine how much AlphaFold-multimer has improved completely de novo homodimer structure prediction where template structures do not exist for either monomer subunits or the homodimer complex. This is another area for future exploration.

One way of exploring the impact of improved monomer structures from AlphaFold2, would be to see how using AlphaFold2 generated monomers affects the prediction of inter-chain contacts using DMPContactClassify, as well as the ranking of docking decoys using these predicted contacts.

Overall, results in this chapter demonstrated promising applications for deep-learning based methods for prediction of the monomer subunits and complex structures of homodimers.
5. Benchmarking of \textit{in silico} heterodimer complex prediction methods

5.1 Introduction and motivation: heterodimer complex prediction

Hetero-oligomers are protein complexes made up of different interacting protein subunits. Heterodimers are the simplest hetero-oligomers and are made up of two interacting protein subunits with different sequence and structure. As is the case for homodimers, accurate prediction of quaternary structure of heterodimer complexes can be very useful for understanding the function and physiological mechanism of action of heterodimers. Knowledge of 3D structures of proteins and their complexes is essential to understanding the molecular basis of their function.

Accurate \textit{in silico} structure prediction of heterodimers has been an important and long-standing challenge (Biasini \textit{et al.}, 2014). The Critical Assessment of Predicted Interactions initiative (CAPRI) has been focused since 2001 on the evaluation of different \textit{in silico} methods for the prediction of protein complexes, including heterodimers (Lensink \textit{et al.}, 2016; Wodak and Janin, 2017; Wodak, Velankar and Sternberg, 2020). At the time the work in this chapter was started, 47 CAPRI prediction rounds had been completed on a total of 160 targets (Wodak, Velankar and Sternberg, 2020). The evaluation meeting that occurred around the same time the work in this chapter was started (7\textsuperscript{th} CAPRI evaluation meeting (Wodak, Velankar and Sternberg, 2020) included 16 targets. Half of these targets were made up of protein-protein hetero-complexes (Wodak, Velankar and Sternberg, 2020).

Figure 5.1 illustrates structures for a number of heterodimer complexes with different structure and function. Heterodimers can have very diverse and complex interfaces, and accurate characterization of these interfaces can be useful to understanding the biology of complexes.
Figure 5.1 Examples of Heterodimer Complexes: a) Alpha-amylase to Alpha-amylase complex from *Bacillus licheniformis* (PDB: 1BPL) b) Calmodulin to calcium-activated potassium channel RSK2 complex from *Rattus norvegicus* (PDB: 1G4Y) c) cell division protein kinase 2 to Cyclin A2 complex (PDB: 1H1S) d) Cytochrome C peroxidase to cytochrome C complex from *Saccharomyces cerevisiae* e) Heat shock protein HSP82 to AHA1 in *Saccharomyces cerevisiae*. Images generated using ChimeraX (Pettersen et al., 2021).

Despite important progress that was demonstrated in the quality of quaternary structure predictions and in the ranking of docked decoys, most of the methods that were evaluated in the CAPRI rounds 38-45 relied on template information for the complex subunits, and sometimes templates for the complexes as a whole (Chakravarty et al., 2020; Dapku纳斯 et al., 2020; Kong et al., 2020; Padhorney et al., 2020; Wodak, Velankar and Sternberg, 2020). Overall, most of the methods assessed in the 7th CAPRI evaluation (CAPRI rounds 38-47) used a combination of template-based modelling and traditional *ab initio*, force-based docking prediction methods to generate structure predictions for the protein
complexes that were evaluated (Lensink et al., 2016; Lee et al., 2017; Dapkūnas et al., 2020; Kong et al., 2020; Padhory et al., 2020; Wodak, Velankar and Sternberg, 2020). These methods are greatly limited by the need for available template structures for the protein subunits, and in many cases, templates for the protein complexes. Unfortunately, templates are not always available for heterodimer complexes, and therefore these methods cannot be used to predict heterodimer complexes where crystal structures or close templates are not available.

The work outlined in this chapter covered the benchmarking of a number of in silico methods for the prediction of heterodimer complex structures. The goal of the experimental work outlined in this chapter was to benchmark various methods for the prediction of inter-chain heterodimer contacts, including CBMs, consensus methods and deep learning predictors. The benchmarking done in this chapter served to illustrate the usefulness of CBMs and deep learning for the prediction of inter-chain contacts. In addition, work in section 5.4 explored the use of inter-chain contact predictions for the purpose of ranking heterodimer docking decoys and to compare contact-based decoy ranking to previously published methods that use electrostatics and shape complementarity for decoy ranking.

At the time the work in this chapter was started, a number of studies had been published which used coevolution information from multiple sequence alignments to improve the prediction of protein monomer structures (Jones et al., 2012; Ekeberg et al., 2013; Kamisetty, Ovchinnikov and Baker, 2013; Kaján et al., 2014; Seemayer, Gruber and Söding, 2014). A number of additional publications had demonstrated the fact that these evolutionary statistical approaches, herein referred to as coevolution-based methods (CBMs), could be extended to the prediction of inter-chain contacts in protein complexes, including heterodimer complexes (Andreani, Faure and Guerois, 2013; Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014; dos Santos et al., 2015; Yu et al., 2016; Uguzzoni et al., 2017).

Unlike homodimers, heterodimer contact prediction with evolutionary statistical methods does not face the difficulty of differentiating between inter-chain and intra-chain contacts as the multiple sequence alignments for the two heterodimer chains are different. Applying CBM approaches to heterodimer inter-chain contact prediction, however, comes
with a different and non-trivial challenge: the generation of accurate paired multiple sequence alignments (Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014; Zhou, Wang and Xu, 2017). Previous studies had demonstrated that deep, accurate and diverse multiple sequence alignments are required for effective use of CBMs for protein contact prediction (Jones et al., 2012; Ekeberg et al., 2013; Kaján et al., 2014; Seemayer, Gruber and Söding, 2014; Jones and Kandathil, 2018). In the case of heterodimers, sequences in the MSAs of each constituent protein chain need to be paired in a way that maximizes the number of interacting homologs (i.e. interlogs) or the pairs or sequences that can be presumed to interact in real biological PPI networks (Zhou, Wang and Xu, 2017). Correct pairing of sequences in MSAs is essential to obtaining correct evolutionary histories and correct coevolution signals for the prediction of inter-chain residue contacts (Hopf et al., 2014).

In section 5.3, three MSA pairing algorithms were compared in terms of performance of CBMs on the paired alignments, as well as the depth of the paired sequences generated by each method. The purpose of comparing MSA pairing algorithms was to determine if there was one method that performed best and that could be used for analysis of the coevolution- and deep-learning based methods benchmarked in section 3.4.

In the years prior to the start of the work described in this chapter, a number of papers had been published exploring the use of CBMs for inter-chain contact prediction in heterodimer complexes (Weigt et al., 2009; Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014; Feinauer et al., 2016). Despite a number of these CBMs having been demonstrated to have promising application for the prediction of heterodimer complexes (Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014), only one method from the 7th CAPRI evaluation, InterEvDock from Andreani and Guerois (Yu et al., 2016; Nadaradjane et al., 2020), used interface coevolution to inform prediction of binding modes for targets in CAPRI rounds 38-45. InterEvDock was one of the best-performing methods in terms of ranking for protein-protein targets, and was the 5th ranked method in the 7th CAPRI evaluation and the second-best method with an openly accessible software or server.

In addition to coevolution-based methods, deep learning methods have shown significant improvements in the prediction of monomer protein structures (Wang et al., 2016; Zhou,
Wang and Xu, 2017; Jones and Kandathil, 2018; Greener, Kandathil and Jones, 2019; Xu and Wang, 2019; Senior et al., 2020; Jumper et al., 2021). At the time the work described in this chapter was started, one method, RaptorX ComplexContact, had been published that explored the application of deep learning methods to the prediction of heterodimer complex contacts (Zhou, Wang and Xu, 2017; Zeng et al., 2018). The method demonstrated better inter-protein contact prediction results than CBMs, and used a phylogeny-based method for the pairing of multiple sequence alignments for eukaryote sequences (Zhou, Wang and Xu, 2017). The method did not have an available code-base however and could not be evaluated as part of the benchmarking work done in this chapter.

Despite the promise of deep learning for protein structure prediction, only one deep learning method was demonstrated at the 7th CAPRI evaluation meeting (Cao and Shen, 2020). The method applied deep learning to the problem of scoring docking decoys models for the purpose of protein complex docking prediction (Cao and Shen, 2020). The method, presented by Cao and Shen, used an energy-based graph convolutional network to predict intra- and inter-molecular energies, binding affinities, and quality measures for scored docking decoys, but did not match the best scoring method, HADDOCK, across the 16 targets evaluated (Cao and Shen, 2020).

At the time the work described in this chapter was started, no comprehensive benchmark was available that compared the performance of the various recently-published coevolution-based methods and deep-learning methods for the prediction of inter-chain contacts and for the docking of heterodimer complexes. In section 3.4, a number of coevolution-based methods, PSICOV, GREMLIN-Complex and EVComplex, were compared to a voting consensus method and two deep-learning methods, DMPfold and DMPfold2 for the prediction of heterodimer inter-chain contacts.

Finally, to determine the impact of predicted contacts on the ranking of docked decoys, contacts from the best contact prediction method were used to rank docking decoys. Decoys were generated by docking two crystal structures of the heterodimer complexes, a crystal structure docked with a DMPfold generated structure, and two DMPfold generated structures. Ranking of docking decoys was done using a traditional
electrostatics and shape-complementarity approach and was compared to ranking done using contacts predicted using the best methods from section 5.4 (DMPfold2).

For MSA pairing, three previously published methods that had publicly available software were compared for the pairing of heterodimer MSAs: GREMLIN-stitch by Ovchinnikov et al., (Ovchinnikov, Kamisetty and Baker, 2014), EVComplex by Hopf et al. (Hopf et al., 2014), and the MI-IPA by Bitbol et al. (Bitbol et al., 2016; Bitbol, 2018).

In terms of inter-chain contact prediction, six methods were compared in terms of inter-chain contact prediction: GREMLIN, PSICOV, EVComplex, a consensus voting method which combined predictions from CBM methods, DMPfold (Greener, Kandathil and Jones, 2019), and DMPfold2 (Kandathil et al., 2022).

To benchmark the generation of docking decoys for heterodimer complexes, docking comparisons were done on six docking protocols which performed well in the 7th CAPRI evaluation, and which had publicly available servers or software. These protocols included InterEVDock (Yu et al., 2016), ZDock (Vreven et al., 2020), HADDOCK (Dominguez, Boelens and Bonvin, 2003), PyDockWeb (Jiménez-García, Pons and Fernández-Recio, 2013), and ClusPro (Padhorny et al., 2020). Each of these protocols was assessed on their ability to generate acceptable docked models in the top 1 and top 10 ranked lists, as well as in the overall list of generated decoys. Although each of these docking methods was evaluated in their respective publications, there was no single consistent, robust benchmark that compared the performance of each of the docking methods for the generation and ranking of docking decoys.
5.2 Benchmark Set for Heterodimer Prediction

Benchmarking

Before conducting an extensive and robust benchmarking of heterodimer complex structure prediction tools, a benchmark set of heterodimers had to be generated. This benchmarking set was the set of heterodimers that would be used for the evaluation of all the tools and servers explored in this chapter. To ensure the benchmarking would be as standardized, robust, and as generalized as possible, it was important to ensure that the heterodimers in the benchmark set represented a diverse or non-redundant and curated set of heterodimer complexes. Specifically, since it is possible for proteins to interact with more than one protein complex through one or more binding sites, it was important to ensure that the inter-protein interfaces represented in our benchmarking set were unique and non-redundant with each other.

To this end, the PiFace database from Cukuroglu et al. (Cukuroglu et al., 2014) of non-redundant and unique interfaces was used to generate the benchmarking set. The PiFace database consists of 22,604 unique interfaces curated from the PDB (Cukuroglu et al., 2014). These interfaces are made up of pairs or interacting protein chains, whose interface represents a unique binding mode as defined by the PiFace clustering method. To generate the unique set of protein pairs, Cukuroglu et al. extracted a set of all possible binary interactions of protein structures (Cukuroglu et al., 2014). Accessible surface area (ASA) of the interface pairs were calculated using NACCESS and all complexes with less than 1Å² ASA values were eliminated, and all interfaces in the PiFace set had at least five interacting residues (Cukuroglu et al., 2014). MultiProt (Shatsky, Nussinov and Wolfson, 2004), which performs structural alignment of multiple protein chains regardless of the order of residues, was used to calculate the pairwise similarity of all interfaces in the dataset (Cukuroglu et al., 2014).

The interface similarity of complexes was determined by calculating the number of structurally matched residues in the interfaces being compared, with a threshold of 3Å RMSD used to differentiate structurally dissimilar interfaces. These structural similarity scores were used to construct a network of interfaces, with interfaces as nodes and interface similarity as edges in the nodes (Cukuroglu et al., 2014). Finally, interfaces in
the network were clustered based on network properties including edge and node betweenness using a community finding algorithm (Cukuroglu et al., 2014). The clustering of interfaces resulted in a set of 22,604 unique interfaces made up of both hetero- and homo-dimer interfaces.

To generate the benchmark set from the PiFace dataset that could be used for the benchmarking work in this chapter, heterodimer interfaces were extracted from the dataset. This was done by extracting the protein dimer complexes for each interface and checking the sequence and structure similarity of the interacting proteins to one another. In this chapter, heterodimers were defined as having protein chains with less than 95% sequence similarity and RMSD greater than 1.5Å to one another. Finally, all complexes that had at least one protein chain with a protein length less than 30 residues were removed from the set of heterodimer interfaces. The resulting set was made up of 4,887 heterodimer interfaces.

At the start of the work completed in this chapter, the intention was to benchmark the performance of DMPfold generated contacts and DMPfold generated structures for heterodimer quaternary structure prediction. For this reason, it was important to ensure that the protein chains involved in the heterodimers being evaluated did not overlap with the DMPfold training set. To do this, all complexes with at least one chain that had greater than 40% sequence similarity to the DMPfold training set were excluded from the benchmarking set. In addition, the ECODanalyse method previously described in Chapter 2 was used to remove all complexes with at least one chain that overlapped in the ECODT (topology) class.

Since the PiFace dataset was released in 2014, and the DMPfold training set was created in 2017, there was a fairly large overlap between the protein chains in the PiFace dataset and the DMPfold training set. After filtering the PiFace heterodimer interfaces against the DMPfold training set, 78 heterodimer complexes were left in the benchmark set. This set of 78 non-redundant heterodimer complexes was the set of heterodimer complexes used for evaluating tools and servers evaluated in this chapter.

It is important to note that, by the nature of the interfaces captured in the PiFace dataset, it was not guaranteed that the heterodimer complexes used for benchmarking in this
chapter were not part of bigger protein complexes in the PDB. Analysis of the biological assembly files for the 78 complexes evaluated in this chapter showed that 32 of the complexes had biological assemblies that were indeed heterodimers, and the remaining 46 complexes were part of higher order protein complexes.

As the goal of the prediction methods in this chapter is to correctly predict the interactions between two different interacting proteins, regardless of their involvement in higher-order protein complexes, both kinds of heterodimer interfaces were kept in the benchmarking set.

### 5.3 Multiple Sequence Alignment Pairing for Heterodimer Complexes

One of the primary goals of the benchmarking work in this chapter was to examine the ability of direct coupling analysis (DCA) and other evolutionary statistical methods to predict inter-chain contacts in heterodimer complexes. To do this, it was important to obtain paired multiple sequence alignments for each of the heterodimer complexes in the benchmark set.

This is an important step in applying coevolution-based methods to inter-chain contact prediction of hetero-oligomeric complexes (Figure 5.2). Correctly paired MSAs maintain the paired evolutionary history of interacting proteins in a heterodimer complex. It is therefore necessary to capture as many sequence pairs that interact in real biological protein-protein networks as possible. Most heterodimer complexes will have protein chain MSAs of different lengths, and often different numbers of paralogs and/or homologs.
Figure 5.2: The MSA pairing problem: MSA pairing involves correct identification of interacting protein homologs from MSAs of interacting proteins.

When using CBMs to predict interacting residues in heterodimer complexes, it is also useful to obtain the deepest possible paired MSAs. Previous publications have shown that CBMs tend to perform better on more accurate, deep, and diverse MSAs. For this reason, the goal of MSA pairing algorithms is accurate predict paired sequences while maximizing the number of successfully paired sequences to improve CBM prediction of inter-chain contacts.

For this reason, benchmarking in this section compared three published MSA pairing methods which had publicly available servers or downloadable software and that could deal with MSAs of different lengths. These methods were the GREMLIN-stitch method (Ovchinnikov, Kamisetty and Baker, 2014), and the MSA pairing method included in the EVCouplings complex prediction server (Hopf et al., 2014), and a modified version of the MI-IPA sequence pairing method developed by Anne-Florence Bitbol (Bitbol, 2018).

Unfortunately, it was not possible to evaluate each of these methods on a particular gold-standard paired MSA dataset for a number of reasons. At the time, no large gold-standard
dataset of known MSA pairings existed that could be used for benchmarking. It was also
difficult to create such a paired dataset for benchmarking the three methods compared in
this section as each method uses different assumptions, alignment databases, and
pairing methods to generate alignments. Finally, the effect of sequence pairing on CBM
contact prediction is dependent on both depth and accuracy of paired MSAs. It is difficult
to determine whether paired MSAs are better for CBM contact prediction based on pairing
accuracy or sequence depth alone.

The overall purpose of MSA pairing is to improve the ability of CBM methods to pick up
on coevolutionary signals of inter-chain contact residues. For this reason, it was possible
to use the precision of CBM-predicted inter-chain contacts as a proxy for the quality of
MSA pairings. Two coevolution-based methods, GREMLIN and PSICOV, were used to
get inter-chain contact predictions from paired MSAs generated with each pairing method.
The precision and accuracy of the top 20 inter-chain contacts predicted for each method
was used as a proxy of the pairing performance of each pairing method.

5.3.1 Methodology

5.3.1.1 MSA Pairing Methods

GREMLIN-stitch

The first method that was evaluated for the pairing of multiple sequence alignments was
the GREMLIN-stitch method included in the GREMLIN complex contact prediction server

To get MSA pairing predictions, protein chain alignments were obtained by running
HHblits using 8 iterations, an E-value threshold of 1E-20, a Neff max of 20. The HHblits
results were filtered using HHfilter, by filtering out sequences with a sequence id of 100,
and a minimum query coverage of 75%.

Once the alignments for each protein chain were obtained, paired MSAs were obtained
using the stitch.pl script from the GREMLIN complex server. This was done using the
default settings for stitch.pl using the uni2loc lookup table and a Δgene range between 1 and 20 (-min 1 -max 20). This MSA pairing method was run over the 78 complexes in the benchmarking set, and 60 complexes had output paired MSAs with MSA depth greater than 30 sequences, which was the minimum number of sequences needed to run the coevolution-based methods.

EVComplex Pairing

The next method that was benchmarked for MSA pairing was the EVComplex pairing method (Hopf et al., 2014). Since this pairing could be obtained from the EVCouplings web server [https://v1.evcouplings.org/complex], these pairings were obtained by running each of the heterodimer complexes in the benchmarking set through the EVComplex server.

The webserver was run using the default cut-offs from the webserver. EVComplex generates MSAs using the jackhmmer method (Johnson, Eddy and Portugaly, 2010) to search the UniProt 2014_02 database with 5 iterations (Hopf et al., 2014). A bit score threshold of 0.5 * monomer sequence length is used to obtain alignments of consistent evolutionary depths (Hopf et al., 2014).

Out of the 78 heterodimers run through the EVComplex server, 56 complexes were successfully paired and had paired MSA depths greater than 30 sequences. This was the depth cut-off that was used for contact prediction with CBMs.

MI-IPA Pairing

The third MSA pairing method evaluated was a modified version of the mutual information iterative pairing algorithm (MI-IPA) developed by Bitbol (Bitbol, 2018). This method was a variation of a previously published IPA method (Bitbol et al., 2016), which used effective interaction energy from a DCA model to pair sequences. MI-IPA instead uses the mutual information (MI) of paired MSAs as the objective function to be maximized (Bitbol, 2018). While both MI-IPA and DCA-IPA had been shown to perform fairly well of sequence
pairing for inter-chain contact prediction, MI-IPA was shown to perform faster and better MSA pairing and resulted in more accurate inter-chain contact predictions (Bitbol, 2018).

The original MI-IPA method was available as a MATLAB codebase that was available online [https://github.com/anneflo/MI_IPA/tree/v1.0]. While the method could be used to perform the sequence pairing for the complexes in the heterodimer benchmark set, there were a number of limitations in the original method.

First, the original method relied on having multiple sequence alignments of equal length, i.e., required the MSAs to have the same number of sequences for each species for both proteins being paired. This was not the case for any of the complexes in our benchmark set, which usually had proteins with MSAs of different length, and often different numbers of paralogs within the same species between the two protein MSAs. It was therefore necessary for the pairing method to be able to deal with MSAs with different numbers of paralogs between protein pairs.

Second, while the example alignments that were provided with the codebase had an approximate depth of 5000 sequences, a number of the alignments for some of the protein chains in the benchmark set were considerably deeper. Out of the 78 complexes in the benchmark set, 25 had at least one protein chain with an MSA sequence depth greater than 10k. As several pairs of scores had to be calculated, calculating paired MSAs turned out to be very computationally expensive for deeper alignments.

For this reason, a modified version of the MI-IPA code was created in Python, that worked exactly the same as the original MI-IPA code with a few changes. One of the limitations of the Hungarian algorithm (HA) is the fact it works on square matrices, which was problematic in cases where unequal numbers of species paralogs were present in the two MSAs being paired. It was possible, however, to get the Hungarian algorithm to work on rectangular matrices resulting from unequal numbers of paralogs in species by simply padding the rows or columns of the matrix with zeros to form square matrices. The first modification made to the original algorithm therefore involved tweaking the code to pad the pairwise matrices to turn rectangular matrices representing MSAs of unequal depth into square matrices. Since the goal of the HA was to maximize the sum of the sequence pairs selected from the matrix, these zero-value rows or columns are simply ignored by
the algorithm. The number of pairs that was added to the concatenated alignment for each species was equal to the minimum number of paralogs between the two protein chains.

The second modification involved changing the initial concatenated alignment used to calculate the mutual information and pairing scores in the first iteration. In the original method, the first iteration uses a random pairing of sequences to calculate the mutual information used to generate confidence scores. In the modified version, all sequences that were singletons (one sequence per species for both protein alignments) were paired and concatenated. This step significantly reduced the amount of time the MI-IPA algorithm took to finish pairing of MSA sequences.

To ensure this modification did not affect the accuracy of the sequence pairing, the modified version was run on the original HK-RR sequence example included in the MI-IPA source code. The modified MI-IPA version was able to reproduce the results from the original method, with a 96% overlap in sequence pairings.

The alignments that were fed into MI-IPA were generated running HHblits on the Uniclust30_2018_08 alignment database with 3 iterations, and an E-value threshold of 0.01, and a minimum coverage threshold of 30%. The MI-IPA method was then used to pair alignments for the 78 benchmark complexes. Paired alignments with depth of at least 30 sequences were obtained for 72 of the complexes. This modified version of MI-IPA was used to generate MSA pairings that could be fed into the CBM tools to obtain inter-chain contact prediction.

5.3.1.2 Direct Coupling Analysis Methods for Evaluation of Paired Alignments

To evaluate the MSA pairing by the pairing methods benchmarked the alignments were concatenated and fed into two direct coupling analysis methods for which software was available: GREMLIN and PSICOV.
To run GREMLIN on the paired alignments, the run_gremlin.sh script obtained from the GREMLIN complex server [http://gremlin.bakerlab.org/cplx_faq.php] was run using a maximum iteration (MaxIter) of 30, and an All-Product Correction (APC) of 0. This was followed by the matrix to score method that was run with a specialized All-Product Correction adopted for inter-protein contact prediction that accounts for differing intra-chain coevolution rates as previously described by Ovchinnikov et al (Ovchinnikov, Kamisetty and Baker, 2014).

The PSICOV software was obtained from the PSIPRED GitHub page [https://github.com/psipred/psicov]. PSICOV was run using the command-line ‘psicov’ command with default rho and estimate prediction, with separation of 12 or more residues.

These CBM methods output contact predictions (or contact probabilities) of all pairs of residues with a sequence separation greater than 12 residues. Inter chain contact predictions were extracted by filtering out all intra-chain residue pairs and ensuring that all remaining residue pairs were on different heterodimer chains.

The remaining inter-chain contact predictions were ranked by the output probabilities or contact prediction scores. The precision of the top 20 predicted inter-chain contacts was extracted for PSICOV and GREMLIN. As the goal of benchmarking was simply to determine whether particular pairing methods resulted in better inter-chain contact prediction performance, this top 20 contact precision could be used as a proxy of the effectiveness of sequence pairing methods.
5.3.2 Results

Paired alignments were run through GREMLIN and PSICOV to extract inter-chain contact predictions. The top 20 ranked inter-chain residue pairs for each method were extracted. The average precisions across the complexes successfully run of these top 20 ranked inter-chain contacts for each alignment pairing method run through each CBM contact prediction method is shown in table 5.1.

54 homodimer complexes had paired MSAs of sufficient depth for all three pairing methods. Table 5-1 shows the precision and recall of each of the pairing methods on these 54 complexes. The performance of PSICOV and GREMLIN Complex for inter-chain contact prediction using paired alignments from each MSA pairing method are shown in table 5.1.

<table>
<thead>
<tr>
<th>Coevolution based method</th>
<th>Top 20 Contact Precision</th>
<th>Top 20 Contact Recall</th>
<th>Number of complexes paired</th>
</tr>
</thead>
<tbody>
<tr>
<td>GREMLIN-stitch GREMLIN Complex</td>
<td>0.1857</td>
<td>0.0082</td>
<td>54</td>
</tr>
<tr>
<td>GREMLIN-stitch PSICOV</td>
<td>0.1762</td>
<td>0.0075</td>
<td>54</td>
</tr>
<tr>
<td>EVComplex pairing GREMLIN Complex</td>
<td>0.2119</td>
<td>0.0088</td>
<td>54</td>
</tr>
<tr>
<td>EVComplex pairing PSICOV</td>
<td>0.1928</td>
<td>0.0081</td>
<td>54</td>
</tr>
<tr>
<td>MI-IPA GREMLIN Complex</td>
<td>0.1952</td>
<td>0.0073</td>
<td>54</td>
</tr>
<tr>
<td>MI-IPA PSICOV</td>
<td>0.1809</td>
<td>0.0078</td>
<td>54</td>
</tr>
</tbody>
</table>

Table 5-1: MSA Pairing Impact on DCA Contact Prediction (overlapping complexes): The precision and recall for the top 20 predicted inter-chain contact residue pairs from GREMLIN Complex and PSICOV for heterodimer benchmark complexes with paired MSAs across all pairing methods

As can be seen from table 5.1, unfortunately the recall for the contact prediction using PSICOV and GREMLIN Complex was low for all three pairing methods. The total number of inter-chain contacts across the 54 complexes was 2732, which meant an average of
50.59 contacts per complex. Overall, for the method with the highest precision and recall, 24 contacts were correctly predicted. This corresponded to 1 complex with 4 correctly predicted inter-chain contacts, 2 complexes with 3 correctly predicted contacts, 4 complexes with 2 correctly predicted contacts, and 6 complexes with 1 correctly predicted inter-chain contact. These low recall figures, also seen in the results from chapters 2 and 3, demonstrate the overall difficulty of correctly predicting interacting residues in protein complexes.

To determine whether differences of precision and recall values between the MSA pairing methods were statistically significant, a paired t-test was used to compare the precision and recall values between the three MSA pairing methods.

<table>
<thead>
<tr>
<th>MSA Pairing Comparison</th>
<th>Top 20 Contact Precision</th>
<th>Top 20 Contact Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVComplex vs. GREMLIN-stitch</td>
<td>0.1405</td>
<td>0.2075</td>
</tr>
<tr>
<td>EVComplex vs. MI-IPA</td>
<td>0.1059</td>
<td>0.3743</td>
</tr>
<tr>
<td>MI-IPA vs. GREMLIN-stitch</td>
<td>0.2075</td>
<td>0.7048</td>
</tr>
</tbody>
</table>

**Table 5-2: Paired t-test p-values for the MSA pairing methods:** p-values of CBM precision and recall for each MSA pairing method compared to the other two MSA pairing methods.

As can be seen from Table 5-2, the differences between the MSA-pairing methods were not statistically significant. The method with the deepest paired alignments was therefore used for further analysis. The EVComplex pairing alignments had an average pairing depth of 11,817 (min: 219, max: 50,950) while MI-IPA alignments had an average pairing depth of 6727 (min: 68, max: 28,105) and GREMLIN-stitch alignments had an average pairing depth of 4,302 (min: 42, max: 22,563). The paired alignments generated using the EVComplex method were deeper overall than those generated using GREMLIN-stitch or MI-IPA. EVComplex pairing was therefore used for further benchmarking of heterodimer contact prediction methods in section 5.4.
5.4 Inter-Protein Contact Prediction for Heterodimer Complexes

Protein-protein interface prediction is an important part of the structural characterization of heterodimer complexes. Having a reliable structural characterization of the interface between heterodimer complexes can give us insight into the orientation of interacting proteins, binding mechanisms of the proteins, the properties of inter-protein interfaces, and insight into residues that may be important to stabilising the complex or understanding biological function of heterodimer interactions.

Overall, most existing inter-chain docking methods predicted correct complex docking for less than 20% of heterodimer complexes. Only one existing docking method, InterEvDock, used coevolution information to perform docking decoy ranking. None of the methods that had been benchmarked at the time used deep-learning contacts or distance prediction to perform docking or ranking of protein-protein complexes.

Previous work (de Vries and Bonvin, 2011; Yu et al., 2016) had demonstrated that it is possible to use inter-chain contact information to filter or rank docking decoys. To do this, the overlap between expected inter-chain contacts and contacts present in the docking decoys can be used as a metric to rank docking decoys. Further, methods such as HADDOCK (Bonvin et al., 2018) can take information such as residues or contacts expected to be involved in the interface (i.e. residue restraints) of a complex to be docked as inputs to guide the docking process.

Inter-chain contacts can therefore be useful to generate better docking decoys and to rank generated decoys based on which are most likely to be closer to the structure (i.e., the orientation) of the native complex. Although a number of papers had been published that demonstrated direct coupling analysis methods could be applied to predict inter-chain contacts in heterodimer complexes (Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014), these methods had not been comprehensively benchmarked on a single set of heterodimer complexes. It may be possible to improve the ranking of docking decoys using improved inter-chain contact predictions.
The work in this section (5.4) therefore explored the application of 6 different protein contact prediction methods for the task of inter-chain contact prediction: GREMLIN Complex (Ovchinnikov, Kamisetty and Baker, 2014), EVComplex (Hopf et al., 2014), PSICOV (Jones et al., 2012), a consensus voting method, DMPfold (Greener, Kandathil and Jones, 2019), and DMPfold2 (Kandathil et al., 2022). It should be noted that the code for DMPfold2 was made available after the work in this section was started and was therefore added to the suite of tools being evaluated a bit later than the first five methods.

DMPfold had been shown in previous CASP rounds to have a good performance for inter-chain contact and distance prediction (Greener, Kandathil and Jones, 2019). Since the code was openly available, and could be run on custom multiple sequence alignments, it could be used to generate predictions for inter-chain contacts. A newer version of DMPfold, DMPfold2, was developed and released after the work in this section was started. DMPfold2 had been shown to have a better performance than DMPfold in most cases in structure prediction of protein monomers (Kandathil et al., 2022). It should be noted that although previously published work had shown that RaptorX-ComplexContact (Zeng et al., 2018) had good performance in terms of heterodimer contact prediction, it was not possible to include it in the benchmarking work here as the code for the method could not be obtained and it took a very long time (between 3 and 6 weeks) to run complex predictions through the server due to large server backlogs.

Overall, the aim of the work completed in this section was to compare the inter-chain contact prediction performance of a number of previously published coevolution-based methods to a consensus voting method and two deep-learning methods. It is hypothesized that higher contact precision and recall of inter-chain contact predictions result in better ranking of docking decoys generated from docking protocols. The deep learning methods had been developed for the prediction of intra-chain predictions, but in this section, the goal was to determine whether they could be extended to the prediction of heterodimer interface contacts.
5.4.1 Methodology

5.4.1.1 Heterodimer MSA Pairing

Before the contact prediction methods explored in this section could be applied to the heterodimer complexes, the multiple sequence alignments had to be paired to optimize the coevolution signals in the paired MSAs. As the EVComplex (Hopf et al., 2014) pairing method had previously been determined to be the best and most effective sequence pairing method, the MSA pairings for each heterodimer complex was extracted from the EVComplex web server [https://v1.evcouplings.org/complex]. Since the EVComplex server had successfully generated paired MSAs of sufficient depth (30 paired sequences) for 56 heterodimer complexes, the benchmarking of inter-chain contact prediction methods was done on these 56 heterodimer complexes.

5.4.1.2 Contact Maps for Benchmark Heterodimer Complexes

As the goal was to predict the inter-chain contacts in the heterodimer complexes, the complex contact information had to be extracted for the heterodimers in the benchmark set. To do this, a rectangular matrix with contact information of the interacting chains was extracted for each complex. All residue pairs with Cβ-Cβ atoms (Cα in the case of glycine) less than 10Å RMSD apart were given a value of 1 while all other residue pairs were given a value of 0. These rectangular contact maps were extracted for all 56 heterodimer complexes with sufficiently deep paired multiple sequence alignments. These contact maps were used to determine the performance of contact prediction methods by calculating the overlap between the top 5, 10, and 15 predicted residue contacts and the set of true contacts in the contact maps.

Figure 5.3 shows the interface contact maps and complex structures for a few of the protein complexes in the Heterodimer Benchmark Set used in this chapter.
Figure 5.3: Examples of heterodimer complexes in the Heterodimer Benchmark Set: Complex structures and inter-chain contact maps for 4 heterodimer complexes in the heterodimer benchmark set used for benchmarking in this chapter.
5.4.1.3 Inter-Protein Contact Prediction for Benchmark Heterodimer Complexes

The contact prediction methods compared in this section included GREMLIN-complex (Ovchinnikov, Kamisetty and Baker, 2014), EVComplex (Hopf et al., 2014), PSICOV, a consensus voting method, DMPfold (Greener, Kandathil and Jones, 2019), and DMPfold2 (Kandathil et al., 2022). For each of these methods, the paired MSAs from EVComplex pairing were used, with a linker of 20 glycine residues used to link paired sequences.

GREMLIN-complex was run using the run_gremlin.sh and mtx2sco.pl scripts downloaded from the GREMLIN website [http://gremlin.bakerlab.org/cplx_faq.php]. GREMLIN (Generative REgularized ModeLs of proteINs) is a maximum-entropy or global statistical model that uses a Markov Random Field with log-linear potentials to generate contact predictions (Kamisetty, Ovchinnikov and Baker, 2013).

To generate pairwise contacts probabilities, the run_gremlin.sh script was first run on the paired multiple sequence alignments for each complex using a maximum iteration (MaxIter) of 30, and an All-Product Correction (APC) of 0. This generated a raw matrix of contact predictions that was then fed into the mtx2sco.pl script with information about the length of the first protein in the alignment, and information about the number of sequences per length. The output of mtx2sco was a corrected APCD matrix which scored the complex residue pairs with a specialised APC to correct for differing rates of intra-chain coevolution for the two proteins in the complex.

EVComplex was run by putting the two fasta sequences for each chain of the heterodimer complexes through the EVComplex server (Hopf et al., 2014). EVComplex uses an underlying statistical model similar to EVCouplings (Marks et al., 2011), which uses a maximum entropy model of protein sequences to identify interacting residues. The server was run using the same method described in section 3.3, with a bit score threshold of 0.5*monomer sequence length run using jackhmmer on the UniProt 2014 database with 5 iterations (Hopf et al., 2014). Predictions of inter-chain contacts were summarized in an output file, *_CouplingsScores.csv, which contained the residue pair contact predictions for the complex in order of descending probability of the residues being in contact as
determined by EVComplex. To get the inter-chain contact predictions, the list was filtered to only contain contacts between two different interacting chains (i.e., filtered by the fields ‘segment_i’ and ‘segment_j’).

PSICOV works slightly differently from DCA methods and uses sparse inverse covariance estimation to predict interacting residues (Jones et al., 2012). PSICOV was run on paired MSAs as previously described in section 3.3, with default rho and estimate prediction, and a minimum sequence separation of 12 residues. Output residue pair predictions were filtered to ensure they were on separate heterodimer chains and were ranked by probability of being in contact as determined by PSICOV.

For each of the coevolution-based methods described above, the top 5, 10, and 15 contacts ranked by probability were extracted for each heterodimer complex.

The fourth method tested for the prediction of inter-chain contact prediction was a consensus voting method that combined the predictions from GREMLIN, EVComplex and PSICOV. Previous work (Jones et al., 2015) had demonstrated that combining predictions from different coevolution based methods could improve the quality of predicted contacts. To obtain the consensus predictions, contacts which were common between at least two of the three lists of the top 5, 10, and 15 ranked contacts for the three CBMs were extracted.

Finally, two deep learning methods, DMPfold and DMPfold2, which had both been shown to perform better than pure coevolution-based methods for intra-chain residue contact and distance prediction were used to predict inter-chain contacts. The code for DMPfold was obtained from the GitHub repository [https://github.com/psipred/DMPfold]. Paired alignments generated by EVComplex pairing with glycine linkers were fed into DMPfold and were used to generate DMPfold structures.

The DMPfold2 code was also obtained from the GitHub repository [https://github.com/psipred/DMPfold2]. The DMPfold2 method performs very fast (few hundred milliseconds) tertiary structure prediction directly from sequence alignments (Kandathil et al., 2022). The method uses learned MSA representations and end-to-end modelling to produce predicted structures from MSA alignments and was demonstrated
to have equivalent or higher accuracy that DMPfold for tertiary protein structure prediction (Kandathil et al., 2022).

DMPfold2 uses a bidirectional gated recurrent unit (biGRU) to embed the MSAs in the vertical and horizontal directions (Kandathil et al., 2022). These embeddings are combined with precision matrices containing the pairwise inverse covariance of 20 amino acids plus gaps for all positions in a protein sequence (441xLxL where L corresponds to the length of the protein). These combined inputs are then run through a residual net followed by a multidimensional scaling, another GRU layer, then a linear layer to get Cα coordinates, after which mainchain and Cβ coordinates are added. Similar to DMPfold, structures were generated by DMPfold2 by providing the same paired alignments with glycine linkers for each benchmark complex.

For both DMPfold and DMPfold2, inter-chain residue distance predictions were extracted from predicted structures by extracting the pairwise Cβ atom distances for residue pairs. Inter-chain distances were obtained by limiting the residue pair information to pairs on different chains of the heterodimer complex. Residue contacts ordering by predicted distances and the top 5, 10, and 15 contacts were extracted. Inter-chain residue contacts were obtained by extracting all residues with Cβ atom distances less than 10Å.

5.4.2 Results

The precision and recall of inter-chain protein lists extracted from all six methods were calculated by getting the overlap between the top 5, 10, and 15 predicted contacts and the true contacts in the native heterodimer complexes. The precision and recall for the contact prediction methods evaluated in this section are shown in Table 5-3 and Figure 5.4.
Table 5-3: Inter-chain contact prediction on benchmarking dataset: The precision and recall for the top 5, 10, and 15 predicted inter-chain contact residue pairs for six benchmarked methods tested on 66 heterodimer complexes in the benchmarking set.

Figure 5.4: Inter-chain contact prediction Precision for: (a) Precision and (b) recall of top 5, 10, and 15 predicted inter-chain contacts from six contact prediction methods on 55 complexes in the heterodimer benchmark set.

Overall, DMPfold2 had the best precision and recall of the methods benchmarked for heterodimer inter-chain protein prediction.
DMPfold2 both performed better than coevolution-based methods and DMPfold in terms of both precision and recall. This could be due to a number of reasons. DMPfold2 had previously been shown to have significantly higher intra-protein structure prediction accuracy than CBM methods and DMPfold. This was due to the use of powerful MSA embedding techniques combined with covariance information, deep residual neural networks and a structure module that greatly improved the prediction of monomer atom coordinates. Results in chapter 3 and chapter 4 of this thesis demonstrated how increasing quality of monomer structure predictions can contribute to better inter-chain contact predictions for complexes. Although that was previously shown to be the case in homodimers, it is possible that heterodimers also benefit from better monomer structure predictions. The results in this section demonstrate a significant increase in inter-chain contact prediction using deep learning and end-to-end structure prediction over traditional coevolution-based methods.

Contacts predicted by DMPfold2 were used for ranking docking decoys in section 5.5

5.5. Heterodimer Complex Docking and Ranking

Complex docking methods are designed to predict the quaternary structures of heterodimer complexes by predicting the relative orientation of constituent monomers and in some cases, the bound conformation of monomers. Docking methods can be provided with crystal structures of monomer structures or predicted monomer structures generated using templated-based methods or de novo structure prediction methods.

In this section, the prediction of quaternary structures of heterodimer complexes using docking methods was benchmarked by exploring the docking of monomer crystal structures as well as the docking of monomer structures generated using DMPfold. Six docking methods were selected for benchmarking based their performance in the 7th CAPRI evaluation (Wodak, Velankar and Sternberg, 2020) and the availability of either downloadable software or web servers that could be used to generate predictions of heterodimer docking orientations. Unfortunately, only 8 hetero-oligomer complexes in total were evaluated during the 7th CAPRI evaluation, and not all evaluated methods were
evaluated on all 8 hetero-oligomer complexes. The goal of this section was therefore to comprehensively benchmark a number of docking methods on the benchmark established here, and to explore the possibility of using predicted inter-protein contacts for docked decoy ranking.

Here, five docking methods, ZDock (Vreven et al., 2020), HADDOCK (Koukos et al., 2020), ClusPro (Padhony et al., 2020), pyDock (Jiménez-García, Pons and Fernández-Recio, 2013) and InterEvDock (Yu et al., 2016) were benchmarked for heterodimer complex decoy generation and ranking of decoys. To measure the quality of docking decoys, DockQ (Basu and Wallner, 2016), a quality measure of protein-protein docking models was used to measure the quality of docked models when compared to native heterodimer complex structures. Structures with a DockQ score above 0.23 are considered to have conformations close enough to the native complex structure to be considered ‘acceptable’ (Basu and Wallner, 2016).

InterEvDock had performed the best in terms of number of acceptable or better models in the top predicted and top 10 predicted docked decoys, the method only outputs 10 docking decoys. ZDock, however, could be used to generate up to 2000 docking decoys for each docking complex and included a much greater proportion of acceptable models in the set of 2000 generated decoys.

ZDock was therefore used to generate 2000 docking decoys for each of the heterodimer complexes. To determine the impact of using inter-chain contact predictions to rank docked models, the top 15 ranked inter-chain contacts predicted by DMPfold2 for each heterodimer complex was used to rank the decoys generated by ZDock. The top 1 and 10 ranked decoys were then evaluated using DockQ to determine the number of complexes with at least one model with an acceptable or better DockQ score. These contact-based decoy rankings were compared to ZDock ranking using the internal ZRank (Vreven et al., 2020) method.

In addition, three types of docking were performed with ZDock: docking with two crystal structures, docking of a crystal structure to a DMPfold generated structure, and the docking of two DMPfold generated structures. 2000 docking decoys were generated for
each docking type and predicted inter-chain contacts were used to rank the docking decoys.

Overall, the work in this section tested the ability of established docking protocols to correctly dock crystal and DMPfold generated structures and the ability of contacts predicted using DMPfold2 to correctly pick docking decoys that are closer native complex structures.

5.5.1 Methodology

5.5.1.1 Docking of Heterodimer Complexes Using Crystal Structures

The first step of benchmarking heterodimer complex docking was running the five docking tools explored herein on the benchmark set of 56 complexes which had paired MSAs and predicted inter-chain contacts from DMPfold2.

Each of the complexes was run through the methods tested here. In this section, the crystal structures of each monomer of the complex (in these cases, the bound form of the monomer randomly translated and rotated around the x, y, z axes) were used for docking. The internal scoring and ranking methods for each tool or server tested were used to extract the top 1 or 10 predicted docking decoys.

To obtain ranked docking decoys using ZDock, the two crystal structures for the constituent monomers of each complex were run through the ZDock server. The CreateComplexes tool was downloaded from the ZDock website [https://www.zdock.umassmed.edu] and was used to extract the top 2000 generated decoys for each complex.

PyDock predictions were obtained by providing the monomer crystal structures to the PyDockWeb server, which outputs the 10 top ranked decoys for each complex.

The ClusPro webserver was used to generate 1000 docking decoys for each complex.
The Haddock2 webserver (van Zundert et al., 2016, p. 2) was used to generate ranked docking decoys using two sets of restraints: residues predicted by the CPORT (de Vries and Bonvin, 2011) server and residues predicted to be involved in the interfaces using DMPfold2 contacts predicted in section 3.4. CPORT is a consensus-based method that combines predictions from six interface prediction web servers to generate predictions of which residues in complexes are involved in protein-protein interfaces (de Vries and Bonvin, 2011). It had been previously shown that CPORT gave more reliable predictions of residues involved in protein interfaces than any of the prediction methods on their own (de Vries and Bonvin, 2011).

HADDOCK performed clustering of docking poses and outputs between 5 and 38 docking decoys for each complex.

The InterEvDock web server was used to generate docking predictions and output 10 docking decoys for each of complex.

For each docking method, the top 1, top 10 and where applicable, the larger docking decoy lists were assessed for docking quality. DockQ was used to assess the quality of the docked models (Basu and Wallner, 2016). DockQ is a continuous score that combines metrics in CAPRI including Fnat (fraction of native contacts included in the docked model), the iRMSD (interface RMSD) and the lRMS (ligand RMSD) (Basu and Wallner, 2016). A DockQ scores between 0.23 and 0.49 corresponds to acceptable quality quaternary predictions, between 0.49 and 0.8 correspond to medium quality predictions, and above 0.8 correspond to high quality docking predictions (Basu and Wallner, 2016).

The number of complexes with at least one acceptable model in the top 1 and top 10 ranked docking predictions was extracted and is shown in the results section below.

5.5.1.2 Predicted DMPfold2 Inter-Chain Contacts for Decoy Ranking

Although the InterEvDock method had the best performance in terms of inclusion of an acceptable model in the top 1 and top 10 ranked models for the benchmark set, ZDock
had a higher inclusion rate of acceptable models in the set of 2000 ZDock generated decoys.

The analysis conducted in this section compared the ranking of 2000 ZDock decoys using the ZRank method to the ranking of decoys using predicted inter-protein contacts from DMPfold2. To do this, the top 15 predicted inter-chain contacts (ranked by contact probability) from DMPfold2 predictions were used. ZDock decoys were then ranked by the overlap between the contacts in the decoys and the predicted DMPfold2 contact list.

The inclusion of an acceptable or better model in the top 1 and top 10 ranked docking decoys was determined and was compared to the performance of ZRank ranking of ZDock decoys.

5.5.1.2 Docking of Crystal Structures and DMPfold generated structures

To determine the ability to do completely de novo prediction of heterodimer complexes, including the constituent monomer structures, docking of DMPfold structures with ZDock was assessed. Since the code for DMPfold2 was developed after the work in this benchmarking assessment was started and the benchmark set had been established, the DMPfold2 training set had structure overlaps in terms of ECOD topology with one or both protein chains in 47 of the 55 benchmark complexes. For this reason, the structures used for docking assessments were generated using DMPfold.

ZDock was used to generate decoys for three cases of docked monomer structures:

- Crystal structures for each monomer chain
- One crystal structure docked to a DMPfold generated structure
- DMPfold generated structures for each monomer chain

2000 decoys were generated for each docking category, and ZRank and DMPfold2 contacts were both used to rank the ZDock docking decoys for decoys generated for each docking case above.
Finally, ZDock docking of DMPfold structures using predicted contacts (i.e., completely \textit{de novo} heterodimer prediction) was compared to DMPfold2. To do this the top ranked docked models generated using ZDock docking was compared to DMPfold2 prediction of the heterodimer complexes. DMPfold2 models were generated for the complexes by running DMPfold2 on the paired alignments for each of the complexes. These DMPfold2 generated complexes were compared to ZDock to determine the ability of the methods to perform completely \textit{de novo} prediction of heterodimer complexes.

5.5.2 Results

5.5.2.1 HADDOCK Docking: CPORT vs. DMPfold 2 contacts

For each of the complexes in the benchmark dataset, HADDOCK docking was performed using two different sets of residue restraints (i.e., residues involved in interface contacts). First, residues obtained from the CPORT server were used for heterodimer docking of monomer crystal structures. Second, residues obtained from the DMPfold2 predictions as outlined in section 3.4 were used for docking of monomer crystal structures.

Table 5-4 shows the number of complexes in the benchmark that had at least one acceptable model in the top 1 and top 10 complexes predicted by HADDOCK using each set of residue restraints.

<table>
<thead>
<tr>
<th>Number of complexes with $\geq$ 1 acceptable model</th>
<th>Complexes docked with CPORT Predicted Residues (/66)</th>
<th>Complexes Docked with DMPfold2 Predicted Residues (/66)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 10 HADDOCK Predictions</td>
<td>21 (36.4%)</td>
<td>25 (37.9%)</td>
</tr>
<tr>
<td>Top HADDOCK Prediction</td>
<td>12 (18.2%)</td>
<td>16 (24.2%)</td>
</tr>
</tbody>
</table>

Table 5-4: HADDOCK Docking – CPORT restraints vs. DMPfold2 restraints: Number of complexes in the benchmarking set that had at least one acceptable docking model in the top 10 and top 1 HADDOCK output docking predictions, after using CPORT predicted residues or DMPfold2 predicted residues as docking restraints.
Docking of the heterodimer complexes was performed using five docking protocols. Each protocol had been ranked in the top 10 CAPRI ranked methods and either had a webserver or downloadable code that could be used for protein docking. The number of complexes with at least one acceptable docking prediction in all output decoys, top 10 and top predicted docked poses output by each docking method is shown in Table 5-5. The final column shows the number of complexes that were successfully docked for each method – some methods did not output docking results for some of the complexes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top Output Docked Decoy</th>
<th>Top 10 Output Docked Decoys</th>
<th>All Output Docked Decoys</th>
<th># Of decoys per complex</th>
<th># Of complexes docked</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDock</td>
<td>15 (23.8%)</td>
<td>24 (38.1%)</td>
<td>51 (80.9%)</td>
<td>2000</td>
<td>63</td>
</tr>
<tr>
<td>HADDOCK</td>
<td>17 (26.5%)</td>
<td>25 (37.9%)</td>
<td>37 (57.8%)</td>
<td>10-35</td>
<td>64</td>
</tr>
<tr>
<td>PyDock</td>
<td>12 (20.0%)</td>
<td>23 (38.3%)</td>
<td>23 (38.3%)</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>ClusPro</td>
<td>14 (24.1%)</td>
<td>26 (46.5%)</td>
<td>40 (69.0%)</td>
<td>1000</td>
<td>58</td>
</tr>
<tr>
<td>InterEvDock</td>
<td>20 (30.3%)</td>
<td>32 (48.5%)</td>
<td>32 (48.5%)</td>
<td>10</td>
<td>66</td>
</tr>
</tbody>
</table>

Table 5-5: Heterodimer docking results for five docking protocols: Number of complexes in the benchmarking set with at least one near-native model in the top 1, top 10 and all output docking decoys for each docking method. Each docking method produces a different number of docking decoys, as can be seen in the second column from the right.

Out of the five docking protocols that were assessed for heterodimer docking, InterEvDock was able to produce near-native docking models for 30.3% and 48.5% of the benchmark complexes in the top 1 and top 10 ranked docking poses, respectively. As InterEvDock combines evolutionary information from sequence alignments with atom-based statistical potentials, this shows that using evolutionary information can improve the ranking of docking poses over simply using traditional ranking methods based on surface potentials, physicochemical properties, or energy-based predictions. In fact, it is possible that adding evolutionary information to previously-used scoring functions such
as IRaPPA could improve the ranking of docking poses and increase the chances of getting better heterodimer docking predictions.

Although InterEvDock was the best performing method for generating near-native predictions for the top 10 predicted docking poses, ZDock produced 2000 docking decoys. This set of docking decoys contained at least one near-native model for 80.3% of the benchmark complexes. ZDock generated decoys could therefore be used to test and compare decoy ranking methods to increase the number of near-native decoys ranked at the top of the ranking lists.

5.5.2.3 Comparison of Docked Crystal Structures and DMPfold Structures

The ability of ZDock to generate ‘acceptable’ near-native docked models for complexes docked using crystal structures was compared to complexes docked using DMPfold generated structures. As previously outlined, three types of docking were performed:

- Crystal structures for each monomer chain
- One crystal structure docked to a DMPfold generated structure
- DMPfold generated structures for each monomer chain

The number of complexes with at least one acceptable model in the ZDock generated set of decoys was obtained. Unfortunately, in the case where two DMPfold generated structures were docked together, only 2 complexes had at least one docking decoy with an acceptable DockQ score in the generated set of 2000 decoys, and both those complexes had 3 or less acceptable decoys in the set, which was considerably lower than the number of acceptable decoys in most of the other docking scenarios.

Further examination revealed that this might be due to the fact that for the 66 complexes in the benchmarking set, none of the complexes had a TM-score greater than 0.5 for both protein chains in the complex. In cases where DMPfold TM-scores are less than 0.5, the DMPfold generated structure is unlikely to have the same topology or fold as the native structure.
21 of the complexes had poor TM-scores (TM-score < 0.5) for both chains, while the remaining 45 other structures had only one chain with a TM-score greater than 0.5. For this reason, further analysis was only done on the first two docking cases: two docked crystal structures, and one crystal structure docked to a DMPfold generated structure.

For the 45 heterodimer complexes that were included in further analysis, the ZDock generated decoy sets had at least one acceptable docking prediction for 36 out of 45 of the complexes (80.0%) for the cases where two crystal structures were used for docking. This number dropped to 19 complexes (42%) in the cases where a crystal structure was docked with a DMPfold generated structure.

5.5.2.4 Decoy Ranking with ZRank, Real Contacts and Predicted Contacts

Next, ranking of docked decoys from ZDock was done using three methods:

- ZDock’s internal ZRank+IRaPPA method
- Real contact overlap: Overlap between inter-chain contacts extracted from the native complex (Cβ atom distance < 10Å) and contacts in the docked decoys
- DMPfold2 contact overlap: Overlap between the top 15 inter-chain contacts predicted in section 3.4 by DMPfold2 and contacts in the docked decoys.

Contact overlap in this case is corresponds to the number of common contacts between the decoy and the list being studied (either real contacts or top DMPfold2 contacts) and ‘number decoy contacts’ corresponds to the number of contacts in the docking decoy being assessed. The equation for contact overlap is shown in equation 5.1.

\[
\text{contact overlap} = \frac{\text{number of common contacts}}{\text{number of decoy contacts}}
\]

The top 10 and top 1 ranked decoys were extracted for each complex, and the inclusion rate of acceptable models in the top 1 and top 10 models are shown in Table 5-6.
ZDock generated decoys were ranked using both real inter-chain contacts extracted from the native complex structures and DMPfold2 predicted contacts. 44 out of the 45 Zdock predicted complexes had paired MSAs from EVComplex and therefore had DMPfold2 contact predictions. Table 5-6 shows the number of complexes that had at least one acceptable model in the overall (2000), top 1 and top 10 inclusion lists.
<table>
<thead>
<tr>
<th>Model Type</th>
<th>Ranking Method</th>
<th>Acceptable Model Inclusion List</th>
<th>Acceptable Model Inclusion Rate (/44)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both Crystal</td>
<td>n/a</td>
<td>2000 (all)</td>
<td>36 (81.8%)</td>
</tr>
<tr>
<td>Crystal + DMPfold</td>
<td>n/a</td>
<td>2000 (all)</td>
<td>19 (43.2%)</td>
</tr>
<tr>
<td>Both Crystal</td>
<td>Real contacts</td>
<td>Top 1</td>
<td>34 (77.3%)</td>
</tr>
<tr>
<td>Both Crystal</td>
<td>Real contacts</td>
<td>Top 10</td>
<td>36 (81.8%)</td>
</tr>
<tr>
<td>Both Crystal</td>
<td>ZRank + IRaPPA</td>
<td>Top 1</td>
<td>10 (22.7%)</td>
</tr>
<tr>
<td>Both Crystal</td>
<td>ZRank + IRaPPA</td>
<td>Top 10</td>
<td>16 (36.4%)</td>
</tr>
<tr>
<td>Both Crystal</td>
<td>DMPfold2 contacts</td>
<td>Top 1</td>
<td>15 (34.1%)</td>
</tr>
<tr>
<td>Both Crystal</td>
<td>DMPfold2 contacts</td>
<td>Top 10</td>
<td>19 (43.2%)</td>
</tr>
<tr>
<td>Crystal + DMPfold</td>
<td>Real contacts</td>
<td>Top 1</td>
<td>15 (34.1%)</td>
</tr>
<tr>
<td>Crystal + DMPfold</td>
<td>Real contacts</td>
<td>Top 10</td>
<td>17 (38.6%)</td>
</tr>
<tr>
<td>Crystal + DMPfold</td>
<td>ZRank + IRaPPA</td>
<td>Top 1</td>
<td>2 (4.4%)</td>
</tr>
<tr>
<td>Crystal + DMPfold</td>
<td>ZRank + IRaPPA</td>
<td>Top 10</td>
<td>4 (8.9%)</td>
</tr>
<tr>
<td>Crystal + DMPfold</td>
<td>DMPfold2 contacts</td>
<td>Top 1</td>
<td>1 (2.3%)</td>
</tr>
<tr>
<td>Crystal + DMPfold</td>
<td>DMPfold2 contacts</td>
<td>Top 10</td>
<td>6 (13.6%)</td>
</tr>
</tbody>
</table>

Table 5-6: ZDock Decoy Ranking using ZRank vs. DMPfold2 predicted inter-chain contacts:
Number of complexes with at least one acceptable docked model in the top 1 and top 10 decoys ranked using ZRank or DMPfold2 contacts. Decoys were either generated by docking two crystal structures or a crystal structure and DMPfold structure.
Overall, the results in Table 5-6 showed that ranking of decoys using DMPfold2 contacts had some advantage over ranking of decoys using ZRank. Unfortunately, in the cases where decoys were generated by docking a DMPfold generated structure with a crystal structure, the inclusion of an acceptable model in the top 1 and top 10 ranked models dropped significantly, although DMPfold2 contact ranking still performed marginally better than ZRank.

The inclusion numbers for the decoys ranked with real contacts shows the upper bounds of the number of complexes that can be correctly ranked using inter-chain contact predictions. Decoys ranked using real contacts had a much larger number of acceptable decoys ranked at the top of the ranking lists than ZRank or DMPfold1 contact predictions. This demonstrates the room for improvement of decoy ranking using better inter-chain contact predictions.

It was observed that complexes that had a higher DMPfold2 contact precision (a higher number of real inter-chain contacts included in the top 15 contacts from DMPfold2) were more likely to have an acceptable model in the top 1 or top 10 ranked decoys.

Figure 5.5 shows plots for DockQ score versus ranking for one complex (2P5T_AB) which had an acceptable model correctly ranked in the top 1 and top 10 ranked models for the crystal docked structures. Each plot in the figure corresponds to docking results of the top 2000 docking decoys generated by docking either two monomer crystal structures or docking a monomer crystal structure with a DMPfold predicted monomer structure. The dots in each plot represent a docking decoy (i.e. a predicted protein complex orientation and interface). The y-axis shows the DockQ score of each docking decoy, while the x-axis corresponds to the overlap between the contacts in the decoy and either the real list of contacts (obtained from the crystal structure) or the predicted list of contacts (obtained from DMPfold2). DockQ scores are divided into acceptable, medium, and high-quality docking models by CAPRI evaluation standards. The DMPfold2 precision for this complex was 0.4667, which meant the overall precision of predicted inter-chain contacts for this particular complex was quite high. The x-axis shows the overlap of the docking decoys with either the real native contacts for the complex or the DMPfold2 predicted contacts for the complex.
Figure 5.5: 2P5T_AB Decoy Ranking: Plots show ranking of docking decoys using either real native inter-chain contacts or DMPfold2 generated contacts. Ranking is in decreasing order from right to left (decoys closer to the right are ranked higher on the decoy ranking lists)

a) Crystal structure decoys ranked by decoy contact overlap with real native contacts
b) Crystal structure decoys ranked by decoy contact overlap with DMPfold2 predicted contacts
c) Crystal-DMPfold structure decoys ranked by overlap with real native contacts
d) Crystal-DMPfold structure decoys ranked by overlap with DMPfold2 predicted contacts
Figure 5.6 shows the top ranked docking model for each of these docking and ranking schemes superimposed with the native structure for 2P5T_AB.

**Figure 5.6: 2P5T_AB Top Ranked Decoy:** Superimposed structures of the native (blue and green) 2P5T_AB complex with the top ranked docking decoy from:

a) Crystal structure decoys ranked by decoy contact overlap with real native contacts  
b) Crystal structure decoys ranked by decoy contact overlap with DMPfold2 predicted contacts  
c) Crystal-DMPfold structure decoys ranked by overlap with real native contacts  
d) Crystal-DMPfold structure decoys ranked by overlap with DMPfold2 predicted contacts
Figure 5.7 shows similar ranking plots for another complex, 1FS0_EG which had correct ranking of docking decoys, and also had a relatively high DMPfold2 precision of 0.5333.

**Figure 5.7: 1FS0_EG Decoy Ranking:** Plots show ranking of docking decoys using either real native inter-chain contacts or DMPfold2 generated contacts. Ranking is in decreasing order from right to left (decoys closer to the right are ranked higher on the decoy ranking lists)

a) Crystal structure decoys ranked by decoy contact overlap with real native contacts
b) Crystal structure decoys ranked by decoy contact overlap with DMPfold2 predicted contacts
c) Crystal-DMPfold structure decoys ranked by overlap with real native contacts
d) Crystal-DMPfold structure decoys ranked by overlap with DMPfold2 predicted contacts
Figure 5.8 shows the top ranked docking model for each of these docking and ranking schemes superimposed with the native structure for 1FS0_EG.

**Figure 5.8: 1FS0_EG Top Ranked Decoys**: Superimposed structures of the native (blue and green) 1FS0_EG complex with the top ranked docking decoy from:

a) Crystal structure decoys ranked by decoy contact overlap with real native contacts
b) Crystal structure decoys ranked by decoy contact overlap with DMPfold2 predicted contacts
c) Crystal-DMPfold structure decoys ranked by overlap with real native contacts
d) Crystal-DMPfold structure decoys ranked by overlap with DMPfold2 predicted contacts
For 1FS0_EG, the best ranked docking decoys for the crystal structure docked decoys were the same (DockQ: 0.534). This was because both the ranking using real native contacts and DMPfold2 predicted contacts was able to identify the best docking decoy from the set of generated decoys (Figure 5.7). This was the same in the case of the Crystal-DMPfold docked structures (DockQ: 0.245), where both real native contacts and DMPfold2 contacts were able to pick the best docking decoys. This shows that for this complex, ranking with inter-chain contacts which had a precision of 0.5333 was able to match ranking with real native contacts.

Figure 5.9 ranking plots for a third complex, 3RKO_FG which had correct ranking of docking decoys, and also had a relatively high DMPfold2 precision of 0.9333.
Figure 5.9: 3RKO_FG Decoy Ranking: Plots show ranking of docking decoys using either real native inter-chain contacts or DMPfold2 generated contacts. Ranking is in decreasing order from right to left (decoys closer to the right are ranked higher on the decoy ranking lists)

a) Crystal structure decoys ranked by decoy contact overlap with real native contacts
b) Crystal structure decoys ranked by decoy contact overlap with DMPfold2 predicted contacts
c) Crystal-DMPfold structure decoys ranked by overlap with real native contacts
d) Crystal-DMPfold structure decoys ranked by overlap with DMPfold2 predicted contacts
Figure 5.10 shows the top ranked docking model for each of these docking and ranking schemes superimposed with the native structure for 3RKO_FG.

**Figure 5.10: 3RKO_FG Top Ranked Decoy:** Superimposed structures of the native (blue and green) 3RKO_FG complex with the top ranked docking decoy from:

a) Crystal structure decoys ranked by decoy contact overlap with real native contacts
b) Crystal structure decoys ranked by decoy contact overlap with DMPfold2 predicted contacts
c) Crystal-DMPfold structure decoys ranked by overlap with real native contacts
d) Crystal-DMPfold structure decoys ranked by overlap with DMPfold2 predicted contacts
Table 5-7 shows similar ranking results for eight other complexes ranked using real and DMPfold2 predicted contacts.

<table>
<thead>
<tr>
<th>Complex name</th>
<th>Contact Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>3RKO_FG</td>
<td>0.9333</td>
</tr>
<tr>
<td>3OE_OP</td>
<td>0.8667</td>
</tr>
<tr>
<td>2OE7_HG</td>
<td>0.7333</td>
</tr>
<tr>
<td>3RKO_EG</td>
<td>0.6000</td>
</tr>
<tr>
<td>3HRD_CB</td>
<td>0.5333</td>
</tr>
<tr>
<td>Complex name</td>
<td>Contact Precision</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>2XZN_K4</td>
<td>0.5333</td>
</tr>
<tr>
<td>2JDI_HG</td>
<td>0.1333</td>
</tr>
<tr>
<td>1DKG_DA</td>
<td>0.2000</td>
</tr>
<tr>
<td>2D2C_PO</td>
<td>0.3333</td>
</tr>
</tbody>
</table>

Table 5.7: Zdock Decoy Ranking using real vs. DMPfold2 predicted inter-chain contacts: Plots for DockQ scores against ranking overlap of docking decoys for real contacts vs. predicted contacts.
As can be seen from the results in table 5.7, higher precision of DMPfold2 predicted contacts usually corresponded to better ranking of docked decoys. This demonstrates the usefulness of improved inter-chain contact prediction for the ranking of docking decoys for both crystal structures and crystal structures docked to DMPfold generated structures.

5.6 Conclusions: Heterodimer Docking and Ranking

The work in chapter 5 explored different aspects of heterodimer docking and docked decoy ranking. The main aim of the work in this section was to benchmark various protein docking methods and to examine the application of coevolution-based and deep-learning predictions of the monomer structures and the inter-chain contacts.

First, the performance of five docking methods, ZDock, pyDock, ClusPro, HADDOCK and InterEvDock were compared for docking and ranking of heterodimers in the benchmark set. Since HADDOCK can take in restraint information indicating residues involved in complex interfaces, HADDOCK docking results generated using CPORT predicted residues was compared to HADDOCK docking results generated using DMPfold2 predicted contacts. Overall, the docking results generated using DMPfold2 were better than those generated using CPORT, demonstrating the usefulness of improved inter-chain contact prediction (and thereby inter-chain residues) for improved docking with HADDOCK.

The comparison between five docking methods showed that InterEvDock, a method that uses evolutionary statistical information to dock and score heterodimer complexes performed better than other \textit{ab-initio} and physics-based docking methods. Overall, InterEvDock worked by combining a free-docking method, FRODOCK, with scoring that combined physico-chemical properties with evolutionary information. The improved performance of InterEvDock demonstrates the value of combining statistical evolutionary information with physical restraints to improve the performance of heterodimer complex docking.
InterEvDock improved upon the previously-published docking methods explored. It was useful, however, to explore whether improved prediction of inter-chain contacts could help improve the ranking of docking decoys.

To do this, the decoys generated by ZDock were ranked using ZRank and IRaPPA and compared to ranking of complexes using overlap with real contacts and predicted contacts. The results showed that, as would be expected intuitively, real contacts were able to correctly rank docking decoys in order of increasing quality. In addition, ranking complexes using overlap of decoy contacts with predicted contacts actually performed slightly better than ZRank and IRaPPA, with more complexes with at least one acceptable near-native model ranked in the top 1 or top 10 models in ranked lists.

Overall, results in this section demonstrate that higher contact precision of predicted inter-chain contacts generally correlated with better ranking of docking decoys and corresponded to an increased probability of capturing acceptable near-native models closer to the top of ranked decoy lists.

In addition to docking with crystal structures, docking of predicted structures was also explored by docking DMPfold generated structures. Unfortunately, due to the fact none of the complexes had DMPfold TM-scores greater than 0.5 (likely to have the correct fold) for both chains, only two complexes that were docked using DMPfold structures for both chains had at least one acceptable model in the set of decoys from ZDock.

For this reason, the cases evaluated in this section were made up of DMPfold generated structures docked to crystal structures. These cases simulate complexes where templates may exist for one of the protein chains, but not the other. Although these cases had much better performance than the cases where two DMPfold-generated chains were docked, the docking performance of these cases was still much lower than cases where two crystal structures were docked.

These results demonstrate the importance of improved prediction of monomer structures for the accurate docking of heterodimer complexes. Since the work in this chapter was completed, a number of methods, including AlphaFold2, trRosetta, and even DMPfold2 have shown significant increases in the quality of monomer structure prediction. Future
work could explore the impact of improved monomer structure prediction on the quality of docking decoys generated by \textit{ab initio} or other docking methods.

Overall, the results in this section show the potential of coevolution-based methods and deep learning to improve heterodimer complex structure prediction where templates do not exist for the heterodimer complex, or do not exist for one of the chains. The results demonstrate the potential of combining traditional physics-based docking methods, interface shapes and physico-chemical properties with coevolution-based information and deep learning to improve heterodimer complex prediction. The results in this section also represents one of the first comprehensive benchmarks at the time comparing \textit{ab initio} docking and ranking methods with methods that combine evolutionary statistical information with deep learning. Future work should comprehensively assess the improvements delivered by newer methods such as AlphaFold-Multimer, and whether including traditional scoring information such as shape complementarity, physico-chemical properties and molecular dynamics can improve heterodimer complex predictions.
6. Heterodimer Prediction with Gated Recurrent Units and Language Models

6.1 Introduction and Motivation

Chapter 5 explored previously-established methods for the prediction of heterodimer quaternary structure as well as more recent deep-learning methods for heterodimer complex structure prediction. The results in Chapter 5 demonstrated improvements in heterodimer docking and ranking using published deep-learning methods (DMPfold and DMPfold2) to perform heterodimer inter-chain contact prediction and demonstrated a correlation between the precision of predicted inter-chain contacts and the ability to correctly rank docked decoys.

The deep-learning methods explored in Chapter 5 (DMPfold and DMPfold2) were trained to perform intra-chain contact prediction of monomer chains. For this reason, it was important to explore how similar deep-learning methods would perform when trained specifically to perform inter-chain contact prediction of heterodimer complexes. The work done in this chapter therefore explored a number of deep-learning tools for the prediction of heterodimer inter-chain contacts.

First, the use of a bidirectional GRU (biGRU) architecture was explored for the embedding of MSAs and subsequent prediction of inter-chain contacts (Kandathil et al., 2022). Gated recurrent units (GRUs) are a variation of recurrent neural networks (RNNs) that aim to solve the vanishing gradient problem of standard RNNs (Cho et al., 2014; Chung et al., 2014). They do this using update and reset gates, which determine what information will be passed to the output and allow the model to store information over several time steps (Chung et al., 2014), or in the case of protein sequences, over several residue sequences (Kandathil et al., 2022). As explained by Kandathil et al., although RNNs in theory avoid the vanishing gradient problem over sequences, in practice, dependencies above windows of a few hundred residues are usually poorly modelled (Kandathil et al., 2022). For this reason, two GRU networks were used to embed MSAs in the vertical and horizontal directions thereby reducing the number of sequential residues the GRUs had...
to embed (Kandathil et al., 2022). Bidirectional GRUs were used successfully in DMPfold2 to embed monomer MSAs for the purpose of monomer structure prediction (Kandathil et al., 2022). The biGRU architecture explored here was similar to that used in DMPfold2, with modifications made to enable the prediction of heterodimer contacts. The goal of this work was to determine whether biGRUs embeddings could be used in combination with other information such as sequence embeddings from language models and inverse covariance information as inputs to residual neural networks to predict heterodimer interface contacts.

During the time the work in this chapter was being completed, a number of sequence embedding models were published and were shown to perform well on protein structure prediction tasks. Two suites of language-based models, Evolutionary Scale Modelling (ESM) (Rives et al., 2021) and ProtTrans (Elnaggar et al., 2020) had been shown to produce high-quality embeddings of protein sequences that could be used for monomer protein contact prediction. The best performing methods from each of these suites, ESM-1b and Prot-T5-XL, both used transformer protein language models to obtain sequence embeddings (Elnaggar et al., 2020; Rives et al., 2021). The ESM-1b model is based on the RoBERTa transformer architecture and training procedure (Liu et al., 2019). The model was trained in an unsupervised manner on the Uniref50 2018_03 protein sequence database using raw protein sequences (Rives et al., 2021). The Prot-T5-XL model also uses a transformer architecture and was trained on the Big Fat Database (BFD) (Elnaggar et al., 2020). Previous papers benchmarking monomer protein contact prediction using sequence embeddings showed that ESM-1b and Prot-T5-XL outperformed previous state-of-the-art (SOTA) single-sequence contact prediction methods (Elnaggar et al., 2020; Rives et al., 2021). Since these sequence embedding methods embedded single sequences and not MSAs for protein chains, they had promising application for heterodimer inter-chain contact prediction as they circumvented the problem of MSA pairing. In this chapter, sequence embeddings of heterodimer chains obtained from ESM-1b and Prot-T5-XL were combined with biGRU embeddings of the heterodimer MSAs chains to obtain heterodimer contact predictions. The goal of this work was to determine if adding single-sequence embeddings of protein chains to MSA embeddings from the biGRU model improved the precision of predicted heterodimer inter-chain contacts.
Work in Chapter 5 demonstrated the difficulty of performing accurate pairings of multiple sequence alignments. MSA pairing is a non-trivial and challenging process, that can have significant impact on the accurate prediction of inter-chain contacts, as demonstrated by the results in Table 5-2. One of the benefits of using biGRU MSA embeddings was the fact it was possible to embed multiple sequence alignments (MSAs) separately, which meant that the MSAs did not have to be paired before being passed to the biGRUs. One of the motivations of this chapter was to compare the performance of the models trained using biGRU embeddings of paired MSAs against biGRU embedding trained using unpaired MSAs. Successful prediction of heterodimer contacts using unpaired sequences could circumvent the MSA pairing problem, thus reducing the difficulty of heterodimer contact prediction.

Overall, the work in this chapter explores the application of newer, and more powerful, deep-learning models including bidirectional GRUs, protein language models, and inverse covariance matrices, for the prediction of heterodimer complex structures. While some of these methods showed some improvement on previously explored heterodimer prediction models, the biGRU + ResNet models explored in this chapter did not improve on inter-chain contact precisions obtained for DMPfold2 in Chapter 5. The ColabFold implementation of AF2 for complex prediction showed significant improvements of predicted complexes over previous non-AF2 SOTA methods for heterodimer prediction.
6.2 Methodology: Interface Contact Prediction and Heterodimer Complex Prediction

6.2.1 Dataset Curation for Heterodimer Complex Prediction

The biGRU and sequence embedding models explored in this chapter were trained, tested, and validated on heterodimer complexes from the PiFace (Cukuroglu et al., 2014) database combined with the set of heterodimer complexes added to the PDB after the PiFace database was released. The set of heterodimer complexes was extracted from the PiFace dataset as outlined in section 5.2. Heterodimers were defined as having protein chains with less than 95% sequence similarity and RMSD greater than 1.5Å to one another. All complexes that had at least one protein chain with a protein length less than 30 residues were removed from the set of heterodimer interfaces. The resulting set contained 4,887 heterodimer interfaces.

Since the PiFace dataset was released in 2014 and only included structures released before October 2012, the PiFace dataset was missing a significant number of heterodimers that had been published between January 2012 and the time the work in this chapter was started, in August 2020. Unfortunately, the code that was used to cluster and curate the PiFace database could not be obtained, and a different curation method was used to curate the complexes added to the PiFace set as outlined below.

Heterodimer complexes released on the PDB between 2\textsuperscript{nd} October 2012 and 12\textsuperscript{th} August 2020 with a resolution less than 3Å were obtained from the RCSB webserver. This query resulted in 5,502 structures. These structures were then filtered for complexes where either chain was less than 30 residues in length or complexes had less than 5 inter-chain contacts. Next, the protein pairs for each of the proteins were clustered by 40% sequence similarity and 50% sequence coverage using mmseqs2. Heterodimers were clustered together if their constituent protein chains were in the same clusters. Representative heterodimers were selected from each cluster, resulting in 3,758 representative complexes.
Once these representative heterodimers were obtained, the complexes were filtered against the set of heterodimers already present in the PiFace set. This was done using both sequence similarity and ECOD topology overlap. It was assumed that protein complexes that have similar interacting subunits are likely to have similar interaction interfaces. The first step of reducing redundancy was therefore to filter out the heterodimer complexes that were made up of protein pairs that overlapped in terms of sequence or structure with protein pairs in the PiFace set.

To filter the heterodimer complexes by sequence overlap with PiFace, the mmseqs2 method (Steinegger and Söding, 2017) was used to cluster the heterodimer sequences by 40% sequence identity and 50% sequence coverage. Filtering of the new heterodimer set was done by removing all instances where both chains of heterodimer complexes were in the same cluster as both chains of heterodimer complexes in the PiFace set. This left a set of 2,892 heterodimer complexes.

Filtering against the PiFace dataset was repeated at a structural level. It was assumed that heterodimers made up of subunits that had the same folds were likely to have similar binding modes to one another. For this reason, the ECODanalyze method used in previous chapters was used to remove complexes where both the subunits overlapped with both subunits of at least one heterodimer complex in the PiFace dataset. This filtering left a set of 1,556 complexes that had no sequence or structure overlap with the PiFace dataset. All heterodimers were then run through the modeller complete_pdb function to fill in any missing residues or gaps in the protein structures (Eswar et al., 2006).

This set of 1,556 complexes was added to the PiFace set of 4,887 heterodimers. This combined set of 6,443 became the heterodimer dataset (HetData) used to train and evaluate deep-learning methods developed in this chapter. The heterodimer complexes in HetDataset were then split into a training, test, and validation set using the ECODclust method outlined in Section 2.2.2.3. This resulted in test and validation sets of 100 heterodimers each, and a training set of 6,243 heterodimers. Figure 6.1 shows the pipeline that was used to curate the HetData set.
**Figure 6.1: HetDataset Generation:** Pipeline used for generation of training, test, and validation sets for training of heterodimer inter-chain contact prediction deep learning models

### 6.2.2 Multiple Sequence Alignments for Training of biGRU models

Before training of the deep learning models was started, multiple sequence alignments had to be generated for all the heterodimers in HetData. MSAs for protein chains were obtained by running HHblits using 3 iterations over the UniClust30 (August 2018) database, with an E-value threshold of 0.001 and a minimum sequence coverage of 50%. These MSAs were used as the input MSAs for all models trained in this chapter.
6.2.3 Bidirectional GRU and Residual Neural Network for Contact Prediction

The primary deep learning model that was trained in this chapter used bidirectional GRUs to embed the multiple sequence alignments of each protein chain of the heterodimer complexes in the horizontal and vertical directions. The inputs to the biGRU model were one-hot encodings of the sequences in the MSAs using 22 residue types which included the 20 standard amino acids plus gaps and unknown residues. Since most machine learning algorithms require numerical input and output variables, it is important to convert the categorical variables in MSAs (i.e. amino acids) into numerical values that can be easily digested by ML methods. In this case, one-hot encoding was chosen over numerical encoding used in previous chapters as it has several advantages. First, it can often better represent the presence or absence of a category. Second, it is often more efficient in terms of memory and computational cost of complex calculations involved in training ML models. In addition, it gets around one problem found with integer encoding (also called label encoding): sometimes ML algorithms implicitly associate higher label values with a ranking of the labels. If one amino acid therefore has a higher integer value encoding the ML model might assume it is somehow bigger or more important than an amino acid with a lower integer value encoding, when this is not the case. One-hot encoding gets around this problem.

One hot encoding of the MSAs involved converting the MSAs with 20 amino acid types plus gaps and unknown residues into binary array of dimension $22 \times L \times D$. Here 22 represents each possible amino acid at each position, and is given a value of 1 if that amino acid is present at that position and 0 if it is not. Since this is done for each amino acid along the length and depth of a protein MSA, L represents the length of the protein, while D represents the depth of an MSA profile. One-hot encoding involves converting categorical variables (in this case amino acids) into a form that is easier for machine learning algorithms to learn. One-hot encoding has been shown to be more effective for representation of categorical variables.
For each MSA, the one-hot encodings were first fed into a stack of 2 GRU layers that embedded the MSAs in the vertical direction. These GRU layers scanned individual MSA columns in the vertical direction to obtain a per-column representations of the MSA (Figure 6.2). The vertical GRU layers both had a hidden dimension of 256 and therefore resulted in fixed depth representations of 256 x L, where L was the length of the protein chain being embedded. The per-column representation of the MSA was then passed to a stack of 2 GRU layers (also with a hidden dimension of 256) to obtain an embedding in the horizontal direction. This resulted in an MSA embedding of dimension 1 x 128 for each embedded MSA (Figure 6.2).

This bidirectional GRU embedding was run on both protein chains in each heterodimer complex, resulting in embeddings of dimension 1x128 for each protein chain. To get a combined representation of the two embeddings, each embedding was stripped in the horizontal direction by the length of the embedded MSA. This resulted in two embeddings, with respective sizes of (2 x M x 128) and (2 x N x 128) where M was the length of the first heterodimer protein chain, and N was the length of the second heterodimer protein chain. The combined representation of the heterodimer complex was then obtained using batch matrix multiplication. To do this, the elements of the second embedding were reordered to get a matrix of dimension (2x128xN). The batch matrix-matrix product of the two matrices was then obtained, which resulted in a complex representation of dimension 2xNxM. This combined representation was then fed into a residual neural network made up of 14 residual blocks with kernel size of 5. The convolutional layers of the residual blocks had an input and output channel size of 32. The output of the residual neural network was a single-channel matrix of size (1xMxN) which contained the probabilities of each residue pair in the heterodimer complex being involved in an inter-chain residue contact. The architecture for the basic biGRU + ResNet model is shown in Figure 6.2.
Figure 6.2: BiGRU + ResNet model: MSA embedding and contact prediction architecture for heterodimer inter-chain contact prediction. Numbers in parenthesis show the output dimensions of each layer or operation. Maxout layers were made up of 2 convolutional layers with a ReLU activation function in-between followed by an instance norm layer.

6.2.4 Language Model Sequence Embeddings

Two transformer protein language models were used to generate sequence embeddings for the heterodimer protein chains. These sequence embeddings were combined with biGRU MSA embeddings and passed to the ResNet model shown in Figure 6.2 to obtain heterodimer contact predictions. These two language models, ESM-1b (Rives et al., 2021) and Prot-T5-XL (Elnaggar et al., 2020) had both been demonstrated to produce high-quality protein sequence embeddings that could be used to perform protein prediction tasks including secondary structure prediction, structure clustering, and protein contact prediction.
The fair-esm python package [https://github.com/facebookresearch/esm] was used to obtain the pretrained ESM-1b model that was used to embed the sequences of the heterodimer protein chains in HetData. The Prot-T5-XL model was downloaded from the ProtTrans repository [https://github.com/agemagician/ProtTrans] and was run in PyTorch v1.4.0. For protein chains of length M and N, these models output sequence embeddings with per-residue representations of size 1028 for ESM-1b and 1024 for Prot-T5-XL. For ESM-1b, where M and N were the lengths of the first and second heterodimer protein chain respectively, the sequence embeddings had dimensions (1 x M x 1028) and (1 x N x 1028). To get the joint sequence embedding of the heterodimer complexes, the second embedding vector was transposed and the matrix-matrix product of the two embeddings was obtained to get a joint embedding representation of dimension (1 x M x N). The process of obtaining and getting the matrix product of the sequence embeddings was identical for Prot-T5-XL, with the only difference being the embedding dimension, which was 1024.

The joint sequence embedding for each sequence embedding model was then concatenated to the biGRU embeddings of the heterodimer MSAs to get an input matrix of dimension (3 x M x N). These concatenated embeddings were then passed to the same residual neural network shown in Figure 6.2 to obtain heterodimer contact predictions. The LM + biGRU + ResNet architecture is shown in Figure 6.3.
Figure 6.3: LM + BiGRU + ResNet model architecture: Architecture of the language model + bidirectional GRU and ResNet model used to obtain inter-chain contact predictions for heterodimers. Embed_dim here corresponds to the embedding dimension of ESM-1b and Prot-T5-XL, which are 1028 and 1024 respectively. M and N correspond to the lengths of the heterodimer protein chains. The ResNet used here was identical to the one in figure 6.2, with the only difference being the dimensions of the inputs.

6.2.5 Sequence pairing for LM + biGRU + ResNet model

Since the models described above use embeddings from unpaired MSAs, the embedding and ResNet models were also run on paired multiple sequence alignments to determine whether MSA pairing resulted in better inter-chain contact predictions from the LM + biGRU + ResNet models. Pairing of multiple sequence alignments was done using the modified MI+IPA method developed in section 5.3.2.1. Paired sequences were concatenated with a 20-residue glycine linker then fed to the biGRU + LM + ResNet model.

For each embedding method, embedding resulted in a vector of size (1 x PL x embed_dim) where PL was the length of the paired alignment and was equal to the length of the 2 sequences and the glycine linker (M + N + 20). Embed_dim corresponded to the embedding dimension, for the biGRU, ESM-1b and Prot-T5-XL respectively. The embedding vector were then striped in the vertical and horizontal directions as per...
The embeddings from the biGRU and language models were concatenated to get an input of dimension (3 x M x N), which was fed into a residual neural network as shown in Figure 6.4. The output of the residual neural network was a matrix of size (1 x PL x PL). Final inter-chain contact predictions were extracted by extracting the rectangular matrix containing inter-chain pair predictions, as shown in figure 6.4. This was the first variation that was trained and tested for inter-chain contact prediction.

A second version of the paired biGRU + LM + ResNet model added an input containing an approximation of inverse covariance, called the precision matrix, as was done in DMPfold2 (Kandathil et al., 2022). This precision matrix was calculated using the “fast_dca” algorithm from trRosetta (Yang et al., 2020). Figure 6.4 shows the architecture used to train the biGRU + LM + ResNet model using paired MSA inputs.

**Figure 6.4: Paired LM + BiGRU + ResNet model architecture:** Architecture of the language model + bidirectional GRU and residual neural network model used to obtain inter-chain contact predictions from paired sequence alignments. Embed_dim here corresponds to the embedding
dimension of ESM-1b and Prot-T5-XL, which are 1028 and 1024 respectively. M and N correspond to the lengths of the heterodimer protein chains, and PL corresponds to the length of the paired sequences with the glycine linker (M+N+20).

6.2.6 Model training of variations of the biGRU + ResNet model

All the models trained and tested in this section were trained in a similar manner. All models were trained on the HetData training data and evaluated on the validation set at each epoch and were trained in PyTorch using the Adam optimizer, with a learning rate of $10^{-4}$. Models were trained over 1000 epochs.

There was a very large class imbalance between positive and negative cases of inter-chain contacts (3132 interface contacts vs 648,729 non-interface contacts in the validation set). The goal was to increase enrichment of inter-chain contacts at the top of the list of residue pairs. For this reason, the margin ranking loss function (with margin = 3) was used to train the models trained in this section. Losses were backpropagated over batch sizes of one – i.e., one heterodimer inter-chain contact map per batch. The precision of the top 500 and 1k ranked contacts predicted for the validation set was used in addition to the overall validation MCC as a model saving criteria.

At each epoch, model weights were saved if the validation MCC, top 500 precision or top 1000 precision on the validation set exceeded the value of these metrics in previous epochs. The model weights that resulted in the best precision for the top 500 and top 1k ranked inter-chain contacts for the validation set were saved as the final model weights for each model. These saved models were used for evaluation of the HetData test set.

6.3 Results: BiGRU + ResNet Models for Inter-Chain Contact Prediction

Once each model was trained on the HetDataset as outlined in section 6.2, each of the models was evaluated on the HetDataset test set. The loss and precision values of each method on the training, validation, and test sets are shown in Table 6-1.
<table>
<thead>
<tr>
<th>Model</th>
<th>Validation precision top 500</th>
<th>Validation precision top 1000</th>
<th>Test Precision top 500</th>
<th>Test Precision top 1000</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>biGRU (unpaired)</code></td>
<td>0.086</td>
<td>0.079</td>
<td>0.080</td>
<td>0.082</td>
</tr>
<tr>
<td><code>biGRU + ESM-1b (unpaired)</code></td>
<td>0.110</td>
<td>0.101</td>
<td>0.098</td>
<td>0.089</td>
</tr>
<tr>
<td><code>biGRU + Prot-T5-XL (unpaired)</code></td>
<td>0.094</td>
<td>0.085</td>
<td>0.102</td>
<td>0.092</td>
</tr>
<tr>
<td><code>biGRU (paired)</code></td>
<td>0.098</td>
<td>0.091</td>
<td>0.092</td>
<td>0.087</td>
</tr>
<tr>
<td><code>biGRU + ESM-1b (paired)</code></td>
<td>0.122</td>
<td>0.118</td>
<td>0.114</td>
<td>0.107</td>
</tr>
<tr>
<td><code>biGRU + Prot-T5-XL (paired)</code></td>
<td>0.114</td>
<td>0.108</td>
<td>0.108</td>
<td>0.103</td>
</tr>
<tr>
<td><code>biGRU + ESM-1b + precision matrix (paired)</code></td>
<td><strong>0.136</strong></td>
<td><strong>0.125</strong></td>
<td><strong>0.128</strong></td>
<td><strong>0.121</strong></td>
</tr>
</tbody>
</table>

**Table 6-1: Training, Validation and Test Metrics:** Results for the training loss and precision scores for each variation of the biGRU + ResNet models trained.

The saved models were also run over all the heterodimers in the test set (100 complexes). The top 5, 10, and 15 ranked interface contacts were extracted for each complex. The average precision of the top 5, 10, and 15 contacts per complex are shown in Table 6-2.
<table>
<thead>
<tr>
<th>Model</th>
<th>Top 5 precision</th>
<th>Top 10 precision</th>
<th>Top 15 precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>biGRU (unpaired)</td>
<td>0.172</td>
<td>0.158</td>
<td>0.121</td>
</tr>
<tr>
<td>biGRU + ESM-1b (unpaired)</td>
<td>0.192</td>
<td>0.183</td>
<td>0.149</td>
</tr>
<tr>
<td>biGRU + Prot-T5-XL (unpaired)</td>
<td>0.186</td>
<td>0.174</td>
<td>0.140</td>
</tr>
<tr>
<td>biGRU + ESM-1b (paired)</td>
<td>0.220</td>
<td>0.195</td>
<td>0.179</td>
</tr>
<tr>
<td>biGRU + Prot-T5-XL (paired)</td>
<td>0.216</td>
<td>0.191</td>
<td>0.177</td>
</tr>
<tr>
<td>biGRU + ESM-1b + precision matrix (paired)</td>
<td>0.222</td>
<td>0.208</td>
<td>0.191</td>
</tr>
</tbody>
</table>

Table 6.2: Precision of top ranked contacts for HetData test set: Average top 5, 10, and 15 precisions for each dimer in the HetData test set.

6.3.1 Paired versus Unpaired Prediction of Inter-Chain Contacts: Binominal Test of Proportions

To determine whether the improvement in contact precision for prediction using paired MSAs over unpaired MSAs was statistically significant, a binomial test of proportions was used to compare the models trained on paired and unpaired MSAs. This meant comparing the values for top 5, top 10 and top 15 precision between the paired and unpaired models tested in Table 6.2.
Comparison | Top 5 p-value | Top 10 p-value | Top 15 p-value
--- | --- | --- | ---
*biGRU + ESM-1b: paired vs. unpaired* | 0.0518 | 0.1534 | 0.0120
*biGRU + Prot-T5-XL: paired vs.* | 0.0394 | 0.0735 | 0.0007

Table 6-3: Binomial Test of Proportions of paired vs. unpaired models: statistical significance of the difference between paired and unpaired MSAs for equivalent biGRU models trained in Table 6.2.

As can be seen from table 6.3, paired and unpaired models for *biGRU + ESM-1b* were only statistically different in terms of performance for the top 15 contacts, while the paired and unpaired models for *biGRU + Prot-T5-XL* were statistically different for both top 5 and top 15 contact prediction performance. Overall, the difference between paired and unpaired biGRU based models was seen mainly at the level of top 15 ranked contacts.

This shows that although biGRUs can be used to embed heterodimer chains separately, inter-chain contact prediction still benefits from performing MSA pairing prior to MSA embedding. Correct MSA pairings therefore are still important to making accurate inter-chain contact predictions. This could be because paired alignments are likely to maintain more information about evolutionary histories.

### 6.4 Conclusion: Deep Learning Methods for Heterodimer Complex Prediction

The results in table 6.1 and 6.2 showed that the biGRU + ResNet models trained with sequence embeddings did better than those trained without sequence embeddings. In addition, the models that used paired MSAs as inputs performed better overall than the models that used unpaired MSAs. This demonstrates that even when using models that can embed protein sequences of the complex separately, it is beneficial to use a paired multiple sequence alignment. The best performing method used biGRU MSA embeddings and ESM-1b sequence embeddings of the paired MSA in addition to a
precision matrix as inputs to the residual neural network. Overall, adding ESM-1b embeddings and pairing MSAs seemed to improve the overall performance of biGRU + ResNet models.

Unfortunately, when comparing the precision values of the methods evaluated in table 6.2 to the precision values of the methods evaluated in Table 5-3, it is apparent that the even the best biGRU + ResNet model does not match the precision of methods explored in chapter 5, especially DMPfold2.

Overall, despite using biGRU MSA embeddings and a residual neural network similar to DMPfold2, the DMPfold2 method had a higher average precision for the top 5, 10, and 15 ranked contacts for each complex. This could be due to a number of reasons. First, DMPfold2 iteratively predicts a distance matrix that is projected into Cα coordinates using multidimensional scaling (MDS) and a coordinate refinement module (Kandathil et al., 2022). Inter-residue distances of these projected Cα coordinates are then iteratively fed in as an input in addition to the precision matrix and biGRU embedding to the residual neural network. This structure refinement may contribute to better inter-chain contact predictions when using DMPfold2 on paired MSAs as was done in Chapter 5. Second, DMPfold2 was trained on a much larger dataset overall (31,159 domains and 6,742 full length chains) than the biGRU+ResNet models trained in this chapter (Kandathil et al., 2022). Finally, DMPfold2 predicts residue-residue distances for both intra-chain residue pairs and inter-chain residue pairs. It is possible that training for monomer residue distances and interface residue distances improves the overall ability of DMPfold2 to do heterodimer contact prediction (Kandathil et al., 2022). Future work could explore the impact of adding a similar MDS and coordinate refinement module at the end of the biGRU + ResNet model explored in this chapter to get Cα coordinate predictions that can be fed back iteratively into the residual neural network. It is possible doing this would improve the precision of heterodimer complex predictions.
7. AlphaFold2 Evaluation for Complex Docking

While work in Chapter 6 was in progress, the AlphaFold2 method was announced at CASP13. AlphaFold2 demonstrated a structure prediction accuracy that was competitive with experimental structures in several cases, and additionally out-performed the next closest-ranked structure prediction methods at CASP13 (AIQuraishi, 2019; Jumper et al., 2021). AlphaFold2 had been demonstrated to have significant improvements in monomer structure prediction (AIQuraishi, 2019, p. 1; Akdel et al., 2021, p. 2). Initial analysis had shown that it was possible to perform structure prediction of heterodimer complexes using AlphaFold2 with an offset of 200 residues in the residue index, or by using a long glycine linker (Baek, 2021a; Moriwaki, 2021). Once the AlphaFold2 code released in July 2021, a number of papers and Google Colab notebooks were released that explored the application of AlphaFold2 for the structural prediction of protein complexes (Akdel et al., 2021; Bryant, Pozzati and Elofsson, 2021; Evans et al., 2021; Mirdita et al., 2022). Initial results of the AlphaFold2-based complex prediction methods seemed to show significant improvements in the ability of modified AlphaFold2 methods to perform end-to-end folding and docking of heterodimers to produce better complex structures that most previous SOTA methods (Baek, 2021a, 2021b; Bryant, Pozzati and Elofsson, 2021; Mirdita et al., 2022).

The code for a modified AlphaFold2 method for heterodimer prediction was available in the form of a Google Colab notebook [https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2_complexes.ipynb]. For this reason, work outlined in this chapter involved running the ColabFold method available from the Colab notebook (Mirdita et al., 2022) on heterodimer complexes that were not included in the original AlphaFold2 training set. The results of this work demonstrated a significant improvement in heterodimer structure prediction over the methods previously explored in this thesis.

The results evaluating the ColabFold implementation of AlphaFold2 showed similar performance to methods benchmarked in the AlphaFold-Multimer paper by Evans et al., that was released while the AF2 complex analysis in this chapter was being completed (Evans et al., 2021). AlphaFold-Multimer was developed by re-training the original
AlphaFold2 deep learning architecture with some modifications for the task of protein complex prediction, including using the UniProt database to generate MSAs and pairing sequences in generated MSAs. AlphaFold-Multimer showed significant improvement in heterodimer complex structure prediction over previously existing methods, including other implementations of AF2 for complex prediction (Evans et al., 2021).

7.1 Methodology: AlphaFold2 for Heterodimer Complex Prediction

The AlphaFold2 method was announced and evaluated in the CASP13 challenge (AlQuraishi, 2019; Jumper et al., 2021). After the AlphaFold2 code was released, a number of Google Colab notebooks were released to perform heterodimer structure prediction (Baek, 2021b; Jumper et al., 2021; Mirdita et al., 2022). One of these Colab notebooks, ColabFold (Mirdita et al., 2022) was used to generate structure predictions for a benchmark set of heterodimers.

To evaluate the ability of AlphaFold2 to perform heterodimer complex structure prediction, a benchmark set was generated. All heterodimers released in the PDB after 29 August 2018 were obtained from the RCSB webserver (this was date the AF2 training data was extracted). This query resulted in 1,426 heterodimers. Heterodimers with lengths less than 30 for either protein chain were filtered out of the heterodimer dataset. Heterodimers with a combined length greater than 1400 were also filtered from the database. Overlap with the AlphaFold2 training set was removed by filtering out all heterodimer complexes where both chains had sequence identity >25% and coverage >50 to chains in the AF2 training set (using mmseqs2 for clustering).

435 complexes were left in the benchmark set after filtering. Next MSAs for the protein chain were obtained using mmseqs2 run on the UniRef90 v. 2020_01 database (Breuza et al., 2016). Sequences were paired using the RoseTTaFold sequence pairing protocol which paired sequences by extracting the prokaryotic sequences and using distances in alphanumeric values of accession numbers to pair the MSA sequences (Baek, DiMaio, et al., 2021). Unfortunately, one of the limitations of this method was the fact it did not
perform pairing of eukaryotic sequences. Heterodimers with paired alignments with a depth less than 10 sequences were removed from the benchmark set. This left 289 complexes in the benchmark set.

Next, the code for the ColabFold method, which modified AlphaFold2 for complex prediction, was downloaded (Baek, 2021b). At the time the work in this chapter was started, the code for AlphaFold2-multimer (AF2-multimer) was not released. The AlphaFold2 model was used to perform prediction of the tertiary and quaternary structures of the heterodimer complexes. Resulting complexes were analysed for DockQ scores and TM-scores. Results for the AlphaFold2 benchmarking are reported in section 7.2.

7.2 Results

7.2.1. AlphaFold2 Heterodimer Structure Prediction

The results of AlphaFold2 structure prediction of the heterodimer complexes in terms of the number of complexes predicted with at least one acceptable-, medium-, or high-quality structure (based on DockQ scores) in the top 5 predicted AlphaFold2 output models is shown in Table 7-1.

<table>
<thead>
<tr>
<th>AlphaFold2 Models</th>
<th>Acceptable or better (/289)</th>
<th>Medium or better (/289)</th>
<th>High or better (/289)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 1</td>
<td>139 (48.1%)</td>
<td>61 (21.1%)</td>
<td>21 (7.3%)</td>
</tr>
<tr>
<td>Top 5</td>
<td>151 (52.2%)</td>
<td>67 (23.2%)</td>
<td>22 (7.6%)</td>
</tr>
</tbody>
</table>

Table 7-1: AlphaFold2 structure prediction on AlphaFold2 Heterodimer benchmark set: Number of structures with acceptable, medium, or high-quality structures scored with DockQ scores. CAPRI qualities correspond to DockQ scores as follows: Incorrect: $0 \leq$ DockQ $< 0.23$ Acceptable: $0.23 \leq$ DockQ $< 0.49$ Medium: $0.49 \leq$ DockQ $< 0.80$ High: $0.80 \leq$ DockQ
Overall, the proportion of correctly predicted heterodimer complexes using the ColabFold version of the AF2 code was higher than all all the methods for de novo complex prediction benchmarked in Chapter 5 (Table 5-6).

Figures 7.1 - 7.10 show 10 heterodimer structures produced using the ColabFold AlphaFold complex code. The figures are ordered by decreasing DockQ scores. TM-scores are reported for the heterodimer chain with lower fold similarity to the protein chains in the native complex. For all 10 figures, the graph in the top left shows the predicted per-residue LDDT (local accuracy) scores for the 5 models generated by AF2 while the heat-maps on the top right show the global predicted aligned error (PAE) scores of residue pairs. Higher PAE values (red) correspond to worse inter-residue distances predictions while lower PAE values (blue) correspond to better inter-residue distance predictions. Residue pairs in blue are likely to be correctly predicted while residue pairs in red are likely to be incorrectly predicted. The structure on the bottom left shows the native structure (light blue and light green) superimposed with the top AF2 predicted structure (dark blue). The structures on the bottom right show the native structure superimposed with the 5 models predicted by AF2. The figures also show the DockQ for the top AF2 predicted model to the native structure.

Two complexes (7D5B_AD and 7D5P_AC) have relatively small interfaces and had the wrong predicted orientation between the protein chains. This could be due to the fact the interfaces for these complexes are relatively small (between 6-10 interface contacts). Results in Chapter 2 showed that complexes with smaller interfaces are less likely to be correctly predicted.

The 4 structures with the highest DockQ scores in the figures below (Figures 7.1 – 7.4) have PAE matrices that are almost completely blue, which corresponds to low PAE values and high predicted inter-residue distance accuracies. In contrast, the complexes with incorrectly predicted AF2 models predictions (Figures 7.6 - 7.10) have higher PAE values (i.e. values in red) in the regions of the PAE value maps that correspond to inter-chain residue pairs. This demonstrates the fact that PAE values could potentially be used to
predict whether AF2 generated heterodimer complexes are correctly predicted: i.e., predicted with correct inter-chain orientations.

**Figure 7.1: 7D9W_AB AlphaFold2 models**: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
Figure 7.2: 6W2L_AB AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
Figure 7.3: 6L4P_AB AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
Figure 7.4: 7LRQ_AB AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) superimposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
Figure 7.5: 6LE5_AB AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
Figure 7.6: 7CSL_AC AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
Figure 7.7: 7D5B_AD AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
Figure 7.8: 7KOZ_CD AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021).
Figure 7.9: 7D5P_AC AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
Figure 7.10: 5R1D_AB AlphaFold2 models: Results of AlphaFold2 heterodimer complex prediction using the ColabFold code. Bottom left: model of top AlphaFold2 complex structure (dark blue) super-imposed with native structure (light green and light blue). Bottom right: all 5 AF2 complex models generated super imposed with native structure (light green and light blue). Structure images generated using ChimeraX (Pettersen et al., 2021)
7.3 Conclusion: AlphaFold2 Complex Docking

Benchmarking of the ColabFold implementation of AlphaFold2 demonstrated significant improvements of AF2 to do heterodimer complex prediction over most methods explored in Chapter 5. Additionally, it appears that PAE values of AF2 models may give some indication of the likelihood of a complex being correctly predicted. This is important as it would give computational biologists some indication whether the complex structures from AlphaFold2 are likely to be correct.

While the work in this chapter was being completed, a new complex prediction method, AlphaFold-multimer (Evans et al., 2021), was published. AlphaFold-multimer used end-to-end protein structure modelling similar to the AlphaFold2 method to perform complex prediction. AlphaFold-multimer adapted the AlphaFold2 architecture for the specific task of protein complex prediction by training the model on paired alignments for homo- and hetero-oligomers (Evans et al., 2021).

AlphaFold2-multimer used paired MSAs generated using the method introduced by Zhou et. al (Zhou, Wang and Xu, 2017). To pair sequences, the UniProt species annotation was used. For sequences from prokaryotic species, pairing was done using the smallest genetic distances, and for sequences from eukaryotic species, sequences were ranked by similarity to the target sequence then pairs of the same rank were concatenated (Evans et al., 2021). In addition, AlphaFold-Multimer used a protein chain cropping strategy similar to that used in the original AlphaFold2 method (Jumper et al., 2021), with modifications made to maximize the coverage of the protein chains and the crop diversity, while balancing the number of crops from interface and non-interface regions (Evans et al., 2021). Overall, results in the AlphaFold-Multimer paper showed the method had the best performance in terms of mean DockQ score compared to other variations of the AlphaFold2 method, and showed very large improvements over previous state of the art methods for heterodimer structure prediction (Evans et al., 2021).

Evaluations of AlphaFold2 multimer demonstrated a significant improvement on previously established methods for the prediction of homodimers, heterodimers, and even higher order complexes (Evans et al., 2021). This represents a significant leap forward in
the *in silico* modelling of heterodimer complexes. Future work could explore the limits of AF2-multimer complex prediction, and what kinds of complexes are more or less likely to be predicted using the AlphaFold method.
8. General conclusions and future outlook

Work preceding this thesis had established the usefulness of covariation-based methods for the prediction of residue-residue interactions in protein structures. The usefulness of these methods for the prediction of protein monomers had been established in publications prior to 2017, and some published studies had demonstrated evolutionary statistical methods could predict residue contacts at protein interfaces. Previously published work applying evolutionary statistical methods to the prediction of inter-protein contacts, however, had been limited by several challenges. In the case of homodimer complexes, evolutionary statistical methods (or CBMs) could not reliably differentiate between inter-chain and intra-chain residue contacts unless known monomeric contacts could be used to filter out intra-chain residues from predicted contact maps. This meant that the prediction of homodimer inter-chain contacts was mainly limited to cases where crystal structures or close templates were available for the monomer subunits, or to specific cases such as trans-membrane proteins which are often made up of simple alpha-helix structures and where lipid-facing residues could be predicted and therefore eliminated from inter-chain contact maps (Polyansky et al., 2014; Wang and Barth, 2015).

In the case of heterodimers, CBMs were limited by the need for deep and accurately paired multiple sequence alignments. Although some of the methods (Hopf et al., 2014; Ovchinnikov, Kamisetty and Baker, 2014) used small genomic distances to pair MSA sequences, this only worked in the case of bacterial sequences, where interacting paralogs could be assumed to be on the same operon. Those papers therefore only focused on predicting inter-chain interactions for protein complexes from bacterial species.

In the years leading up to the start of this thesis, meta-predictors and deep-learning methods had been increasingly used for the prediction of protein monomer contacts and had resulted in significantly better contact predictions for protein domains and protein chains. No papers at the time had extended the application of these meta-predictors or deep-learning methods to the problem of protein complex prediction.
The work in this thesis therefore aimed to combine coevolution-based predictions with deep-learning tools for the specific goal of performing de novo prediction of homodimers and heterodimers. The ability of these methods to accurately classify and predict complex contacts was explored, as well as the usefulness of using contact predictions for the purposes of ranking docking decoys to obtain near-native predictions.

Work in chapters 2-4 focused on exploring the application of CBMs and deep learning for the prediction of homodimer complex structures.

Work done in chapter 2 explored a CNN-based approach (CNNContactClassify) and a residual neural network approach (DMPContactClassify) to classify known homodimer contacts as inter-chain or intra-chain contacts. Both methods took in contact predictions from CBMs, as well as secondary structure predictions, solvent accessible surface area predictions, and residue information as inputs and used deep learning models to classify inter-chain vs. intra-chain residue-residue contacts. The results of the chapter showed that deep learning models could be used to some extent to differentiate between the two contact types, with about half of the tested homodimers having a MCC above 0.3 when classifying the two contact types. The results in chapter 2 demonstrate that at least for some homodimer types in the test set, the model was able to somewhat accurately differentiate between inter-chain and intra-chain contacts. The results in chapter 2 also demonstrated that certain factors contributed to the increased likelihood of accurate inter-chain contact prediction. First, structures with bigger interface to monomer ratios (i.e., the ratio of contacts involved in the interface to the number of contacts involved in the monomer subunits) tended to have higher MCCs. Dimers with large interaction interfaces are more likely to correspond to obligate or permanent protein dimers as opposed to transient or intermediate protein dimers. For this reason, it is usually easier to predict obligate dimers than transient dimers. Second, it was observed that better inter-chain contact prediction results were often observed for interfaces that were made up of interacting alpha-helices than for interfaces made up of interacting beta sheets or disordered regions. Finally, classification of interface residues for globular and rigid proteins tended to be more accurate than for more flexible proteins. This chapter was an important proof of concept to demonstrate that it was possible to differentiate the two contact types.
Work in chapter 3 explored several deep-learning models for the prediction of homodimer contacts. Unfortunately, the results of the chapter showed that the methods explored in the chapter were not able to differentiate between interface residues and non-interface residues. Overall, the models trained to perform simultaneous prediction of inter- and intra-chain contacts performed very poorly on the task of predicting inter-chain contacts. The best performing method for predicting inter-chain contacts used predicted inter-residue distances from DMPfold as an input to the DMP_CP model and resulted in much better inter-chain predictions. The best performing method therefore needed some previous information about potential monomer contacts to be able to better predict the interface contacts of the homodimers. Additionally, a significant correlation was found between the accuracy of the DMPfold predicted model (i.e., lower RMSD to native structures) and the precision of the inter-chain contact predictions from DMP_CP. This indicates that accurate prediction of the monomer subunit structure can be important to helping the deep-learning model predict inter-chain contacts. A few factors contribute to the accurate prediction of the monomer subunits of the homodimer complexes, and thereby the overall accuracy of the interface prediction. First, the overall secondary structure makeup of the monomer (i.e., if the monomer has several disordered regions) greatly affects the ability to accurately predict the monomer structure as proteins with more disordered regions are more difficult to accurately model. Second, the overall conformational change between the monomer and dimer versions of the protein also plays a role in the ability to accurately predict the monomer structure (proteins that undergo large conformational changes upon dimerization may be more difficult to predict). Finally, the depth of the multiple sequence alignments can greatly affect the accuracy of the monomer structure prediction. Since the overall accuracy of complex interface prediction depends highly on the quality of the monomer structure prediction, further work could focus on testing various monomer structure prediction methods and determining how the accuracy of the monomer structure predictions affects the overall prediction of homodimer complexes.

Chapter 4 used inter-chain contacts predicted using the DMPfold + DMP_CP method from Chapter 3 to rank docking decoys generated with ZDock. The results in the chapter showed that ranking using DMPfold + DMP_CP predicted contacts performed a bit worse than ZRank physics-driven ranking in ranking acceptable or better docking decoys at the top of ranking lists. In cases where the precision of predicted inter-chain contacts from
DMP_CP was higher than 0.3333, however, contact-based ranking often performed better than ZDock in ranking better-quality models at the top of the decoy ranking lists. These results demonstrate the importance of correctly identifying interface residue-residue contacts for the correct docking of protein complexes and shows that the ability to make precise inter-chain contact predictions (as explored in chapter 3) has a direct impact on the ability to rank near-native decoys at the top of ranking lists. These results could indicate one of the possible explanations for why AF2-multimer outperforms existing homodimer prediction tools. Since AlphaFold2 was shown to greatly outperform most existing methods in terms of monomer structure prediction in CASP13, it is possible that this improvement in monomer structure prediction improves the overall ability of the model to correctly predict interaction interfaces. Future work in this area could use AF2 to make predictions of monomer subunits of homodimers see whether the quality of monomer structure prediction correlates to the quality of AlphaFold2 homodimer predictions.

Work in chapters 5-7 focused on exploring tools for heterodimer structure prediction.

Chapter 5 involved benchmarking several existing tools for MSA sequence pairing, inter-chain contact prediction, and docking of heterodimer complexes. The results in this chapter demonstrated the importance of MSA pairing for the correct prediction of inter-chain protein contacts. In addition, similar to homodimers, accurate prediction of inter-chain contacts had a significant impact on the correct ranking of acceptable or better docking models and the results in this chapter demonstrate the importance of inter-chain contact precision for the correct ranking of docking decoys. Again, as was observed in chapters 3 and 4, results in chapter 5 demonstrated the importance of accurate monomer structures for protein-protein docking. This indicates that the accuracy of prediction of the monomer subunits of protein complexes is essential to the accurate docking of both homodimers and heterodimers. Docking decoys generated by docking DMPfold predicted structures generally resulted in much poorer complex structure predictions than decoys generated using crystal structures. Again, this chapter demonstrates the importance of high-quality predictions of constituent monomers to the improved prediction of heterodimer complex structures. As AF2 multimer also performs very well on heterodimer complex prediction, this could again be one of the reasons AF2 multimer is able to outperform most existing methods on the problem of heterodimer complex prediction and is often able to predict the correct orientation of dimer proteins. It is important to note,
however, that there are specialized cases where tools trained on specific families outperform AlphaFold multimer. One example is antibody docking, as AF2 multimer does not perform as well on as other antibody-specific prediction tools on the problem of antibody loop prediction (Evans et al., 2021; Abanades et al., 2022; Ruffolo et al., 2022).

Chapter 6 of this thesis explored the application of an MSA-embedding bidirectional GRU along with sequence embeddings and inverse covariance information for the prediction of heterodimer interface contacts. Although in theory, the biGRU methods could make predictions on unpaired multiple sequence alignments, the results in this chapter showed that the biGRU methods performed better on paired MSAs than unpaired MSAs for interface contact prediction. This shows that MSA pairing is still important for contact prediction even when using MSA embedding methods. Unfortunately, the overall inter-chain contact prediction precision of the methods trained in chapter 6 did not match the best inter-chain prediction precisions in chapter 5 (from DMPfold and DMPfold2). Further work could be done to determine whether incorporating a structure module and an iterative prediction process like those used in DMPfold and DMPfold2 could improve the overall heterodimer complex predictions of biGRU-based methods.

Finally, work in chapter 7 explored the application of a version of AlphaFold2 adapted for complex prediction (ColabFold) for the prediction of heterodimer complexes. In terms of the proportion of complexes that were predicted with acceptable or better dimer structures by CAPRI standards, AlphaFold seemed to perform better than most previously reported heterodimer complex prediction tools. Similar to the results in chapters 3-6, the quality of the monomer structure prediction was often correlated to the ability of AlphaFold 2 to correctly predict the complex structure. In addition, AlphaFold PAE scores seem have some correlation with the quality of the predicted complex models. This is useful because, for de novo heterodimer prediction, it may be possible to determine how reliable a AlphaFold prediction is based on the PAE maps. Future work needs to be done to do very rigorous benchmarking of AlphaFold2 multimer against other heterodimer prediction tools, and to study complexes for which AlphaFold2 does not give good predictions. For example, is already known that AF2 multimer does not outperform state of the art antibody structure prediction tools, for CDR loop prediction. Granular analysis of cases where AlphaFold does not give good complex predictions may give hints about what factors determine whether AlphaFold will be successful in complex prediction or not.
Overall, there were some different and common challenges that had to be solved for the prediction of homodimer complexes compared to heterodimer complexes. In the case of homodimers, the challenge of deconvoluting inter-chain contacts from intra-chain contacts was important to being able to correctly predict the complex structure. Here, being able to accurately model the structure of the monomer subunits was important to differentiating between the two contact types. In the case of heterodimers, the challenge of generating correctly paired, deep multiple sequence alignments was essential to being able to perform inter-chain contact predictions for heterodimers. The quality of the monomer structure prediction was shown in chapters 3-6 to be essential to the prediction of both homodimer and heterodimer complexes.

Finally, the nature of dimer complexes plays an important role in determining the ability of computational tools to perform accurate complex predictions. Obligate complexes, which tend to have large interface-to-monomer ratios and have more evolutionarily conserved interacting residue pairs tended to be easier to predict than transient dimer complexes, which often have smaller interface-to-monomer ratios and fewer conserved interacting residue pairs. In addition, dimers that undergo large conformational changes upon binding tend to be much more difficult to predict than dimers that have smaller differences between the bound and unbound versions of the constituent monomers. Overall, the most challenging complexes to predict tend to be complexes that are more transient in nature, and complexes that undergo large conformational changes upon binding. These are the complexes that future methods can focus on improving.

8.2 Limitations and Future Directions

There are a number of limitations in the analysis done in this PhD thesis. This section outlines these limitations, as well as future research directions that might address some of these limitations.

First, for the benchmarking of existing tools used in chapter 4 (docking of homodimer complexes) and chapter 5 (heterodimer contact prediction and docking), the benchmarking involved using code and webservers that were available at the time the work was done, and that were either openly accessible or accessible with academic
credentials. Since the work in chapter 5 was completed, there have been a number of updates made to existing docking tools, including HADDOCK (now HADDOCK3), ClusPro (now ClusPro 2.0), and InterEvDock (now InterEvDock2). The benchmarking work done in this thesis is therefore not up to date with the newest versions of the docking servers and may therefore not reflect improvements in these methods since that time. In addition, the benchmarking of docking protocols was limited to tools that ranked within the top 10 tools of the CAPRI predictor category from the 7th CAPRI evaluation (Wodak, Velankar and Sternberg, 2020). For this reason, tools that were not evaluated during the CAPRI assessment, or did not rank in the top 10 CAPRI were not included in the benchmarking. One such method that was not benchmarked here was RosettaDock (Lyskov and Gray, 2008; Marze et al., 2018). Comparison of RosettaDock 3.2 and RosettaDock4.0 (Marze et al., 2018) showed that RosettaDock had comparable performance to ClusPro, ZDock, and HADDOCK. The comparison involved in the paper by Marze et al., however, compared performance based on self-reported results from papers evaluating each method. This meant that the methods had been evaluated on different docking datasets. For this reason, it is difficult to know how RosettaDock compares to the methods evaluated in chapter 5. Future work may therefore involve comparing the newer versions of docking servers to RosettaDock, all evaluated on the same benchmark set similar to the work done in chapter 5.

Second, for the evaluation of AlphaFold complex prediction done in chapter 7, the evaluation was done using an AlphaFold2 method modified to predict heterodimer complexes (Baek, 2021b). At the time the evaluation work was done for chapter 7, the code for AlphaFold multimer had not yet been released. The method evaluated in chapter 7, which was part of the ColabFold suite of tools (Mirdita et al., 2022), was different from the AF2-multimer method in that it did not involve retraining AlphaFold2 on protein complexes as was done for AF2-multimer. For this reason, the results in chapter 7 may not reflect the SOTA in AlphaFold protein complex prediction. In addition, since a threshold of 25% sequence identity was used to filter chains overlapping with the AlphaFold2 training set, it is possible that monomers in the same fold as some of the multimer protein chains were in the benchmark set. This means that although AlphaFold2 was not trained on multi-chain complexes, the results in chapter 7 could have benefitted from some protein chains having overlaps in terms of structure with the AlphaFold2 training set. Future work needs to be done using AF2-multimer and ensuring more
rigorous cross-validation in relation to the training set. Further work also needs to be done to determine cases where AF2-multimer does not successfully predict protein dimers, to determine reasons what factors determine the success of AF2-multimer. This could be important to developing tools to address the cases of protein complex prediction where AF2-multimer is not able to produce accurate predictions.

Third, since the work in chapter 5 was completed, a number of new inter-chain contact prediction and complex docking tools that use deep learning for rigid body docking have been released. These include the DeepComplex server (Quadir, Roy, Soltanikazemi, et al., 2021) and EquiDock (Ganea et al., 2021) have been released. DeepComplex uses DNCON2_inter (Quadir et al., 2021), to predict inter-chain contacts for homodimers and heterodimers using deep CNNs. The DeepComplex method uses inter-chain contact predictions from DNCON2_inter to build dimer complex structures using CNS. EquiDock uses independent SE(3)-equivariant graph matching models to perform end-to-end rigid protein docking. The EquiDock method is one of the first methods to use SE(3)-equivariant graph neural networks to perform protein docking. Future follow-up work should compare these newer protein-protein docking techniques to previously established docking methods, such as those explored in chapter 5 and RosettaDock. In addition, it would be useful to know how methods developed chapter 3 (DMPContactClassify and DMPfold+DMPContactClassify) compare to the performance of DNCON2_inter for the prediction of homodimer inter-chain contacts (Quadir, Roy, Halfmann, et al., 2021). A similar comparison, following up on the comparison done in section 5.4, could be done comparing the best heterodimer inter-chain contact predictions from DMPfold2 to DNCON2_inter.

Overall, although the work in this thesis demonstrates the benefits of using coevolution-based methods combined with deep-learning for the prediction of dimer complex contacts, further work needs to be done to determine current state of the art in complex prediction, as well as remaining challenges for the prediction of homo- and hetero-oligomer complexes.
8.2 Final Conclusions

The goal of this thesis was to explore the use of coevolution-based methods and deep-learning techniques for the prediction of protein dimer contacts, and to determine the impact of these contact predictions on complex prediction tasks such as protein dimer docking. The work outlined in this thesis demonstrates a number of important insights into the problem of \textit{in silico }protein docking.

It is shown that the use of deep-learning models such as CNNs and ResNets improves the ability of computational techniques to differentiate between inter-chain and intra-chain contacts in homodimers, thereby addressing an important problem in previously published coevolution-based methods for homodimer interface prediction. In addition, results in chapter 3, 4, and 5 clearly demonstrate the importance of accurate monomer structure and contact predictions on subsequent protein complex prediction. Results in chapter 2 show a correlation between the quality of predicted monomer structures, and the precision of methods such as DMPContactPredict to accurately predict inter-chain homodimer contacts. The results in chapters 4 and 5 demonstrate the direct impact of improved inter-chain contact prediction precision on ability to rank docking decoys to get near-native complex structure predictions. Overall, improved monomer structure and contact prediction has a direct impact on the quality of protein complex prediction.

Chapter 7 demonstrates that although AlphaFold2 improves on previously used protein docking methods for heterodimer complex structure prediction, there remain complexes that AlphaFold2 was not able to predict correctly.

Overall, the work in this thesis shows the great potential of combined coevolution-based prediction and deep-learning for the improved prediction of near-native structures for dimer complexes. Several important advances have been made using deep learning for protein complex prediction, and the work done in this thesis demonstrates where some of these advances have been useful, and where further development could be done to improve the overall ability of the field to perform accurate and reliable prediction of protein complexes. Future challenges that need to be addressed include exploring specific cases
of protein dimer docking where current SOTA, like AF2-multimer, fail to make correct predictions and developing techniques to better predict those complexes.
Convolutional layers are the primary building blocks of convolutional neural networks (CNNs). Convolutional layers involve performing convolutions over an input layer. A convolution involves taking the dot product of a kernel, $K$, made up of learnable parameters and a restricted portion of an input receptive field as shown in figure A.1. During a forward pass through a convolutional layer, the kernel is passed over the horizontal and vertical dimensions of an input at a stride, $s$, to produce a 2-dimensional activation map, which has the same depth as the input. The stride determines the units by which a convolution shifts over the input.
Figure A 1: Kernel and receptive field: 3x3 kernel (square in bold) passes over the receptive field at a particular stride, $s$, to produce activation (yellow). Here, the kernel size is 3x3 and the depth of the kernel is the same as the depth of the input (i.e., the number of channels in the input).

Convolutional layers are powerful because they are able to capture spatial dependencies in two dimensional inputs.

**Dilations of Convolutional Layers**

Dilated convolutions “inflate” the convolution kernel by inserting holes between kernel elements, thus expanding the kernel in the horizontal and vertical directions. A dilation rate ($d$) indicates how much a kernel is widened. A convolution with a dilation rate equal to one corresponds to a normal convolution. A dilation rate greater than one involves skipping $d-1$ units per convolution input, i.e., inserting $d-1$ spaces inserted between kernel elements. Figure A-2, taken from (Cui et al., 2019), demonstrates the effect of dilations on the convolution kernel.
Figure A 2: Kernel dilation: Impact of dilation rate on the size of convolution kernel (blue). Image obtained from (Cui et al., 2019)

Dilated convolutions allow exponential expansion of the receptive field of a kernel without loss of resolution or coverage of the inputs (Cui et al., 2019). This means the field of a convolutional kernel can be expanded without having to increase the number of parameters of a model. Dilated convolutions allows the detection of fine details in inputs by processing inputs in higher resolutions and allows for a broader view of inputs to capture more contextual information (Cui et al., 2019).

A3. Contribution Statement

All analysis work and results generated in this PhD were generated by Nikita Desai unless otherwise indicated.

The original code for CNNContactClassify was written by David Jones in Lasagne and was adapted into PyTorch by Nikita. CNNContactPredict was adapted by Nikita Desai from the PyTorch CNNContactClassify code. The code for DMPContactClassify was adapted from the original DeepMetaPSICOV code written in Pytorch by David Jones, with modifications made to adapt the network to homodimer classification. Modifications to
original DMP code for homodimer prediction were made by Nikita Desai. All training strategies used to train models were designed by Nikita.

Modified version of MI-IPA used to create MSA pairings used in chapter 5 and 6 was adapted from the original MI-IPA (Bitbol et al., 2016) code by Nikita Desai.

The biGRU model for heterodimer MSA embedding in chapter 6 was written by David Jones. ResNet model added to the biGRU model was adapted from the DMPfold2 code by Nikita Desai. Code to add sequence embeddings as inputs to biGRU inputs was written by Nikita.

Training of all models developed for this thesis was performed by Nikita Desai. All other tools and methods tested in this thesis were developed by others as indicated by references. All tools and methods analysed here were run by Nikita.

**Software availability**

Code developed over the course of this PhD is available through the GitHub account of the author [https://github.com/ndesai1](https://github.com/ndesai1). Code repositories are available for the following tools: CNNContactClassify, DMPContactClassify, CNNContactPredict, DMPContactPredict, biGRU + ResNet methods
Bibliography


Baek, M. (2021a) ‘Adding a big enough number for “residue_index” feature is enough to model hetero-complex using AlphaFold’, 20 July. Available at: https://twitter.com/minkbaek/status/1417538291709071362.


Symposium on Performance Analysis of Systems and Software (ISPASS), Santa Rosa, CA, USA: IEEE, pp. 55–64. Available at: https://doi.org/10.1109/ISPASS.2017.7975270.


Moriwaki, Y. (2021) ‘AlphaFold2 can also predict heterocomplexes’, 19 July. Available at: https://twitter.com/Ag_smith/status/1417063635000598528.


Skwark, M.J. et al. (2014) ‘Improved Contact Predictions Using the Recognition of Protein Like Contact Patterns’, PLoS Computational Biology. Edited by G. Wei, 10(11), p. e1003889. Available at: https://doi.org/10.1371/journal.pcbi.1003889.


Wang, S. *et al.* (2016) ‘Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model’. Available at: https://doi.org/10.48550/ARXIV.1609.00680.


