

IMPROVING PROTEIN STRUCTURE PREDICTION USING AMINO ACID CONTACT & DISTANCE PREDICTION

by
SHUANGXI JI

A thesis submitted to
the University of Birmingham
for degree of
DOCTOR OF PHILOSOPHY (Sc, PhD)

School of Biosciences
College of Life and Environment Sciences
the University of Birmingham
April 2019

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

ABSTRACT

With more and more protein sequences generated, one of the most pressing tasks in bioinformatics has become to interpret these data. This thesis concerns how to predict the 3D structure of a protein relying on its sequence only, which is a long-standing problem in computational biology. A commonly adopted intermediate step for this task is to predict pairwise amino acid contacts based on the query sequence. Due to the simplicity of the current algorithms, which include statistical models and machine learning techniques, the accuracy of contact prediction is still low for many proteins. Also, these available algorithms are unable to predict amino acid distances (distance longer than contact). Thus, the lack of high quality and enough geometry constraints make it difficult for 3D structure prediction for many proteins. To deal with the current limitations of amino acid constraint and structure prediction, a state-of-the-art deep neural network based amino acid contact & distance prediction algorithm, DeepCDpred, is proposed in this thesis. For a given query protein sequence, the geometry constraints predicted by DeepCDpred are fed into a Rosetta *ab initio* modelling protocol for protein structure prediction. In addition, a neural network based method is proposed to evaluate the quality of predicted structures.

The accuracies of amino acid contact and distance predictions, the quality of structure predictions and the accuracy of confidence score predictions were evaluated by a test set of 108 protein chains whose experimental structures are known. Any sequence in the test

set shares no greater than 25% sequence identity with any sequence in the training set, which was used to train DeepCDpred. The accuracy of amino acid contact predictions of DeepCDpred is just slightly worse than a newly published method, RaptorX; but exceeds all others mentioned in this thesis. Thanks to the predicted extra distance constraints and the Rosetta *ab initio* modelling protocol, the structure prediction quality based on the algorithms proposed in this study is better than that from the RaptorX server. A blind test, which was done with a yet to be released protein, was also used to validate the effectiveness of DeepCDpred. The protein classes of structures predicted with amino acid contact constraints from MetaPSICOV (the amino acid contact predictor, which DeepCDpred is most often compared with in this thesis), are analysed and compared to the predictions based on contact constraints from DeepCDpred, and also to the predictions based on both contact and distance constraints from DeepCDpred. An online server, <http://proteincoevolution.bham.ac.uk>, is programmed and released to make the proposed methods for amino acid contact and distance predictions, structure prediction and structure confidence prediction accessible to average users, and it is expected beneficial to the research community.

ACKNOWLEDGEMENTS

Foremost, I would like to thank my first supervisor Peter J. Winn. This thesis, as well as the four years of study and research would not be possible without his help and guidance. Owing to his substantial investment of effort and time, I have a much better understanding about protein data mining today than the beginning I came to his group. I truly appreciate the time I had to learn from him and was a member of his group. I also want to thank Dr. Sam Butterworth, my second supervisor, for his ideas on the sub-project of ligand matching, also advices on other topics in the PhD project. Dr. Eva Hyde, my internal assessor, I appreciate her corrections and comments on my three-month, nine-month and twenty-one-month reports, as well as for providing a blind test opportunity to my protein structure prediction algorithm.

Also, I must thank current and former members of Peter Winn's group. Especially I would like to thank Bhima Auro for introducing me to the previous work he had done on the building of the three-way cross-reference and the solvent accessibility work and for taking time to answer my questions, and Tugce Oruc for her contribution of successfully writing a python script that makes it possible to include metagenomics sequence into my amino acid contact/distance prediction algorithm, as well as her sacrifice of time on interpreting some results in my project. I would still like to thank these people for their help during my research and life in the University of Birmingham: Andrew Edmondson, a specialist from IT Services of Birmingham University, Edwin Aponte Angarita, a PhD student from Peter Winn's group, Rohit Farmer, a graduated PhD student from Peter Winn's group,

Dr. Jan-Ulrich Kreft, from the Centre for Computational Biology, Feng Dong and Joanna Summers from Kreft's group.

This thesis is dedicated to my wife, Mengyang Jia, who has given me endless support and patiently listened to my complaints and encouraged me, and my parents, for their genuine love and support.

CONTENTS

Abstract	i
Acknowledgements	iii
List of Figures	xi
List of Tables	xvii
Abbreviations	xx
1 Introduction	1
1.1 Proteins	1
1.2 Public Protein Databases	5
1.2.1 Protein Sequence Database	6
1.2.2 Protein Structure Database: the Protein Data Bank	7
1.2.3 Protein Family Database	8
1.3 Protein Data Analysis and Bioinformatics Tools	9
1.3.1 Protein Sequence Alignment	9
1.3.2 Protein Secondary Structure Prediction	11
1.3.3 Protein Three-dimensional Structure Prediction	12
1.4 The Scope and Contributions of This Thesis	14
1.4.1 The Scope of This Thesis	14
1.4.2 Contributions of This Thesis	16
2 Background	18
2.1 Overview of This Chapter	18
2.2 Coevolution	19
2.3 Coevolution and Amino Acid Contact	20

2.4	Review of Using Coevolution to Predict Amino Acid Contact & Protein Structure and to Study Protein Function	22
2.4.1	Using Coevolution to Predict Amino Acid Contact	22
2.4.2	Using Coevolution to Predict 3D Structure of Protein	27
2.4.3	Using Coevolution to Study Protein Function	28
2.4.4	Summary of This Section	29
2.5	Protein Sequence Alignment and Homology Detection	30
2.5.1	From Amino Acid Substitution Models to BLAST	32
2.5.2	Profile-Sequence Comparison and PSI-BLAST	34
2.5.3	Profile HMM-Sequence Comparison	35
2.5.4	Profile-Profile Hidden Markov Models	38
2.6	Review of Amino Acid Coevolution Analysis	41
2.6.1	Local Statistics Models	43
2.6.2	Disentangling Directly Coupled Positions from the Network of Indirectly Correlated Positions	44
2.6.3	Phylogenetic Bias Correction	45
2.6.4	Global Statistical Models	47
2.6.5	Maximum Entropy Principle	49
2.6.6	Sequence Reweighting	54
2.6.7	Mean Field DCA	56
2.6.8	Sparse Inverse Covariance Estimation (PSICOV)	58
2.6.9	Pseudo-likelihood Maximization DCA	62
2.6.10	Machine Learning Based Predictors	67
2.7	Feature Selection and Machine Learning	73
2.7.1	Secondary Structure Prediction	73
2.7.2	Other Features	77
2.7.3	Machine Learning and Artificial Neural Network	78
2.8	Protein Structure Prediction	84
2.8.1	Introduction	84
2.8.2	Template-Based Modelling	87
2.8.3	Template-free Modelling	92
2.8.4	CASP	98
2.8.5	Protein Structure Comparison	99
2.9	Summary of This Chapter	102
3	Method Overview and Materials	104
3.1	Overview of This Chapter	104
3.2	Aims of This Thesis	105
3.3	Structures of The Methods Proposed In This Thesis	105

3.3.1	Definitions of Amino Acid Contact and Amino Acid Coupled at A Distance	106
3.3.2	The Structure of DeepCDpred	107
3.3.3	The Structure of DeepCDpred_AbInitio	110
3.3.4	The Structure of The Confidence Prediction Model	113
3.4	Materials	114
3.4.1	Data	114
3.4.2	Software	120
3.4.3	Computing Resources	123
3.5	Features and Feature Vector	123
3.5.1	Features	124
3.5.2	The Feature Vectors	127
4	Method Development	131
4.1	Overview of This Chapter	131
4.2	The Development of DeepCDpred	131
4.2.1	Introduction	131
4.2.2	Technical Details of DeepCDpred	134
4.2.3	Training Process and Parameter Optimization of DeepCDpred	135
4.2.4	Explorations to Improve the Performance of DeepCDpred	139
4.2.5	Contact and Distance Prediction Assessment	142
4.2.6	Realization of DeepCDpred	144
4.2.7	Contribution Analysis of the Features of DeepCDpred	145
4.2.8	Differences between DeepCDpred and MetaPSICOV	147
4.3	The Development of DeepCDpred_AbInitio	149
4.3.1	Introduction	149
4.3.2	Rosetta <i>Ab Initio</i> Modelling with the Top 1.5L Contact Predictions	151
4.3.3	Rosetta <i>Ab Initio</i> Modelling with the Contact Predictions Determined by Score Cut-off	152
4.3.4	Rosetta <i>Ab Initio</i> Modelling with Both Contact Predictions and Distance Predictions from DeepCDpred	152
4.3.5	Rosetta <i>Ab Initio</i> Modelling Protocol	157
4.4	The Development of the Method to Estimate the Quality of Predicted Structures	158
4.5	Comparisons of Contact and Structure Prediction Between DeepCDpred, RaptorX and NeBcon	160
5	Results	161
5.1	Overview of This Chapter	161
5.2	Percentage of Amino Acid Contact in Real Proteins	163

5.3	Characterisation of the Training/Validation Set and the Test Set	164
5.4	Comparisons of the Speed and of the Inter-residue Contact Prediction Accuracy between PSICOV and QUIC	169
5.5	Amino Acid Contact and Distance Predictions of DeepCDpred	173
5.5.1	Parameter Optimization of DeepCDpred	174
5.5.2	Comparison of Contact Prediction Accuracies Between a Single Network and the Average of Four Networks	175
5.5.3	Comparison of Contact Prediction Accuracies between DeepCDpred and Other Algorithms and Distance Prediction Accuracy of DeepCDpred	176
5.5.4	Examples of Contact Prediction Comparisons Between MetaPSICOV and DeepCDpred and Distance Prediction of DeepCDpred	177
5.5.5	Feature Contribution Ranking Analysis for DeepCDpred	180
5.5.6	Limitations of the Contact Prediction Selection Method of “Top L Terminology”	183
5.5.7	Accuracies of Contact and Distance Predictions of DeepCDpred Based on Score Cut-offs	185
5.6	Protein Structure Prediction	187
5.6.1	The Variation between Different <i>ab initio</i> Predictions by Rosetta	188
5.6.2	Comparison of Structure Prediction Based on the Top-ranked 1.5L Contact Predictions from DeepCDpred and MetaPSICOV	190
5.6.3	Comparison of Structure Predictions Based on Predicted Contacts That Have the Same Contact Prediction Accuracy	195
5.6.4	Summary of Comparisons of Structure Predictions Based on Contacts Predicted by MetaPSICOV and DeepCDpred	207
5.6.5	Comparisons of Structure Predictions Between Using the Top-ranked 1.5L DeepCDpred Contacts and Using the Combination of the Top-ranked 1.5L DeepCDpred Contacts & Score Cut-off Selected DeepCDpred Distances	209
5.6.6	Comparison of Structure Predictions between Using DeepCDpred Contacts Selected by Score Cut-off and Using Both DeepCDpred Contacts & Distances Selected by Score Cut-off	214
5.6.7	Summary of Structure Prediction Based on Contacts and Distances from DeepCDpred	220
5.7	TM-score Predictions	224
5.8	Examples of Some of the Best Protein Structure Predictions	232
5.9	A Blind Test	235
5.10	Comparisons of Amino Acid Contact and Structure Predictions among DeepCDpred, RaptorX and NeBcon	241
5.11	Improving the Accuracies of the Amino Acid Contact and Distance Predictions of DeepCDpred by Using Metagenomics Data	248

5.12	Improving the Accuracy of Amino Acid Contact Predictions of DeepCDpred by Using Networks with 5 Hidden Layers	250
5.13	Online Server: PROTEINCOEVOLUTION.BHAM.AC.UK	252
6	Discussion	254
6.1	Sequence Alignment Methods	256
6.2	Available Sequences	257
6.3	Interpretation of Long-distance Couplings and Improvement to Distance Prediction	258
6.4	The Test Set of DeepCDpred	262
6.5	Protein Model Selection	263
6.6	Feature Optimization	266
6.7	The Training Strategy of DeepCDpred	268
6.8	The Comparisons of Contact and Structure Predictions Between DeepCDpred and Other Algorithms	270
6.9	Other Machine Learning Algorithms and Best Model Selection Strategies	272
6.10	The Online Server: PROTEINCOEVOLUTION.BHAM.AC.UK	277
7	Conclusion	279
	Publications	282
	Appendix A Supplementary Materials	284
	Appendix B Supplementary Results	286
B.I	Sequence Identity Distribution Between the Test Protein Chains & the Training/Validation Protein Chains	286
B.II	Parameter Optimization of DeepCDpred	287
B.III	Comparisons of Contact Prediction Accuracy, Model File Size and Average Contact Prediction Speed between DeepCDpred, SVM and Random Forest	289
B.IV	Structure Selected by Lowest Rosetta Energy VS. True Best Structure	291
B.V	Accuracy of Amino Acid Contact Prediction After Adopting A New Feature Vector	293
B.VI	Raw Data of Figure 5.17, Figure 5.19 and Figure 5.21	294
	Appendix C Training/Validation, And Test Set	301
	Appendix D Protein Classification	315

LIST OF FIGURES

1.1	The four levels of structure of a protein	3
1.2	Depiction of dihedral angles ϕ and ψ in a polypeptide chain (a), and an example of Ramachandran plot that shows all of the ϕ - ψ angles within a protein (b).	4
2.1	Some structural constraints of a protein are recorded in the alignment of homologues, from which they can be inferred and used for the structure prediction.	21
2.2	Cost of DNA sequencing has decreased dramatically in the past more than ten years, especially since 2008.	26
2.3	A section of the MSA of Pfam PF00103.	31
2.4	The way of building a multiple sequence alignment by BLAST or PSI-BLAST.	35
2.5	Diagram of profile HMM for protein homologous sequence detection.	36
2.6	An illustration of the indirect coupling caused by the transfer of direct couplings.	45
2.7	Diagram of the overlap of the correct contact predictions (a), and incorrect contact predictions (b) by PSICOV, EVfold (mfDCA) and CCMpred (plmDCA) from 1050 predictions.	68
2.8	A three-hidden-layer deep neural network.	84
2.9	The gap between the numbers of protein sequences and structures are becoming larger over time.	86
2.10	From protein sequence to protein structure.	87
2.11	Diagram of a protein folding funnel.	95
3.1	The overall structure of DeepCDpred.	108
3.2	Overall diagram of DeepCDpred_AbInitio pipeline, including the step of DeepCDpred for inter-residue contact and distance predictions, and the step of structure prediction by using the obtained geometry constraints from the former.	112

3.3	The architecture of the three-layer neural network model chosen for predicting TM-score.	113
3.4	Diagram of selecting the test set, training set and validation set of DeepCDpred.	116
3.5	Diagram of the feature vector for each residue pair of stage 1 networks of DeepCDpred (graph a) and the concatenation of feature vectors of all the residue pairs (inputs) of all the proteins in the training/validation set to form the training/validation data (graph b).	129
4.1	Diagram of the development of DeepCDpred.	133
4.2	Diagram of the development of DeepCDpred_AbInitio.	150
4.3	The distribution of inter-residue distance in terms of sequence separation (measured in amino acid number, aa no.).	155
4.4	Beta strand pairing in an anti-parallel sheet.	157
4.5	Diagram of the development of the structure prediction quality evaluation model.	159
5.1	The percentage of contacting amino acid pairs in 250 unrelated protein chains in terms of one amino acid separation and five amino acid separation (graph a); the distribution of the number of amino acids of these chains (graph b).	163
5.2	The distributions of globular and membrane proteins in the training/validation set and the test set.	165
5.3	The distributions of protein stoichiometry in the training/validation set, the test set of DeepCDpred, as well as the 2,957 protein chains, in which the training/validation data were chosen from.	165
5.4	Examples of the six protein classes.	167
5.5	Protein class distributions of the 1,066 proteins chains in the training/validation set (graph a), and of the 108 protein chains in the test set (graph (b)).	168
5.6	Comparisons of contact prediction accuracy and speed between PSICOV and QUIC.	170
5.7	The comparison of amino acid contact prediction accuracies between the optimized one-hidden network and the optimized two-hidden-layer network.	174
5.8	Contact prediction accuracy calculated from averaging output scores from four networks vs contact prediction accuracy calculated with the output score from each individual network for both stage 1 and stage 2.	176

5.9	Comparison of contact prediction accuracies between DeepCDpred and previous algorithms shown in a; b is the accuracy of DeepCDpred distance predictions in three bins.	177
5.10	Comparison of the top L/3 predictions between MetaPSICOV and DeepCDpred for three example proteins.	179
5.11	Examples of distance bin 8-13 Å prediction from DeepCDpred.	180
5.12	Accuracy of contact prediction changes with each type of feature removed from the stage 1 networks of DeepCDpred.	181
5.13	Results of contact (a) and distance (b) predictions of DeepCDpred shown in the form of accuracy versus neural network output score cut-off.	186
5.14	Two independent sets of prediction calculations for the 108 protein test set give little variation on average.	189
5.15	The accuracy of structure predictions with the top 1.5L contacts predicted by MetaPSICOV, compared to those by DeepCDpred. The selected models are those with the lowest Rosetta energy score.	191
5.16	The top 1 models of four selected proteins in the test set are aligned to their respective experimental structures.	194
5.17	The comparison of structure prediction accuracies between MetaPSICOV and DeepCDpred after the outliers were removed.	195
5.18	Finding the equivalent minimum scores for DeepCDpred and MetaPSICOV based on the same average accuracies of the top-ranked 1.5L contact predictions (71.8%, predicted by DeepCDpred in graph a; 62.8%, predicted by MetaPSICOV in graph b) of the 108 test proteins.	197
5.19	The comparison of structure prediction accuracies between feeding MetaPSICOV predicted contacts (score \geq 0.56) and feeding top DeepCDpred predicted contacts (score \geq 0.40) into the same Rosetta <i>ab initio</i> protocol.	198
5.20	Superimpositions of the top 1 models (the model with the lowest Rosetta energy) from three proteins (left side of the dash line) for which the folds are better-predicted with the contacts (score \geq 0.40) from DeepCDpred than with the contacts (score \geq 0.56) from MetaPSICOV, and the top 1 model from one protein (right side of the dash line) that is more accurately predicted with the contacts (score \geq 0.56) from MetaPSICOV than with the contacts (score \geq 0.40) from DeepCDpred with the respective experimental structures.	201
5.21	The comparison of structure prediction accuracies after the outliers were removed.	202

5.22	The comparison of structure prediction accuracies between feeding MetaPSICOV predicted contacts (score \geq 0.40) and feeding top DeepCDpred predicted contacts (score \geq 0.26) to the same Rosetta <i>ab initio</i> protocol, best predictions were picked out by the lowest Rosetta energy score.	203
5.23	Superimpositions of the top 1 model of three proteins (left side of the dash line) which are more accurately predicted with the contacts (score \geq 0.26) from DeepCDpred than with the contacts (score \geq 0.40) from MetaPSICOV, and the top 1 model of one protein (right side of the dash line) that is more accurately predicted with the contacts (score \geq 0.40) from MetaPSICOV than with the contacts (score \geq 0.26) from DeepCDpred with the respective experimental structures.	206
5.24	The comparison of structure prediction accuracies after the outliers were removed.	207
5.25	Comparisons of the quality of structure predictions based on the three contact selection strategies for each of the two compared algorithms.	208
5.26	The comparison of the structure prediction accuracies between feeding the top 1.5L DeepCDpred predicted contacts and feeding the combination of top 1.5L DeepCDpred predicted contacts & neural network score selected distances to the same Rosetta <i>ab initio</i> protocol.	210
5.27	Superimpositions of the top 1 model of three proteins (left side of the dash line) which are more accurately predicted with the top-ranked 1.5L contacts plus distances from DeepCDpred than with only the top-ranked 1.5L contacts from DeepCDpred, and top 1 model (right side of the dash line) of one protein that is more accurately predicted with only the top-ranked 1.5L contacts than with the contacts plus distances, with the respective experimental structures.	213
5.28	The comparison of structure prediction accuracies after the outliers were removed.	214
5.29	The comparison of structure prediction accuracies between feeding the contacts (score \geq 0.40) from DeepCDpred and feeding the contacts (score \geq 0.40) & distances from DeepCDpred to the same Rosetta <i>ab initio</i> protocol.	216

5.30	Superimpositions of the top 1 model of three proteins (left side of the dash line) which are more accurately predicted with contacts (score \geq 0.40) plus distances from DeepCDpred than with contacts (score \geq 0.40) from DeepCDpred, and the top 1 model of one protein (right side of the dash line) that is more accurately predicted with the contacts than with the contacts plus distances, with the respective experimental structures.	219
5.31	The comparison of structure prediction accuracies after the outliers were removed.	220
5.32	Comparison of TM-scores of the model (selected by the lowest Rosetta energy score) for each of the 108 test proteins predicted based on the two combinations of DeepCDpred contacts and distances constraints.	221
5.33	Distribution of the TM-scores of the top 1 model (selected by the lowest Rosetta energy score) predicted based on the two combinations of DeepCDpred contact plus distance constraints.	222
5.34	Real TM-score of the top 1 model (selected by the lowest Rosetta energy score) versus the Nf for each of 108 proteins in the test set.	223
5.35	Predicted TM-scores versus real TM-scores from the structure predictions based on the contact and distance constraints from DeepCDpred for the proteins in the test set of DeepCDpred.	225
5.36	Protein class distribution of the protein chains whose predicted vs real TM-scores are between the lines of $y = x \pm 0.1$	227
5.37	Predicted TM-scores versus real TM-scores from the structure predictions based on the contact constraints from MetaPSICOV for the proteins in the test set of DeepCDpred.	228
5.38	Protein class distribution of the protein chains whose predicted vs real TM-scores are between the lines of $y = x \pm 0.1$	231
5.39	TM-score versus Nf for the six example proteins and the blindly tested protein.	232
5.40	Comparisons of structure predictions of six proteins based on DeepCDpred predicted constraints (left) and MetaPSICOV predicted constraints (right).	234
5.41	Amino acid contact map of Q9FLY6 predicted by DeepCDpred.	237
5.42	The comparison of the average distance distribution from the predicted top 5 models of Q9FLY6 and the contact strength from the contact prediction of the same protein.	237
5.43	Overlaying the predicted top 1 model of Q9FLY6 to the experimental structure (coloured as grey) with the low variation region residues coloured as red and the high variation region residues coloured as magenta.	238

5.44	Overlaying the predicted top 1 model (coloured as blue) of Q9FLY6 to the experimental structure (coloured as red).	240
5.45	Comparison of the contact prediction accuracies of RaptorX, DeepCDpred and NeBcon for eight test proteins based on the top-ranked 1.5L contact predictions.	242
5.46	Comparisons of quality of structure predictions between DeepCDpred and RaptorX based on variant contact/distance constraints and structure simulation protocols.	244
5.47	Superimposition between the top 1 predicted model of the blind test protein Q9FLY6 with the experimental structure.	245
5.48	Six examples of structure predictions from RaptorX server.	246
5.49	The comparison of the accuracies of amino acid contact and distance predictions between adding metagenomics data to UniProtKB and using UniProtKB only.	249
5.50	Predictions from a two stage neural network trained by using the same feature vector but with five hidden layers.	251
5.51	The home page of the amino acid contact & distance prediction and protein structure prediction server, proteincoevolution.bham.ac.uk.	253
6.1	Two building block types of ResNet.	274
B.1	Distribution of pairwise sequence identity between the 108 test protein chains and the 1066 training/validation protein chains of DeepCDpred.	287
B.2	Amino acid contact prediction accuracy comparison between the optimized one-hidden network and the optimized two-hidden-layer network.	288
B.3	Comparisons of amino acid contact prediction accuracy, model file size and average prediction speed between DeepCDpred, an SVM and a random forest model.	290
B.4	The difference between the best structure selected by the lowest Rosetta energy score and the true best structure among the 100 candidates which is picked out by comparing with the experimental structure.	292
B.5	Contact prediction accuracy comparison between the five-hidden-layer neural networks of DeepCDpred and the modified five-hidden-layer neural networks with new feature vector.	294

LIST OF TABLES

2.1	Comparisons between PconsC, PconsC2 and MetaPSICOV.	71
3.1	Features of DeepCDpred.	125
3.2	Features of the model confidence estimation method.	127
4.1	Training function names and the corresponding optimization algorithms in the neural network toolbox of MATLAB.	138
5.1	The protein classes of the outliers in the comparison of contact prediction accuracies between PSICOV and QUIC.	172
5.2	Comparison of contact prediction accuracies between between MetaPSICOV and DeepCDpred.	177
5.3	How the removal of one feature changes the accuracy of the stage 1 network of DeepCDpred.	182
5.4	Four examples show the limitations of selecting the top-ranked 1.5L contact predictions.	185
5.5	The classes of the proteins whose structures are better-predicted with the top-ranked 1.5L contacts of DeepCDpred than with the top-ranked 1.5L contacts from MetaPSICOV, or vice versa.	192
5.6	The protein classes of the proteins whose structures are better-predicted with the contacts (score \geq 0.40) from DeepCDpred than with the contacts (score \geq 0.56) from MetaPSICOV, or vice versa.	199
5.7	The classes of the proteins whose structures are better-predicted with the contacts (score \geq 0.26) from DeepCDpred than those with the contacts (score \geq 0.40) from MetaPSICOV, or vice versa.	204
5.8	The classes of the proteins whose structures are better-predicted with the top-ranked 1.5L contacts plus distances from DeepCDpred than with the top-ranked 1.5L contacts from DeepCDpred only, or vice versa.	211

5.9	The classes of the proteins whose structures are significantly better-predicted with the contacts (score \geq 0.40) & distances from DeepCDpred than with the contacts (score \geq 0.40) from DeepCDpred, or vice versa.	217
5.10	Comparison of the distance prediction accuracy between proteins which appear in the top three rows and the last row of Table 5.9.	218
5.11	The classes of the proteins whose top 1 models' TM-scores are poorly predicted with the TM-score prediction network model. The top 1 models are predicted with the contacts and distances from DeepCDpred.	226
5.12	The classes of the proteins whose top 1 model's TM-scores are poorly predicted by the TM-score prediction network model. The top 1 models are predicted with the top-ranked 1.5L contacts from MetaPSICOV.	229
5.13	Information about the blind test protein.	236
5.14	Comparisons between the top 1 model quality of Q9FLY6 and the top 1 model quality of the β proteins in the test set of DeepCDpred whose Nf values are similar to that of Q9FLY6.	240
5.15	PDB ID list of the 29 protein chains with Nf values less than 64.	248
A.1	Mappings of converting amino acids to numbers.	284
A.2	The positions of the nine beta strands in the experimental structure of Q9FLY6.	285
B.1	Raw TM-scores of the boxplots shown in Figure 5.17. Each row corresponds to the same protein; different rows represent different proteins.	294
B.2	Raw TM-scores of the boxplots shown in Figure 5.19.	296
B.3	Raw TM-scores of the boxplots shown in Figure 5.21. Each row corresponds to the same protein; different rows represent different proteins.	298
C.1	PDB ID list of the 250 protein chains in the amino acid contact percentage calculation.	302
C.2	PDB ID list of the 221 protein chains in the speed and contact prediction accuracy comparisons between PSICOV and QUIC.	303
C.3	PDB ID list of the test set of DeepCDpred.	304
C.4	PDB ID list of the training/validation set of DeepCDpred.	304
C.5	PDB ID list of the training/validation set of Feature Contribution Analysis.	309
C.6	PDB ID list of the training/validation set of Feature Contribution Analysis.	312

C.7	PDB ID list of training/validation set of the TM-score prediction network.	314
D.1	PDB ID list of the training/validation set of DeepCDpred.	315
D.2	Protein classes of the 108 protein chains in the test set of DeepCDpred.	327

ABBREVIATIONS

PDB	Protein Data Bank
NMR	Nuclear Magnetic Resonance
MSA	Multiple Sequence Alignment
PFAM	Protein Families database
PSSM	Position specific scoring matrix
BLAST	Basic Local Alignment Search Tool
PSIBLAST	Position Specific Iterated BLAST
DELTA-BLAST	Domain Enhanced Lookup Time Accelerated BLAST
HHblits	HMM-HMM Based Lightning fast Iterative sequence Search
SCOP	Structural Classification of Proteins
HMM	Hidden Markov Model

MUSCLE	Multiple Sequence Comparison by Log-Expectation
PSP	Protein Structure Prediction
SS	Secondary Structure
DSSP	Dictionary of Protein Secondary Structure
PSIPRED	PSI-blast based secondary structure PREDiction
SCA	Statistical Coupling Analysis
PCA	Principal Component Analysis
MI	Mutual Information
APC	Average Product Correction
DI	Direct Information
DCA	Direct Coupling Analysis
mpDCA	Message Passing DCA
nmfDCA	Naive Mean Field DCA
plmDCA	Pseudo-likelihood Maximisation DCA
FN	Frobenius Norm
DeepCNF	Deep Convolutional Neural Fields

SPIDER 2	Structural Property prediction with Integrated DEep neuRal network 2
ANN	Artificial neural network
CASP	Critical Assessment of Techniques for Protein Structure Prediction
MC	Monte Carlo
MD	Molecular Dynamics
RMSD	Root Mean Square Deviation
BP	Backpropagation
MLP	Multi-Layer Perceptron
SVM	Support Vector Machine
QUIC	Quadratic Approximation for Sparse Inverse Covariance Estimation
PSICOV	Protein Sparse Inverse COVariance
MetaPSICOV	Meta Protein Sparse Inverse COVariance
DeepCDpred	Deep Contact & Distance Prediction
RL	Reinforcement Learning
MSE	Mean Squared Error
SGD	Stochastic Gradient Descent

BFGS Broyden-Fletcher-Goldfarb-Shanno

ROC Receiver Operating Characteristic

CHAPTER 1

INTRODUCTION

1.1 Proteins

One cannot emphasize too much the importance of proteins to life. They are the workhorse molecules in all organisms and carry out a great variety of functions: some act as building blocks, together with other molecules to make the structure of cells; some work as catalysts, accelerating almost all the biochemical reactions in cells, which are many orders of magnitude faster than catalysts that humans can devise; some help cells to communicate, to move, to respond to stimuli; some others interact with DNA to regulate programs of development.

Proteins are long sequences formed out of 20 naturally occurring amino acid residues that

adopt a unique three-dimensional (3D) structure in native physiological conditions ([Anfinsen 1973](#)). It is these amino acids that result in the diversity of the protein world. These amino acids share a basic structure with an amino ($-\text{NH}_2$) and a carboxyl ($-\text{COOH}$) group connecting to the same carbon atom, called the C_α ; they are thus α -amino acids. While the third covalent bond of the C_α connects to a hydrogen atom, the fourth substituent is the *side-chain*, which causes different chemical and physical properties. All amino acids have chiral α atoms, except for glycine, which has a hydrogen side chain.

Proteins are usually described at four levels, known as primary, secondary, tertiary, and quaternary structure ([Figure 1.1](#)). The complexity seen between the secondary and tertiary structural levels can be further characterized by the super-secondary elements and domains. In the native state, the primary structure folds into local secondary structures including α helix and β strand. Secondary structures can be further packed into tertiary structure due to van der Waals forces, the hydrophobic effect, hydrogen bonding and electrostatic interactions between the atoms. Many proteins contain more than one polypeptide chain. Protein quaternary structure refers to the arrangement and interaction of multiple polypeptide chains.

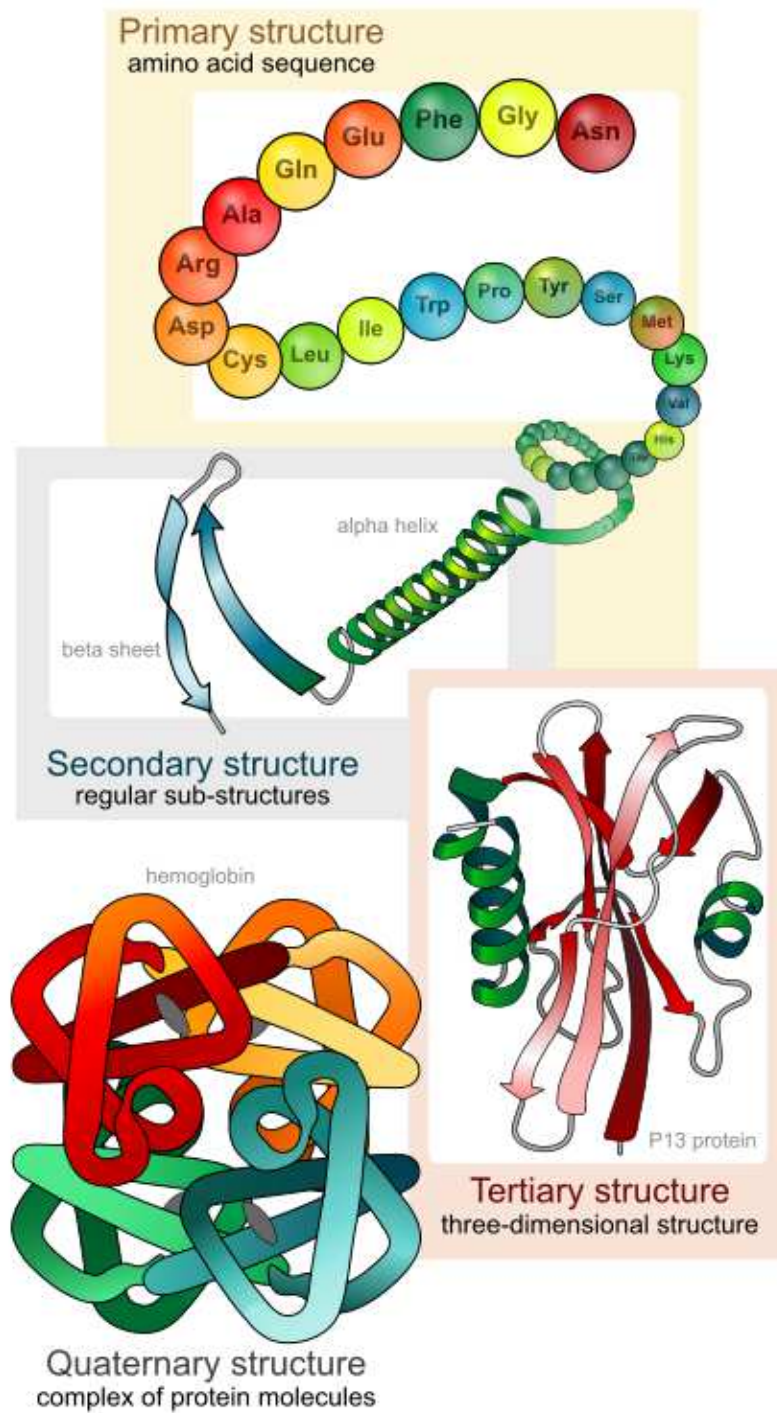


Figure 1.1. The four levels of structure of a protein (reproduced from ([Wikipedia/Protein Structure 2018](#))).

In a protein structure, adjacent amino acids are connected by a peptide bond. The planarity of the peptide bond makes the dihedral angle, called ω , close to 0 degree (*cis*) or more often, 180 degrees (*trans*); ω is formed by the four nearby atoms $C\alpha_{(-1)}-C_{(-1)}-N-C\alpha$ on the backbone (Figure 1.2a). Besides ω , there are another two dihedral angles defined in a protein backbone, ϕ and ψ . ϕ is formed by the four atoms $C_{(-1)}-N-C\alpha-C$; and ψ is formed by $N-C\alpha-C-N_{(+1)}$ (Figure 1.2a). Unlike ω , the angles of ψ and ϕ are flexible; different combinations of ψ and ϕ define different local conformations of the protein structure.

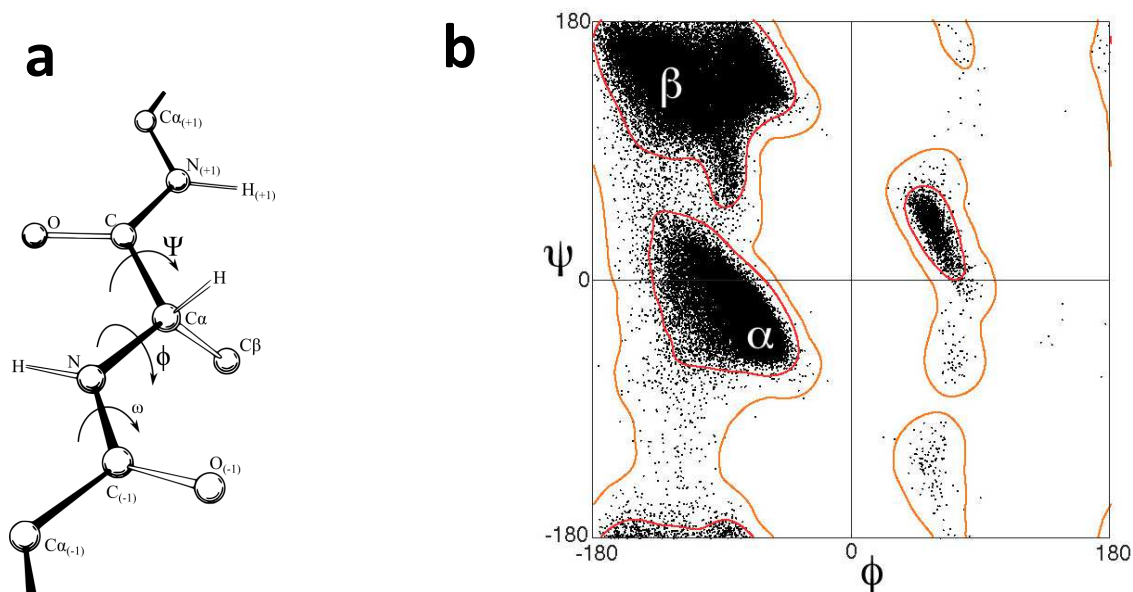


Figure 1.2. Depiction of dihedral angles ϕ and ψ in a polypeptide chain (a), and an example of Ramachandran plot that shows all the ϕ - ψ angles within a protein (b). (graph a is reproduced from ([Wikipedia/Dihedral Angle 2018](#)) and graph b is reproduced from ([Wikipedia/Ramachandran Plot 2018](#))).

The diagram of Ramachandran plot visualizes all the possible values of ψ and ϕ angles within a protein structure. This plot is named after the biophysicist G.N. Ramachandran

(1922 - 2001) ([Wikipedia/Ramachandran 2018](#)). A typical Ramachandran plot is shown in Figure 1.2b. In the plot, both ψ and ϕ range from -180° to 180° . Some combinations of the two angles are not possible, due to the steric hindrance; some other combinations are energetically favourable, as shown in the black areas on the plot. The ψ and ϕ degrees of the two common secondary structures α helix and β strand can be clearly seen on the plot. For α helix, average values of ψ and ϕ are about -57° and -47° , as shown in the middle-left black area. This is the right-hand α helix, which is the common type; another less common type is the left-hand α helix, and the favourable combination of ψ and ϕ is shown in the upper-right area. For a parallel β sheet, the average values are about -119° and $+113^\circ$, and an antiparallel β sheet, -139° and 135° . The combinations of ψ and ϕ of these two types of β sheet are shown in the upper-left area of the plot.

From the secondary structure of a protein, one can define its fold or topology. Two proteins sharing a common fold do not necessarily require the two structures are identical; instead, it requires their secondary structures have the same composition and the same arrangement ([Lo Conte et al. 2000](#)). Thus two proteins sharing the same fold may have different loop structures.

1.2 Public Protein Databases

Various protein databases were involved in the studies of this thesis, which include protein sequence, structure and family databases.

1.2.1 Protein Sequence Database

Thanks to the fast development of gene sequencing techniques, protein sequences are accumulating at an ever-increasing speed. Protein sequence databases usually provide functions or tools to search, compare and analyse these sequences. There are mainly two protein sequence databases that are widely used, UniProt ([Uniprot 2018](#)) and NCBI *nr* ([NCBI 2018](#)).

UniProt is composed of three core databases: UniProtKB (includes manually curated and reviewed sequences in SwissProt, and unreviewed, automatically annotated sequences in TrEMBL), UniParc, and UniRef ([UniProt 2015](#)). By September 2017, there were more than 550,000 protein sequences in UniProtKB/SwissProt, and about 90 million sequences deposited in the collection of UniProtKB/TrEMBL. Unlike UniProtKB, UniParc (UniProt Archive) does not provide annotations. It just stores all publicly available non-redundant protein sequences ([Wu et al. 2006](#)). In UniParc, all identical sequences over the full length are merged into one sequence entry. UniRef (UniProt Reference Clusters) provides clustered sequences from UniProtKB and selected UniParc records ([Suzek et al. 2015](#)). UniRef100 database combines identical sequences into a single UniRef entry ([Mirdita et al. 2017](#)). UniRef90 is created by clustering UniRef100 sequences such that each cluster contains the sequences that have at least 90% pairwise sequence identity and 80% overlap with the longest sequence ([Mirdita et al. 2017](#)). UniRef50 is built by clustering UniRef90 sequences that have at least 50% sequence identity with each other and 80% overlap with

the longest sequence in every cluster (Mirdita et al. 2017; Uniref 2018). The server of UniProt provides cross-references to other protein sources, such as PDB structures and protein families (e.g., Pfam).

NCBI (National Center for Biotechnology Information) also provides a rich sequence database, *nr*, which contains protein sequences from GenBank translations, as well as the sequences from other databases, such as PDB, UniProt/SwissProt, etc.

Both UniRef (e.g., UniRef90) and *nr* are commonly used as the sequence search sources for programs in the BLAST family, such as BLAST and PSI-BLAST.

1.2.2 Protein Structure Database: the Protein Data Bank

The Protein Data Bank (PDB) is the major source of macromolecular 3D structures in the world, including proteins and nucleic acids (Rose et al. 2015). By the end of 2016, there were about 125,000 structure entries deposited in PDB. These structures are mainly resolved by X-ray crystallography and NMR spectroscopy. One can search a structure on the PDB server (PDB 2018) based on its four-character identifier (pdb id). The structure can be viewed and analysed on the server or downloaded to the local computer to be viewed and analysed by programs such as PyMol (DeLano 2002) and Jmol (Jmol 2018). There are two types of structures available in PDB, the asymmetric unit and the biological unit. Like UniProtKB, the PDB server also provides cross-references to other databases, such as the protein family databases (e.g., Pfam, SCOP and CATH), and protein sequence

databases (e.g., UniProt). One can view annotations of a protein structure from other sources.

1.2.3 Protein Family Database

Protein family databases are useful to assign functions to uncharacterized proteins (Louie et al. 2008). A protein family derived from sequences is usually classified by a profile, which is built from a multiple-sequence alignment (MSA) (Xu and Xu 2004). A good example is Pfam, which builds MSAs of protein domain families by using the profile hidden Markov model (profile HMM) (Bateman et al. 2004; Finn et al. 2014, 2010; Punta et al. 2012). Protein domains are usually sections of protein sequence or structure that could evolve independently and “*have an independent function or contribute to the function of a multidomain protein in cooperation with other domains*” (Vogel et al. 2004).

The structure-structure comparison yields protein structure families. CATH (Class, Architecture, Topology and Homologous superfamily) and SCOP (Structural Classification of Proteins) are the two most important protein structure classification schemes (Csaba et al. 2009). Both of them classify protein structures which are generally deposited in PDB, in a hierarchical manner. CATH sorts protein structures into four major levels – class, architecture, topology and superfamily (Orengo et al. 1997); SCOP organizes protein structures into class, fold, superfamily and family, an alternative four-level classification (Lo Conte et al. 2000). Regarding to how to classify proteins, CATH uses an

automated process; while, SCOP mainly relies on expert knowledge ([Hadley and Jones 1999](#)).

1.3 Protein Data Analysis and Bioinformatics Tools

To deal with the overwhelming abundance of protein data, a collection of bioinformatics tools and analysis methods have emerged. In the following, those related to this work are briefly introduced; more detailed information about them can be found in the next chapter.

1.3.1 Protein Sequence Alignment

Sequence alignment is a fundamental concept in protein data analysis, since it is the basis of many further studies. Probably the basic and most common requirement of protein data analysis is to search for similar sequences from a certain database to a query sequence. The most widely used computational tool for this task is BLAST (Basic Local Alignment Search Tool) ([Altschul et al. 1990](#)) and its variants – PSI-BLAST (Position-Specific Iterative BLAST) ([Altschul et al. 1997](#)) and DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST) ([Boratyn et al. 2012](#)). HMMER (Hidden Markov Model Tool Suite) is widely used as well ([Finn et al. 2011](#); [Johnson et al. 2010](#); [Soding 2005](#)). It probabilistically models the local interactive constraint between neighbouring residues

and was shown to be more sensitive and accurate at finding similar sequences for a query, but slower in speed, compared to BLAST and PSI-BLAST ([Madera and Gough 2002](#)). The recently developed HHblits (HMM–HMM–Based Lightning fast Iterative sequence Search) algorithm was even proved to outperform HMMER in remote protein homologue detection, as well as being faster than the latter ([Remmert et al. 2012](#)). Both CDD (Conserved Domains Database) and Pfam use HMMs to detect known protein domains in a query sequence.

An accurate multiple sequence alignment (MSA) is a critical step in both phylogenetic analysis and amino acid coevolution analysis. Suboptimality of the MSA could reduce the accuracy of the resultant phylogenetic tree ([Ogden and Rosenberg 2006](#)). Section 2.6 of the next chapter reviews both local and global statistical models for amino acid coevolution analysis. All of these approaches start from the MSA of the target (query) protein sequence. Alignment errors could lead to erroneous observations of correlated mutation ([Dickson et al. 2010](#)). One of the most widely used approaches for aligning multiple sequences is a heuristic method known as the progressive sequence alignment ([Ari and Goldman 2005](#)). This method involves the construction of a coarse-guide tree in the first stage. The tree determines how the sequences are added to the alignment – the most similar sequences are added at the beginning and more distant sequences are added successively. The final alignment from this method is not guaranteed to be globally optimal, notably when errors are made at any time in expanding the MSA; they are then propagated through to the final alignment. The two best-known programs using

this method are CLUSTALW (Weighted UPGMA (Unweighted Pair Group Method with Arithmetic Mean) CLUSTER analysis of the pairwise ALignments) (Thompson et al. 2002, 1994) and MUSCLE (MULTiple Sequence Comparison by Log-Expectation) (Edgar 2004).

Tools in the BLAST family make an MSA by aligning all significant hits to the query. HMM builds an MSA by matching each hit to a profile HMM by using the Viterbi algorithm (Eddy 1998). As a comparison, HHblits aligns all found HMMS by an HMM-HMM comparison algorithm to make an MSA (Remmert et al. 2012; Soding 2005).

1.3.2 Protein Secondary Structure Prediction

Protein secondary structure (SS) represents the local conformations of amino acids. It is formed by hydrogen bonds between N–H and C=O on the backbone. Ss have a regular geometry that allows only certain values of dihedral angles (as already shown in Figure 1.2). Different amino acids have different natural propensities for a given secondary structure element. For instance, methionine, alanine, leucine, glutamate and lysine prefer helical conformations (α helix); aromatic residues (tyrosine, phenylalanine and tryptophan) and β -branched amino acids (threonine, valine and isoleucine) tend to be on β strands (Chou and Fasman 1974).

Knowing the secondary structure of a protein allows a general structural classification from α protein, β protein, $\alpha + \beta$ protein and α/β protein (Lo Conte et al. 2000). Moreover, secondary structure plays a critical role in discovering how proteins fold (Zhou and Karplus

1999). The accuracy of protein secondary structure prediction could impact the accuracy of protein 3D structure prediction (Fischer and Eisenberg 1996; Rohl et al. 2004) and amino acid solvent exposure prediction (Heffernan et al. 2016). Predicted secondary structures have also proved to be useful in protein sequence alignment (Soding 2005).

A detailed description of SS prediction approaches can be found in Chapter 2 (Subsection 2.7.1).

1.3.3 Protein Three-dimensional Structure Prediction

Secondary structure provides a coarse-grained picture of a protein. However, in order to understand the precise function of a protein, its 3D structure is required. In 2017, experiments, such as nuclear magnetic resonance (nmr) and X-ray crystallography, are still the main approaches to solve the structure of proteins. However, due to the labour, time and money costs involved, only a very limited number of proteins have structure data in PDB as compared with the exponentially growth of the number of protein sequences in UniProt.

Thus, it is a great temptation to predict the 3D structure of a protein solely relying on its primary sequence. In fact, it has long been recognised to be achievable in theory (Anfinsen 1973). Unfortunately, according to the calculations performed by Levinthal (Zwanzig et al. 1992), there are too many conformations even for a common protein to

find the native state with random searches. Thus, predicting protein structure remains one of the biggest open research issues in computational biology.

Great progress has been made in this field, and many algorithms have been proposed in the last three decades. These algorithms can, in principle, be divided into two categories, template-based and template-free modelling. Methods in the former group, given a target sequence, identify evolutionarily related (homology modelling) or unrelated (threading) templates with solved structure, and then construct structure models based on the frames provided by these templates. Template-free modelling methods do not rely on template structures. Among them, pure *ab initio* modelling methods, which do not use any prior structural information, have been developed. However, the expensive computing cost limits this method only to very short chains (< 90 residues) to achieve a reasonable accuracy ($< 4 \text{ \AA } C_\alpha$ RMSD) at present (Kallberg et al. 2012). Other *ab initio* methods that assemble short structural fragments or use geometry constraints obtained from coevolution analysis to constrain the building of a model structure, become more and more successful (Ovchinnikov et al. 2017b), with results of RMSD better than $3 \text{ \AA } C_\alpha$ (Ovchinnikov et al. 2016).

In Section 2.8 of Chapter 2, some of the popular protein structure prediction methods will be introduced in detail.

1.4 The Scope and Contributions of This Thesis

1.4.1 The Scope of This Thesis

The main content of this thesis introduces the author's work on solving the long-standing problem of protein structure prediction relying only on sequence. The method developed in this thesis has two steps. In the first step, a deep neural network based algorithm (DeepCDpred, Deep Contact & Distance Prediction) was proposed to predict inter amino acid contacts and distances. In this algorithm, features, such as coevolutionary couplings, the predicted protein secondary structure, sequence profile calculated from the target sequence or the MSA of the target sequence were used. In the second step, the geometry constraints predicted from DeepCDpred were fed into a Rosetta *ab initio* modelling (Fleishman et al. 2011; Leaver-Fay et al. 2011; Rohl et al. 2004) protocol to generate protein structures. The putative best-predicted structure was then selected as the one with the lowest Rosetta energy score. In order to assign a confidence value to the predicted structure to estimate how reliable it is, a TM-score prediction algorithm was developed.

In the next chapter (Chapter 2), the concept of coevolution, the relationship between coevolution and amino acid contact, and previous studies that employ coevolution for amino acid contact, structure and function predictions are introduced or reviewed. Since sequence alignment and homology detection are the necessary steps for amino acid coevolution analysis, the methods in these fields are then briefly introduced. After these,

both local and global statistical models for amino acid coevolution analysis are reviewed (as mentioned in the above paragraph, the coevolutionary couplings calculated from these models were used as features of DeepCDpred). The development of machine learning, especially neural networks, is then presented. Other features used in the input of DeepCDpred, such as the secondary structure prediction, are reviewed. In the final section of this chapter, the methods of protein structure prediction and structure similarity comparison are reviewed.

In Chapter 3 and Chapter 4, the materials and model development of DeepCDpred are introduced, respectively. The materials include the descriptions of the training set, validation set and test set of DeepCDpred, the software that generate features for DeepCDpred, and the structures of the methods proposed in this study. Chapter 4 includes the implementation of DeepCDpred, the evaluation of the performance of DeepCDpred, feature contribution analysis, how to select contact and distance predictions from DeepCDpred for protein structure prediction, the Rosetta *ab initio* modelling protocol used in this study, and the building of the TM-score prediction model.

In Chapter 5, the main results of this thesis are displayed, including: (a) the accuracy of contact predictions of the test proteins of DeepCDpred, as compared with that predicted by other algorithms; (b) the accuracy of distance predictions of DeepCDpred; (c) the contribution analysis of the features used in DeepCDpred; (d) the accuracy of protein structure predictions of the test proteins with the contacts predicted by DeepCDpred, as compared with that based on the contact predictions from MetaPSICOV (Meta Protein

Sparse Inverse COVariance) (Jones et al. 2015); (e) the accuracy of structure predictions of these proteins based on both the contact and distance predictions from DeepCDpred; (f) the result of TM-score predictions; (g) the structure prediction of DeepCDpred in a blind test; (h) the comparisons of contact and structure predictions between DeepCDpred and the two recently published methods (i.e., NeBcon (He et al. 2017) and RaptorX (Wang et al. 2017b)); (i) with metagenome data employed as the homologue search source, the accuracies of contact and distance predictions by DeepCDpred; (j) the accuracy of contact predictions of the test proteins by a deeper version of DeepCDpred which replaces the two-hidden-layer and sigmoid activation functions in the hidden layers with five-hidden-layer and ReLU (Nair and Hinton 2010) activation functions; (k) the introduction to the online server (www.proteincoevolution.bham.ac.uk) designed and programmed to implement DeepCDpred and the protein structure prediction in this study.

The Chapter 6 discusses the limitations and the possible ways to improve this work, and Chapter 7 concludes this study.

1.4.2 Contributions of This Thesis

The main contributions of this thesis include: (a) the development of a new deep neural network based amino acid contact & distance prediction algorithm, DeepCDpred; (b) implementing a Rosetta *ab initio* modelling protocol that uses the predicted contact & distance constraints for protein structure prediction; (c) proposing a TM-score prediction

method that provides a confidence for the structure prediction; (d) testing two geometry constraint selection methods to see which one leads to better structure prediction; (e) designing and programming an online website that implements DeepCDpred, protein structure prediction and TM-score prediction.

CHAPTER 2

BACKGROUND

2.1 Overview of This Chapter

In this chapter, the background of coevolution and its relationship with amino acid contacts are introduced first, followed by the review of the previous studies using amino acid coevolution to predict amino acid contacts, protein structure and function. The methods commonly used in the process of inferring coevolutional couplings from the target protein sequence are briefly introduced, which include protein sequence homology detection, local/global statistical models of coevolution analysis, protein secondary structure prediction, machine learning and feature selection. The final section of this chapter reviews methods of protein structure prediction, discusses three representatives of structure modelling algorithms (MODELLER, I-TASSER and Rosetta) and two approaches of structure similarity measurement (RMSD and TM-score).

2.2 Coevolution

Coevolution is a prevalent biological phenomenon that exists at species, organism, and molecular levels, and “*is a fundamental component of the theory of evolution*” (de Juan et al. 2013). It refers to the coordinated changes of multiple biological entities under selective pressures, typically to maintain or to refine functional interactions among them (de Juan et al. 2013).

The concept of coevolution can be traced back to Darwin’s *On the Origin of Species* (1859), in which he mentioned the evolutionary interaction between orchids and pollinators. The formation of the term of ‘coevolution’ was introduced in 1964 by Ehrlich and Raven (Ehrlich and Raven 1964). A widely accepted definition of coevolution is ‘reciprocal evolutionary change in interacting species’ (Lovell and Robertson 2010; Thompson 1994). It implies that a change in one species could alter selection pressure on another species. Conversely, the change that selection pressure caused in the second species could change the selection pressure of the first species (Lovell and Robertson 2010). Examples of coevolution from paired species include predator and prey, parasite and the host.

As for coevolution at the molecular level, a mutation of a residue at one site could change the fitness to a function of the protein which applies selection pressure to change a related residue to the first one (structural nearby or functional related). The related residue in response to this pressure could in turn affect the evolution of the residue at the first site.

Chakrabarti *et al.* ([Chakrabarti and Panchenko 2010](#)) conducted a large-scale study of coevolving site identification by 803 protein families. The authors found 15% active sites are coevolving with other sites; the number is changed to 11%, 11% and 9% for functionally important, protein binding and ligand binding sites, respectively. More discussions of the mechanisms of coevolution can be found in ([Lovell and Robertson 2010](#)) and ([Pazos and Valencia 2008](#)).

Within the literature, there are several synonyms for coevolution, including covariation and correlated mutation. In order to avoid confusion, coevolution and correlated mutation are mainly used in this thesis. The difference between them is that coevolution refers to the biological phenomenon, but correlated mutation is used when the coordinated changes of residues at multiple sites (positions) in an MSA are being discussed. For simplicity and efficiency of calculation, researchers have mainly focused on pairwise coevolution so far. In this thesis, all the words of ‘coevolution’ and ‘correlated mutation’ refer to the interaction between two entities, except the ones in the brief introduction to the SCA (statistical coupling analysis) in this chapter.

2.3 Coevolution and Amino Acid Contact

Amino acid coevolution is suggestive of compensatory substitutions that occur between coupled residues (e.g., those in close proximity or acting together at binding sites) due to the folding, structural, or functional constraints of a protein or a protein complex

(Chakrabarti and Panchenko 2010; Gobel et al. 1994; Neher 1994; Shindyalov et al. 1994; Taylor and Hatrick 1994). It can result in correlated columns in the MSA of homologous protein sequences. Homologous sequences can be considered as a record of the natural sampling of the sequence space available to folded functional proteins. By inverting the observation of covarying positions, structural or functional interdependencies between amino acids can be inferred from patterns of correlated mutations within the MSA. Importantly, the concept of amino acid coevolution provides a direct link between sequence and 3D structure and can be turned into a protein structure predictive method (Figure 2.1).

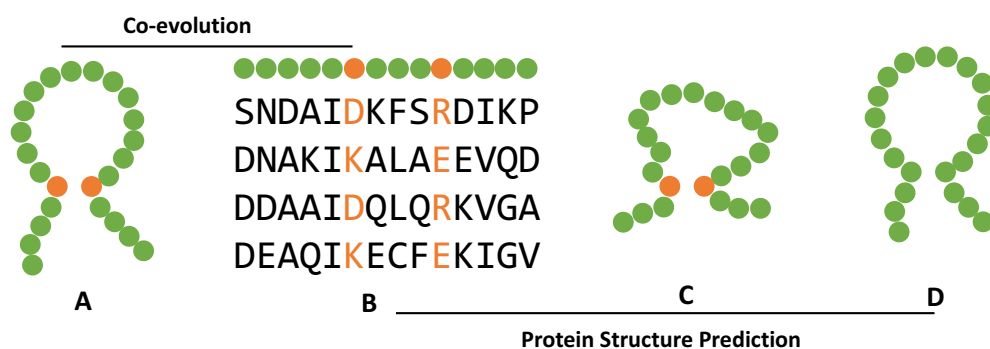


Figure 2.1. Some structural constraints of a protein are recorded in the alignment of homologues, from which they can be inferred and used for the structure prediction. From A to B, coevolutionary pressure between physically interacting amino acid residues (orange circles in A) in the 3D structure of a protein leaves a visible record of amino acid correlated mutation in the MSA (two orange columns in B). The inverse problem of inferring direct coevolutionary couplings from the alignment (from B to C) could be achieved by several algorithms, including the global statistical model introduced in this chapter. Once evolutionary couplings are determined (orange circles in C), they can be used to predict the unknown 3D structure of a protein (D) from a set of sequences alone.

2.4 Review of Using Coevolution to Predict Amino Acid Contact & Protein Structure and to Study Protein Function

2.4.1 Using Coevolution to Predict Amino Acid Contact

Before introducing how to use coevolution analysis to predict inter-amino-acid contacts (hereinafter referred to as amino acid contacts for simplicity), it is necessary to make it clear how an amino acid contact is defined and classified. In the literature, a widely used definition is to choose a cut-off of 8\AA between the C_β atoms (or C_α in the case of glycine) of the two residues ([Ekeberg et al. 2013](#); [He et al. 2017](#); [Jones et al. 2012, 2015](#); [Skwark et al. 2013](#); [Wang et al. 2017b](#)). This definition is also employed by the Residue-Residue prediction category of the Critical Assessment of protein Structure Prediction (CASP) competition ([Monastyrskyy et al. 2014](#)).

Amino acid coevolution analysis is mainly a mathematical process that tries to find co-varying positions based on an MSA. Since coevolving positions have been shown to be much more likely to be spatially proximal or just in contact in the protein structure than by chance, a substantial effort has been invested in developing more and more advanced statistical algorithms to infer coevolutionary signals during the past more than twenty years ([de Oliveira et al. 2016](#); [Goh et al. 2000](#); [Jones et al. 2015](#); [Kosciolek and Jones](#)

2016; Martin et al. 2005; Shindyalov et al. 1994; Wang et al. 2017b). The inferred inter-residue amino acid contacts have further aided protein structure prediction (de Oliveira et al. 2016; Hopf et al. 2012, 2014; Jones et al. 2015; Marks et al. 2011, 2012a,b; Morcos et al. 2011).

The first algorithm using evolutionary information to predict inter-residue contact was introduced by Gobel *et al.* (Gobel et al. 1994). It and other early proposed approaches were originally developed to detect pairs of positions in an MSA that have similar amino acid substitution patterns (Gobel et al. 1994; Neher 1994; Taylor and Hatrick 1994). Substitution patterns could be recognized based on an amino acid substitution matrix (e.g., BLOSUM62), and the similarity between them can be evaluated by a linear correlation (e.g., Pearson correlation coefficient). The common feature of these approaches is that they all treat each pair of positions as independent from the other pairs in the MSA being considered. Thus, they are called local statistical models, which are different from the recently developed approaches, the global statistical models. The early models only achieved very limited success in predicting inter-residue contacts – only 20% to 40% of all predicted contacts are correct (Burger and van Nimwegen 2008; Fariselli et al. 2001; Pollastri and Baldi 2002).

The following reasons explain why amino acid contact inferred from coevolution is not easy.

(a). Correlation might not be a good indicator of direct coevolutionary signals; a typical

example is that two distant residues correlate only because they are both in contact with a third residue. (b). Some contacting residues are too conserved to show sensible variations, in which case only a low correlation signal may be detected. (c). MSAs might have only a few sequences so that the correlated mutation signal cannot be inferred efficiently; (d). A similar situation is that MSAs might have lots of sequences but all of them are very similar. (e). Sequencing is biased toward the organisms that are of research's interest, which potentially leads to biased coevolution signals. (f). In homo-dimer proteins, it is hard to distinguish intra-protein coevolution signals from those due to inter-protein contacts. (g). Last but not least, coevolution signals may not necessarily originate from amino acid contacts but from protein functional constraints.

However, the adoption of global statistical models and machine learning techniques in recent years has made contact prediction far more accurate than the earlier attempts. The global statistical models assume that each sequence in the MSA of the target sequence is a sample of a multivariate probability distribution, and by maximizing the Shannon entropy of the distribution with the constraints of the first-order and second-order observed amino acid frequencies from the MSA, the direct coupling between each pair of residues in the target sequence can be obtained ([Marks et al. 2011](#)) (Section 2.6). Machine learning techniques take coevolutionary couplings predicted from these global models, as well as other protein primary and secondary structure related features as inputs and output a final contact score for each pair of residues in the target sequence (Subsection 2.6.10). All these methods can effectively disentangle the direct pairwise couplings from the background

phylogenetic noise (see Section 2.6 for details).

Predicted contacts are then used as geometry constraints to predict protein 3D structure through the provision of *a priori* structural information by using *ab initio* modelling. Such information can sharply reduce conformation search space and computational complexity, and thus make it possible to predict protein structure on a personal computer. Owing to the development of high throughput gene sequencing technologies and the decreasing cost of sequencing in the past more than twenty years (Figure 2.2), more and more organisms have been sequenced. There are a large number of genome sequencing projects, approximately 10,691 completed and 47,118 ongoing (JGI 2018) by the date of July 1st, 2017, which have resulted in the knowledge of more than 16,000 protein families (Finn et al. 2016). Many of these protein families contain from 10^3 to 10^5 homologous sequences. The vast number of available homologous sequences could improve the amino acid coevolution-based inter-residue contact prediction and the devising of protein structure prediction algorithms for both globular and membrane proteins (Hopf et al. 2012; Marks et al. 2011; Ovchinnikov et al. 2017b).

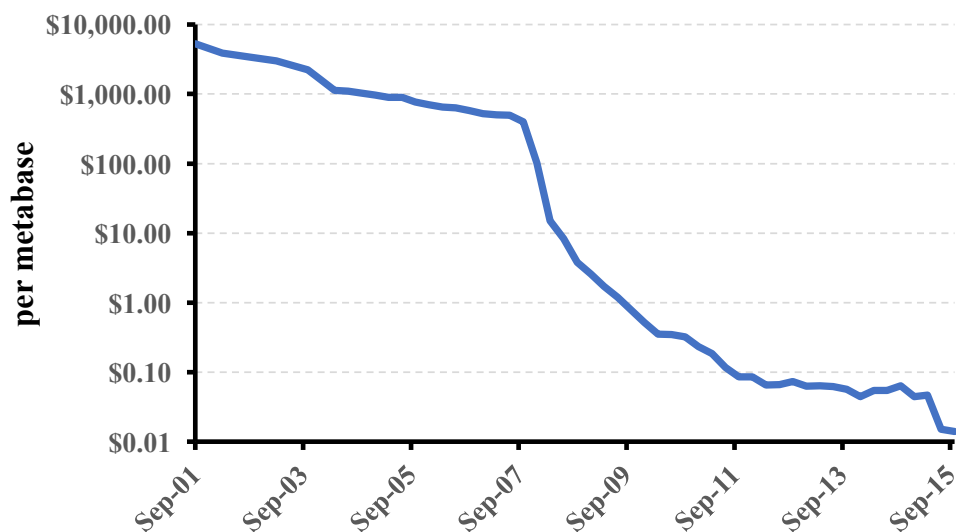


Figure 2.2. Cost of DNA sequencing has reduced dramatically in the past more than ten years, especially since 2008. This graph is produced based on the data downloaded from <https://www.genome.gov/sequencingcostsdata/> (last accessed: September 1st, 2017).

Predicted amino acid contacts are not only helpful for protein model generation but can also be used in many other steps within the process of protein structure prediction. Due to the fact that contacts are structural features, they are more conserved than sequences and can thus help to identify remote structural templates (Ovchinnikov et al. 2017b). This might enable homology modelling in cases where templates cannot be identified by sequence information alone. Additionally, predicted structural representatives for protein families might be used as template structures to model the remaining family members. This could reveal structural information for hundreds or thousands of proteins in a family where there is none available at the moment.

2.4.2 Using Coevolution to Predict 3D Structure of Protein

Experimental structure determination could also benefit from accurately predicted contacts and structures. In NMR, the coevolutionary couplings can serve as an additional signal and improve the quality of resultant structures (Tang et al. 2015). In X-ray crystallography, models from contact-based structure prediction can be used to solve the phase problem (Safarian et al. 2016). Besides these, a study has illustrated the possibility of combining X-ray structures deposited in PDB with coevolutionary-based contact predictions to find other conformations that the proteins sample (Morcos et al. 2013).

Disordered proteins do not have fixed structures, but they can assume specific structural states when interacting with other molecules. It was shown that these states could be identified by DCA(direct coupling analysis)-based contact predictions. This achievement revealed possible folds that have not been observed before (Toth-Petroczy et al. 2016).

In addition, the application of contact prediction with global statistical models extends far beyond the tertiary structure prediction of a single protein chain. When combining the alignments of two protein families, contacts on the interface could be detected, serving as the constraints for protein-protein docking (Burger and van Nimwegen 2008; Hopf et al. 2014; Ovchinnikov et al. 2014; Pazos et al. 1997; Yeang 2007). It was also applied on the nucleotide level, where contacts in RNA molecules could be inferred and used to predict tertiary structures of RNA (De Leonardis et al. 2015); by combining the alignments with those of protein-coding genes, it revealed protein-RNA interactions (Weinreb et al. 2016).

In a recent large-scale study, DCA has been used to detect epistatic effects on a nucleotide level between genes of streptococcus bacteria, where whole genome alignments were served as inputs (Skwark *et al.* 2017). The resulting network revealed strong couplings between genes related to antibiotic resistance, facilitating the identification of novel drug targets.

2.4.3 Using Coevolution to Study Protein Function

Besides amino acid contact prediction, coevolution was also employed for protein function study. One example is the Statistical Coupling Analysis (SCA). SCA was proposed to detect networks of residue positions that are coevolving. But it usually focuses on the patterns associated with functions (de Juan *et al.* 2013). The first implementation of SCA was introduced in 1999 by Ranganathan *et al.* (Lockless and Ranganathan 1999); it analyses how the amino acid frequency change at one position causes a statistical perturbation to the amino acid frequencies of functionally related positions. It was further developed in 2009 by Halabi *et al.* (Halabi *et al.* 2009), also from the lab of Ranganathan. It was enriched with a noise reducing method, PCA (Principal Component Analysis). The function of the noise reducing lies in the existence of a hierarchical structure in the interaction network of amino acid residues. This hierarchy can be extracted by an orthogonal transformation of the covariance matrix. It was shown that the top principal components of the covariance matrix have biological meanings and they correspond to the so-called functional sectors (Halabi *et al.* 2009). Results showed that each sector has a specific functional role, and is physically connected with others in the tertiary structure.

Thus, it is suggested that a protein sector is an evolutionary unit of protein structure (Halabi et al. 2009; McLaughlin et al. 2012).

Both versions of SCA were successful in searching for clusters of coevolving residues that contribute to protein folding or allosteric interactions (de Juan et al. 2013; Halabi et al. 2009). However, a lack of comprehensive benchmarking is a major limitation for the applications of SCA-based methods (de Juan et al. 2013). It has also been shown that they are not competitive for predicting protein contacts as compared with DCA methods (Kukic et al. 2014; Walsh et al. 2009).

2.4.4 Summary of This Section

The field of statistical analysis of coevolutionary signals between residues is active and thriving. It is expected that new coevolution-based methodologies will continue to be developed and improved. One aspect is to investigate the amino acid non-contact prediction. Due to the obvious fact that non-contact residues are much more numerous than contact residues (see Figure 5.1 in the Results chapter (Chapter 5) for the percentage of amino acid contact pairs in an analysis of 250 unrelated protein chains), if inter-residue distances other than contacts can be precisely predicted for a given query protein, it could be easier to carry out accurate structure prediction. We are not the first group to notice the potential benefits of inter-residue distances. Two papers from the group of Pollastri (from the University College Dublin, Ireland) have already been published to introduce

the ideas of both predicting real-value residue-residue distance map and further using the map to predict the protein structure (Kukic et al. 2014; Walsh et al. 2009). However, their results are very poor – the predicted distance is about 6Å difference compared to native distance on average and structure predictions only have an average TM-score of about 0.23 against the native structures.

In this thesis, both inter-residue contact predictions and non-contact (or distance) predictions are produced by machine learning models. Some features in these models take advantage of the latest development of the study of amino acid coevolution. The protein structure prediction results show that the incorporation of non-contact predictions can improve the quality of predicted protein models (see Subsection 5.6.6 in Chapter 5).

2.5 Protein Sequence Alignment and Homology Detection

Protein sequence alignment and homology detection are the essential steps of all coevolution-based statistical amino acid contact prediction algorithms (introduced in Section 2.6), as well as the machine learning based amino acid contact/distance prediction models (of course, including the DeepCDpred algorithm proposed in this study). Specifically, for all the amino acid coevolution analysis, the target protein sequence is firstly searched against a sequence database, and the significantly similar sequences (hits,

or homologues) are aligned to build an MSA. The subsequent calculations are based on this MSA.

Proteins sharing a common ancestor are considered to be homologous. Homologous proteins that arose through speciation are orthologs; or from gene duplication are paralogs. Orthologous sequences thus have identical or almost identical functions in different species; whereas paralogous sequences can undergo differentiation resulting in different functions. A set of homologous sequences composes a protein family. It is favourable to define consensus amino acid residues of a family and map residues of the individual proteins onto them for statistical analysis. This leads to a symbol matrix, as already mentioned, the MSA.

In an MSA, equivalent residues are placed in the same column. Hence, the variations of the amino acids in the columns tell a story of evolutionary mutations that may spread to millions of years (Figure 2.3). A well-known online MSA database is Pfam (Finn et al. 2016), which contains a large number of protein families (more than 16,000 by the writing of this thesis), with up to several hundred thousand sequences for each family.

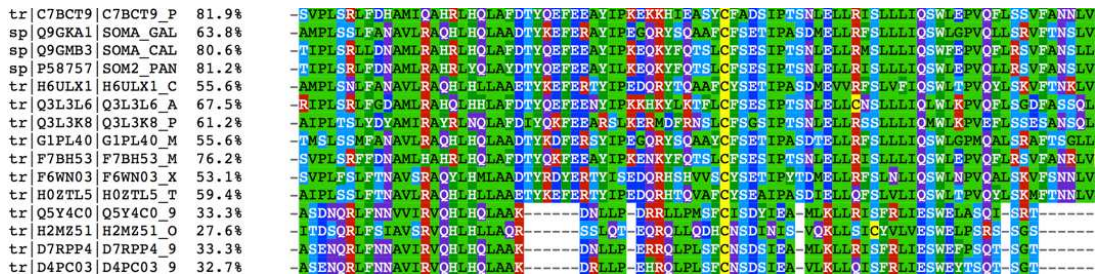


Figure 2.3. A section of the MSA of Pfam PF00103.

Before building an MSA, it is necessary to search for homologous sequences in a sequence database for a given query protein. This step is called homology detection. Homology detection has become a prerequisite procedure in the fields of computational biology such as protein evolution, and structure and function prediction of new proteins. The central idea of protein homology detection is to measure the sequence similarity (other information may also be incorporated, e.g., secondary structure prediction) between the query and sequences from a dataset or database with respect to a certain scoring scheme.

Lots of computational methods have been developed for homology detection. Probably, the simplest one in this field is pairwise sequence alignment with constant match/mismatch and gap scores. It can be achieved by dynamic programming algorithms such as Needleman-Wunsch ([Needleman and Wunsch 1970](#)) or Smith-Waterman ([Smith and Waterman 1981](#)). The idea behind dynamic programming is to find an optimal solution for aligning each position in one sequence to each position in another sequence with respect to an underlying scoring scheme. In the following, some of the sophisticated and widely-used protein homology detection tools developed in the past three decades and how to align the searched significant hits to generate an MSA are introduced.

2.5.1 From Amino Acid Substitution Models to BLAST

“Amino acid substitution models estimate the replacement rate of an amino acid residue by another” ([Pavlopoulou and Michalopoulos 2011](#)). The Point Accepted Mutation (PAM)

matrix is constructed relying on the sequence alignments of closely related sequences ([Dayhoff and Schwartz 1978](#); [Pavlopoulou and Michalopoulos 2011](#)). It was shown that PAM matrix cannot well approximate changes over long evolutionary timescales ([Risler et al. 1988](#)). Using the PAM matrix, Risler *et al.* aligned two pairs of protein sequences, the N and the C-terminal sections of chymotrypsin and elastase respectively, which have low sequence identity but similar 3D topologies ([Risler et al. 1988](#)). The reference alignments were obtained by superimposing the 3D structures of each pair. Their results showed that PAM could not align them well. To solve this problem, the BLOck SUBstitution Matrix (BLOSUM) was built based on conserved, functionally important regions (no gaps in the regions) found in the alignments of remotely related sequences ([Henikoff and Henikoff 1992](#)). For example, the BLOSUM62 is derived from observed substitutions in ungapped alignments that share at least 62% sequence identity. Other amino acid substitution matrices in the BLOSUM family can be calculated in the same way.

The widely used program of BLAST (Basic Local Alignment Search Tool) employs BLOSUM62 as the default amino acid substitution matrix for detecting homologues for a query sequence ([Pearson 2013](#)). With this matrix, BLAST uses a heuristic searching strategy to find short matches to the target from a sequence database.

2.5.2 Profile-Sequence Comparison and PSI-BLAST

The sensitivity of homology detection was further improved by incorporating the information from multiple sequences instead of just one. A sequence profile, or a position-specific scoring matrix (PSSM), is a description of the consensus of an MSA, capturing the variability of sequences in a family through position-specific amino acid frequencies, which are stored as a $N \times 20$ matrix (N is the columns of the MSA and 20 is all the types of amino acids). This makes it a much more sensitive and specific method than position-independent scoring systems for database searching.

PSI-BLAST (Position-Specific Iterative BLAST) derives a PSSM from the alignment of significant hits of one round of BLASTP (Altschul et al. 1997). The PSSM is then used to further search the sequence database for new hits, and is updated for subsequent iterations with the newly identified sequences. It is important to note that gaps are still scored equally, rather than according to their different positions in the sequence (Altschul et al. 1997; Edgar and Sjolander 2004). The reasons why PSI-BLAST does not use position-specific gap scores include two aspects: (a) there is no good theory for deriving such a gap score system (Altschul et al. 1997) and (b) “*eschewing the position-specific gap costs could help to make a reasonable estimate of the statistical significance of the resulting local alignments*” (Altschul et al. 1997). However, it was shown that not using position-specific gap scores is one reason that PSI-BLAST is less sensitive than profile HMMs (introduced in the next section) (Soding 2005).

After homologue detection, both BLAST and PSI-BLAST simply collapse the pairwise alignments between the query and each hit into a multiple sequence alignment by aligning all of the hits to the query (Figure 2.4).

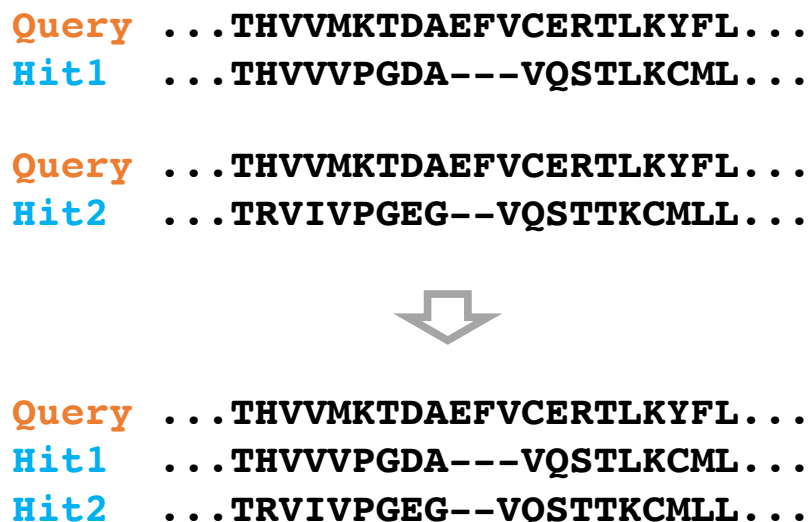


Figure 2.4. The way of building a multiple sequence alignment by BLAST or PSI-BLAST. It aligns all of the hits into an MSA according to the pairwise alignments of each hit with the query.

2.5.3 Profile HMM-Sequence Comparison

More accurate homology detection algorithms use profile HMMs (hidden Markov models) to include position-specific gap scores and neighbouring dependencies. To arrive at a description of how profile HMMs detect remote sequences, the basics of an HMM are introduced below.

An HMM (Krogh et al. 1994) is the statistical modelling of a stochastic process for which the outputs are observable, while the internal states of the model producing them remain

hidden. It also has the Markov property, which says that the states of the system in the future only depend on the current state.

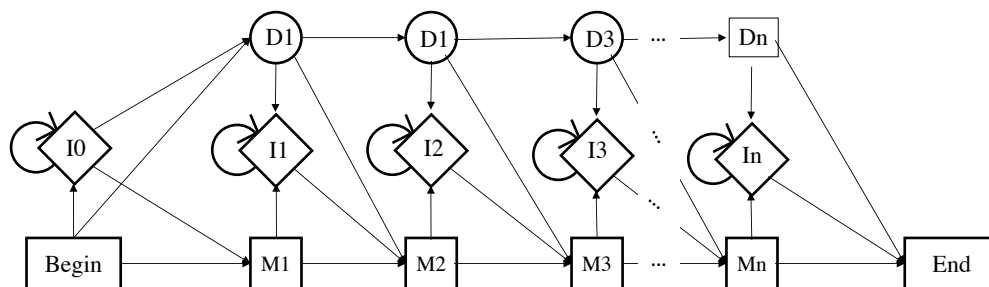


Figure 2.5. Diagram of profile HMM for protein homologous sequence detection.

When an HMM is employed for protein remote homologue detection, the internal states are matches (M), insertions (I), and deletions (D) (Figure 2.5). The match state is used to model consensus amino acids within a family; while the insertion and deletion states represent additional and skipped amino acids relative to the family, respectively.

Like sequence profiles, a profile HMM is derived from the MSA of homologous proteins. It provides a statistical framework for the amino acid frequencies in the columns of the profile and additionally contains position specific insertion and deletion probabilities (one of the differences to PSI-BLAST). Specifically, in a profile HMM, there are two types of probabilities – emission probabilities and transition probabilities. The emission probabilities represent the chances that the match, deletion or insertion state emits a certain residue type at a given position on the sequence. The transition probabilities describe the probability of each of the states (match, deletion and insertion) at one position on the

sequence changes to each of the states at the next position. For an MSA, the transition and emission probabilities can be learned from the sequences. Aligning a target sequence to the profile HMM is to use the two sets of probabilities to score each possible path (a path, as indicated by arrows in Figure 2.5, represents the states of all positions that can emit the target sequence) and then find the best path (Figure 2.5) that maximizes the score by the Viterbi algorithm (Eddy 1998), a variant of dynamics programming.

HMMER3 is a software suite for HMM-sequence comparison (Eddy 2009). The iterative search method in HMMER3 is called Jackhmmer. In the first iteration, a profile HMM is built using a simple scoring scheme (BLOSUM62) for the query sequence. With this profile HMM, a sequence database is searched and all hits that pass the inclusion threshold (i.e., E-value) are added to the query sequence in a multiple alignment and a profile is then made from the alignment. This new profile is further used as the input for the next search iteration. Iterations continue until no new sequences are found or the maximum number of iterations is reached.

Since PSI-BLAST and HMMER3 are based on profile-to-sequence and HMM-to-sequence comparisons, respectively, they have the advantage over HHsearch and HHblits (which will be mentioned below) of being able to search raw sequence databases, although the latter are more sensitive for detecting and aligning remote homologous proteins.

2.5.4 Profile-Profile Hidden Markov Models

Recent efforts have been made to improve the sensitivity of profile HMMs by aligning a profile HMM built from the query sequence with another profile HMM built from a training profile. HHsearch (Soding 2005) is among these approaches. HHsearch performs local alignment between pairs of profile HMMs. For this purpose, both the query protein and the database must be represented in the HMM format. Besides the HMM pair comparison, HHsearch is further improved with the incorporation of a secondary structure term into its alignment scoring function. Each HMM in HHsearch is therefore built from two components:

1. A sequence profile, computed by PSI-BLAST with multiple iterations;
2. Secondary structure, predicted by PSIPRED (PSI-blast based secondary structure PREDiction) for the sequence alignment (McGuffin et al. 2000) or computed by DSSP (Dictionary of Protein Secondary Structure) if there is at least one structure of the proteins in the alignment available (Kabsch and Sander 1983).

Due to the better performance in detecting remote homologues, HHsearch is often used for homology modelling (Bordoli et al. 2009; Kelley et al. 2015; Meier and Soding 2015). Besides the standalone toolkit that can be downloaded from <https://github.com/soedinglab/hh-suite> (last check: November 2018), there is also a web

server (HHpred) available at <https://toolkit.tuebingen.mpg.de/#/tools/hhpred> (last check: November 2018) that runs HHsearch.

One major drawback of HHsearch is that it is generally “*too slow for iteratively searching through large sequence databases such as UniProt*” (Remmert et al. 2012). Another algorithm named HHblits (HMM-HMM Based Lightning-fast ITERative Sequence search), from the same laboratory, solved this problem. It was shown that HHblits is faster, and constructs multiple alignments with significantly better quality than PSI-BLAST and HMMER3, and not compromising on sensitivity compared to HHsearch (Remmert et al. 2012). Unlike HHsearch, HHblits can also build MSAs beginning with a single protein sequence.

To search homologues, HHblits first changes the query sequence (or query MSA) to a profile HMM by putting pseudocounts of amino acids that are physicochemically similar to amino acids in the query (Remmert et al. 2012). HHblits then searches a profile HMM database by using HHsearch and appends sequences from the significant hits based on which the new HMM for the next iteration is constructed. To accelerate speed, the probabilities of the 20 amino acids in each HMM column are discretized into an alphabet of 219 discretized states. The 219 states are encoded by 219 ASCII characters, with each approximating a typical amino acid probability vector in the columns of the profile HMMs. The authors clustered amino acid distributions of the columns of a large training profile HMM set (built from the NCBI *nr* database) into 219 clusters and each cluster is represented by a character (Remmert et al. 2012). Then, the score of each HMM column in the

query MSA is computed with each of the 219 discretized states, which produces a 219-row extended sequence profile. The profile is then aligned to a profile HMM database. Statistically significant profile HMMs that have passed a prefilter are aligned again ([Remmert et al. 2012](#)). The use of HHblits requires a special format sequence database, in which a large number (usually millions) of profile HMMs are indexed([Remmert et al. 2012](#)). Each profile HMM is built with several sequences. Fortunately, users can download such a database together with the HHblits program suite ([HHblits 2018](#)). The name of the latest database is “uniclust30_2017_10”.

2.6 Review of Amino Acid Coevolution Analysis

There have been two types of coevolution analysis used for amino acid contact prediction, i.e., local statistical models and global statistical models. These two types of models share in common that they are pure statistical approaches and have a concrete theory to identify the coevolutionary signals from an MSA of the query protein. None of them uses other features of the query protein (e.g., the amino acid profile or secondary structure prediction) to help the identification of the coevolutionary couplings. However, the recently developed machine learning models incorporate coevolutionary couplings predicted by local statistical models or/and global statistical models, as well as 1D and 2D properties of the query protein. These machine learning models are not designed for coevolution analysis, but for amino acid contact/distance prediction. However, since they use coevolutionary couplings, they are also called coevolution based amino acid contact prediction models.

Representatives of the three types of coevolution based amino acid contact prediction models are reviewed in this section. Among them, both predictions from the local and global statistical models were used as features in the contact/distance prediction algorithm developed in this study (DeepCDpred). The predictions from the machine learning algorithm, MetaPSICOV, were used to benchmark the performance of DeepCDpred.

All of the local and global statistical models below use the MSA of the query sequence as

the input and output a matrix of pairwise amino acid coevolutionary couplings. The value of the couplings is used to indicate how likely a pair of residues is in contact. An MSA of detected significant homologues of the query protein sequence is the start point for all of the models reviewed in this section. In order to carry out mathematical calculations, the MSA is firstly converted to a numerical matrix whose values are from 0 to 20 (where 0 to 19 represent 20 letters in the MSA and 20 stands for the gap). This conversion scheme is adopted from the one previously used in PSICOV (Jones et al. 2012) and the detail of the mappings can be found in Table A.1 of Appendix A. For ease of description, the following notations are made:

1. M and L are the number of rows and the number of columns in the MSA, which represent the number of sequences and the number of amino acids in the query protein, respectively;
2. Uppercase $\mathbf{X} = (X_1, \dots, X_L) \in \mathbb{R}^L$ is an L dimensional random variable vector with X_i corresponding to i^{th} column of the MSA, where $1 \leq i \leq L$;
3. Lowercase $\mathbf{x} = (x_1, \dots, x_L) \in \mathbb{R}^L$ represents an observation of \mathbf{X} and a row (that is, a sequence) in the MSA; $\mathbf{x}^m = (x_1^m, \dots, x_L^m)$ is the m^{th} observation of \mathbf{X} and m^{th} row in the MSA, where $1 \leq m \leq M$;
4. $f_i(a)$ is the observed frequency of amino acid type a in i^{th} column; $f_{ij}(a, b)$ is the observed joint frequency of amino acid type a occurring in column i and b occurring in column j ;

2.6.1 Local Statistics Models

The first attempts to study amino acid coevolution used local statistical models. They assume that the dependency of any pair of positions is independent of all the other pairs. These methods have been widely used until about 9 years ago (1999). Pearson correlation coefficient, observed minus expected squared (OMES) and mutual information are three representatives among them and mutual information is probably the most widely used (de Juan et al. 2013). The following provides a brief introduction to the application of mutual information in the amino acid coevolution study.

Mutual information detects covarying positions in an MSA based on the strategy that it measures whether the presence of the amino acids at one position is a good prediction of the presence of amino acids at another position. Specifically, for an MSA, one can calculate the empirical mutual information MI_{ij} between any column pairs i and j using one-site and two-site amino acid frequencies $f_i(a)$ (as well as $f_j(b)$) and $f_{ij}(a, b)$,

$$MI_{ij} = MI_{ji} = \sum_{ab} f_{ij}(a, b) \log \left(\frac{f_{ij}(a, b)}{f_i(a) f_j(b)} \right) \quad (2.1)$$

MI_{ij} is bounded between $[0, \min(H_i, H_j)]$ (Martin et al. 2005) and equals to zero if and only if the distributions of the two columns are independent. Here, H_i and H_j are the empirical entropies of column i and column j , respectively. A high score of MI_{ij} is usually used as an indicator of coupling (Gloor et al. 2005). It is easier to understand and make comparisons to bound $MI_{ij} \in [0, 1]$. Thus, normalized mutual information

$NMI_{ij} = \frac{MI_{ij}}{\min(H_i, H_j)}$ was proposed and used (Martin et al. 2005). $NMI_{ij} = 1$ if column i and column j are perfectly correlated.

However, MI suffers from a problem related to the high degree of amino acid conservation at some sites. For example, if position i is always leucine and position j always lysine, NMI_{ij} would be equal to one although no variation between the two positions is observed. By definition, MI also does not take into account which residues are present in both columns in an MSA. It just treats different amino acid types as different symbols (numbers). Thus, the biochemical changes are ignored when MI is used to assess the similarity of mutational patterns between two positions. Besides the two limitations, MI also cannot remove phylogenetic biases and indirect interaction effects, which will be discussed in the next two sections.

2.6.2 Disentangling Directly Coupled Positions from the Network of Indirectly Correlated Positions

Indirect interaction is also called chaining correlation or transitive correlation. Due to the essence of correlation, the mutation between two residues may present significant correlation even if they are not close in structure. In the simplest situation with three residues, A , B , and C , if both residue pairs AB and BC are coevolving together directly, there may be an apparent correlation between A and C (“indirect interaction”) regardless of whether these residues interact (Figure 2.6). This additional correlation appears because

the two real couplings share residue B . It is obvious that the above local statistical models are not able to remove these artefacts – they just reflect how strong the correlation between two sites is, but do not distinguish where the correlation comes from. Thus, the coupling matrix obtained by local statistical models mixes both direct and indirect couplings (Figure 2.6, right).

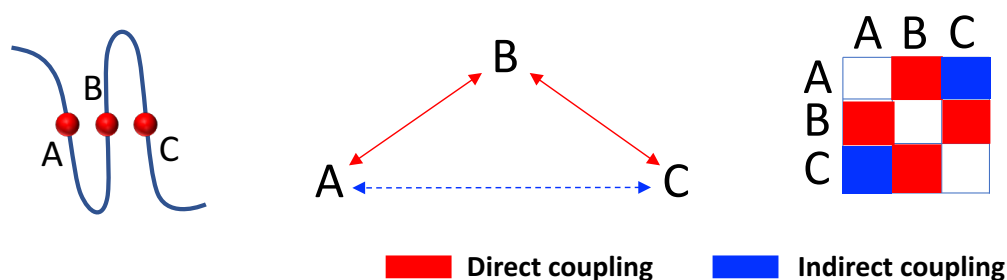


Figure 2.6. An illustration of the indirect coupling caused by the transfer of direct couplings.

The mixture of these indirect correlations makes it difficult to identify the directly coupled sites. However, since the direct couplings are more useful for predicting spatially close amino acid residues in protein structures, approaches need to distinguish direct from indirect couplings (de Juan et al. 2013). The methods discussed in the Global Statistical Models subsection (2.6.4) have shown good performance in dealing with this problem.

2.6.3 Phylogenetic Bias Correction

Because of the evolutionary relationships among species, in an MSA, the aligned sequences do not represent independent samples – sequences from some taxa may be over-represented and other sequences under-represented, as driven by the research interests of the scientific

community. This violates the assumption that the measured sequences are independent and identically distributed. It might happen that two positions just keep their respective ancestral amino acid with no mutual influence. The two positions then show a high degree of correlation resulted from phylogeny alone. This effect is called phylogenetic noise or bias (Baker and Porollo 2016). Since the calculations of both local statistical models and global statistical models (see detail in the next section) are based on MSAs, they all suffer from this problem.

A practical solution for this bias can be dealt with by an approach called Average Product Correction (APC) (Dunn et al. 2008). An instructive derivation of APC found in (Burger and van Nimwegen 2010) is followed here. Assume that the coupling J_{ij} of positions i and j is made of two parts, J_{ij}^r , corresponding to a real (or observed) mutual influence (can be indirect or direct), and $B_i B_j$, a product of single-site characteristics that represents the background bias:

$$J_{ij} = J_{ij}^r + B_i B_j \quad (2.2)$$

One hopes to correct such a background by estimating the contribution of it and subtracting it from the coupling score. It is now assumed that $J_{ij}^r \ll B_i B_j$. Then the one and two-site averages of J_{ij} will also be dominated by the single site contributions (average denoted by \bullet):

$$\begin{cases} J_{i\bullet} \approx B_i B_\bullet \\ J_{\bullet\bullet} \approx (B_\bullet)^2 \end{cases} \quad (2.3)$$

where $J_{i\bullet} = \frac{1}{N} \sum_{j=1}^N J_{ij}$, $B_\bullet = \frac{1}{N} \sum_{i=1}^N B_i$ and $J_{\bullet\bullet} = \frac{1}{N} \sum_{j=1}^N J_{i\bullet}$.

Therefore,

$$J_{ij}^r = J_{ij} - \frac{J_{i\bullet} J_{\bullet j}}{J_{\bullet\bullet}} \quad (2.4)$$

Previous studies have shown that couplings corrected by APC could improve amino acid contact prediction (Buslje et al. 2009; Dunn et al. 2008; Gomes et al. 2012).

2.6.4 Global Statistical Models

Breakthroughs to address the transitive interaction problem of the local statistical models (e.g., mutual information) and predict much more accurate amino acid contacts were only made with the adoption of global statistical models. These techniques view all residues in a whole network simultaneously and treat pairs of correlated residues as dependent on one another. They try to find the smallest set of direct couplings that can explain the observed correlations in the MSA by applying the maximum entropy principle.

From this subsection (2.6.4) to the subsection of “Pseudo-likelihood Maximization DCA” (2.6.9), the background of global statistical models (2.6.4), the optimization strategy (maximum entropy principle, 2.6.5), and three representatives (2.6.7, 2.6.8 and 2.6.9) of the global statistical models are introduced. These three models were used in the pipeline of the proposed algorithm in this thesis, DeepCDpred (precisely, PSICOV was replaced with QUIC for speed consideration. See the Model Development chapter (Chapter 4) and the Results chapter (Chapter 5) for details). The subsection of “Sequence Reweighting” (2.6.6) introduces the preprocessing that is necessary before feeding the MSA of the target

sequence to the three statistical models to generate the coevolutionary couplings for each pair of residues in the target sequence.

Applying the maximum entropy principle to the study of amino acid coevolution was initially attempted by the work of Lapedes *et al.* (Lapedes *et al.* 1999). However, due to a large number of parameters in the proposed model and lack of optimization algorithms, this work was unrecognized at that time. Ten years later, Weigt *et al.* (Weigt *et al.* 2009) also applied the maximum entropy principle to infer the amino acid coevolution, but used a more computationally efficient message-passing algorithm. Once again, due to the scaling of the computational complexity, this work still only analysed 60 positions of an MSA. This method was termed message-passing Direct Coupling Analysis (mpDCA) because of its ability to distinguish direct from indirect coevolutional couplings.

The next several years documented a significant step forward for the development of global statistical models, or more precisely, DCA related models. Just like mpDCA, most of these methods concretize the maximum entropy principle with the observed one-site and two-site amino acid frequencies from an MSA. The representatives among them are mean-field DCA (mfDCA), PSICOV (Protein Sparse Inverse COVariance) and pseudo-likelihood maximization DCA (plmDCA). mfDCA was implemented in a software named FreeContact (Kajan *et al.* 2014), which is capable of running in parallel and similarly, plmDCA was implemented in CCMpred (Seemayer *et al.* 2014) to run in parallel. The original release of PSICOV was already programmed to use multiple CPU cores. The difference between these methods are: mfDCA essentially takes the inverse of the correlation

matrix and is thus the fastest and is capable of processing large proteins in a reasonable period of time (Morcos et al. 2011); PSICOV assumes the global model with a multivariate Gaussian distribution and approximates the likelihood function of the distribution by using the GLASSO algorithm (Friedman et al. 2008). It was shown that PSICOV can make better inter-residue contact prediction than mfDCA (Jones et al. 2012); however, it also suffers from much slower speed than both the mfDCA implementation of FreeContact and the plmDCA implementation of CCMpred (Kajan et al. 2014; Seemayer et al. 2014); plmDCA simplifies the likelihood function of the global model with an easy-to-calculate pseudo-likelihood and achieves better residue contact prediction than both of mfDCA and PSICOV (Ekeberg et al. 2014, 2013; Jones et al. 2015).

In the following subsection, the maximum entropy principle is introduced and how to apply it to the MSA data to produce a global statistical model is explained.

2.6.5 Maximum Entropy Principle

The principle of maximum entropy seeks a solution to the problem of how one selects the best probabilistic model, which is consistent with certain constraints, from many models. The answer is one should choose the model that assigns probabilities in a probability space as evenly as possible. Mathematically, this means to find the probability P that maximizes the formula $S(P)$ with constraints $C(P) = 0$, where $S(P)$ is the Shannon entropy. In other words, maximum entropy is also called the least constrained approach.

A multivariate probabilistic model $P(X_1, \dots, X_L)$ assigns a probability to any amino acid sequence $A = (a_1, \dots, a_L)$ based on the empirical frequency counts in the MSA. More precisely, in order to be consistent with the MSA, the probabilistic model is chosen to reproduce the empirical one-site and two-site amino acid frequency counts:

$$S = - \sum_X P(X) \log P(X) \quad (2.5)$$

subject to

$$\begin{aligned} \sum_{\{a_k | k \neq i\}} P(a_1, \dots, a_L) &= f_i(a_i) \\ \sum_{\{a_{k,l} | k \neq i, l \neq j\}} P(a_1, \dots, a_L) &= f_{ij}(a_i, a_j) \end{aligned} \quad (2.6)$$

where $X = (X_1, \dots, X_L)$ is the variable vector; $f_i(a_i)$ is the frequency of proteins having amino acid a in column i of the MSA, and $f_{ij}(a_i, a_j)$ counts the fraction of proteins with amino acid a in column i and amino acid b in column j .

Maximizing S with the constraints can be carried out through the introduction of Lagrange multipliers. Now, there are two choices to progress calculations. One is to view the distribution $P(X)$ as discrete so that each random variable can only choose values from a finite set (e.g., 20 amino acids and a gap, that is $\{1, \dots, 21\}$); the second is to consider the distribution as continuous. For the former, after some calculations, the Potts distribution can be obtained (Marks et al. 2011):

$$P(a_1, \dots, a_L) = \frac{1}{\mathbf{Z}} \exp \left(\sum_{i=1}^L \sum_{j=i+1}^L J_{ij}(a_i, a_j) + \sum_{i=1}^L h_i(a_i) \right) \quad (2.7)$$

where the a_i, a_j can take any value from an alphabet of size $q = 21$ (20 amino acids and one gap symbol) and the $J_{ij}(a_i, a_j)$ and $h_i(a_i)$ are real numbers indexed by the positions i and j and the amino acids a_i and a_j .

The normalization constant \mathbf{Z} is defined as

$$\mathbf{Z} = \sum_{(a_1 \dots, a_L)} \exp \left(\sum_{i=1}^L \sum_{j=i+1}^L J_{ij}(a_i, a_j) + \sum_{i=1}^L h_i(a_i) \right) \quad (2.8)$$

\mathbf{Z} is also known as the partition function and ensures the sum of the probabilities equals 1; $\mathbf{x} = (a_1 \dots, a_L)$ runs through all possible sequences. Notice that the outer sum in Equation 2.8 contains q^N ($q=21$) terms, corresponding to all possible sequences. This means that an exact and direct calculation of \mathbf{Z} is impossible even for small proteins. Methods therefore must generally avoid an exhaustive evaluation of \mathbf{Z} and one thus needs to rely on approximate methods for inferring the model parameters.

In physics, the Potts model is a generalization of the Ising model. The latter describes a lattice of classical spins that can each be in one of two states (Wu 1982). The generalization is that in the Potts model the spins can be in one of $q > 2$ states. Inspired by statistical physics, the exponent in Equation 2.7 is often called (with a minus sign) the Hamiltonian:

$$-\mathbf{H}(\mathbf{x} = (a_1 \dots, a_L)) = \left(\sum_{i=1}^L \sum_{j=i+1}^L J_{ij}(a_i, a_j) + \sum_{i=1}^L h_i(a_i) \right) \quad (2.9)$$

In the Potts model, J_{ij} is the coupling strength between spins i and j , and h_i is the strength of the external field at position i on the lattice (Ekeberg et al. 2013). When this model is applied to the amino acid coevolution analysis, the local couplings and fields

describe the preferences of positions to carry certain amino acids. Specifically, a large $h_i(a)$ is a bias of position i toward preferring amino acid type a , and a large $J_{ij}(a, b)$ translates into a desire for positions i and j to jointly carry amino acid types a and b . J_{ij} indicates the direct interaction between positions i and j . Hence, the aim here is to infer the parameter set $\{h, J\}$ (h is useful to characterize the conservation states of all the positions in a query protein) from the observed one-site and two-site amino acid frequencies, i.e., inferring the model parameters from observations of the system, which is also called the inverse Potts problem.

If the distribution $P(\mathbf{X})$ is considered as continuous rather than the discrete distribution that leads to the the Potts model, it takes the form of a multi-variate Gaussian distribution (Friedman et al. 2008; Stein et al. 2015):

$$P(\mathbf{X}) = \frac{1}{\sqrt{(2\pi)^L \det \Sigma}} \exp\left(-\frac{1}{2}(\mathbf{X} - \mu)^T \Sigma^{-1}(\mathbf{X} - \mu)\right) \quad (2.10)$$

where μ and Σ is the mean vector and symmetric covariance matrix of the population, respectively; $\det \Sigma$ is the determinate of Σ . Note that Σ is positive definite and thus $\det \Sigma > 0$.

The proof of why $P(\mathbf{X})$ takes the form of Equation 2.10 can be found in the paper (Stein et al. 2015).

Equation 2.10 is well determined. The inverse covariance matrix, Σ^{-1} , is also termed as the precision matrix. Unlike the covariance matrix, in which each element is mixed with direct and indirect interactions between two random variables, it measures the direct

couplings within random variable pairs, after effects from other variables removed. This property is desired in amino acid contact prediction. More details can be found in the section about PSICOV (subsection 2.6.8).

The Number of Free Parameters In Equation 2.7

In the Potts model, it is easy to obtain that the parameters h_i and \mathbf{J}_{ij} have the following vector forms:

$$h_i = \begin{pmatrix} h_{i,1} \\ h_{i,2} \\ \vdots \\ h_{i,q} \end{pmatrix} \quad \text{and} \quad \mathbf{J}_{ij} = \begin{pmatrix} J_{ij,(1,1)} & J_{ij,(1,2)} & \cdots & J_{ij,(1,q)} \\ J_{ij,(2,1)} & J_{ij,(2,2)} & \cdots & J_{ij,(2,q)} \\ \vdots & \vdots & \ddots & \vdots \\ J_{ij,(q,1)} & J_{ij,(q,2)} & \cdots & J_{ij,(q,q)} \end{pmatrix}$$

The calculation of how many free parameters there are in $\{h, J\}$ – which can suggest the complexity of the inverse problem to some extent – is as follows:

The pairs (i, j) and (j, i) are considered to be the same, and no pairs of the type (i, i) are included. Thus, the number of pairs equals the number of ways in which one can choose two elements from a collection of L without replacement, $\frac{L(L-1)}{2}$. Because there are L positions, each with a field vector (h_i) of length q , and $\frac{L(L-1)}{2}$ residues pairs, each with a coupling matrix of size q^2 , the total number of parameters is $Lq + \frac{L(L-1)q^2}{2}$. But, it turns out that the parameter set $\{h, J\}$ is over-parameterized. Because in Equation 2.6, $f_i(q)$

is implied given $f_i(1), \dots, f_i(q-1)$, since $\sum_{a=1}^q f_i(a) = 1$; and similarly, the coupling constraints $\sum_{b=1}^q f_{ij}(a, b) = f_i(a)$ also exist. These constraints lead to the real number of free parameters in $\{h, J\}$ being $L(q-1) + \frac{L(L-1)(q-1)^2}{2}$ (Cocco et al. 2013; Ekeberg et al. 2013; Morcos et al. 2011; Weigt et al. 2009). A way to solve this dimensional excess is to fix some values of the parameters, for example by setting

$$J_{ij}(q, l) = J_{ij}(l, q) = h_i(q) = 0 \quad (2.11)$$

for all i, j , and l , where $1 \leq i, j \leq L, i \neq j, 1 \leq l \leq q$. It makes all biases and couplings with the last state as a reference. This would make the solution of the inverse Potts problem unique.

In summary, global statistical models, which are based on the maximum entropy principle, represent an MSA as a 21-state Potts model. Each sequence of the MSA can be thought as a sample taken from a Potts model probability distribution (Ekeberg et al. 2014). Thus, if a Potts model can be approximated with respect to the observed sequences, this model could be used to identify coevolving positions.

2.6.6 Sequence Reweighting

This procedure aims to mitigate the phylogenetic bias. As mentioned in the subsection of Phylogenetic Bias Correction (Subsection 2.6.3), protein sequences from a database (e.g., uniprot) are not independently distributed. A good example is the homologous sequences of a target protein; some of them could have a very high identity. However, as described in

the above subsection (2.6.5), the samples (or observations) of the global statistical models should be independently distributed. A commonly used approach to alleviate the problem resulted from similar sequences is called the sequence reweighting (Ekeberg et al. 2014; Hopf et al. 2014; Jones et al. 2012). The essence of this approach is to assign a weight to each sequence in an MSA based on how many similar sequences are related to it. The more similar sequences a sequence has, the lower the weight it obtains. Two sequences are considered to be similar if more than a fraction of σ ($0 \leq \sigma \leq 1$) of all the positions in the alignment of these two sequences have the same amino acids. Explicitly, each sequence \mathbf{x}^b is assigned a weight $w_b = 1/m_b$, where m_b is the number of sequences in the MSA that are similar to \mathbf{x}^b :

$$m_b = |\{a \in \{1, \dots, M\} : \text{similarity}(\mathbf{x}^a, \mathbf{x}^b) \geq \sigma\}|$$

Here, appropriate values for σ are in the range of 0.7 – 0.9 (Ekeberg et al. 2013; Morcos et al. 2011; Weigt et al. 2009).

Using this technique, the one-site and two-site amino acid frequencies are modified to

$$\begin{aligned} f_i(k) &= \frac{1}{M_{\text{eff}}} \sum_{b=1}^M w_b I(x_i^b = k) \\ f_{ij}(k, l) &= \frac{1}{M_{\text{eff}}} \sum_{b=1}^M w_b I(x_i^b = k) I(x_j^b = l) \end{aligned} \tag{2.12}$$

respectively. Here $M_{\text{eff}} = \sum_{b=1}^M w_b$ is the effective number of sequences and $I(x_i^b = k)$ is the indicator function with the value of 1 if $x_i^b = k$, otherwise 0 (the same with $I(x_j^b = l)$).

2.6.7 Mean Field DCA

Mean-Field Direct Coupling Analysis (mfDCA), proposed by Morcos *et al.* (Morcos *et al.* 2011), whose definition is followed below, was the first fast and efficient method to infer the couplings J_{ij} in Equation 2.7, for a given MSA. The idea behind this approach is a Taylor-expansion of the Legendre transform of the term $F = -\ln Z$ (F is the free energy of the system) around zero (Morcos *et al.* 2011).

Specifically, at the beginning, a perturbation parameter ε is introduced to control the strength of the interaction term in the Hamiltonian:

$$\mathbf{Z}(\varepsilon) = \sum_{\mathbf{x}} \exp \left(\varepsilon \sum_{i=1}^L \sum_{j=i+1}^L J_{ij}(x_i, x_j) + \sum_{i=1}^L h_i(x_i) \right) \quad (2.13)$$

Then the Legendre transform of $-\ln Z$ is considered:

$$G(\varepsilon) = \ln Z(\varepsilon) - \sum_{i=1}^L \sum_{a_i=1}^{q-1} h_i(a_i) P_i(a_i) \quad (2.14)$$

where P is the multi-variate probability distribution defined in Equation 2.7.

Approximates $G(\varepsilon)$ to the first order in ε by using a Taylor series expansion:

$$G(\varepsilon) = G(0) + \varepsilon \left. \frac{\partial G(\varepsilon)}{\partial \varepsilon} \right|_{\varepsilon=0} + \mathcal{O}(\varepsilon^2) \quad (2.15)$$

In this approximation, the key result is obtained:

$$(C^{-1})_{ij}(a_i, a_j) = -J_{ij}(a_i, a_j) \quad (2.16)$$

where empirical covariance matrix is defined as

$$C_{ij}(a_i, a_j) = f_{ij}(a_i, a_j) - f_i(a_i) f_j(a_j) \quad (2.17)$$

A Potts model describing an MSA with sequence length of 50-300 amino acids includes about $10^5 - 10^7$ parameters. However, a typical protein family has only from hundreds to thousands of sequences. The limited number of samples leads to the empirical covariance matrix \mathbf{C} in the above inversion problem usually being not of full rank, and therefore, Equation 2.16 is not well-defined. Morcos *et. al.* introduced a pseudo-count approach in which the one-site and two-site amino acid frequencies are adjusted by a parameter λ (Morcos *et al.* 2011). This method is a modified version of the reweighting scheme as described in the previous section:

$$\begin{aligned} f_i(k) &= \frac{1}{\lambda + M_{\text{eff}}} \sum_{m=1}^M w_m \left(\frac{\lambda}{q} + I(x_i^m = k) \right) \\ f_{ij}(k, l) &= \frac{1}{\lambda + M_{\text{eff}}} \sum_{m=1}^M w_m \left(\frac{\lambda}{q^2} + I(x_i^m = k) I(x_j^m = l) \right) \end{aligned} \quad (2.18)$$

The pseudo-count is used to ameliorate the statistical noise due to under-sampled sequences in the MSA and could also increase the rank of the correlation matrix, therefore promotes the invertibility of the covariance matrix \mathbf{C} (Ekeberg *et al.* 2013; Morcos *et al.* 2011). Here, $I(x_i^m = k)$ is an indicator function taking value 1 if and only if the amino acid at position i in sequence m is k , and otherwise taking value 0.

In summary, inferring coupling using mfDCA involves the following three steps:

1. compute the observed one-site and two-site residue frequencies, f_i and f_{ij} , from the MSA using Equation 2.18;
2. calculate the empirical correlation matrix $C_{ij}(a_i, a_j) = f_{ij}(a_i, a_j) - f_i(a_i) f_j(a_j)$;

3. invert the matrix C_{ij} to obtain the couplings $J_{ij}(a_i, a_j)$.

The task left is how to combine the couplings between positions i and j to a contact measuring score. Morcos *et al.* (Morcos *et al.* 2011) used a quantity termed direct information DI_{ij} , similar to the mutual information between site i and j , based on a two-site probability model:

$$P_{ij}^d(a, b) = \frac{1}{Z_{ij}} \exp\left(J_{ij}(a, b) + \hat{h}_i(a) + \hat{h}_j(b)\right) \quad (2.19)$$

where Z_{ij} is calculated in a similar way to Equation 2.8, but limited to sites i and j ; \hat{h} is inferred so that the empirical one-site frequencies are recovered with this two-site probability model (Morcos *et al.* 2011). The direct information (DI) between sites i and j is defined as:

$$DI_{ij} = \sum_{a,b} P_{ij}^d(a, b) \ln \frac{P_{ij}^d(a, b)}{f_i(a)f_j(b)} \quad (2.20)$$

The DI_{ij} , sorted by their numerical values, stand for the evolutionary coupling strength for the initial MSA.

2.6.8 Sparse Inverse Covariance Estimation (PSICOV)

In the *Maximum Entropy Principle* subsection (2.6.5) of this chapter, it was discussed that if each sequence of an MSA is considered as being sampled from a continuous multivariate distribution, and with the first moment and second moment of the observations

(e.g., the sequences in the MSA) as constraints, unlike the Potts model, the probability adopts the form of a multi-variate Gaussian distribution:

$$P(\mathbf{X}) = \frac{1}{\sqrt{(2\pi)^N \det \Sigma}} \exp\left(-\frac{1}{2}(\mathbf{X} - \mu)^T \Sigma^{-1}(\mathbf{X} - \mu)\right) \quad (2.21)$$

where μ and Σ is the mean vector and symmetric covariance matrix of the population, respectively; $\det \Sigma$ is the determinate of Σ . Note that Σ is positive definite and thus $\det \Sigma > 0$.

Using this special distribution as a start-point, Jones *et al.* proposed a successful amino acid contact prediction algorithm termed PSICOV (Protein Sparse Inverse Covariance Estimation) in 2012 (Jones et al. 2012).

The inverse of the covariance matrix $\Theta = \Sigma^{-1}$ is termed as the precision matrix. From it, one can define the so-called partial correlation coefficient connected to the two variables X_i and X_j as:

$$\rho_{X_i, X_j} = -\frac{\Theta_{X_i, X_j}}{\sqrt{\Theta_{X_i, X_i}} \sqrt{\Theta_{X_j, X_j}}} \quad (2.22)$$

The partial correlation coefficient matrix has the property that each element in it measures the correlation of two random variables after the influence of the other variables has been taken away (Peng et al. 2009).

In order to be consistent with the notation in the previous subsection (Subsection 2.6.7), let C be the empirical covariance matrix again. Due to the large size of C and the limited number of sequences in the MSA for typical proteins, C is not invertible (Jones et al.

2012). One practical solution to calculate the inverse covariance matrix is to minimize the negative log-likelihood with a l_1 norm penalty regularizer (to overcome the overfitting problem) (Jones et al. 2012):

$$\text{obj} = \sum_{ij=1}^d C_{ij} \Theta_{ij} - \log \det \Theta + \rho \|\Theta\|_1 \quad (2.23)$$

where $\|\Theta\|_1$ is the l_1 norm – the sum of the absolute values of the elements in Θ ; ρ controls the sparsity of Θ —a larger ρ makes more of the elements in Θ approach 0; $d = 21 \times L$ and L is the number of amino acids in the query protein sequence. The empirical covariance matrix C with size of $(21 \times L) \times (21 \times L)$ is defined:

$$C_{i,j}^{a,b} = f(x_i^a, x_j^b) - f(x_i^a) f(x_j^b) \quad (2.24)$$

where $1 \leq a, b \leq 21$ and $1 \leq i, j \leq L$.

A brief explanation to the regularization is appropriate here. It is not only used in Equation 2.23, but also used in the function optimization of Subsection 2.6.9 (next subsection) and the training process of the DeepCDpred (4.2.3), the amino acid contact/distance prediction algorithm, proposed in this thesis. The idea behind regularization is as described in Occam’s razor – among many models, the one that makes the fewest assumptions should be chosen. As for solving a model with multiple parameters by optimization, regularization could be achieved by adding a penalty to the parameters. In the optimization process, the penalty term forces some of the parameters to approach zero. Two widely used regularization forms are L_1 and L_2 . L_1 minimizes the sum of the absolute values

of the parameters (e.g., Equation 2.23); while L_2 minimizes the sum of the square of the parameters (e.g., Equation 2.33).

The optimization of the objective function has been a research focus for many years. One famous algorithm is termed *graphical LASSO* (*least absolute shrinkage and selection operator*), which uses a coordinate descent procedure. In the original paper of PSICOV, Jones *et al.* (Jones *et al.* 2012) also adopted this algorithm. Using the l_1 norm to constrain the solution of the inverse of the covariance matrix to be sparse not only accelerates the calculation by allowing null rows to be skipped in the *graphical LASSO* procedure, but also constrains the statistical model to be as simple as possible (Tetchner *et al.* 2014). Sparse priors are also used in plmDCA; see the next subsection for detail. After solving the partial correlation coefficient matrix, PSICOV also attempts to reduce the effects of phylogenetic bias by applying average product correction (APC) as already mentioned in Subsection 2.6.3.

Unlike the *graphical LASSO* algorithm, which essentially solves the optimization problem of Equation 2.23 using a block-wise coordinate descent method, another algorithm proposed in 2014 named QUIC (Quadratic approximation for sparse Inverse Covariance estimation) was proved to achieve a much faster convergence, as well as a considerable improvement of performance due to a smart partitioning of variables into fixed and free sets (Hsieh *et al.* 2014). Because of these advantages, PSICOV was replaced by QUIC for amino acid coevolutionary couplings calculation in this thesis. See the Model Development chapter (Chapter 4) for more information.

2.6.9 Pseudo-likelihood Maximization DCA

Pseudo-likelihood maximization DCA, or plmDCA, represents the most successful pure mathematical approach in amino acid contact prediction so far (Jones et al. 2015). The idea of this approach is to use the pseudo-likelihood as an alternative to the full likelihood (Besag 1977). To do this, note that:

$$P(\mathbf{X}; \theta) = \prod_i P(X_i | X_1, \dots, X_{i-1}; \theta) \quad (2.25)$$

via the chain rule. Here, $\theta = \{h, J\}$ is the parameter set in the distribution. Consider the following approximation:

$$\begin{aligned} P(\mathbf{X}; \theta) &\approx \prod_i P(X_i | X_1, \dots, X_{i-1}, \dots, X_L; \theta) \\ &= \prod_i P(X_i | X_{-i}; \theta) \end{aligned} \quad (2.26)$$

where the conditioning over additional variables is added. Here,

$X_{-i} = \{X_1, \dots, X_{i-1}, \dots, X_N\}$; it is the collection of all the variables except X_i . Then, the following equation (Equation 2.27) can be obtained:

$$\begin{aligned} P(X_i = x_i | X_{-i} = \mathbf{x}_{-i}) &= \frac{\exp\left(h_i(x_i) + \sum_{j=1, j \neq i}^L J_{ij}(x_i, x_j)\right)}{\sum_{l=1}^q \exp\left(h_i(l) + \sum_{j=1, j \neq i}^L J_{ij}(l, x_j)\right)} \\ &= \frac{1}{\mathbf{Z}_i} \exp\left(h_i(x_i) + \sum_{j=1, j \neq i}^L J_{ij}(x_i, x_j)\right) \end{aligned} \quad (2.27)$$

where $\mathbf{Z}_i = \sum_{l=1}^q \exp \left(h_i(l) + \sum_{j=1; j \neq i}^L J_{ij}(l, x_j) \right)$ and for convenience, $J_{ij}(x_i, x_j)$ means $J_{ji}(x_j, x_i)$ when $j < i$ (this notation is reasonable, since the coupling between the residue x_i at position i and the x_j at position j is the same as the coupling between the residue x_j at position j and the x_i at position i , i.e., the coupling is undirected).

This quantity does not contain the forbidding and bothersome normalization. In a sense, normalization is still going on though; the denominator \mathbf{Z}_i can be seen as the ‘new \mathbf{Z} ’, specified to the position i . The dependent variable l takes on just q states (contrasted to with q^L states in the original \mathbf{Z}), so this normalization is compatible with a large L .

Based on the above analysis, given an MSA, the negative logarithm pseudo log-likelihood function corresponding to the i^{th} position reads

$$-\log \text{PL}_i = - \sum_{m=1}^M w_m \left[h(x_i^m) + \sum_{j=1, j \neq i}^L J_{ij}(x_i^m, x_j^m) - \log \mathbf{Z}_i \right] \quad (2.28)$$

where

$$\mathbf{Z}_i = \frac{1}{M_{\text{eff}}} \sum_{m=1}^M w_m \ln \left[\sum_{l=1}^q \exp \left(h_i(l) + \sum_{j=1, j \neq i}^L J_{ij}(l, x_j^m) \right) \right]. \quad (2.29)$$

The w_m is the weight for the m^{th} sequence in the MSA.

Sequential and Parallel Negative Logarithm Pseudo-likelihood Minimization

There are two ways to approximate the parameter set $\{h, \mathbf{J}\}$. The straightforward one is to minimize the combined L negative logarithm pseudo-likelihood functions:

$$\{h^\#, \mathbf{J}^\#\} = \arg \min_{h, \mathbf{J}} \sum_{i=1}^L (-\log \text{PL}_i) \quad (2.30)$$

and solve the parameter set with only one optimization process. This is called the sequential negative logarithm pseudo-likelihood minimization. GREMLIN uses this strategy and was shown to be able to achieve more accurate amino acid contact prediction than both mfDCA and PSICOV (Balakrishnan et al. 2011). However, this optimization has the drawback of being slow (Ekeberg et al. 2014).

Another method is to minimize each $-\log \text{PL}_i$ separately, one is independent of another. Many modern computers have multiple CPU cores. So, it is possible to send the calculation of each $-\log \text{PL}_i$ to a different core and join the individual parameter subset $\{h_i^\#, J_{i*}^\#\}$ to form the final $\{h^\#, J^\#\}$. This method is called the parallel negative logarithm pseudo-likelihood minimization, since it can be easily programmed using parallel computing. Obviously, it is much faster than the former one. However, this method also suffers from a problem. For any inferred coupling $\mathbf{J}_{ij}^\#$, it can come from both $J_{i*}^\#$ and $J_{*j}^\#$, which are named $J_{ij}^{\#i}$ and $J_{ij}^{\#j}$. The two $\mathbf{J}_{ij}^\#$ can be different. A compromise proposed by Ekeberg *et al.* (Ekeberg et al. 2014) was shown to achieve almost the same amino acid contact prediction accuracy as the sequential one:

$$\mathbf{J}_{ij}^\# = \frac{1}{2} \left(J_{ij}^{\#i} + J_{ij}^{\#j} \right). \quad (2.31)$$

Due to the speed advantage, the following description of plmDCA only focuses on the parallel one.

Although one can find the solutions of $\{h, J\}$ by gradient descent, practically, due to the huge number of free parameter in the set, overfitting cannot always be avoided (Ekeberg

et al. 2014). The technique of regularization partially solves the issue (Ekeberg et al. 2014, 2013) of overfitting.

Regularization

In Equation 2.28, l_2 regularization, which is the sum of all squares, was used. The l_2 regularizer forces a finite fraction of parameters to assume value zero, thus effectively reducing the number of parameters. Instead of minimizing $-\log \text{PL}_i$, one minimizes $-\log \text{PL}_i + R_{l_2}$ with

$$R_{l_2}(h, J) = \lambda_h \sum_{i=1}^L \sum_{a=1}^q \|h_i(a)\|_2^2 + \lambda_J \sum_{1 \leq i < j \leq N} \sum_{a,b=1}^{q,q} \|J_{ij}(a, b)\|_2^2 \quad (2.32)$$

λ_h and λ_J are regularization strengths for the couplings and the fields to be specified by the user. Suitable values were found to be $\lambda_h = \lambda_J = 0.01$ (Ekeberg et al. 2014, 2013).

So, instead of minimizing $-\log \text{PL}_i$,

$$\begin{aligned} g_i^{(reg)} &= -\log \text{PL}_i + \lambda_h \|h_i\|_2^2 + \lambda_J \sum_{j=1, i \neq r}^L \|J_{ij}\|_2^2 \\ &= -\frac{1}{M_{\text{eff}}} \sum_{m=1}^M w_m \left\{ h_i(x_i^m) + \sum_{j=1, i \neq j}^N J_{ij}(x_i^m, x_j^m) - \ln \sum_{l=1}^q \exp \left[h_i(l) + \sum_{j=1, i \neq j}^L J_{ij}(l, x_j^m) \right] \right\} \\ &\quad + \lambda_h \|h_i\|_2^2 + \lambda_J \sum_{j=1, i \neq r}^L \|J_{ij}\|_2^2 \end{aligned} \quad (2.33)$$

is minimized.

From the above equation, the partial derivatives are calculated as (cited from (Ekeberg et al. 2014)):

$$\frac{\partial g_i^{(\text{reg})}}{\partial h_i(a)} = -\frac{1}{M_{\text{eff}}} \sum_{m=1}^M w_m \left(I[x_i^m = a] - \frac{\exp[h_i(a) + \sum_{j=1, i \neq j}^L J_{ij}(a, x_j^m)]}{\sum_{l=1}^q \exp[h_i(l) + \sum_{j=1, i \neq j}^L J_{ij}(l, x_j^m)]} \right) + 2\lambda_h h_r(s) \quad (2.34)$$

$$\frac{\partial g_r^{(\text{reg})}}{\partial J_{ij}(a, b)} = -\frac{1}{M_{\text{eff}}} \sum_{m=1}^M w_m I[x_i^m = a] \left(I[x_j^m = b] - \frac{\exp[h_i(a) + \sum_{j=1, i \neq j}^L J_{ij}(a, x_j^m)]}{\sum_{l=1}^q \exp[h_i(l) + \sum_{j=1, i \neq j}^L J_{ij}(l, x_j^m)]} \right) + 2\lambda_J J_{ij}(a, b) \quad (2.35)$$

$g_r^{(\text{reg})}$ is a smooth function, which means that minimizing $g_r^{(\text{reg})}$ is to find the point at which these derivatives are all zeros (Ekeberg et al. 2014).

Scoring

This step deals with how to choose values from the inferred parameter set $\{h, J\}$ to make amino acid contact prediction. Weigt *et al.* used the direct information (DI) to measure the interaction strength between any pair of amino acid sites in their pioneering mfDCA work (Morcos et al. 2011). However, instead of DI, Ekeberg et al. used Frobenius norm (FN) followed by the APC correction (Ekeberg et al. 2014, 2013). There is not a clear theoretical reason why one should favour this approach over DI, but the results in the study by (Ekeberg et al. 2014, 2013) indicate that it seems to achieve better amino acid contact prediction in the context of plmDCA, and thus it was also the scoring function of choice in this work.

For any pair of sites i and j , the Frobenius norm is defined as

$$\text{FN}_{ij} = \sqrt{\sum_{a=1, b=1}^{q, q} J_{ij}(a, b)^2} \quad (2.36)$$

After being corrected by the APC, the final score which indicates the amino acid coupling strength is

$$S_{ij} = \text{FN}_{ij} - \frac{\text{FN}_{\bullet j} \text{FN}_{i \bullet}}{\text{FN}_{\bullet \bullet}}. \quad (2.37)$$

2.6.10 Machine Learning Based Predictors

All of the above three global statistical approaches were shown to be able to detect structural amino acid contacts more accurately than local statistical approaches (Ekeberg et al. 2013; Jones et al. 2015; Morcos et al. 2011). plmDCA was reported the best among them. It is worth knowing whether plmDCA could predict all of the contacts that mfDCA and PSICOV predict, or in other words, whether mfDCA or PSICOV could predict some contacts which plmDCA couldn't. Some groups have investigated the differences in the contacts predicted by these three approaches (Jones et al. 2015; Skwark et al. 2013). Figure 2.7a shows the overlap of correctly predicted contacts and Figure 2.7b illustrates the overlap of the incorrectly predicted contacts from the three approaches for the same set of MSAs. It is clear to find that the different methods result in a number of correct predictions, as well as incorrect predictions, which are unique to that method, even if the

majority of correct contacts are identified by all of the three. So, it is a good idea to combine the three approaches to generate a meta-predictor via machine learning techniques with the aim of improving amino acid contact precision.

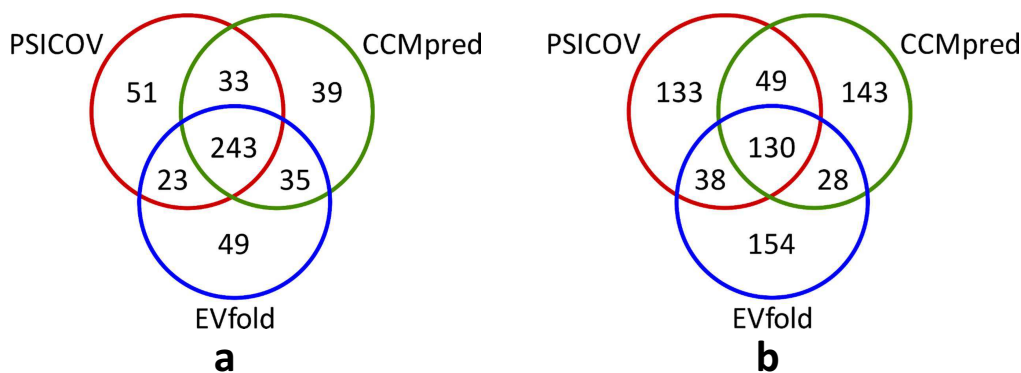


Figure 2.7. Diagram of the overlap of the correct contact predictions (a), and incorrect contact predictions (b) by PSICOV, EVfold (mfDCA) and CCMpred (plmDCA) from 1050 predictions. This figure is adopted from the study by (Tetchner 2016).

The first developed machine learning based contact predictors that combine the amino acid coevolutionary couplings from multiple global statistical models by machine learning were PconsC (Skwark et al. 2013) and a modified version of PconsC (PconsC2) (Skwark et al. 2014). These methods combine the coevolutionary couplings from plmDCA and PSICOV that are calculated based on eight different MSAs for a query protein sequence by using a random forest classifier. Of the eight MSAs, four are generated by JackHMMER (an iterative profile HMM program in the HMMER suite), with different expectation values (10^{-40} , 10^{-10} , 10^{-4} , 1), and four are generated by another program, HHblits, with the same four expectation values (Skwark et al. 2013). Besides the eight contact predictions, PconsC2 also used predicted SS, accessible solvent area and amino acid profile as features. In addition,

PconsC2 also changed the architecture compared to PconsC: a five-layer stack of random forests was used. From the second layer, the outputs from the preceding layer were fed into the input of the current layer, together with the above-mentioned features. PconsC and PconsC2 improved amino acid contact prediction performance compared to the individual statistical methods (Skwark et al. 2013, 2014). However, they both suffer from the drawback of being slow for contact prediction calculation (Michel et al. 2014; Wang et al. 2017a). The later proposed MetaPSICOV approach (Jones et al. 2015), which combines predictions from mfDCA, PSICOV and plmDCA along with other structural predictions such as the secondary structure prediction and solvent accessibility prediction, and protein primary structure properties such as amino acid profiles, reduces a lot of computational time (Jones et al. 2015), as compared with PconsC and PconsC2. MetaPSICOV uses feedforward neural networks and can make better amino acid contact predictions than the original PconsC (Jones et al. 2015) and PconsC2 (de Oliveira et al. 2016).

MetaPSICOV consists of two stages. In stage 1, the neural network has a single hidden layer with sigmoid activation and 55 hidden units. For each pair of sites, the input layer is fed with a 672-dimensional feature vector for each pair of residue on the target sequence and outputs a value in the range of 0–1 to indicate how likely the residue pair is in contact (0 indicates very unlikely and 1 very likely to be in contact). Stage 2 is another neural network with the same architecture as stage 1. The main difference is that the contact predictions generated by stage 1 are used as an input feature of stage 2. This iteration strategy allows stage 2 to be able to predict more accurate amino acid contacts than

stage 1 in most cases. By combining machine learning techniques and coevolution-based predictions, MetaPSICOV is able to successfully deal with proteins with different lengths (the number of amino acids). In the paper of MetaPSICOV ([Jones et al. 2015](#)), the authors report that, if the MSA of the query protein has poor quality, MetaPSICOV downweights the coevolutional coupling signal and promotes the weights of other structural properties (e.g., the secondary structure prediction, amino acid profile). Conversely, if the MSA has sufficient homologous sequences, the coevolutional coupling signal is upweighted.

Summarized comparisons between the above three machine learning based amino acid contact prediction approaches can be found in [Table 2.1](#).

Table 2.1. Comparisons between PconsC, PconsC2 and MetaPSICOV.

#	PconsC	PconsC2	MetaPSICOV
Machine Learning Type	Random Forest (100 decision trees)	5 Layers of Random Forests (each layer has 100 decision trees)	Two stages of three-layer feedforward neural networks
Feature	PSICOV and plmDCA contact predictions based on 8 MSAs	PSICOV and plmDCA contact predictions based on 8 MSAs, sequence separation, predicted SS and accessible solvent area, amino acid profile. These features are used in the first layer; from the second layer, these features together with the preceding layer outputs are used in the inputs.	mfDCA, PSICOV and plmDCA contact predictions, MI, statistical potential, position entropy, amino acid profile, predicted SS and accessible solvent area, sequence separation, sequence length, the number of sequences in the MSA, the number of effective sequences in the MSA. These features are used in the neural network of stage 1; for stage 1 neural network, besides these features, the output from stage 1 is also used.
Training Set Size	48 proteins	150 proteins	672 proteins
Performance	Better than mfDCA (evfold), PSICOV and plmDCA (Skwark et al. 2013)	Better than PconsC (Skwark et al. 2014)	Better than both PconsC and PconsC2 (de Oliveira et al. 2016; Jones et al. 2015)

Besides the above three methods, there are another three recently published amino acid contact prediction algorithms, namely plmConv (Golkov et al. 2016), NeBcon (He et al. 2017) and RaptorX (Wang et al. 2017b). They all claim to make more accurate amino acid contact predictions than MetaPSICOV. The following is a brief introduction to them.

In the paper of plmConv, a 2D three-layer convolutional network was employed and co-evolutional couplings calculated by plmDCA were included in the features. Mean squared error was used as the loss function, and the Adam algorithm (Kingma and Ba 2014) as the training function during the network training process.

NeBcon includes two steps. In the first step, contact scores are firstly predicted by eight representative predictors, which include PSICOV, CCMpred, Freecontact, MetaPSICOV, etc., for the query sequence; a naive Bayes classifier is then used to calculate the posterior probability score for each contact score matrix. In the second step, six features (secondary structure prediction, solvent accessibility prediction, Shannon entropy, residue separation, residue composition and residence; here residue composition is the amino acid frequency of the MSA of the query sequence and the residence is binary value to indicate the location of the residue pair being considered) are extracted from the query sequence; they, together with the posterior probability scores, are fed into a neural network to report the final contact score.

RaptorX predicts amino acid contacts by combining both sequence information and evolutionary coupling; it uses an ultra-deep neural network (60 layers in total) model formed by two deep residual neural networks. In the first residual neural network, 1D features including the sequence profile (the amino acid frequency), the secondary structure prediction and the solvent accessibility prediction are used, and the output of this network is converted to a 2D matrix. This output, together with pairwise features, are used as the input in the second residual neural network. The pairwise features include coevolutional

couplings predicted by CCMpred, pairwise contact and distance potentials. The output from this network reports the final contact score.

2.7 Feature Selection and Machine Learning

As mentioned in the above paragraphs of the previous section, the latest coevolution based amino acid contact prediction methods use machine learning models. Besides the coevolutionary couplings calculated from the local/global statistical algorithms, other protein 1D or 2D properties are also usually calculated and used as features. In this section, some of the commonly used features are introduced (these features are also used in DeepCDpred). The brief introduction to the development of machine learning, especially the neural network model, could help readers to understand the models proposed in this study.

2.7.1 Secondary Structure Prediction

The secondary structure of a protein not only provides an approximate idea about the overall structural category it belongs to, but also can define geometry constraints for the tertiary structure prediction when its known 3D structure is not available in PDB.

Protein secondary structure prediction requires the definition of the secondary structure. The most commonly used standard is the secondary structure assignment method, DSSP

(Dictionary of Secondary Structure of Proteins) ([Kabsch and Sander 1983](#)), which defines eight states of secondary structure: H (α -helix), G (3_{10} helix), I (π -helix), E (extended strand in parallel and/or anti-parallel β -strand conformation), B (β -bridge), S (bend), T (turn) and C (coil) based on hydrogen bonding patterns. These eight states are further simplified into three states – helix (H, G and I), strand (E and B) and coil (S, T and C) ([Muppalaneni and Gunjan 2015](#)).

In order to assign one type of secondary structure for each residue in a protein, a number of sophisticated computational approaches have been developed in the past several decades for protein secondary structure prediction. The approaches can be classified into two main groups: simple statistical methods and machine learning based methods. The highlight of the former is that the biological meanings are comprehensible ([Chou and Fasman 1974](#)). They generally assign secondary structures for a given protein according to the statistical propensities of amino acid residues towards a specific secondary structure element. However, the maximum Q3 accuracy of them is only around 65% ([Muggleton et al. 1992](#)). Here, Q3 is the total number of correctly predicted residue states divided by the total number of residues. By contrast, a major advantage of machine learning, especially deep learning based methods, is that different information can be incorporated into a prediction model ([Zhang 2015](#)). Among them, neural-network based models have seen the highest reported accuracy ([Yang et al. 2016](#)). These methods typically rely on a sequence profile (position-specific substitution matrix, PSSM) derived from multiple sequence alignment of homologous sequences. Using PSSM implicitly assumes that homologous sequences

have the same secondary structures (Yang et al. 2016).

In a newly published paper, some protein secondary structure prediction methods were reviewed (Yang et al. 2016). The authors compared seven state-of-the-art algorithms – Jpred 4 (Drozdetskiy et al. 2015), SCORPION (Yaseen and Li 2014), Porter 4.0 (Mirabello and Pollastri 2013), PSIPRED 3.3 (Buchan et al. 2010; McGuffin et al. 2000), SPINE X (Dor and Zhou 2007), SPIDER2 (Structural Property prediction with Integrated DEep neuRal network 2) (Heffernan et al. 2015) and DeepCNF (Deep Convolutional Neural Fields) (Wang et al. 2016b) by using the same test set (115 X-ray crystallography solved proteins released between 1 January 2016 and 20 September 2016). The three-state accuracies reported in this paper were 77.1% by Jpred 4, 80.1% by SPINE X, 80.2% by PSIPRED 3.3, 81.7% by SCORPION, 81.9% by SPIDER2, 82.0% by Porter 4.0 and 82.3% by DeepCNF. Here, the prediction accuracy is measured by the Q3 score. Except Porter 4.0, all of the algorithms used PSSM calculated from a sequence database with PSI-BLAST as inputs for each machine learning model (Buchan et al. 2010; Dor and Zhou 2007; Drozdetskiy et al. 2015; Heffernan et al. 2015; McGuffin et al. 2000; Wang et al. 2016b; Yaseen and Li 2014). Porter 4.0 used the frequencies of 25 amino acids (20 standard amino acids plus B (D or N), U (selenocysteine), X (unknown), Z (E or Q) and the gap) as inputs. In addition, Jpred 4 also used the HMM profile that is obtained from the MSA of the target sequence as input features (Drozdetskiy et al. 2015). SPINE X also used seven physical parameters including the graph shape index, hydrophobicity, volume, polarizability, isoelectric point, helix probability, and sheet probability, as the

input features (Faraggi et al. 2012). As for outputs, all of these algorithms generated a three-state probability vector for the target sequence; each position in the vector includes three elements which represent the probabilities of the secondary structure states, helix and strand, and coil/loop. SPIDER2 (Heffernan et al. 2015) also reported the ϕ and ψ angles. DeepCNF (Wang et al. 2016b) reported an eight-state probability vector for the eight-state secondary structure prediction (the definition of the eight states was already mentioned at the beginning of this section).

At the time of developing the code of the proposed amino acid contact/distance prediction algorithm (i.e., DeepCDpred) in this thesis, the codes of Porter 4.0 and DeepCNF were not available, so SPIDER2 was used for the secondary structure prediction and embedded in the automatic amino acid contact/distance prediction pipeline of DeepCDpred. In future work, Porter 4.0 and DeepCNF will be tried to replace SPIDER2 to improve contact/distance accuracy. Please see Section 6.6 in Chapter 6 (Discussion chapter) for more discussions.

SPIDER2 (Heffernan et al. 2015) applied deep neural networks, which refer to neural networks with more than one hidden layer, for secondary structure prediction. Three hidden layers were used in SPIDER2 and each consisted of 150 neurons. Weights were learned by the standard backward propagation algorithm. SPIDER2 used an iterative simultaneous improvement of secondary structure, backbone torsion angles and solvent accessibility (solvent accessible surface area) by using three iterations. The original publication reported that the method achieved an average Q3 accuracy of 81.8% for an independent

test on 1199 high-resolution proteins (<2.0 Å) (Heffernan et al. 2015).

DeepCNF was published in 2016, one year later than SPIDER2. It combined a deep convolutional network and a conditional random field. The deep convolutional network (7-layer) captured long-range sequence information and the conditional random field models the SS labels of nearby residues (Heffernan et al. 2015). In the paper, the authors claimed that DeepCNF predicts better 3-state SS than JPRED, SPINE-X and RaptorX-SS8 on the CASP10, CASP11 and CAMEO (<https://www.cameo3d.org> (last check: November 2018); like CASP, CAMEO is a world-wide protein structure prediction competition platform) proteins.

Secondary structure prediction has already been used in the machine learning based amino acid contact prediction algorithms (Cheng and Baldi 2007; Jones et al. 2015; Skwark et al. 2014; Wang et al. 2017b).

2.7.2 Other Features

Besides the secondary structure prediction, a couple of other structural properties have been included in machine learning based contact prediction models. For example, amino acid solvent accessibility, site entropy and amino acid composition, sequence separation (number of amino acids along the sequence) between the two residues, the number of sequences in the MSA were scored and calculated (Jones et al. 2015). These properties aim at capturing different structural characteristics of a protein. Usually, a local window

with a specific width is used for the position of the residue being considered, to include the effect of neighbouring residues (Jones et al. 2015). That is, properties of interest within the window are taken as features that will be fed into a machine learning model. After the model has been trained and validated, a structural target (e.g., amino acid contact or hydrogen bonding pattern in the structure) will be obtained from the output.

2.7.3 Machine Learning and Artificial Neural Network

Machine learning (ML) can be defined as “the study of computational methods and the construction of computer algorithms and programs capable of learning from their own previous experience, in order to improve their performance at a defined task” (Mitchell 1997). It has become more and more popular in recent years, in some areas even surpassing human-level performance (He et al. 2015; Tang and Xiaoou 2014). Representative applications of ML include the development of self-driving cars (e.g., Google’s self-driving car), automatic language translation (Zhang and Zong 2015), automatic text generation (Yu et al. 2017), object recognition in images and videos (Kang et al. 2016; Ren et al. 2015; Szegedy et al. 2015) and disease prediction in healthcare (Chen et al. 2017).

ML can be generally grouped into three categories.

1. Supervised learning: in this case, each input in the training data is associated with a desired output. The main aim is to learn a function (continuous or discrete) from the inputs and the desired outputs which minimize the difference (“loss”) between the

desired outputs and the predicted outputs. Predictions could then be made with this function for new inputs. The main tasks associated with this kind of learning are classification and regression.

2. Unsupervised learning: in this case there is no outcome and the aim is to describe the associations and patterns among a set of inputs ([Hastie et al. 2001](#)). The principal tasks associated with this kind of learning are clustering and association rules.
3. Reinforcement learning: in this case, there is also no supervisor, but a reward system instead; the system provides a feedback (reward signal) to the agent to indicate how well it is doing, and the agent needs to take actions to maximize the expected cumulative reward.

This thesis focuses on a supervised classification task. Classification is an example of pattern recognition. Classes (or groups) are defined by the categories of the desired outputs in the training data. By learning a model between the inputs and outputs in the training data, the aim of classification is to determine which class a new input belongs to. Groups can be a simple binary partition (e.g., a pair of amino acid residues in “contact” or “not in contact” , the problem addressed in this thesis), or a complex with multiple partitions. A good example of the multiple-category classification is recognizing handwritten digits (the 10 numbers from 0 to 9) in the MNIST database ([Schmidhuber 2015a](#)).

Conventional machine learning algorithms include decision trees, naïve-Bayes classifiers, support vector machines, Logistic regression and neural networks (NNs). NNs are the

method used in this thesis. The core idea of NNs is inspired by biology and attempts to mimic the behaviour of neurons in the brain. They are made of basic units (the neurons), which propagate signals to the other connected units. NNs can be viewed as general input-output relationship estimators. The estimation is achieved after a learning process from a set of samples in the training set. NNs are commonly used in classification, regression and clustering. This section briefly describes the characteristics of NNs.

Usually, an NN organizes the neuron units into several layers. The first layer is called the input layer, and the last layer is the output layer. The layers (if any) between these two are called hidden layers. For a feedforward NN, the input feature vectors from the input layer through the hidden layer(s), and finally to the output layer. For the recurrent NN, the output from the hidden layer(s) not only goes to the next layer (another hidden layer or output layer), but also returns to the input layer after a time delay and then combines with new inputs to go into the NN again.

Neural networks are usually trained by backpropagation and optimized by stochastic gradient descent. The learning process is divided into the forward pass and the backward pass. In the forward pass, the input goes through the network and produces an output. The output is then compared to the target data by using a loss function and the resultant gradient is propagated through the network in the backward pass. In the process, the weights of each layer are slightly changed in a way that reduces the error. This is repeated until a stop condition is reached. Simple stop conditions can be a threshold to the error on the training set, or the number of iterations.

Although an NN model may not be the best choice in some tasks ([Caruana and Niculescu-Mizil 2006](#)), it does have some advantages for the amino acid contact/distance prediction study that will be introduced in the later chapters. Two main characteristics of this study are the large number of inputs (tens of millions) and high dimensions (≈ 750) of the feature vector in the training dataset. NN models are known to be more suitable for dealing with large datasets ([Chau et al. 2014](#)). The use of mini-batch and stochastic gradient descent optimization could easily speed up the training process of a deep network model on a platform with multi-CPU or multi-GPU. See the results shown in [B.III](#) (Appendix [B](#)), trained with the same set, DeepCDpred produced a higher accuracy of amino acid contact prediction than an SVM model and a random forest model on the same test set.

The research community has seen very fast development of both NN related libraries and newly proposed network models. Such libraries include Tensorflow ([tensorflow 2018](#)) (backed by Google, supports Python, Java, R and C++), Keras ([Keras 2018](#)) (supports Python, Java and R), Deeplearning4j ([Deeplearning4j 2018](#)) (supports Java and Scala), PyTorch ([PyTorch 2018](#)) (backed by Facebook, supports Python), CNTK ([CNTK 2018](#)) (backed by Microsoft, supports Python, Java, C++ and C#/.NET) and Caffe ([caffe 2018](#)) (supports Python and C++). The neural network toolbox and statistics & machine learning toolbox in MATLAB also provide easy to use functions for deep learning studies. The first version of Tensorflow was released in November of 2015, and now (later 2018), it has already become the most popular deep learning framework estimated by the number of stars on Github ([github/tensorflow 2018](#)).

Furthermore, deep learning models have begun to outperform humans on some tasks regarding performance ([Karam and Lina 2017](#)) in recent years. A well-known example is the AI (artificial intelligence) Go (an ancient Chinese game) player AlphaGo (developed by the company of DeepMind) which beat a top world ranking player. Go uses a convolutional network guided tree search strategy to find the best play positions in the next steps ([Silver et al. 2016](#)). Another example is the image classification on the ImageNet dataset ([Wikipedia/ImageNet 2018](#)). The residual network proposed in 2015 achieved an error rate of 3.57%, which is lower than the human error rate of 5.1% ([He et al. 2016](#)).

Limitations of neural networks also exist. A large number of parameters and the somewhat non-schematic training procedure are regarded as downsides of NNs. Some of the parameters include the overall architecture (i.e. the number of hidden layers and hidden units), the activation functions, the weight initialization, the learning rate and the choice of momentum. Activation functions may themselves have additional parameters. When the network goes deeper (contains multiple hidden layers), activation functions such as ReLU ([Nair and Hinton 2010](#)), ELU ([Djork-Arne Clevert 2015](#)) and SeLU ([Klambauer et al. 2017](#)) may reduce the vanishing gradient problem and give the network a better performance than traditional sigmoid functions. Due to the shapes of different activation functions, for pattern recognition problems, sigmoid (two categories) or softmax (two or more categories) activation functions should be used in the output layer, while for function regression problems, linear activation should be used instead.

Classical neural networks are considered to have shallow architectures often composed of

0 or 1 hidden layers. Deep Neural Networks (DNNs, i.e. with more than one hidden layer, Figure 2.8) and other related deep learning techniques have become popular tools in recent years for the studies of many problems including the tasks of speech recognition, image recognition, object detection and natural language processing (LeCun et al. 2015; Schmidhuber 2015a). Classic neural networks are often limited by the complexity of the pattern they try to learn. These models need careful selections of input features. While, DNNs have been shown to be better than classic neural networks in many fields, since they are able to learn intermediate representations, and each layer learns slightly more detailed and more abstractions from the input data than the previous layer (Bengio et al. 2013; LeCun et al. 2015). Thus, DNNs with multiple layers can recognise very complicated patterns within the input data (Bengio 2009).

In summary, neural networks are a powerful approach to deal with multiclass classification problems. Their setup allows to capture high order correlations from inputs. Through the introduction of more hidden layers, this ability can be increased even further, also referred to as the deep learning. However, introducing more hidden layers could also increase the risk of making the system over specified and prone to overfitting. This problem can be mitigated by the strategies of “early stopping”, regularization and dropout (Srivastava et al. 2014). Early stopping is realized by introducing a separate validation set and checking the prediction accuracy on it after every several epochs (e.g. 6) when the network is training. The training process is stopped when the accuracy of prediction on the validation set ceases to improve for a specific number of continuous epochs. The goal

of regularization (e.g., L_1 or L_2) on the weights is to impose the value of some weights to approach zero during the training process.

Input Layer Hidden Layer Hidden Layer Hidden Layer Output Layer

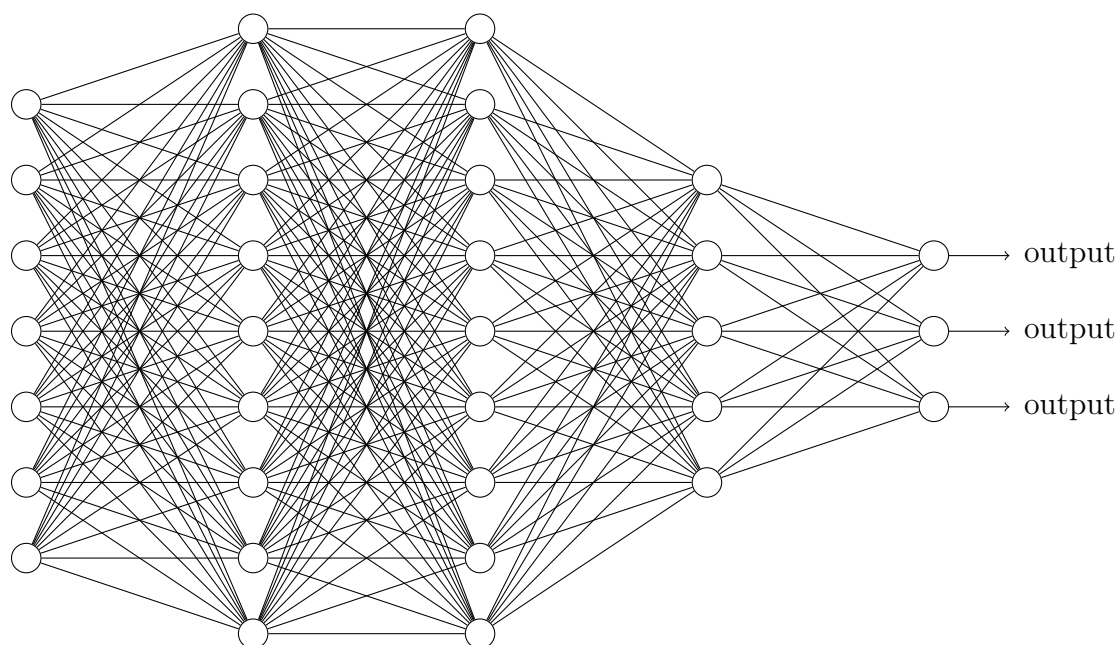


Figure 2.8. A three-hidden-layer deep neural network

2.8 Protein Structure Prediction

2.8.1 Introduction

In this section, the background of computational algorithm based protein structure prediction is introduced. It is followed by the introduction and discussion of the two types of protein structure prediction approaches. Then, the well-known competition in the field of protein structure prediction, CASP (Critical Assessment of protein Structure Prediction),

is briefly introduced. In the end, two methods of protein structure similarity comparisons are described and compared.

As mentioned in Chapter 1, protein sequences are accumulated with an ever-increasing speed. The development of metagenome sequencing in recent years even accelerates this trend (Mitchell et al. 2016; Oulas et al. 2015). However, experimentally resolved structures are accrued much slower. For example, by the end of 2016, the number of protein structures deposited in PDB was about 125,000, accounting for only 0.14% of sequences in the UniProtKB database at that time. The increasing gap between the number of protein structures and the deluge of protein sequences is clearly displayed in Figure 2.9.

As the experimental determination of protein structures by either X-ray crystallography or NMR is time-consuming and expensive, developing an efficient computer-based algorithm to predict 3D structures from sequences is attractive and worthwhile.

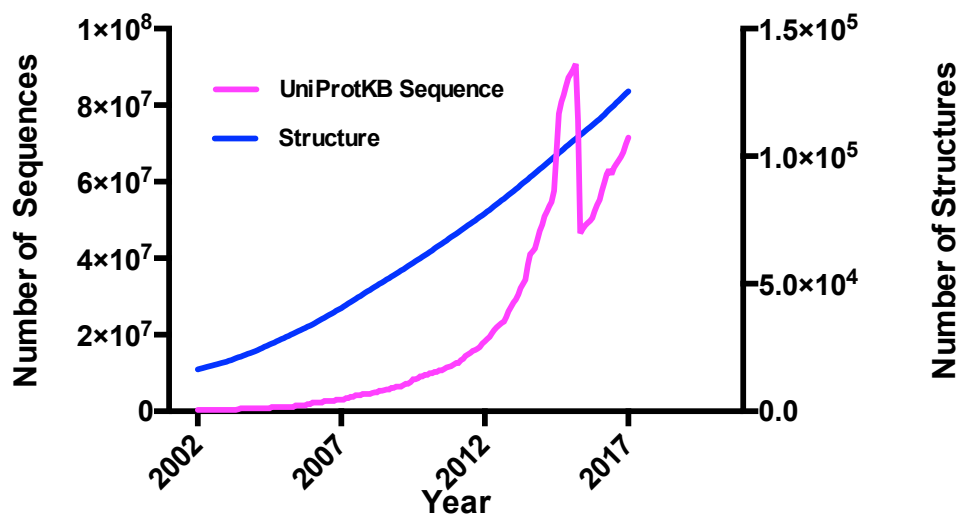


Figure 2.9. The gap between the numbers of protein sequences and structures are becoming larger over time. The sharp drop in UniProtKB entries resulted from a proteome redundancy minimization procedure implemented in March 2015 (The UniProt 2017). Data are obtained from <http://www.rcsb.org/pdb/statistics/holdings.do> & <http://www.uniprot.org/statistics/> (last check: November 2018).

Decades of intense research in bioinformatics has brought about a huge progress in dealing with the ever-increasing amount of protein data. In an attempt to bridge the ‘sequence-structure gap’, computational approaches have long been devised with the aim of predicting the tertiary structure of proteins from the sequences (Figure 2.10). A number of different approaches have been proposed over the years. They can be broadly divided into two categories: template-based modelling and template-free modelling, which are briefly introduced in the next two subsections.

LFKLGAENIFLGRKAATKEEAIRFAGEQLVKGGYVEPEYVQAMLDREKL
TPTYLGESIAVPHGTVEAKDRVLKTGVVFCQYPEGVRFGEEDDIARLV
IGIAARNNEHIQVITSLTNALDDESVIERLAHTTSVDEVLELLAGRK

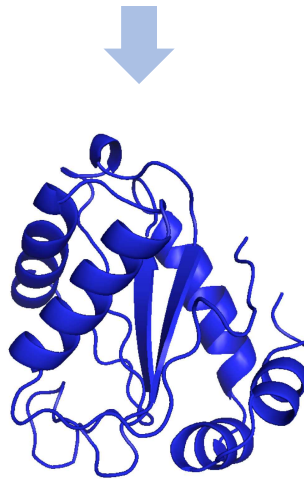


Figure 2.10. From protein sequence to protein structure. Anfinsen theory suggests that in the native environment, the 3D structure of a protein is uniquely determined by its amino acid sequence. The sequence and structure of the protein pdbid 1a3a (IIA mannitol from *Escherichia coli*) are used here.

2.8.2 Template-Based Modelling

Compared to free modelling, template-based modelling still holds the leading role in predicting the tertiary structure of a protein regarding the prediction accuracy. Template-based modelling refers to building a 3D model for a given protein sequence based on the information from previously solved 3D structures in PDB.

To predict a protein structure by template-based modelling, two requirements have to be met (Fiser 2010). Firstly, the sequence to be modelled must have at least one template

of known structure; and secondly, it must be possible to compute an accurate alignment between the target sequence and the template structure. The alignment provides structural equivalences that could be used as geometry constraints for the target sequence. The accuracy of the alignment becomes even more important when the sequence identity between the target and template falls to less than 40% (Fiser 2010). Depending on whether there is detectable sequence similarity between the target and the sequence of the template (in other words, whether the target sequence and the template sequence belong to the same protein family based on their sequence similarity), template-based modelling can be further categorized into two types – homology modelling and threading.

Homology modelling, also known as comparative modelling, is usually used when the sequence identity between two proteins is above $\approx 30\%$ (Rost 1999). Structural similarity can usually be then assumed. The principle idea behind homology modelling is that protein structure is more conserved than sequence, and most protein pairs were found structurally similar when the sequence identity is higher than 30% (Rost 1999). When one structure has been experimentally determined in a protein family, other members of this family can be modelled guided by information from it (the known structure).

As the sequence identity falls below $\approx 30\%$ (in the so-called ‘twilight zone’ (Rost 1999)), homology modelling becomes unreliable, since protein sequences have accumulated so many mutations that the relationship between sequence and structure similarity gets increasingly blurred (Bordoli et al. 2009). Threading is used for this situation. It is devised to match the target sequence directly onto the 3D structures with the objective of

scoring how likely they are. The idea of threading is based on the observation that there are many fewer distinct folds than sequences due to physical constraints in the structure space (Govindarajan et al. 1999). Thus, threading is also named fold recognition.

Generally, the operating procedure of template-based modelling consists of four steps:

1. finding the proteins of known structure related to the target sequence (for homology modelling, this step tries to find homologues with known structures; whereas for threading modelling, with looser constraints, it tries to find proteins with known structures that likely have a similar arrangement of secondary structures);
2. aligning each residue of the target sequence onto the structures of templates;
3. generating a backbone of the target protein by copying aligned regions, or by satisfying some kinds of spatial constraints with templates;
4. modelling unaligned loops and side-chains.

For the first two steps, homology modelling algorithms usually detect template structures in PDB with PSI-BLAST, HMMER, HHsearch or HHBlits. The alignment between the sequence of the target protein and the sequence of the template structure can also be obtained by these programs. Threading algorithms commonly use score functions to detect template structure(s) from a protein structure database (e.g. PDB, SCOP and CATH)

for the target sequence. These score functions include terms such as sequence profile-profile alignment (Soding 2005), secondary structure match (Khor et al. 2015), inter-residue contact match (Zhang and Skolnick 2004b), and statistical potential (Skolnick and Kihara 2001). However, none of the threading methods that use only one term of the score function can outperform others for all of the target protein structure predictions (Khor et al. 2015). For this reason, meta-methods that combine the output models from multiple programs have been developed (Ginalski et al. 2003; Wu and Zhang 2007). Among them, LOMETS (Local Meta-Threading Server) (Wu and Zhang 2007), which selects consensus models from nine threading servers, is used as the template identification program of I-TASSER (Iterated Threading ASSEMBly Refinement) (Roy et al. 2010; Yang et al. 2015). The latter was ranked as No.1 server in the CASP experiments (more introduction about CASP can be found in Subsection 2.8.4 of this chapter) from CASP6 (2006) to CASP12 (2016), except CASP9 (2010) in which it was ranked as No.2 (CASP 2018). Due to this great success, a brief description of I-TASSER is introduced as follows.

After the template structure(s) is identified by LOMETS, I-TASSER uses a Monte Carlo simulation method (more explanations about Monte Carlo simulation can be found in the next section), TASSER (Zhang and Skolnick 2004a), for template assembly. For regions not covered by the template(s), *ab initio* modelling is used. After this step, multiple full-length models are generated, which are then clustered by SPICKER (Zhang and Skolnick 2004c). Centroids obtained by averaging the 3D coordinates in each cluster are used to search for similar PDB structures with TM-align (Zhang and Skolnick 2005). Geometry

constraints from the template(s) identified by LOMETS and new PDB structures are then used to generate new models. This round of “*iteration is to remove steric clashes as well as to refine the global topology of the cluster centroids*” (Roy et al. 2010). Low-energy models are selected to feed into REMO (Li and Zhang 2009) to generate the final full-length models.

MODELLER (Sali and Blundell 1993) is one of the most frequently used template-based modelling programs. It builds models of the target protein by deriving constraints from the target-template alignment(s). It assumes that the spatial distances and angles between equivalent residues are similar (Fiser 2010). The distance and angle constraints can also be expanded by adding non-bonded atom-atom contacts, bond lengths and bond angles (Webb and Sali 2014). To account for these uncertainties of the quality and reliability of the alignment(s), each constraint is specified by a probability density function (pdf). By using this method (the use of pdf), MODELLER allows the incorporation of information from multiple templates. Taking the distance between two given atoms in the target sequence as an instance, it is assumed to be similar to the corresponding distances in the template structures and is thus modelled by a Gaussian distribution. If enough distance constraints are specified, the 3D structure of the query protein can be accurately predicted. In the optimization step, MODELLER determines the best query structure by maximizing the joint probability.

Template-based modelling has both advantages and limitations. Because of the much smaller conformational searching space compared to *ab initio* modelling, template-based

modelling, especially homology modelling, achieves more accurate protein structure predictions. For example, depending on the degree of similarity between target and template sequences, structures predicted by homology modelling are generally within 3.5 Å, sometimes even within 1 Å backbone RMSD (Kopp and Schwede 2004). However, the applicability of template-based modelling (including both MODELLER and I-TASSER) is limited to those protein sequences that have reliable template structures. At present, the chance of finding a related template structure for a protein sequence chosen randomly from a genome varies roughly from 30% to 80% (Fiser 2010); the difference in the chance is due to the differences in the genome being considered (Fiser 2010). Because this approach builds protein models for the query sequence by copying the structure(s) from the template(s), an inherent drawback of it is that the models generated have a strong bias toward the template structure(s), rather than toward the native structure of the query protein (Read and Chavali 2007). Therefore, template-based modelling techniques should be refined to generate models closer to the native structures.

2.8.3 Template-free Modelling

When there are no structural analogues available in PDB, or they cannot be detected by threading modelling, one has to predict the structure of the target protein from scratch based on some basic physical principles. Thus, this type modelling is also called *ab initio* or *de novo* modelling, which is helpful to understand the physicochemical principle of how a protein adopts its specific fold in nature (Lee et al. 2017). The methods in this

category can also be divided into two types: pure physics-based free modelling and prior-knowledge-based free modelling.

In a pure physics-based *ab initio* method, interactions between atoms should be calculated according to classical mechanics or quantum mechanics. Since this method requires expensive computing resources, it has not been widely used to predict protein structures (Rigden 2009). A practical way to do the pure physics-based *ab initio* protein modelling is to use an empirical force field with selected atom types, which is applied in software such as AMBER (Salomon-Ferrer et al. 2013). There are two types of algorithms that are widely used in this protein structure modelling category, Monte Carlo simulation (MC) and molecular dynamics (MD). The former generally starts with building a global energy function to account for all conformational states of the target protein and then devises an efficient search strategy capable of quickly identifying low energy states (Figure 2.11). In detail, a random move on a fragment of the backbone or on the side chain is applied; an energy function is used to calculate the energy of the structure before and after the move; after comparing the energy change, the move is rejected or accepted with the Metropolis criterion. For MD simulation, the forces on each atom of a protein are solved by using a force field; the result is then used to calculate the position and the velocity of the atom at the next time point; the energy of the system at the current is calculated with a potential energy function. Due to the huge conformational search space even for an average protein, as compared with template-based modelling, this approach requires vast computational

resources and has thus achieved only very limited success. Some examples of pure physics-based modelling include (a) the 36-residue peptide of villin headpiece simulated with MD on a 256-CPU computer for two months, which finally reached an RMSD of 4.5Å to the native structure (Duan and Kollman 1998), (b) the mini protein of tc5b simulated to within 1 Å RMSD to the native NMR structure with a supercomputer (Chowdhury et al. 2003), (c) a 102-residue $\alpha+\beta$ protein from *T. maritima* topologically correctly predicted with a 7.3 Å C_α RMSD, and a phosphate transport system regulator PhoU which is a 235-residue mainly α -helical protein, also from *T. maritima*, announced by (Oldziej et al. 2005) that the authors “*predicted the topology of the whole six-helix bundle correctly within 8 Å RMSD*”. Despite these successes, “*physics-based folding is far from routine for general protein structure prediction of normal size proteins, mainly because of the prohibitive computing demand*” (Zhang and Wu 2009). In a recently published paper (Krupa et al. 2016), the authors introduced their MD simulation results of 55 proteins, which are the targets of CASP11. The average size of the proteins was 251 amino acids and the simulations were done with the coarse-grained UNited RESidue (UNRES) force field (Sieradzan et al. 2015), developed by the same group. Although the authors had completed the simulation of each protein in three weeks and the best RMSD of 3.8 Å was achieved for a 97-residue protein (T0769), over 4000 CPUs were used for the computation. This type of computing device is not accessible to an average laboratory.

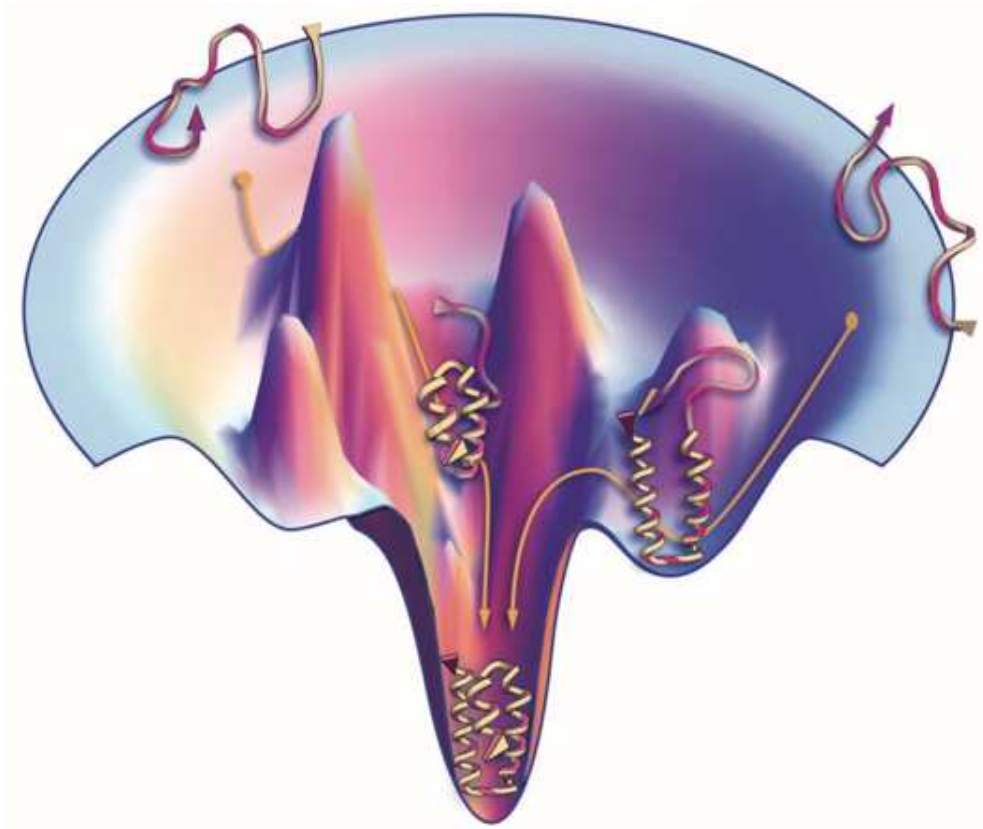


Figure 2.11. Diagram of a protein folding funnel. This figure shows how a protein folds into its native state through minimizing the free energy. This figure is reproduced from Dill and Chan ([Dill and MacCallum 2012](#))

The most successful free modelling techniques use prior knowledge ([Lee et al. 2017](#)). The prior knowledge can be small amino acid residue fragments with known structure, empirical energy terms derived from the solved structures deposited in PDB, or inter-residue contact constraints, etc. Probably the most successful method of this type is the AbinitioRelax program in the Rosetta protein modelling software suite, developed by David Baker and co-workers ([Leaver-Fay et al. 2011](#); [Rohl et al. 2004](#)). It makes use of an assembly strategy to combine fragment structures of unrelated proteins with similar local sequences to the query sequence by using Bayesian scoring functions ([Simons et al. 1997](#)).

A Bayes-based theorem gives the probability of a structure depending on its amino acid sequence. The 3D structures are generated by splicing together fragments and evaluating them using scoring functions.

Usually, Rosetta AbinitioRelax uses both three and nine continuous amino acid residues as the fragments. These structural fragments are obtained by searching all three- and nine-residue windows of the target protein sequence against the PDB database (exclusively X-ray structures with 2.5 Å or better) (Rohl et al. 2004). The initial search is carried out in a centroid mode and an optional subsequent model refinement is done in a full atom mode (Malmstrom 2005). In the centroid mode, the backbone heavy atoms of each residue (except glycine) of the target sequence remain, but the side-chain is simplified to a pseudo-atom (centroid) whose properties are determined by the identity of the residue (Rohl et al. 2004). The pseudo-atom is located at the side-chain centre of mass (for glycine, the C_α atom is chosen as the pseudo-atom) (Rohl et al. 2004). The energy function in this step includes solvation, electrostatics, hydrogen bonds between β strands, and steric clashes; all of these terms are knowledge-based (Kaufmann et al. 2010). After a large-scale conformation search with MC (described in the previous section), a set of selected low-resolution centroid mode models are fed into an all-atom mode refinement procedure by using a physics-based energy function that includes van der Waals interactions, hydrogen bonding, beta sheet pairing and pairwise solvation free energy. For the conformational search, multiple rounds of MC are employed again. Along with the predicted structures, Rosetta also outputs each individual energy score and the total energy score which is the

combination of all the individual energy scores for every structure. The structure with the lowest total energy score is likely to approach the native structure of the protein. Since the Rosetta AbinitioRelax was used for the protein structure prediction study in this thesis, a Rosetta *ab initio* modelling protocol can be found in Subsection 4.3.5 of Chapter 4.

The biological phenomenon of amino acid coevolution has caught the attention of researchers in the past 20 years. Coevolving residues are often found to be spatially proximal in the protein structure (Marks et al. 2012b). Recently, evolutionary analysis has made good progress in contact prediction by using global statistical models and machine learning algorithms (de Juan et al. 2013; Jones et al. 2015), which has led to the rapid development of protein structure prediction algorithms that use such predicted contacts as distance constraints. The archetypes include EvFold (Hopf et al. 2012; Marks et al. 2011), GDFuzz3D (Pietal et al. 2015), CONFOLD (Adhikari et al. 2015) and CoinFold (Wang et al. 2016a). Compared to the Rosetta *ab initio* program, these algorithms are able to not only sharply reduce the protein conformational search space and require much fewer computational resources, but also make good quality structure predictions (de Juan et al. 2013; Lee et al. 2017).

2.8.4 CASP

Critical Assessment of protein Structure Prediction (CASP) is a world-wide and double-blind competition to evaluate the state-of-the-art protein structure modelling methods. It has taken place biannually since 1994 (Moult 2005). CASP offers participating groups an opportunity to objectively assess their structure prediction methods, delivers information of what progress has been made in this field, and highlights where the future research may be most productively focused for both the research community and the public users.

The idea of CASP is to assign all the groups amino acid sequences (targets) whose structures have been solved experimentally, but have not yet been released to the public. All the groups are then challenged to predict 3D structures using their favourite algorithms. Subsequently, all the final models are sent back to the organizers and compared to the native structures by independent assessors. In the end, rankings (e.g., by group or best model) can be generated according to different quality measures.

CASP includes several categories: template-based modelling predictions, template-free modelling predictions, model refinement, model quality assessment (evaluating the accuracy of a model) and amino acid contact prediction (Moult et al. 2016a). In the CASP12 held in 2016, the most exciting result was the accurate template-free prediction of a large protein's 3D structure (256-residue) with the amino acid contact constraints predicted from GREMLIN (an implementation of plmDCA) (Monastyrskyy et al. 2016; Moult et al. 2016b). As a comparison, the predicted amino acid contact did not lead to improved

structure predictions until this round of CASP (Moult et al. 2016a). In addition, the model quality evaluation also marked an improvement in this round. The best single-model (assessing the quality of a model based on its properties) based method was shown to be as effective as clustering-based methods (assessing the quality of a model by comparing its similarity with other models of the same target protein) to pick out the best model among multiple candidates (Kryshtafovych et al. 2016).

In the CASP11 (2014), the most impressive improvement was made in the model refinement category (Moult et al. 2014), where a group used the MD simulation with the CHARMM36 force field and the TIP3 water model to successfully improve the structure predictions of all of the targets for the first time (Mirjalili et al. 2014).

2.8.5 Protein Structure Comparison

In this thesis, the protein structure predictions based on the algorithms proposed in this study need to be compared with the experimentally solved structures to evaluate how successful they are. Two of the structure similarity comparison methods commonly used today, root-mean-square deviation (RMSD) and template modelling score (TM-score) are introduced in this section. Which one is the better will also be discussed.

RMSD. RMSD is the most commonly used similarity measure between the structures of two proteins. RMSD values are presented in Å and calculated by

$$\text{RMSD}(v, w) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|v_i - w_i\|^2} \quad (2.38)$$

where v and w are sets of n corresponding atom positions in two superimposed proteins. Given the corresponding positions, a superposition with the minimal RMSD is found by translating and rotating the protein centroids onto each other. The computation of the optimal translation and rotation can be done with the Kabsch algorithm ([Kabsch 1976, 1978](#)).

The advantages of RMSD are that the value is straightforward and easy to understand, and the calculation process is relatively simple compared to the TM-score. An RMSD of $\approx 0\text{Å}$ means two structures are identical. However, its disadvantages should not be ignored.

Firstly, RMSD is sensitive to outliers. According to the formula of RMSD, a small number of atoms with high flexibility could lead to a large RMSD; e.g., a large RMSD between two protein structures may just be caused by a change at the position of a flexible loop. RMSD is also protein length dependent; one cannot judge the RMSD values of two pairs of proteins of different lengths ([Kufareva and Abagyan 2012](#)). One also cannot find an RMSD cut-off to judge if two structures compared have the same fold based on RMSD only. An RMSD of $\approx 0\text{Å}$ indicates the two structures have the same fold. If one (or both)

of them has flexible loops, and they have the same secondary structure composition and arrangement (which means they share the same fold), the RMSD between them might be a large value.

TM-score. TM-score tries to solve the above problems of RMSD. It was introduced along with the TM-align algorithm that can calculate TM-score efficiently ([Zhang and Skolnick 2005](#)). The formula of TM-score is

$$\text{TM-score} = \text{Max} \left[\frac{1}{L_{\text{Target}}} \sum_i^{L_{\text{ali}}} \frac{1}{1 + \left(\frac{d_i}{d_0(L_{\text{Target}})} \right)^2} \right] \quad (2.39)$$

where L_{Target} is the number of residues in the target protein; L_{ali} is the number of matched residues between the target and the query; d_i is the distance between the i^{th} matched residues pair. $d_0(L_{\text{Target}}) = 1.24 \sqrt[3]{L_{\text{Target}} - 15} - 1.8$ is a normalization term to eliminate the dependence of the obtained score on the target protein length. It is derived from the analysis of large sets of related and unrelated protein structures with different numbers of amino acids. The maximization is with respect to all superpositions of the target and template proteins.

It is very useful to know what different ranges of TM-score stand for. The TM-score value is in the range of $(0, 1]$. When $\text{TM-score} = 1$, it means the two structures are perfectly matched or they are identical; when $\text{TM-score} < 0.17$, it means they are just random

structures; roughly, when TM-score > 0.5 , the two structures likely have the same fold (Zhang and Skolnick 2004b, 2005).

2.9 Summary of This Chapter

This chapter introduces the backgrounds of the methods developed in this thesis. The main aim of this thesis is to present and evaluate the amino acid contact/distance prediction algorithm (DeepCDpred) and the *ab initio* structure prediction method which requires the predicted geometry constraints from DeepCDpred. DeepCDpred uses neural network models to make amino acid contact and distance constraint predictions for a query (target) protein sequence. It starts with searching the homologous sequences of the query sequence from a sequence database and building an MSA (Section 2.5). From the MSA, the features of the inputs of the neural networks are calculated. The features fed into DeepCDpred include coevolutionary couplings (amino acid coevolution is introduced in Section 2.2) calculated from the MI (Subsection 2.6.1) and the three global statistical models (mfDCA, PSICOV and plmDCA, Subsection 2.6.4), and secondary structure prediction (Subsection 2.7.1). Other features are mentioned in Subsection 2.7.2. The background of machine learning, specially neural networks, is introduced in Subsection 2.7.3. Since DeepCDpred needs to be compared with other algorithms, three recently published amino acid contact prediction algorithms (PconsC, PconsC2 and MetaPSICOV) are reviewed in Subsection 2.6.10. Among the three algorithms, MetaPSICOV is the most

accurate. Thus, the results of the comparisons between DeepCDpred and MetaPSICOV are displayed in Chapter 5 are the comparisons between DeepCDpred and MetaPSICOV.

The background of protein structure prediction introduced in Section 2.8 would help the readers understand the protein structure prediction method proposed in this study. Also, the introduction to RMSD and TM-score (Subsection 2.8.5) is necessary to explain why TM-score is preferred for protein structure comparisons in this study (Chapter 5).

CHAPTER 3

METHOD OVERVIEW AND MATERIALS

3.1 Overview of This Chapter

This chapter firstly introduces the aims of this thesis and the structures of the methods proposed in this thesis, which include DeepCDpred, the amino acid contact/distance prediction algorithm, DeepCDpred_AbInitio which employs the constraints predicted from DeepCDpred and a Rosetta *ab initio* modelling protocol to predict protein structures, and a protein model quality evaluation method, being used to achieve these aims. It is followed by the description of the materials required by these methods, which include data resources, software and computing resources. The final section of this chapter presents how the data and software are assembled to build the feature sets of the inputs of DeepCDpred and of the model confidence score prediction method.

3.2 Aims of This Thesis

Three aims were planned for this thesis:

1. to propose a more accurate algorithm for amino acid contact predictions which is also capable of predicting amino acid long-range distances;
2. to use a Rosetta *ab initio* modelling protocol and the amino acid contact/distance prediction obtained in aim 1 for protein structure prediction, which could test the effectiveness of the predicted constraints;
3. to propose a confidence prediction method which evaluates the quality of the predicted protein models.

3.3 Structures of The Methods Proposed In This Thesis

This section displays the structure of each method developed in this thesis and provides necessary explanations. Detailed information about how each method was developed can be found in the next chapter (Chapter 4).

3.3.1 Definitions of Amino Acid Contact and Amino Acid Coupled at A Distance

Before introducing the structures of the methods proposed in this thesis, it is necessary to present the definitions of amino acid contact and amino acid coupled at a long-range distance.

A variety of thresholds were used to define whether two residues are in contact. Here, the widely-used definition ([Kamisetty et al. 2013](#); [Pietal et al. 2015](#); [Wang and Xu 2013](#)) was adopted: two residues are considered to be in contact if the distance of their C_β (C_α for glycine) atoms is no greater than 8\AA in the experimental structure. MetaPSICOV also used this standard ([Jones et al. 2015](#)). Any residue pair with sequence separation less than 5 amino acids is removed from the training and the prediction in this work, since they are expected to be very close in a structure (e.g., residue pairs within an α helix) ([Jones et al. 2015](#); [Wang et al. 2017b](#)). This could also remove strong but trivial couplings between nearby residues, which might introduce bias to other residue pairs in the training and prediction process otherwise.

Besides the contact bin of $(0-8\text{\AA}]$, three other distance bins are taken into consideration, i.e. $(8-13\text{\AA}]$, $(13-18\text{\AA}]$, $(18-23\text{\AA}]$, also measured by the $C_\beta-C_\beta$ (C_α for glycine) distance. Predicting the three distance bins provides more constraints for protein structure prediction, and the choice of bin size of 5\AA for each bin is based on the consideration that a

larger bin size could not provide precise constraints for structure prediction, and a smaller bin size may increase the difficulty of classification due to the limited data in the bin.

3.3.2 The Structure of DeepCDpred

The overall structure of the amino acid contact/distance prediction algorithm, DeepCDpred, is shown in Figure 3.1. The pipeline of DeepCDpred is summarized in Figure 3.1a, and the explanations are as follows.

At the beginning, homologous protein sequences of the query (target) sequence were detected and an MSA was built. These were done by using HHblits. Direct couplings were then inferred from three global statistical models, mfDCA (from the FreeContact software), QUIC and plmDCA (from the CCMPred software), based on the MSA. Other features such as amino acid profiles, statistical potential, mutual information, the number of sequences and the number of effective sequences were also calculated based on the alignment. Meanwhile, the secondary structure and the solvent accessibility were predicted by SPIDER2. All of these features, together with the protein length (the number of amino acids), were fed into four groups of neural network models to output an inter-residue contact score and three other distance scores for each residue pair of the query sequence. DeepCDpred consists of four groups of neural network models (Figure 3.1b). Each group is responsible for the prediction of inter-residue contacts in a specific distance bin.

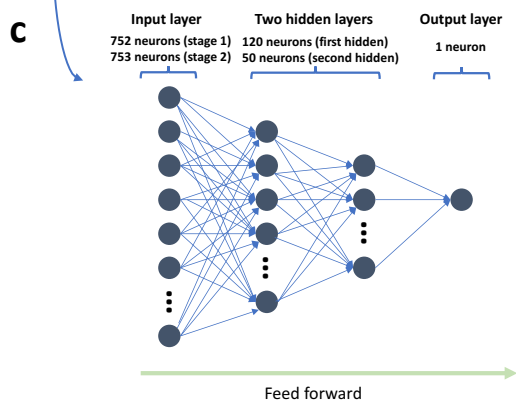
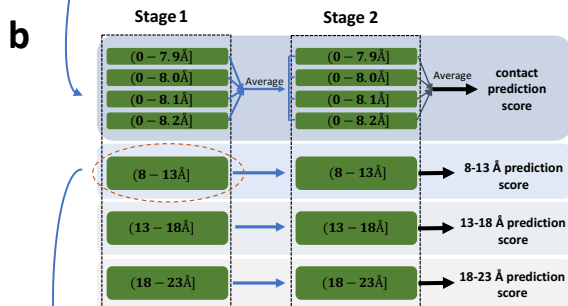
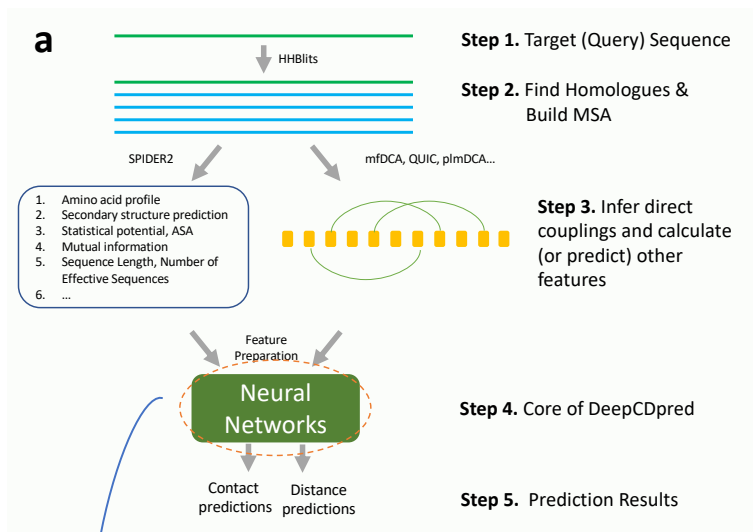


Figure 3.1. The overall structure of DeepCDpred.

(a). The overview structure of DeepCDpred. (b). The neural networks for contact and distance prediction. Each green rectangle represents a neural network; the contact prediction model is the average of the outputs from four networks and each was trained with a different contact cut-off (i.e. 0-8Å, 0-7.9Å, 0-8.1Å and 0-8.2Å); for distance prediction, only one network was used; for both contact and distance prediction, the strategy of two-stage was employed in DeepCDpred; the contact/distance prediction from stage 1 was used as an extra feature fed into the network of stage 2; the final contact/distance prediction score for each residue pair was taken from the output of the network of stage 2. (c). The architecture of the networks in (b), which includes the structures of both stage 1 and stage 2.

In each group or each prediction task, a two-stage structure was used. This means the prediction results (contact or distance prediction scores) from stage 1 were used as a feature in the model of stage 2; the output scores from stage 2 were the final contact/distance

predictions (Figure 3.1b).

In the contact prediction group, for both stage 1 and stage 2, there are four neural network models. In addition to the distance range of $0 - 8\text{\AA}$, three other similar ranges were also chosen, i.e. $0 - 7.9\text{\AA}$, $0 - 8.1\text{\AA}$ and $0 - 8.2\text{\AA}$, for amino acid contact prediction. Each of the distance ranges was used as the classification target to train a neural network model (e.g., for the range of $0 - 8\text{\AA}$, the target was set to 1 if the distance of a residue pair in the training set is in this range and 0 if is out of this range when training the neural network). In other words, the four networks in stage 1 of contact prediction used the same feature inputs, but different classification targets. The contact prediction scores from the four networks were averaged and used as features in the inputs of the networks in stage 2. Again, the four networks in stage 2 of contact prediction used the same feature inputs, but different targets (also $0 - 7.9\text{\AA}$, $0 - 8.0\text{\AA}$, $0 - 8.1\text{\AA}$ and $0 - 8.2\text{\AA}$). The final neural network output from this group was the average of the outputs from the four networks. This strategy of combining different contact thresholds followed its successful implementation in MetaPSICOV (Jones et al. 2015). In Figure 5.8 of the Results chapter (Chapter 5), for both stage 1 and stage 2, a comparison between the contact prediction accuracy from each model in the four networks and the result calculated by averaging the outputs from the four networks is shown. For each of the other three groups of inter-residue distance prediction, only one neural network model was trained in each stage; the classification target was defined by the corresponding distance bin range. The combination of multiple neural network models were also tried, but it did not show significant improvement of

distance prediction (data not shown in this thesis).

Figure 3.1c shows the architecture of the neural networks used in the four groups. There are two hidden layers: the first has 120 neurons and the second has 50 neurons. The output layer has only one neuron to report the prediction score (between 0 and 1).

3.3.3 The Structure of DeepCDpred_AbInitio

Figure 3.2 shows the overview of DeepCDpred_AbInitio, which includes DeepCDpred, the protein structure prediction step that is based on the predicted constraints from DeepCDpred, the secondary structure prediction from SPIDER2, and a Rosetta *ab initio* protocol.

The first part of DeepCDpred_AbInitio is the already-mentioned DeepCDpred. From DeepCDpred, not only the final predictions of contacts and distances for each residue pair in the query (target) sequence, but also the intermediate result of the secondary structure prediction for the query (target) sequence, were exported. All of them were used in Rosetta *ab initio* structure modelling. As will be introduced in Section 4.3 (the next chapter), two methods to construct Rosetta constraints were used. Meanwhile, the three-mer and nine-mer fragments were generated for the query (target) protein by using the Perl script of ‘make_fragments.pl’, which can be found in the Rosetta suite. Two groups of structure predictions were made based on constraints from two different methods, respectively. In

each group, the model with the lowest Rosetta energy score was selected; and the TM-score was predicted based on the TM-score prediction neural network model (the model confidence evaluation method that will be introduced later). The quality of the predicted best structures between the two groups is compared in the Results chapter (Chapter 5).

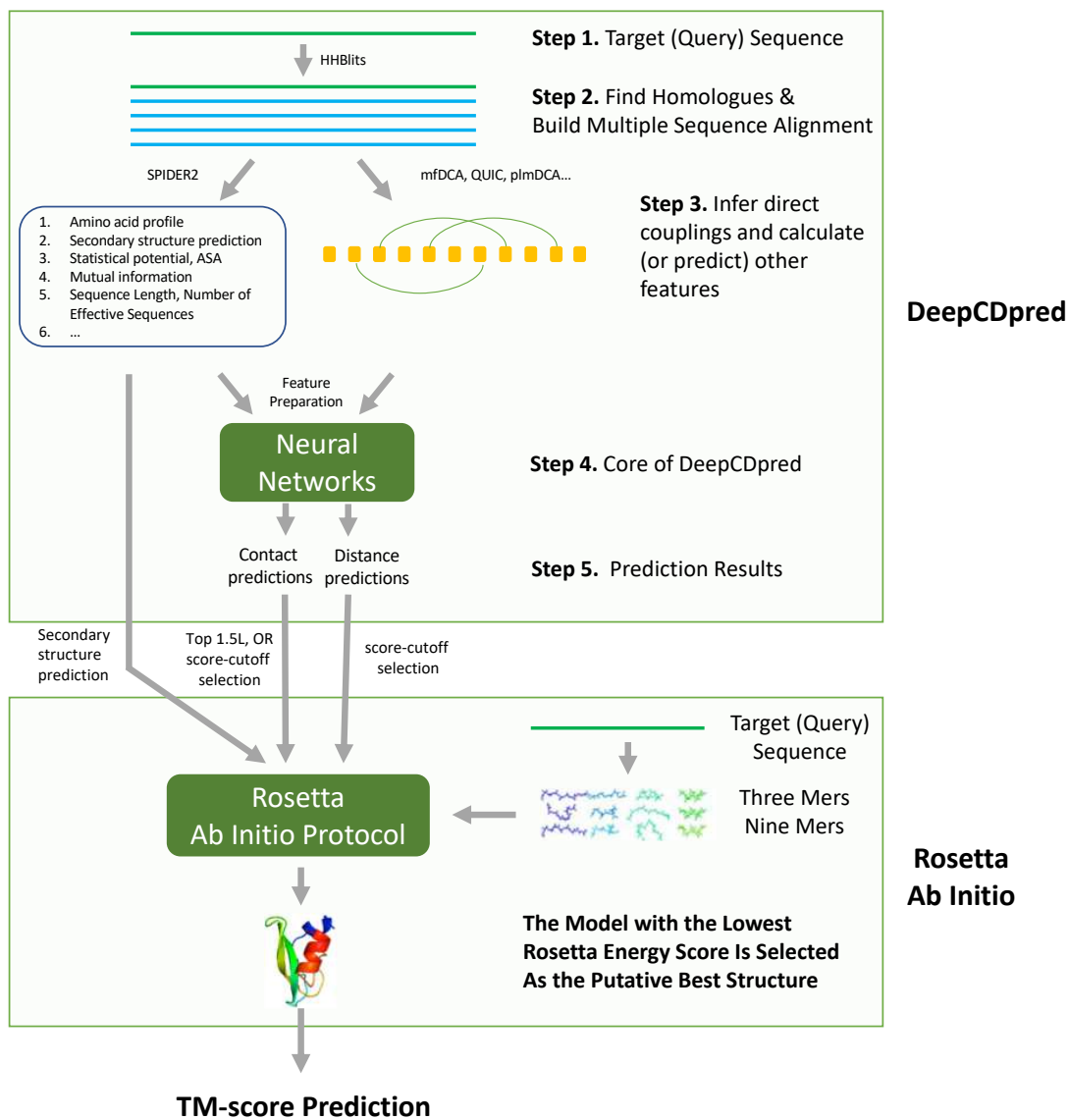


Figure 3.2. Overall diagram of DeepCDpred_AbInitio pipeline, including the step of DeepCDpred for inter-residue contact and distance predictions, and the step of structure prediction by using the obtained geometry constraints from the former.

3.3.4 The Structure of The Confidence Prediction Model

A confidence value should be assigned to each structure prediction to indicate its expected similarity to the native structure. It might be useful for a user to select or filter out predicted structures.

Since the TM-score of a model with respect to the crystal structure is a good measure of the quality of the model, as described earlier, the quality prediction model was trained to predict this TM-score (subsection 2.8.5). The model is a classic three-layer feedforward neural network as shown in Figure 3.3. The input layer has 7 dimensions; the hidden layer contains 5 neurons and the output layer only has one neuron to report the predicted TM-score.

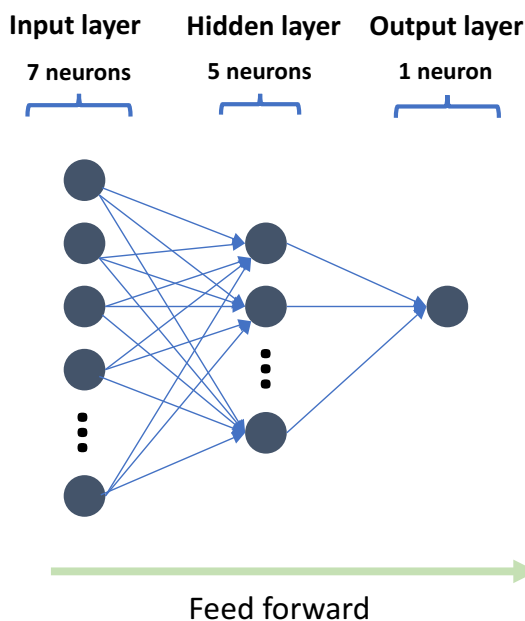


Figure 3.3. The architecture of the three-layer neural network model chosen for predicting TM-score.

3.4 Materials

3.4.1 Data

A. Homologous Sequence Search Database

The homologous sequence search data source was downloaded from http://wwwuser.gwdg.de/%7Ecompbiol/data/hhsuite/databases/hhsuite_dbs/ (last check: November 2018). The file name is “uniprot20_2016_02”. It was built by the HHblits group (i.e. Söding and his coworkers). Both DeepCDpred and MetaPSICOV (the algorithm DeepCDpred is compared with in this thesis) used HHblits to search for homologous sequences from this data source and build an MSA for each target sequence. The settings of HHblits are introduced in the Software section (the next section).

A metagenomics sequence dataset was downloaded from https://metaclust.mmseqs.com/current_release/metaclust_2017_05.fasta.gz (last check: November 2018); it was concatenated with uniprot100 (downloaded from <ftp://ftp.uniprot.org/pub/databases/uniprot/uniref/uniref100/uniref100.fasta.gz>, March 2017) and used as an extra protein sequence source for homology detection. It is worth noting that uniprot sequences do not come from metagenomics projects (http://www.uniprot.org/help/sequence_origin (last check: November 2018)), and thus the concatenation does not generate sequence redundancies. This dataset

contains about 600,000,000 sequences. As a comparison, uniprot100 itself only had about 100,000,000 sequence entries by the March of 2017. Since uniprot20 is a subset of uniprot100 and “uniprot20_2016_02” was built in 2016, it must have fewer sequences than uniprot100. This concatenated larger data set does not support an HHblits search, but supports an HMMER search (HMMER will be introduced in the Software section); the latter is much slower than the former. Thus, this dataset was only used to explore the possibilities of improving the performance of DeepCDpred (visit Section 5.11 (Chapter 5) for the result).

B. Test Set, Training and Validation Set of DeepCDpred

The test dataset was created based on MetaPSICOV’s test set, since the latter is a set of diverse proteins (Jones *et al.* 2015) and it is convenient to compare the amino acid contact prediction performance between DeepCDpred and MetaPSICOV based on the same data. Originally, there were 150 protein chains. A filtering process was made on them by removing any chain similar to sequence(s) (sequence identity no less than 25%) in the training set of SPIDER2. After this step, 108 protein chains ranging from 52 to 266 amino acids remained and were used as the test set of this work. According to the study done by Jones *et al.* (Jones *et al.* 2012), these protein chains are X-ray crystallographic structures and with the resolution $< 1.9\text{\AA}$. Every pair of proteins in this test set has 25% or less sequence identity to each other. Since the algorithm developed in this thesis is called DeepCDpred, these 108 chains are named the test set of DeepCDpred (Figure 3.4). The

SPIDER2 training set was downloaded from http://sparks-lab.org/server/SPIDER2/dat/seq+ss_train.txt (last check: November 2018). It consists of 4,590 protein chains.

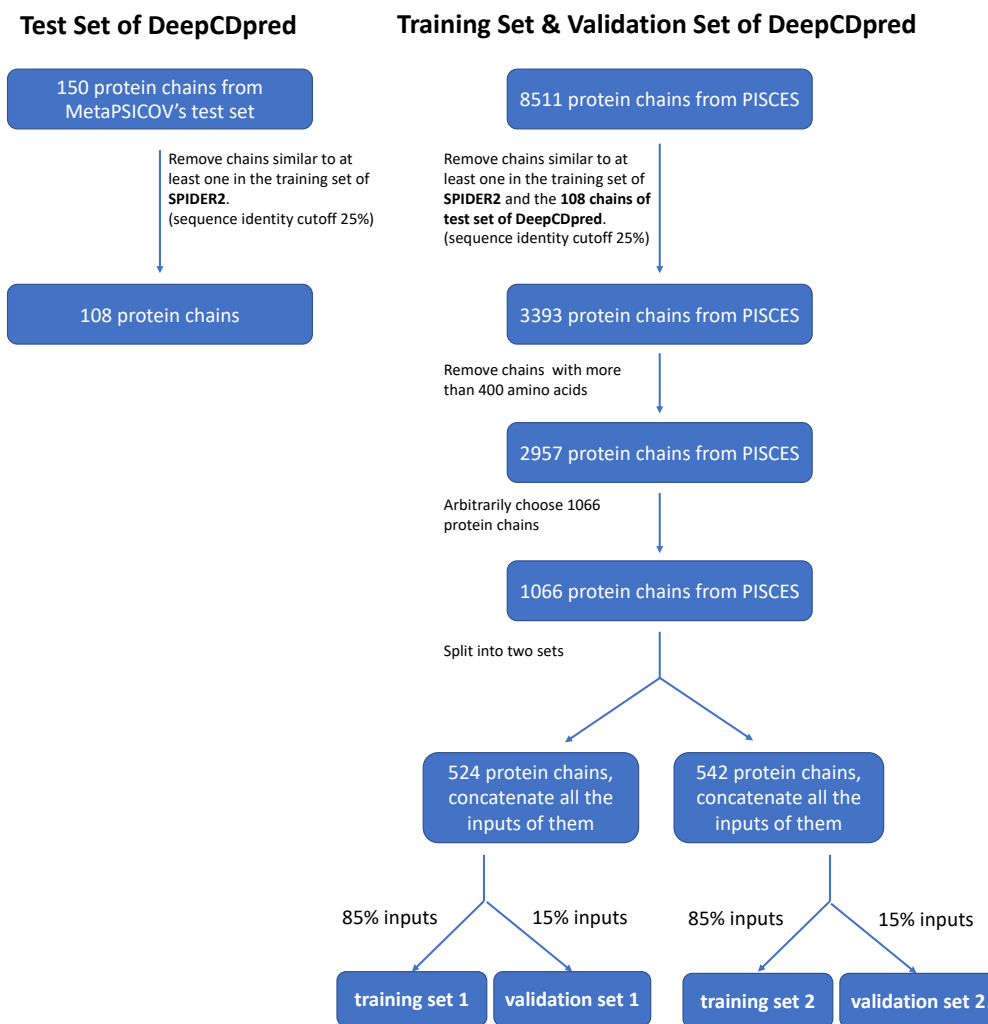


Figure 3.4. Diagram of selecting the test set, training set and validation set of DeepCDpred.

The pdb id list of these 108 protein chains can be found in Table C.3 of Appendix C.

For building of both the training set and the validation set of DeepCDpred, the protein chains from the PISCES set (November 2016) (Wang and Dunbrack 2003) were used.

PISCES is a subset of the sequences culled from the entire PDB according to the experimental type (e.g. X-ray crystallography, NMR), the structure quality and the maximum mutual sequence identity. In PISCES, the chains were determined by X-ray crystallography, with no less than 2Å resolution and with no more than 25% pairwise sequence identity. There were 8,511 protein chains in the PISCES dataset. After removing sequences similar (pairwise sequence identity > 25%) to the ones in both the test set of 108 chains and the training set of SPIDER2, there were 3393 chains in the remaining PISCES. In order to include more proteins in the training process, chains that have more than 400 amino acids were removed before going to the next step (a large protein generates much more feature inputs to the training process of DeepCDpred than a small protein, which can be easily understood in Subsection 3.5.1). Then 1,066 chains from the remaining 2,957 chains were arbitrarily chosen as the data resource of the training set and validation set. Their chain lengths range from 16 to 394 amino acids. The pairwise sequence identity between the test set and the training set (including validation set) is $12.4\% \pm 3.6\%$ (Figure B.1 in Appendix B). Due to the memory limit of the computer used in this work, the 1,066 chains had to be split into two sets with set one, 524 chains, and set two, 542 chains, in the training process. For each set, all of the neural network inputs of the chains were concatenated (7,215,900 inputs in set one and 4,435,101 in set two; there were more inputs in set one, since more large chains are in this set); each input has 752-dimensional features in the training process of stage 1 and 753-dimensional features in stage 2. How the features were defined for each stage can be found in Section 4.2 of Appendix 4. For

both the two sets, 85% of the inputs were randomly chosen as the training set and the rest 15% were used as the validation set; the latter was used to prevent overfitting in the training process via early stopping. Dividing the training set and validation set based on all of the inputs rather than dividing the proteins (e.g. 85% of the proteins used as the training set and 15% as the validation set) was for the diversity consideration – the 85% inputs in the training set may come from all of the 1,066 proteins and the 15% inputs in the validation set are also possible from all of the 1,066 proteins. A diagram summarizing the construction of the test set, the training set and the validation set can be found in [Figure 3.4](#).

The pdb id list of these 1066 protein chains can be found in [Table C.4](#) of [Appendix C](#).

As mentioned in the above two paragraphs, when the test set, the training set and the validation set were prepared, sequences similar to any sequences in the SPIDER2 training set were removed from these three sets. The secondary structure prediction from SPIDER2 was used as a feature in DeepCDpred. The secondary structure definition in the training set of SPIDER2 is from experimental structures. Thus, if similar sequences in the test set, the training set and the validation set of DeepCDpred were not removed, they would introduce a bias.

It should be stated clearly how the similar sequence detection method was used to construct the test set, training set and validation set works. As mentioned above, there were three places that use similar sequence detection: removing sequences from MetaPSICOV's

test set that are similar to those in the training set of SPIDER2 to construct the test set of DeepCDpred; removing sequences from PISCES that are similar to those in the test set of DeepCDpred; and removing sequences from the remaining PISCES sequences that are similar to those in the training set of SPIDER2. Since the methods used in these three places are the same, only the first is introduced here. Firstly, the SPIDER2 training set sequences were converted into a BLAST searchable database with the program makeblastdb from the BLAST suite ([BLAST 2018](#)); then each sequence in MetaPSICOV's test set (150 chains) was searched against this database using Blastpgp (also from the BLAST suite) with the default options (including the E-value, which was 10.0) except option '-m' (alignment view option) which was set to 8 (means the tabular output). Significant hits found were output to a plain text format file. The sequence identity between the query sequence and the hit from the SPIDER2 training sequences could be found in the third column of the output file. For any query sequence, if at least one hit was found with sequence identity >25%, this query was removed from the test set. After this filtering step, 108 protein chains remained and formed the test set of DeepCDpred.

As a comparison, MetaPSICOV used a training set of 624 protein chains; the resolutions of these chain structures are less than 1.5Å and their lengths range from 50 to 500 amino acids ([Jones et al. 2015](#)).

C. Test Set, Training and Validation Set of the Model Confidence Estimation Method

The test set for the TM-score prediction method had the same 108 protein chains as the test set of DeepCDpred. 161 protein chains randomly chosen from the training/validation set of DeepCDpred were used as the training/validation set of this method. Similar to DeepCDpred, the 161 inputs were concatenated; 85% of the inputs were chosen randomly as the training set and the other 15% as the validation set, for the purpose of early stopping, to prevent overfitting.

The pdb id list of these 161 protein chains can be found in Table C.7 of Appendix C.

3.4.2 Software

A. Programs for Homologous Sequence Searching and MSA Construction

HHblits. HHSuite (version 2.0.16) was downloaded from Github (<https://github.com/soedinglab/hh-suite> (last check: November 2018)). HHblits requires a program-specific protein sequence database of HMM profiles. Here, the version released in February 2016 (file name: uniprot20_2016_02) was used. The parameter settings used in HHblits were iteration: 4 and e-value: 0.001 (these are the default values, and were also used in MetaPSICOV (Jones et al. 2015)), minimum coverage with the query sequence: 60% (this value is not the default, but was used by MetaPSICOV (Jones et al. 2015)), and maximum pairwise sequence identity: 90% (this value was chosen based on some tests of contact prediction with the proteins from the training/validation set by using CCMpred). The

values chosen for ‘iteration’ and ‘maximum pairwise sequence identity’ in MetaPSICOV were 3 and 99%, respectively.

HMMER. HMMER 3.1b1 was downloaded. The parameter setting for e-value was 1.0; other parameters were chosen as the default values.

B. Programs for Amino Acid Coevolutionary Coupling Inference

The following four programs, MI_APC, FreeContact, QUIC and CCMpred, were used to calculate pairwise coevolutionary couplings for each pair of residues in the target sequence, which were used as features in the feature set of DeepCDpred. They all take the MSA as inputs.

MI_APC. Mutual Information with the Average Product Correlation (MI_APC), as described by Dunn *et al.* (Dunn *et al.* 2008), was calculated by using a script in the MetaPSICOV source code, which was downloaded from <http://bioinfadmin.cs.ucl.ac.uk/downloads/MetaPSICOV/> (last check: November 2018).

FreeContact. mfDCA couplings were calculated by using the implementation of package FreeContact (downloaded from <ftp://roslab.org/free/freecontact-1.0.21.tar.xz> (last check: November 2018)) with default parameters.

QUIC. The source code of QUIC was downloaded from http://www.cs.utexas.edu/~sustik/QUIC/QUIC_MEX_1.1.tar (last check: November 2018). It consisted of a mixture of MATLAB and C scripts. In this study, the code was rewritten solely in C. To speed up

the calculation, OpenMP was also employed to allow it to run in parallel. The parameter of tolerance which controls the speed of convergence was chosen as 0.004. Several proteins from the training/validation set of DeepCDpred were arbitrarily chosen to test both the original QUIC and the rewritten version, in order to ensure that the calculation results from them were the same. The tolerance value is a trade-off between calculation speed and contact prediction accuracy and was optimized with an arbitrarily chosen subset of 221 proteins from the 1066 protein chains (for more information about the 221 proteins, please refer to Section 5.4 in the Results chapter and the Table C.2 in Appendix C).

CCMpred. CCMpred was downloaded from <https://github.com/soedinglab/CCMpred> (last check: November 2018) and run with default parameters.

B. Programs for the Secondary Structure Prediction

The following programs of Blastpgp and SPIDER2 were used to predict the secondary structure and the accessible solvent area of the target sequence. Specifically, Blastpgp was used to search homologues for the target sequence and build a PSSM; SPIDER2 was then used to predict the secondary structure and the accessible solvent area based on that PSSM.

Blastpgp. Blastpgp of version 2.2.26 was used in this study and uniref90 (November 2016) was used as the sequence database.

SPIDER2. The source code of SPIDER2 and the training dataset of SPIDER2 were downloaded from <http://sparks-lab.org/server/SPIDER2/> (last check: November

2018). The protein secondary structure and the accessible solvent area predictions for the target sequence were generated by using default settings.

B. Program for *Ab Initio* Protein Structure Modelling

Rosetta. The source code of version 3.7 was downloaded from <https://rosettacommons.org> (last check: November 2018) and compiled into executable files supported by the openMPI library to allow parallel calculations on multiple CPU cores.

3.4.3 Computing Resources

The neural network models in this study were trained on a desktop machine with 128GB RAM and two Intel *Xeon* E5-2630 v3 2.4 GHz processors (16 cores in total). Structure predictions were carried out on BlueBEAR, a supercomputer in the University of Birmingham, and for each protein 50 CPU cores were used.

3.5 Features and Feature Vector

This section firstly explains in detail how the features in the neural networks of DeepCD-pred and the network of the model confidence estimation method were calculated. Next, it explains how the features were combined to form the feature vector (or the feature set).

3.5.1 Features

Features of DeepCDpred

Features are the basis of many machine learning algorithms; feature selection is thus a crucial step. The features of DeepCDpred were inspired by the earlier studies of MetaP-SICOV (Jones et al. 2015) and SVMcon (Cheng and Baldi 2007). Table 3.1 lists all of the 13 types of features used in stage 1 networks of DeepCDpred. For stage 2 networks, besides the features used in stage 1, the output from stage 1 was also used as a feature. In the table, the third column also explains how each type of feature was calculated. “Based on MSA” means the corresponding feature was calculated based on the MSA by using the script developed in this study. The amino acid profile is the amino acid frequency distribution at each position of the target sequence. It was easily calculated based on the MSA of the target sequence. The number of sequences in the MSA was calculated directly based on the MSA. The number of effective sequences in the MSA was calculated according to the formula in subsection 2.6.6 (sequence reweighting); the sequence identity cut-off was chosen as 0.8, which means if the identity between two sequences in the MSA is greater than 80%, they are regarded as the same sequence. With the amino acid profile, the position entropy of each position could be calculated according to the Shannon entropy formula $S = -\sum_a f(a) \log(f(a))$, where a is any of the 21 amino acid types (gap is considered as the 21st amino acid). The statistical contact potential between a pair of residues from the target sequence is calculated by averaging contact potentials of

all the pair of residues in the two columns of the MSA. The contact potential matrix for the 20 standard amino acids used in this study came from the work of Betancourt and Thirumalai (Betancourt and Thirumalai 1999). As for other features, each was calculated with the programs listed in the table.

Table 3.1. Features of DeepCDpred.

#	Feature Name	How Calculated?
1	Chain Length	No. of aa in the target sequence [#]
2	Amino Acid Profile	Based on MSA
3	No. of Sequences in MSA	Based on MSA
4	No. of Effective Sequences in MSA	Based on MSA
5	Sequence Separation	Distance of the residue pair on sequence
6	Contact Potential	Based on MSA ^{##}
7	Position Entropy	Based on MSA
8	MI	MI_APC
9	Secondary Structure Prediction	SPIDER2
10	Accessible Solvent Area	SPIDER2
11	EvFold Coevolution Coupling	FreeContact
12	QUIC Coevolution Coupling	QUIC
13	plmDCA Coevolution Coupling	CCMpred

[#]: aa, amino acid.

^{##}: contact potential matrix was adopted from the study of Betancourt and Thirumalai (Betancourt and Thirumalai 1999).

As described above, the DeepCDpred feature vector requires multiple features to be abstracted from the MSA prior to training or prediction, i.e. pre-processing of the MSA. Moreover, some of this pre-processing is undertaken by using neural networks, e.g. SPIDER2 to predict the solvent accessibility and secondary structure. However, the ultra

deep convolutional networks are likely to be capable of learning abstracted features directly from the input data. With networks like ResNet (He et al. 2016) and SENet (Hu et al. 2017), the greater the depth of the network, the greater the capacity to learn abstracted features. Classic feed forward networks can also learn this in principle, but facing the problem of vanishing gradient when getting deeper (Nair and Hinton 2010). It should be possible to go directly from MSA to contact prediction without the need for pre-processing the MSA. Skipping the preprocessing of the MSA would speed up training, testing and predicting. More discussions about end-to-end learning can be found in Section 6.6 and Section 6.9 of Chapter 6.

Features of the Model Confidence Estimation Method

Table 3.2 lists all of the types of features used in this neural network method. For each generated structure, the Rosetta energy score was assigned by the AbinitioRelax program in the Rosetta suite.

These features were chosen for the following reasons: the chain length indicates the size of the protein; the number of sequences and the number of the effective sequences define the quality of the MSA of the query sequence, which could affect the amino acid contact prediction accuracy (see Subsection 5.5.5), and thereafter the quality of the structure; the accuracy of the secondary structure prediction also affects the accuracy of the amino acid contact prediction (see Subsection 5.5.5); the Rosetta energy score is Rosetta's own measure of structure quality; a lower score would likely indicate a better model.

Table 3.2. Features of the model confidence estimation method.

#	Feature Name	How Calculated?
1	Chain Length	No. of aa in the target sequence
2	No. of Sequences in MSA	Based on MSA
3	No. of Effective Sequences in MSA	Based on MSA
4	Secondary Structure Prediction	SPIDER2
5	Rosetta Energy Score	Rosetta AbinitioRelax

3.5.2 The Feature Vectors

The Feature Vector of DeepCDpred

In stage 1 of each neural network of DeepCDpred, for each pair of residue positions, a feature vector with 752 dimensions was used as an input. The position pair (any pair in the target sequence) being considered here is denoted by (i, j) . Some features in the neural network model of this stage were based on the properties of the amino acids adjacent in the sequence to the one of immediate interest. Two windows of length 9 amino acids are centred at i and j respectively, and another window of length 5 is located at the middle point $(i + j)/2$. For each of the columns $(2 \times 9 + 5)$ in the three windows, the amino acid composition consisting of the relative frequencies of the 20 amino acids and a 21st position for the gap were calculated; three values of the probabilities of helix, strand and coil and one value of the predicted solvent accessibility (solvent accessible surface area) were imported from the predictions of SPIDER2 (Heffernan et al. 2015). The Shannon entropy for this position was also included. The last binary value (position tag) indicated

whether this position is outside the length of the query sequence, since for the four N-terminal positions and four C-terminal positions, the window centred at any of these is not intact. Across the three windows, there were thus 621 local features, $(2 \times 9 + 5) \times (21 + 3 + 1 + 1 + 1) = 621$. Other features were either only related to positions i or j , or global properties of the MSA. Statistical contact potential and mutual information with the average product correction were included ($1 + 1 = 2$ elements in the feature vector). The covariance scores from FreeContact (mfDCA), QUIC ([Hsieh et al. 2014](#)) and CCMpred (plmDCA) were also among the features. For the former two, only the values located at (i, j) were used; while for the couplings predicted by CCMpred, a square window of size 9×9 centred at the position pair (i, j) was used and all contact scores predicted by this software in this window were included in the feature vector. The length of the protein chain, the number of sequences and the effective number of sequences in the MSA, the mean values of all the alpha helix, beta strand, coil, solvent accessibility scores and position entropies were used as well (another 8 elements in the feature vector). The other elements in the feature vector were used to encode the sequence separation between the two positions (Figure 3.5).

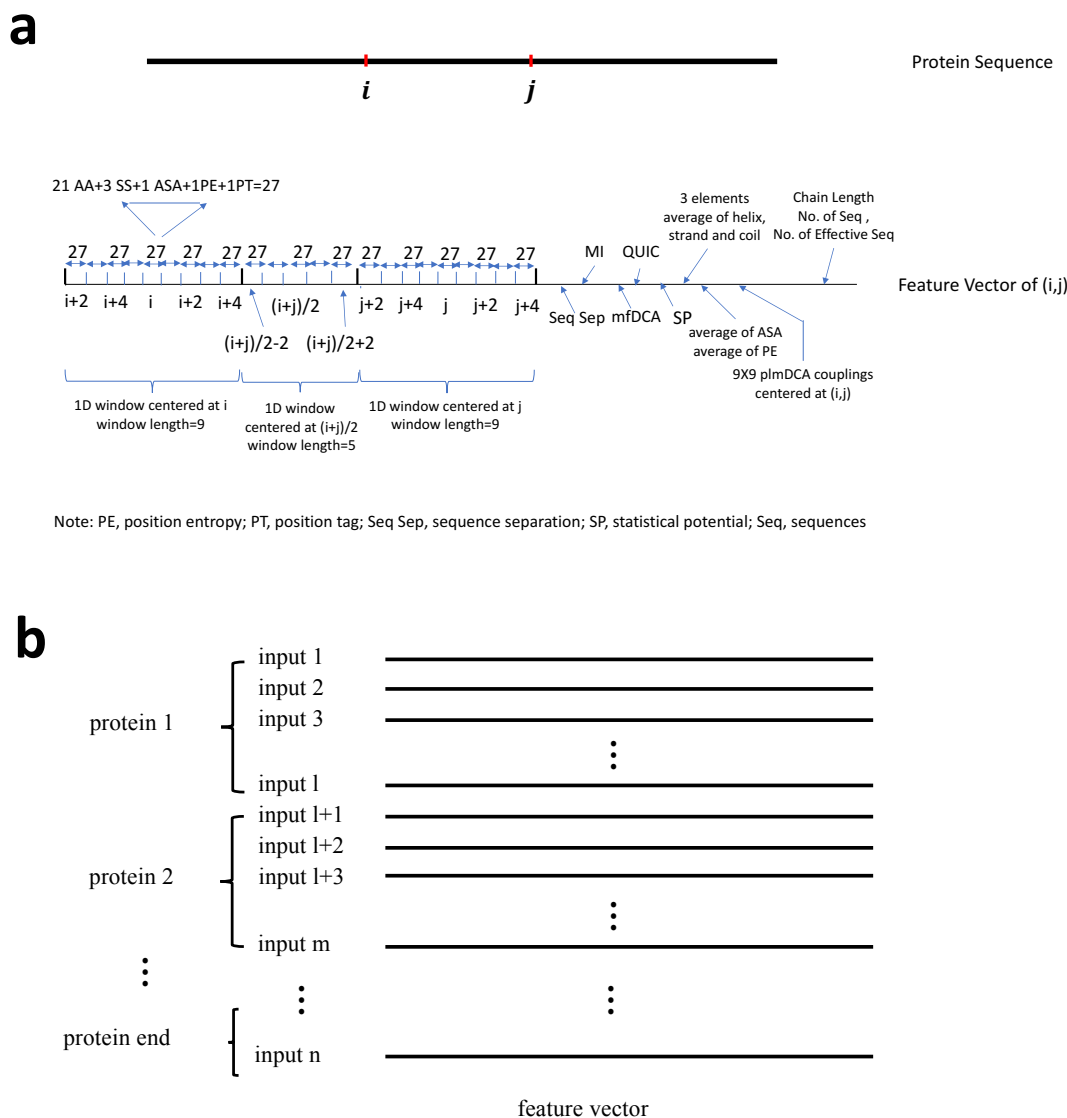


Figure 3.5. Diagram of the feature vector for each residue pair of stage 1 networks of DeepCDpred (graph a) and the concatenation of feature vectors of all the residue pairs (inputs) of all the proteins in the training/validation set to form the training/validation data (graph b).

In stage 2, the feature vector dimension was changed to 753. The predicted residue contact scores from stage 1 were used in this stage. The square window centred at position pair

(i, j) contained the 9×9 prediction scores from stage 1, rather than the scores from CCMpred. The coupling score of CCMpred between positions i and j was also included, which led to one more element in the feature vector as compared with that in stage 1.

The Feature Vector of the Model Confidence Estimation Method

85% of the 161 protein chains were randomly chosen as the training set and the other 15% as the validation set to prevent overfitting.

Each input has 7 dimensions – chain length, the number of sequences in the MSA, the number of effective sequences in the MSA, Rosetta energy score of the target model, and the other three elements that are the mean of the helix probability predicted by SPIDER2 for all positions in the alignment, the mean beta strand probability, and the mean coil probability (since the model was predicted with the constraints from DeepCDpred, the secondary structure prediction was the same as the one used in the feature of DeepCDpred). Then, the inputs were constructed by concatenating the feature vectors of the 161 training/validation protein chains into one matrix (161×7).

CHAPTER 4

METHOD DEVELOPMENT

4.1 Overview of This Chapter

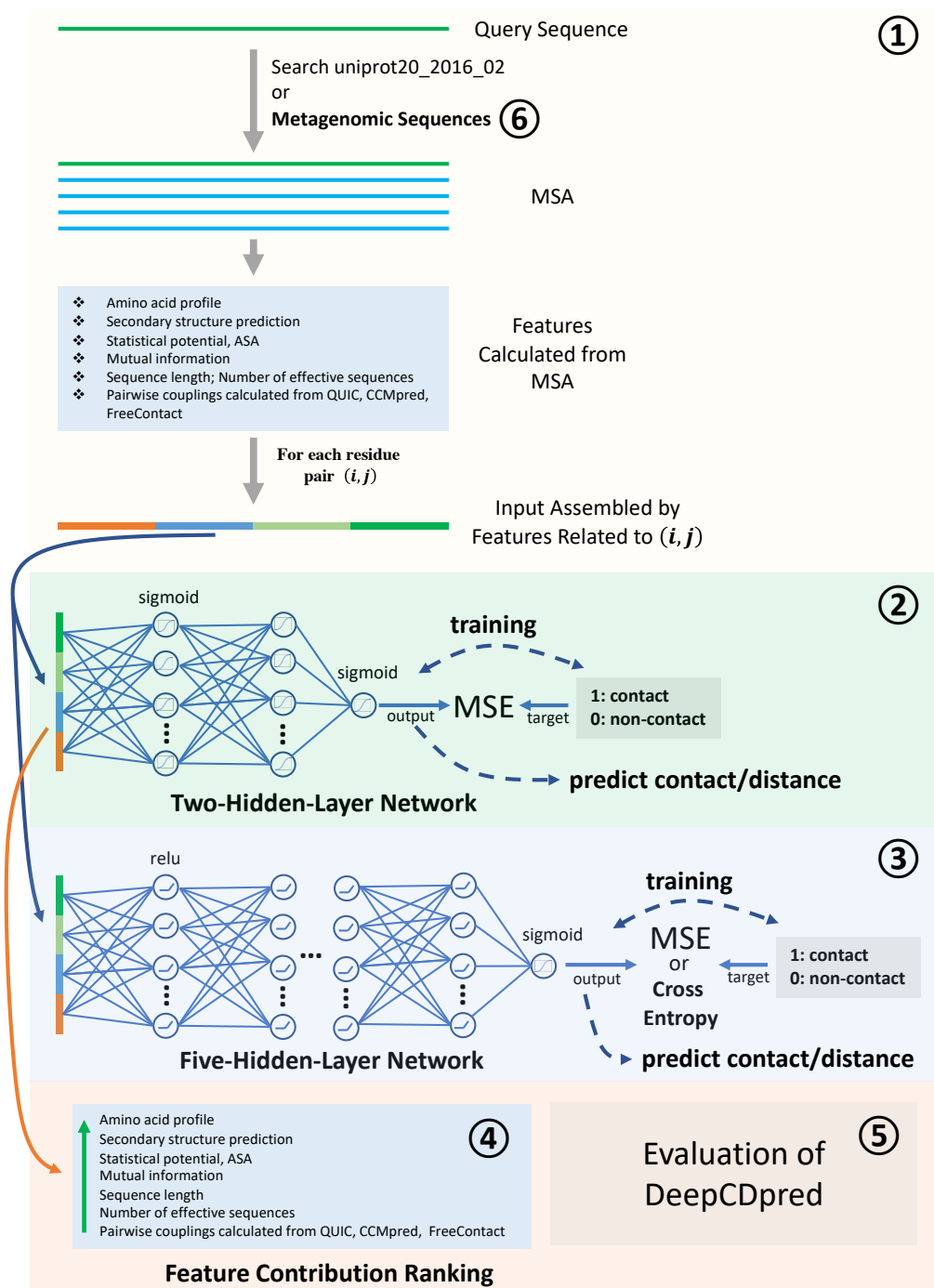
This chapter introduces the development of the three methods proposed in this thesis (i.e. DeepCDpred, DeepCDpred_AbInitio, and the protein model quality estimation method).

4.2 The Development of DeepCDpred

4.2.1 Introduction

An overview of the structure and feature vectors of DeepCDpred was introduced in the previous chapter. This section introduces the implementation of DeepCDpred (Figure 4.1),

which includes the technical details of the network models in DeepCDpred (e.g. the activation function and the loss function) (Subsection 4.2.2), how these networks were trained and optimized (Subsection 4.2.3), how the performance of DeepCDpred was evaluated (Subsection 4.2.5), and the realization of DeepCDpred (Subsection 4.2.6, this part is not mentioned in the diagram figure below). In addition, the explorations of improving the performance of DeepCDpred by using metagenomic sequences and a deeper architecture (Subsection 4.2.4), the feature contribution analysis that ranks the features of DeepCDpred based on their significance to the contact prediction accuracy (Subsection 4.2.7), and the difference between DeepCDpred and MetaPSICOV are also introduced (Subsection 4.2.8, this part is not mentioned in the diagram figure)).



- ①: introduced in Chapter 3
 ②: subsections 4.2.2 and 4.2.3
 ③: subsection 4.2.4

- ④: subsection 4.2.7
 ⑤: subsection 4.2.5
 ⑥: subsection 4.2.4

Figure 4.1. Diagram of the development of DeepCDpred. The numbers indicate which subsection each part will be introduced in.

4.2.2 Technical Details of DeepCDpred

The features as described in Subsection 3.5.2 are fed into the input layer of DeepCDpred. The number of neurons used in the input layer is the same as the dimensions of the feature vector (i.e., 752 for stage 1 and 753 for stage 2); the two hidden layers have 120 and 50 neurons, respectively; the output layer has only one neuron to report the inter-residue contact/distance prediction score. Compared to a classic three-layer feedforward neural network, the neural network used in DeepCDpred is called a deep network. Subsection 2.7.3 of Chapter 2 has explained that deep neural networks are superior to conventional neural network methods.

For each network in DeepCDpred, logistic sigmoid functions were used as the activation functions for all the layers. Since the value range of the logistic sigmoid function is (0, 1), the output from the single output layer neuron is also in the range (0, 1). For the contact prediction network in DeepCDpred, the value represents the score of how likely it is that a given pair of residues is in contact. If the value is close to 1, it means the residues are likely to be in contact; if the value approaches 0, it means the residues are not likely to be in contact. Similarly, for a network predicting an inter-residue distance, the final neuron's output indicates how likely it is that the distance between the two C_β atoms (C_α for glycine) of the two residues being considered is within the distance bin of that network. For both contact and distance predictions made by DeepCDpred, the residue pairs were ranked based on the corresponding network output scores from the highest to

lowest. The prediction result for each distance bin (including the contact bin (0-8Å)) was written as a separate file on the disk with the format:

```
residueA residueB DistanceLowerBond DistanceUpperBond score1  
residueC residueD DistanceLowerBond DistanceUpperBond score2.
```

When it is contact prediction, “DistanceLowerBond” and “DistanceUpperBond” were replaced with 0 and 8, respectively; when it is distance prediction, the two parameters were replaced with the distance bin range.

4.2.3 Training Process and Parameter Optimization of DeepCDpred

The following training process was used for both the amino acid residue contact and distance prediction networks of DeepCDpred. All of the networks were trained by using the Neural Network Toolbox of MATLAB (version 2015b).

Since the machine used in this study (128 gigabytes RAM) could not hold all of the 1066 proteins in the memory at the same time during the neural network training, the training dataset had to be divided into two groups, 524 proteins in group one and 542 proteins in group two. A neural network model was trained with group one and then the converged model was re-trained with group two. The training algorithm for all the models was the conjugate gradient backpropagation with Powell-Beale restarts (training function ‘traincgb’ in MATLAB) (Powell 1977). The initial weights and biases were generated

as random numbers between 0 and 1. The strategy of batch training was used, which means the weights and biases were updated only after all inputs from the training set were processed by the network and the current cumulative error were calculated. For this reason, the training required a very large RAM. The number of validation checks (or early stopping checks) was set to 6 (the default value in the Neural Network Toolbox of MATLAB), which means, when the loss function on the validation set failed to decrease on 6 successive epochs, the training process would stop. Another parameter, the L_2 regularization of the loss function, which also prevents overfitting, was set to 0.01. Please refer to Section 5.12 of the Results chapter (Chapter 5) to check the primary results and Section 6.7 in the Discussion chapter for the analysis.

There are multiple parameters involved in either the architecture or the training process of DeepCDpred. All of them could affect the performance of DeepCDpred. The parameters are: (1) the number of hidden layers in the neural network models (both contact and distance prediction models), (2) the number of neurons in each hidden layer, (3) the regularization in the training process, (4) the training function, (5) the size of the square window in the feature vector.

The reader has already noticed that the values of the parameters were explicitly stated in the above sections and the previous chapter for the convenience of introducing DeepCDpred. However, it is necessary to make it clear how these parameters were optimized.

For the first two parameters, the initial values were chosen as the values reported in

the MetaPSICOV paper (Jones et al. 2015). The initial number of hidden layers was 1, but increasing numbers were tested. In the one hidden-layer architecture, the number of neurons in the hidden layer was varied; the networks were tested, in which the number of neurons in the hidden layer were 55, 80, 100, 110, 120, 130 etc. Several architectures were tested for the two-hidden-layer variants of DeepCDpred, with the numbers of neurons in the two layers being 50 and 45, 70 and 30, 100 and 50, 110 and 50, 120 and 50, 120 and 55, 130 and 55. The number of neurons in the three hidden-layer architecture was tried with the combinations: (a) 120, 50 and 30, (b) 120, 50 and 50. The regularization values of 0.02, 0.01, 0.008, 0.005, 0.001 were tested for all of the above architectures. All of the training functions available in the Neural Network Toolbox of MATLAB were tried (namely, 'trainlm', 'trainbfg', 'trainrp', 'trainscg', 'traincgb', 'traincgf', 'traincgp', 'trainoss' and 'traingdx'; please go to the page of <https://uk.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html> (last check: November 2018) for detailed descriptions of these functions) for all of the above architectures.

Table 4.1. Training function names and the corresponding optimization algorithms in the neural network toolbox of MATLAB.

Function Name	Algorithm
trainlm	Levenberg-Marquardt algorithm
trainbfg	BFGS [#] Quasi-Newton algorithm
trainrp	Resilient backpropagation
trainscg	Scaled conjugate gradient
traincgb	Conjugate gradient with Powell/Beale restarts
traincgf	Fletcher-Powell conjugate gradient
traincgp	Polak-Ribiere conjugate gradient
trainoss	One-step secant backpropagation
traingdx	Variable learning rate backpropagation

[#]: Broyden-Fletcher-Goldfarb-Shanno.

Explanations about the above algorithms can be found on <https://ww2.mathworks.cn/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html> (last check: November 2018).

Besides the logistic sigmoid function, activation functions in all of the layers were also tried with hyperbolic tangent sigmoid functions. The size of the square window used to include CCMpred calculated coevolutional couplings and the contact/distance scores, which were predicted from stage 1, were tested with 7×7 , 9×9 and 11×11 .

At one time, only one of the above changes was made and the others remained the same. Since the neural networks used in this work were initialized with random weights and bias, it was expected that the network could produce different prediction results when it was trained multiple times with the same parameters. In fact, the differences among these were

slight (visit Subsection [5.5.2](#) to check the evidence). Thus, each network architecture was trained three times based on one parameter set. It is necessary to select the best neural network from the different instances. For this aim, a subset of 435 protein chains from the training/validation set was arbitrarily chosen. The neural network that achieved the best amino acid contact/distance predictions on these protein chains was chosen among the multiple runs.

For all the parameters mentioned above, the values were optimized based on improving the contact prediction accuracy of the 435 proteins chosen for validation. If changing a parameter led to an improvement in accuracy of 0.5% or greater, the current value of the parameter was replaced with the new one; if not, the current value would remain.

The pdb ids of the 435 protein chains can be found in Table [C.6](#) of Appendix [C](#).

4.2.4 Explorations to Improve the Performance of DeepCDpred

Two methods were tried to potentially improve DeepCDpred. One of the methods replaced the HHblits homologous sequence database with the metagenomics data but kept the above trained two-hidden-layer DeepCDpred. The other method used five-hidden-layer networks rather than the original two-hidden-layer networks in the contact prediction network group (the network group refers to the four contact prediction networks in Figure [3.1b](#)) of DeepCDpred. ReLU activation functions were also adopted to replace the sigmoid functions in the new hidden layers. The new version of DeepCDpred were

retained with the same training/validation set. Descriptions of the two methods are given below and the corresponding results of them can be found in Section 5.11 and Section 5.12 of the Results chapter (Chapter 5).

Improving the Accuracy of Amino Acid Contact/Distance Prediction of DeepCDpred by Using Metagenomics Data

All of the above amino acid contact/distance predictions were based on the protein sequence dataset, UniRefKB. As the number of protein sequences in metagenome sequence projects is growing faster than that in the UniRefKB database, protein families with a limited number of homologous sequences in the UniRefKB might benefit from adding extra sequences from metagenomics data for amino acid contact prediction and structure prediction (Ovchinnikov et al. 2017b). This study has tested to combine the protein sequences from the metagenomics data with those from UniRefKB as a protein sequence searching dataset, and use it in the amino acid contact/distance prediction procedure of DeepCDpred. The metagenomics data was introduced in Subsection 3.4.1 of the previous chapter. Since the metagenomics data cannot be searched by HHblits, the HMMER tool (Finn et al. 2011) was employed for the homologue search.

Improving the Accuracy of Amino Acid Contact Prediction of DeepCDpred by Using Networks with Five Hidden Layers

As mentioned in Subsection 4.2.3 of this chapter, neural networks in DeepCDpred were trained with the functions from the Neural Network Toolbox of MATLAB. The advantages

of this toolbox include: it is easy to implement, train and test the network models, and it is also simple for coding. Also, since MATLAB provides many functions for data processing, the feature preparation is easily done by MATLAB. But the limitations of this toolbox are crucial: it lacks latest activation functions (e.g., ReLU ([Nair and Hinton 2010](#)), ELU ([Djork-Arne Clevert 2015](#)) and SeLU ([Klambauer et al. 2017](#))), and network training functions (e.g., stochastic gradient descent, or sgd, with mini-batch ([Schmidhuber 2015b](#))) for feedforward networks. The use of sigmoid or hyper tangent activation functions in deep networks has the vanishing gradient problem (visit [Section 6.9](#) in the Discussion chapter ([Chapter 6](#)) for an explanation to this problem), which means adding more hidden layers could hardly improve the performance of a network, or even make it worse. The newly proposed activation functions mentioned above can alleviate this problem. The training function of sgd with a mini-batch could speed up the training process and requires less RAM.

In this section, the two-hidden-layer networks in both stage 1 and stage 2 were replaced with five-hidden-layer ones (seven layers in total); the sigmoid activation function was used at the output layer; the other layers used ReLU activation functions. sgd was used as the training function with the mini-batch size of 32. The implementation used the Python neural network Keras library ([Keras 2018](#)) with Tensorflow ([tensorflow 2018](#)) as the backend. The training/validation set, test set, and training process are the same as those of the two-hidden-layer networks. Again, 15% of the inputs are used as the validation set and the other 85% as the training set, and the final contact score is the

average of four outputs from the four networks, with each using a different amino acid contact cut-off (i.e. $0 - 7.9\text{\AA}$, $0 - 8.0\text{\AA}$, $0 - 8.1\text{\AA}$ and $0 - 8.2\text{\AA}$) to define the classification of the targets.

In the new five-hidden-layer networks, both the loss functions of cross entropy and MSE (mean squared error) were employed, and the accuracies of the amino acid contact prediction based on them were compared.

4.2.5 Contact and Distance Prediction Assessment

In this work, two methods were used to evaluate whether residue contact and distance predictions are successful or not. The first one, commonly adopted by previous studies (including MetaPSICOV) (Jones et al. 2015; Skwark et al. 2013), selected a number of predictions based on the length of the query protein (e.g., $L/10$, $L/5$, $L/4$, $L/3$, $L/2$, L , $1.5L$ etc., where L is the length of the query protein, that is, the number of amino acids). Before the assessment, residue pairs were ranked by the predicted contact scores and then selections were made by choosing a certain number of residue pairs with high contact scores. For each selection, a measure to assess the accuracy of the predicted contacts or distances was calculated by using the formula:

$$\text{Accuracy} = \frac{\text{True Positives In the Selected Predictions}}{\text{All Selected Predictions}} \times 100\% \quad (4.1)$$

where True Positives (TPs) are predictions that are observed contacting residue pairs in the experimental structure. Accuracy scores are between 0% and 100%; 0% indicates all the predictions are incorrect, and 100% indicates all the predictions are correct. This strategy was used to compare the contact prediction accuracies of DeepCDpred and the previous algorithms as described in the section Review of Amino Acid Coevolution Analysis (Section 2.6). The advantages of this measure are: (a) it is easy to calculate; (b) it is widely used in the research community, which makes it easier to compare contact accuracy with other algorithms; and (c) the accuracy is well recognized or accepted by the community.

In this section, another contact/distance selection strategy is also introduced, which is based on the neural network output scores. Specifically, for a query protein, the score of each contact/distance prediction is compared to a score cut-off; the predictions with scores above the cut-off are selected. The logic of this method is as follows.

The contact/distance prediction scores of a query protein could be affected by the “quality” of its MSA (‘quality’ indicates the factors related to the MSA that could impact the residue contact and distance predictions, such as the proportion of gaps in each column and the effective number of sequences in the MSA; the result shown in Subsection 5.5.6 of Chapter 5 proves this statement). Thus, for a query protein with a few effective sequences in its MSA compared to its sequence length (\mathbf{Nf}), the contact/distance prediction scores should on average be smaller than the one with many effective sequences. Therefore, based on this selection method, the number of contact/distance predictions that can be selected

is determined by the intrinsic quality of the query protein's homologous sequences. Check Subsection 5.5.6 in the Results chapter (Chapter 5) for the evidence. However, this method also has limitations. Firstly, for the query protein with low-quality MSAs, it may improve the contact/distance accuracy based on the above analysis, but it also reduces the number of contact/distance predictions, which means there are fewer constraints fed into the structure modelling software (i.e. Rosetta) for structure prediction. Whether more false positives or fewer predictions is worse for structure prediction is unknown. Secondly, it is hard to compare the contact/distance accuracies among different algorithms according to this measure, since the scores from these algorithms are calculated with different methods, which are not equivalent.

Based on the above analysis, the widely used contact/distance selection method of choosing $L/10$, $L/5$, $L/4$, $L/3$, $L/2$, L , $1.5L$ predictions were used to compare the performance of DeepCDpred with other algorithms. However, the accuracies of structure predictions were compared based on two ways to select contacts: one was to use neural network output scores; the other was to use a fixed number of top predictions. (check Subsection 5.6.2 and Subsection 5.6.3 in the Results chapter).

4.2.6 Realization of DeepCDpred

The source code of DeepCDpred was written in Python. Neural networks trained with MATLAB code were converted to Python readable files. Some protein chains from the

training/validation set were used to check if the conversion was correct. Prerequisites (e.g. database and software) were downloaded to the local machine. Users can go to the website of <http://proteincoevolution.bham.ac.uk> to access DeepCDpred. When a job with one query protein sequence is submitted to the server, DeepCDpred will run and prediction results will be sent to the email address provided by the user in the job. More information about the server can be found in Section 5.13 of the next chapter.

4.2.7 Contribution Analysis of the Features of DeepCDpred

Different features from the input feature vector make different contributions to a neural network model. Some features might even make no contributions. It is important to rank the contributions of features of DeepCDpred. Knowing the ranking is useful for removing the non-contributive features and improving the effectiveness of the most important features. For example, if the feature of secondary structure predictions is among the top-ranked features, more accurate secondary structure prediction algorithms might improve the performance of DeepCDpred.

In this study, to speed up the calculations, only 524 protein chains were selected from the whole training/validation set introduced in Subsection 3.4.1 of Chapter 3. Limiting the number of chains allows the time-consuming steps of feature vector creation and training to be performed more quickly and thus to explore more possibilities. These chains were actually the first group of the training/validation set of DeepCDpred (see Subsection 4.2.3

‘Training Process of DeepCDpred’) and the lengths range from 16 to 394 amino acids. The pdb id list of the training/validation set can be found in Table C.5 of Appendix C). Also, only networks of contact prediction in stage 1 were trained, again, to speed up the calculations. Each time, only one type of feature was removed from the total 752-dimensional feature vector. For example, when the importance of coevolutionary coupling calculated from QUIC was evaluated, this type of feature was removed from the feature vector of DeepCDpred, and all of the remaining features were kept; four neural network models representing four target range (i.e., $0 - 7.9\text{\AA}$, $0 - 8.0\text{\AA}$, $0 - 8.1\text{\AA}$ and $0 - 8.2\text{\AA}$) were trained based on the new features from the 524 protein chains. Similarly, 85% of the inputs were randomly selected as the training set and the other 15% as the validation set. All of the settings in the training process were the same as those of DeepCDpred.

In addition to the coevolutionary coupling calculated from QUIC, the contributions from other features, i.e. the coevolutionary coupling calculated from CCMpred (plmDCA), the coevolutionary coupling calculated from FreeContact (mfDCA), mutual information, statistical potential, secondary structure prediction, solvent accessibility, amino acid profile, site entropy, the number of effective sequences, the number of sequences, the length of the protein chain and sequence separation, were also evaluated in the same way. It is worth noting that the indices of the inputs in the training set (85%) and the indices of the inputs in the validation set (15%) were randomly chosen from all of the indices of the residue pairs of the 524 protein chains; however, these two sets of indices were divided only once, and for the networks in each feature removal, they were kept. In other words, for all the

networks, the inputs in the training set (also the validation set) were from the same set of residue pairs of the 524 protein chains, but with a different feature removed from the feature vector of DeepCDpred. The contribution ranking of these features is shown in Subsection 5.5.5 of Chapter 5.

4.2.8 Differences between DeepCDpred and MetaPSICOV

Although the development of DeepCDpred was inspired by MetaPSICOV and they do share some things in common (the reader can find these by comparing the introduction of MetaPSICOV in Subsection 2.6.10 and the introduction of DeepCDpred in the above paragraphs and the previous chapter), they are different regarding the following aspects.

- a. MetaPSICOV is only capable of predicting inter-residue contacts, while DeepCDpred can make both inter-residue contact and distance predictions (four scores output from the four groups of neural networks for each pair of residues in the target sequence, which represent how likely the spatial distance of the residue pair is in the range of 0–8Å, 8–13Å, 13–18Å, and 18–23Å).
- b. The architectures of the neural network models in DeepCDpred and MetaPSICOV are different. The former uses two hidden layers with 120 and 50 neurons, respectively, while the latter adopts only one hidden layer with 55 neurons. A new version of DeepCDpred, as described above, has five hidden layers.

- c. Some features are different in the two algorithms. In MetaPSICOV, PSIPRED ([McGuffin et al. 2000](#)) and SOLVPRED ([Jones et al. 2015](#)) were used for the secondary structure and the solvent accessibility prediction, respectively; while, DeepCDpred uses the software of SPIDER2 for the preparation of both kinds of features. In Subsection 2.7.1, it was shown that SPIDER2 is better than PSIPRED by citing the paper ([Heffernan et al. 2015](#)). Another feature is that the correlated mutation scores were predicted by QUIC in DeepCDpred; MetaPSICOV used the scores predicted by PSICOV. As described in Subsection 2.6.8, the algorithm of QUIC has the same objective function as PSICOV, but with a much faster speed of optimization. The result in Section 5.4 of the Result chapter (Chapter 5) has proved this point, and at the same time, QUIC makes very similar inter-residue contact prediction accuracy. Therefore, PSICOV was replaced by QUIC in DeepCDpred.
- d. The strategy of arranging features is different in both algorithms. Unlike using a square window of size 9×9 centred at the position pair (i, j) and including all the contact scores predicted from CCMpred in this window, the feature vector of stage 1 in MetaPSICOV only includes the contact score of CCMpred at (i, j) ; this is the main reason why feature vector size in MetaPSICOV is less than that in DeepCDpred in stage 1. In stage 2, MetaPSICOV uses an 11×11 square window to include the contact scores predicted from stage 1, while DeepCDpred still adopts the 9×9 window. In this stage, the mid-point window of five was used in DeepCDpred, while MetaPSICOV does not have this window.

4.3 The Development of DeepCDpred_AbInitio

4.3.1 Introduction

The method of predicting protein structure based on constraints (including both contact and distance prediction) predicted by DeepCDpred and Rosetta *ab initio* modelling is called DeepCDpred_AbInitio.

In this thesis, in addition to the comparison of contact prediction accuracy of DeepCDpred with other methods, it's also worth comparing the structure prediction quality based on the contact/distance predictions from DeepCDpred to other methods. As discussed in the above Subsection 4.2.5 of 'Contact and Distance Prediction Assessment', two methods for contact prediction selection were used in this study. Thus, there were two ways to compare the quality of the structure predictions among different algorithms based on the selections: one was selecting the same number of contact predictions from different methods and using the same Rosetta *ab initio* protocol; the other was selecting the contact predictions that have the same expected contact accuracy and using the same Rosetta *ab initio* protocol.

Besides the comparisons of the quality of the structure predictions based on the contact predictions from different algorithms, comparisons were also made between the quality of the structure predictions from DeepCDPred based on contact prediction only and that based on both contact and distance predictions.

To quantify the spatial similarity of a predicted structure and the corresponding experimentally determined structure, a TM-score was reported. Details about TM-score have already been introduced in Subsection 2.8.5 of Chapter 2.

In the next four subsections, how to select the contact and distance predictions to prepare the geometry constraints for Rosetta *ab initio* modelling and the Rosetta *ab initio* modelling protocol are introduced. The relations of the four subsections are indicated in Figure 4.2, which shows that all of the protein structure prediction methods introduced in Subsection 4.3.2, Subsection 4.3.3 and Subsection 4.3.4 use the same Rosetta modelling protocol.

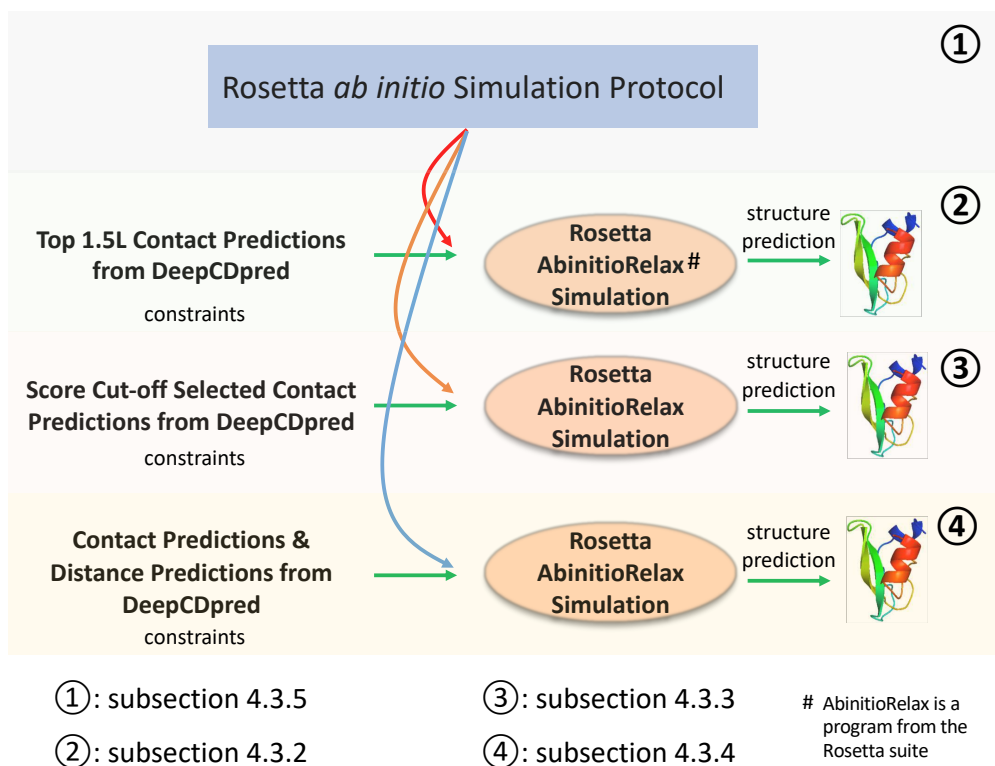


Figure 4.2. Diagram of the development of DeepCDpred_AbInitio.

4.3.2 Rosetta *Ab Initio* Modelling with the Top 1.5L Contact Predictions

This method of employing the predicted contacts for structure prediction is commonly used in recent studies (Ovchinnikov et al. 2015, 2017a,b). In earlier studies, the top L (Michel et al. 2014) contact predictions were used for structure predictions rather than the top 1.5L predictions. In this thesis, the top 1.5L ranked contact predictions from DeepCDpred and MetaPSICOV were selected for each protein chain in the test set. The ‘BOUNDED’ score function (https://www.rosettacommons.org/docs/latest/rosetta_basics/file_types/constraint-file, last check: November 2018) was used to construct constraint inputs for Rosetta *ab initio* modelling, with the upper bound and lower bound set to 8 and 0, respectively.

In Section 5.10 of the Results chapter (Chapter 5), comparisons of the quality of structure predictions are made between DeepCDpred and RaptorX by using the top 1.5L ranked predicted contacts from each algorithm for eight proteins.

In order to make the comparisons fairly, for the different constraints from these algorithms, all of the structure predictions used the same Rosetta *ab initio* modelling protocol, which will be introduced later in this section.

4.3.3 Rosetta *Ab Initio* Modelling with the Contact Predictions Determined by Score Cut-off

As mentioned in Subsection 4.2.5, there are some good reasons to suppose that the method of selecting contact predictions based on neural network scores is superior to that of simply taking the top 1.L predictions, regarding to protein structure predictions. With this method, contact predictions selected from DeepCDpred and MetaPSICOV were included as constraints in Rosetta *ab initio* modelling by using the same ‘BOUNDED’ score function as mentioned above. Since MetaPSICOV and DeepCDpred are two different algorithms, one cannot choose the cut-offs with the same value. Instead, cut-offs of MetaPISCOV and DeepCDpred were selected that gave equivalent expected accuracies of contact predictions, as reported in the Results chapter (Chapter 5). All Rosetta protocols were the same, as described in Subsection 4.3.5, except by which C_β carbons were constrained.

4.3.4 Rosetta *Ab Initio* Modelling with Both Contact Predictions and Distance Predictions from DeepCDpred

In addition to the comparisons of structure predictions above based on contact predictions only, for DeepCDpred, the comparison of structure predictions was made between using contact predictions only and using both contact predictions and distance predictions. The

purpose of this comparison is to assess whether the distances predicted by DeepCDpred are useful for improving structure predictions.

The two constraint selecting methods, i.e. by using a neural network score cut-off and taking the top 1.5 L predictions, potentially allow four combinations to select contacts and distances together for structure predictions. In this study, only two groups were made using the predicted contacts and distances. In the first group, contact predictions from DeepCDpred were selected as the top 1.5L ranked ones for each protein, and distance predictions from DeepCDpred were selected by setting a score cut-off for each distance bin for each protein; in the second group, both contact and distance predictions were selected by the score cut-off standard.

It is worth noting that there are three groups of comparisons of structure predictions to evaluate the effectiveness of the distance prediction from DeepCDpred (in this paragraph, all the contact and distance predictions are from DeepCDpred): (a) the comparison of structure predictions between using the top 1.5L contact predictions only and using the 1.5L contact predictions & the distance predictions selected by score cut-offs; (b) the comparison of structure predictions between using contact predictions only, selected by a score-cut-off, and using contact predictions selected by a score cut-off & distance predictions selected by score cut-offs as well; (c) the comparison of structure predictions between using the two methods for contact & distance prediction selections. The results of (a), (b) and (c) are shown in Subsection [5.6.5](#), Subsection [5.6.6](#), and Subsection [5.6.7](#) of the Results chapter (Chapter [5](#)), respectively.

In the calculations, the following two steps are used before the distance predictions are added into the Rosetta *ab initio* modelling.

1. Redundancy Reduction in Inter-Residue Distance Prediction of DeepCD-pred

Redundancy is a major problem when predicting distance. The sources of it include: (a). the two amino acids are on the same alpha helix or beta strand; (b). the sequence separation of the two amino acids is too small (i.e., the two positions are so close in sequence that they inevitably appear in a spatial distance bin); (c). they are neighbors of two contacting residues.

Two steps were taken to deal with this issue. Firstly, residue pairs predicted by SPIDER2 on the same alpha helix or beta strand were removed from the training and the prediction processes. Secondly, a minimum cut-off of sequence separation was set for each distance bin. The three cut-offs come from the analysis of Figure 4.3. In this figure, the inter-residue distance distribution versus sequence separation of 435 experimental protein structures from the training/validation set was calculated (i.e., less than 25% sequence identity between any pair of proteins in the 435 structures). All of the residue pairs appearing on the same predicted alpha helix or beta strand had already been removed before the analysis. The pdb ids of the 435 proteins are listed in Table C.6 of Appendix C.

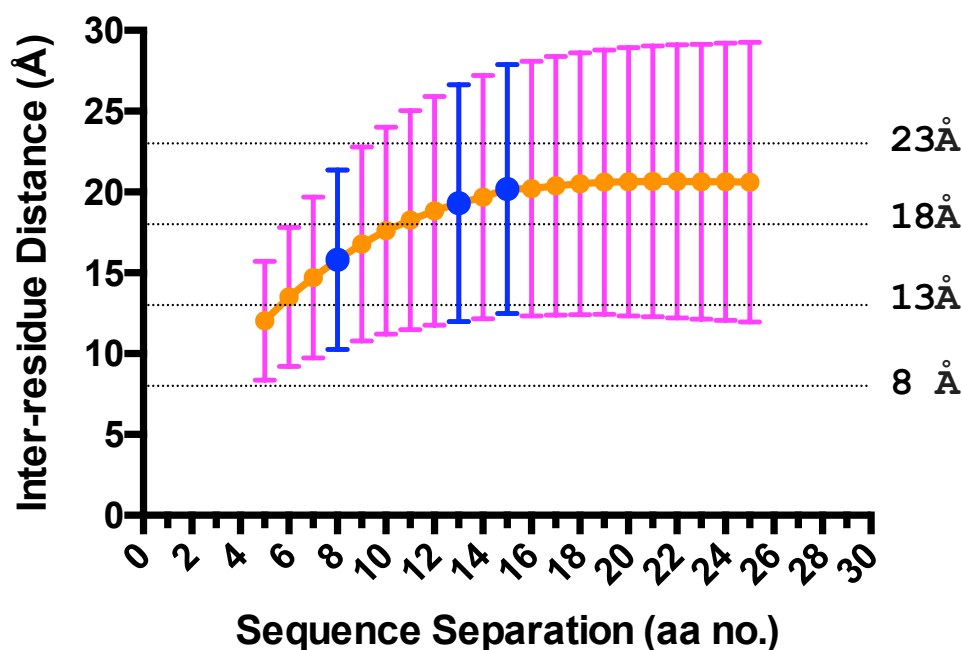


Figure 4.3. The distribution of inter-residue distance in terms of sequence separation (measured in amino acid number, aa no.). The result comes from the calculation of 435 experimental protein structures in the training/validation set and is shown as mean \pm std. Three blue highlighted sequence separations (8, 13 and 15) were chosen as the cut-offs for the distance predictions in bins (8 – 13Å], (13 – 18Å] and (18 – 23Å] respectively.

The usefulness of a distance prediction is defined by whether it is out of the expected range. To make it clearer, the distance bin (8 – 13Å] is taken as an example. If the sequence separation of a position pair is five amino acids, without any prediction, it is expected that their distance in space is likely to be in the range of (8 – 13Å], from Figure 4.3. When the sequence separation is eight (or even seven), the distance between them is likely to be in the range of (13 – 18Å] on average (see the left blue highlighted bar in the figure). Therefore, if DeepCDpred predicts that it has a big chance (high output value from the neural network) of being in (8 – 13Å], the prediction is significant

(which means the prediction is out of expectation and thus nontrivial). When the sequence separation is larger, the prediction of the two positions in this distance bin is even more significant. From the figure, a sequence separation of seven amino acids can also be used as the cut-off for the distance bin $(8 - 13\text{\AA}]$. However, eight was chosen (left blue highlighted bar) in this study as a trade-off between making the predictions more significant and keeping as many predictions as possible. For the same reason, thirteen (middle blue highlighted bar) was chosen as the minimum sequence separation for predicting residue pairs in distance bin $(13 - 18\text{\AA}]$. As for the bin $(18 - 23\text{\AA}]$, the situation is different. Since from sequence separation of fifteen (right-most blue highlighted bar) amino acids to twenty-five, the average inter-residue distance hardly changes and always stays in this bin, redundancy cannot be avoided effectively in the distance bin of $(18 - 23\text{\AA}]$ and was thus accepted. A sequence separation cut-off of fifteen was set in this bin to keep as many predictions as possible.

2. Beta Strand Pairing

A beta sheet is characterized by two beta strands running in the same (parallel)/opposite (anti-parallel) direction held together by hydrogen bonds. The length of the hydrogen bond is generally around 3.0 \AA . Therefore, it is more accurate to constrain two residues than a simple contact in a protein structure. The idea in the state-of-the-art beta strand pairing program, bCov ([Savojardo et al. 2013](#)), was adopted, and a small program was developed to replace the PSICOV residue contact scores used in bCov with the contact scores predicted from DeepCDpred.

After the beta strand pairings were generated, all the contact and distance predictions that have both positions on the two strands (e.g., AB and CD in Figure 4.4) were removed, since the hydrogen bonding pattern could determine the topology of the beta sheet, and the redundant constraints will be less precise than the hydrogen bond constraints. Figure 4.4 is an example of anti-parallel beta sheet prediction.

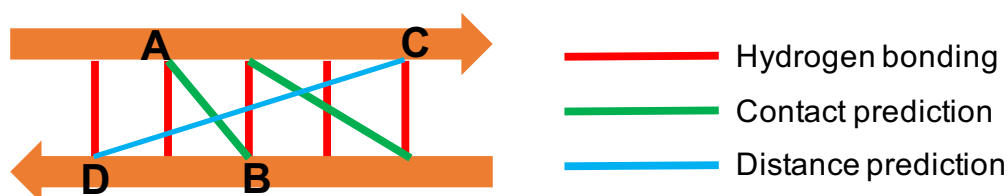


Figure 4.4. Beta strand pairing in an anti-parallel sheet. After the hydrogen bonding pattern has been inferred, all of the contact and distance predictions related to this beta sheet are removed.

4.3.5 Rosetta *Ab Initio* Modelling Protocol

For the prediction of protein 3D structures, the *ab initio* simulation program, AbinitioRelax from the Rosetta software suite, was used together with the predicted contacts, distances and beta sheets predicted as described above and the secondary structure prediction obtained from SPIDER2. The first step was to use the fragment generating program ‘make_fragments.pl’ from the Rosetta suite to create the three-residue and nine-residue fragments. Then, all of the above predictions together with the query protein sequence were fed into AbinitioRelax by using the following parameters (Rosetta simulation protocol):

```
-in:file:fasta PATH_TO_QUERY_SEQUENCE_FILE
-in:file:frag3 PATH_TO_THREE_RESIDUE_FRAGMENTS_FILE
-in:file:frag9 PATH_TO_NINE_RESIDUE_FRAGMENTS_FILE
-abinitio:relax
-nstruct 100
-out:pdb
-out:overwrite
-database PATH_TO_ROSETTA_DATABASE_DIRECTORY
-cst_fa_file PATH_TO_RESTRAINT_FILE
-use_filters true
-psipred_ss2 PATH_TO_SECONDARY_STRUCTURE_PREDICTION_FILE
-abinitio::increase_cycles 20
-abinitio::rg_reweight 0.5
-abinitio::rsd_wt_helix 0.5
-abinitio::rsd_wt_loop 0.5
-constraints:cst_weight 0.5
-constraints:cst_fa_weight 0.5
```

100 candidate structures for each target protein were generated and the one with the lowest Rosetta energy score was picked out as the top 1 model.

4.4 The Development of the Method to Estimate the Quality of Predicted Structures

Logistic sigmoid functions were used as the activation functions in both the hidden layer and the output layer. L_2 regularization was set to 0 and the mean squared error was used

as the loss function. MATLAB's default parameters were used for other parameters. All of the training functions in the neural network toolbox of MATLAB were tested with other parameters keeping the same. The best one was selected by comparing the prediction accuracy on the validation set (the validation set was the same for the comparison; the data of the comparison is not shown; finally, the function of 'traincgf' was chosen). Before fixing the number of neurons in the hidden layer to be five, other numbers, such as eight, ten and fifteen were also tested.

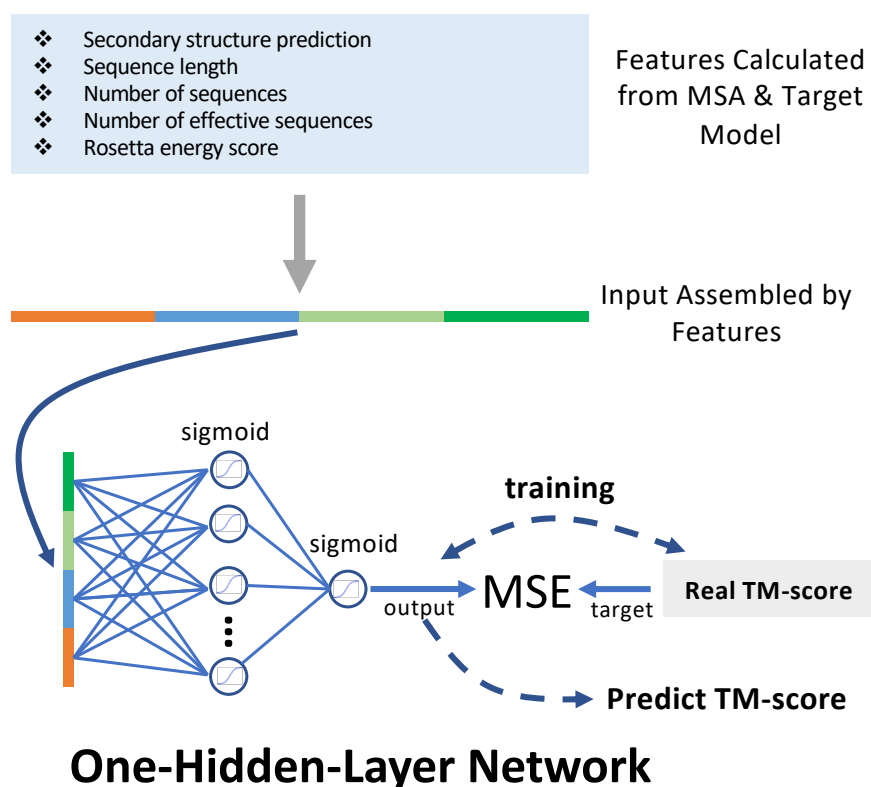


Figure 4.5. Diagram of the development of the structure prediction quality evaluation model.

The test set of this model is the same as the test set of DeepCDpred.

4.5 Comparisons of Contact and Structure Prediction Between DeepCDpred, RaptorX and NeBcon

Both amino acid contact and structure predictions were compared among DeepCDpred and the two newly published algorithms, RaptorX and NeBcon, which were introduced in Subsection [2.6.10](#), on eight proteins. The results can be found in Section [5.10](#) (Chapter [5](#)).

CHAPTER 5

RESULTS

5.1 Overview of This Chapter

This chapter presents results on the following:

1. the proportion of possible amino acid pairs that are in contact in 250 unrelated (‘unrelated’ is defined as any pair of sequences in the 250 proteins with no greater than 25% sequence identity) real protein structures;
2. the analysis of the composition of the training/validation and test sets used by DeepCDpred;
3. comparisons of the speed and of the inter-residue contact prediction accuracy of PSI-COV and QUIC;

4. the accuracy of the contact predictions of DeepCDpred as compared with MetaPSICOV as well as the accuracy of DeepCDpred's distance predictions;
5. the analysis of the contribution of different features to the quality of DeepCDpred's contact prediction;
6. comparisons of structure predictions between using DeepCDpred contacts and using MetaPSICOV contacts based on two methods of contact selection;
7. comparisons of the quality of structures as predicted by DeepCDpred between using contact predictions only and using contact and distance predictions together;
8. a true blind test of the DeepCDpred_AbInitio structure prediction;
9. testing the ability to predict the TM-score of a structure model;
10. comparisons of the contact accuracy and the structure prediction between DeepCDpred and two other recently published algorithms, RaptorX and NeBcon;
11. improving contact/distance predictions of DeepCDpred by using metagenomics sequences;
12. improving contact predictions of DeepCDpred by using neural networks with five hidden layers and ReLU activation functions;
13. the introduction of the online server developed in this work.

Other results can be found in [Appendix B](#).

5.2 Percentage of Amino Acid Contact in Real Proteins

The percentage of amino acid pairs that are in contact in 250 unrelated protein chains is shown in Figure 5.1a with sequence separation of one amino acid (means that all the amino acid pairs are included) and five amino acids (the minimum sequence separation of contact prediction of DeepCDpred). The averages are 8.2% and 3.0%, respectively. The error bars represent the standard deviation (4.6% and 1.6%, respectively). These proteins were randomly selected from the 1,066 training/validation proteins of DeepCDpred. Figure 5.1b shows the distribution of the number of amino acids per chain. The counts for the four bins are 76, 150, 14 and 10, respectively. The full list of the pdb ids of the 250 protein chains can be found in Table C.1 of Appendix C.

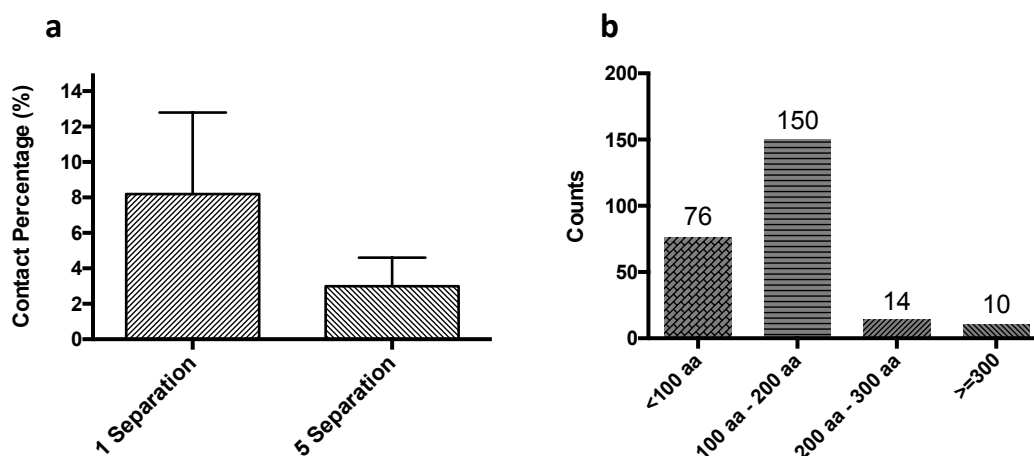


Figure 5.1. The percentage of contacting amino acid pairs in 250 unrelated protein chains in terms of one amino acid separation and five amino acid separation (graph a); the distribution of the number of amino acids of these chains (graph b).

5.3 Characterisation of the Training/Validation Set and the Test Set

The training/validation set and the test set are characterised by globular/membrane, oligomeric state and classes of the proteins presented. Globular proteins dominate both the training/validation set and the test set (Figure 5.2). Only eight membrane proteins are present in the training/validation set, while one is included in the test set, representing 0.8% and 0.9% in percentage, respectively. The percentage of the membrane chains in PISCES, the source database, is 1.3%. In addition, the number of chains from multimeric proteins is about three times the ones from monomers in the training/validation set; on the contrary, in the test set, the majority of chains are from monomers (80.6% versus 19.4%). As a comparison, the percentage of monomer chains in the 2,957 protein chains, in which the 1,066 training/validation protein chains were chosen from (the explanation about the 2,957 chains was mentioned in Subsection 3.4.1), is 44.3%. The result is shown in Figure 5.3. In Section 6.3 of the Discussion chapter (Chapter 6), the impact of the big difference of the monomer chain composition in the training/set and the test set on the contact prediction accuracy is analysed.

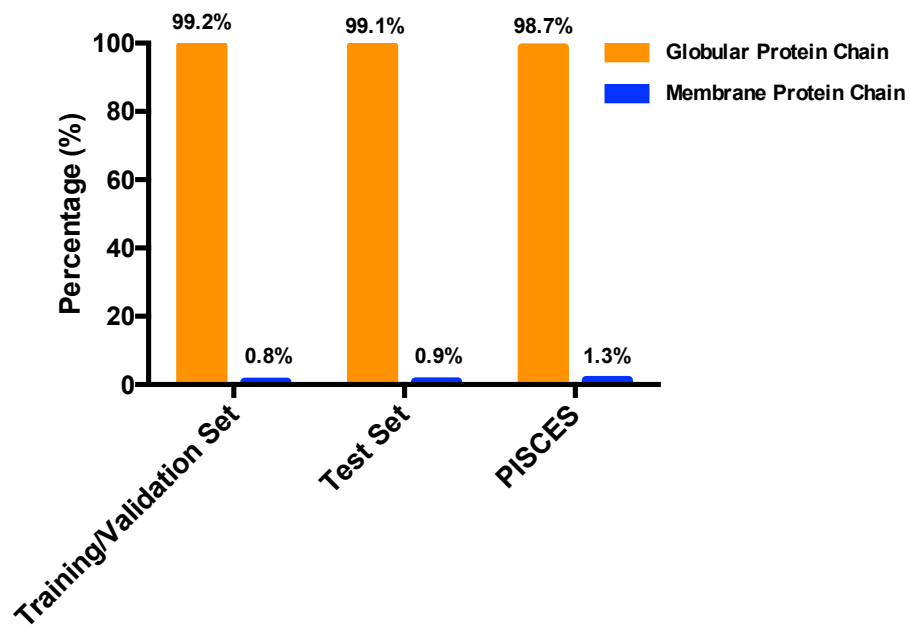


Figure 5.2. The distributions of globular and membrane proteins in the training/validation set and the test set.

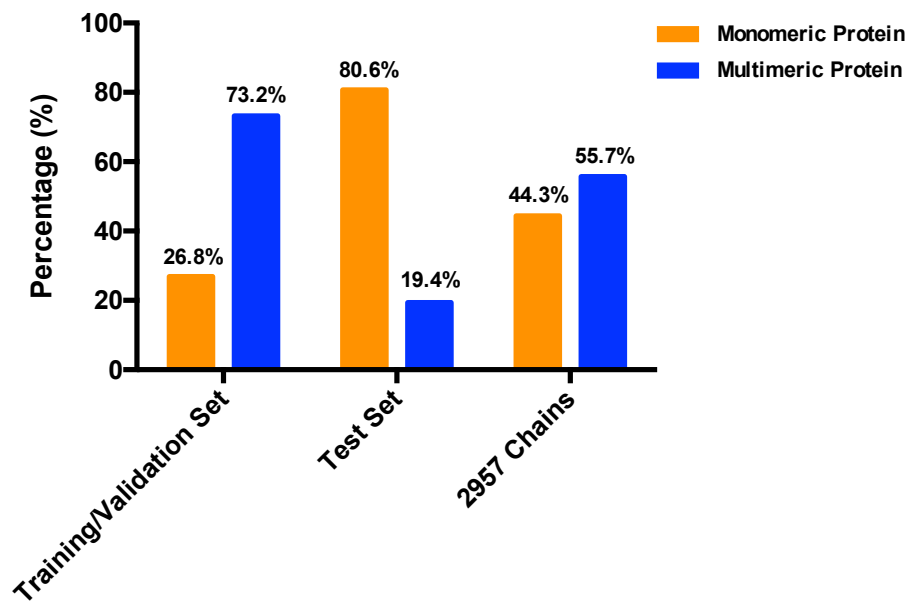


Figure 5.3. The distributions of protein stoichiometry in the training/validation set, the test set of DeepCDpred, as well as the 2,957 protein chains, from which the training/validation data were chosen.

The pdb id list of the membrane structures in PDB was downloaded from <http://blanco.biomol.uci.edu/mpstruc/listAll/pdbIdList> (downloaded on November 14, 2017). It was used to check how many chains in the training/validation set, the test set, and PISCES, respectively, are membrane proteins. Meanwhile, the full pdb id list of the monomer structures was obtained from the PDB website (<https://www.rcsb.org/pdb/home/home.do>, November 14, 2017). After the monomers were identified, the rest in each set were deemed to be multimers.

Although the 108 test proteins of DeepCDpred taken from MetaPSICOV were indicated by the studies of Jones *et al.* (Jones *et al.* 2012, 2015) as monomers (the studies didn't provide supporting evidence to the claim that MetaPSICOV's test proteins are all monomeric), Figure 5.3, which summarises the relevant data from PDB, lists 19.4% of them as being multimeric. There is thus a discrepancy.

Proteins can also be classified as α proteins, β proteins, α/β proteins, $\alpha+\beta$ proteins, coiled-coil proteins (hereinafter referred to as coil proteins) and membrane proteins. The definitions of the protein classes $\alpha+\beta$, α/β , and coil that are used here are adopted from SCOP (Lo Conte *et al.* 2000). The difference between $\alpha+\beta$ proteins and α/β proteins are that β sheets in the former are mainly antiparallel, while in the latter, they are mainly parallel. Coil proteins are dominated by coils. With the definitions, each protein in both the training/validation set and the test set of DeepCDpred was checked on the SCOP website (<http://scop.mrc-lmb.cam.ac.uk/scop/index.html>, last check: November 2018) to determine the class type. Examples of these six classes are shown in Figure 5.4. The

pdb ids of them are shown at the bottom of the figure. These proteins are all from the test set. The class distributions of the proteins in both the training set and the test set are shown in Figure 5.5. The numbers on the bars are the percentages of proteins in each class. It clearly shows $\alpha+\beta$ proteins dominate both the training/validation set and the test set. The main difference between Figure 5.5a and Figure 5.5b is the percentage of α and β proteins – 25% and 9% versus 11% and 16%. The protein class for each individual protein chain in both the two sets can be found in Table D.1 and Table D.2 of Appendix D.

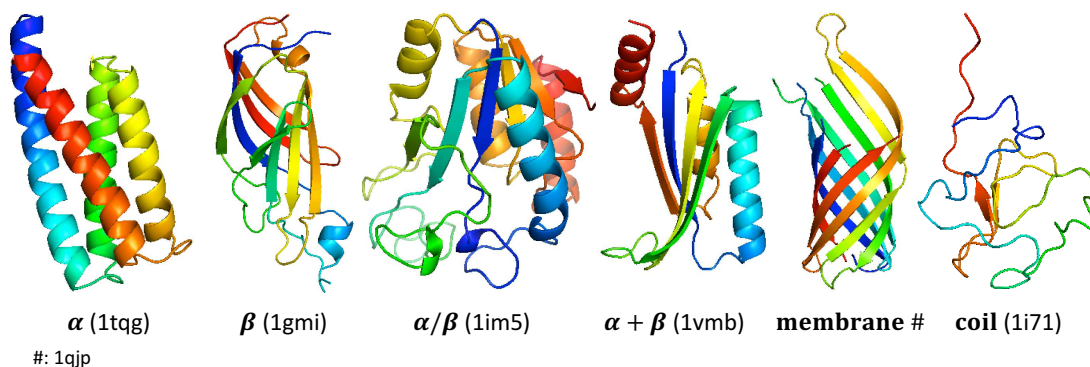


Figure 5.4. Examples of the six protein classes. pdb ids are shown in the brackets.

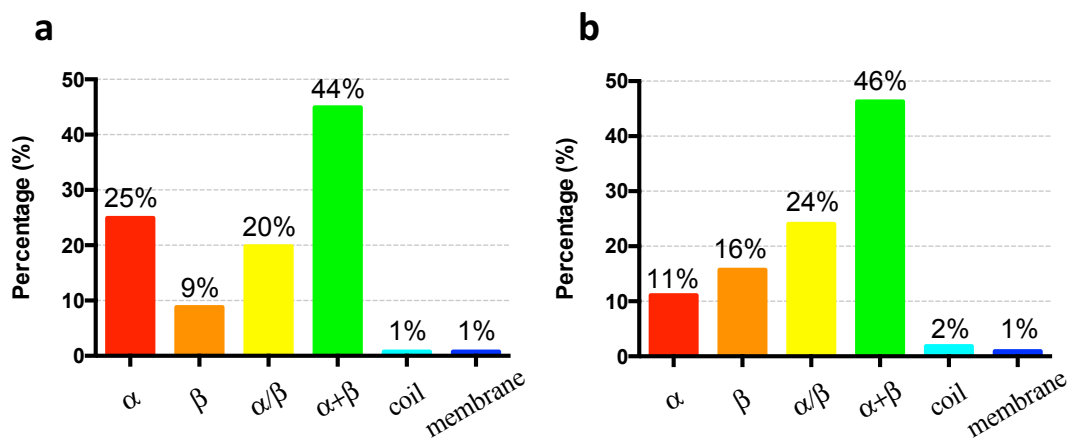


Figure 5.5. Protein class distributions of the 1,066 proteins chains in the training/validation set (graph a), and of the 108 protein chains in the test set (graph b). Numbers above the histograms are the percentages of the protein chains in each class.

5.4 Comparisons of the Speed and of the Inter-residue Contact Prediction Accuracy between PSICOV and QUIC

To replace PSICOV, QUIC is expected to be faster than PSICOV on the same data set and the same computing device, but maintains a similar amino acid contact prediction accuracy. 221 protein chains from the training/validation set, with lengths ranging from 50 to 386 amino acids, were arbitrarily chosen for both comparisons. The pdb ids of these proteins are listed in Table C.2 of Appendix C.

For the comparison of the accuracies of inter-residue contact predictions, the top $1.5L$ (L is the chain length) contact predictions, ranked according to the predicted coupling score from each chain, were selected for each method. The reason for choosing the number of the top $1.5L$ is that it was chosen as the number of the top-ranked contact predictions for the structure predictions (Subsection 5.6.2). The accuracies for both methods are shown in Figure 5.6a, where the accuracies of contact predictions for each individual protein chain from both methods are very close.

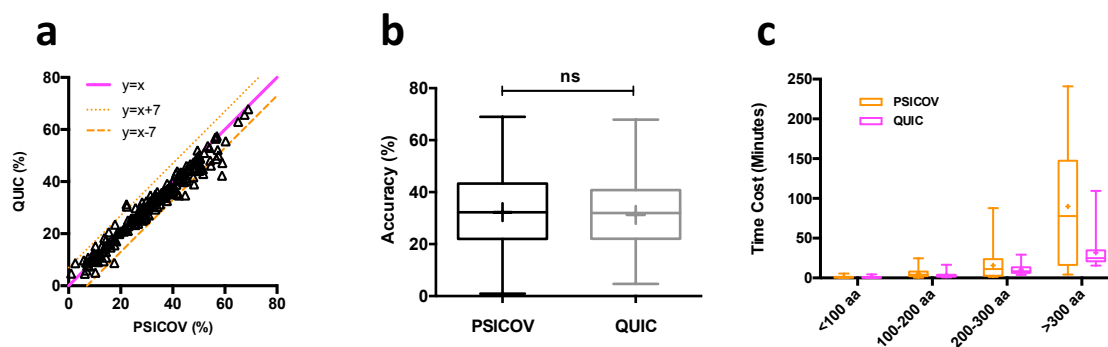


Figure 5.6. Comparisons of contact prediction accuracy and speed between PSICOV and QUIC. 221 proteins were chosen for the comparison, and the accuracies of the top 1.5L amino acid contact predictions for both PSICOV and QUIC of each protein are shown in the graph of (a). (b) the boxplot form of graph (a); no significant difference (ns) is found between the accuracies of the two methods by a paired t-test ($p > 0.05$). In the boxplots, whiskers represent the maximum and minimum values and the middle line in the box stands for median and cross for mean. (c) speed comparisons between PSICOV and QUIC for the 221 proteins in each bin of protein length.

In Figure 5.6a, the two lines of $y = x + 7$ and $y = x - 7$ highlight the proteins that deviate greatly from $y = x$, and thus which proteins are predicted better by one algorithm as compared with the other are shown. Notably, 94% of the proteins are located between the two lines, and 6% (13 in number) are outliers.

The protein class distributions of α , β , α/β , $\alpha+\beta$ and coil of the 221 chains are 14%, 14%, 27%, 43% and 2%, respectively. The classes of α , β , α/β and $\alpha+\beta$ proteins are among the outliers whose amino acid contacts were predicted more accurately with PSICOV than QUIC (percentages of α , β , α/β , $\alpha+\beta$ and coil are 18%, 18%, 18%, 46% and 0%, respectively). As compared with the distribution of the five classes in the 221 proteins, the percentages of α , β , and $\alpha+\beta$ proteins are (slightly) higher; but α/β protein is lower. A χ^2 test shows the two protein class distributions have no significant difference ($p = 0.25$

> 0.05). For the outliers predicted more accurately by QUIC, there are only two proteins from two classes (α/β and $\alpha+\beta$). The set is too small to draw a conclusion. The protein classes, together with stoichiometry and protein types, of these 13 proteins are listed in Table 5.1. For the stoichiometry, 99.5% (220 in number) of the 221 protein chains are monomers, and only 0.5% are polymers (only 1 in number). For the protein type, 99.5% (220 in number) of the 221 protein chains are globular proteins, and 0.5% are membrane proteins (only 1 in number). It is clear that the outliers do not have stoichiometry and protein type specificity too.

A paired t-test showed that there is no significant difference between the accuracies of the two methods ($p > 0.05$), which is also evident in Figure 5.6b.

As for the comparison of running speed, QUIC is faster than PSICOV for all the bins of chain lengths (Figure 5.6c). The largest difference of speed appears when the protein chain length is greater than 300 amino acids, and QUIC just took less than 1/4 time of PSICOV on average (24.4 minutes vs. 106.7 minutes). For all of the protein chains used, QUIC took 6.9 minutes on average, while PSICOV took 17.9 minutes. This speed comparison was performed on a Linux machine with an 8-core i7-3770 processor and a 32 GB of RAM.

Table 5.1. The protein classes of the outliers in the comparison of contact prediction accuracies between PSICOV and QUIC. Accuracy difference is defined as the accuracy predicted with PSICOV subtracting the accuracy predicted with QUIC for the same protein. Rows in the table are ranked by accuracy difference from the highest to the lowest.

PDB ID	Accuracy Difference #	PSICOV Top 1.5L	QUIC Top 1.5L	Protein Class	Stoichiometry	Protein Type
1zgk	16.6%	58.9%	42.3%	β	monomer	globular
1yfq	11.9%	59.1%	47.2%	β	monomer	globular
1hh8	10.1%	44.8%	34.7%	α	monomer	globular
1cjw	9.2%	48.2%	39.0%	$\alpha+\beta$	monomer	globular
2j5y	8.8%	17.6%	8.8%	α	monomer	globular
1gz2	8.7%	51.7%	43.0%	$\alpha+\beta$	monomer	globular
2hzc	8.5%	54.6%	46.2%	$\alpha+\beta$	monomer	globular
1h0p	8.1%	56.8%	48.7%	$\alpha+\beta$	monomer	globular
1aoe	7.6%	42.4%	34.7%	α/β	monomer	globular
1jvw	7.5%	57.5%	50.0%	$\alpha+\beta$	monomer	globular
1r85	7.0%	41.9%	34.9%	α/β	monomer	globular
2gke	-8.0%	22.4%	30.4%	$\alpha+\beta$	monomer	globular
2h1v	-9.1%	22.2%	31.3%	α/β	monomer	globular

#: Accuracy difference is defined as the accuracy predicted with PSICOV subtracting the accuracy predicted with QUIC for the same protein.

5.5 Amino Acid Contact and Distance Predictions of DeepCDpred

The following results are discussed in this section.

1. Optimizing the parameters of DeepCDpred.
2. Comparison of the contact prediction accuracies between a single network and the average of four networks.
3. Comparisons of the contact prediction accuracies between DeepCDpred and other algorithms, namely mfDCA, QUIC, plmDCA, and MetaPSICOV, based on the number of top-ranked contact predictions ($L/10$, $L/5$, $L/4$, $L/3$, $L/2$, L , $1.5L$).
4. The distance prediction accuracy of DeepCDpred.
5. Examples of comparisons of contact prediction accuracies between MetaPSICOV and DeepCDpred.
6. Feature contribution ranking analysis for DeepCDpred.
7. Limitations of the method of contact prediction selection for choosing top-ranked predictions (e.g. top $1.5L$).
8. The accuracies of contact and distance predictions of DeepCDpred based on the prediction selection strategy of score cut-off.

5.5.1 Parameter Optimization of DeepCDpred

This subsection only displays the result of the comparisons of the amino acid contact prediction accuracies between the optimized one-hidden-layer network and the optimized two-hidden-layer network. Figure 5.7 shows the comparison result of stage 2 of DeepCDpred. For the top-ranked 1.5L predictions, the two-hidden-layer network can ensure 0.8% higher contact prediction accuracy for the 108 test proteins as compared with the one-hidden-layer network. It is worth noting that both the two networks are individual networks (see the next subsection); both were trained with the contact range defined as $0 - 8\text{\AA}$.

This is the reason why DeepCDpred was chosen as a two-hidden-layer architecture.

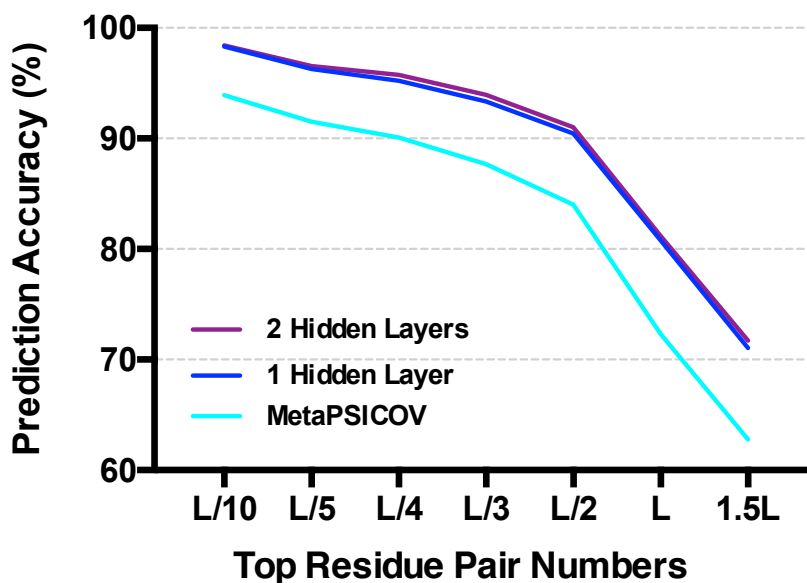


Figure 5.7. The comparison of amino acid contact prediction accuracies between the optimized one-hidden network and the optimized two-hidden-layer network. The result of MetaPSICOV for the same proteins are used as a reference.

Another result of choosing a different number of neurons in the two hidden layers is shown in Figure B.2 of Appendix B.

The optimized values of other parameters have already been explicitly stated in the previous two chapters.

5.5.2 Comparison of Contact Prediction Accuracies Between a Single Network and the Average of Four Networks

In the contact prediction part of DeepCDpred, four networks were used; the final contact prediction score for each residue pair comes from the average of the outputs from the four networks (each with a slightly different distance bin range, i.e. $0 - 7.9\text{\AA}$, $0 - 8.0\text{\AA}$, $0 - 8.1\text{\AA}$ and $0 - 8.2\text{\AA}$). The contact prediction accuracy determined by the output of each individual network contact range versus by the average output is shown in Figure 5.8 (left graph: stage 1; right graph: stage 2). For both stage 1 and stage 2, the accuracy of the contact predictions generated by averaging the outputs of the four networks is $\approx 1.5\%$ higher than that achieved by using any individual network.

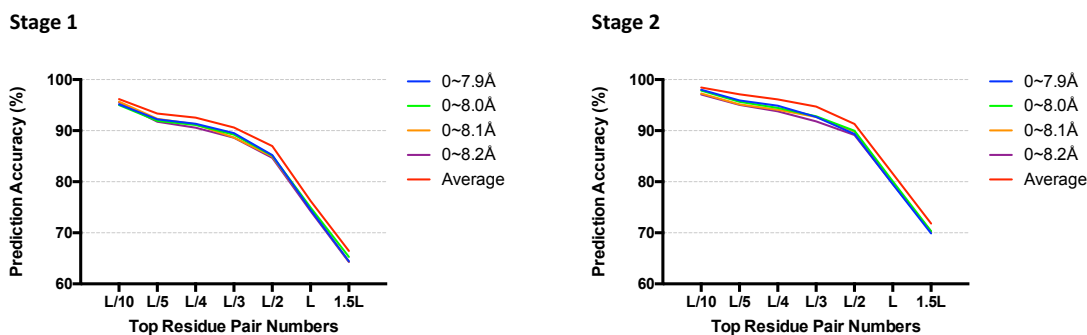


Figure 5.8. Contact prediction accuracy calculated from averaging output scores from four networks vs contact prediction accuracy calculated with the output score from each individual network for both stage 1 and stage 2. The contact predictions made by averaging of four networks is $\approx 1.5\%$ higher than any individual network for both stages in terms of accuracy.

5.5.3 Comparison of Contact Prediction Accuracies between DeepCDpred and Other Algorithms and Distance Prediction

Accuracy of DeepCDpred

The comparison of the amino acid contact prediction accuracies among mfDCA, QUIC, plmDCA, MetaPSICOV and DeepCDpred for the 108 proteins in the test set are shown in Figure 5.9a. It is clear that DeepCDpred’s predictions are the most accurate ones no matter what number of top predictions is chosen (here, the number is one of L/10, L/5, L/4, L/3, L/2, L, 1.5L). The FreeContact (Kajan et al. 2014) and CCMpred (Seemayer et al. 2014) implementations of the mfDCA and plmDCA methods were used. A detailed comparison between MetaPSICOV and DeepCDpred is shown in Table 5.2. Unlike

MetaPSICOV and other algorithms in Figure 5.9a, DeepCDpred also predicts distant amino acid couplings (distance predictions) in three bins, as shown in Figure 5.9b.

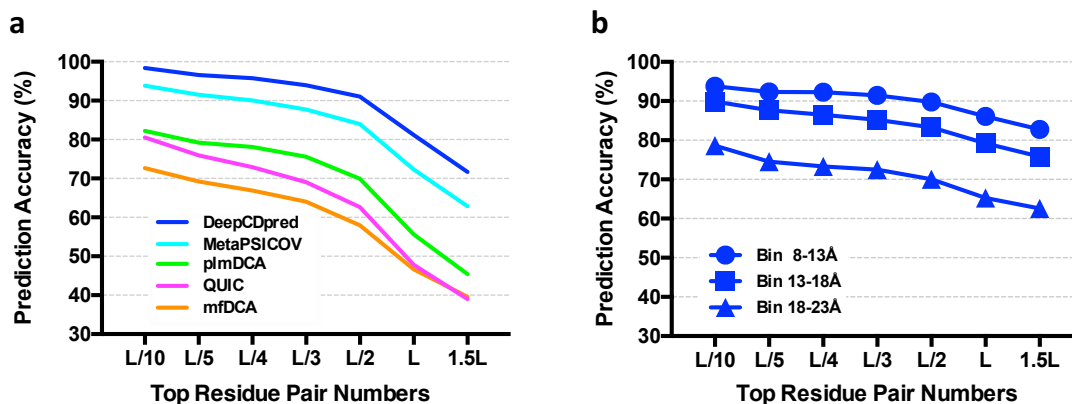


Figure 5.9. Comparison of contact prediction accuracies between DeepCDpred and previous algorithms shown in a; b is the accuracy of DeepCDpred distance predictions in three bins.

Table 5.2. Comparison of contact prediction accuracies between between MetaPSICOV and DeepCDpred.

#	L/10	L/5	L/4	L/3	L/2	L	1.5L
MetaPSICOV	93.9%	91.5%	90.1%	87.7%	84.0%	72.3%	62.8%
DeepCDpred	98.4%	96.6%	95.8%	94.0%	91.1%	81.2%	71.8%

5.5.4 Examples of Contact Prediction Comparisons Between MetaPSICOV and DeepCDpred and Distance Prediction of DeepCDpred

Figure 5.10 shows the amino acid contact predictions of three protein chains based on MetaPSICOV (Figure 5.10A), and DeepCDpred (Figure 5.10B), respectively. The pdb

ids of these proteins are 1tqg, 1t8k and 1iib. These proteins were selected arbitrarily from the test set (108 protein chains) of DeepCDpred. For each protein chain, the top $1/3L$ predictions from each algorithm were selected (more predictions drawn on the graph could make them hard to distinguish). Comparing Figure 5.10A with Figure 5.10B, DeepCDpred generates more true-positive and fewer false-positive predictions than MetaPSICOV does in the top $L/3$ contact predictions.

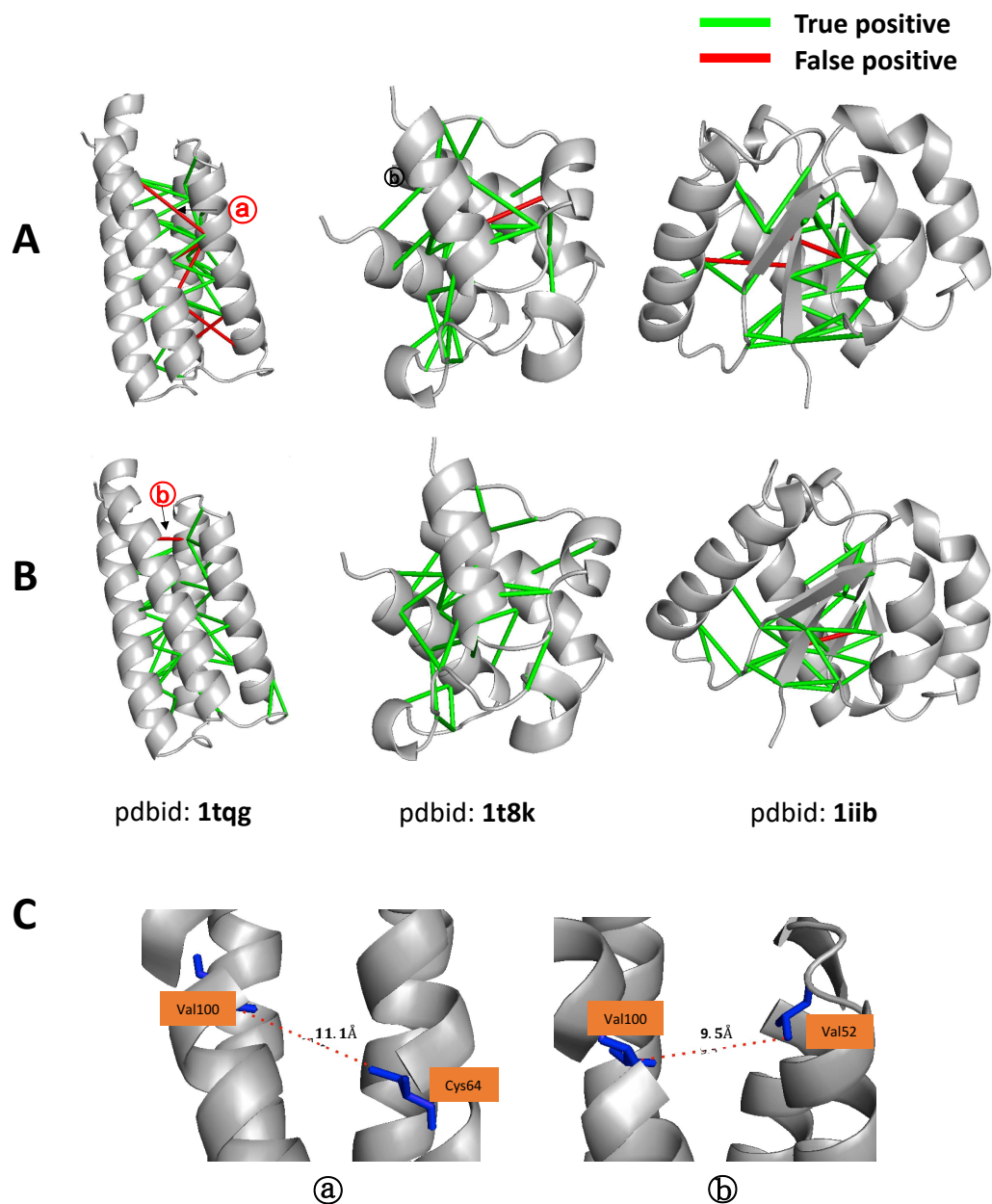


Figure 5.10. Comparison of the top 1/3 predictions between MetaPSICOV and DeepCDpred for three example proteins. A: MetaPSICOV and B: DeepCDpred. The top L/3 predictions were selected for each protein and drawn on the plots. The background structures are the experimental structures. Contact predictions of DeepCDpred have less false positives. C: A close up view of false positive example from each of MetaPSICOV and DeepCDpred, a and b are as indicated on 1tqg in A and B.

Figure 5.11 visualizes the distance predictions in the bin of 8-13 Å for the same proteins as noted in Figure 5.10. All these predictions are from DeepCDpred. Again, the top 1/3L predictions were selected for each protein chain.

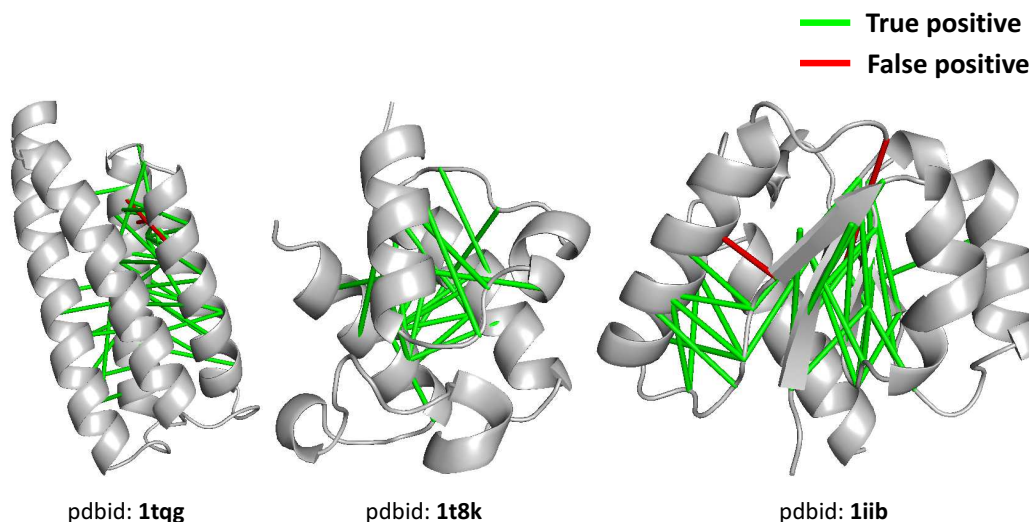


Figure 5.11. Examples of distance bin 8-13 Å prediction from DeepCDpred. The proteins shown here are the same as those in Figure 5.10. The background structures are also the protein experimental structures. The colour scheme of true positives and false positives is also the same as that in Figure 5.10.

5.5.5 Feature Contribution Ranking Analysis for DeepCDpred

The contribution ranking of the features of DeepCDpred was evaluated by comparing with the ‘residual’ networks, i.e. a network with one feature was removed from the full stage 1 network with two hidden layers, as described in Subsection 4.2.7. All of the networks were trained with the same training/validation set (group 1 in the training/validation set of DeepCDpred).

There are 13 types of features in the networks of stage 1 of DeepCDpred, namely, amino acid profile, secondary structure prediction, asa (accessible surface area) prediction, positional entropy, statistical potential, sequence separation, EVfold coevolutionary coupling, QUIC coevolutionary coupling, CCMpred coevolutionary coupling, MI (mutual information), sequence length (or chain length), the number of effective sequences, and the number of sequences. Thus, there are 13 ‘residual’ networks. The contact prediction accuracies of the test set from the 13 ‘residual’ networks and the ‘intact’ network are shown in Figure 5.12 and Table 5.3.

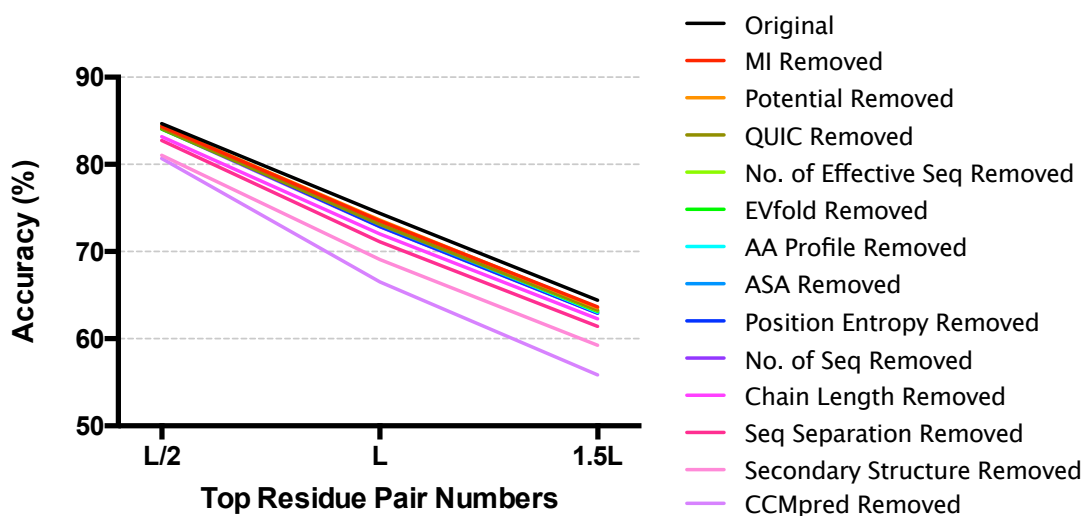


Figure 5.12. Accuracy of contact prediction changes with each type of feature removed from the stage 1 networks of DeepCDpred. All of the networks were trained with the same training/validation set. “Original” represents the network trained with the same architecture and features of a standard stage 1 DeepCDpred network; other “residual” networks only keep the architecture of stage 1 networks of DeepCDpred, but with one type of feature removed, as implied by the labels in the legend. The network listed in the legends are ranked by the contact prediction accuracy of the top 1.5L predictions. In the figure, “Seq” means “Sequence” and “AA” means “amino acid”.

Table 5.3. How the removal of one feature changes the accuracy of the stage 1 network of DeepCDpred. The networks are ranked by the contact prediction accuracy of the top-ranked 1.5 predictions.

Network	L/10	L/5	L/4	L/3	L/2	L	1.5L
Original	95.4%	91.6%	90.8%	89.1%	84.7%	74.4%	64.4%
MI Removed	95.2%	92.2%	90.6%	88.4%	84.3%	73.5%	63.6%
Potential Removed	95.2%	92.1%	90.8%	88.7%	84.4%	73.6%	63.6%
QUIC Removed	94.0%	90.8%	89.3%	87.9%	84.1%	73.1%	63.3%
No. of Effective Seq Removed	95.7%	92.0%	90.8%	89.0%	84.4%	73.2%	63.2%
EVfold Removed	95.6%	91.7%	90.9%	88.0%	84.0%	73.2%	63.2%
AA Profile Removed	94.2%	91.1%	90.2%	88.3%	84.3%	73.3%	63.0%
ASA Removed	95.3%	91.7%	90.4%	88.0%	84.3%	73.1%	63.0%
Position Entropy Removed	95.3%	91.6%	90.3%	87.8%	84.3%	72.9%	63.0%
No. of Seq Removed	94.6%	91.7%	90.4%	88.1%	84.1%	72.9%	62.9%
Chain Length Removed	94.8%	91.5%	90.1%	87.6%	83.2%	72.0%	62.3%
Seq Separation Removed	94.5%	91.1%	89.4%	86.7%	82.7%	71.1%	61.4%
Secondary Structure Removed	93.1%	89.6%	88.2%	85.8%	81.0%	69.1%	59.2%
CCMpred Removed	93.8%	90.9%	89.1%	86.1%	80.7%	66.5%	55.9%

In the figure, in order to make the lines clearer, only the accuracies of the top-ranked L/2, L and 1.5L predictions are shown. Table 5.3 also lists the contact prediction accuracies for the top-ranked L/10, L/5, L/4 and L/3 contacts. The figure legend lists the networks

in the order of the contact accuracy of the top 1.5L predictions from the highest to the lowest; so do the rows in Table 5.3.

The results indicate that taking away any of the features could decrease the contact prediction accuracy of DeepCDpred. The coupling calculated from CCMpred and the secondary structure prediction are the top two features that most affect the contact prediction performance of DeepCDpred.

In Section 6.6 of Chapter 6, how to improve the contact prediction accuracy (as well as probably distance prediction) based on these results will be discussed.

5.5.6 Limitations of the Contact Prediction Selection Method of “Top L Terminology”

The meaning of the ‘Top L Terminology’, just as shown in the above subsections, is to select the predicted contacts (or distances) based on a fraction of the top-ranked L predictions. As analysed in the Model Development chapter (Subsection 4.2.7), this method has the limitations of that it could miss out on true-positive predictions for some proteins that have ‘good quality’ MSAs, and also wrongly include false-positive predictions for some proteins that have ‘bad quality’ MSAs. Here, the ‘quality’ of an MSA is defined by its Nf value: $Nf = M_{eff}/\sqrt{L}$, where M_{eff} is the number of effective sequences in the MSA. This definition is adopted by Ovchinnikov *et al.* (Ovchinnikov *et al.* 2017b). In their paper, the authors discovered an approximately linear relationship between the Nf

value and the protein structure prediction quality. When $N_f > 64$, the predicted structure is likely to have the same fold as the native structure.

Four example proteins are shown here to illustrate the limitation. In Table 5.4, the N_f values of proteins 1bdo and 1eaz (pdb id) are $\gg 64$, while the N_f values of 1j3a and 1beh < 64 . From Table 5.2, the average contact prediction accuracy of the top-ranked 1.5L of the test set is 71.8%. For the first two proteins, the contact prediction accuracies of the top 1.5L are 90.1% and 85.2%, respectively. Notably, both values are higher than the average accuracy. When more top-ranked predictions are selected until the contact prediction accuracies decrease to $\approx 71.8\%$, 214 and 234 predictions can be selected, which are significantly more than the top 1.5L counts, 121 and 155, respectively. The minimum network scores of these predictions decrease to 0.20 and 0.16, respectively (shown in the brackets of the ‘Accuracy Above Cutoff’ column).

However, for the other two proteins in the table, 1j3a and 1beh, the accuracies of the top 1.5L contact predictions are lower than the average. If the contact predictions of the ones with network scores ≥ 0.49 and 0.38 respectively, which increase the accuracy to $\approx 71.8\%$, the number of selections are only 147 and 208, respectively (less than the top 1.5L contact counts, 194 and 277, respectively, check the ‘Count of Top 1.5L’ column).

Table 5.4. Four examples show the limitations of selecting the top-ranked 1.5L contact predictions. Count of Top 1.5L means the number of contacts in the top 1.5L predictions. Top 1.5L Accuracy means the contact prediction accuracy of the top 1.5L contacts; the numbers in the brackets are the minimum contact neural network scores in the top 1.5L contacts. Count Above Cutoff means the number of contacts when the top-ranked contact predictions are selected based on a neural network score cutoff which makes the accuracy of the selected predictions approximating to the average top 1.5L contact prediction accuracy of the test set of DeepCDpred. The accuracies of the selected contacts based on this method are shown in the column of “Accuracy Above Cutoff”, which are very close to 71.8%; the numbers in the brackets are the minimum contact neural network scores in the selected contacts for each protein. The last column is the Nf values that indicate the MSA quality of each protein.

PDB ID	Length	Count of Top 1.5L	Top 1.5L Accuracy	Count Above Cutoff	Accuracy Above Cutoff	Nf
1bdo	80	121	90.1%(0.48)	214	71.5%(0.20)	819
1eaz	103	155	85.2%(0.50)	234	71.8%(0.16)	482
1j3a	129	194	61.9%(0.38)	147	71.4%(0.49)	18
1beh	184	277	65.0%(0.27)	208	71.6%(0.38)	42

5.5.7 Accuracies of Contact and Distance Predictions of Deep-CDpred Based on Score Cut-offs

There is another way to select all of the predictions with neural network output scores above a cut-off for each protein chain. The results of contact and distance predictions from DeepCDpred based on this strategy are shown in Figure 5.13.

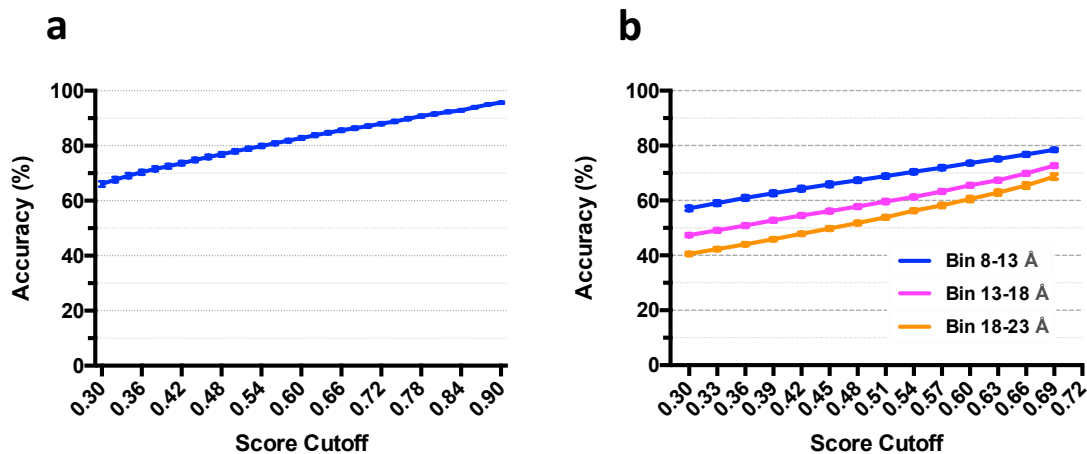


Figure 5.13. Results of contact (graph a) and distance (graph b) predictions of DeepCDpred shown in the form of accuracy versus neural network output score cut-off. The error bar in each plot is measured by MSE (mean squared error).

5.6 Protein Structure Prediction

Based on the constraints of predicted inter-residue contact and distance and the Rosetta *ab initio* modelling protocol introduced in the Model Development chapter (Chapter 4), protein structure predictions of the test set are presented, including:

1. the variation between different *ab initio* structure predictions with Rosetta;
2. the comparison of structure predictions based on the top-ranked 1.5L contact predictions of DeepCDpred and MetaPSICOV;
3. the comparison of structure predictions based on the predicted contacts of DeepCDpred and MetaPSICOV that have the same average contact prediction accuracy;
4. the comparison of structure predictions between employing the top-ranked 1.5L contact predictions of DeepCDpred and employing the top-ranked 1.5L contact predictions plus distance predictions selected based on a score cut-off of DeepCDpred;
5. the comparison of structure predictions between employing the contact predictions of DeepCDpred selected based on a score cut-off and employing the contact predictions selected based on a score cut-off plus distance predictions of DeepCDpred selected based on score cut-offs;
6. the relationship between the quality of the predicted structure and the Nf value;

7. examples of structure predictions based on the contact and distance constraints of DeepCDpred.

5.6.1 The Variation between Different *ab initio* Predictions by Rosetta

Rosetta *ab initio* modelling uses the Monte Carlo simulation to find the native structure, so the models generated from different modelling runs but with the same constraints and modelling protocol might vary greatly. Thus, it is important to verify how large the variation of the structure predictions is for a set of proteins before proceeding to assessing the effects of different contact/distance prediction procedures on structure predictions.

The proteins were chosen as the test set of DeepCDpred and for each protein in the test set, the contact predictions from DeepCDpred with contact score ≥ 0.4 were selected and used as the constraints for Rosetta *ab initio* modelling. The modelling protocol was introduced in the Model Development chapter (Subsection 4.3.5). For each protein, 100 candidate structures were generated, and the one with the lowest Rosetta energy score was selected as the choice (the top 1 model). Then, a second run, which took all of the settings (both the constraints and the modelling protocol) from the first run, was used to generate another 100 structures, from which the one with the lowest energy was selected.

For each protein in the test set, the two lowest energy structures from the two runs, were compared to the experimental structure to calculate the TM-score by using TM-align. The

variation of the two TM-scores for each protein are shown in Figure 5.14. No significant difference (ns) is found between the TM-scores from the two runs by using a paired t-test ($p > 0.05$). The TM-score averages of the two runs are 0.671 and 0.669, respectively – the difference is only 0.002. This provides a baseline for any variance among runs with different contact prediction protocols and allows to distinguish differences due to changes in contact predictions from variations associated with the basic modelling protocol.

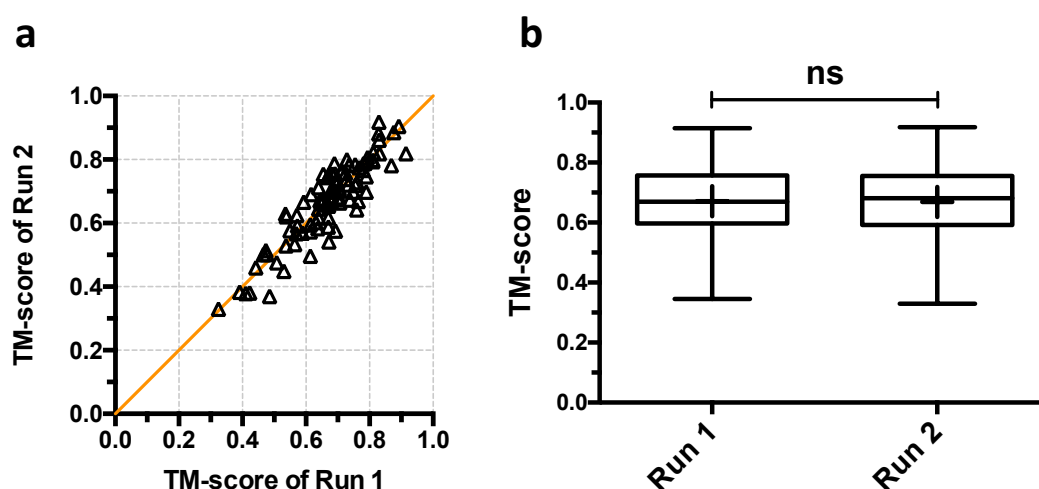


Figure 5.14. Two independent sets of prediction calculations for the 108 protein test set give little variation on average. DeepCDpred protocol settings were used in both cases with the contacts selected that were above the 0.4 contact neural network score cut-off; (a) comparison of TM-score of individual proteins between independent runs. (b) comparison of the distribution of scores for independent runs. A paired t-test indicates no significant difference (ns) ($p > 0.05$) between the two respective runs of the TM-scores of the lowest Rosetta energy structures with respect to the experimental structures. Whiskers indicate the minimum and maximum TM-score values in each group; middle lines in the boxes are the median values and the crosses represent the two means.

5.6.2 Comparison of Structure Prediction Based on the Top-ranked 1.5L Contact Predictions from DeepCDpred and MetaPSICOV

The top-ranked 1.5L contact predictions of DeepCDpred and MetaPSICOV were chosen and fed into the Rosetta *ab initio* modelling protocol introduced in Chapter 4, respectively. For each protein chain in the test set, the top 1 models with the lowest Rosetta energy scores (in the rest of this chapter, all of the terms ‘the top 1 model’ have the same meaning as the one here) based on the two algorithms were selected. A comparison of the two sets of predictions is given in Figure 5.15.

In Figure 5.15a, there is a strong bias of the TM-score distribution toward the side of DeepCDpred. The boxplots in Figure 5.15b further support the above argument. The TM-scores of the best structures predicted with the top-ranked 1.5L contact predictions from DeepCDpred are significantly higher than those with the top-ranked 1.5L contact predictions from MetaPSICOV through a paired t-test ($p < 0.001$).

A question is raised here: are there proteins of a certain class predicted by DeepCDpred contacts better than that by MetaPSICOV contacts or vice versa? In order to answer this question, the TM-score distribution in Figure 5.15a is analysed. Two lines of $y = x + 0.1$ and $y = x - 0.1$ are drawn on the graph. 85% of the proteins that are better-predicted with contacts from MetaPSICOV appear in the area formed by $y = x$ and $y = x - 0.1$,

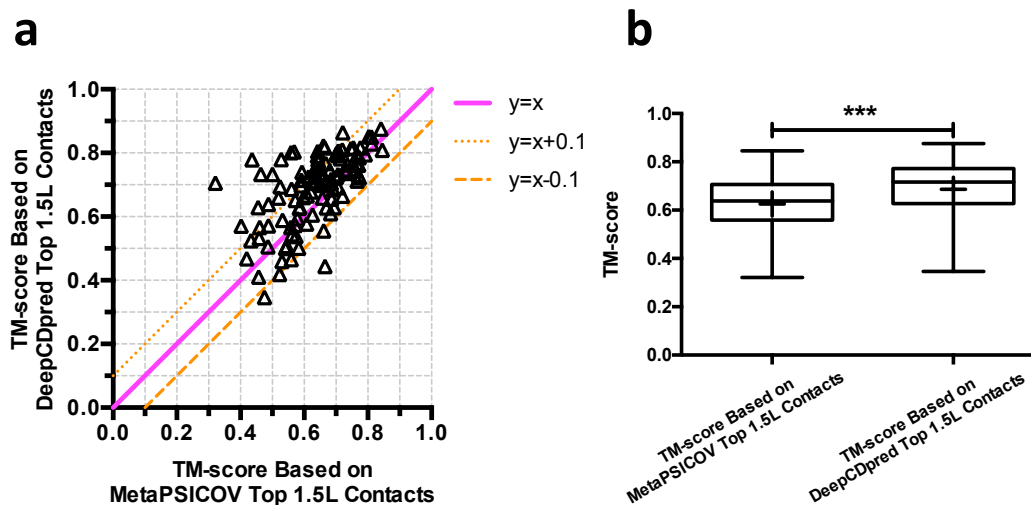


Figure 5.15. The accuracy of structure predictions with the top 1.5L contacts predicted by MetaPSICOV, compared to those by DeepCDpred. The selected models are those with the lowest Rosetta energy score. (a), scatter plot of the comparison; each triangle represents one protein in the test set and the majority of the proteins have a better-predicted structure by DeepCDpred constraints rather than that by MetaPSICOV constraints. (b), boxplots of the comparison in (a); a significant difference ($p < 0.001$) was found between the two groups of structure predictions. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.14b.

and 65% of the proteins that are better-predicted with contacts of DeepCDpred appear between $y = x$ and $y = x + 0.1$. The proteins located outside the area of $y = x + 0.1$ and $y = x - 0.1$ are considered to be outliers. The proteins above $y = x + 0.1$ are the ones whose top 1 models are significantly better-predicted with contacts from DeepCDpred, while the proteins below $y = x - 0.1$ are the ones whose top 1 models are significantly better-predicted with contacts from MetaPSICOV. Table 5.5 lists these outliers and the classifications of them.

From the table, the percentages of the protein classes α , β , α/β , $\alpha+\beta$, coil and membrane of the proteins whose top 1 models are better-predicted with the top 1.5L contacts from

Table 5.5. The classes of the proteins whose structures are better-predicted with the top-ranked 1.5L contacts of DeepCDpred than with the top-ranked 1.5L contacts from MetaPSICOV, or vice versa. TM-score difference is defined as the TM-score of the top 1 model predicted with the top-ranked 1.5L contacts of DeepCDpred subtracting the TM-score of the top 1 model (lowest Rosetta energy model) predicted with the top-ranked 1.5L contacts from MetaPSICOV for the same protein. Rows in the table are ranked by TM-score difference from the highest to the lowest.

PDB ID	TM-score Difference	TM-score of DeepCDpred	TM-score of MetaPSICOV	Protein Class
1k7j	0.38	0.70	0.32	$\alpha+\beta$
1htw	0.34	0.77	0.43	α/β
1vmb	0.27	0.73	0.46	$\alpha+\beta$
1g2r	0.25	0.77	0.52	α/β
1avs	0.25	0.80	0.55	α
1c44	0.24	0.80	0.56	$\alpha+\beta$
1ctf	0.23	0.73	0.50	$\alpha+\beta$
1vfy	0.17	0.62	0.45	coil
1mug	0.17	0.69	0.52	α/β
1dix	0.17	0.57	0.40	$\alpha+\beta$
1jos	0.16	0.80	0.64	$\alpha+\beta$
1bdo	0.16	0.82	0.66	β
1ku3	0.15	0.81	0.66	α
1d4o	0.15	0.63	0.48	α/β
1chd	0.15	0.79	0.64	α/β
1kq6	0.14	0.73	0.59	$\alpha+\beta$
1vp6	0.14	0.86	0.72	$\alpha+\beta$
1qjp	0.14	0.65	0.51	membrane [#]
1p90	0.13	0.77	0.64	$\alpha+\beta$
1eaz	0.13	0.77	0.64	$\alpha+\beta$
1fx2	0.12	0.68	0.56	$\alpha+\beta$
1m8a	0.12	0.71	0.59	$\alpha+\beta$
1aba	0.12	0.75	0.63	$\alpha+\beta$
1jo0	0.11	0.74	0.63	$\alpha+\beta$
1i1n	0.11	0.72	0.61	α/β
1ktg	0.11	0.75	0.64	$\alpha+\beta$
1cxy	0.10	0.69	0.59	$\alpha+\beta$
1gzc	0.10	0.56	0.46	β
1ej8	-0.11	0.41	0.52	β
1xff	-0.11	0.55	0.66	$\alpha+\beta$
1i71	-0.13	0.34	0.47	coil
2phy	-0.22	0.44	0.66	$\alpha+\beta$

[#]: 1qjp is a beta out membrane protein (<https://www.rcsb.org/pdb/explore/explore.do?structureId=1qjp> (last check: November 2018)).

DeepCDpred are 7%, 7%, 21%, 57%, 4% and 4%, respectively. By comparing these findings with the protein class distributions in the training/validation set of DeepCDpred (25%, 9%, 20%, 44%, 1% and 1% for the six classes in the training/validation set), it can be seen that α protein is under represented; $\alpha+\beta$ is over represented. A χ^2 test shows the two protein class distributions are significantly different ($p = 0.0078 < 0.05$). As for the proteins whose structures are better-predicted with the top 1.5L contacts from MetaPSICOV (last four rows in the table), the set is too small to draw a conclusion.

It is shown in Table 5.5 that there are six proteins (1k7j, 1htw, 1vmb, 1vfy 1dix and 1d4o) for which the correct folds were not predicted using MetaPSICOV contacts, but such were predicted using the contacts from DeepCDpred. There were only two proteins (1ej8 and 2phy) whose folds were not correctly predicted using DeepCDpred contacts but such were with MetaPSICOV.

To provide a direct view of the comparison of structure predictions based on DeepCDpred and MetaPSICOV constraints, the top 1 models of the proteins in the first three rows and the last row of Table 5.5 are shown in Figure 5.16, aligned with their respective experimental structures. It is interesting to see the results of 1vmb; specifically, the top 1 model predicted with DeepCDpred's constraints cannot overlap the experimental structure on the top α helix very well, but the top 1 model of this protein predicted with MetaPSICOV's constraints can, although the former is closer to the native structure.

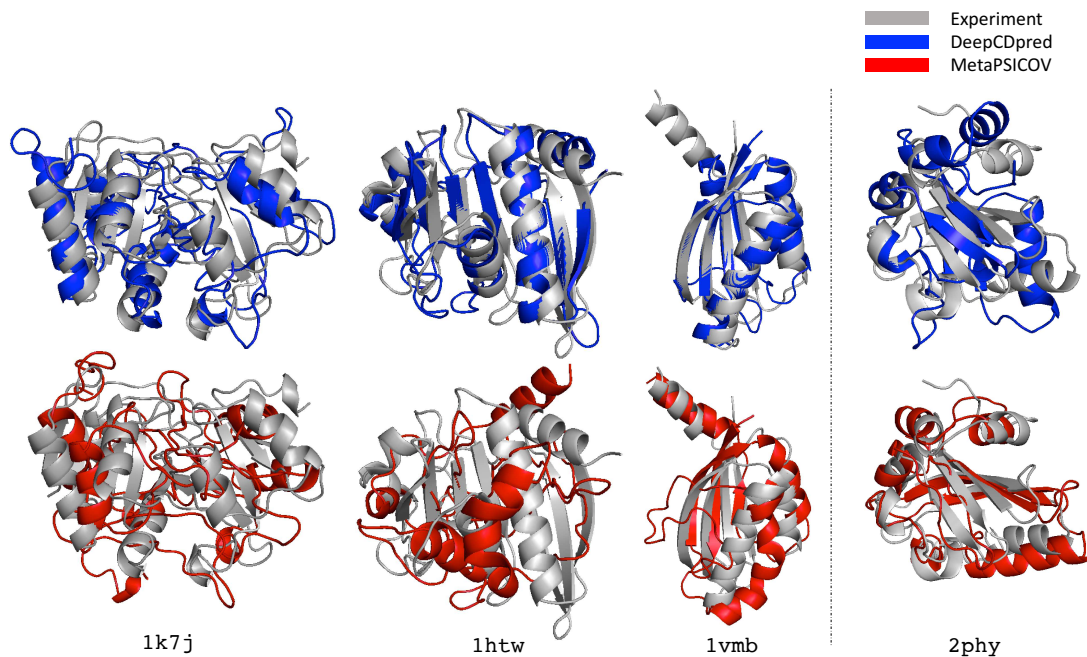


Figure 5.16. The top 1 models of four selected proteins in the test set are aligned to their respective experimental structures. The models on the left side of the dash line are better-predicted with DeepCDpred’s constraints, while, the model on the right side of the dash line is better-predicted with MetaPSICOV’s constraints. The proteins are selected according to Table 5.5.

After the outlier proteins in Figure 5.15a were removed from the test set of 108 proteins, there was still a significant difference ($p < 0.001$) between the TM-scores of the two groups (Figure 5.17, raw data of the boxplots can be found in Table B.1 of Appendix B).

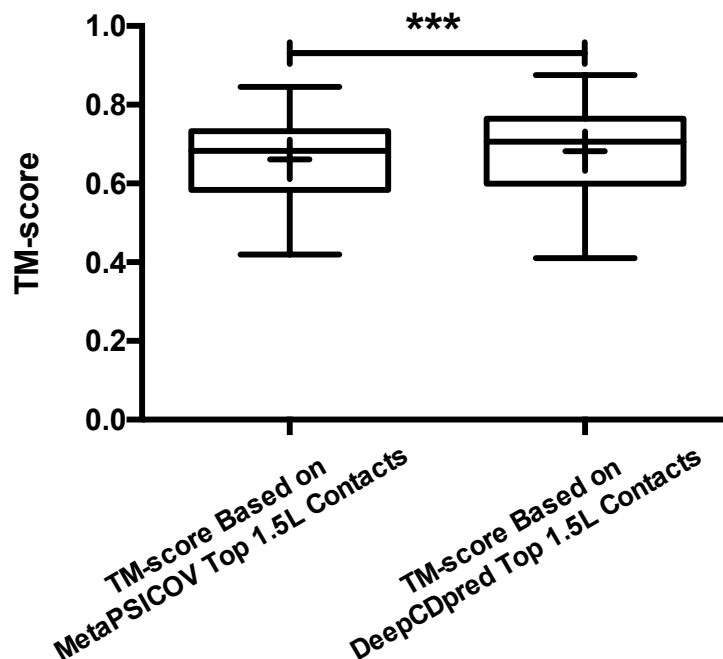


Figure 5.17. The comparison of structure prediction accuracies between MetaPSICOV and DeepCDpred after the outliers in Figure 5.15a were removed. A significant difference ($p < 0.001$) was found between the two groups of structure predictions by a paired t-test. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.15b.

5.6.3 Comparison of Structure Predictions Based on Predicted Contacts That Have the Same Contact Prediction Accuracy

As mentioned in the Model Development chapter (Chapter 4) and Subsection 5.5.6 of this chapter, in addition to selecting the top-ranked 1.5L contact predictions, another contact selection strategy based on a neural network score cut-off was also tested. In this section, the structure prediction results of both MetaPSICOV and DeepCDpred considering this method are introduced.

The relationship between the score cut-off and the contact prediction accuracy for both MetaPSICOV and DeepCDpred can be seen in Figure 5.18. The 108 proteins from the test set of DeepCDpred were used for the calculation. The step size of score cut-off was 0.02. As explained in Figure 5.18, two sets of equivalent-score cut-offs for DeepCDpred and MetaPSICOV were obtained based on the top ranked 1.5L contact predictions of the two algorithms, respectively: 0.40 for DeepCDpred and 0.56 for MetaPSICOV, and 0.26 for DeepCDpred and 0.40 for MetaPSICOV.

Based on the two pairs of score cut-offs, two sets of structure prediction comparisons were performed for the 108 proteins. In set 1, contact predictions from DeepCDpred with scores of greater than 0.40 were chosen, and contact predictions from MetaPSICOV with scores of greater than 0.56 were chosen, respectively. Also, in set 2, contact predictions from DeepCDpred with scores of greater than 0.26 were chosen, and contact predictions from MetaPSICOV with scores of greater than 0.40 were chosen, respectively.

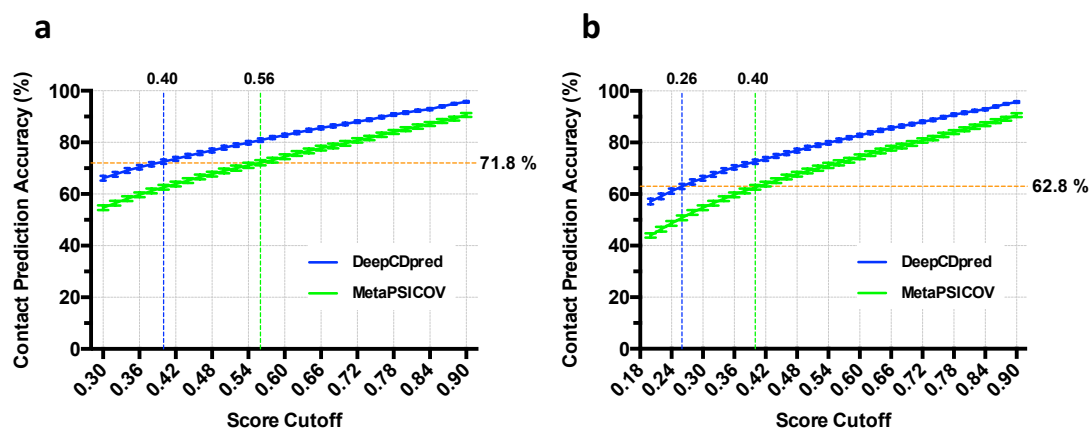


Figure 5.18. Finding the equivalent minimum scores for DeepCDpred and MetaPSICOV based on the same average accuracies of the top-ranked 1.5L contact predictions (71.8%, predicted by DeepCDpred in graph a; 62.8%, predicted by MetaPSICOV in graph b) of the 108 test proteins. To make sure the two contact score cut-offs of MetaPSICOV and DeepCDpred are equivalent, a horizontal line which represents a contact prediction accuracy (orange dashed lines in graph a and b) is drawn; the crossover points between this line and contact prediction accuracy curves of MetaPSICOV and DeepCDpred are determined. Two horizontal lines, $y = 71.8\%$ and $y = 62.8\%$, which represents the top-ranked 1.5L contact prediction accuracy of DeepCDpred and MetaPSICOV respectively, are shown in graph a and b. Thus, two pairs of equivalent score cut-offs are determined: 0.40 for DeepCDpred and 0.56 for MetaPSICOV having an expected accuracy of 71.8%, and 0.26 for DeepCDpred and 0.40 for MetaPSICOV having an expected accuracy of 61.8%.

The results of the first set are shown in Figure 5.19. In Figure 5.19a, there is a strong bias of the TM-score distribution toward the side of DeepCDpred. The boxplots in Figure 5.19b further support the above argument. The TM-score of the structures predicted with the top contact predictions from DeepCDpred are significantly higher than those with the top-ranked contact predictions from MetaPSICOV; a paired t-test indicates $p < 0.001$ (raw data of the boxplots can be found in Table B.2 of Appendix B).

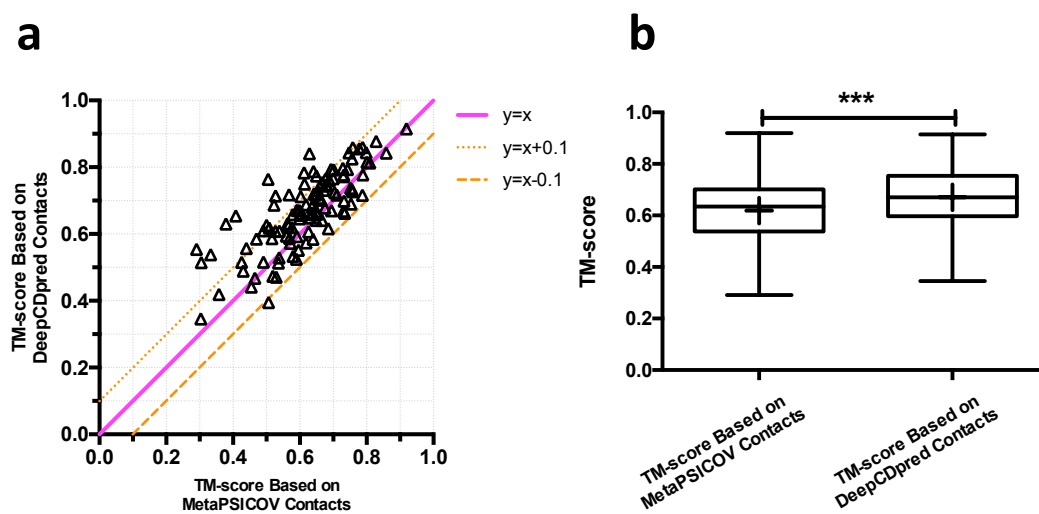


Figure 5.19. The comparison of structure prediction accuracies between feeding MetaPSICOV predicted contacts (score \geq 0.56) and feeding DeepCDpred predicted contacts (score \geq 0.40) into the same Rosetta *ab initio* protocol. Best predictions were picked out by the lowest Rosetta energy score. (a), scatter plot of the comparison; each triangle represents one protein in the test set and the majority of the proteins have better structure predictions when using DeepCDpred predicted contacts (score \geq 0.40) as constraints. (b), boxplots of the comparison in (a); a significant difference ($p < 0.001$) was found between the two groups of structure predictions. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.17.

Two lines of $y = x + 0.1$ and $y = x - 0.1$ are drawn on Figure 5.19a. The proteins located outside the area of $y = x + 0.1$ and $y = x - 0.1$ are considered to be outliers (81% of proteins are between the two lines versus 29% of proteins are outside the two lines in Figure 5.19a). Table 5.6 lists the outliers and the classifications of these proteins. Rows in the table are ranked by the TM-score difference from the highest to the lowest.

From the table, the percentages of the classes α , β , α/β , $\alpha+\beta$, coil and membrane of the proteins whose top 1 models are better-predicted with the contacts (score ≥ 0.40) from DeepCDpred are 5%, 22%, 0%, 68%, 5% and 0%, respectively. As compared with the

Table 5.6. The protein classes of the proteins whose structures are better-predicted with the contacts (score \geq 0.40) from DeepCDpred than with the contacts (score \geq 0.56) from MetaPSICOV, or vice versa. TM-score difference is defined as the TM-score of the top 1 model predicted with the contacts (score \geq 0.40) from DeepCDpred subtracting the TM-score of the top 1 model predicted with the contacts (score \geq 0.56) from MetaPSICOV for the same protein. Rows in the table are ranked by TM-score difference from the highest to the lowest.

PDB ID	TM-score Difference	TM-score of DeepCDpred	TM-score of MetaPSICOV	Protein Class
1dix	0.26	0.55	0.29	$\alpha+\beta$
1nps	0.25	0.76	0.51	β
1vfy	0.25	0.63	0.38	coil
1tif	0.24	0.65	0.41	$\alpha+\beta$
1mk0	0.21	0.84	0.63	$\alpha+\beta$
1aap	0.21	0.51	0.30	β
1dqg	0.21	0.54	0.33	β
1j11	0.18	0.71	0.53	$\alpha+\beta$
1bkr	0.17	0.78	0.61	α
1mug	0.17	0.69	0.52	$\alpha+\beta$
1cjw	0.15	0.72	0.57	$\alpha+\beta$
1atz	0.15	0.79	0.64	$\alpha+\beta$
1c44	0.14	0.75	0.61	$\alpha+\beta$
1c52	0.13	0.63	0.50	$\alpha+\beta$
1w0h	0.12	0.77	0.65	$\alpha+\beta$
1hfc	0.12	0.61	0.49	$\alpha+\beta$
1m4j	0.12	0.56	0.44	$\alpha+\beta$
1hxn	0.12	0.59	0.47	β
1p90	0.11	0.71	0.60	$\alpha+\beta$
1beb	0.11	0.62	0.51	β
1j3a	-0.11	0.40	0.51	α

protein class distributions in the training/validation set of DeepCDpred (25%, 9%, 20%, 44%, 1% and 1% for the six classes in the training/validation set), the percentages of α and α/β proteins are found to be lower; however, β and $\alpha+\beta$ proteins are over represented. A χ^2 test shows the two protein class distributions are significantly different ($p < 0.05$). As for the proteins whose structures are better-predicted with the contacts (score ≥ 0.56) from MetaPSICOV (the last row in the table), the set is too small to draw a conclusion.

There are eight proteins (1dix, 1vfy, 1tif, 1aap, 1dgg, 1hfc, 1m4j and 1hxn) for which the correct fold cannot be predicted (TM-score<0.5) with the contacts from MetaPSICOV, but can be predicted with the contacts from DeepCDpred; there is only one protein (1j3a) for which the correct fold can be predicted with the contacts from MetaPSICOV, but not with the contacts from DeepCDpred.

To give a direct view of the comparison of structure predictions based on DeepCDpred and MetaPSICOV constraints, the top 1 models (again, the model with the lowest Rosetta energy) of the proteins in the first three rows and the last row of Table 5.6 are shown in Figure 5.20, aligned to their respective experimental structures.

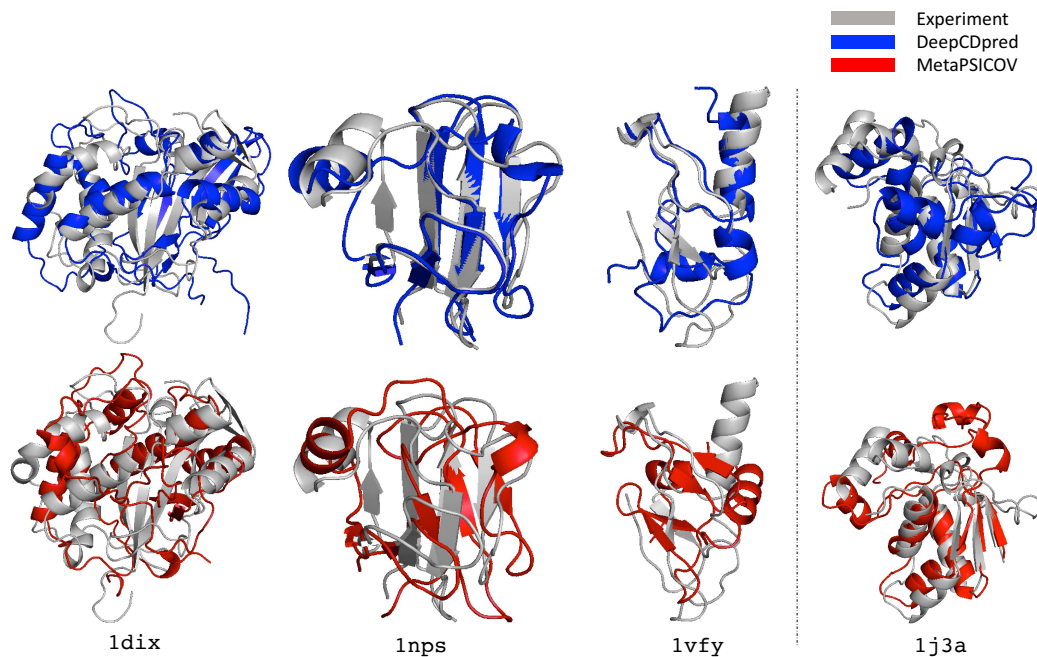


Figure 5.20. Superimpositions of the top 1 models from three proteins (left side of the dash line) for which the folds are better-predicted with the contacts ($\text{score} \geq 0.40$) from DeepCDpred than with the contacts ($\text{score} \geq 0.56$) from MetaPSICOV, and the top 1 model from one protein (right side of the dash line) that is more accurately predicted with the contacts ($\text{score} \geq 0.56$) from MetaPSICOV than with the contacts ($\text{score} \geq 0.40$) from DeepCDpred, with the respective experimental structures. The four proteins are selected according to Table 5.6.

After the outlier proteins in Figure 5.19a were removed from the test set of 108 proteins, there was still a significant difference ($p < 0.001$) presented between the TM-scores of the two groups (Figure 5.21, raw data of the boxplots can be found in Table B.3 of Appendix B).

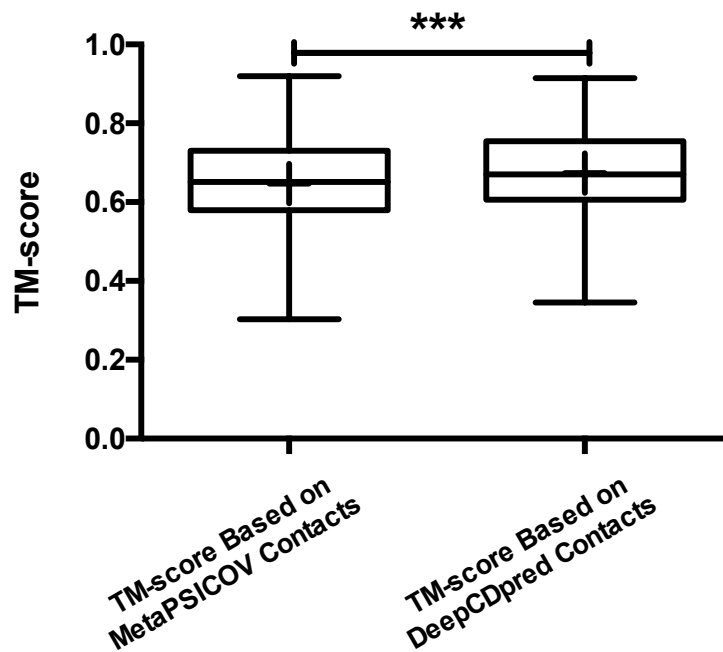


Figure 5.21. The comparison of structure prediction accuracies between MetaPSICOV and DeepCDpred after the outliers in Figure 5.19a were removed. A significant difference ($p < 0.001$) was found between the two groups of structure predictions by a paired t-test. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.19b.

The results of the second set of TM-score comparisons are shown in Figure 5.22. In Figure 5.22a, there is a strong bias of the TM-score distribution toward the side of DeepCDpred. The boxplots in Figure 5.22b further support the above argument. Notably, the TM-scores of the structures predicted with the top contact predictions from DeepCDpred are significantly higher than those predicted with the top-ranked contact predictions from MetaPSICOV, with a paired t-test indicating $p < 0.001$.

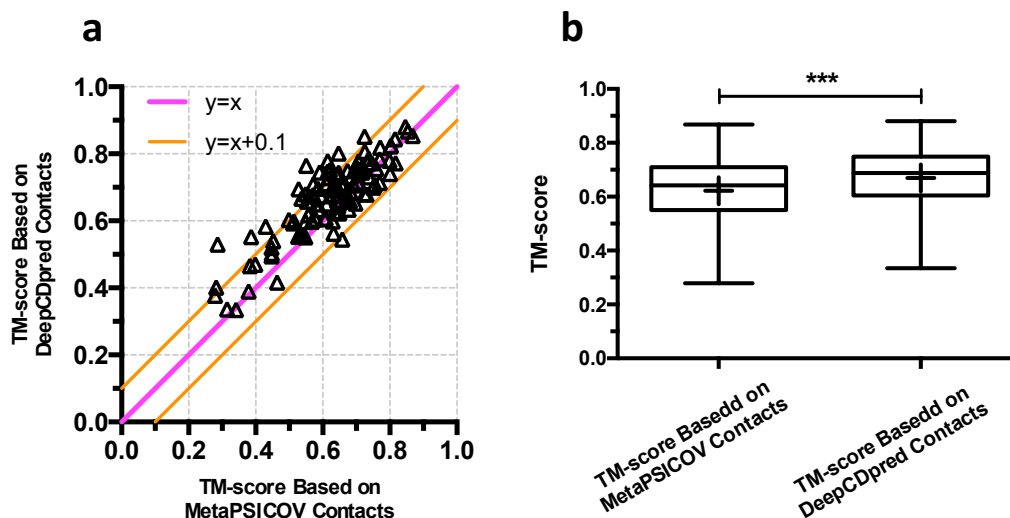


Figure 5.22. The comparison of structure prediction accuracies between feeding MetaPSICOV predicted contacts (score \geq 0.40) and feeding DeepCDpred predicted contacts (score \geq 0.26) to the same Rosetta *ab initio* protocol. Best predictions were picked out by the lowest Rosetta energy score. (a) scatter plot of the comparison; each triangle represents one protein in the test set and majority proteins have better structure prediction when using DeepCDpred predicted contacts (score \geq 0.26) as constraints. (b) boxplots of the comparison in (a); a significant difference ($p < 0.001$) was found between the two groups of structure predictions. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.21.

Two lines of $y = x + 0.1$ and $y = x - 0.1$ are drawn on Figure 5.22a. The proteins located outside the area of $y = x + 0.1$ and $y = x - 0.1$ are considered to be outliers (80% of proteins are between the two lines versus 20% of proteins are outside the two lines in Figure 5.22a). Table 5.7 lists the outliers and the classifications of these proteins.

Table 5.7. The classes of the proteins whose structures are better-predicted with the contacts (score \geq 0.26) from DeepCDpred than those with the contacts (score \geq 0.40) from MetaPSICOV, or vice versa. TM-score difference is defined as the TM-score of the top 1 model predicted with the contacts (score \geq 0.26) from DeepCDpred subtracting the TM-score of the top 1 model predicted with the contacts (score \geq 0.40) from MetaPSICOV for the same protein. Rows in the table are ranked by TM-score difference from the highest to the lowest.

PDB ID	TM-score Difference	TM-score of DeepCDpred	TM-score of MetaPSICOV	Protein Class
1gzc	0.24	0.53	0.29	$\alpha+\beta$
1jyh	0.21	0.76	0.55	$\alpha+\beta$
1g2r	0.17	0.78	0.61	α/β
1dqg	0.16	0.55	0.39	β
1gz2	0.16	0.69	0.53	$\alpha+\beta$
1ktg	0.15	0.80	0.65	$\alpha+\beta$
1m4j	0.15	0.58	0.43	$\alpha+\beta$
1ne2	0.15	0.74	0.59	α/β
1dmg	0.14	0.75	0.61	α/β
1qjp	0.14	0.71	0.57	membrane [#]
1hfc	0.13	0.67	0.54	α/β
1jl1	0.13	0.68	0.55	$\alpha+\beta$
1mk0	0.13	0.85	0.72	$\alpha+\beta$
2phy	0.13	0.75	0.62	$\alpha+\beta$
1kqr	0.12	0.40	0.28	β
1d1q	0.11	0.73	0.62	α/β
1fk5	0.11	0.70	0.59	α
1fvq	0.11	0.67	0.56	$\alpha+\beta$
1lo7	0.11	0.66	0.55	$\alpha+\beta$
1tif	0.11	0.68	0.57	$\alpha+\beta$
1htw	0.10	0.71	0.61	α/β
1i71	0.10	0.38	0.28	coil
1roa	0.10	0.60	0.50	$\alpha+\beta$
1vmb	0.10	0.74	0.64	$\alpha+\beta$
1wjx	-0.12	0.54	0.66	$\alpha+\beta$

[#]: 1qjp is a β out-membrane protein (<https://www.rcsb.org/pdb/explore/explore.do?structureId=1qjp>, last check: November 2018).

From the table, the percentages of the protein classes α , β , α/β , $\alpha+\beta$, coil and membrane of the proteins whose top 1 models are better-predicted with the contacts (score \geq 0.26)

from DeepCDpred are 4%, 8%, 25%, 55%, 4% and 4%, respectively. By comparing these values with the protein class distributions in the training/validation set of DeepCDpred (25%, 9%, 20%, 44%, 1% and 1% for the six classes in the training/validation set), it is seen that the α protein is under-represented; both the α/β and $\alpha+\beta$ proteins are slightly over-represented. A χ^2 test shows the two protein class distributions are significantly different ($p = 0.0078 < 0.05$). As for the proteins whose structures are better-predicted with the contacts (score ≥ 0.40) from MetaPSICOV (the last row in the table), the set is too small to draw a conclusion.

In this case, there are 3 proteins (1gzc, 1dqg and 1m4j) whose folds can not be predicted correctly (TM-score <0.5) with MetaPSICOV contacts, but can be predicted with DeepCDpred contacts. Conversely, there are no proteins whose folds can be predicted correctly with MetaPSICOV contacts, but cannot be with DeepCDpred contacts.

To attain a direct view of the comparison of structure predictions based on DeepCDpred and MetaPSICOV constraints, the top 1 models of the proteins in the first three rows and the last row of Table 5.7 are shown in Figure 5.23, aligned with their respective experimental structures.

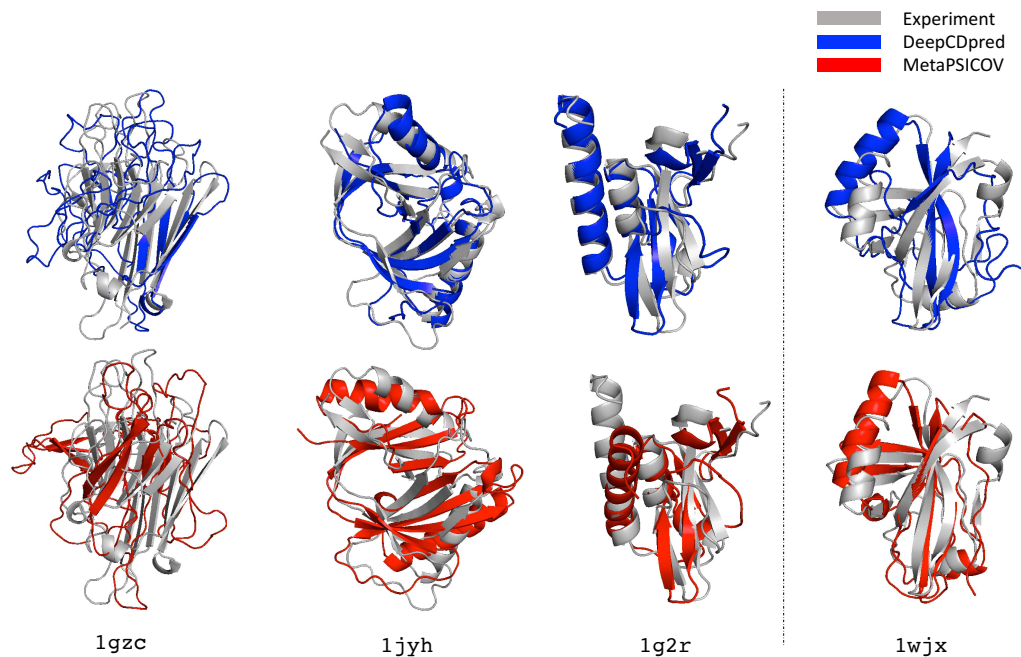


Figure 5.23. Superimpositions of the top 1 models of three proteins (left side of the dash line) which are more accurately predicted with the contacts ($\text{score} > 0.26$) from DeepCDpred than with the contacts ($\text{score} \geq 0.40$) from MetaPSICOV, and the top 1 model of one protein (right side from the dash line) that is more accurately predicted with the contacts ($\text{score} \geq 0.40$) from MetaPSICOV than with the contacts ($\text{score} \geq 0.26$) from DeepCDpred with the respective experimental structures. The four proteins are selected according to Table 5.7.

After the outlier proteins in Figure 5.22a were removed from the test set of 108 proteins, it is clear that there was still a significant difference ($p < 0.001$) between the TM-scores of the two groups (Figure 5.24).

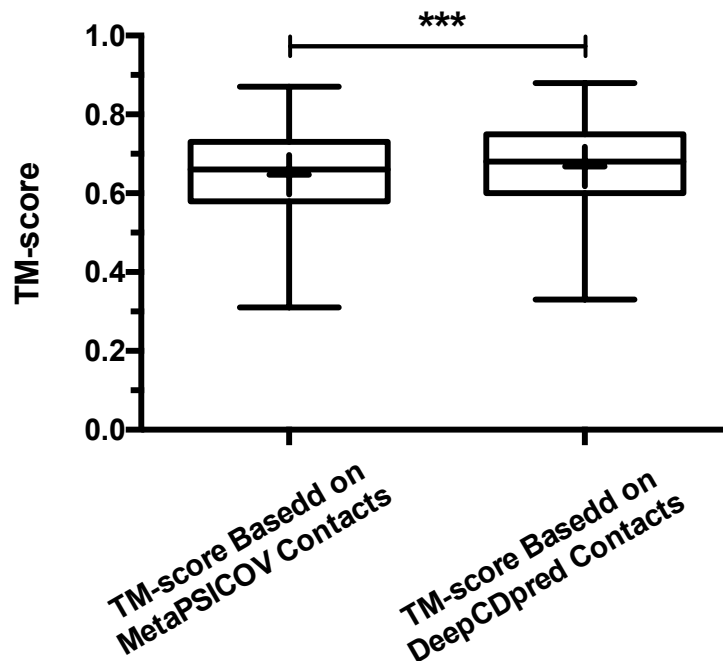


Figure 5.24. The comparison of structure prediction accuracies between MetaPSICOV and DeepCDpred after the outliers in Figure 5.22a were removed. A significant difference ($p < 0.001$) was found between the two groups of structure predictions by a paired t-test. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.22b.

5.6.4 Summary of Comparisons of Structure Predictions Based on Contacts Predicted by MetaPSICOV and DeepCDpred

In the above two subsections, three groups of comparisons of structure predictions were made between MetaPSICOV and DeepCDpred for the proteins in the test set based on different selection strategies of contact predictions. For the contacts predicted by each method (MetaPSICOV or DeepCDpred), it is also interesting to compare the three selection ways to find out that which of them leads to the best structure predictions. It's

worth noting that the results shown here (Figure 5.25) are taken from the above two subsections.

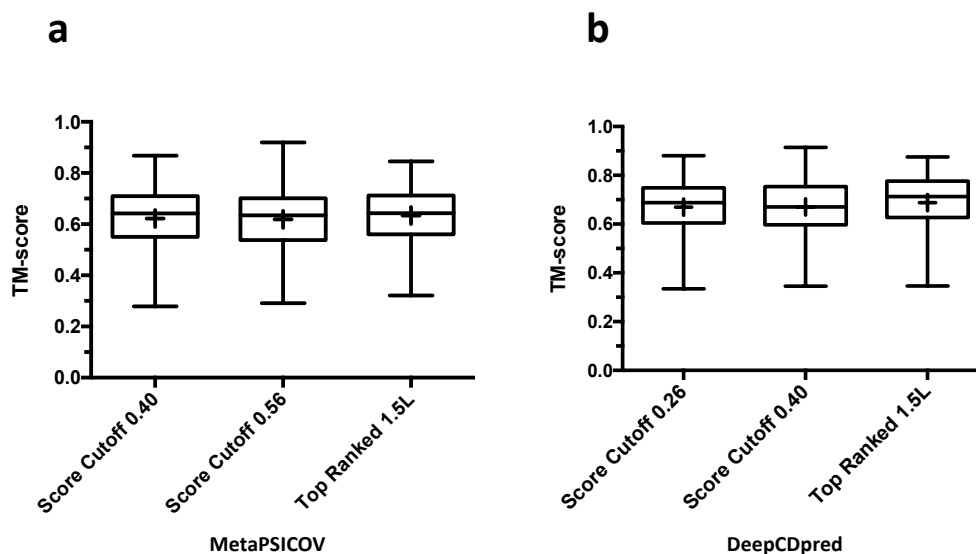


Figure 5.25. Comparisons of the quality of structure predictions based on the three contact selection strategies for each of the two algorithms compared. (a) the comparisons based on the contact predictions from MetaPSICOV; three boxplots represent the TM-score distributions of the top 1 models of the 108 proteins based on the three contact selection strategies (two are score cut-offs based; the score cut-offs are 0.40 and 0.56, respectively, and the third is the top-ranked 1.5L). (b) the comparisons among the contact predictions from DeepCDpred based on the three contact selection strategies (two are score cut-offs based; the score cut-offs are 0.26 and 0.40, respectively, and the third is the top-ranked 1.5L).

Figure 5.25a and Figure 5.25b show the comparison results of structure predictions based on the contact predictions from MetaPSICOV and DeepCDpred, respectively. In Figure 5.25a, the average TM-scores of the three groups are 0.622, 0.619, and 0.634, respectively; a one-way ANOVA test shows there is no significant difference ($p > 0.05$) among the three averages. In Figure 5.25b, the average TM-scores of the three groups are 0.669,

0.670 and 0.688, respectively; again, a one-way ANOVA test indicates there is no significant difference ($p > 0.05$) among the three averages.

5.6.5 Comparisons of Structure Predictions Between Using the Top-ranked 1.5L DeepCDpred Contacts and Using the Combination of the Top-ranked 1.5L DeepCDpred Contacts & Score Cut-off Selected DeepCDpred Distances

After comprehensive comparisons of the quality of protein structure predictions between using contact predictions from DeepCDpred and from MetaPSICOV, it is also important to compare the structure predictions based on the contact predictions only as well as based on both contact and distance predictions from DeepCDpred. A unique feature of DeepCDpred is the distance prediction; if the predicted distances make no contribution to the structure prediction, the importance of this point will be greatly reduced.

As mentioned in Subsection 4.3.4 of the previous chapter, there are four ways to combine the contact and distance predictions. However, the results of only two of them will be introduced in this and the next two subsections. The two combinations of contact and distance predictions are: (1) the top-ranked 1.5L contacts plus the score cut-off based distances; (2) the score cut-off based contacts plus the score cut-off based distances. This

subsection introduces the results of the structure predictions of the test proteins based on the first combination, as compared with the structure predictions based on the top-ranked 1.5L DeepCDpred contacts only, shown in Figure 5.26. Here, the contact/distance predictions in each of the distance bins were accepted based on a minimum neural network output score of 0.60. The structure prediction result based on the second combination will be introduced in the next section.

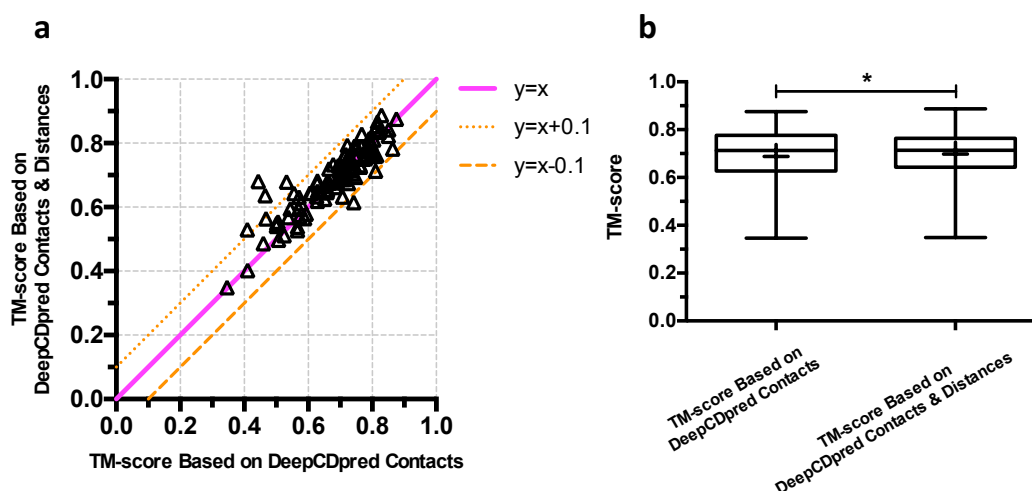


Figure 5.26. The comparison of the structure prediction accuracies between feeding the top 1.5L DeepCDpred predicted contacts and feeding the combination of the top 1.5L DeepCDpred predicted contacts & neural network score selected distances to the same Rosetta *ab initio* protocol. The predictions were picked out by the lowest Rosetta energy score. (a) scatter plot of the comparison; each triangle represents one protein in the test set and the majority of proteins have better structure prediction when using the top 1.5L DeepCDpred predicted contacts as constraints. (b) boxplots of the comparison in (a); a significant difference ($p < 0.001$) was found between the two groups of structure predictions. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.25.

Figure 5.26a shows a bias of the TM-score distribution toward the y -axis side, which means the quality of the best structure predictions based on both the top ranked 1.5L contacts and score cut-off selected distances from DeepCDpred are generally better than

those based on only the top ranked 1.5L contacts from DeepCDpred. The boxplots in Figure 5.26b further support the above argument. The TM-scores of the top 1 models predicted with both the contact and distance predictions from DeepCDpred are significantly higher than those with the contact predictions only from DeepCDpred, as indicated by a paired t-test ($p < 0.001$).

Again, the two lines of $y = x + 0.1$ and $y = x - 0.1$ are drawn on Figure 5.26a. The proteins located outside the area of $y = x + 0.1$ and $y = x - 0.1$ are considered to be outliers (95% of proteins are between the two lines versus 5% of proteins are outside the two lines in Figure 5.26a). Table 5.8 lists the outliers and their classifications.

Table 5.8. The classes of the proteins whose structures are better-predicted with the top-ranked 1.5L contacts plus distances from DeepCDpred than with the top-ranked 1.5L contacts from DeepCDpred only, or vice versa. TM-score difference is defined as the TM-score of the top 1 model predicted with the top-ranked 1.5L contacts plus distances from DeepCDpred subtracting the TM-score of the top 1 model predicted with the top-ranked 1.5L contacts from DeepCDpred only for the same protein. Rows in the table are ranked by TM-score difference from the highest to the lowest.

PDB ID	TM-score Difference	TM-score of DeepCDpred with Contacts & Distances	TM-score of DeepCDpred with Contacts	Protein Class
2phy	0.24	0.68	0.44	$\alpha + \beta$
1tif	0.17	0.64	0.47	$\alpha + \beta$
1fk5	0.15	0.68	0.53	α
1ej8	0.11	0.53	0.42	β
1c9o	-0.13	0.61	0.74	β

From the table, the percentages of the protein classes α , β , α/β , $\alpha + \beta$, coil and membrane of the proteins whose structures are better-predicted with the contacts & distances from

DeepCDpred are 25%, 25%, 0%, 50%, 0% and 0%, respectively. As compared with the protein class distributions in the training/validation set of DeepCDpred (25%, 9%, 20%, 44%, 1% and 1% for the six classes in the training/validation set), α/β is lower again but the set size is too small (only four proteins) to draw a conclusion. As for the proteins whose structures are better-predicted with the contacts from DeepCDpred only (last row in the table), the set is also too small to draw a conclusion.

There are 3 proteins (2phy, 1tif and 1ej8) for which the correct fold cannot be predicted (TM-score<0.5) with the predicted contacts only from DeepCDpred, but can be predicted with both the contacts and distances from DeepCDpred. There is no protein for which the correct fold can be predicted with DeepCDpred contacts only but cannot with DeepCDpred contacts & distances.

To give a direct view of the comparison of structure predictions based on DeepCDpred contact constraints and DeepCDpred contact & distance constraints, the top 1 models of the proteins in the first three rows and the last row of Table 5.8 are shown in Figure 5.27, aligned to their respective experimental structures.

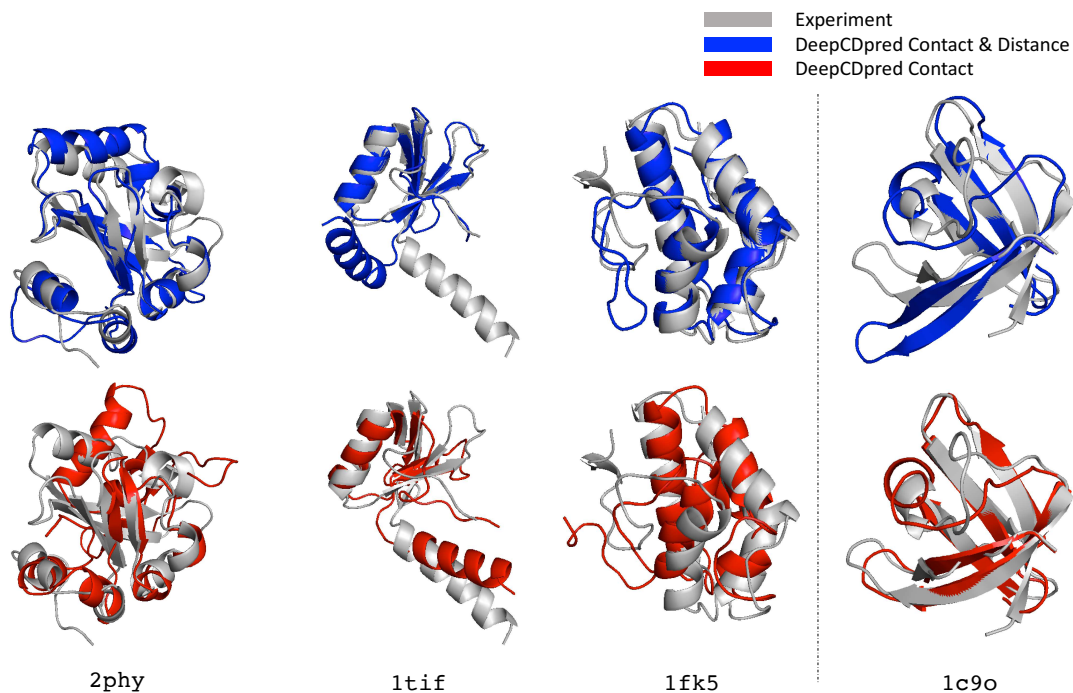


Figure 5.27. Superimpositions of the top 1 models of three proteins (left side of the dash line) which are more accurately predicted with the top-ranked 1.5L contacts plus distances from DeepCDpred than with only the top-ranked 1.5L contacts from DeepCDpred, and top 1 model (right side of the dash line) of one protein that is more accurately predicted with only the top-ranked 1.5L contacts than with the contacts plus distances, with the respective experimental structures. pdb ids are selected according to Table 5.8.

After the outlier proteins in Figure 5.26a were removed from the test set of 108 proteins, there was no significant difference ($p = 0.16 > 0.05$) between the TM-scores of the two groups (Figure 5.28), which means the outliers contribute to the significance of the statistical analysis in Figure 5.26.

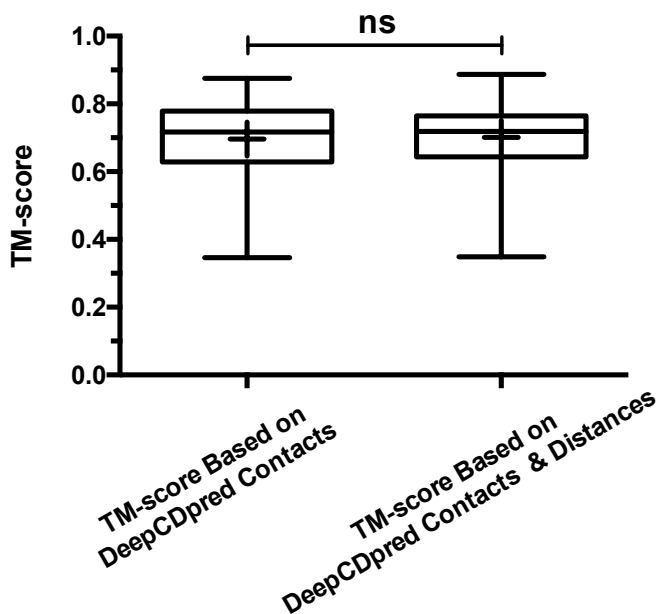


Figure 5.28. The comparison of structure prediction accuracies between DeepCDpred contacts only and DeepCDpred contacts & distances after the outliers in Figure 5.26a were removed. No significant difference (ns, $p > 0.05$) was found between the two groups of structure predictions by a paired t-test. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.26b.

5.6.6 Comparison of Structure Predictions between Using DeepCDpred Contacts Selected by Score Cut-off and Using Both DeepCDpred Contacts & Distances Selected by Score Cut-off

This section introduces the results of the structure prediction of the proteins in the test set based on both the contact and distance predictions from DeepCDpred. The contacts are selected based on a minimum neural network score of 0.40, and predictions in each distance bin are accepted based on a minimum neural network score of 0.60. The reason for choosing the minimum score of 0.40 instead of 0.26 for the contacts is that the former

achieves slightly higher TM-scores of structure predictions on average (0.670 vs. 0.669), though the difference is not significant, as shown in Figure 5.25.

Structure predictions of the proteins in the test set by using only DeepCDpred contact predictions with a score cut-off of 0.40 have already been illustrated in Figure 5.19 and Figure 5.25. They are compared to the structure predictions of the same proteins based on the contacts and distances predicted by DeepCDpred, shown in Figure 5.29. Figure 5.29a shows a strong bias of the TM-score distribution toward the contact & distance side. The boxplots in Figure 5.29b further support the above argument. The TM-scores of the structures predicted with the contact & distance predictions from DeepCDpred are significantly higher than those achieved with the contact predictions only from DeepCDpred, as determined by a paired t-test ($p < 0.001$).

The outliers of the TM-scores of the structure predictions of both with and without distance constraints are analysed. The method is the same as the one used in the previous subsections (Subsection 5.6.2 and Subsection 5.6.3). Two lines of $y = x + 0.1$ and $y = x - 0.1$ are drawn on Figure 5.29a. The proteins located outside the area of $y = x + 0.1$ and $y = x - 0.1$ are considered to be outliers (94% of proteins are between the two lines versus 6% of proteins are outside the two lines in Figure 5.29a). Proteins in the upper-left area are significantly better-predicted with both the contacts and distances from DeepCDpred; proteins in the bottom-right area are significantly better-predicted with contacts only. Table 5.9 lists the outliers and the classifications of these proteins.

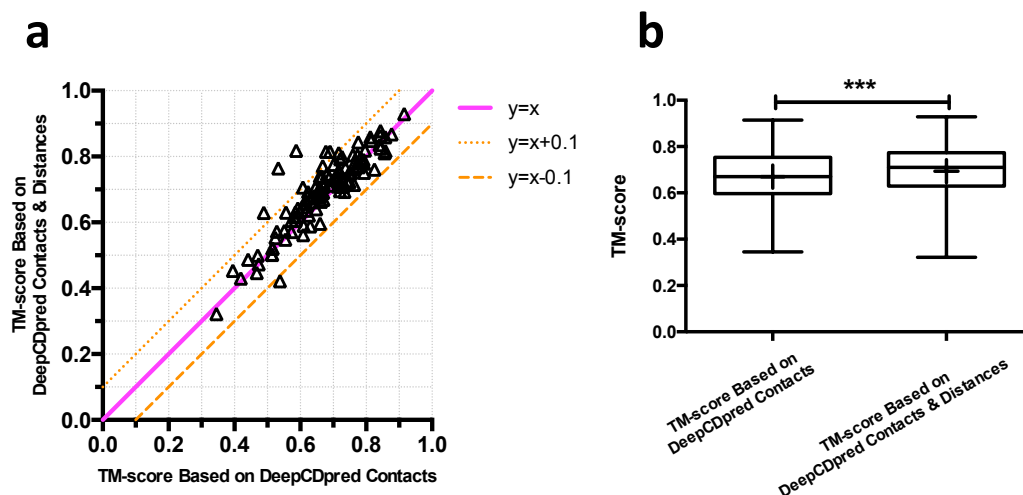


Figure 5.29. The comparison of structure prediction accuracies between feeding the contacts ($\text{score} \geq 0.40$) from DeepCDpred and feeding the contacts ($\text{score} \geq 0.40$) & distances from DeepCDpred to the same Rosetta *ab initio* protocol. The predictions were picked out by the lowest Rosetta energy score. (a) scatter plot of the comparison; each triangle represents one protein in the test set and the majority of the proteins have better structure prediction when using the contacts ($\text{score} \geq 0.40$) & distances from DeepCDpred as constraints. (b) boxplots of the comparison in (a); a significant difference ($p < 0.001$) was found between the two groups of structure predictions via a paired t-test. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.28.

From the table, the percentages of the protein classes α , β , α/β , $\alpha+\beta$, coil and membrane of the proteins whose structures are better-predicted with the contacts & distances from DeepCDpred are 17%, 33%, 0%, 50%, 0% and 0%, respectively. As compared with the protein class distributions in the training/validation set of DeepCDpred (25%, 9%, 20%, 44%, 1% and 1% for the six classes in the training/validation set), both the percentages of the α and α/β proteins are lower, especially in the case of the latter; conversely, that of β protein is higher. However, since the set size is small, it is hard to draw a conclusion. As for the proteins whose structures are better-predicted with only the contacts from DeepCDpred (last row in the table), the set is also too small to draw a conclusion. There

Table 5.9. The classes of the proteins whose structures are significantly better-predicted with the contacts (score \geq 0.40) & distances from DeepCDpred than with the contacts (score \geq 0.40) from DeepCDpred, or vice versa. TM-score difference is defined as the TM-score of the top 1 model predicted with the contacts & distances from DeepCDpred subtracting the TM-score of the top 1 model predicted with the contacts from DeepCDpred for the same protein. “significantly” here means the absolute value of TM-score difference \geq 0.1 (outside the area formed between $y = x + 0.1$ and $y = x - 0.1$ in Figure 5.29a. Rows in the table are ranked by TM-score difference from the highest to the lowest.

PDB ID	TM-score Difference	TM-score of DeepCDpred with Contacts & Distances	TM-score of DeepCDpred with Contacts	Protein Class
1kq6	0.23	0.76	0.53	$\alpha+\beta$
1avs	0.23	0.82	0.59	α
1d4o	0.14	0.63	0.49	α/β
1jos	0.14	0.81	0.68	$\alpha+\beta$
1vjk	0.12	0.81	0.69	$\alpha+\beta$
1chd	0.10	0.77	0.67	α/β
1dqq	-0.12	0.42	0.54	β

is one protein (1d4o) for which the correct fold can not be predicted (TM-score $<$ 0.5) with the predicted contacts from DeepCDpred, but can be predicted with both the contacts and distances from DeepCDpred. There is one protein (1dqq) for which the correct fold can be predicted with the contacts from DeepCDpred, but can not be predicted with both the contacts and distances from DeepCDpred. To explore the reason why additional constraints make the structure prediction of 1dqq worse, the prediction accuracy of the distance constraints used in the structure prediction are calculated. The result is shown in Table 5.10. From the table, the accuracies of the distance constraint predictions of the three distance bins are lower than the average accuracies of the 108 test proteins, as well as lower than the accuracies of the three proteins appearing on the top three rows of

Table 5.9.

Table 5.10. Comparison of the distance prediction accuracy between proteins which appear in the top three rows and the last row of Table 5.9. In the last row, the average means the average accuracy of distance predictions of the 108 test proteins by DeepCDpred .

PDB ID	8-13Å	13-18Å	18-23Å
1avs	80.0%	79.6%	62.9%
1k6g	76.5%	73.9%	59.5%
1d4o	80.9%	77.5%	61.4%
1dqg	52.5%	58.7%	44.4%
Average	73.6%	65.5%	60.6%

To give a direct view of the comparison of structure predictions based on DeepCDpred contact and DeepCDpred contact & distance constraints, the top 1 models of the proteins in the first three rows and the last row of Table 5.9 are shown in Figure 5.30, aligned with their respective experimental structures.

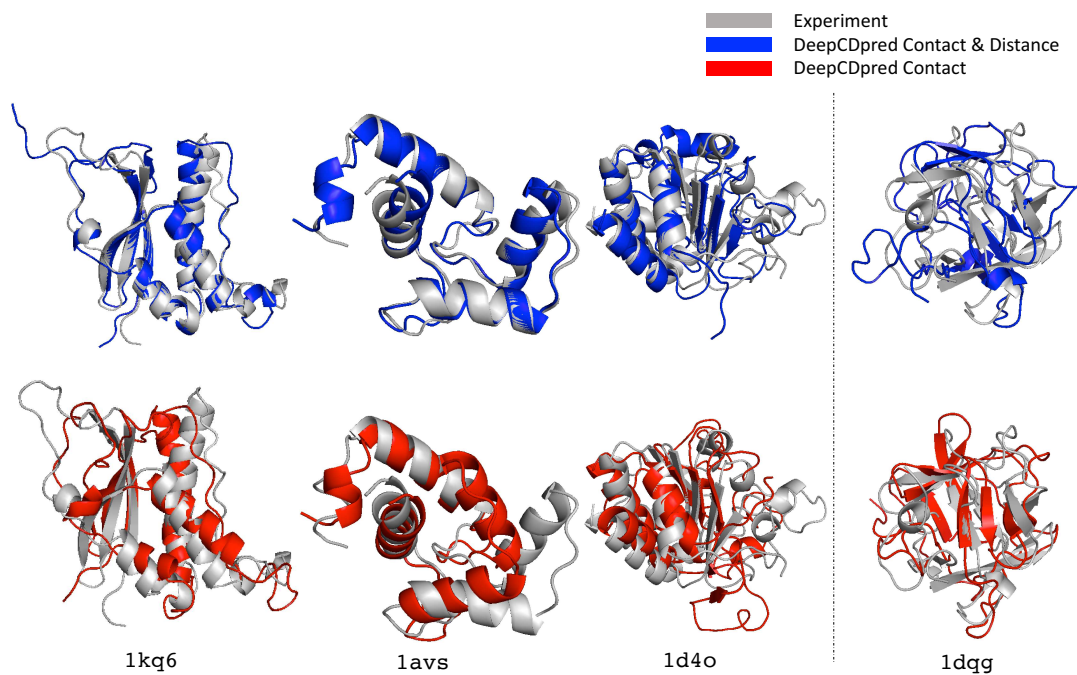


Figure 5.30. Superimpositions of the top 1 model of three proteins (left side of the dash line) which are more accurately predicted with contacts (score \geq 0.40) plus distances from DeepCDpred than with contacts (score \geq 0.40) from DeepCDpred, and the top 1 model of one protein (right side of the dash line) that is more accurately predicted with the contacts than with the contacts plus distances, with the respective experimental structures. pdb ids are selected according to Table 5.9.

Notably, after the outlier proteins in Figure 5.29a were removed from the test set of 108 proteins, there was still a significant difference ($p < 0.001$) between the TM-scores of the two groups (Figure 5.31).

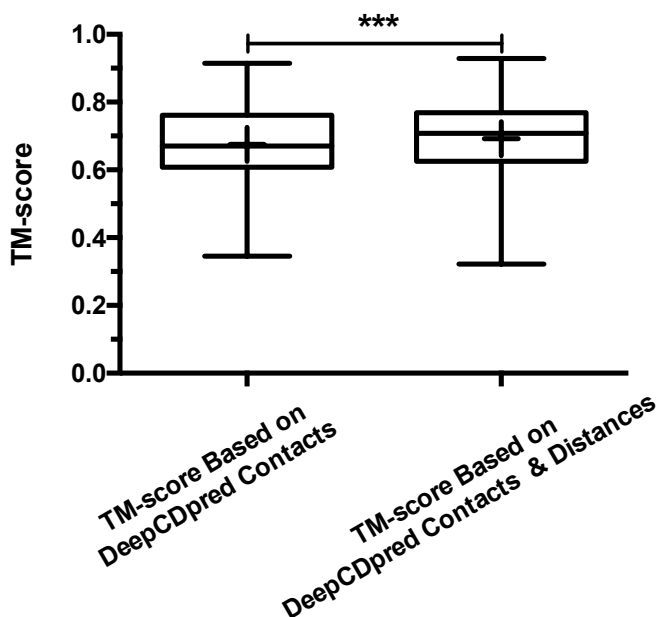


Figure 5.31. The comparison of structure prediction accuracies between DeepCDpred contacts only and DeepCDpred contacts & distances after the outliers in Figure 5.29a were removed. A significant difference ($p < 0.001$) was found between the two groups of structure predictions by a paired t-test. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.29b.

5.6.7 Summary of Structure Prediction Based on Contacts and Distances from DeepCDpred

In this subsection, the structures predicted by using the combination of the top 1.5L DeepCDpred contact constraints and score cut-off (score cut-off of ≥ 0.6) selected DeepCDpred distance constraints are compared to the ones predicted by using the same distance constraints but different DeepCDpred contact constraints selected with scores of ≥ 0.4 . Meanwhile, the distributions of the TM-scores of the predicted structures for the protein chains in the test set, as well as the relationships between the TM-score and the corresponding Nf value, which measures the quality of the MSA of a target protein, based on the two constraint selections methods, are introduced.

The comparison of the TM-scores of the structure predictions with the contact and distance constraints predicted by DeepCDpred based on the two constraint selection methods is shown in Figure 5.32. In fact, the two boxplots in the figure are just copied from the right boxplot of Figure 5.26b and the right boxplot of Figure 5.29b. A paired t-test shows there is no significant difference between the two groups of TM-scores ($p > 0.05$). The averages in the two boxplots are 0.697 and 0.694, respectively.

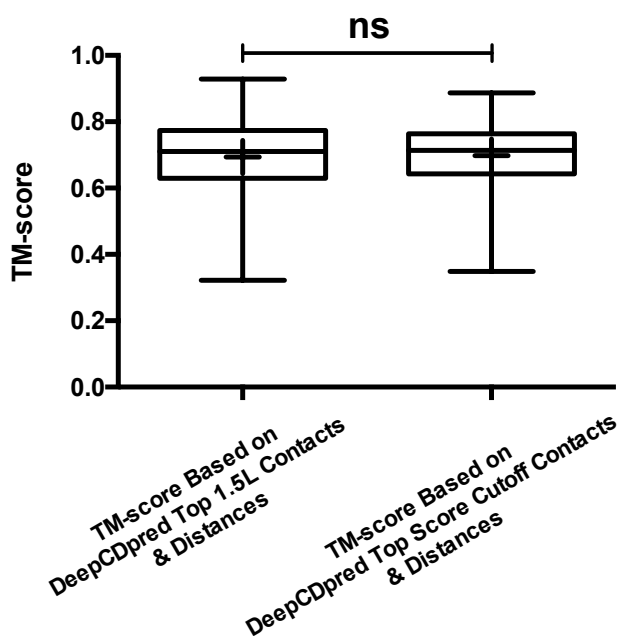


Figure 5.32. Comparison of TM-scores of the model (selected by the lowest Rosetta energy score) for each of the 108 test proteins predicted based on the two combinations of DeepCDpred contacts & distances constraints. The first combination is the top-ranked 1.5L contacts plus distances (selected by a minimum neural network score of 0.6 for all of the three bins) (left box), and the second combination is the contacts selected by a minimum neural network score of 0.4 plus the same distances (right box). A paired t-test shows there is no significant difference between the two groups of TM-scores. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.31.

The histograms of the TM-scores of the two groups are shown in Figure 5.33a and Figure 5.33b. There are five proteins predicted by the first constraint selection method that have incorrect folds with the experimental structures (Figure 5.33a), which are compared to eight wrongly predicted ones by the second constraint selection method (Figure 5.33b). The numbers displayed on the bars are the counts of proteins in each TM-score bin.

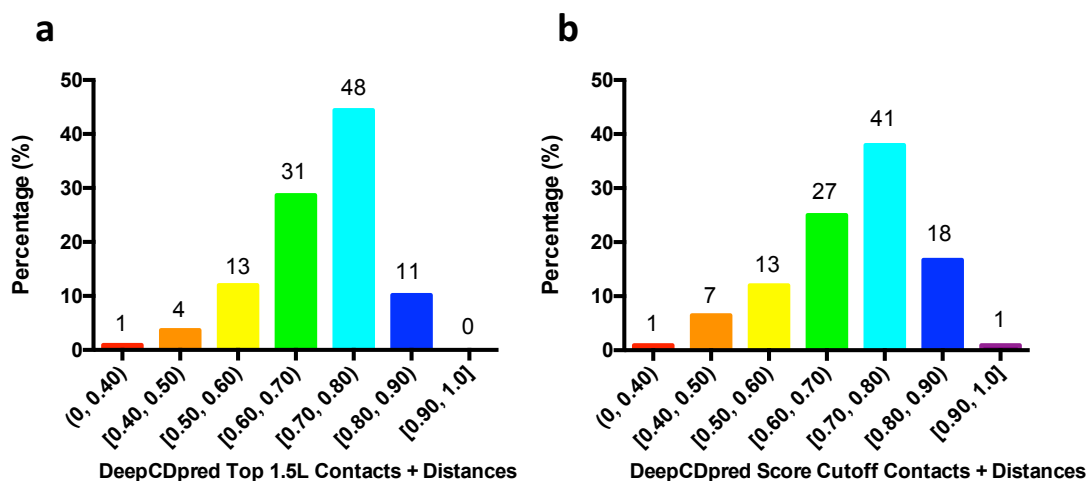


Figure 5.33. Distribution of the TM-scores of the top 1 model (selected by the lowest Rosetta energy score) predicted based on the two combinations of DeepCDpred contact plus distance constraints. The numbers displayed above the bars are the counts of proteins in each TM-score bin.

The relationship between the Nf value and the TM-score value of the top 1 model for each protein against the experimental structure in the test set based on the two selection methods are also investigated. It is worth noting that the TM-score here is the real TM-score, not the predicted one (introduced in the next section). The result is shown in Figure 5.34. In Figure 5.34a, 100% of the predictions of the proteins in the test set have the correct fold when the Nf values of these proteins >64 . When $Nf \leq 64$, 82.1% of the predictions still have the same folds with the experimental ones. In Figure 5.34b, when $Nf > 64$, 97.5% of the models have the same folds with the experimentally solved ones. It is also noteworthy that 78.5% of the top 1 models have the correct folds even when $Nf \leq 64$.

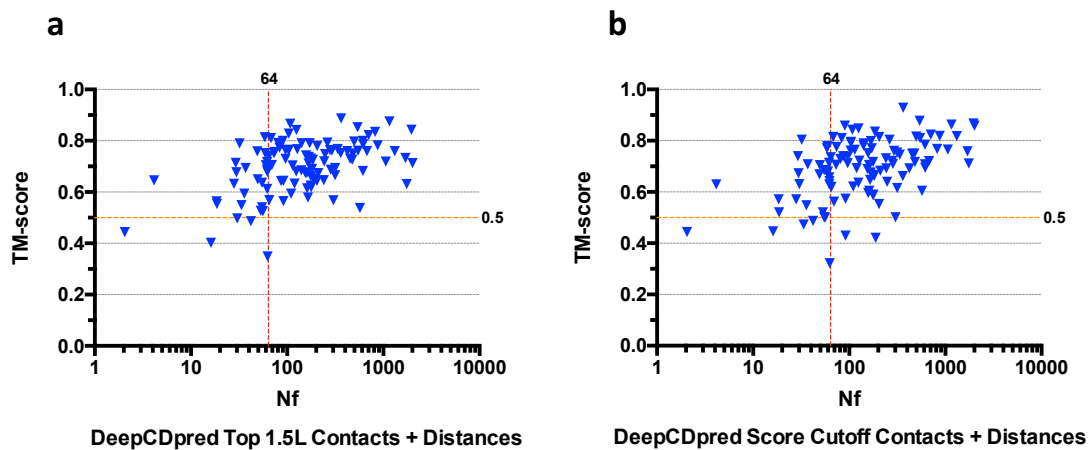


Figure 5.34. Real TM-score of the top 1 model (selected by the lowest Rosetta energy score) versus the Nf for each of 108 proteins in the test set. As explained in Subsection 2.8.5 of Chapter 2, TM-score of 0.5 was chosen as the cut-off to between correct folds and incorrect folds. Constraints of contacts/distances predicted by DeepCDpred were fed into the Rosetta Abinitio protocol.

5.7 TM-score Predictions

Taking the model with the lowest Rosetta energy score for each test protein yielded 108 structures to test the TM-score prediction network. The structure predictions used DeepCDpred predicted contacts and distances based on score cut-offs of 0.4 and 0.6, respectively. The real TM-scores of the top 1 models were already shown in Figure 5.32b. The ‘real TM-score’ refers to the structure comparison between the predicted top 1 model and the corresponding experimental structure.

Comparisons of the predicted TM-score and the real TM-score of the structure predictions are shown in Figure 5.35a. It is observed that 63% of the predicted TM-scores are in the range of the real TM-score ± 0.1 . The correlation coefficient between the predicted TM-score and real TM-score is adjusted- $R^2 = 0.46$. Although there is a significant difference between the averages of the predicted and the real TM-scores (by a paired t-test, $p < 0.001$), the difference of the two averages is only 0.07, which is less than 0.1 (Figure 5.35b). Specifically, the averages of the predicted TM-scores and the real TM-scores are 0.63 and 0.70, respectively.

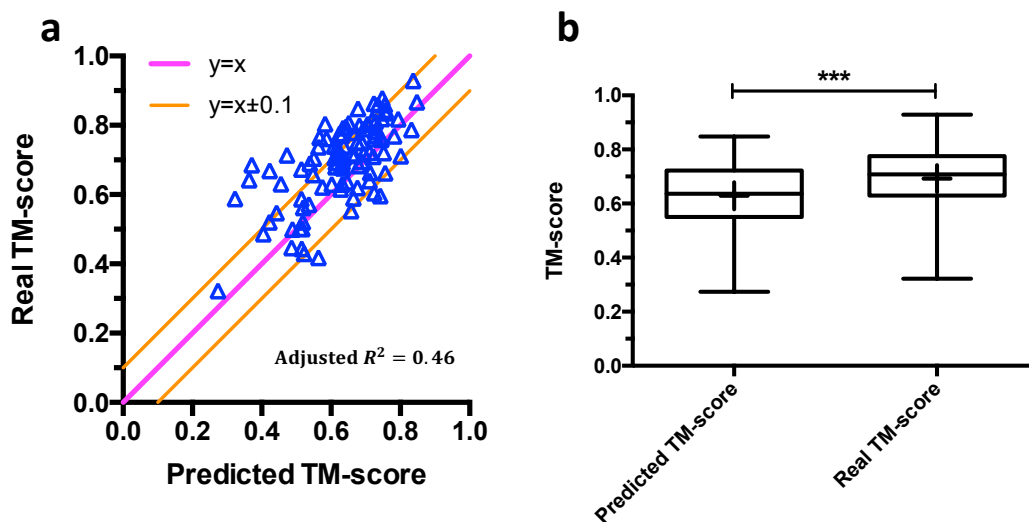


Figure 5.35. Predicted TM-scores versus real TM-scores from the structure predictions based on the contact and distance constraints from DeepCDpred for the proteins in the test set of DeepCDpred. Constraints of contacts and distances fed into the Rosetta *ab initio* protocol were predicted by DeepCDpred. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.32.

The outliers of the proteins whose top 1 models' TM-scores are poorly predicted with the network model, are determined by the two lines of $y = x \pm 0.1$. The pdb ids of them are listed in Table 5.11, together with the protein classes.

The percentages of the protein classes α , β , α/β , $\alpha+\beta$, coil and membrane of the proteins whose top 1 models' TM-score predictions are under-estimated (triangles at the upper left of $y = x + 0.1$) are 12%, 8%, 16%, 60%, 4% and 0%, respectively. As compared with the protein class distributions in the training/validation set of DeepCDpred (25%, 9%, 20%, 44%, 1% and 1% for the six classes in the training/validation set), the percentage of α proteins is found to be lower; but that of $\alpha+\beta$ is higher. A χ^2 test shows the two protein class distributions have no significant difference ($p = 0.07 > 0.05$). As for the proteins

Table 5.11. The classes of the proteins whose top 1 models' TM-scores are poorly predicted with the TM-score prediction network model. The top 1 models are predicted with the contacts and distances from DeepCDpred. TM-score difference is defined as the real TM-score subtracting the predicted TM-score for the same protein. The outliers of proteins located outside the two lines of $y = x \pm 0.1$ in Figure 5.35a are listed in this table. Rows in the table are ranked by the TM-score difference from the highest to the lowest.

PDB ID	TM-score Difference	Predicted TM-score	Real TM-score	Protein Class
1tif	0.31	0.37	0.69	$\alpha+\beta$
1fk5	0.28	0.36	0.64	α
1vfy	0.27	0.32	0.59	coil
1g2r	0.25	0.42	0.67	α/β
1m8a	0.24	0.47	0.71	$\alpha+\beta$
1jo0	0.22	0.58	0.80	$\alpha+\beta$
1kq6	0.20	0.57	0.76	$\alpha+\beta$
1hxn	0.18	0.46	0.63	β
1k6k	0.17	0.68	0.85	α
1nps	0.17	0.59	0.76	β
1p90	0.17	0.56	0.74	$\alpha+\beta$
1d1q	0.16	0.63	0.79	α/β
1fvq	0.16	0.51	0.67	$\alpha+\beta$
1vjk	0.16	0.65	0.81	$\alpha+\beta$
1bkr	0.15	0.64	0.80	α
1d0q	0.15	0.54	0.69	$\alpha+\beta$
1i1n	0.15	0.62	0.78	α/β
1nb9	0.15	0.55	0.70	$\alpha+\beta$
1chd	0.14	0.63	0.77	$\alpha+\beta$
1r26	0.14	0.72	0.86	α/β
1cc8	0.13	0.75	0.88	$\alpha+\beta$
1xff	0.13	0.61	0.74	$\alpha+\beta$
1i4j	0.12	0.62	0.74	$\alpha+\beta$
1jyh	0.12	0.68	0.80	$\alpha+\beta$
1pch	0.12	0.74	0.86	$\alpha+\beta$
1smx	-0.11	0.66	0.55	β
1g9o	-0.12	0.72	0.60	$\alpha+\beta$
1dqg	-0.15	0.56	0.42	β
1fx2	-0.15	0.74	0.60	$\alpha+\beta$

whose top 1 models' TM-score predictions are over-estimated (triangles at the lower-right side of $y = x + 0.1$, and the last four rows in the table), the size of the data set (only four

proteins) is too small to draw a conclusion.

The protein class distribution for the protein chains whose predicted TM-scores versus real-scores are between the lines of $y = x \pm 0.1$ are shown in Figure 5.36. Again, As compared with the protein class distributions in the training/validation set of DeepCDpred, the percentages of α and α/β proteins are lower; but that of β protein is higher. A χ^2 test shows the two the distributions of the two protein classes have no significant difference ($p = 0.17 > 0.05$).

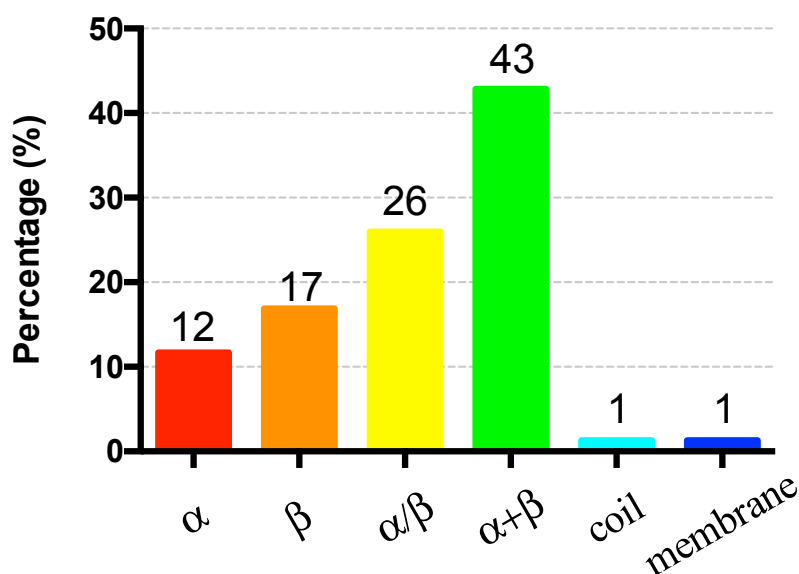


Figure 5.36. Protein class distribution of the protein chains whose predicted vs real TM-scores are between the lines of $y = x \pm 0.1$ in Figure 5.35a. Numbers above the bars are the counts of proteins in each class.

To test whether or not the TM-score prediction method can work with contact prediction data from other methods, the TM-scores of the structure predictions based on the top-ranked 1.5L contact predicted from MetaPSICOV were also used to check the performance of the TM-score prediction network. The result is shown in Figure 5.37. Similarly, two

lines of $y = x \pm 0.1$ are drawn on the scatter (Figure 5.37a) and, notably, 68% of the predicted TM-scores are in the range of the real TM-score ± 0.1 . Additionally, the correlation coefficient between the predicted TM-score and real TM-score is adjusted- $R^2 = 0.43$. There is no significant difference between the averages of the predicted and the real TM-scores (by a paired t-test, $p > 0.05$). The averages of the real and the predicted TM-scores are 0.62 and 0.63, respectively, which means the difference is 0.01 (Figure 5.37).

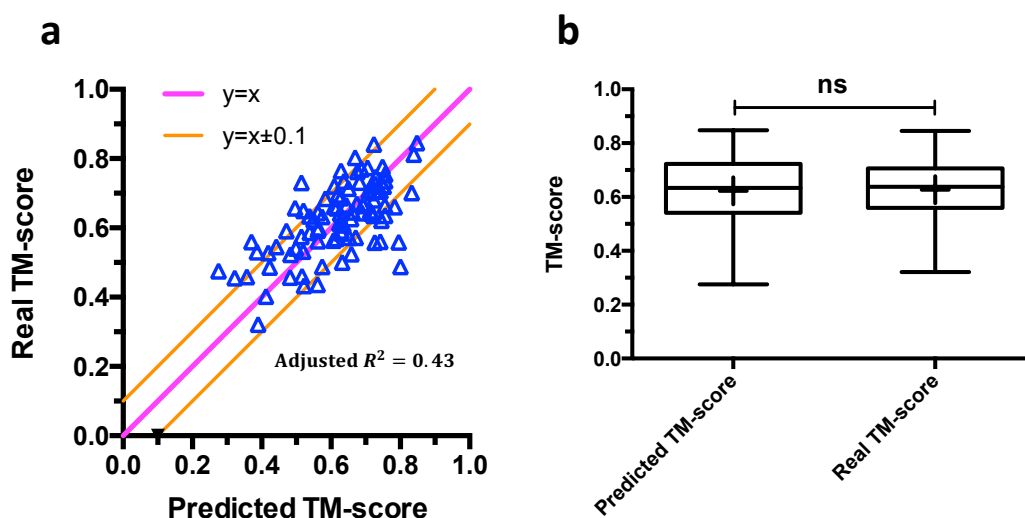


Figure 5.37. Predicted TM-scores versus real TM-scores from the structure predictions based on the contact constraints from MetaPSICOV for the proteins in the test set of DeepCDpred. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.35b.

The outliers of proteins whose top 1 models' TM-scores are poorly predicted with the network model, are determined by the two lines of $y = x \pm 0.1$, and listed in Table 5.12, together with the protein classes.

Table 5.12. The classes of the proteins whose top 1 models' TM-scores are poorly predicted by the TM-score prediction network model. The top 1 models are predicted with the top-ranked 1.5L contacts from MetaPSICOV. TM-score difference is defined as the real TM-score subtracting the predicted TM-score for the same protein. The outlier proteins located outside the two lines of $y = x \pm 0.1$ in Figure 5.37a are listed in this table. Rows in the table are ranked by TM-score difference from the highest to the lowest.

PDB ID	TM-score Difference	Predicted TM-score	Real TM-score	Protein Class
1hh8	0.31	0.49	0.80	β
1k7j	0.27	0.32	0.59	coil
1avs	0.24	0.56	0.80	$\alpha+\beta$
1htw	0.22	0.44	0.66	$\alpha+\beta$
1fx2	0.18	0.56	0.74	β
1g9o	0.17	0.56	0.73	α/β
1vmb	0.14	0.46	0.61	coil
1atz	0.13	0.70	0.83	α
1cke	0.13	0.50	0.63	$\alpha+\beta$
1mug	0.13	0.52	0.66	$\alpha+\beta$
1aba	0.12	0.64	0.76	α/β
1bdo	0.12	0.66	0.78	α
1cjw	0.12	0.62	0.75	α/β
1kw4	-0.11	0.72	0.61	$\alpha+\beta$
1g2r	-0.11	0.53	0.42	β
1m8a	-0.12	0.59	0.47	$\alpha+\beta$
1r26	-0.12	0.84	0.72	α/β
1vfy	-0.13	0.46	0.32	α/β
1k6k	-0.13	0.80	0.67	α/β
1roa	-0.13	0.65	0.52	$\alpha+\beta$
1beh	-0.14	0.53	0.39	$\alpha+\beta$
1d1q	-0.14	0.76	0.63	$\alpha+\beta$
1fvg	-0.16	0.66	0.50	α/β
1tif	-0.19	0.56	0.37	α
1i71	-0.2	0.48	0.28	$\alpha+\beta$
1c9o	-0.22	0.73	0.51	α

From the table, the percentages of the protein classes α , β , α/β , $\alpha+\beta$, coil and membrane of the proteins whose top 1 models' TM-score predictions are under-estimated (triangles at the upper-left of $y = x + 0.1$) are 15%, 15%, 23%, 31%, 15% and 0%, respectively. As

compared with the protein class distributions in the training/validation set of DeepCDpred (25%, 9%, 20%, 44%, 1% and 1% for the six classes in the training/validation set), the percentages of the α and α/β proteins are seen to be lower, while those of the $\alpha+\beta$ and coil proteins are higher. A χ^2 test shows the two protein class distributions are significantly different ($p < 0.05$). As for the proteins whose top 1 model's TM-score predictions are over-estimated (triangles at the bottom-right of $y = x - 0.1$), the percentages of the protein classes α , β , α/β , $\alpha+\beta$, coil and membrane of the proteins are 15%, 8%, 31%, 46%, 0% and 0%, respectively. Again, by comparing with the protein class distributions in the training/validation set of DeepCDpred (25%, 9%, 20%, 44%, 1% and 1% for the six classes in the training/validation set), the percentage of α protein is found to be lower, and that for α/β protein is higher. A χ^2 test shows the distributions of the two protein classes have no significant difference ($p = 0.22 > 0.05$).

The protein class distributions for the protein chains whose predicted TM-scores versus real-scores are between the lines of $y = x \pm 0.1$ are shown in Figure 5.38. As compared with the protein class distributions in the training/validation set of DeepCDpred, it is clear that the percentage of α protein is lower, and those of the β and α/β proteins are higher. A χ^2 test shows the distributions of the two protein classes have no significant difference ($p = 0.10 > 0.05$).

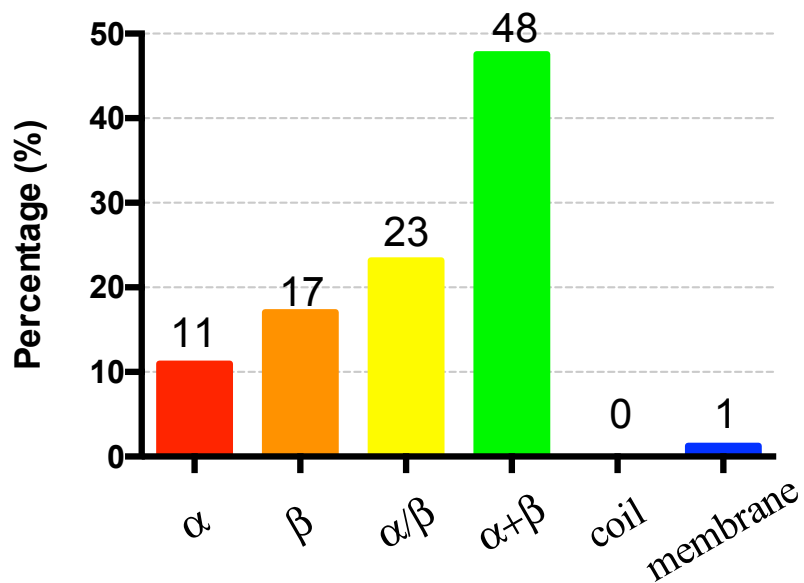


Figure 5.38. Protein class distribution of the protein chains whose predicted vs real TM-scores are between the lines of $y = x \pm 0.1$ in Figure 5.37a. Numbers above the bars are the counts of proteins in each class.

5.8 Examples of Some of the Best Protein Structure Predictions

This section introduces the structure prediction results of six proteins from the test set to illustrate the performance of DeepCDpred_AbInitio. For the six selected proteins, the relationship between the real TM-score of the predicted model and the Nf value is shown in Figure 5.39. In the figure, the TM-score of the blind test protein Q9FLY6 (uniprot id) and the Nf value are also included. The detailed result of the structure prediction of this protein will be introduced in the next section. The blue triangles in the figure represent other protein chains in the test set. The Nf cutoff value of 64 is also drawn in the figure; it is stated in a previous research (Ovchinnikov et al. 2017b) that above this value, the predicted structure is likely to have the same fold as the native structure.

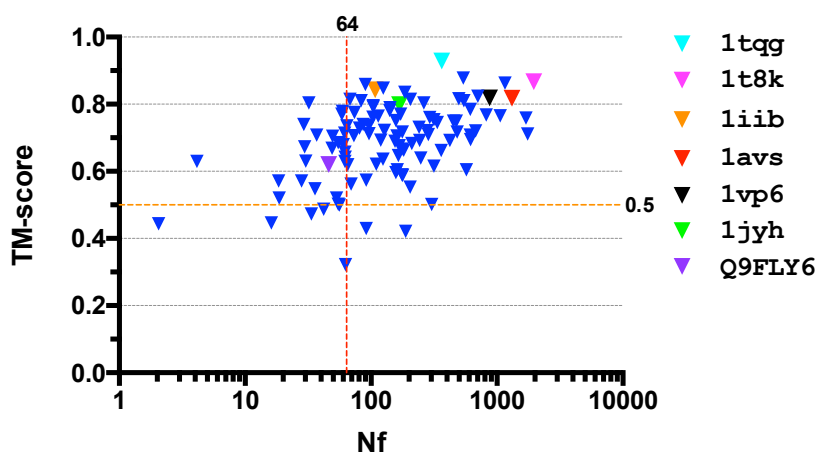


Figure 5.39. TM-score versus Nf for the six example proteins and the blindly tested protein.

The structure predictions of these six protein chains are shown in Figure 5.40 (left), overlapping with the corresponding experimental structures. They were made by using the Rosetta *ab initio* modelling protocol described in the Model Development chapter (the previous chapter), and by using the contact and distance constraints from DeepCDpred. The constraints are selected by using the score cut-off based method (a minimum neural network score of 0.4 for contact and 0.6 for all of the three distance bins). As a comparison, the top 1 models of these proteins predicted with the contact constraints (a minimum neural network score of 0.56) from MetaPSICOV, overlapping with the experimental structures, are shown in Figure 5.40 (right). Clearly, the top 1 models of the six proteins are better-predicted with DeepCDpred predicted constraints than with MetaPSICOV predicted constraints (as indicated by the real TM-score values).

The six proteins shown here were chosen since they were predicted well, which cover a range of Nf values and are from different protein classes, including α (1tqg, qt8k and 1avs), α/β (1iib) and $\alpha + \beta$ (1vp6 and 1jyh) proteins.

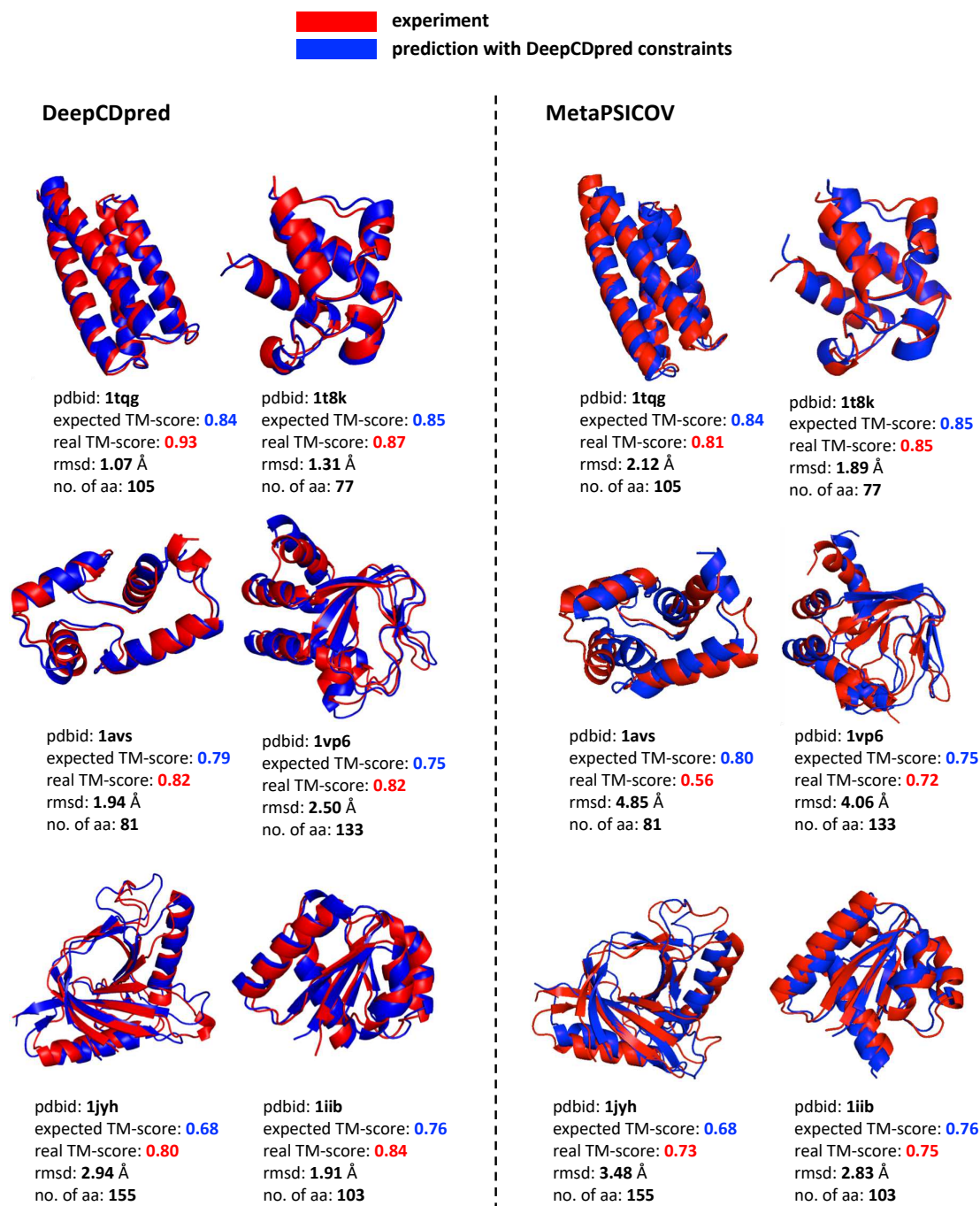


Figure 5.40. Comparisons of structure predictions of six proteins based on DeepCDpred predicted constraints (left) and MetaPSICOV predicted constraints (right). Real TM-scores between the prediction and the experimental structure, predicted TM-scores and RMSD are also shown in the figure. In the figure, no. of aa means the number of amino acids in the protein chain.

5.9 A Blind Test

In addition to the tests with proteins of known structure, a blind test with a protein of genuinely unknown structure can leave no possibility for unconscious bias to creep in. The blind test was conducted by predicting a protein chain whose 3D structure was resolved recently, yet to be released to the public. The protein does not have any homologues in PDB (it was confirmed in two ways: (1) by using HHblits to search the PDB sequences with default settings and no hits found; (2) no structures listed in the Pfam family website for the family which this protein belongs to (http://pfam.xfam.org/family/Self-incomp_S1, last check: November 2018)). Therefore, comparative modelling is not a good option to predict its 3D structure.

The uniprot accession code of the protein is Q9FLY6. It has 112 amino acids as listed below.

```
>sp|Q9FLY6|21-132
```

```
CKEIEIVIKNTLGPSRILQYHCRSGNTNVGVQYLNFKGTRIIKFKDDGTERS SRWNCLFRQ
```

```
GINMKFFTEVEAYRPDLKHPLCGKRYELSARMDAIYFKMDERPPQPLNKWRS.
```

After the steps of inter-residue contact and distance predictions and structure prediction by using DeepCDpred_AbInitio, the top 1 model was selected by choosing the one with the lowest Rosetta energy score. The contact and distance constraints fed into the Rosetta *ab initio* modelling protocol were selected based on the score cut-off method: a minimum score of 0.4 for contacts and minimum scores of 0.6 for all of the three-bin distances.

Detailed information about this protein is shown in Table 5.13. The TM-score was also predicted, as presented later.

Table 5.13. Information about the blind test protein.

UniProt Code	No. of amino acids	No. of Homologous Sequences	No. of Effective Sequences	Nf	Experimental Structure Determination
Q9FLY6	112	1,049	488	46	NMR

In order to depict the quality of the prediction in more detail, the top 5 of the 100 candidate models were sorted according to the Rosetta energy score. The last four models in the top 5 were aligned to the top 1 by the coordinates of all C_{α} atoms. Then, the average distance at each residue among the five models was calculated. The regions with large distances are expected to be flexible and hard to predict accurately; the regions with small distances are expected to be conserved and easy to predict precisely. Another calculation based on the inter-amino-acid contact prediction was also made. Like the way to evaluate the distances among the top 5 models, this is another way to evaluate which regions are easy to predict and which regions are hard. The calculation was straightforward: for all residue pairs including the residue of interest, the neural network scores output by DeepCDpred that were above a threshold of 0.4 were summed. The value of the sum is called the contact strength. Regions with higher contact strength are expected to be predicted more precisely.

The predicted amino acid residue contact map of the blind test protein is shown in Figure 5.41. The average distance among the top 5 predicted models and the contact strength calculated from the contact map are shown in Figure 5.42.

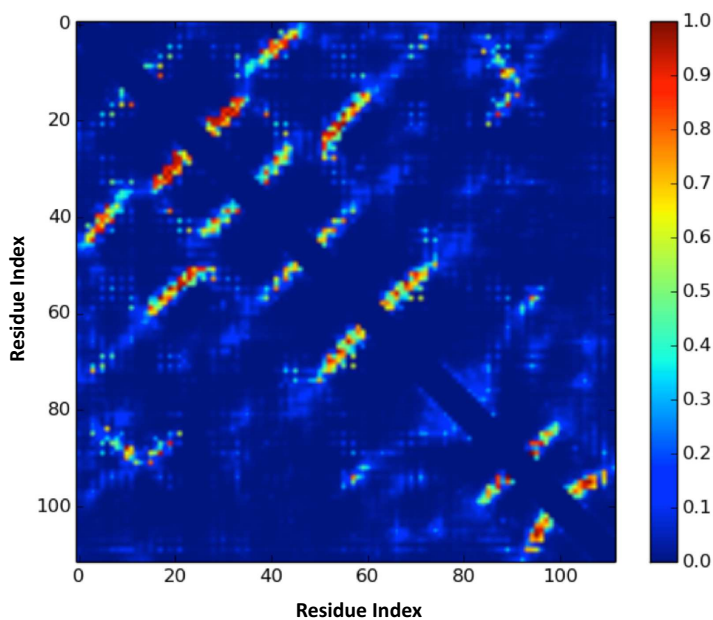


Figure 5.41. Amino acid contact map of Q9FLY6 predicted by DeepCDpred.

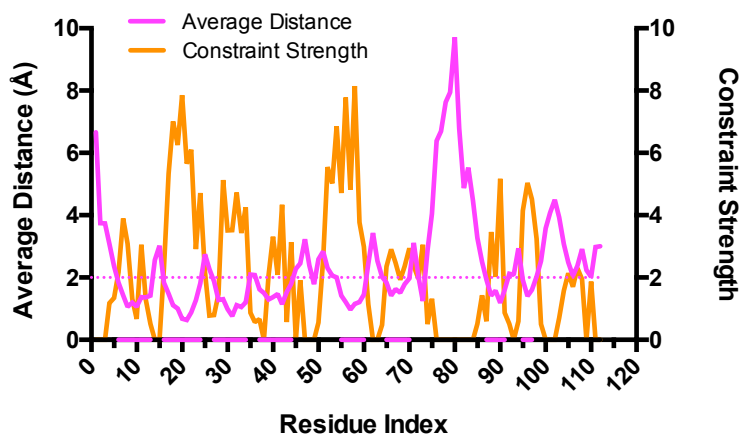


Figure 5.42. The comparison of the average distance distribution from the predicted top 5 models of Q9FLY6 and the contact strength from the contact prediction of the same protein. The average distance = 2.0\AA is arbitrarily selected as the cut-off to separate the two regions. The residue indices at which the average distance $< 2.0\text{\AA}$ are coloured as magenta on the $y = 0$ axis.

The superimposition of the predicted top 1 model from DeepCDpred_AbInitio with the experimental NMR structure (the NMR structure is usually an ensemble; the structure shown here is the most representative in the ensemble) is shown in Figure 5.43. The top 1 model was selected by the lowest Rosetta energy score from the 100 candidate models. Low variation regions (or highly constrained residues) are indicated in red, and high variation regions (or poorly constrained residues) are indicated in magenta. The positions of both regions are inferred from the above figure by using a threshold of the average distance, 2.0 Å (Figure 5.42). The TM-score between the top 1 model and the NMR structure is 0.62, which means they have the same fold. The predicted TM-score, which is 0.60, is very close to the real TM-score. The whole backbone C_{α} RMSD between the two structures is 4.6Å.

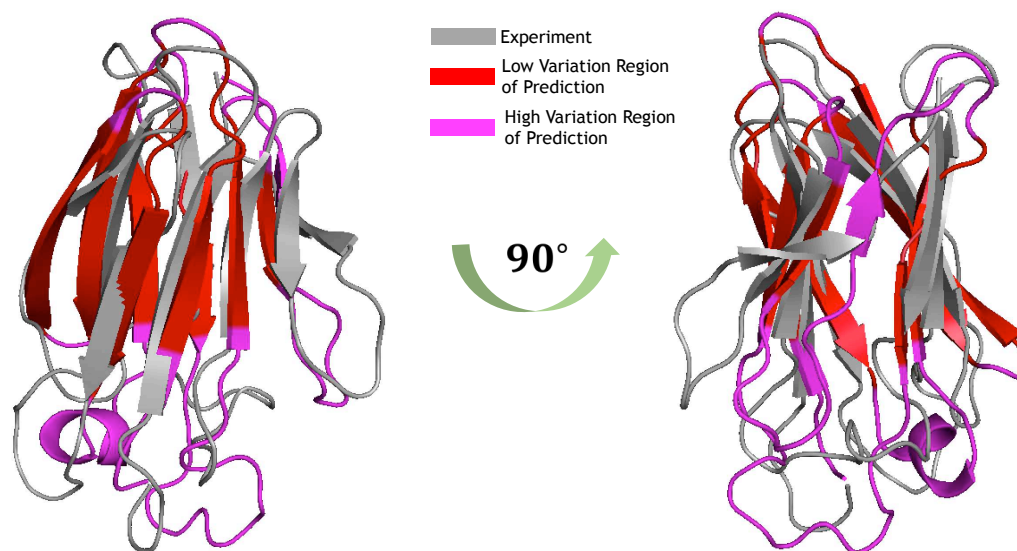


Figure 5.43. Overlaying the predicted top 1 model of Q9FLY6 to the experimental structure (coloured as grey) with the low variation region residues coloured in red and the high variation region residues coloured in magenta. The low and high variation regions of the predicted structure are defined in Figure 5.42.

The top 1 model covers all of the beta strands in the experimental structure; the connections of the nine beta strands are all correctly predicted. In order to depict the structure in more detail, TM-score and RMSD (backbone C α) between the top 1 model and the experimental structure were calculated for the nine beta strands only with the resultant values of 0.61 and 2.91Å, respectively. The amino acid positions of the nine beta strands in the experimental structure are shown in Table A.2 (Appendix A). These results show that TM-score calculation is barely affected by the flexible coils (from 0.62 to 0.61); however, after removing the coils (the residues are not in the beta strands), the RMSD value is significantly improved (from 4.6Å to 2.9Å).

For the β proteins in the test set, together with Q9FLY6, the variation of the TM-score (between the top 1 predicted model and the corresponding experimental model) against the Nf value is summarised in Table 5.14. Comparing to 2uca, 1i1j, 1ej8, and 1gzc, Q9FLY6 is demonstrated to have the smallest Nf, but the best top 1 model prediction; however, comparing to 1hxn, Q9FLY6 has a larger Nf, but a slightly worse top 1 model prediction. In Figure 5.39 of the previous section, the TM-score versus Nf of Q9FLY6 was already shown along with all of the protein chains in the test set.

Table 5.14. Comparisons between the top 1 model quality of Q9FLY6 and the top 1 model quality of the β proteins in the test set of DeepCDpred whose Nf values are similar to that of Q9FLY6. Rows are sorted by Nf column from the largest to the lowest.

PDB ID/UniProt Code	TM-score With Experiment Structure	Nf	Protein Class
2cua	0.64	92	β
1i1j	0.56	92	β
1ej8	0.53	56	β
1gzc	0.53	53	β
Q9FLY6	0.62	46	β
1beh	0.49	42	β
1hxn	0.63	30	β
1f10	0.52	19	β

As a comparison, the top 1 model predicted with MetaPSICOV's top 1.5L contact constraints and the same Rosetta *ab initio* protocol is shown in Figure 5.44. The TM-score between it and the experimental structure is 0.43, which means this model cannot recover the correct fold of this blind test protein.

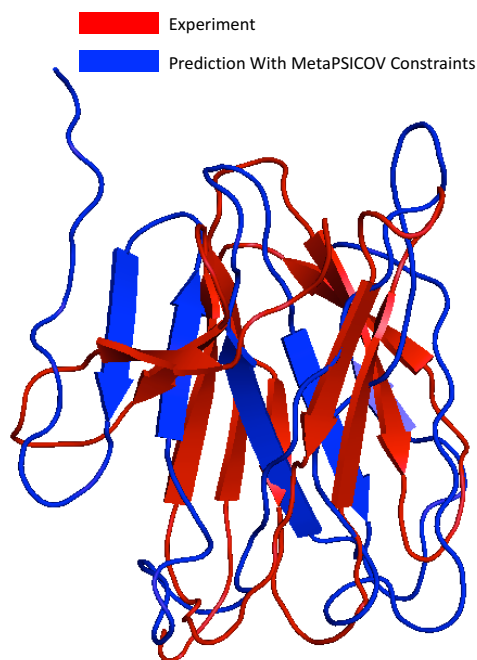


Figure 5.44. Overlaying the predicted top 1 model (coloured as blue) of Q9FLY6 to the experimental structure (coloured as red). The structure prediction was made by MetaPSICOV's top 1.5L contact constraints and the Rosetta *ab initio* modelling protocol introduced in the Model Development chapter (Chapter 4). The TM-score between them is 0.43.

5.10 Comparisons of Amino Acid Contact and Structure Predictions among DeepCDpred, RaptorX and NeBcon

In Chapter 2, three recently published algorithms of amino acid contact predictions, plmConv (Golkov et al. 2016), NeBcon (He et al. 2017) and RaptorX (Wang et al. 2017b) were introduced. Since the paper of plmConv does not release the source code or web server, it is impossible to compare the performance of DeepCDpred to plmConv. Fortunately, the latter two (NeBcon and RaptorX) do provide online web servers. The results of contact and structure predictions introduced in this subsection are based on the comparisons among DeepCDpred and these two algorithms.

Since it is a time-consuming process from uploading protein sequences to the server of each of the two methods to receiving the results of the contact predictions and/or structure predictions, only eight protein chains were selected in the comparisons. Besides the protein Q9FLY6, another seven protein chains were randomly picked out from the test set. It is noteworthy to point out that RaptorX predicts both amino acid contacts and protein structures; but NeBcon only predicts contacts.

In Figure 5.45, the boxplots show the prediction accuracies of the top-ranked 1.5L amino

acid contacts from the three methods; among them, RaptorX has the best contact prediction performance and NeBcon has the worst. In detail, prediction accuracies of the top 1.5L contacts predicted by RaptorX are significantly higher than those achieved by using DeepCDpred for the 8 proteins ($p < 0.05$); the accuracies of the latter are significantly higher than those from NeBcon ($p < 0.001$). The comparisons are made by paired t-tests. The average accuracies of the three methods are 78.8%, 75.0% and 58.2%, respectively. Thus, the difference in the results between RaptorX and DeepCDpred is small.

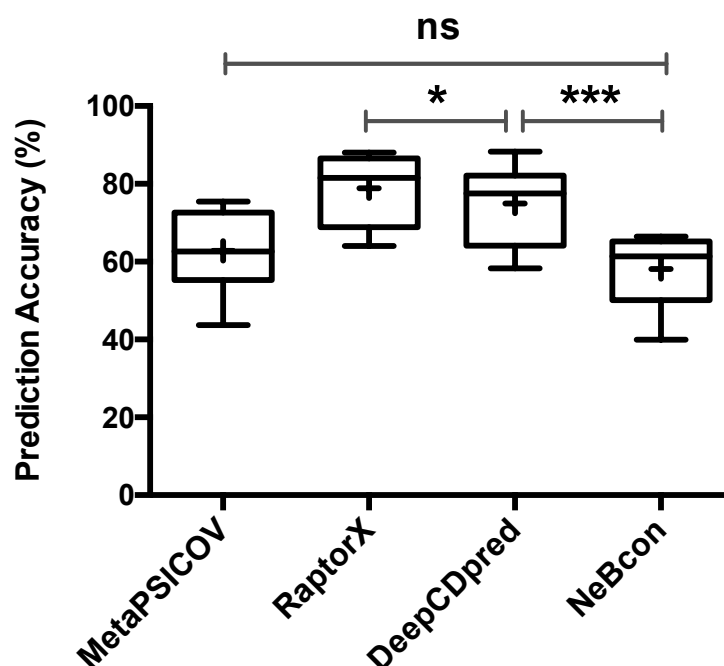


Figure 5.45. Comparison of the contact prediction accuracies of RaptorX, DeepCDpred and NeBcon for eight test proteins based on the top-ranked 1.5L contact predictions. *: $p < 0.05$; ***: $p < 0.001$, significances were calculated with paired t-tests. The PDB ID of the seven of the eight proteins are 1aap, 1d1q, 1h2e, 1hdo, 1hh8, 1tqg and 1w0h; the UniProt accession code of the remaining one protein is Q9FLY6, which has been solved by one of our co-workers of this study, but not yet published. Whiskers, middle lines and crosses have the same meanings as those in Figure 5.37.

Since the contact prediction of NeBcon is very poor as compared with RaptorX and DeepCDpred, it is simply removed from the comparisons of structure predictions. Also, for simplicity and fairness, the following method of constraint selection is used for the comparisons of structure predictions:

- a. RaptorX: the top ranked 1.5L contact predictions;
- b. DeepCDpred (contact only): the top ranked 1.5L contact predictions;
- c. DeepCDpred: the top ranked 1.5L contact predictions and score cut-off selected distance (minimum score of 0.60 for each distance bin).

The Rosetta *ab initio* modelling protocols are the same for the three groups, besides which, the structure predictions downloaded from the RaptorX server are also included in the comparison. The structure predictions downloaded directly from the RaptorX server are worse than the other two methods (Figure 5.46, $p < 0.05$).

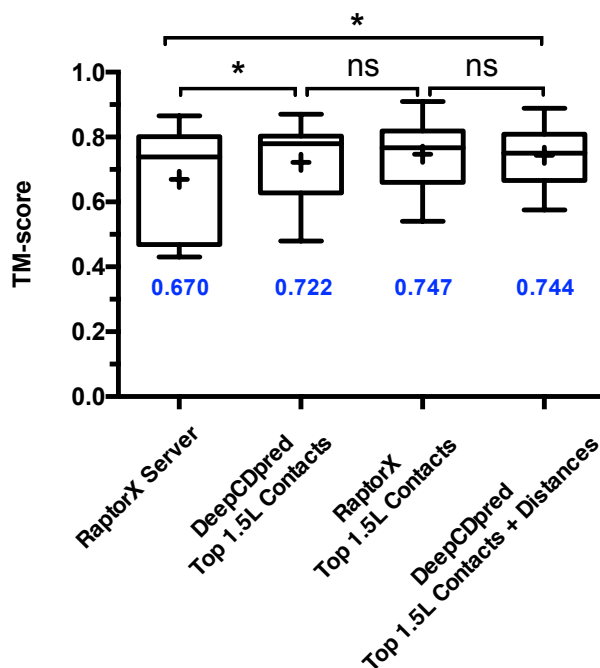


Figure 5.46. Comparisons of quality of structure predictions between DeepCDpred and RaptorX based on variant contact and distance constraints and structure simulation protocols. The TM-scores of the models predicted by the RaptorX server for the eight proteins are shown in the left-most boxplot. The other three boxplots correspond to the structure predictions by using the same structure simulation protocol, but with different constraints. Blue numbers are the TM-score averages of the four groups, respectively. Paired t-tests are used to compare the TM-score difference among the groups.

In Figure 5.46, when using the same Rosetta modelling protocol with the 1.5L contact constraints from DeepCDpred or RaptorX, there is no significant difference in terms of the quality of the models ($p > 0.05$). Adding DeepCDpred distance constraints alongside the DeepCDpred contact constraints does not significantly improve the structural models as compared with RaptorX contact constraints alone. As an example, the structure predicted by the RaptorX server is shown in Figure 5.47 aligned with the experimental structure. The TM-score between them is 0.43, which means that the predicted structure fails to recover the correct fold of the protein. Errors can also be found in the figure. Again, the top 1 model predicted by using Rosetta *ab initio* modelling protocol was selected by the lowest Rosetta energy.

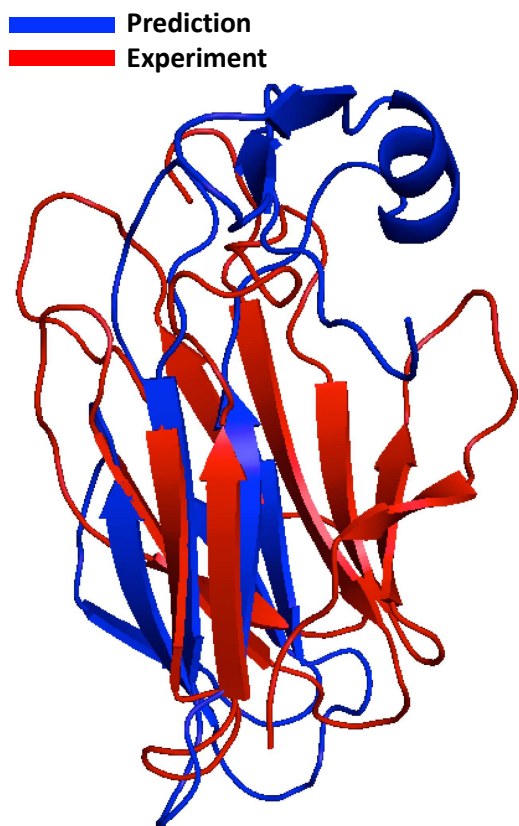


Figure 5.47. Superimposition between the top 1 predicted model of the blind test protein Q9FLY6 and the experimental structure. The model is predicted by the online RaptorX server.

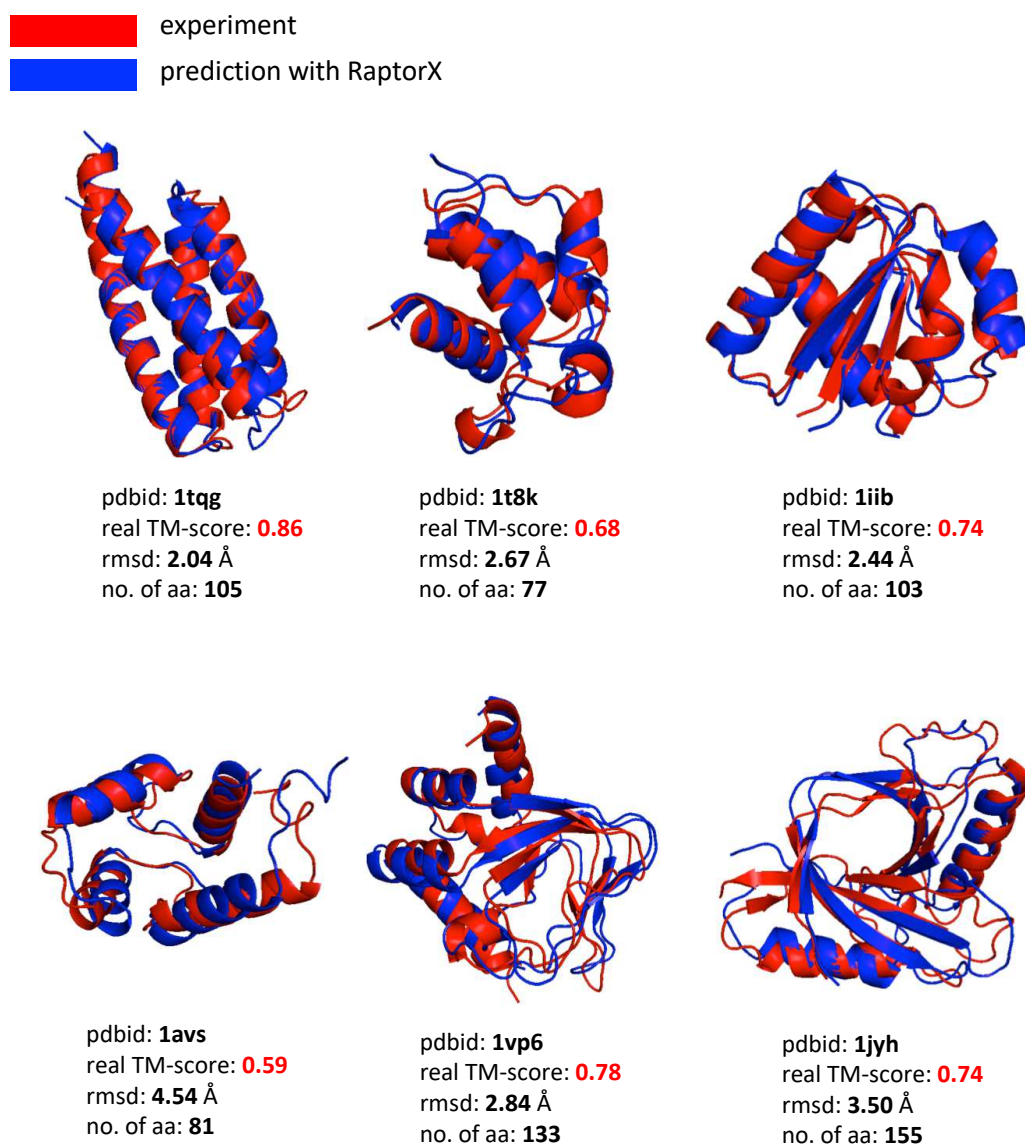


Figure 5.48. Six examples of structure predictions from RaptorX server. Real TM-scores between the predictions and the experimental structures, predicted TM-scores and RMSD are also shown in the figure. In the figure, no. of aa means the number of amino acids in the protein chain.

With the predicted constraints, the RaptorX server uses the CNS program ([Brunger et al. 1998](#)) to do structure modelling. The putatively best model for each protein chain

as reported by RaptorX is selected according to the CNS build-in energy function. CNS is suggested to be not as good as Rosetta *ab initio* for building *ab initio* protein models ([Wang et al. 2017b](#)). More discussions about the results of this section can be found in Section 6.8 of the Discussion chapter (Chapter 6).

5.11 Improving the Accuracies of the Amino Acid Contact and Distance Predictions of DeepCDpred by Using Metagenomics Data

It has been shown that metagenomics data could be used to enrich sequence alignments (Ovchinnikov et al. 2017b). This section investigates whether the metagenomics data is useful for improving DeepCDpred. For simplicity, protein chains from the test set were only used if a UniRefKB search with HHblits produced a sequence alignment with Nf less than 64. Based on this selection criteria, there are 29 chains available. The pdb ids of these proteins are listed in Table 5.15.

Table 5.15. PDB ID list of the 29 protein chains with Nf values less than 64.

PDB ID/CHAIN NAME								
1aoeA	1bebA	1behA	1chdA	1ctfA	1d4oA	1dixA	1dmgA	1ej8A
1fk5A	1f10A	1fvga	1g2rA	1gzca	1htwA	1hxnA	1i71A	1j3aA
1jo0A	1k7jA	1kqrA	1lm4A	1m4jA	1m8aA	1nb9A	1roaA	1tifa
1whiA	1wjxA							

The results are shown in Figure 5.49. For the contact predictions, the accuracies at L/10 and L/5 become slightly worse after using extra sequences, but get better at L and 1.5L. The total effect is that the contact prediction gets better, since the accuracies at L/10 and L/5 are already very high (95%) and at L and 1.5L, much more predictions are included

and thus the prediction improvement is more significant. It can be also obtained that distance predictions (all of the three bins) become better after adding the metagenomics sequences.

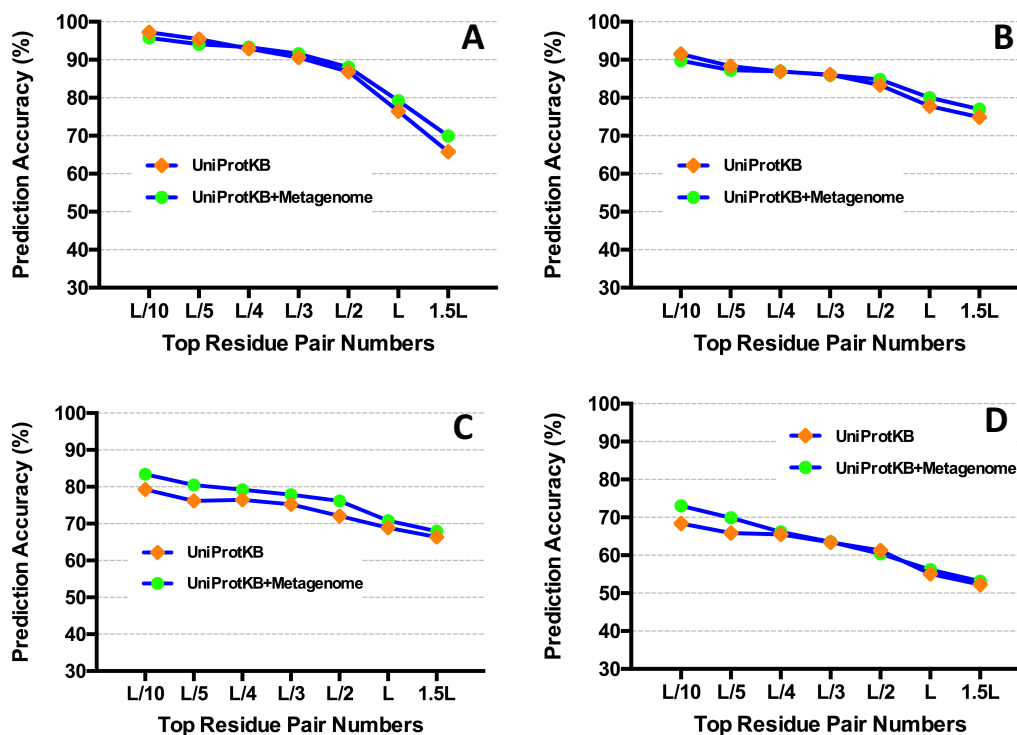


Figure 5.49. The comparison of the accuracies of amino acid contact and distance predictions between adding metagenomics data to UniProtKB and using UniProtKB only. A: contact (0-8 Å); B: distance bin 8-13 Å; C: distance bin 13-18 Å and D: distance bin 18-23 Å.

5.12 Improving the Accuracy of Amino Acid Contact Predictions of DeepCDpred by Using Networks with 5 Hidden Layers

As mentioned in previous chapters, a deeper network might be expected to produce more accurate predictions of amino acid contact and distance. Two types of five-hidden-layer networks were trained here, one using cross entropy as the loss function and the other using MSE. The same with the previous version of DeepCDpred, each of them was trained for four times, and each time there was a slightly different contact range ($0-7.9\text{\AA}$, $0-8.0\text{\AA}$, $0-8.1\text{\AA}$ and $0-8.2\text{\AA}$); the final contact prediction score was the average of the four network output scores. The result of contact prediction accuracy comparison between them, together with the previous two-hidden-layer version of DeepCDpred and MetaPSICOV for the 108 test proteins is shown in Figure 5.50. Both the two new networks produce $\approx 1.5\%$ higher contact prediction accuracy than the previous version of DeepCDpred for both the top-ranked L and the 1.5L contact predictions.

Due to the time limit, the work of replacing the two-hidden-layer networks of the three-bin distance predictions with the five-hidden-layer networks has not been done. This step will be completed in the future work (refer to the Discussion chapter (Section 6.9) for more information). The difference of the contact prediction accuracies between the two

versions of the five-hidden-layer DeepCDpred is less than 0.5% for all of the 7 top-ranked predictions.

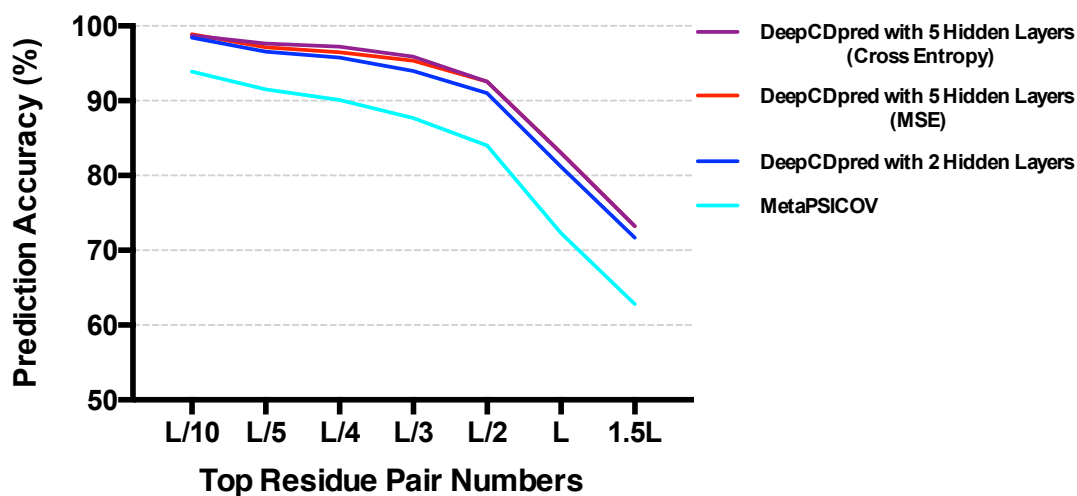



Figure 5.50. Predictions from a two stage neural network trained by using the same feature vector as the DeepCDpred network previously described, but with five hidden layers. The network consists of 5 ReLU (hidden layers) and 1 sigmoid (output layer) activation functions for processing in addition to the input and output layers. Two versions of the five-hidden-layer networks were trained, one with cross-entropy as the loss function, and the other using MSE, which is the same as the two-hidden-layer version of DeepCDpred.

5.13 Online Server: PROTEINCOEVOLUTION.BHAM.AC.UK

This study programmed and released the online server of amino acid contact and distance predictions to implement DeepCDpred. This server can also carry out protein structure predictions based on the predicted geometry constraints. However, due to the computing capacity of the laboratory conducting this research, ability to make such predictions is currently limited. Only two jobs per user are allowed to be queued on the server at any one time.

The homepage is shown in Figure 5.51. A user can make a query by pasting/uploading a protein sequence in FASTA format. The job name option allows the user to have an easy-to-recognise name, and the email address is required if the user wants a reminder of the URL address of the result page of the job; otherwise, he/she has to remember it. Other functions of this server include browsing processed jobs, and displaying the results of the contact/distance predictions and structure predictions. An explanation for DeepCDpred is provided on the ALGORITHM page of the website. A user can use the contact information provided in the ABOUT&CONTACT US to contact the developer. More details of this server can be easily found when a user logs in.



Amino acid contact prediction & protein structure prediction

[SUBMIT A JOB](#)
 [SEARCH JOBS](#)
 [ALGORITHMS](#)
 [NEWS](#)
 [ABOUT ME](#)

Amino acid contact/distance prediction and protein structure prediction.

Job name (optional)

E-mail (optional)

Paste/upload ONE protein sequence of length 20-500 amino acids in [FASTA](#) format

Example

no file selected
 predicts structure?

Options

hhblits options:

hhblits database

hhblits E-value (0-1)

Minimum coverage with query sequence (60-100, %)

Maximum pairwise sequence identity (50-99, %)

Minimum sequence identity with query sequence (0-50, %)

Iteration (1-8)

hhsuite version: 2.0.16; xplor-nih version: 2.48, released in 2018.

Figure 5.51. The home page of the amino acid contact & distance prediction and protein structure prediction server, proteincoevolution.bham.ac.uk.

CHAPTER 6

DISCUSSION

In the previous chapters, the inter-amino-acid contact/distance prediction algorithm, DeepCDpred, was explained and evaluated both by using a subset of protein chains from the test set of MetaPSICOV and by blindly using a protein solved by another research group in the University of Birmingham that has yet to have its coordinates been released. It is the second-best contact predictor at the moment and only slightly worse than the algorithm of RaptorX published about one year ago. However, since DeepCDpred is capable of predicting inter-residue distance, it produces more geometry constraints. Protein structure predictions based on the constraints from DeepCDpred and the Rosetta *ab initio* protocol introduced in this thesis are significantly better than those from the RaptorX server. The RaptorX server uses CNS to predict protein structures, and CNS is thought to produce worse models than Rosetta *ab initio* does ([Wang et al. 2017b](#)). Thus, in order to remove this bias, protein structures were also predicted by using the

constraints from RaptorX and the Rosetta *ab initio* protocol; the result showed that there is no difference between models produced by using DeepCDpred contact & distance predictions as compared with the RaptorX contact predictions. Besides the structure modelling protocol, another advantage of DeepCDpred is the much simpler architecture than that of RaptorX in terms of contact prediction accuracy, which means that DeepCDpred may require less time for both training and predicting. The online server of proteincoevolution.bham.ac.uk was programmed and released to the public; it is expected to be beneficial to the research community. In the previous chapter, it was also shown that adding protein sequences from metagenomics data could slightly increase the accuracy of the predictions of amino acid contacts and distances for those proteins with a limited number of diverse sequences in their MSAs. Since the use of metagenomics data is not the focus of this study, whether the extra improvements could lead to better structure predictions or not has not been checked yet. The Python code that searches homologous sequences from the combined protein sequence dataset (UniRefKB+Metagenome) was contributed by Tugce Oruc, a first-year PhD student from the same research group that the author of this thesis belongs to.

There has clearly been great progress made in inferring coevolutionary signals from aligned sequences of proteins in the same family, especially in the past two decades. Using inter-residue contact/distance predictions as geometry constraints in template-free protein structure prediction has become a standard technique ([de Juan et al. 2013](#); [Kosciolek and Jones 2016](#)). It is now possible to achieve accurate structure predictions for protein

families that were impossible to obtain before ([Ovchinnikov et al. 2017b](#)). Nevertheless, some of the most significant developments have only arisen in the past eight years or so. Machine learning methods for amino acid contact/distance prediction have only been developed in the last few years. As such, this field is relatively young. Therefore, current methods, including DeepCDpred, proposed in this thesis, still have a number of limitations that will require further exploration and technological refinement to address. The following aspects are potential areas for the improvement of DeepCDpred in the future.

6.1 Sequence Alignment Methods

As coevolution-based approaches rely solely on an MSA to identify covariation between amino acid positions, the quality of the MSA is important. It has been shown that alignment errors can result in erroneous observations of correlated mutation ([Dickson et al. 2010](#)). In this work, DeepCDpred requires HHblits to generate alignments, since it can find more remote homologues than PSSM profiles-based algorithms (e.g., PSI-BLAST) or profile HMMs (already mentioned in the Background chapter (Chapter 2)).

Nevertheless, even with the capability to detect very remote and divergent sequences, it is hard to find a balance between increasing the number of sequences and being at the risk of adding noisy sequences. Better ways for sequence selection and alignment generation are likely to benefit contact/distance prediction in the future work. As for this aspect, the recently published machine learning approaches, PconsC and PconsC2

([Skwark et al. 2013, 2014](#)), may provide some inspirations. Instead of building one single accurate alignment, the researchers for these approaches used eight different MSAs (e.g. from both HHblits and Jackhmmer with different E-values) for a query protein sequence and joined the coevolutionary coupling predictions of PSICOV and plmDCA based on these alignments by using a random forest classifier. Although PconsC and PconsC2 were found to be more effective at predicting amino acid contacts than earlier methods, the calculations are time-consuming. Hopefully, faster approaches to detect homologous sequences and build sequence alignments will come up soon.

6.2 Available Sequences

It was shown that the accuracy of the predictions of coevolution-based amino acid contact and distance is correlated with the value of N_f ([Ovchinnikov et al. 2017b](#)), as well as the accuracy of the predictions from machine learning based methods, such as DeepCDpred, which use coevolutional couplings as input features (shown in this study). It is worth noting that protein sequences from metagenomics projects are accumulating at a much faster speed than those of traditional sequencing projects ([Ovchinnikov et al. 2017b](#)). Recent efforts have also been made regarding sequencing the genomes of organisms in high-temperature ([Inskeep et al. 2010](#)), the Arctic ([Jeon et al. 2009](#)), and deep seas ([Sogin et al. 2006](#)) environments.

Such metagenomics projects have greatly increased the availability of bacterial sequences. However, sequences from higher organisms (e.g. eukaryotes) are less accessible, as they suffer from much fewer sequenced genomes from different species as compared with the sequences of bacterial proteins. The current release of the UniProt-TrEMBL database (September 2017) is made up of about 66% bacterial sequences, compared to 28% from eukaryotes (<http://www.ebi.ac.uk/uniprot/TrEMBLstats>, last check: November 2018). Too few available sequences likely lead to inter-residue contact/distance prediction accuracies becoming too low to be useful in 3D modelling. Besides increasing the availability of non-redundant sequences for the current approaches, if there were a new algorithm capable of calculating contact/distance maps from 100 homologous sequences that are as accurate as those computed from 1000 sequences, it can then be expected that coevolution based *ab initio* modelling would become even more useful. Unfortunately, there is a long way to achieve this, in terms of the development trends in recent years, and it may be an impossible task due to the systematic biases that arise from missing data ([Martin et al. 2005](#)).

6.3 Interpretation of Long-distance Couplings and Improvement to Distance Prediction

It is necessary to understand where the long-distance couplings come from. In the Method Development chapter (Chapter 4), it was mentioned that some of them may arise from

the two residues on each strand of a beta sheet. Some other explanations can be found in the literature as introduced below.

1. Conformational change. If two or more experimentally solved protein structures have identical or very similar sequences, but they cannot be superimposed, it can be said that each of the structures represents a different functional conformation of the same protein. When applying DeepCDpred (or another accurate contact prediction algorithm such as MetaPSICOV) to make amino acid contact/distance prediction for this protein, some residue pairs with a high contact score may have a long distance in space when referring to one structure, but be close in another. Anishchenko *et al.* ([Anishchenko et al. 2017](#)) found only a small fraction (0.5%) of strongly coupled residue pairs are associated with conformational change. In their study, the coupling was measured by the CCMpred output score (coevolution score), which is different from the neural network score of DeepCDpred. However, conformational change may also be a reason for the long-distance coupling captured by DeepCDpred, since it uses the coevolutional couplings predicted from CCMpred as input features.
2. Structural variation within a protein family. From the same paper ([Anishchenko et al. 2017](#)), it was found that 91% of directly coevolving residue pairs in the 5 – 15 Å range are in contact in at least one homologous structure, which means that structural variation in the family could result in some degree of strong contact between coupled

residue pairs that are not directly in contact in a reference structure. For the long-distance couplings with a distance of greater than 15 Å, 19% of them arose from the structural variation in the same family ([Anishchenko et al. 2017](#)).

3. Homo-oligomeric interaction. It was also found some long-distance couplings are associated with homo-oligomeric interactions ([Anishchenko et al. 2017](#); [dos Santos et al. 2015](#); [Uguzzoni et al. 2017](#)). The fact may be that these couplings are only caused by the residues within the same chain, since in homo-dimers intra-protein couplings are not distinguishable from those due to the inter-protein interaction.

It may be possible to find out different alternative conformations by analysing the contact/distance predictions. If some residue pairs have both high scores of contact and distance predictions, it is reasonable to suggest that this residue pair contributes to the conformational change of the protein. In the future, such an analysis should be investigated. The structure variation issue is difficult to be solved by analysing contact and distance predictions. However, if more sophisticated structure simulation protocols are proposed in the future, it is expected the predicted structures are closer to the native ones.

The results of the structure predictions from DeepCDpred_AbInitio have already shown the extra constraints from distance prediction can improve the quality of the predicted structures. Filtering out the false positives of distance predictions (e.g. contacting residue pairs) could be an important step for more improvement of structure prediction that should be solved in the future. Based on the above explanations, DeepCDpred could be

modified in the following ways for improving amino acid distance prediction, which could also result in better amino acid contact prediction.

In order to include more data, multimeric protein chains were not removed in both the training set (including validation set) and the test set. In the Results chapter, it shows that about 1/5 chains in the training set are from multimers, while this fraction changes to more than 1/4 for the chains in the test set. The homo-oligomeric interactions may introduce biases into the training of the neural networks in DeepCDpred, as well as the predictions when applying DeepCDpred to the test set. Thus, DeepCDpred is unlikely to be optimised for either the predictions of amino acid contacts and distances of monomeric protein chains or the predictions of amino acid contacts and distances of multimeric protein chains. A good way that can be implemented in the future is to separate these two types of chains. That is, the version of DeepCDpred should be trained exclusively with monomeric protein chains to make contact/distance predictions for monomeric protein chains, and it is the same for multimeric chains.

Another possible way to improve distance predictions is to use a different statistical potential. The statistical potential of amino acid contact proposed in the study by (Bentancourt and Thirumalai 1999) was used in the input feature vector of DeepCDpred for both contact and distance (three bins) predictions. It is reasonable to believe that this type of potential cannot efficiently capture long-range amino acid interactions. Instead, a distance-dependent statistical potential seems to be more suitable for this task. Fortunately, such a potential was proposed in the paper by (Zhao and Xu 2012) and is capable

of measuring how favourable the interaction between two atoms is at a distance of up to more than 14 Å. Although the results shown in Figure 5.12 and Table 5.3 prove that statistical potential contributes only a little to the amino acid prediction in DeepCDpred, it is worth replacing the fixed contact potential matrix with it to update DeepCDpred in the future; maybe the new version would offer a greater contribution to the contact prediction. Further discussions about the features of DeepCDpred can be found in Section 6.6.

6.4 The Test Set of DeepCDpred

As described in Chapter 3, 108 protein chains originally from the test set MetaPSICOV (150 chains) were used as the test set of DeepCDpred. Not only DeepCDpred but also RaptorX used the aforementioned 150 protein chains as the test set to compare the performance of amino acid contact predictions to that of MetaPSICOV. These structures can be easily downloaded from the MetaPSICOV online server (http://bioinf.cs.ucl.ac.uk/software_downloads/ last check: November 2018). At the beginning of the development of DeepCDpred, the author of this thesis did not have enough experience with collecting structures from the CASP. In the future work, the contact prediction performance between DeepCDpred and other algorithms (e.g. RaptorX) could be compared with the CASP proteins. The author of this thesis participated in the CASP competition of 2018

in the contact prediction category. The results of competition ranking were not released at the time of writing this thesis.

6.5 Protein Model Selection

As mentioned many times in both Chapter 4 and Chapter 5, ranking predicted models and selecting the presumably best one were completed with the lowest Rosetta energy score in this thesis. It is worth noting that there are some protein structure quality assessment programs, such as ModFold (Maghrabi and McGuffin 2017; McGuffin et al. 2013), Qprob (Cao and Cheng 2016), and ProQ2 (Uziela and Wallner 2016). Among them, ModFOLD6 (the latest generation of ModFOLD) is a leading server of protein structure quality estimation tested in CASP12 (Maghrabi and McGuffin 2017).

The programs of protein quality assessment can be divided into two categories: consensus-model based and single-model based. Usually, the former is more accurate than the latter (Cao and Cheng 2016). Consensus-model based methods require multiple models (also called “decoys”, e.g. the 100 candidate structures generated from DeepCDpred_AbInitio for each target protein) of the target protein; pairwise similarities of the decoys are calculated and the similarity score of each decoy as compared with the others is then inferred (Konopka et al. 2012). The decoys are subsequently ranked according to their similarity scores. This kind of method is generally works slower than single-model based approaches,

which usually rank each decoy based on its features, such as the secondary structure prediction, the atom-atom and/or residue-residue interaction, the sequence profile, and the solvent accessibility prediction (Cao et al. 2017; Uziela and Wallner 2016). This type of method does not require information from other decoys. Single-model based methods are generally faster and may work better when a large proportion of the multiple models are poorly predicted (Cao et al. 2015).

In the future work, some of the latest servers or programs for assessing the quality of protein models, no matter whether they are consensus-model based or single-model based, should be tried in order to select the best model from the structure predictions of DeepCDpred_AbInitio.

Besides ranking the models and selecting the best one, another important task is to provide a confidence score for the predicted model(s). The confidence score represents how accurate the model is – that is, how close the model is to the native structure. The method proposed in this thesis is a neural network model; it predicts the TM-score which ranges from 0 to 1. The result shown in Section 5.7 (the Results chapter) indicates that the predictions are well-correlated with the real values for both the structure predictions based on the constraints from DeepCDpred and MetaPSICOV; the majority of differences between the predictions and the real corresponding ones fall in a range of less than 0.1. In addition, the model is simple and the prediction is fast. However, the result also clearly shows that predicted TM-scores from the models are generally smaller than the real corresponding ones for the models based on DeepCDpred constraints. Possible ways

to improve the TM-score prediction include using more features in the network model, such as the DOPE score calculated by MODELLER ([Webb and Sali 2014](#)), using a more complicated network architecture, and adopting a larger training set. A network with a more complicated architecture (e.g. more hidden layers) does not necessarily perform better than a simple one, because of the problems of vanishing gradients and overfitting. However, due to the use of ‘skip connections’, ResNet and its variants have indicated that a deeper architecture could achieve a lower error rate in image classification ([He et al. 2016](#); [Huang et al. 2016](#)). Thus, they are worth trying on TM-score predictions with a deep architecture. More accurate methods for confidence score predictions should also be studied (e.g. not restricted to predictions of TM-score). Some published paper may provide clues regarding how to make improvements.

In the study by Roy *et al.* ([Roy et al. 2011](#)), all the decoys of the target protein are clustered and the average distance (RMSD) of the decoys in the top cluster to the centroid of this cluster is calculated; a confidence score for the top predicted model is predicted based on this distance. In another paper ([Xu and Zhang 2013](#)), the confidence score is calculated based on the normalized cluster size of the top decoy cluster and a so-called ‘f-score’, which measures the quality of the threading fragments (threading is used for structure prediction in this paper).

6.6 Feature Optimization

Analysis of the importance of different features in the DeepCDpred feature vector, shown in Subsection 5.5.5 of Chapter 5, could provide clues about how to optimize the feature selections to improve contact predictions. For example, The feature category of amino acid profile, or frequency, does not affect the accuracy of contact predictions very much; however, there are 483 elements in the 752-dimensional feature vector related to the frequency (Subsection 3.5.2), which is a very large fraction. The feature category of the coevolutionary couplings from CCMpred contribute the most to the contact prediction, particularly compared to the coevolutionary couplings from EVFold and QUIC. It is not only because the former is more accurate than the latter two, but probably also a 9×9 square window is used to include 81 couplings; instead, only one coupling is included for both EVFold and QUIC for a residue pair in the feature vector. It is reasonable to remove the feature category of amino acid profile, and to use two square windows to include more couplings (neighbouring couplings) for EVFold and QUIC, respectively. This idea for the contact prediction has also been tested. The feature vector in the stage 1 networks of DeepCDpred was replaced with the new one. The result of this test is shown in Figure B.5 of Appendix B, and indicates improvement in accuracy of 3.5% for the top 1.5L contact predictions.

From Figure 5.12 and Table 5.3 in the Results chapter, the secondary structure prediction can also be said to play a critical role in the contact prediction of DeepCDpred. If a

more accurate program of protein secondary structure predictions was used rather than SPIDER2, the contact (probably also distance) prediction performance of DeepCDpred could be improved. In fact, there is such a program available, called DeepCNF (Wang *et al.* 2016c), which was already mentioned in Subsection 2.7.1 (the brief review of protein secondary structure prediction). In the future work, this program should be used to replace SPIDER2 to generate the secondary structure prediction for the feature selection of DeepCDpred.

It is noteworthy that deep networks developed in the fields such as image classification and natural language processing do not require the extraction of features. Instead, features are automatically learned during the training process. This strategy of learning is known as end-to-end learning (Glasmachers 2017). A comparable work by Golkov *et al.* (Golkov *et al.* 2016), which was mentioned in the previous chapters, uses a quasi-end-to-end learning approach. Here, ‘quasi’ means that the input of the model is not the query sequence or the MSA of the query sequence, but rather the pairwise covariance matrix inferred from the MSA by plmDCA (Golkov *et al.* 2016). The matrix may not represent all of the information in the MSA that is useful for contact prediction. Thus, incorporating the features of amino acid frequency, secondary structure prediction, and solvent accessibility prediction into the inputs may improve the accuracy of contact predictions. An idea of using a real end-to-end learning to predict contact/distance can be found in Section 6.9.

6.7 The Training Strategy of DeepCDpred

As introduced in Chapter 4, neural networks in the original version of DeepCDpred were trained with the conjugate gradient descend function 'traincgb' in the neural network toolbox of MATLAB. This function uses the batch training strategy that the values of all the parameters (weights and biases) in a neural network are updated only after all of the inputs have passed the process of the loss function optimization in a training epoch. Obviously, this strategy is memory consuming, especially when the training dataset is very large. The training/validation set of DeepCDpred includes 1,066 protein chains, which represents more than 10 million of inputs (residue pairs). The machine used for the training procedure has 128GB of RAM, but this is still not large enough. That is why the training/validation set was split into two groups.

The version of MATLAB used in this work was 2015b, which does not support the stochastic gradient descent optimization or mini-batch for feedforward neural network training. The reasons for choosing the MATLAB neural network toolbox, rather than any of the libraries available in other languages (such as Keras in Python), are: (a) the author of this thesis was more familiar with MATLAB than other languages when this project was started, and (b) other libraries such as Keras were not as widely used as it is used today (the first version of Tensorflow was released on November 9, 2015). As mentioned in Chapter 4, the batch training with MATLAB's neural network toolbox requires a large RAM size. As an estimation, the 11,651,001 inputs, with 752 dimensions each, of the

1,066 training proteins cost $11651001 \times 752 \times 8$ bytes (double-precision floating-point format (MATLAB 2019)) = 65.3 gigabytes. Additionally, weights and biases of the network also use some space of the RAM (in fact, these RAM costs are small, i.e. 0.7 megabytes and 0.8 megabytes for the two-hidden-layer and five-hidden-layer DeepCDpred, respectively). Such memory limitations could be overcome via the use of mini-batches for training, which are now routinely implemented in deep learning libraries (e.g. Keras, tensorflow and pytorch).

ReLU activation function, available in Keras, alleviated the vanishing gradient problem that neural networks with sigmoid activation functions usually encounter (Nair and Hinton 2010). This problem is probably the reason why the three-hidden-layer networks were also considered for contact and distance prediction but no improvements were found and thus a switch was made to the two-hidden-layer architecture. ReLU was introduced in MATLAB with the version of 2016a (<https://uk.mathworks.com/help/nnet/ref/nnet.cnn.layer.relu.html>, last check: November 2018). With ReLU as the activation function in the hidden-layer, more layers can be added in the neural network using the Keras library. A small improvement was found for amino acid contact predictions based on the same test set when comparing three-hidden-layer to two-hidden-layer, and four-hidden-layer to three-hidden-layer networks, respectively. The five-hidden-layer architecture was finally chosen for the amino acid contact predictions, and the results are shown in Section 5.12. The accuracy is about 1.5% higher than that with the previous version of DeepCDpred. In the future work, the distance prediction networks should also

adopt this deeper architecture with ReLU (or its variants) as activation functions.

In the old version of DeepCDpred, the loss function was chosen as the MSE (mean squared error). It was shown that the choice of cross-entropy could lead to a better result than the use of MSE for pattern recognition (Golik et al. 2013). In the algorithm of plmConv (Golkov et al. 2016), MSE was also used as the loss function in a convolutional network to predict amino acid contacts. In the new version of DeepCDpred, both MSE and cross-entropy were tried; the difference between the accuracies of the amino acid contact predictions based on them is very small (less than 0.5%).

6.8 The Comparisons of Contact and Structure Predictions Between DeepCDpred and Other Algorithms

The accuracies of amino acid contact predictions were compared between DeepCDpred and two other algorithms, NeBcon and RaptorX, as shown in Chapter 5. There are several aspects of the results worth discussing here.

Since both RaptorX and NeBcon require sending sequences to an online server, which is time-consuming, only eight proteins are selected to participate in the comparisons. Among the proteins, seven were randomly chosen from the test set of DeepCDpred and the other one is the blind test protein (Q9FLY6). These proteins may not truly represent the overall performance of RaptorX and DeepCDpred_AbInitio. Future work should include all of

the proteins in the test set of DeepCDpred, or a large set of proteins that are independent from the training/validation sets of DeepCDpred, RaptorX and NeBcon.

Another thing that should be clarified is that whether the seven proteins are in the training set of RaptorX, and NeBcon or not could not be determined by the time of writing this thesis, since the pdb id list of the training sets of the two algorithms could not be found. The blind test protein is not in the training set, since it had not been released at the time of writing, nor had its homologous proteins in PDB (later November 2017). If some (or even all) of the seven proteins are in RaptorX's, or NeBcon's training set, it could result in a bias toward RaptorX or NeBcon in both the contact and the structure predictions as compared with DeepCDpred, which means the difference between them might be smaller in the comparisons. The performance of DeepCDpred might be better than that of RaptorX based on RaptorX's own structure prediction protocol (RaptorX server) and DeepCDpred_AbInitio's *ab initio* protocol. Similarly, if some of these test proteins are in NeBcon's training set, the true contact prediction accuracy of NeBcon could be even worse.

The contact prediction accuracy of NeBcon is discussed here. NeBcon claims to be able to predict more accurate contacts than MetaPSICOV in the paper (He et al. 2017) published based on the test set of MetaPSICOV, but the results obtained in this study show that there is no significant difference between them based on a subset of the test set of MetaPSICOV (Section 5.10).

6.9 Other Machine Learning Algorithms and Best Model Selection Strategies

One way to improve the contact/distance prediction is to combine different mathematical models. The models should be as diverse as possible in order to capture different aspects of the input data. In the contact prediction step of DeepCDpred, the final model is obtained by averaging the outputs from several individual neural network models. This strategy can improve the accuracy of contact predictions in comparison by with using any individual network model as shown in Subsection 5.5.2. In the future, other types of model combinations are worth trying. For example, each prediction model can be created through the use of different machine learning algorithms or by training with different subsets of the input data. There are also other ways to join all of the individual models into a final model, rather than by simply averaging. One example is that the predictions from each individual model become the input features of a new machine learner. The advantage of this method would be that additional information can be used to aid the learning process.

Another way is to use more advanced deep network models. The vanishing gradient problem is a major barrier that impeded the development of deep neural networks in the past (Hochreiter 1998). Traditionally, networks are trained by the method of gradient descent based backpropagation, and often sigmoid or hyperbolic tangent activation functions are

used. The parameters (weights and biases) in previous layers are updated by multiplying the derivatives of activation functions of all the succeeding layers. Since all of the derivatives are small values (especially at the two sides of a sigmoid function), if the network is very deep, the weights and biases of the first several layers receive extremely small updates such that they can hardly be updated. New activation functions proposed in recent years, such as ReLU (Nair and Hinton 2010), ELU (Djork-Arne Clevert 2015), and SeLU (Klambauer et al. 2017) are able to alleviate the vanishing gradient problem. In Section 5.12 of the Results chapter, some primary results shows the adoption of ReLU activation functions might improve the contact prediction accuracy. However, the improvement of accuracy may also attribute to the use of more hidden layers. These results are limited to the amino acid contact predictions. In the future work, which of them contributes more and the amino acid distance prediction based on the same architecture should be explored.

The problem of vanishing gradient was further overcome by the introduction of the concept of the residual network (ResNet) (He et al. 2016). A ResNet is stacked by multiple building blocks. Each block uses a so-called skip connection that directly merges the input (the output of the previous block) to the output. Two types of building blocks were used in the study by (He et al. 2016), whose detailed architectures are shown in Figure 6.1. Before ResNet was proposed, networks included at most ≈ 10 hidden layers (He et al. 2016); conversely, ResNet can employ even more than 1,000 hidden layers (Zhang et al. 2017). With the ability to learn extreme abstract representations of objects, ResNet won

the first place of the image classification task in the 2015 ILSVRC (ImageNet Large Scale Visual Recognition Competition) (He et al. 2016). In 2016 and 2017, some variants of ResNet network models (e.g. DenseNet (Huang et al. 2016), ResNeXt (Xie et al. 2016) and Wide-ResNet (Zagoruyko and Komodakis 2016)) were proposed, and they were shown to be more accurate for image classification.

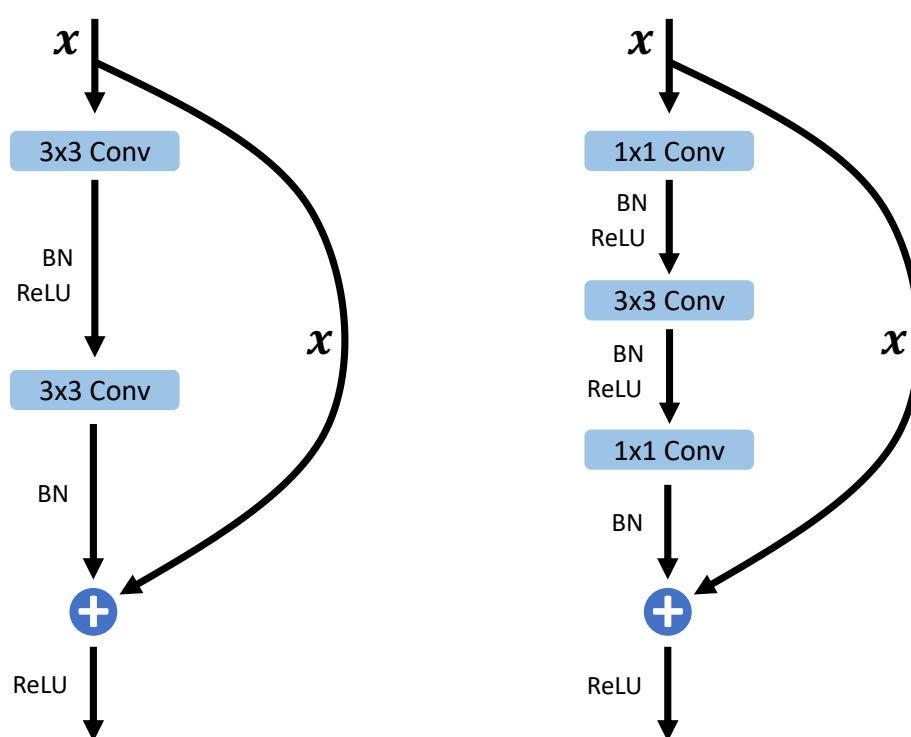


Figure 6.1. Two building block types of ResNet. Conv means a convolutional layer; 3×3 and 1×1 are the sizes (width, height) of the corresponding convolutional layers; BN means the batch normalization (Ioffe and Szegedy 2015) operation; ReLU is the ReLU activation function; ‘+’ means the add operation. This figure is reproduced from Figure 5 in (He et al. 2016).

Using the ResNet or its variants for contact/distance prediction should also be tested based on a modified feature vector. RaptorX has already used the ResNet. However, each

network in the ResNet is a 2D convolutional network, which is widely used in 2D image processing. For the features of the coupling matrices which are calculated from EVFold, QUIC and CCMpred, it is probably suitable to use a 2D convolutional network. However, other features, such as the secondary structure prediction and solvent accessibility, are essentially 1D arrays and so careful considerations about how to feed them into a 2D convolutional network are necessary. One idea is introduced below.

RaptorX uses a combination of 1D ResNet (i.e. the convolution filter is 1D) and 2D ResNet to predict amino acid contacts. The 1D ResNet is fed with sequential features, such as the sequence profile and secondary structure predictions. The output is transformed into a multi-channel 2D feature matrix, merged with other 2D features (e.g. the coupling predictions from CCMpred), and is then fed into the 2D ResNet. The transformation is performed in a way similar to the outer product (a detailed description can be found elsewhere ([Wang et al. 2017b](#))). Some ideas could be tested to improve the accuracies of amino acid contact and distance predictions based on the network model used in RaptorX. Since the variants of ResNet, such as ResNeXt ([Xie et al. 2016](#)) and Wide-ResNet ([Sergey Zagoruyko 2016](#)) could achieve a lower error rate on image classification, they might also make better contact/distance predictions than RaptorX. The 1D branch of RaptorX could additionally be replaced with a multi-layered bi-directional LSTM (long short term memory) recurrent network, which might capture longer couplings on the sequential signals.

For the above ResNet and its variants, different channels in the same convolutional layer

have the same weight. A new network block, termed SENet (Squeeze-and-Excitation Network) (Hu et al. 2017), could be embedded into ResNet and ResNeXt. It learns the different contributions of the convolutional features of the channels. By incorporating SENet in ResNet or ResNeXt, lower error rates are achieved on image classification (Hu et al. 2017). Thus, SENet based ResNet or ResNeXt could also be tested for contact and distance predictions.

By the time of writing this thesis, all of the published machine learning based methods of amino acid contact and distance predictions, including DeepCDpred, require some features used as inputs. These features are generally calculated from the MSA of the target protein sequence. An idea of the end-to-end learning should be tried to use the target sequence itself or the MSA of the sequence as the only input. Just like what the network models in the NLP (natural language processing) field do, an embedding layer that captures the biological, or more precisely, the evolutionary relations among different amino acid types (e.g. use the BLOSUM62 matrix as the weights of the embedding layer) are placed right after the input layer. The embedding layer is followed by multi-layered bidirectional LSTM, or 1D convolutional networks. The 1D output from the networks is then transformed into a 2D array either by following RaptorX's method, or just by repeating it along with a new dimension until the new dimension has the same size as the length of the sequence. The 2D array can be further fed into ResNet, ResNeXt, or SENet based ResNet or ResNeXt. The target is the contact or distance matrix.

The use of ROC (Receiver operating characteristic) curves may be useful to select the

best machine learning model. Take the number of neurons in the first hidden layer in DeepCDpred as an example: it is necessary to draw an ROC curve based on the true positives versus the false positives of contact and distance predictions for multiple values (e.g. 50, 80, 100, 120, 130). The best model is then chosen in conjunction with the number which produces the true-positive point versus the false-positive point that is closest to the top-left corner of the ROC plot. The problem with this method is that it requires changing the value of each parameter for multiple times. Since the time cost of training the neural networks in DeepCDpred is expensive, it could be helpful to select the best values for several parameters, but this is not feasible for all of the parameters on a CPU machine. However, training DeepCDpred on a multi-GPU platform may make it possible, since it is generally much faster to train a network model, especially a convolutional network, with a GPU than with a CPU. Multi-GPU could provide support to train the model in parallel, which makes the training process even faster.

6.10 The Online Server: [PROTEINCOEVOLUTION.BHAM.AC.](http://PROTEINCOEVOLUTION.BHAM.AC.UK)

[UK](http://PROTEINCOEVOLUTION.BHAM.AC.UK)

The website of proteincoevolution.bham.ac.uk was designed to provide an easy access to DeepCDpred contact/distance and structure predictions. It employs two Linux machines from the Centre for Computational Biology at the University of Birmingham in Birmingham, UK for the calculations. Since students in the centre are also using the machines

for their projects, the website has to limit the access to the users who want to predict protein structures – notably, structure predictions require more computing resources than contact and distance predictions by DeepCDpred. An idea to solve this problem is to use cloud-based servers, such as the EC2 of AWS (Amazon Web Services). Again, since protein structure predictions require many CPU powers, the price of this service is beyond the budget of the laboratory at this time.

CHAPTER 7

CONCLUSION

This thesis introduces the work of amino acid contact/distance prediction based on the method proposed in this thesis, DeepCDpred, and how it was used as the geometry constraint for the long-standing computational biology problem of protein structure prediction. The feature contribution analysis shows that at least for the amino acid contact prediction, coevolutional couplings calculated from CCMpred play the most important role in the neural network model of DeepCDpred. In addition, in order to estimate the quality of the structure prediction, a TM-score prediction method was proposed. Compared with other algorithms, the accuracy of the amino acid contact prediction of DeepCDpred is just slightly worse than a newly published method, RaptorX, but exceeds all others mentioned in this thesis.

For fairness, the structure predictions based on the predicted constraints from DeepCDpred and those based on the predicted constraints from MetaPSICOV, were compared

using two constraint selection methods. The result showed that the conventional method of selecting the top-ranked 1.5L contacts has no difference with that of selecting the contacts based on a score cut-off for protein structure prediction. The additional distance prediction from DeepCDpred improved structure prediction compared with that based only on the contact prediction from DeepCDpred. A blind test also proved constraints predicted by DeepCDpred were more effective than MetaPSICOV for structure prediction.

The adoption of strategies of mini-batch, stochastic gradient descent and ReLU activation functions even made the amino acid contact prediction of DeepCDpred more accurate. The feature contribution analysis revealed that couplings calculated from CCMpred and the secondary structure prediction from SPIDER2 are the top 2 features for the performance of amino acid contact prediction of DeepCDpred. future work that improves the quality of them may thus improve the contact prediction accuracy of DeepCDpred.

Based on the analysis in the Discussion chapter (Chapter 6), some future work which uses the metagenome sequence data and more advanced deep learning models, might improve the accuracy of amino acid contact/distance prediction of DeepCDpred and may thus result in better protein structure prediction. In addition, model quality assessment methods, such as ModFold6, should be tested to select the best-predicted structure, rather than just choose the one with the lowest Rosetta energy score used in this thesis. Also, methods introduced by Roy *et al.* and Xu *et al.* (Roy *et al.* 2011; Xu and Zhang 2013) should be tried to predict a more accurate confidence score for structure prediction.

An online server, <http://proteincoevolution.bham.ac.uk>, was programmed and released to make the algorithms of amino acid contact and distance predictions, structure prediction, and TM-score prediction accessible to average users, which may be beneficial to the research community.

PUBLICATIONS

Ji, S., Oruc, T., Mead, L., Rehman, M. F., Thomas, C. M., Butterworth, S., and Winn, P. J. (2019). Deep CDpred: Inter-residue distance and contact prediction for improved prediction of protein structure. *PLoS ONE*, 14(1):e0205214.

Rajasekar, K. V., Ji, S., Coulthard, R. J., Ride, J. P., Reynolds, G. L., Winn, P. J., Wheeler, M. J., Hyde, E. I., Smith, L. J., Coulthard-Graf, R., and Heidelberg, E. (2019). Structure of SPH (Self-Incompatibility Protein Homologue) proteins, a widespread family of small, highly stable, secreted proteins. *Biochemical Journal*, 476(5):809-826.

Appendices

APPENDIX A

SUPPLEMENTARY MATERIALS

Table A.1. Mappings of converting amino acids to numbers.

Amino Acid	Number
A	0
R	1
N	2
D	3
C	4
Q	5
E	6
G	7
H	8
I	9
L	10
K	11
M	12
F	13
P	14
S	15
T	16
W	17
Y	18
V	19
- #	20

#: gap.

Table A.2. The positions of the nine beta strands in the experimental structure of Q9FLY6. The positions are determined by observing the experimental structure with PyMol ([DeLano 2002](#)).

Strand Index	Residue Index
1	4 5 6 7 8 9 10 11
2	17 18 19 20 21 22 23
3	32 33 34 35
4	40 41 42 43 44
5	55 56 57 58 59 60
6	66 67 68 69 70 71
7	86 87 88 89 90 91
8	95 96 97 98
9	106 107 108

APPENDIX B

SUPPLEMENTARY RESULTS

B.I Sequence Identity Distribution Between the Test Protein Chains & the Training/Validation Protein Chains

Figure [B.1](#) shows distributions of the pairwise sequence identities of the 108 test protein chains and the 1066 training/validation protein chains of DeepCDpred. The mean and standard deviation of the pairwise sequence identities is 12.4%, and 3.6%, respectively.

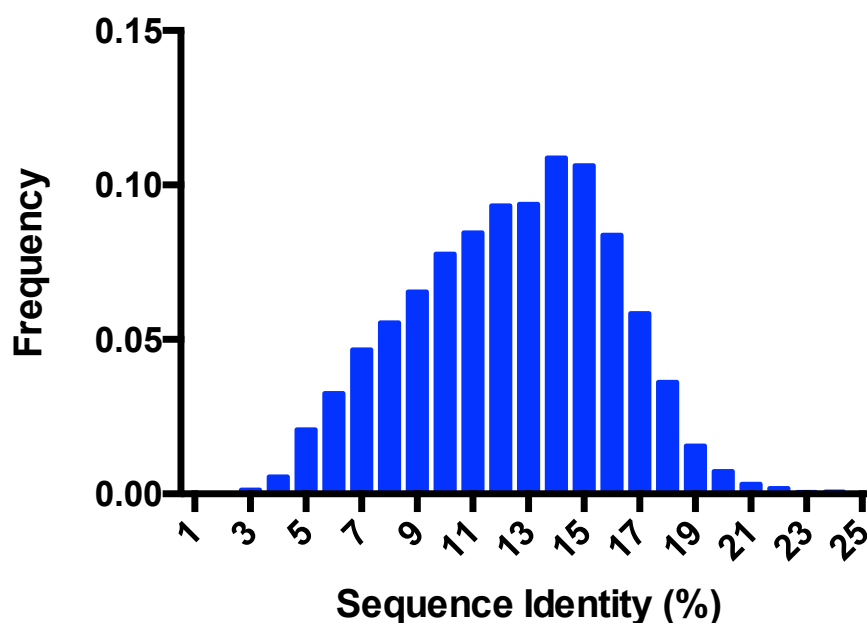


Figure B.1. Distribution of pairwise sequence identity between the 108 test protein chains and the 1066 training/validation protein chains of DeepCDpred.

B.II Parameter Optimization of DeepCDpred

This section shows the result of the amino acid contact prediction accuracy comparison between the optimized two-hidden-layer network of DeepCDpred with 120 and 50 neurons in the hidden layers and a two-hidden-layer network with 100 and 30 neurons in the hidden layers. Figure B.2 shows the stage 2 results of the comparisons. For the top ranked 1.5L predictions, the choosing of 120 and 50 neurons makes 0.7% higher contact prediction accuracy than with 100 and 30 neurons for the 108 test proteins. It is worth noting that

both the two networks were trained with the contact range defined as $0 - 8\text{\AA}$ and the 1066 proteins in the training/validation set.

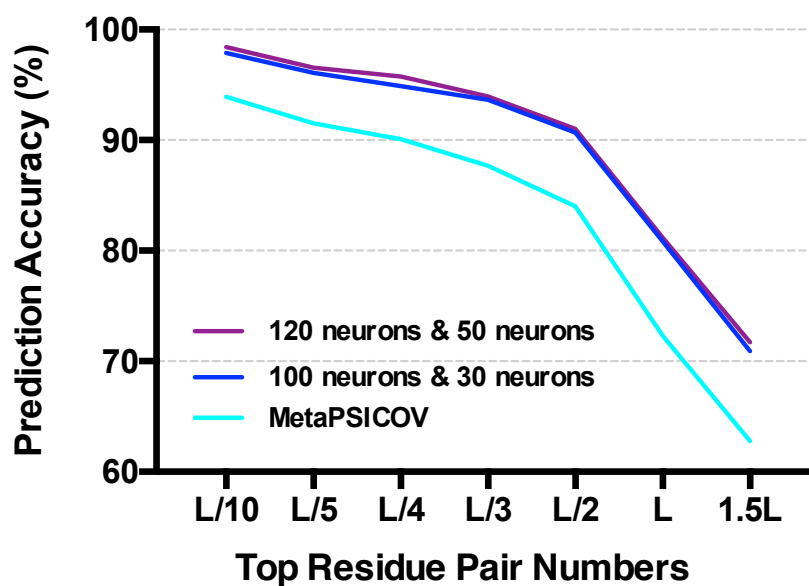


Figure B.2. Amino acid contact prediction accuracy comparison between the optimized two-hidden network with 120 and 50 neurons of DeepCDpred and the two-hidden-layer network with the numbers of neurons replaced with 100 and 30. The result of MetaPSICOV for the same proteins are used as a reference.

B.III Comparisons of Contact Prediction Accuracy, Model File Size and Average Contact Prediction Speed between DeepCDpred, SVM and Random Forest

DeepCDpred, an SVM and a random forest model were trained with the same inputs from the same set of proteins (435 proteins, PDB IDs are listed in Table C.5). The inputs were introduced in Chapter 3. The SVM model and the random forest model were trained with the functions, “fitcsvm” and “TreeBagger”, respectively, from the machine learning toolbox of MATLAB. Parameter settings for the SVM training include kernel, rbf; kfold, 2 (2-fold cross-validation). Other parameters were chosen as the default. Parameter settings for the training of the random forest model include the number of decision trees, 10; kfold, 2 (2-fold cross-validation). Other parameters were also chosen as the default.

The results of comparisons of the amino acid contact prediction accuracy, the model file size and the average prediction speed between DeepCDpred, the SVM and the random forest model are shown in Figure B.3.

From the figure, DeepCDpred makes significantly better amino acid contact predictions, but uses much less disk space than both the SVM model and the random forest model. The process of making predictions of DeepCDpred is also faster. Although the parameters

of the SVM and the random forest model are not optimized through adjusting, the obvious advantages of DeepCDpred pushed the author of this work to choose DeepCDpred as the algorithm for amino acid contact/distance prediction.

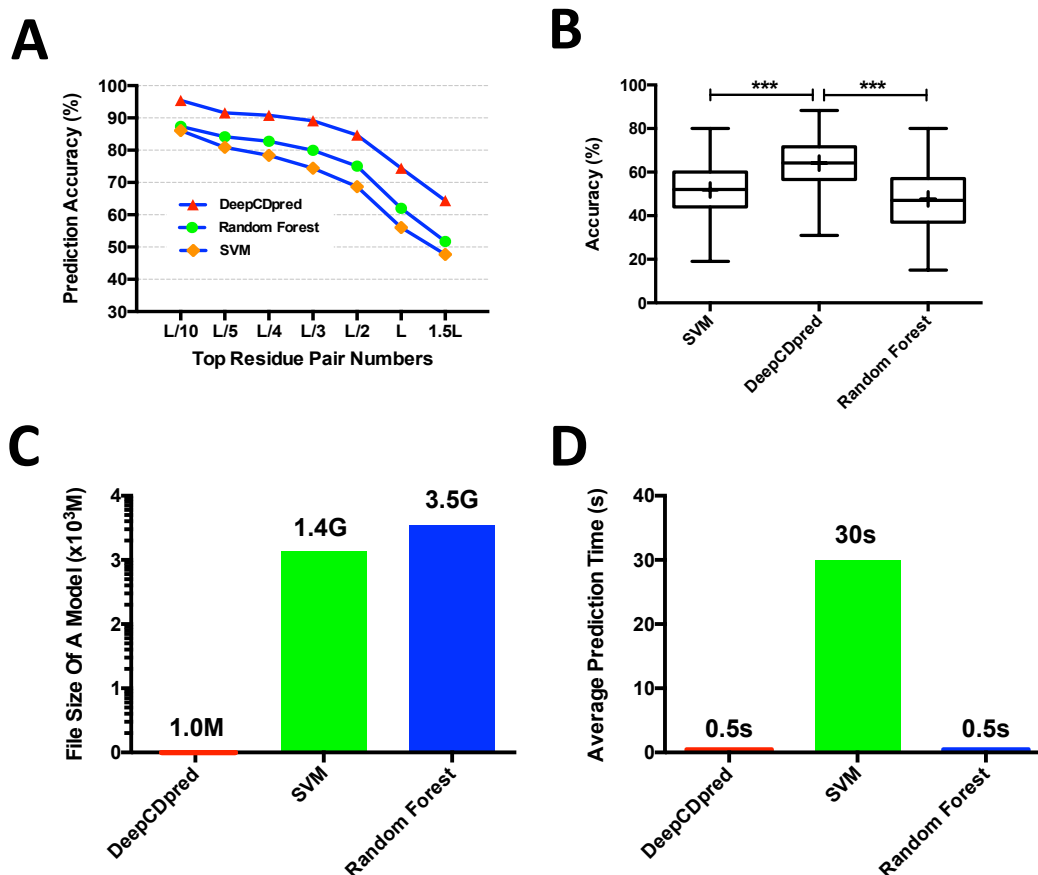


Figure B.3. Comparisons of amino acid contact prediction accuracy, model file size and average prediction speed between DeepCDpred, an SVM and a random forest model. A, amino acid contact prediction accuracy comparisons for the top L/10, L/5, L/4, L/3, L/2, L and 1.5L predictions between the three algorithms; B, the accuracy of the top 1.5L amino acid contacts predicted by DeepCDpred is significantly higher than both that predicted by the SVM and the random forest model; C, the file size of a neural network model of DeepCDpred is only about 1/1000 of the SVM, or the random forest model (here, M means megabytes; G means gigabytes); D, when making predictions, DeepCDpred only takes 1/60 time comparing to the SVM on average for the 108 test proteins, and almost the same time comparing to the random forest model on average based on the same test proteins.

B.IV Structure Selected by Lowest Rosetta Energy VS. True Best Structure

In Figure B.4a, the TM-score of the structure selected by the lowest Rosetta energy is compared with the TM-score of the predicted structure with the highest TM score for each protein chain in the test set of DeepCDpred (true best structure). The scatter plot in graph a shows that the latter TM-score is always no smaller than the former. Graph b is the box plot of graph a. In Figure B.4b, whiskers indicate the minimum and maximum TM-score values in each group; middle lines in the boxes are the median values and the crosses represent the two means. The means, medians, and standard deviation of the two groups are 0.69, 0.12 and 0.71, 0.72, 0.11 and 0.74, respectively.

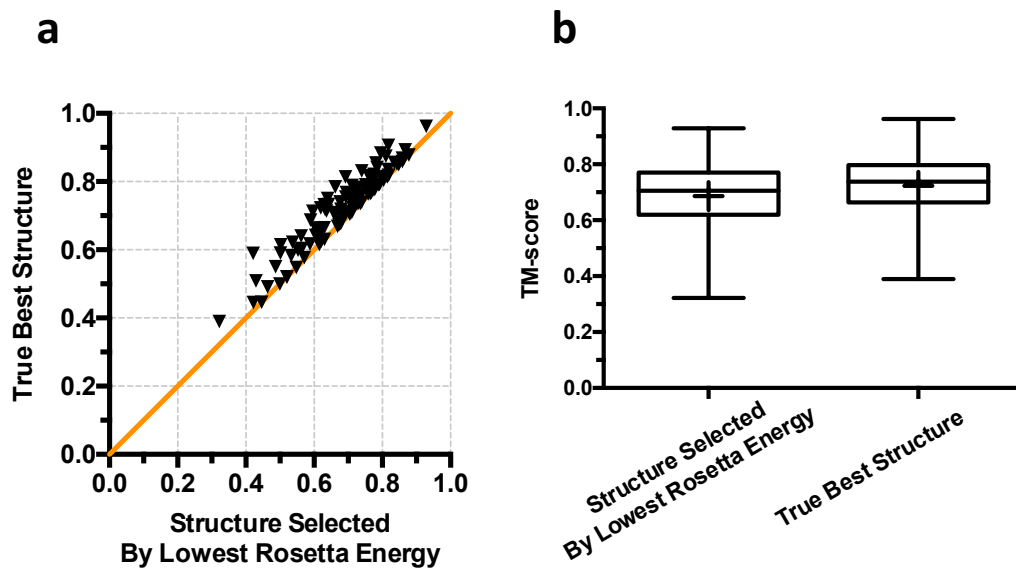


Figure B.4. The difference between the best structure selected by the lowest Rosetta energy score and the true best structure among the 100 candidates which is picked out by comparing with the experimental structure. (a), scatter plot of the comparison; each triangle represents one protein in the test set and majority proteins have better structure prediction when using top 1.5L DeepCDpred predicted contacts as constraints. (b), boxplots of the comparison in (a); whiskers indicate the minimum and maximum TM-score values in each group; middle lines in the boxes are the median values and the crosses represent the two means.

B.V Accuracy of Amino Acid Contact Prediction After Adopting A New Feature Vector

A modification of the feature vector of DeepCDpred was tried. The new feature vector was constructed based on the feature contribution analysis of the original feature vector, which was already introduced in section 5.5.5 (Chapter 5). In detail, the amino acid profiles were removed from the original feature vector; for both the coevolutional couplings calculated from EVfold and QUIC, like CCMpred, a square window of size 9×9 was used for each to include all the neighbouring couplings in the window. Finally, the new feature vector has 429-dimension.

The stage 1 neural networks for contact prediction in DeepCDpred were trained with the same parameter settings and architecture as introduced in section 5.12 (Chapter 5). The result is shown in Figure B.5. The accuracy of the top ranked 1.5L contact predictions from the new networks is 3.6% higher than the original ones (68.6% versus 72.2%).

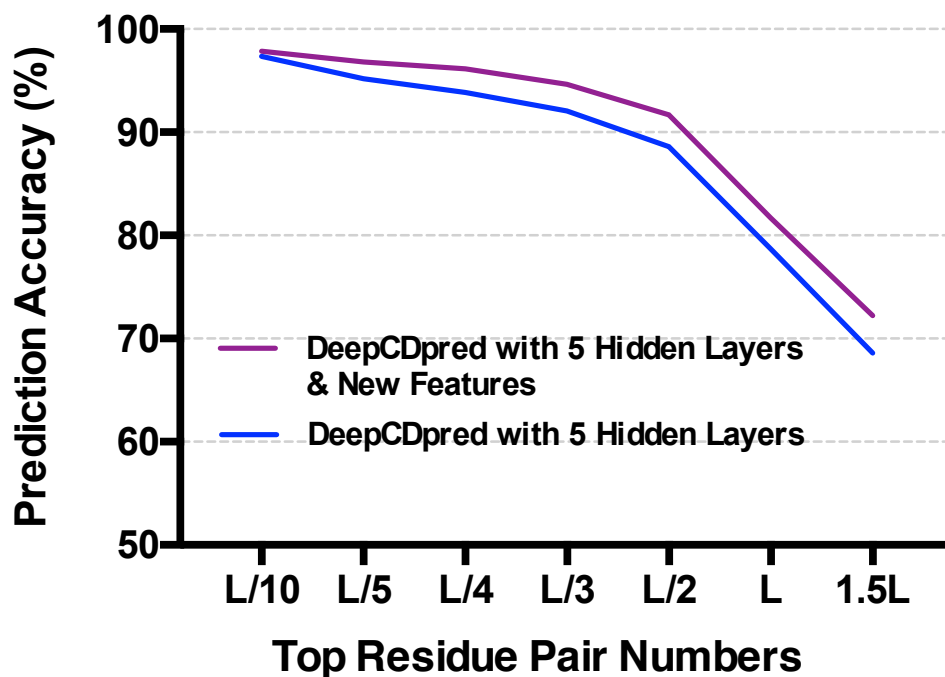


Figure B.5. Contact prediction accuracy comparison between the five-hidden-layer neural networks of DeepCDpred and the modified five-hidden-layer neural networks with new feature vector.

B.VI Raw Data of Figure 5.17, Figure 5.19 and Figure 5.21

Table B.1. Raw TM-scores of the boxplots shown in Figure 5.17. Each row corresponds to the same protein; different rows represent different proteins.

TM-score of MetaPSICOV	TM-score of DeepCDpred
0.46055	0.56607
0.66830	0.76727
0.70738	0.80632
0.70196	0.79862
0.70137	0.79583
0.43286	0.52488
0.63399	0.72525
0.71984	0.80990

0.70558	0.79221
0.48756	0.57128
0.62266	0.70447
0.62818	0.70728
0.57244	0.65109
0.65748	0.73576
0.63519	0.70865
0.45888	0.53201
0.71414	0.78226
0.74741	0.81354
0.68015	0.74466
0.75445	0.81658
0.59881	0.66072
0.53171	0.58903
0.62128	0.67824
0.64982	0.70600
0.60980	0.66530
0.64879	0.69893
0.41924	0.46774
0.80294	0.85084
0.58630	0.62854
0.58444	0.62629
0.77632	0.81622
0.81154	0.85049
0.66391	0.69942
0.84098	0.87524
0.68448	0.71738
0.73532	0.76251
0.70592	0.73238
0.76634	0.79279
0.70100	0.72299
0.72770	0.74794
0.48628	0.50579
0.68577	0.70449
0.72714	0.74476
0.81246	0.82864
0.57873	0.59351
0.77281	0.78526
0.73007	0.74176
0.73452	0.74430
0.55755	0.56599
0.73119	0.73589
0.76427	0.76711
0.79192	0.79437
0.65727	0.65844
0.57229	0.57285
0.69152	0.68420
0.77108	0.76197
0.70741	0.69153
0.76190	0.74513
0.56188	0.54339
0.62619	0.60522
0.60755	0.57721

0.68655	0.65555
0.84550	0.80861
0.57500	0.53770
0.54546	0.50721
0.76932	0.73057
0.54125	0.50033
0.66958	0.62796
0.45717	0.41076
0.77391	0.72194
0.76639	0.71180
0.72005	0.66436
0.69474	0.62871
0.53001	0.45918
0.68331	0.60992
0.58272	0.50103
0.55923	0.46555

Table B.2. Raw TM-scores of the boxplots shown in Figure 5.19. Each row corresponds to the same protein; different rows represent different proteins.

TM-score of MetaPSICOV	TM-score of DeepCDpred
0.64608	0.71686
0.30478	0.51435
0.66640	0.64066
0.63931	0.58448
0.53604	0.51316
0.64105	0.78702
0.56911	0.58696
0.78895	0.77715
0.50825	0.61806
0.45478	0.44142
0.61283	0.78301
0.61398	0.74975
0.50015	0.62583
0.61859	0.57368
0.74702	0.84330
0.73033	0.66833
0.56777	0.71713
0.65097	0.66560
0.57887	0.65848
0.63665	0.66275
0.69442	0.66943
0.65066	0.72063
0.43069	0.48893
0.29097	0.55430
0.69970	0.76561
0.33301	0.53817
0.59962	0.65628
0.59130	0.67080
0.52923	0.47052
0.55952	0.59391
0.59133	0.59258

0.49065	0.51589
0.56292	0.63045
0.60671	0.66008
0.59271	0.67021
0.51614	0.58612
0.42564	0.51485
0.75657	0.82521
0.75177	0.73690
0.49060	0.61019
0.70375	0.72412
0.74286	0.79312
0.46986	0.58532
0.35762	0.41916
0.66371	0.70114
0.61844	0.68788
0.59845	0.67016
0.30313	0.34556
0.73119	0.77622
0.69330	0.79319
0.53768	0.52887
0.67809	0.74766
0.52694	0.71416
0.68876	0.75976
0.78725	0.71625
0.67494	0.67776
0.73035	0.77161
0.75804	0.72649
0.80089	0.81480
0.64888	0.64096
0.57984	0.53304
0.64202	0.71987
0.80959	0.81252
0.73439	0.66205
0.65363	0.73008
0.43980	0.55673
0.67661	0.76355
0.62871	0.84068
0.52264	0.68670
0.59910	0.64535
0.50516	0.76377
0.63637	0.65905
0.70156	0.79188
0.60232	0.71224
0.75836	0.85832
0.63228	0.64843
0.85859	0.84238
0.53871	0.60878
0.75553	0.73378
0.58961	0.52434
0.82806	0.87707
0.40824	0.65442
0.91935	0.91494
0.68174	0.74374

0.37870	0.62999
0.75335	0.69005
0.52639	0.60780
0.72911	0.79646
0.64958	0.77354
0.46522	0.46744
0.65107	0.65904
0.57608	0.61912
0.68546	0.61571
0.67263	0.73058
0.66638	0.75466
0.63636	0.68040
0.79872	0.84365
0.56524	0.62361
0.78769	0.85909
0.50669	0.39514
0.51905	0.47359
0.71065	0.71647
0.62581	0.60646
0.65714	0.70721
0.77816	0.85550
0.57152	0.57250
0.73229	0.69851
0.59606	0.55114

Table B.3. Raw TM-scores of the boxplots shown in Figure 5.21. Each row corresponds to the same protein; different rows represent different proteins.

TM-score of MetaPSICOV	TM-score of DeepCDpred
0.75836	0.85832
0.69330	0.79319
0.74702	0.84330
0.70156	0.79188
0.42564	0.51485
0.66638	0.75466
0.67661	0.76355
0.52639	0.60780
0.57887	0.65848
0.59130	0.67080
0.64202	0.71987
0.59271	0.67021
0.77816	0.85550
0.65363	0.73008
0.59845	0.67016
0.78769	0.85909
0.68876	0.75976
0.64608	0.71686
0.53871	0.60878
0.51614	0.58612
0.65066	0.72063
0.67809	0.74766
0.61844	0.68788

0.75657	0.82521
0.56292	0.63045
0.72911	0.79646
0.69970	0.76561
0.68174	0.74374
0.35762	0.41916
0.56524	0.62361
0.43069	0.48893
0.67263	0.73058
0.59962	0.65628
0.60671	0.66008
0.74286	0.79312
0.65714	0.70721
0.82806	0.87707
0.59910	0.64535
0.73119	0.77622
0.79872	0.84365
0.63636	0.68040
0.57608	0.61912
0.30313	0.34556
0.73035	0.77161
0.66371	0.70114
0.55952	0.59391
0.63665	0.66275
0.49065	0.51589
0.63637	0.65905
0.70375	0.72412
0.56911	0.58696
0.63228	0.64843
0.65097	0.66560
0.80089	0.81480
0.65107	0.65904
0.71065	0.71647
0.80959	0.81252
0.67494	0.67776
0.46522	0.46744
0.59133	0.59258
0.57152	0.57250
0.91935	0.91494
0.64888	0.64096
0.53768	0.52887
0.78895	0.77715
0.45478	0.44142
0.75177	0.73690
0.85859	0.84238
0.62581	0.60646
0.75553	0.73378
0.53604	0.51316
0.69442	0.66943
0.66640	0.64066
0.75804	0.72649
0.73229	0.69851
0.61859	0.57368

0.59606	0.55114
0.51905	0.47359
0.57984	0.53304
0.63931	0.58448
0.52923	0.47052
0.73033	0.66833
0.75335	0.69005
0.58961	0.52434
0.68546	0.61571
0.78725	0.71625
0.73439	0.66205

APPENDIX C

TRAINING/VALIDATION, AND TEST SET

In this chapter, all PDB ID and chain names of the proteins used in the amino acid contact percentage calculation, in the training/validation set and the test set used for DeepCDpred, and the TM-score prediction neural network model are listed. For all the three following tables, the entries of PDB ID and chain name are separated by commas; the four characters represent the PDB ID, and the fifth character is the chain name.

Table C.1. PDB ID list of the 250 protein chains in the amino acid contact percentage calculation.

PDB ID/CHAIN NAME
4wzxE, 4axyA, 4w8pB, 4reyB, 4wndB, 4zgmB, 4lkuA, 1n0wB,
4wolA, 4ub8R, 3wwqC, 4txrC, 1htrP, 1nh2B, 4rkhC, 4zceB,
1i7wB, 4od8D, 4x86B, 3h6pA, 1isuA, 4r2yA, 2yh9A, 1gu4A,
1xiwB, 1eayC, 1whzA, 4qxbB, 1tafA, 4cvoA, 4rgdA, 1tafB,
1egwA, 4uafE, 1uv7A, 3e19A, 1hq1A, 1dp7P, 1oeyA, 1n7sC,
4nc7A, 3w61A, 1d3bB, 4zeyA, 4tpsB, 1r6jA, 4m1gA, 1w53A,
1wlzA, 3n5bB, 4cayA, 1wv9A, 1t0pB, 3mhsE, 4lwsB, 3kojA,
4csrA, 1o5uA, 1nkpA, 1q8bA, 4xa1A, 1xiwA, 1b9wA, 1uw4A,
4ku0A, 1bteA, 3dwgC, 4zv0B, 4u7iA, 1q08A, 1dfuP, 3dluA,
4z8tB, 3glaA, 4gneA, 3lwcA, 4qkwA, 3f8bA, 1lxjA, 4zhbA,
4czxB, 3kmaA, 1nlqA, 1xlqA, 4hlyA, 1hxrA, 4nn5A, 1p57A,
1r75A, 3u2aA, 1n13B, 3ju3A, 4otmA, 3kdfD, 4kqdA, 2zqmA,
4ue0A, 4rleA, 1vkeA, 3lyxA, 1sd4A, 3h7hA, 2a0bA, 1u5fA,
4qblA, 1t1jA, 1fc3A, 2xdpA, 1gy7A, 4mypA, 3bedA, 4tshA,
4bhUA, 1lr0A, 4h87A, 1ez3A, 4ounA, 1y7rA, 2yqyA, 3f8xA,
1ixlA, 1s5uA, 3hsrA, 1ogdA, 1od3A, 3dlqI, 1wvhA, 4zdsA,
1vsrA, 4htuA, 1n12A, 3do8A, 3oopA, 1z1sA, 1ccwA, 4mtmA,
4p5eA, 1jmvA, 3ht1A, 1fm0E, 1r0uA, 4mtuA, 1f2tB, 1h6hA,
1kxgA, 4rptA, 1yocA, 3v4gA, 4un1A, 1f2tA, 4pp8C, 1idpA,
1dzkA, 4zeqA, 3eytA, 4u5rA, 1v4pA, 4gqzA, 3vygB, 3l2hA,
4v3iA, 1t9iA, 1l6A, 1guiA, 1g5tA, 1rylA, 4lzkA, 3anoA,
1np6A, 1y9iA, 1sz7A, 4okeA, 4kt6B, 4bi3A, 1u7pA, 4llyA,
3nl9A, 3ktaB, 1gheA, 1ynbA, 1g12A, 4yp6A, 4mi4A, 1gu9A,
4qm6A, 1fpoA, 1cczA, 3v6gA, 4w4kB, 1xg0C, 1g3kA, 1rttA,
4yz6A, 3q3jB, 4x2hB, 4hiaA, 4aciA, 1wubA, 4tq2A, 3wisA,
4p82A, 4jj0A, 1jh6A, 4bjaA, 4bu0A, 3shoA, 2zfdA, 1h6fA,
3rnqB, 3vp5A, 4yepA, 3fxaA, 1qqp1, 3u3zA, 1n9pA, 4lviA,
1pp0A, 1v2xA, 2vzyA, 1ui0A, 4xbaB, 1nxmA, 2xblA, 3r5gA,
3l8dA, 4nn5B, 3rnrA, 4g6iA, 2w7zA, 2xbuA, 1jm1A, 1l6rA,
1lfpA, 1kzqA, 1n57A, 1m6yB, 1kpgA, 1in1A, 1mnnA, 1lc0A,
1k8wA, 1ntyA, 1jl0A, 1n7zA, 1l7aA, 1ixhA, 1j4aA, 1juhA,
1kq3A, 1mtyB

Table C.2. PDB ID list of the 221 protein chains in the speed and contact prediction accuracy comparisons between PSICOV and QUIC.

PDB ID/CHAIN NAME
1a6mA, 1a70A, 1abaA, 1ag6A, 1aoeA, 1atzA, 1avsA, 1bebA,
1behA, 1bkrA, 1brfA, 1bsgA, 1c44A, 1c52A, 1cc8A, 1chdA,
1cjlA, 1ckeA, 1cxyA, 1cznA, 1d1qA, 1d4oA, 1dbxA, 1dixA,
1dlwA, 1dmgA, 1dqgA, 1e5kA, 1eaqA, 1eazA, 1eb6A, 1ej0A,
1ej8A, 1fcyA, 1fk5A, 1fl0A, 1fnaA, 1fvga, 1g2rA, 1g61A,
1g9oA, 1gbsA, 1gmiA, 1gmxA, 1gqvA, 1gu2A, 1guuA, 1gz2A,
1h0pA, 1h12A, 1h2eA, 1h4gA, 1h4xA, 1h98A, 1hfcA, 1hh8A,
1htwA, 1hxnA, 1i1jA, 1i1nA, 1i27A, 1i5gA, 1i71A, 1ihzA,
1iibA, 1im5A, 1iwdA, 1j3aA, 1jbeA, 1jbkA, 1jfuA, 1jfxA,
1jl1A, 1jo0A, 1jo8A, 1josA, 1jvwA, 1jwqA, 1jyhA, 1k5cA,
1k6kA, 1k7cA, 1k7jA, 1ka1A, 1kidA, 1kmtA, 1kq6A, 1kqrA,
1ktgA, 1ku3A, 1kw4A, 1l9lA, 1lm4A, 1lniA, 1lpyA, 1lwbA,
1lyvA, 1m1qA, 1m4jA, 1m55A, 1m8aA, 1mk0A, 1mn8A, 1mugA,
1muwA, 1n8vA, 1nb9A, 1npsA, 1nuyA, 1ny1A, 1nz0A, 1o1zA,
1o2dA, 1o4yA, 1o7qA, 1odmA, 1oh4A, 1p90A, 1pbjA, 1pchA,
1pkoA, 1qf9A, 1qtwA, 1r26A, 1r85A, 1roaA, 1rw1A, 1rybA,
1sauA, 1svyA, 1t8kA, 1tifA, 1tqgA, 1tqhA, 1tt8A, 1tzvA,
1ucsA, 1vfyA, 1vhuA, 1vjkA, 1vlyA, 1vmbA, 1vykA, 1w0hA,
1w0nA, 1w1hA, 1w66A, 1wc2A, 1wcwA, 1wdpA, 1wjxA, 1wkcA,
1xbiA, 1xdnA, 1xdzA, 1xkrA, 1xmka, 1xmtA, 1xqoA, 1yfqA,
1yiiA, 1z0nA, 1zgjA, 1zzkA, 2b3hA, 2b97A, 2bkxA, 2c4bA,
2c60A, 2cb8A, 2cg7A, 2ciwA, 2ckkA, 2cs7A, 2cuaA, 2endA,
2erfA, 2fb6A, 2fbaA, 2fcjA, 2fmaA, 2fsqA, 2gkeA, 2gkpA,
2gqtA, 2gsoA, 2h1vA, 2hbaA, 2hzcA, 2i5vA, 2ia7A, 2iayA,
2ic6A, 2j5yA, 2j8bA, 2jekA, 2jliA, 2lisA, 2mhrA, 2nr7A,
2nszA, 2ofzA, 2ovjA, 2p0nA, 2p51A, 2phyA, 2pneA, 2q52A,
2qfeA, 2qjzA, 2qngA, 2qvka, 2r16A, 2r31A, 2r75A, 2rctA,
2rdqA, 2rffA, 2tpsA, 2v9vA, 2ve8A

Table C.3. PDB ID list of the test set of DeepCDpred.

PDB ID/CHAIN NAME
1a3aA, 1cc8A, 1dsxA, 1gzcA, 1im5A, 1ku3A, 1p90A, 1vjkA, 1aapA, 1chdA, 1eazA, 1h2eA, 1j3aA, 1kw4A, 1pchA, 1vmbA, 1abaA, 1cjwA, 1ej8A, 1h4xA, 1jfuA, 1lm4A, 1qf9A, 1vp6A, 1ag6A, 1ckeA, 1f6bA, 1hdoA, 1jl1A, 1lo7A, 1qjpA, 1w0hA, 1aoeA, 1ctfA, 1fcyA, 1hfcA, 1jo0A, 1m4jA, 1r26A, 1whiA, 1atzA, 1cxyA, 1fk5A, 1hh8A, 1jo8A, 1m8aA, 1roaA, 1wjxA, 1avsA, 1cznA, 1f10A, 1htwA, 1josA, 1mk0A, 1rw1A, 1wkcA, 1bdoA, 1d0qA, 1fvgaA, 1hxnA, 1jwqA, 1mugA, 1smxA, 1xffA, 1bebA, 1d1qA, 1fx2A, 1i1jA, 1jyhA, 1nb9A, 1svyA, 2cuaA, 1behA, 1d4oA, 1g2rA, 1i1nA, 1k6kA, 1ne2A, 1t8kA, 2phyA, 1bkrA, 1dixA, 1g9oA, 1i4jA, 1k7jA, 1npsA, 1tifA, 1c44A, 1dlwA, 1gmiA, 1i58A, 1kq6A, 1nrvA, 1tqgA, 1c52A, 1dmgA, 1gmxA, 1i71A, 1kqrA, 1ny1A, 1tqhA, 1c9oA, 1dggA, 1gz2A, 1iibA, 1ktgA, 1o1zA, 1vfyA

Table C.4. PDB ID list of the training/validation set of DeepCDpred.

PDB ID/CHAIN NAME
1a62A, 1cukA, 1dvoA, 1f3uB, 1gprA, 1hxiA, 1ixlA, 1juvA, 1lf7A, 1mvlA, 1nnxA, 1oi0A, 1pocA, 1qkrA, 1rssA, 1sgwA, 1tc5A, 1u2hA, 1uujA, 1vi6A, 1wmhA, 1x2iA, 1y63A, 1ae9A, 1cv8A, 1dwkA, 1f46A, 1gs9A, 1hxrA, 1izmA, 1jyaA, 1lkiA, 1my7A, 1np6A, 1on2A, 1pp0A, 1qqp1, 1rttA, 1sh8A, 1tfeA, 1u2wA, 1uuyA, 1vimA, 1wmhB, 1x3kA, 1y7rA, 1alyA, 1cxqA, 1dxgA, 1f60B, 1gu4A, 1hztA, 1j0pA, 1jyoA, 1lkkA, 1mzwB, 1nqzA, 1oo0B, 1pqhA, 1qv1A, 1rutX, 1sj1A, 1th7A, 1u5fA, 1uv7A, 1vj1A, 1wmxA, 1x6oA, 1y88A, 1ayoA, 1cy5A, 1dzkA, 1fc3A, 1gu9A, 1i07A, 1j24A, 1k3sA, 1lqvA, 1n0wB, 1nrjB,

1oqjA,1psrA,1qw2A,1rxdA,1sqwA,1tigA,1u7kA,1uw4A,
1vkeA,1wocA,1x91A,1y9iA,1b4fA,1d2oA,1e0bA,1fltX,
1guiA,1i12A,1j27A,1k4nA,1lr0A,1n12A,1ntvA,1orsC,
1pt6A,1qwdA,1rxqA,1ss4A,1tiqA,1u7pA,1uxoA,1vkiA,
1wolA,1xauA,1y9wA,1b9wA,1d2sA,1e30A,1fm0E,1gutA,
1i2tA,1j3wA,1k8kE,1lr5A,1n13B,1nu0A,1oruA,1pvmA,
1qzgA,1rylA,1sviA,1tjlA,1u84A,1v0aA,1vkkA,1wouA,
1xe1A,1yd9A,1bgcA,1d2zB,1eayC,1fpoA,1gxuA,1i4uA,
1j77A,1k8kG,1lshB,1n1fA,1nxmA,1ou8A,1pyoB,1qzmA,
1ryqA,1sz7A,1tp6A,1u9kA,1v2xA,1vl7A,1wpaA,1xe7A,
1ynbA,1bm8A,1d3bA,1ef1C,1fpzA,1gy7A,1i7wB,1j7dA,
1kgdA,1lu4A,1n62A,1nznA,1ow1A,1pzwA,1r0dA,1rz3A,
1t07A,1ts9A,1u9lA,1v4pA,1vmgA,1wpbA,1xfSA,1yocA,
1bm9A,1d3bB,1egwA,1g12A,1h2sB,1i8aA,1j8bA,1khyA,
1luzA,1n71A,1o13A,1ow4A,1q08A,1r0uA,1s12A,1t0pB,
1tu1A,1uebA,1v5iB,1vqsA,1wrda,1xg0C,1z1sA,1bteA,
1d4tA,1elkA,1g3kA,1h4aX,1id0A,1je5A,1klxA,1lxjA,
1n7sC,1o4wA,1oz9A,1q0pA,1r6jA,1s29A,1t1jA,1tu9A,
1ui0A,1v6pA,1vsrA,1ws8A,1xhdA,1zavA,1btkA,1ddwA,
1elwA,1g5tA,1h6fA,1idpA,1jf3A,1koeA,1ly1A,1n9pA,
1o50A,1p57A,1q1fA,1r75A,1s3cA,1t3yA,1tuaA,1ujcA,
1v74B,1w2wB,1wu9A,1xhnA,1ze3H,1bxyA,1dfuP,1epfA,
1g6gA,1h6hA,1ifrA,1jh6A,1kptA,1m0dA,1ng2A,1o5uA,
1p6oA,1q40B,1r77A,1s4kA,1t4aA,1tuhA,1ukka,1v96A,
1w4sA,1wubA,1xiwA,1zpvA,1byfA,1dg6A,1ew4A,1g8eA,
1h8pA,1igqA,1jhgA,1kt6A,1m1fA,1ng6A,1o6dA,1p9gA,
1q42A,1r7jA,1s5uA,1t6sA,1tuvA,1unnC,1v9yA,1w53A,

1wurA,1xiwB,1zuoA,1byrA,1dj8A,1ez3A,1g8qA,1h97A,
1io0A,1jhjA,1kxgA,1m4iA,1nh2B,1o7iA,1p9hA,1q8bA,
1r9wA,1s7iA,1t6t1,1tuwA,1uptB,1vbwA,1wdcB,1wv9A,
1x1qA,2a0bA,1c1yB,1dk8A,1f1mA,1gheA,1hfeS,1iq4A,
1jhsA,1kxoA,1m70A,1nh2C,1oa8A,1pbwA,1q8dA,1rg8A,
1s9uA,1t82A,1tvga,1urqA,1vcaA,1wdjA,1wvhA,1xo5A,
2a5dB,1c7kA,1dm9A,1f2tA,1gmuA,1hq1A,1iqzA,1jidA,
1kzfa,1maiA,1nh2D,1ocyA,1pcfA,1q9uA,1rh6A,1sd4A,
1t92A,1twuA,1uscA,1vctA,1wehA,1wwcA,1xteA,2aalA,
1ccwA,1dowA,1f2tB,1go3E,1hruA,1irqA,1jiwI,1l3kA,
1mjnA,1nkiA,1od3A,1pdoA,1qcsA,1rliA,1seiA,1t9iA,
1txlA,1usmA,1vgjA,1whzA,1wwzA,1y02A,2anrA,1cczA,
1dp7P,1f39A,1go3F,1htrP,1isuA,1jkeA,1l6pA,1mk4A,
1nkpA,1oeyA,1pkhA,1qf8A,1rocA,1sfpA,1tafA,1u0sA,
1ut7A,1vh5A,1wljA,1wy3A,1y0hA,1ci4A,1dunA,1f3uA,
1gp0A,1huwA,1it2A,1jmvA,1lb6A,1mvfD,1nlqA,1ogdA,
1pmhX,1qftA,1rowA,1sgmA,1tafB,1u14A,1utgA,1vi0A,
1wlzA,1wz3A,1y1xA,1n7kA,1nijA,1nnfA,1nuuA,1n2zA,
1i24A,1ii5A,1ixhA,1j5wA,1jdwA,1jovA,1juhA,1jykA,
1k3yA,1k8wA,1ko7A,1kwfA,1l7aA,1ls1A,1lzlA,1mixA,
1mtpA,1n3lA,1n7zA,1njrA,1nnwA,1i60A,1in4A,1izcA,
1j7xA,1jl0A,1jr2A,1jw9B,1jztA,1k77A,1kcmA,1kpgA,
1kzqA,1lbuA,1luaA,1m0kA,1mnnA,1mtyB,1n57A,1n93X,
1nlfA,1nszA,1i88A,1inlA,1j1tA,1jayA,1jm1A,1jr7A,
1jx6A,1k0iA,1k8kC,1khxA,1kq3A,1l3lA,1lc0A,1lucA,
1m6sA,1mpgA,1mtzA,1n62C,1nfpA,1nlsA,1ntyA,1i9zA,
1iuqA,1j4aA,1jb7B,1jmkC,1jtvA,1jyeA,1k3xA,1k8kD,

1kjqa,1kqfC,1l6rA,1lfpA,1lv7A,1m6yB,1mrzA,4zgmB
2vv6A,2zxyA,3cjeA,3es1A,3fttA,3ht1A,3l2hA,3n79A,
3rs1A,3witA,4bg7A,4ejrA,4hlyA,4ktwA,4m5dB,4ndsA,
4ouhA,4q2uA,4r7kA,4tpvA,4uuuA,4wzxA,4ynhA,4zkyA,
2vzyA,3ajvA,3ck1A,3eusA,3fxaA,3hv2A,3l51A,3n9uC,
3s9dA,3wmiA,4bhuA,4errA,4hs2A,4ku0A,4m62S,4ne3A,
4ounA,4q4w4,4r8hA,4tq1B,4uyiA,4wzxE,4ynxA,4zldA,
2w5eA,3anoA,3cnuA,3eytA,3fynA,3i96A,3l8dA,3nklA,
3s9dB,3wmiB,4bi3A,4eskA,4htuA,4ku0D,4m8aA,4nf1A,
4owtB,4q5eA,4rbrA,4tq2A,4v3iA,4x2hA,4yp6A,4zqaA,
2w7zA,3anpC,3cqbA,3f13A,3g13A,3ia1A,3l9fA,3n19A,
3shoA,3wmvA,4bjaA,4eunA,4hvyA,4kv2B,4m91A,4n19A,
4owtC,4qamB,4rcjA,4tsdB,4w4kA,4x2hB,4ytdA,4zv0A,
2wtgA,3b09A,3cu3A,3f14A,3g14A,3ihtA,3laeA,3nznA,
3sxmA,3wn7B,4bu0A,4evxA,4im6A,4l1jA,4macA,4nn5A,
4p1mA,4qasA,4reyB,4tshA,4w4kB,4x33A,4ytwA,4zv0B,
2wvbA,3b7hA,3d0wA,3f2iA,3g8zA,3ilxA,3lazA,3oa4A,
3t9yA,3wqbB,4bvxB,4evyA,4j39A,4l5eA,4makA,4nn5B,
4p3aA,4qblA,4rfuA,4tx4B,4w78A,4x3iA,4ytwB,4zvcA,
2x78A,3b8lA,3ddtA,3f42A,3ghjA,3imoA,3lluA,3oopA,
3tboA,3wvaA,4bwcA,4f7uB,4jdeB,4l8pA,4mi4A,4nutB,
4p3fA,4qboA,4rgdA,4tx5A,4w78B,4x86A,4yv4A,4zylA,
2xblA,3bb9A,3dewA,3f4aA,3glA,3ju3A,3llvA,3op9A,
3tj8A,3wvzA,4c5eE,4ffuA,4jemA,4l9nA,4mlmA,4nv4A,
4p3vA,4qbsA,4rguA,4txrA,4w8pB,4x86B,4ywkA,5a0rA,
2xbuA,3bbyA,3df8A,3f5oA,3glvA,3jygA,3lqnA,3p1xA,
3u28C,3wwqC,4c6sA,4fvdA,4jf3A,4l9uA,4mn5A,4o1rA,

4p5eA,4qdnA,4rhsA,4txrC,4wh5A,4x9zA,4yx1A,5a3dA,
2xcjA,3bd1A,3dlqI,3f6gB,3grdA,3k12A,3lw3A,3pluA,
3u2aA,3wwtB,4ca1A,4g6iA,4jj0A,4lflA,4mnnA,4o3vB,
4p5nB,4qe0A,4rkhC,4u1eG,4wjtA,4xalA,4yz6A,5a6wA,
2xdpA,3bedA,3dluA,3f8bA,3grzA,3k6gA,3lwcA,3q3jB,
3u3zA,3wydA,4cayA,4g6xA,4jj9A,4ljiA,4mqvB,4o4oA,
4p82A,4qftA,4rleA,4u3sB,4wksA,4xb6A,4z04A,5ajjA,
2yh9A,3bguA,3dmcA,3f8xA,3gwnA,3kbqA,3lx3A,3q87A,
3v4gA,3x38A,4cayC,4gdoA,4jo7A,4lkuA,4mt8A,4o66A,
4p9iA,4qkdA,4ro3A,4u5hA,4wndB,4xbaB,4z3xE,2yilA,
3bhqA,3do8A,3fanA,3gwyA,3kdfD,3lypA,3qbmA,3v6gA,
3zhoA,4cbuG,4gn5A,4js0B,4lloA,4mtmA,4o7jB,4pasA,
4qkwA,4rp3A,4u5rA,4wolA,4xhtA,4z8tB,2yleA,3blnA,
3dwgC,3fauA,3gy9A,3kg0A,3lyxA,3qdlA,3vbjA,3zieA,
4cngA,4gneA,4jw0B,4loob,4mtuA,4o8yB,4pdcE,4qlpA,
4rptA,4u7iA,4wp9A,4xinA,4zbhA,2yqyA,3blzA,3e05A,
3ff2A,3gydA,3kgzA,3m9lA,3qv1G,3vcxA,3zihA,4cryB,
4gqmA,4jzuA,4lowA,4mxtA,4od8D,4peoA,4qlpB,4rs7R,
4uafE,4wpyA,4xo1A,4zc4A,2yskA,3bm7A,3e19A,3fh1A,
3h2bA,3kmaA,3md1A,3qvaA,3vfzA,3zqsA,4csrA,4gqzA,
4k02A,4lviA,4mypA,4oi3A,4phjA,4qm6A,4rt1A,4ub8R,
4wsfA,4xrmA,4zceB,2yveA,3bn7A,3e57A,3fjsA,3h6pA,
3kojA,3mgdA,3r0nA,3vp5A,4a1kA,4csrB,4h2wC,4k12A,
4lwsA,4mzgB,4oieA,4pibA,4qndA,4rt4E,4ue0A,4wv4A,
4xu6A,4zcnA,2yvqA,3bwvA,3ec6A,3flhA,3h7hA,3kopA,
3mhsE,3r3cA,3vrdA,4a6hA,4cvoA,4h3kB,4k12B,4lwsB,
4n0hF,4ojuA,4pp8C,4qttB,4s1aA,4ue8B,4wvrD,4xzfA,

4zdsA, 2yzjA, 3c3pA, 3ej9A, 3fn2A, 3ha2A, 3kosA, 3mnmA,
 3r5gA, 3vygB, 4aciA, 4cxfB, 4h3uA, 4k9zA, 4lyyA, 4n6qA,
 4okeA, 4pz1A, 4qu6A, 4s2xA, 4un1A, 4ww7B, 4yepA, 4zeqA,
 2z3jA, 3c57A, 3ejvA, 3fpnA, 3ha9A, 3ktaA, 3mqqa, 3rfiA,
 3w0tA, 4aikA, 4cybA, 4h87A, 4kqdA, 4lzkA, 4n7cA, 4okvE,
 4pzjA, 4qusA, 4s3oC, 4un1B, 4wy4A, 4yh8A, 4zeyA, 2zejA,
 3c7xA, 3ek3A, 3frqA, 3hf5A, 3ktaB, 3mtqA, 3rnqB, 3w61A,
 4axyA, 4czxB, 4hfsA, 4krdB, 4lzxB, 4nb5A, 4otmA, 4q0yA,
 4qxbB, 4tkcA, 4un2B, 4wy4B, 4yh8B

Table C.5. PDB ID list of the training/validation set of Feature Contribution Analysis.

PDB ID/CHAIN NAME
1a62A, 1cukA, 1dvoA, 1f3uB, 1gprA, 1hxiA, 1ixlA, 1juvA, 1lf7A, 1mvlA, 1nnxA, 1oi0A, 1pocA, 1qkrA, 1rssA, 1sgwA, 1tc5A, 1u2hA, 1uujA, 1vi6A, 1wmhA, 1x2iA, 1y63A, 1ae9A, 1cv8A, 1dwkA, 1f46A, 1gs9A, 1hxrA, 1izmA, 1jyaA, 1lkiA, 1my7A, 1np6A, 1on2A, 1pp0A, 1qqp1, 1rttA, 1sh8A, 1tfeA, 1u2wA, 1uuyA, 1vimA, 1wmhB, 1x3kA, 1y7rA, 1alyA, 1cxqA, 1dxgA, 1f60B, 1gu4A, 1hztA, 1j0pA, 1jyoA, 1lkkA, 1mzwB, 1nqzA, 1oo0B, 1pqhA, 1qv1A, 1rutX, 1sj1A, 1th7A, 1u5fA, 1uv7A, 1vj1A, 1wmxA, 1x6oA, 1y88A, 1ayoA, 1cy5A, 1dzkA, 1fc3A, 1gu9A, 1i07A, 1j24A, 1k3sA, 1lqvA, 1n0wB, 1nrjB, 1oqjA, 1psrA, 1qw2A, 1rxdA, 1sqwA, 1tigA, 1u7kA, 1uw4A, 1vkeA, 1wocA, 1x91A, 1y9iA, 1b4fA, 1d2oA, 1e0bA, 1fltX, 1guiA, 1i12A, 1j27A, 1k4nA, 1lr0A, 1n12A, 1ntvA, 1orsC, 1pt6A, 1qwdA, 1rxqA, 1ss4A, 1tiqA, 1u7pA, 1uxoA, 1vkiA,

1wolA,1xauA,1y9wA,1b9wA,1d2sA,1e30A,1fm0E,1gutA,
1i2tA,1j3wA,1k8kE,1lr5A,1n13B,1nu0A,1oruA,1pvmA,
1qzgA,1rylA,1sviA,1tjlA,1u84A,1v0aA,1vkkA,1wouA,
1xe1A,1yd9A,1bgcA,1d2zB,1eayC,1fpoA,1gxuA,1i4uA,
1j77A,1k8kG,1lshB,1n1fA,1nxmA,1ou8A,1pyoB,1qzmA,
1ryqA,1sz7A,1tp6A,1u9kA,1v2xA,1vl7A,1wpaA,1xe7A,
1ynbA,1bm8A,1d3bA,1ef1C,1fpzA,1gy7A,1i7wB,1j7dA,
1kgdA,1lu4A,1n62A,1nznA,1ow1A,1pzwA,1r0dA,1rz3A,
1t07A,1ts9A,1u9lA,1v4pA,1vmgA,1wpbA,1xfsA,1yocA,
1bm9A,1d3bB,1egwA,1g12A,1h2sB,1i8aA,1j8bA,1khyA,
1luzA,1n71A,1o13A,1ow4A,1q08A,1r0uA,1s12A,1t0pB,
1tu1A,1uebA,1v5iB,1vqsA,1wrda,1xg0C,1z1sA,1bteA,
1d4tA,1elkA,1g3kA,1h4aX,1id0A,1je5A,1klxA,1lxjA,
1n7sC,1o4wA,1oz9A,1q0pA,1r6jA,1s29A,1t1jA,1tu9A,
1ui0A,1v6pA,1vsrA,1ws8A,1xhdA,1zavA,1btka,1ddwA,
1elwA,1g5tA,1h6fA,1idpA,1jf3A,1koeA,1ly1A,1n9pA,
1o50A,1p57A,1q1fA,1r75A,1s3cA,1t3yA,1tuaA,1ujcA,
1v74B,1w2wB,1wu9A,1xhnA,1ze3H,1bxyA,1dfuP,1epfA,
1g6gA,1h6hA,1ifrA,1jh6A,1kptA,1m0dA,1ng2A,1o5uA,
1p6oA,1q40B,1r77A,1s4kA,1t4aA,1tuhA,1ukka,1v96A,
1w4sA,1wubA,1xiwA,1zpvA,1byfA,1dg6A,1ew4A,1g8eA,
1h8pA,1igqA,1jhgA,1kt6A,1m1fA,1ng6A,1o6dA,1p9gA,
1q42A,1r7jA,1s5uA,1t6sA,1tuvA,1unnC,1v9yA,1w53A,
1wurA,1xiwB,1zuoA,1byrA,1dj8A,1ez3A,1g8qa,1h97A,
1io0A,1jhjA,1kxgA,1m4iA,1nh2B,1o7iA,1p9hA,1q8bA,
1r9wA,1s7iA,1t6t1,1tuwA,1uptB,1vbwA,1wdcB,1wv9A,
1xlqA,2a0bA,1c1yB,1dk8A,1f1mA,1gheA,1hfeS,1iq4A,

1jhsA,1kxoA,1m70A,1nh2C,1oa8A,1pbwA,1q8dA,1rg8A,
1s9uA,1t82A,1tvga,1urqA,1vcaA,1wdjA,1wvhA,1xo5A,
2a5dB,1c7kA,1dm9A,1f2tA,1gmuA,1hq1A,1iqzA,1jidA,
1kzfa,1maiA,1nh2D,1ocyA,1pcfA,1q9uA,1rh6A,1sd4A,
1t92A,1twuA,1uscA,1vctA,1wehA,1wwcA,1xteA,2aalA,
1ccwA,1dowA,1f2tB,1go3E,1hruA,1irqA,1jiwI,1l3kA,
1mjnA,1nkiA,1od3A,1pdoA,1qcsA,1rliA,1seiA,1t9iA,
1txlA,1usmA,1vgjA,1whzA,1wwzA,1y02A,2anrA,1cczA,
1dp7P,1f39A,1go3F,1htrP,1isuA,1jkeA,1l6pA,1mk4A,
1nkpA,1oeyA,1pkhA,1qf8A,1rocA,1sfpA,1tafA,1u0sA,
1ut7A,1vh5A,1wljA,1wy3A,1y0hA,1ci4A,1dunA,1f3uA,
1gp0A,1huwA,1it2A,1jmvA,1lb6A,1mvfD,1nlqA,1ogdA,
1pmhX,1qftA,1rowA,1sgmA,1tafB,1u14A,1utgA,1vi0A,
1wlzA,1wz3A,1y1xA,1i24A,1ii5A,1ixhA,1j5wA,1jdwA,
1jovA,1juhA,1jyK,1k3yA,1k8wA,1ko7A,1kwfA,1l7aA,
1ls1A,1lzlA,1mixA,1mtpA,1n3lA,1n7zA,1njrA,1nwwA,
1i60A,1in4A,1izcA,1j7xA,1jl0A,1jr2A,1jw9B,1jztA,
1k77A,1kcmA,1kpgA,1kzqA,1lbuA,1luaA,1m0kA,1mnnA,
1mtyB,1n57A,1n93X,1nlfA,1nszA,1i88A,1inlA,1j1tA,
1jayA,1jm1A,1jr7A,1jx6A,1k0iA,1k8kC,1khxA,1kq3A,
1l3lA,1lc0A,1lucA,1m6sA,1mpgA,1mtzA,1n62C,1nfpA,
1nlsA,1ntyA,1i9zA,1iuqA,1j4aA,1jb7B,1jmkC,1jtvA,
1jyeA,1k3xA,1k8kD,1kjqa,1kqfC,1l6rA,1lfpA,1lv7A,
1m6yB,1mrzA,1n2zA,1n7kA,1nijA,1nnfA,1nuuA

Table C.6. PDB ID list of the training/validation set of Feature Contribution Analysis.

PDB ID/CHAIN NAME
1a62A, 1cukA, 1dvoA, 1f3uB, 1gprA, 1hxiA, 1ixlA, 1juvA, 1lf7A, 1mvlA, 1nnxA, 1oi0A, 1pocA, 1qkrA, 1rssA, 1sgwA, 1tc5A, 1u2hA, 1uujA, 1vi6A, 1wmhA, 1x2iA, 1y63A, 1ae9A, 1cv8A, 1dwkA, 1f46A, 1gs9A, 1hxrA, 1izmA, 1jyaA, 1lkiA, 1my7A, 1np6A, 1on2A, 1pp0A, 1qqp1, 1rttA, 1sh8A, 1tfeA, 1u2wA, 1uuyA, 1vimA, 1wmhB, 1x3kA, 1y7rA, 1alyA, 1cxqA, 1dxgA, 1f60B, 1gu4A, 1hztA, 1j0pA, 1jyoA, 1lkkA, 1mzwB, 1nqzA, 1oo0B, 1pqhA, 1qv1A, 1rutX, 1sj1A, 1th7A, 1u5fA, 1uv7A, 1vj1A, 1wmxA, 1x6oA, 1y88A, 1ayoA, 1cy5A, 1dzkA, 1fc3A, 1gu9A, 1i07A, 1j24A, 1k3sA, 1lqvA, 1n0wB, 1nrjB, 1oqjA, 1psrA, 1qw2A, 1rxDA, 1sqwA, 1tigA, 1u7kA, 1uw4A, 1vkeA, 1wocA, 1x91A, 1y9iA, 1b4fA, 1d2oA, 1e0bA, 1fltX, 1guiA, 1i12A, 1j27A, 1k4nA, 1lr0A, 1n12A, 1ntvA, 1orsC, 1pt6A, 1qwdA, 1rxqA, 1ss4A, 1tiqA, 1u7pA, 1uxoA, 1vkiA, 1wolA, 1xauA, 1y9wA, 1b9wA, 1d2sA, 1e30A, 1fm0E, 1gutA, 1i2tA, 1j3wA, 1k8kE, 1lr5A, 1n13B, 1nu0A, 1oruA, 1pvmA, 1qzgA, 1rylA, 1sviA, 1tjlA, 1u84A, 1v0aA, 1vkkA, 1wouA, 1xe1A, 1yd9A, 1bgcA, 1d2zB, 1eayC, 1fpoA, 1gxuA, 1i4uA, 1j77A, 1k8kG, 1lshB, 1n1fA, 1nxmA, 1ou8A, 1pyoB, 1qzmA, 1ryqA, 1sz7A, 1tp6A, 1u9kA, 1v2xA, 1vl7A, 1wpaA, 1xe7A, 1ynbA, 1bm8A, 1d3bA, 1ef1C, 1fpzA, 1gy7A, 1i7wB, 1j7dA, 1kgdA, 1lu4A, 1n62A, 1nznA, 1ow1A, 1pzwA, 1r0dA, 1rz3A, 1t07A, 1ts9A, 1u9lA, 1v4pA, 1vmgA, 1wpbA, 1xfSA, 1yocA, 1bm9A, 1d3bB, 1egwA, 1g12A, 1h2sB, 1i8aA, 1j8bA, 1khyA,

1luzA, 1n71A, 1o13A, 1ow4A, 1q08A, 1r0uA, 1s12A, 1t0pB,
1tu1A, 1uebA, 1v5iB, 1vqsA, 1wrdA, 1xg0C, 1z1sA, 1bteA,
1d4tA, 1elkA, 1g3kA, 1h4aX, 1id0A, 1je5A, 1klxA, 1lxjA,
1n7sC, 1o4wA, 1oz9A, 1q0pA, 1r6jA, 1s29A, 1t1jA, 1tu9A,
1ui0A, 1v6pA, 1vsrA, 1ws8A, 1xhdA, 1zavA, 1btkA, 1ddwA,
1elwA, 1g5tA, 1h6fA, 1idpA, 1jf3A, 1koeA, 1ly1A, 1n9pA,
1o50A, 1p57A, 1q1fA, 1r75A, 1s3cA, 1t3yA, 1tuaA, 1ujcA,
1v74B, 1w2wB, 1wu9A, 1xhnA, 1ze3H, 1bxyA, 1dfuP, 1epfA,
1g6gA, 1h6hA, 1ifrA, 1jh6A, 1kptA, 1m0dA, 1ng2A, 1o5uA,
1p6oA, 1q40B, 1r77A, 1s4kA, 1t4aA, 1tuhA, 1ukkA, 1v96A,
1w4sA, 1wubA, 1xiwA, 1zpvA, 1byfA, 1dg6A, 1ew4A, 1g8eA,
1h8pA, 1igqA, 1jhgA, 1kt6A, 1m1fA, 1ng6A, 1o6dA, 1p9gA,
1q42A, 1r7jA, 1s5uA, 1t6sA, 1tuvA, 1unnC, 1v9yA, 1w53A,
1wurA, 1xiwB, 1zuoA, 1byrA, 1dj8A, 1ez3A, 1g8qA, 1h97A,
1io0A, 1jhjA, 1kxgA, 1m4iA, 1nh2B, 1o7iA, 1p9hA, 1q8bA,
1r9wA, 1s7iA, 1t6t1, 1tuwA, 1uptB, 1vbwA, 1wdcB, 1wv9A,
1xlqA, 2a0bA, 1c1yB, 1dk8A, 1f1mA, 1gheA, 1hfeS, 1iq4A,
1jhsA, 1kxoA, 1m70A, 1nh2C, 1oa8A, 1pbwA, 1q8dA, 1rg8A,
1s9uA, 1t82A, 1tvga, 1urqA, 1vcaA, 1wdjA, 1wvhA, 1xo5A,
2a5dB, 1c7kA, 1dm9A, 1f2tA, 1gmuA, 1hq1A, 1iqzA, 1jidA,
1kzfA, 1maiA, 1nh2D, 1ocyA, 1pcfA, 1q9uA, 1rh6A, 1sd4A,
1t92A, 1twuA, 1uscA, 1vctA, 1wehA, 1wwcA, 1xteA, 2aalA,
1ccwA, 1dowA, 1f2tB, 1go3E, 1hruA, 1irqA, 1jiwI, 1l3kA,
1mjnA, 1nkiA, 1od3A, 1pdoA, 1qcsA, 1rliA, 1seiA, 1t9iA,
1txlA, 1usmA, 1vgjA, 1whzA, 1wwzA, 1y02A, 2anrA, 1cczA,
1dp7P, 1f39A, 1go3F, 1htrP, 1isuA, 1jkeA, 1l6pA, 1mk4A,
1nkpA, 1oeyA, 1pkhA, 1qf8A, 1rocA, 1sfpA, 1tafA, 1u0sA,

1ut7A,1vh5A,1wljA,1wy3A,1y0hA,1ci4A,1dunA,1f3uA,
1gp0A,1huwA,1it2A,1jmvA,1lb6A,1mvfD,1nlqA,1ogdA,
1pmhX,1qftA,1rowA,1sgmA,1tafB,1u14A,1utgA,1vi0A,
1wlzA,1wz3A,1y1xA

Table C.7. PDB ID list of training/validation set of the TM-score prediction network.

PDB ID/CHAIN NAME
1a62A,1bm8A,1c1yB,1cxqA,1d4tA,1dowA,1e0bA,1egwA, 1es5A,1f1mA,1hq1A,1i07A,1dp7P,1e30A,1guiA,1gy7A, 1f5vA,1fpoA,1g3kA,1g8eA,1go3F,1gu9A,1gxyA,1h6hA, 1id0A,1io0A,1izcA,1ae9A,1bm9A,1c7kA,1cy5A,1ddwA, 1ekqA,1es9A,1f2tA,1f60B,1fpzA,1g5tA,1g8qA,1gp0A, 1h72C,1hruA,1i12A,1idpA,1iq4A,1izmA,1alyA,1bteA, 1ccwA,1d2oA,1g60A,1ga8A,1j0pA,1ayoA,1f39A,1fiuA, 1dfuP,1dunA,1eayC,1elkA,1euvA,1f2tB,1fc3A,1ft5A, 1gppA,1gutA,1h0hB,1h8pA,1htrP,1i2tA,1ifrA,1iqzA, 1btkA,1cczA,1d2sA,1dg6A,1dvoA,1eejA,1elwA,1ew4A, 1ftrA,1g66A,1gheA,1gprA,1gv9A,1h2sB,1h97A,1huwA, 1i4uA,1igqA,1eerB,1eokA,1h32A,1h99A,1d3bA,1dk8A, 1irqA,1j24A,1b4fA,1bxyA,1ci4A,1d2zB,1dj8A,1dwkA, 1ez3A,1f3uA,1fjhA,1fviA,1g6gA,1gl4A,1gs5A,1gvfA, 1hxiA,1i60A,1ii5A,1isuA,1j27A,1b9wA,1byfA,1cukA, 1dxgA,1ef1C,1epfA,1eziA,1f3uB,1fltX,1fyeA,1g6hA, 1gmuA,1gs9A,1bgcA,1byrA,1fm0E,1g12A,1inlA,1ixlA, 1gvnB,1h4aX,1hfeS,1hxrA,1i7wB,1in4A,1it2A,1j3wA, 1cv8A,1d3bB,1dm9A,1dzkA,1efdN,1eq2A,1f00I,1f46A, 1g8aA,1go3E,1gu4A,1gxuA,1h6fA,1hq0A,1hztA,1i8aA, 1j77A

APPENDIX D

PROTEIN CLASSIFICATION

Table D.1. PDB ID list of the training/validation set of DeepCDpred.

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
1b4fA	α	1sgwA	α/β	1y9wA	$\alpha+\beta$
1bgcA	α	1sviA	α/β	1yocA	$\alpha+\beta$
1ci4A	α	1t1jA	α/β	1z1sA	$\alpha+\beta$
1cy5A	α	1t3yA	α/β	1zavA	$\alpha+\beta$
1d2zB	α	1t6t1	α/β	1zpvA	$\alpha+\beta$
1dj8A	α	1tigA	α/β	1zuoA	$\alpha+\beta$
1dk8A	α	1u7pA	α/β	2a5dB	$\alpha+\beta$
1dowA	α	1ui0A	α/β	2anrA	$\alpha+\beta$
1ef1C	α	1ujcA	α/β	2vv6A	$\alpha+\beta$
1elkA	α	1usmA	α/β	2vzyA	$\alpha+\beta$
1elwA	α	1uuyA	α/β	2w5eA	$\alpha+\beta$
1ez3A	α	1uxoA	α/β	2wvbA	$\alpha+\beta$
1flmA	α	1v2xA	α/β	2yh9A	$\alpha+\beta$
1fc3A	α	1v96A	α/β	2yilA	$\alpha+\beta$
1fpoA	α	1vbwA	α/β	2yleA	$\alpha+\beta$
1g12A	α	1vi6A	α/β	2yzjA	$\alpha+\beta$

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
1g8eA	α	1vimA	α/β	2z3jA	$\alpha+\beta$
1g8qA	α	1vkiA	α/β	3anoA	$\alpha+\beta$
1go3F	α	1vsrA	α/β	3b8lA	$\alpha+\beta$
1gs9A	α	1w2wB	α/β	3bb9A	$\alpha+\beta$
1gu4A	α	1wehA	α/β	3bguA	$\alpha+\beta$
1gu9A	α	1wljA	α/β	3blnA	$\alpha+\beta$
1h97A	α	1wouA	α/β	3blzA	$\alpha+\beta$
1hfeS	α	1wv9A	α/β	3bm7A	$\alpha+\beta$
1hq1A	α	1xhdA	α/β	3bn7A	$\alpha+\beta$
1htrP	α	1y63A	α/β	3c7xA	$\alpha+\beta$
1huwA	α	1y88A	α/β	3cjeA	$\alpha+\beta$
1hxiA	α	1yd9A	α/β	3ck1A	$\alpha+\beta$
1i2tA	α	2aalA	α/β	3cnuA	$\alpha+\beta$
1i7wB	α	2vliA	α/β	3cu3A	$\alpha+\beta$
1irqA	α	2x78A	α/β	3ddtA	$\alpha+\beta$
1it2A	α	2xblA	α/β	3df8A	$\alpha+\beta$
1izmA	α	2xbuA	α/β	3dluA	$\alpha+\beta$
1j0pA	α	2yvqA	α/β	3dmcA	$\alpha+\beta$
1j77A	α	2zejA	α/β	3dwgC	$\alpha+\beta$
1jf3A	α	3ajvA	α/β	3e19A	$\alpha+\beta$
1jhgA	α	3bbyA	α/β	3e57A	$\alpha+\beta$
1k8kE	α	3bedA	α/β	3ek3A	$\alpha+\beta$
1k8kG	α	3bwvA	α/β	3elsA	$\alpha+\beta$
1khyA	α	3c3pA	α/β	3en8A	$\alpha+\beta$
1klxA	α	3c85A	α/β	3eytA	$\alpha+\beta$
1kwfA	α	3c97A	α/β	3f14A	$\alpha+\beta$
1lkiA	α	3cqbA	α/β	3f42A	$\alpha+\beta$
1m70A	α	3do8A	α/β	3f5oA	$\alpha+\beta$
1mtyB	α	3e05A	α/β	3f6gB	$\alpha+\beta$
1mzwB	α	3ec6A	α/β	3f8bA	$\alpha+\beta$
1n0wB	α	3ej9A	α/β	3f8xA	$\alpha+\beta$
1n1fA	α	3ejvA	α/β	3fanA	$\alpha+\beta$
1n7sC	α	3f13A	α/β	3fauA	$\alpha+\beta$

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
1n93X	α	3f2iA	α/β	3ff2A	$\alpha+\beta$
1ng6A	α	3f4aA	α/β	3fh1A	$\alpha+\beta$
1nh2B	α	3flhA	α/β	3fn2A	$\alpha+\beta$
1nkpA	α	3fttA	α/β	3fpnA	$\alpha+\beta$
1nznA	α	3fxaA	α/β	3fryA	$\alpha+\beta$
1on2A	α	3fynA	α/β	3g14A	$\alpha+\beta$
1ow4A	α	3g13A	α/β	3g8zA	$\alpha+\beta$
1pbwA	α	3glvA	α/β	3ghjA	$\alpha+\beta$
1psrA	α	3grzA	α/β	3grdA	$\alpha+\beta$
1q08A	α	3h2bA	α/β	3gwyA	$\alpha+\beta$
1q1fA	α	3ha2A	α/β	3gy9A	$\alpha+\beta$
1q8dA	α	3hv2A	α/β	3gydA	$\alpha+\beta$
1qkrA	α	3ia1A	α/β	3h7hA	$\alpha+\beta$
1qv1A	α	3ihtA	α/β	3ha9A	$\alpha+\beta$
1r0dA	α	3ilxA	α/β	3hf5A	$\alpha+\beta$
1r7jA	α	3ju3A	α/β	3hmzA	$\alpha+\beta$
1rxqA	α	3kbqA	α/β	3ht1A	$\alpha+\beta$
1s29A	α	3kdfD	α/β	3i96A	$\alpha+\beta$
1s9uA	α	3kgzA	α/β	3imoA	$\alpha+\beta$
1sd4A	α	3kosA	α/β	3jygA	$\alpha+\beta$
1sgmA	α	3l8dA	α/β	3k12A	$\alpha+\beta$
1t07A	α	3lluA	α/β	3kg0A	$\alpha+\beta$
1tafA	α	3llvA	α/β	3kojA	$\alpha+\beta$
1tafB	α	3lypA	α/β	3kopA	$\alpha+\beta$
1tjlA	α	3m9lA	α/β	3ktaA	$\alpha+\beta$
1tu9A	α	3mtqA	α/β	3ktaB	$\alpha+\beta$
1u2wA	α	3nklA	α/β	3l2hA	$\alpha+\beta$
1u7kA	α	3q3jB	α/β	3l51A	$\alpha+\beta$
1u84A	α	3rpeA	α/β	3laeA	$\alpha+\beta$
1u9lA	α	3shoA	α/β	3lqnA	$\alpha+\beta$
1uptB	α	3u3zA	α/β	3lw3A	$\alpha+\beta$
1urqA	α	3vbjA	α/β	3lx3A	$\alpha+\beta$
1utgA	α	3w61A	α/β	3lyxA	$\alpha+\beta$

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
1uuja	α	3wisA	α/β	3md1A	$\alpha+\beta$
1v74B	α	3wydA	α/β	3mgdA	$\alpha+\beta$
1vi0A	α	3zhoA	α/β	3mqqA	$\alpha+\beta$
1vkeA	α	3zieA	α/β	3n5bB	$\alpha+\beta$
1vmgA	α	3zihA	α/β	3n72A	$\alpha+\beta$
1w53A	α	4bfcA	α/β	3n79A	$\alpha+\beta$
1wdcB	α	4bu0A	α/β	3n9uC	$\alpha+\beta$
1wlzA	α	4c6sA	α/β	3nznA	$\alpha+\beta$
1wolA	α	4cbuG	α/β	3oa4A	$\alpha+\beta$
1wpaA	α	4cngA	α/β	3p1xA	$\alpha+\beta$
1wpbA	α	4eunA	α/β	3pluA	$\alpha+\beta$
1wrdA	α	4fvdA	α/β	3q87A	$\alpha+\beta$
1wu9A	α	4h3kB	α/β	3qdlA	$\alpha+\beta$
1wy3A	α	4htuA	α/β	3qvaA	$\alpha+\beta$
1x2iA	α	4im6A	α/β	3r3cA	$\alpha+\beta$
1x3kA	α	4jemA	α/β	3r5gA	$\alpha+\beta$
1x91A	α	4jj9A	α/β	3rnrA	$\alpha+\beta$
1xg0C	α	4ktwA	α/β	3rs1A	$\alpha+\beta$
1xo5A	α	4lflA	α/β	3s9dB	$\alpha+\beta$
1y1xA	α	4m1aA	α/β	3t9yA	$\alpha+\beta$
1y9iA	α	4m1gA	α/β	3tboA	$\alpha+\beta$
1ynbA	α	4m62S	α/β	3tj8A	$\alpha+\beta$
2a0bA	α	4mnnA	α/β	3u2aA	$\alpha+\beta$
2vklA	α	4p5eA	α/β	3v4gA	$\alpha+\beta$
2wtgA	α	4p82A	α/β	3vcxA	$\alpha+\beta$
2xcjA	α	4qasA	α/β	3w0tA	$\alpha+\beta$
2yqyA	α	4qblA	α/β	3wqbB	$\alpha+\beta$
2yskA	α	4qm6A	α/β	3wvaA	$\alpha+\beta$
2yveA	α	4qttB	α/β	3wvzA	$\alpha+\beta$
2zfdA	α	4qu6A	α/β	3zqsA	$\alpha+\beta$
2zqmA	α	4rcjA	α/β	4a1kA	$\alpha+\beta$
2zxyA	α	4rfuA	α/β	4a6hA	$\alpha+\beta$
3anpC	α	4s1aA	α/β	4bg7A	$\alpha+\beta$

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
3b09A	α	4tpsB	α/β	4bi3A	$\alpha+\beta$
3b7hA	α	4u5rA	α/β	4bvxB	$\alpha+\beta$
3bd1A	α	4wsfA	α/β	4bwcA	$\alpha+\beta$
3bhqA	α	4yp6A	α/β	4ca1A	$\alpha+\beta$
3c57A	α	4ywkA	α/β	4cryB	$\alpha+\beta$
3d0wA	α	4zylA	α/β	4ejrA	$\alpha+\beta$
3dewA	α	1a62A	$\alpha+\beta$	4evyA	$\alpha+\beta$
3dllI	α	1ae9A	$\alpha+\beta$	4f7uB	$\alpha+\beta$
3eusA	α	1bm8A	$\alpha+\beta$	4ffuA	$\alpha+\beta$
3frqA	α	1bm9A	$\alpha+\beta$	4g6iA	$\alpha+\beta$
3ft7A	α	1btkA	$\alpha+\beta$	4g6xA	$\alpha+\beta$
3gwnA	α	1bxyA	$\alpha+\beta$	4gn5A	$\alpha+\beta$
3h6pA	α	1byfA	$\alpha+\beta$	4gqzA	$\alpha+\beta$
3hsrA	α	1byrA	$\alpha+\beta$	4h3uA	$\alpha+\beta$
3k6gA	α	1c1yB	$\alpha+\beta$	4hfsA	$\alpha+\beta$
3kz3A	α	1cv8A	$\alpha+\beta$	4hhvA	$\alpha+\beta$
3l1nA	α	1d3bA	$\alpha+\beta$	4hiaA	$\alpha+\beta$
3l9fA	α	1d4tA	$\alpha+\beta$	4hlyA	$\alpha+\beta$
3mhsE	α	1ddwA	$\alpha+\beta$	4hvyA	$\alpha+\beta$
3nl9A	α	1dm9A	$\alpha+\beta$	4j39A	$\alpha+\beta$
3oopA	α	1dp7P	$\alpha+\beta$	4jj0A	$\alpha+\beta$
3op9A	α	1dunA	$\alpha+\beta$	4jzuA	$\alpha+\beta$
3qbmA	α	1dvoA	$\alpha+\beta$	4k02A	$\alpha+\beta$
3qv1G	α	1dwkA	$\alpha+\beta$	4k9zA	$\alpha+\beta$
3rfiA	α	1dzkA	$\alpha+\beta$	4kqdA	$\alpha+\beta$
3s9dA	α	1e0bA	$\alpha+\beta$	4ktbA	$\alpha+\beta$
3sxmA	α	1e30A	$\alpha+\beta$	4kv2B	$\alpha+\beta$
3v6gA	α	1eayC	$\alpha+\beta$	4l8pA	$\alpha+\beta$
3vfzA	α	1egwA	$\alpha+\beta$	4lloA	$\alpha+\beta$
3vp5A	α	1ew4A	$\alpha+\beta$	4lowA	$\alpha+\beta$
3vrdA	α	1f2tA	$\alpha+\beta$	4lviA	$\alpha+\beta$
3vygB	α	1f2tB	$\alpha+\beta$	4lyyA	$\alpha+\beta$
3whjA	α	1f3uA	$\alpha+\beta$	4lzkA	$\alpha+\beta$

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
3wmiA	α	1f46A	$\alpha+\beta$	4m5dB	$\alpha+\beta$
3wmiB	α	1f60B	$\alpha+\beta$	4m8aA	$\alpha+\beta$
3wn7B	α	1f9vA	$\alpha+\beta$	4m91A	$\alpha+\beta$
3wwqC	α	1fm0E	$\alpha+\beta$	4macA	$\alpha+\beta$
3wwtB	α	1g3kA	$\alpha+\beta$	4makA	$\alpha+\beta$
3x38A	α	1g5hA	$\alpha+\beta$	4mi4A	$\alpha+\beta$
4aciA	α	1g6gA	$\alpha+\beta$	4mn5A	$\alpha+\beta$
4aikA	α	1gheA	$\alpha+\beta$	4mtmA	$\alpha+\beta$
4b4sA	α	1go3E	$\alpha+\beta$	4mtuA	$\alpha+\beta$
4bjaA	α	1gprA	$\alpha+\beta$	4mxtA	$\alpha+\beta$
4cayA	α	1gutA	$\alpha+\beta$	4mzgB	$\alpha+\beta$
4csrA	α	1gxmA	$\alpha+\beta$	4n6qA	$\alpha+\beta$
4csrB	α	1gxuA	$\alpha+\beta$	4n7cA	$\alpha+\beta$
4cvoA	α	1gy7A	$\alpha+\beta$	4ndhA	$\alpha+\beta$
4cxfB	α	1h6fA	$\alpha+\beta$	4ndsA	$\alpha+\beta$
4cybA	α	1h6hA	$\alpha+\beta$	4nf1A	$\alpha+\beta$
4czxB	α	1h8pA	$\alpha+\beta$	4nv4A	$\alpha+\beta$
4d2hA	α	1hruA	$\alpha+\beta$	4o3vB	$\alpha+\beta$
4errA	α	1hztA	$\alpha+\beta$	4o4oA	$\alpha+\beta$
4evxA	α	1i4uA	$\alpha+\beta$	4o66A	$\alpha+\beta$
4gdoA	α	1i88A	$\alpha+\beta$	4o7jB	$\alpha+\beta$
4gqmA	α	1i9zA	$\alpha+\beta$	4o8yB	$\alpha+\beta$
4h2wC	α	1id0A	$\alpha+\beta$	4oi3A	$\alpha+\beta$
4hs2A	α	1idpA	$\alpha+\beta$	4okeA	$\alpha+\beta$
4jf3A	α	1inlA	$\alpha+\beta$	4otmA	$\alpha+\beta$
4jo7A	α	1iq4A	$\alpha+\beta$	4otnA	$\alpha+\beta$
4k12B	α	1iqzA	$\alpha+\beta$	4ou6A	$\alpha+\beta$
4krdB	α	1ixlA	$\alpha+\beta$	4ouhA	$\alpha+\beta$
4kt6B	α	1j1tA	$\alpha+\beta$	4owtB	$\alpha+\beta$
4l1jA	α	1j27A	$\alpha+\beta$	4p1mA	$\alpha+\beta$
4l5eA	α	1j3wA	$\alpha+\beta$	4p3vA	$\alpha+\beta$
4l9nA	α	1j5wA	$\alpha+\beta$	4p5nB	$\alpha+\beta$
4l9uA	α	1j7dA	$\alpha+\beta$	4pdcE	$\alpha+\beta$

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
4ljiA	α	1j7xA	$\alpha+\beta$	4peoA	$\alpha+\beta$
4lkuA	α	1j8bA	$\alpha+\beta$	4pibA	$\alpha+\beta$
4lwsA	α	1jdwA	$\alpha+\beta$	4pp8C	$\alpha+\beta$
4lwsB	α	1jh6A	$\alpha+\beta$	4q0yA	$\alpha+\beta$
4lzxB	α	1jhjA	$\alpha+\beta$	4q29A	$\alpha+\beta$
4mlmA	α	1jhsA	$\alpha+\beta$	4q2lA	$\alpha+\beta$
4mqvB	α	1jidA	$\alpha+\beta$	4q5eA	$\alpha+\beta$
4mt8A	α	1jiwI	$\alpha+\beta$	4qamB	$\alpha+\beta$
4n0hF	α	1jl0A	$\alpha+\beta$	4qboA	$\alpha+\beta$
4nb5A	α	1jm1A	$\alpha+\beta$	4qbsA	$\alpha+\beta$
4nc7A	α	1jr7A	$\alpha+\beta$	4qe0A	$\alpha+\beta$
4ne3A	α	1juhA	$\alpha+\beta$	4qftA	$\alpha+\beta$
4nl9A	α	1jyaA	$\alpha+\beta$	4qkdA	$\alpha+\beta$
4nn5A	α	1jyoA	$\alpha+\beta$	4qlpA	$\alpha+\beta$
4nutB	α	1k3xA	$\alpha+\beta$	4qlpB	$\alpha+\beta$
4od8D	α	1k3yA	$\alpha+\beta$	4qusA	$\alpha+\beta$
4okvE	α	1k4nA	$\alpha+\beta$	4r2yA	$\alpha+\beta$
4ounA	α	1k8kC	$\alpha+\beta$	4r7kA	$\alpha+\beta$
4owtC	α	1k8kD	$\alpha+\beta$	4r8hA	$\alpha+\beta$
4p3aA	α	1k8wA	$\alpha+\beta$	4rhsA	$\alpha+\beta$
4p3fA	α	1kcmA	$\alpha+\beta$	4rleA	$\alpha+\beta$
4pasA	α	1khxA	$\alpha+\beta$	4ro3A	$\alpha+\beta$
4phjA	α	1kjqA	$\alpha+\beta$	4rptA	$\alpha+\beta$
4pz1A	α	1ko7A	$\alpha+\beta$	4rt1A	$\alpha+\beta$
4pzjA	α	1koeA	$\alpha+\beta$	4s2xA	$\alpha+\beta$
4q2uA	α	1kt6A	$\alpha+\beta$	4s3oC	$\alpha+\beta$
4qdnA	α	1kxoA	$\alpha+\beta$	4tpsA	$\alpha+\beta$
4qkwA	α	1kzfA	$\alpha+\beta$	4tpvA	$\alpha+\beta$
4qxbB	α	1l3kA	$\alpha+\beta$	4tq2A	$\alpha+\beta$
4r3qA	α	1l3lA	$\alpha+\beta$	4tsdB	$\alpha+\beta$
4rbrA	α	1l6pA	$\alpha+\beta$	4tx4B	$\alpha+\beta$
4rgdA	α	1lb6A	$\alpha+\beta$	4u1eG	$\alpha+\beta$
4rguA	α	1lbuA	$\alpha+\beta$	4u3sB	$\alpha+\beta$

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
4rkhC	α	1lf7A	$\alpha+\beta$	4uafE	$\alpha+\beta$
4rp3A	α	1lfpA	$\alpha+\beta$	4ue0A	$\alpha+\beta$
4rs7R	α	1lkkA	$\alpha+\beta$	4un1A	$\alpha+\beta$
4rt4E	α	1lqvA	$\alpha+\beta$	4un1B	$\alpha+\beta$
4tq1B	α	1lr0A	$\alpha+\beta$	4uuuA	$\alpha+\beta$
4tshA	α	1lr5A	$\alpha+\beta$	4uyiA	$\alpha+\beta$
4tx5A	α	1lshB	$\alpha+\beta$	4w78A	$\alpha+\beta$
4txrA	α	1lu4A	$\alpha+\beta$	4w78B	$\alpha+\beta$
4txrC	α	1luzA	$\alpha+\beta$	4wh5A	$\alpha+\beta$
4u5hA	α	1lxjA	$\alpha+\beta$	4wjtA	$\alpha+\beta$
4u7iA	α	1m1fA	$\alpha+\beta$	4wp9A	$\alpha+\beta$
4ue8B	α	1maiA	$\alpha+\beta$	4wpyA	$\alpha+\beta$
4un2B	α	1mixA	$\alpha+\beta$	4ww7B	$\alpha+\beta$
4v3iA	α	1mk4A	$\alpha+\beta$	4x2hA	$\alpha+\beta$
4w4kA	α	1mnnA	$\alpha+\beta$	4x2hB	$\alpha+\beta$
4w4kB	α	1mpgA	$\alpha+\beta$	4xb6A	$\alpha+\beta$
4w8pB	α	1mtpA	$\alpha+\beta$	4xbaB	$\alpha+\beta$
4wksA	α	1mvfD	$\alpha+\beta$	4xo1A	$\alpha+\beta$
4wv4A	α	1n13B	$\alpha+\beta$	4xzfA	$\alpha+\beta$
4wy4A	α	1n62A	$\alpha+\beta$	4yepA	$\alpha+\beta$
4wy4B	α	1n62C	$\alpha+\beta$	4yh8A	$\alpha+\beta$
4wy4C	α	1n71A	$\alpha+\beta$	4ynxA	$\alpha+\beta$
4wy4D	α	1n7zA	$\alpha+\beta$	4ytwB	$\alpha+\beta$
4wzxA	α	1n9pA	$\alpha+\beta$	4yx1A	$\alpha+\beta$
4wzxE	α	1ng2A	$\alpha+\beta$	4yz6A	$\alpha+\beta$
4x3iA	α	1nh2D	$\alpha+\beta$	4z04A	$\alpha+\beta$
4x86A	α	1nijA	$\alpha+\beta$	4z3xE	$\alpha+\beta$
4x86B	α	1nkiA	$\alpha+\beta$	4zkyA	$\alpha+\beta$
4xalA	α	1nnwA	$\alpha+\beta$	4zv0A	$\alpha+\beta$
4xhtA	α	1nnxA	$\alpha+\beta$	5a6wA	$\alpha+\beta$
4xrmA	α	1nszA	$\alpha+\beta$	1alyA	β
4yh8B	α	1ntvA	$\alpha+\beta$	1ayoA	β
4yiiA	α	1ntyA	$\alpha+\beta$	1b9wA	β

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
4ynhA	α	1nxmA	$\alpha+\beta$	1bteA	β
4ytdA	α	1o50A	$\alpha+\beta$	1cczA	β
4ytwA	α	1o5uA	$\alpha+\beta$	1d2oA	β
4yv4A	α	1oa8A	$\alpha+\beta$	1d2sA	β
4z8tB	α	1ocyA	$\alpha+\beta$	1d3bB	β
4zc4A	α	1oeyA	$\alpha+\beta$	1dg6A	β
4zdsA	α	1oo0B	$\alpha+\beta$	1dxgA	β
4zeqA	α	1oqjA	$\alpha+\beta$	1epfA	β
4zeyA	α	1oruA	$\alpha+\beta$	1f39A	β
4zgmB	α	1ou8A	$\alpha+\beta$	1f3uB	β
4zhbA	α	1ow1A	$\alpha+\beta$	1fftX	β
4zieA	α	1p57A	$\alpha+\beta$	1gp0A	β
4zldA	α	1p9gA	$\alpha+\beta$	1guiA	β
4zqaA	α	1pcfA	$\alpha+\beta$	1h4aX	β
4zv0B	α	1pkhA	$\alpha+\beta$	1hxrA	β
4zvcA	α	1pmhX	$\alpha+\beta$	1i07A	β
5a0rA	α	1pocA	$\alpha+\beta$	1i8aA	β
5a3dA	α	1pp0A	$\alpha+\beta$	1ifrA	β
5ajjA	α	1pvmA	$\alpha+\beta$	1igqA	β
1c7kA	α/β	1pyoB	$\alpha+\beta$	1jovA	β
1ccwA	α/β	1pzwA	$\alpha+\beta$	1kxgA	β
1cukA	α/β	1q40B	$\alpha+\beta$	1kzqA	β
1cxqA	α/β	1q42A	$\alpha+\beta$	1my7A	β
1dfuP	α/β	1q8bA	$\alpha+\beta$	1n12A	β
1fp2A	α/β	1q9uA	$\alpha+\beta$	1nh2C	β
1fpzA	α/β	1qcsA	$\alpha+\beta$	1nlqA	β
1g5tA	α/β	1qf8A	$\alpha+\beta$	1nlsA	β
1gmuA	α/β	1qftA	$\alpha+\beta$	1o7iA	β
1gpjA	α/β	1qqp1	$\alpha+\beta$	1od3A	β
1i12A	α/β	1qw2A	$\alpha+\beta$	1p9hA	β
1i24A	α/β	1qwdA	$\alpha+\beta$	1r0uA	β
1i60A	α/β	1qzgA	$\alpha+\beta$	1r75A	β
1ii5A	α/β	1r6jA	$\alpha+\beta$	1r77A	β

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
1in4A	α/β	1r9wA	$\alpha+\beta$	1rg8A	β
1io0A	α/β	1rh6A	$\alpha+\beta$	1sfpA	β
1iuqA	α/β	1rocA	$\alpha+\beta$	1t0pB	β
1ixhA	α/β	1rowA	$\alpha+\beta$	1tvgaA	β
1izcA	α/β	1rssA	$\alpha+\beta$	1u2hA	β
1j24A	α/β	1rutX	$\alpha+\beta$	1uebA	β
1j4aA	α/β	1rylA	$\alpha+\beta$	1v0aA	β
1jayA	α/β	1ryqA	$\alpha+\beta$	1v6pA	β
1jb7B	α/β	1s12A	$\alpha+\beta$	1vcaA	β
1je5A	α/β	1s5uA	$\alpha+\beta$	1wmxA	β
1jkeA	α/β	1s7iA	$\alpha+\beta$	1wocA	β
1jmkC	α/β	1seiA	$\alpha+\beta$	1wwcA	β
1jmvA	α/β	1sh8A	$\alpha+\beta$	1xauA	β
1jr2A	α/β	1sj1A	$\alpha+\beta$	1xe1A	β
1jtvA	α/β	1sqwA	$\alpha+\beta$	1xiwB	β
1juvA	α/β	1ss4A	$\alpha+\beta$	1ze3H	β
1jw9B	α/β	1sz7A	$\alpha+\beta$	2w7zA	β
1jx6A	α/β	1t4aA	$\alpha+\beta$	2xdpA	β
1jyeA	α/β	1t6sA	$\alpha+\beta$	3es1A	β
1jykA	α/β	1t82A	$\alpha+\beta$	3fjsA	β
1jztA	α/β	1t92A	$\alpha+\beta$	3glaA	β
1k0iA	α/β	1t9iA	$\alpha+\beta$	3kmaA	β
1k3sA	α/β	1tc5A	$\alpha+\beta$	3lazA	β
1k77A	α/β	1tfeA	$\alpha+\beta$	3lwcA	β
1kgdA	α/β	1th7A	$\alpha+\beta$	3mnmA	β
1kpgA	α/β	1tiqA	$\alpha+\beta$	3r0nA	β
1kptA	α/β	1tp6A	$\alpha+\beta$	3rnqB	β
1kq3A	α/β	1ts9A	$\alpha+\beta$	3u28C	β
1l6rA	α/β	1tu1A	$\alpha+\beta$	3witA	β
1l7aA	α/β	1tuaA	$\alpha+\beta$	3wmvA	β
1lc0A	α/β	1tuhA	$\alpha+\beta$	4bhuaA	β
1ls1A	α/β	1tuvA	$\alpha+\beta$	4c5eE	β
1luaA	α/β	1tuwA	$\alpha+\beta$	4db5A	β

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
1lucA	α/β	1twuA	$\alpha+\beta$	4eskA	β
1lv7A	α/β	1txlA	$\alpha+\beta$	4gneA	β
1ly1A	α/β	1u0sA	$\alpha+\beta$	4h87A	β
1lzlA	α/β	1u14A	$\alpha+\beta$	4jdeB	β
1m0dA	α/β	1u5fA	$\alpha+\beta$	4jw0B	β
1m4iA	α/β	1u9kA	$\alpha+\beta$	4k12A	β
1m6sA	α/β	1ukkA	$\alpha+\beta$	4ku0A	β
1m6yB	α/β	1unnC	$\alpha+\beta$	4ku0D	β
1mjnA	α/β	1uscA	$\alpha+\beta$	4mypA	β
1mrzA	α/β	1ut7A	$\alpha+\beta$	4nn5B	β
1mtzA	α/β	1uv7A	$\alpha+\beta$	4o1rA	β
1mvlA	α/β	1uw4A	$\alpha+\beta$	4oieA	β
1n2zA	α/β	1v4pA	$\alpha+\beta$	4ojuA	β
1n3lA	α/β	1v5iB	$\alpha+\beta$	4p9iA	β
1n57A	α/β	1v9yA	$\alpha+\beta$	4reyB	β
1n7kA	α/β	1vctA	$\alpha+\beta$	4tkcA	β
1nfpA	α/β	1vgjA	$\alpha+\beta$	4usoA	β
1njrA	α/β	1vh5A	$\alpha+\beta$	4wvrD	β
1nlfA	α/β	1vj1A	$\alpha+\beta$	4x33A	β
1nnfA	α/β	1vkkA	$\alpha+\beta$	4x9zA	β
1np6A	α/β	1vl7A	$\alpha+\beta$	4xinA	β
1nqzA	α/β	1vqsA	$\alpha+\beta$	4yl8B	β
1nrjB	α/β	1w4sA	$\alpha+\beta$	4zbhA	β
1nu0A	α/β	1wdjA	$\alpha+\beta$	4zceB	β
1nuuA	α/β	1whzA	$\alpha+\beta$	4zcnA	β
1o13A	α/β	1wmhA	$\alpha+\beta$	1isuA	coil
1o4wA	α/β	1wmhB	$\alpha+\beta$	4axyA	coil
1o6dA	α/β	1ws8A	$\alpha+\beta$	4cayC	coil
1ogdA	α/β	1wubA	$\alpha+\beta$	4js0B	coil
1oi0A	α/β	1wurA	$\alpha+\beta$	4looB	coil
1oz9A	α/β	1wvhA	$\alpha+\beta$	4q4w4	coil
1p6oA	α/β	1wwzA	$\alpha+\beta$	4uqzB	coil
1pdoA	α/β	1wz3A	$\alpha+\beta$	4wndB	coil

Continued on next page

Table D.1 – continued from previous page

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
1pqhA	α/β	1x6oA	$\alpha+\beta$	1h2sB	membrane
1pt6A	α/β	1xe7A	$\alpha+\beta$	1kqfC	membrane
1q0pA	α/β	1xf5A	$\alpha+\beta$	1m0kA	membrane
1qzmA	α/β	1xhnA	$\alpha+\beta$	1orsC	membrane
1rliA	α/β	1xiwA	$\alpha+\beta$	4qndA	membrane
1rttA	α/β	1xlqA	$\alpha+\beta$	4ub8R	membrane
1rxdA	α/β	1xteA	$\alpha+\beta$	4wolA	membrane
1rz3A	α/β	1y02A	$\alpha+\beta$	4xu6A	membrane
1s3cA	α/β	1y0hA	$\alpha+\beta$	1y7rA	$\alpha+\beta$
1s4kA	α/β				

Table D.2. Protein classes of the 108 protein chains in the test set of Deep-CDpred.

PDB ID	Protein Class	PDB ID	Protein Class	PDB ID	Protein Class
1a3a	$\alpha+\beta$	1fx2	$\alpha+\beta$	1lm4	$\alpha+\beta$
1aap	β	1g2r	α/β	1lo7	$\alpha+\beta$
1aba	$\alpha+\beta$	1g9o	$\alpha+\beta$	1m4j	$\alpha+\beta$
1ag6	β	1gmi	β	1m8a	$\alpha+\beta$
1aoe	α/β	1gmx	α/β	1mk0	$\alpha+\beta$
1atz	α/β	1gz2	$\alpha+\beta$	1mug	α/β
1avs	α	1gzc	β	1nb9	$\alpha+\beta$
1bdo	β	1h2e	$\alpha+\beta$	1ne2	α/β
1beb	$\alpha+\beta$	1h4x	α/β	1nps	β
1beh	β	1hdo	α/β	1nrv	$\alpha+\beta$
1bkr	α	1hfc	α/β	1ny1	α/β
1c44	$\alpha+\beta$	1hh8	α	1o1z	α/β
1c52	$\alpha+\beta$	1htw	α/β	1p90	$\alpha+\beta$
1c9o	β	1hxn	β	1pch	$\alpha+\beta$
1cc8	$\alpha+\beta$	1i1j	β	1qf9	α/β
1chd	α/β	1i1n	α/β	1qjp	membrane
1cjw	$\alpha+\beta$	1i4j	$\alpha+\beta$	1r26	α/β
1cke	α/β	1i58	$\alpha+\beta$	1roa	$\alpha+\beta$
1ctf	$\alpha+\beta$	1i71	coil	1rw1	$\alpha+\beta$
1cxy	$\alpha+\beta$	1iib	α/β	1smx	β
1czn	α/β	1im5	α/β	1svy	$\alpha+\beta$
1d0q	$\alpha+\beta$	1j3a	α	1t8k	α
1d1q	α/β	1jfu	$\alpha+\beta$	1tif	$\alpha+\beta$
1d4o	α/β	1j11	$\alpha+\beta$	1tqg	α
1dix	$\alpha+\beta$	1jo0	$\alpha+\beta$	1tqh	α/β
1dlw	α	1jo8	β	1vfy	coil
1dmg	α/β	1jos	$\alpha+\beta$	1vjk	$\alpha+\beta$
1dqg	β	1jwq	a/b	1vmb	$\alpha+\beta$
1dsx	$\alpha+\beta$	1jyh	$\alpha+\beta$	1vp6	$\alpha+\beta$
1eaz	$\alpha+\beta$	1k6k	α	1w0h	$\alpha+\beta$
1ej8	β	1k7j	$\alpha+\beta$	1whi	$\alpha+\beta$
1f6b	$\alpha+\beta$	1kq6	$\alpha+\beta$	1wjx	$\alpha+\beta$
1fcy	α	1kqr	β	1wkc	α/β
1fk5	α	1ktg	$\alpha+\beta$	1xff	$\alpha+\beta$
1fl0	β	1ku3	α	2cua	β
1fvq	$\alpha+\beta$	1kw4	α	2phy	$\alpha+\beta$

REFERENCES

- Adhikari, B., Bhattacharya, D., Cao, R., and Cheng, J. (2015). Confold: Residue-residue contact-guided ab initio protein folding. *Proteins*, 83(8):1436–49.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–10.
- Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–402.
- Anfinsen, C. (1973). Principles that govern the folding of protein chains. *Science*, 181(4096):223–230.
- Anishchenko, I., Ovchinnikov, S., Kamisetty, H., and Baker, D. (2017). Origins of co-evolution between residues distant in protein 3d structures. *Proc Natl Acad Sci U S A*.
- Ari, L. and Goldman, N. (2005). An algorithm for progressive multiple alignment of sequences with insertions. *PNAS*, 102(30):10557–10562.
- Baker, F. N. and Porollo, A. (2016). Coeviz: a web-based tool for coevolution analysis of protein residues. *BMC Bioinformatics*, 17:119.
- Balakrishnan, S., Kamisetty, H., Carbonell, J. G., Lee, S. I., and Langmead, C. J. (2011). Learning generative models for protein fold families. *Proteins*, 79(4):1061–78.
- Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L., Studholme, D. J., Yeats, C., and Eddy, S. R. (2004). The pfam protein families database. *Nucleic Acids Res*, 32(Database issue):D138–41.
- Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2(1):1–127.

- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell*, 35(8):1798–828.
- Besag, J. (1977). Efficiency of pseudolikelihood estimation for simple gaussian fields. *Biometrika*, 64(3):616–618.
- Betancourt, M. R. and Thirumalai, D. (1999). Pair potentials for protein folding: choice of reference states and sensitivity of predicted native states to variations in the interaction schemes. *Protein Sci*, 8(2):361–9.
- BLAST (Accessed: 01/12/2018). <ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>.
- Boratyn, G. M., Schaffer, A. A., Agarwala, R., Altschul, S. F., Lipman, D. J., and Madden, T. L. (2012). Domain enhanced lookup time accelerated blast. *Biol Direct*, 7:12.
- Bordoli, L., Kiefer, F., Arnold, K., Benkert, P., Battey, J., and Schwede, T. (2009). Protein structure homology modeling using swiss-model workspace. *Nat Protoc*, 4(1):1–13.
- Brunger, A. T., Adams, P. D., Clore, G. M., DeLano, W. L., Gros, P., Grosse-Kunstleve, R. W., Jiang, J. S., Kuszewski, J., Nilges, M., Pannu, N. S., Read, R. J., Rice, L. M., Simonson, T., and Warren, G. L. (1998). Crystallography & nmr system: A new software suite for macromolecular structure determination. *Acta Crystallogr D Biol Crystallogr*, 54(Pt 5):905–21.
- Buchan, D. W., Ward, S. M., Lobley, A. E., Nugent, T. C., Bryson, K., and Jones, D. T. (2010). Protein annotation and modelling servers at university college london. *Nucleic Acids Res*, 38(Web Server issue):W563–8.
- Burger, L. and van Nimwegen, E. (2008). Accurate prediction of protein-protein interactions from sequence alignments using a bayesian method. *Molecular Systems Biology*, 4.
- Burger, L. and van Nimwegen, E. (2010). Disentangling direct from indirect co-evolution of residues in protein alignments. *PLoS Comput Biol*, 6(1):e1000633.
- Buslje, C. M., Santos, J., Delfino, J. M., and Nielsen, M. (2009). Correction for phylogeny, small number of observations and data redundancy improves the identification of co-evolving amino acid pairs using mutual information. *Bioinformatics*, 25(9):1125–31.
- caffe (Accessed: 8/1/2018). <http://caffe.berkeleyvision.org>.
- Cao, R., Adhikari, B., Bhattacharya, D., Sun, M., Hou, J., and Cheng, J. (2017). Qacon: single model quality assessment using protein structural and contact information with machine learning techniques. *Bioinformatics*, 33(4):586–588.

- Cao, R., Bhattacharya, D., Adhikari, B., Li, J., and Cheng, J. (2015). Large-scale model quality assessment for improving protein tertiary structure prediction. *Bioinformatics*, 31(12):i116–23.
- Cao, R. and Cheng, J. (2016). Protein single-model quality assessment by feature-based probability density functions. *Sci Rep*, 6:23990.
- Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd international conference on Machine learning*, pages 161–168.
- CASP (Accessed: 01/12/2018). <http://www.predictioncenter.org>.
- Chakrabarti, S. and Panchenko, A. R. (2010). Structural and functional roles of coevolved sites in proteins. *PLoS One*, 5(1):e8591.
- Chau, A. L., Li, X., and Yu, W. (2014). Support vector machine classification for large datasets using decision tree and fisher linear discriminant. *Future Generation Computer Systems*, 36:57–65.
- Chen, M., Hao, Y., Hwang, K., Wang, L., and Wang, L. (2017). Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*, 5:8869–8879.
- Cheng, J. and Baldi, P. (2007). Improved residue contact prediction using support vector machines and a large feature set. *BMC Bioinformatics*, 8:113.
- Chou, P. Y. and Fasman, G. D. (1974). Prediction of protein conformation. *Biochemistry*, 13(2):222–245.
- Chowdhury, S., Lee, M. C., Xiong, G., and Duan, Y. (2003). Ab initio folding simulation of the trp-cage mini-protein approaches nmr resolution. *Journal of Molecular Biology*, 327(3):711–717.
- CNTK (Accessed: 8/1/2018). <https://github.com/Microsoft/CNTK>.
- Cocco, S., Monasson, R., and Weigt, M. (2013). Inference of hopfield-potts patterns from covariation in protein families: calculation and statistical error bars. *Journal of Physics: Conference Series*, 473:012010.
- Csaba, G., Birzele, F., and Zimmer, R. (2009). Systematic comparison of scop and cath: a new gold standard for protein structure analysis. *BMC Struct Biol*, 9:23.
- Dayhoff, M. O. and Schwartz, R. M. (1978). A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5:345–358.

- de Juan, D., Pazos, F., and Valencia, A. (2013). Emerging methods in protein co-evolution. *Nat Rev Genet*, 14(4):249–61.
- De Leonardis, E., Lutz, B., Ratz, S., Cocco, S., Monasson, R., Schug, A., and Weigt, M. (2015). Direct-coupling analysis of nucleotide coevolution facilitates rna secondary and tertiary structure prediction. *Nucleic Acids Res*, 43(21):10444–55.
- de Oliveira, S. H., Shi, J., and Deane, C. M. (2016). Comparing co-evolution methods and their application to template-free protein structure prediction. *Bioinformatics*.
- Deeplearning4j (Accessed: 8/1/2018). <https://deeplearning4j.org>.
- DeLano, W. (2002). Pymol: An open-source molecular graphics tool. *CCP4 Newsletter On Protein Crystallography*, 40:82–92.
- Dickson, R. J., Wahl, L. M., Fernandes, A. D., and Gloor, G. B. (2010). Identifying and seeing beyond multiple sequence alignment errors using intra-molecular protein covariation. *PLoS One*, 5(6):e11082.
- Dill, K. A. and MacCallum, J. L. (2012). The protein-folding problem, 50 years on. *Science*, 338(6110):1042–6.
- Djork-Arne Clevert, Thomas Unterthiner, S. H. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv:1511.07289*.
- Dor, O. and Zhou, Y. (2007). Achieving 80structure prediction by large-scale training. *Proteins*, 66(4):838–45.
- dos Santos, R. N., Morcos, F., Jana, B., Andricopulo, A. D., and Onuchic, J. N. (2015). Dimeric interactions and complex formation using direct coevolutionary couplings. *Scientific Reports*, 5.
- Drozdetskiy, A., Cole, C., Procter, J., and Barton, G. J. (2015). Jpred4: a protein secondary structure prediction server. *Nucleic Acids Res*, 43(W1):W389–94.
- Duan, Y. and Kollman, P. A. (1998). Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, 282(5389):740–4.
- Dunn, S. D., Wahl, L. M., and Gloor, G. B. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340.
- Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics*, 14(9):755–63.
- Eddy, S. R. (2009). A new generation of homology search tools based on probabilistic inference. *Genome Inform*, 23(1):205–11.

- Edgar, R. C. (2004). Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, 32(5):1792–7.
- Edgar, R. C. and Sjolander, K. (2004). A comparison of scoring functions for protein sequence profile alignment. *Bioinformatics*, 20(8):1301–8.
- Ehrlich, P. R. and Raven, P. H. (1964). Butterflies and plants: A study in coevolution. *Evolution*, 18(4):586.
- Ekeberg, M., Hartonen, T., and Aurell, E. (2014). Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *Journal of Computational Physics*, 276:341–356.
- Ekeberg, M., LÅvkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013). Improved contact prediction in proteins: Using pseudolikelihoods to infer potts models. *Phys. Rev. E*, 87:012707.
- Faraggi, E., Zhang, T., Yang, Y., Kurgan, L., and Zhou, Y. (2012). Spine x: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *J Comput Chem*, 33(3):259–67.
- Fariselli, P., Olmea, O., Valencia, A., and Casadio, R. (2001). Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. *Proteins*, Suppl 5:157–62.
- Finn, R. D., Bateman, A., Clements, J., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Heger, A., Hetherington, K., Holm, L., Mistry, J., Sonnhammer, E. L., Tate, J., and Punta, M. (2014). Pfam: the protein families database. *Nucleic Acids Res*, 42(Database issue):D222–30.
- Finn, R. D., Clements, J., and Eddy, S. R. (2011). Hmmer web server: interactive sequence similarity searching. *Nucleic Acids Res*, 39(Web Server issue):W29–37.
- Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., Potter, S. C., Punta, M., Qureshi, M., Sangrador-Vegas, A., Salazar, G. A., Tate, J., and Bateman, A. (2016). The pfam protein families database: towards a more sustainable future. *Nucleic Acids Res*, 44(D1):D279–85.
- Finn, R. D., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J. E., Gavin, O. L., Gunasekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E. L., Eddy, S. R., and Bateman, A. (2010). The pfam protein families database. *Nucleic Acids Res*, 38(Database issue):D211–22.
- Fischer, D. and Eisenberg, D. (1996). Protein fold recognition using sequence-derived predictions. *Protein Sci*, 5(5):947–55.

- Fiser, A. (2010). Template-based protein structure modeling. *Methods Mol Biol*, 673:73–94.
- Fleishman, S. J., Leaver-Fay, A., Corn, J. E., Strauch, E. M., Khare, S. D., Koga, N., Ashworth, J., Murphy, P., Richter, F., Lemmon, G., Meiler, J., and Baker, D. (2011). Rosettascripts: a scripting language interface to the rosetta macromolecular modeling suite. *PLoS One*, 6(6):e20161.
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–41.
- Ginalski, K., Elofsson, A., Fischer, D., and Rychlewski, L. (2003). 3d-jury: a simple approach to improve protein structure predictions. *Bioinformatics*, 19(8):1015–8.
- github/tensorflow (Accessed: 01/12/2018). <https://github.com/tensorflow/tensorflow>.
- Glasmachers, T. (2017). Limits of end-to-end learning. *arXiv:1704.08305*.
- Gloor, G. B., Martin, L. C., Wahl, L. M., and Dunn, S. D. (2005). Mutual information in protein multiple sequence alignments reveals two classes of coevolving positions. *Biochemistry*, 44(19):7156–65.
- Gobel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins*, 18(4):309–17.
- Goh, C. S., Bogan, A. A., Joachimiak, M., Walther, D., and Cohen, F. E. (2000). Co-evolution of proteins with their interaction partners. *Journal of Molecular Biology*, 299(2):283–293.
- Golik, P., Doetsch, P., and Ney, H. (2013). Cross-entropy vs. squared error training: a theoretical and experimental comparison. *INTERSPEECH*, pages 1756–1760.
- Golkov, V., Skwark, M. J., Golkov, A., Dosovitskiy, A., Brox, T., Meiler, J., and Cremers, D. (2016). Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images. *30th Conference on Neural Information Processing Systems*.
- Gomes, M., Hamer, R., Reinert, G., and Deane, C. M. (2012). Mutual information and variants for protein domain-domain contact prediction. *BMC Res Notes*, 5:472.
- Govindarajan, S., Recabarren, R., and Goldstein, R. A. (1999). Estimating the total number of protein folds. *Proteins*, 35(4):408–14.
- Hadley, C. and Jones, D. T. (1999). A systematic comparison of protein structure classifications: Scop, cath and fssp. *Structure*, 7(9):1099–1112.

- Halabi, N., Rivoire, O., Leibler, S., and Ranganathan, R. (2009). Protein sectors: evolutionary units of three-dimensional structure. *Cell*, 138(4):774–86.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*.
- He, B., Mortuza, S. M., Wang, Y., Shen, H., and Zhang, Y. (2017). Nebcon: Protein contact map prediction using neural network training coupled with naïve bayes classifiers.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv*, page 1502.01852.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. pages 770–778.
- Heffernan, R., Dehzangi, A., Lyons, J., Paliwal, K., Sharma, A., Wang, J., Sattar, A., Zhou, Y., and Yang, Y. (2016). Highly accurate sequence-based prediction of half-sphere exposures of amino acid residues in proteins. *Bioinformatics*, 32(6):843–849.
- Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., Sattar, A., Yang, Y., and Zhou, Y. (2015). Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci Rep*, 5:11476.
- Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, 89(22):10915–9.
- HHblits (Accessed: 8/1/2018). http://wwwuser.gwdg.de/~compbiol/uniclust/2017_10/.
- Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116.
- Hopf, T. A., Colwell, L. J., Sheridan, R., Rost, B., Sander, C., and Marks, D. S. (2012). Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, 149(7):1607–21.
- Hopf, T. A., Scharfe, C. P., Rodrigues, J. P., Green, A. G., Kohlbacher, O., Sander, C., Bonvin, A. M., and Marks, D. S. (2014). Sequence co-evolution gives 3d contacts and structures of protein complexes. *Elife*, 3.
- Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., and Ravikumar, P. (2014). Quic: Quadratic approximation for sparse inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1):2911–2947.

- Hu, J., Shen, L., and Sun, G. (2017). Squeeze-and-excitation networks. *arXiv:1709.01507*.
- Huang, G., Liu, Z., Maaten, L. v. d., and Weinberger, K. Q. (2016). Densely connected convolutional networks. *arXiv:1608.06993*.
- Inskip, W. P., Rusch, D. B., Jay, Z. J., Herrgard, M. J., Kozubal, M. A., Richardson, T. H., Macur, R. E., Hamamura, N., Jennings, R., Fouke, B. W., Reysenbach, A. L., Roberto, F., Young, M., Schwartz, A., Boyd, E. S., Badger, J. H., Mathur, E. J., Ortmann, A. C., Bateson, M., Geesey, G., and Frazier, M. (2010). Metagenomes from high-temperature chemotrophic systems reveal geochemical controls on microbial community structure and function. *PLoS One*, 5(3):e9773.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*.
- Jeon, J. H., Kim, J. T., Kang, S. G., Lee, J. H., and Kim, S. J. (2009). Characterization and its potential application of two esterases derived from the arctic sediment metagenome. *Mar Biotechnol (NY)*, 11(3):307–16.
- JGI (Accessed: 01/12/2018). <https://gold.jgi.doe.gov>.
- JMol (Accessed: 8/1/2018). <http://jmol.sourceforge.net>.
- Johnson, L. S., Eddy, S. R., and Portugaly, E. (2010). Hidden markov model speed heuristic and iterative hmm search procedure. *BMC Bioinformatics*, 11:431.
- Jones, D. T., Buchan, D. W. A., Cozzetto, D., and Pontil, M. (2012). Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190.
- Jones, D. T., Singh, T., Kosciulek, T., and Tetchner, S. (2015). Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*, 31(7):999–1006.
- Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A32:922–923.
- Kabsch, W. (1978). A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A34:827–828.
- Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–637.
- Kajan, L., Hopf, T. A., Kalas, M., Marks, D. S., and Rost, B. (2014). Freecontact: fast and free software for protein contact prediction from residue co-evolution. *BMC Bioinformatics*, 15:85.

- Kallberg, M., Wang, H., Wang, S., Peng, J., Wang, Z., Lu, H., and Xu, J. (2012). Template-based protein structure modeling using the raptorx web server. *Nat Protoc*, 7(8):1511–22.
- Kamisetty, H., Ovchinnikov, S., and Baker, D. (2013). Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era. *Proc Natl Acad Sci U S A*, 110(39):15674–9.
- Kang, K., Ouyang, W., Li, H., and Wang, X. (2016). Object detection from video tubelets with convolutional neural networks. *arXiv preprint arXiv*, page 1604.04053.
- Karam, S. D. and Lina (2017). A study and comparison of human and deep learning recognition performance under visual distortions. *arXiv:1705.02498*.
- Kaufmann, K. W., Lemmon, G. H., Deluca, S. L., Sheehan, J. H., and Meiler, J. (2010). Practically useful: what the rosetta protein modeling suite can do for you. *Biochemistry*, 49(14):2987–98.
- Kelley, L. A., Mezulis, S., Yates, C. M., Wass, M. N., and Sternberg, M. J. (2015). The phyre2 web portal for protein modeling, prediction and analysis. *Nat Protoc*, 10(6):845–58.
- Keras (Accessed: 8/1/2018). <https://keras.io>.
- Khor, B. Y., Tye, G. J., Lim, T. S., and Choong, Y. S. (2015). General overview on structure prediction of twilight-zone proteins. *Theor Biol Med Model*, 12:15.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *eprint arXiv:1412.6980*, page 1404.3840.
- Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *arXiv:1706.02515*.
- Konopka, B. M., Nebel, J. C., and Kotulska, M. (2012). Quality assessment of protein model-structures based on structural and functional similarities. *BMC Bioinformatics*, 13:242.
- Kopp, J. and Schwede, T. (2004). Automated protein structure homology modeling: a progress report. *Pharmacogenomics*, 5(4):405–16.
- Kosciolek, T. and Jones, D. T. (2016). Accurate contact predictions using covariation techniques and machine learning. *Proteins-Structure Function and Bioinformatics*, 84:145–151.
- Krogh, A., Brown, M., Mian, I. S., Sjolander, K., and Haussler, D. (1994). Hidden markov models in computational biology. applications to protein modeling. *J Mol Biol*, 235(5):1501–31.

- Krupa, P., Mozolewska, M. A., Wisniewska, M., Yin, Y., He, Y., Sieradzan, A. K., Ganzynkiewicz, R., Lipska, A. G., Karczynska, A., Slusarz, M., Slusarz, R., Gieldon, A., Czaplewski, C., Jagiela, D., Zaborowski, B., Scheraga, H. A., and Liwo, A. (2016). Performance of protein-structure predictions with the physics-based unres force field in casp11. *Bioinformatics*, 32(21):3270–3278.
- Kryshtafovych, A., Barbato, A., Monastyrskyy, B., Fidelis, K., Schwede, T., and Tramontano, A. (2016). Methods of model accuracy estimation can help selecting the best models from decoy sets: Assessment of model accuracy estimations in casp11. *Proteins-Structure Function and Bioinformatics*, 84:349–369.
- Kufareva, I. and Abagyan, R. (2012). Methods of protein structure comparison. *Methods Mol Biol*, 857:231–57.
- Kukic, P., Mirabello, C., Tradigo, G., Walsh, I., Veltri, P., and Pollastri, G. (2014). Toward an accurate prediction of inter-residue distances in proteins using 2d recursive neural networks. *Bmc Bioinformatics*, 15.
- Lapedes, A. S., Giraud, B., Liu, L., and Stormo, G. D. (1999). Correlated mutations in models of protein sequences: phylogenetic and structural effects. *Statistics in molecular biology and genetics*, 33:21.
- Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., Kaufman, K., Renfrew, P. D., Smith, C. A., Sheffler, W., Davis, I. W., Cooper, S., Treuille, A., Mandell, D. J., Richter, F., Ban, Y. E., Fleishman, S. J., Corn, J. E., Kim, D. E., Lyskov, S., Berrondo, M., Mentzer, S., Popovic, Z., Havranek, J. J., Karanicolas, J., Das, R., Meiler, J., Kortemme, T., Gray, J. J., Kuhlman, B., Baker, D., and Bradley, P. (2011). Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol*, 487:545–74.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–44.
- Lee, J., Freddolino, P. L., and Zhang, Y. (2017). Ab initio protein structure prediction. pages 3–35.
- Li, Y. and Zhang, Y. (2009). Remo: A new protocol to refine full atomic protein models from c-alpha traces by optimizing hydrogen-bonding networks. *Proteins*, 76(3):665–76.
- Lo Conte, L., Ailey, B., Hubbard, T. J., Brenner, S. E., Murzin, A. G., and Chothia, C. (2000). Scop: a structural classification of proteins database. *Nucleic Acids Res*, 28(1):257–9.
- Lockless, S. W. and Ranganathan, R. (1999). Evolutionarily conserved pathways of energetic connectivity in protein families. *Science*, 286(5438):295–9.

- Louie, B., Tarczy-Hornoch, P., Higdon, R., and Kolker, E. (2008). Validating annotations for uncharacterized proteins in *shewanella oneidensis*. *OMICS*, 12(3):211–5.
- Lovell, S. C. and Robertson, D. L. (2010). An integrated view of molecular coevolution in protein-protein interactions. *Mol Biol Evol*, 27(11):2567–75.
- Madera, M. and Gough, J. (2002). A comparison of profile hidden markov model procedures for remote homology detection. *Nucleic Acids Res*, 30(19):4321–8.
- Maghrabi, A. H. A. and McGuffin, L. J. (2017). Modfold6: an accurate web server for the global and local quality estimation of 3d protein models. *Nucleic Acids Res*.
- Malmstrom, L. (2005). *Genome-wide structural and functional protein characterization by ab initio protein structure prediction*. Department of Electrical Measurements Lund Institute of Technology Lund University.
- Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C. (2011). Protein 3d structure computed from evolutionary sequence variation. *PLoS One*, 6(12):e28766.
- Marks, D. S., Hopf, T. A., and Sander, C. (2012a). Protein structure prediction from sequence variation. *Nat Biotechnol*, 30(11):1072–80.
- Marks, D. S., Hopf, T. A., and Sander, C. (2012b). Protein structure prediction from sequence variation. *Nat Biotechnol*, 30(11):1072–80.
- Martin, L. C., Gloor, G. B., Dunn, S. D., and Wahl, L. M. (2005). Using information theory to search for co-evolving residues in proteins. *Bioinformatics*, 21(22):4116–24.
- MATLAB (Accessed: 15/3/2019). https://uk.mathworks.com/help/matlab/matlab_prog/floating-point-numbers.html.
- McGuffin, L. J., Bryson, K., and Jones, D. T. (2000). The psipred protein structure prediction server. *Bioinformatics*, 16(4):404–5.
- McGuffin, L. J., Buenavista, M. T., and Roche, D. B. (2013). The modfold4 server for the quality assessment of 3d protein models. *Nucleic Acids Res*, 41(Web Server issue):W368–72.
- McLaughlin, R. N., J., Poelwijk, F. J., Raman, A., Gosal, W. S., and Ranganathan, R. (2012). The spatial architecture of protein function and adaptation. *Nature*, 491(7422):138–42.
- Meier, A. and Soding, J. (2015). Automatic prediction of protein 3d structures by probabilistic multi-template homology modeling. *PLoS Comput Biol*, 11(10):e1004343.

- Michel, M., Hayat, S., Skwark, M. J., Sander, C., Marks, D. S., and Elofsson, A. (2014). Pconsfold: improved contact predictions improve protein models. *Bioinformatics*, 30(17):i482–8.
- Mirabello, C. and Pollastri, G. (2013). Porter, paleale 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics*, 29(16):2056–8.
- Mirdita, M., von den Driesch, L., Galiez, C., Martin, M. J., Soding, J., and Steinegger, M. (2017). Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res*, 45(D1):D170–D176.
- Mirjalili, V., Noyes, K., and Feig, M. (2014). Physics-based protein structure refinement through multiple molecular dynamics trajectories and structure averaging. *Proteins*, 82 Suppl 2:196–207.
- Mitchell, A., Bucchini, F., Cochrane, G., Denise, H., ten Hoopen, P., Fraser, M., Pesseat, S., Potter, S., Scheremetjew, M., Sterk, P., and Finn, R. D. (2016). Ebi metagenomics in 2016—an expanding and evolving resource for the analysis and archiving of metagenomic data. *Nucleic Acids Res*, 44(D1):D595–603.
- Mitchell, T. (1997). *Machine Learning*. McGraw Hill.
- Monastyrskyy, B., D’Andrea, D., Fidelis, K., Tramontano, A., and Kryshtafovych, A. (2014). Evaluation of residue-residue contact prediction in casp10. *Proteins*, 82 Suppl 2:138–53.
- Monastyrskyy, B., D’Andrea, D., Fidelis, K., Tramontano, A., and Kryshtafovych, A. (2016). New encouraging developments in contact prediction: Assessment of the casp11 results. *Proteins*, 84 Suppl 1:131–44.
- Morcos, F., Jana, B., Hwa, T., and Onuchic, J. N. (2013). Coevolutionary signals across protein lineages help capture multiple protein conformations. *Proc Natl Acad Sci U S A*, 110(51):20533–8.
- Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc Natl Acad Sci U S A*, 108(49):E1293–301.
- Moult, J. (2005). A decade of casp: progress, bottlenecks and prognosis in protein structure prediction. *Curr Opin Struct Biol*, 15(3):285–9.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2014). Critical assessment of methods of protein structure prediction (casp)—round x. *Proteins*, 82 Suppl 2:1–6.

- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2016a). Critical assessment of methods of protein structure prediction: Progress and new directions in round xi. *Proteins*, 84 Suppl 1:4–14.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2016b). Critical assessment of methods of protein structure prediction: Progress and new directions in round xi. *Proteins*, 84 Suppl 1:4–14.
- Muggleton, S., King, R. D., and Sternberg, M. J. E. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, 5(7):647–657.
- Muppalaneni, N. B. and Gunjan, V. K. (2015). *Computational Intelligence Techniques for Comparative Genomics*. Springer Singapore.
- Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, pages 807–814, USA. Omnipress.
- NCBI (Accessed: 8/1/2018). <https://www.ncbi.nlm.nih.gov/refseq/>.
- Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.
- Neher (1994). How frequent are correlated changes in families of protein sequences? *PNAS*, 91(1):5.
- Ogden, T. H. and Rosenberg, M. S. (2006). Multiple sequence alignment accuracy and phylogenetic inference. *Syst Biol*, 55(2):314–28.
- Oldziej, S., Czaplewski, C., Liwo, A., Chinchio, M., Nancias, M., Vila, J. A., Khalili, M., Arnautova, Y. A., Jagielska, A., Makowski, M., Schafroth, H. D., Kazmierkiewicz, R., Ripoll, D. R., Pillardy, J., Saunders, J. A., Kang, Y. K., Gibson, K. D., and Scheraga, H. A. (2005). Physics-based protein-structure prediction using a hierarchical protocol based on the unres force field: assessment in two blind tests. *Proc Natl Acad Sci U S A*, 102(21):7547–52.
- Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. (1997). Cath—a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–108.
- Oulas, A., Pavloudi, C., Polymenakou, P., Pavlopoulos, G. A., Papanikolaou, N., Koutoulas, G., Arvanitidis, C., and Iliopoulos, I. (2015). Metagenomics: tools and insights for analyzing next-generation sequencing data derived from biodiversity studies. *Bioinform Biol Insights*, 9:75–88.

- Ovchinnikov, S., Kamisetty, H., and Baker, D. (2014). Robust and accurate prediction of residue-residue interactions across protein interfaces using evolutionary information. *Elife*, 3:e02030.
- Ovchinnikov, S., Kim, D. E., Wang, R. Y., Liu, Y., DiMaio, F., and Baker, D. (2016). Improved de novo structure prediction in casp11 by incorporating coevolution information into rosetta. *Proteins*, 84 Suppl 1:67–75.
- Ovchinnikov, S., Kinch, L., Park, H., Liao, Y., Pei, J., Kim, D. E., Kamisetty, H., Grishin, N. V., and Baker, D. (2015). Large-scale determination of previously unsolved protein structures using evolutionary information. *Elife*, 4:e09248.
- Ovchinnikov, S., Park, H., Kim, D. E., DiMaio, F., and Baker, D. (2017a). Protein structure prediction using rosetta in casp12. *Proteins*.
- Ovchinnikov, S., Park, H., Varghese, N., Huang, P.-S., Pavlopoulos, G. A., Kim, D. E., Kamisetty, H., Kyrpides, N. C., and Baker, D. (2017b). Protein structure determination using metagenome sequence data. *Science*, 355(6322):294–298.
- Pavlopoulou, A. and Michalopoulos, I. (2011). State-of-the-art bioinformatics protein structure prediction tools (review). *Int J Mol Med*, 28(3):295–310.
- Pazos, F., HelmerCitterich, M., Ausiello, G., and Valencia, A. (1997). Correlated mutations contain information about protein-protein interaction. *Journal of Molecular Biology*, 271(4):511–523.
- Pazos, F. and Valencia, A. (2008). Protein co-evolution, co-adaptation and interactions. *EMBO J*, 27(20):2648–55.
- PDB (Accessed: 8/1/2018). <https://www.rcsb.org/pdb/home/home.do>.
- Pearson, W. R. (2013). Selecting the right similarity-scoring matrix. *Curr Protoc Bioinformatics*, 43:3 5 1–9.
- Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *J Am Stat Assoc*, 104(486):735–746.
- Pietal, M. J., Bujnicki, J. M., and Kozlowski, L. P. (2015). Gdfuzz3d: a method for protein 3d structure reconstruction from contact maps, based on a non-euclidean distance function. *Bioinformatics*, 31(21):3499–505.
- Pollastri, G. and Baldi, P. (2002). Prediction of contact maps by giohmms and recurrent neural networks using lateral propagation from all four cardinal corners. *Bioinformatics*, 18.
- Powell, M. J. D. (1977). Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12(1):241–254.

- Punta, M., Coghill, P. C., Eberhardt, R. Y., Mistry, J., Tate, J., Bournsnel, C., Pang, N., Forslund, K., Ceric, G., Clements, J., Heger, A., Holm, L., Sonnhammer, E. L., Eddy, S. R., Bateman, A., and Finn, R. D. (2012). The pfam protein families database. *Nucleic Acids Res*, 40(Database issue):D290–301.
- PyTorch (Accessed: 8/1/2018). <http://pytorch.org>.
- Read, R. J. and Chavali, G. (2007). Assessment of casp7 predictions in the high accuracy template-based modeling category. *Proteins*, 69 Suppl 8:27–37.
- Remmert, M., Biegert, A., Hauser, A., and Soding, J. (2012). Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nat Methods*, 9(2):173–5.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc.
- Rigden, D. J. (2009). *From Protein Structure to Function with Bioinformatics*. Springer.
- Risler, J. L., Delorme, M. O., Delacroix, H., and Henaut, A. (1988). Amino acid substitutions in structurally related proteins a pattern recognition approach. *Journal of Molecular Biology*, 204(4):1019–1029.
- Rohl, C. A., Strauss, C. E. M., Misura, K. M. S., and Baker, D. (2004). Protein structure prediction using rosetta. *Methods Enzymol*, 383:66–93.
- Rose, P. W., Prlic, A., Bi, C., Bluhm, W. F., Christie, C. H., Dutta, S., Green, R. K., Goodsell, D. S., Westbrook, J. D., Woo, J., Young, J., Zardecki, C., Berman, H. M., Bourne, P. E., and Burley, S. K. (2015). The rcsb protein data bank: views of structural biology for basic and applied research and education. *Nucleic Acids Res*, 43(Database issue):D345–56.
- Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein Eng*, 12(2):85–94.
- Roy, A., Kucukural, A., and Zhang, Y. (2010). I-tasser: a unified platform for automated protein structure and function prediction. *Nat Protoc*, 5(4):725–38.
- Roy, A., Xu, D., Poisson, J., and Zhang, Y. (2011). A protocol for computer-based protein structure and function prediction. *J Vis Exp*, (57):e3259.
- Safarian, S., Rajendran, C., Muller, H., Preu, J., Langer, J. D., Ovchinnikov, S., Hirose, T., Kusumoto, T., Sakamoto, J., and Michel, H. (2016). Structure of a bd oxidase indicates similar mechanisms for membrane-integrated oxygen reductases. *Science*, 352(6285):583–6.

- Sali, A. and Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol*, 234(3):779–815.
- Salomon-Ferrer, R., Case, D. A., and Walker, R. C. (2013). An overview of the amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3(2):198–210.
- Savojardo, C., Fariselli, P., Martelli, P. L., and Casadio, R. (2013). Bcov: a method for predicting beta-sheet topology using sparse inverse covariance estimation and integer programming. *Bioinformatics*, 29(24):3151–7.
- Schmidhuber, J. (2015a). Deep learning in neural networks: an overview. *Neural Netw*, 61:85–117.
- Schmidhuber, J. (2015b). Deep learning in neural networks: an overview. *Neural Netw*, 61:85–117.
- Seemayer, S., Gruber, M., and Soding, J. (2014). Ccmpred—fast and precise prediction of protein residue-residue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–30.
- Sergey Zagoruyko, N. K. (2016). Wide residual networks. *arXiv:1605.07146*.
- Shindyalov, I., Kolchanov, N., and Sander, C. (1994). Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations. *Protein Eng*.
- Sieradzan, A. K., Krupa, P., Scheraga, H. A., Liwo, A., and Czaplewski, C. (2015). Physics-based potentials for the coupling between backbone- and side-chain-local conformational states in the united residue (unres) force field for protein simulations. *Journal of Chemical Theory and Computation*, 11(2):817–831.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–9.
- Simons, K. T., Kooperberg, C., Huang, E., and Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *J Mol Biol*, 268(1):209–25.
- Skolnick, J. and Kihara, D. (2001). Defrosting the frozen approximation: Prospector—a new approach to threading. *Proteins*, 15(42):319–31.
- Skwark, M. J., Abdel-Rehim, A., and Elofsson, A. (2013). Pconsc: combination of direct information methods and alignments improves contact prediction. *Bioinformatics*, 29(14):1815–6.

- Skwark, M. J., Croucher, N. J., Puranen, S., Chewapreecha, C., Pesonen, M., Xu, Y. Y., Turner, P., Harris, S. R., Beres, S. B., Musser, J. M., Parkhill, J., Bentley, S. D., Aurell, E., and Corander, J. (2017). Interacting networks of resistance, virulence and core machinery genes identified by genome-wide epistasis analysis. *PLoS Genet*, 13(2):e1006508.
- Skwark, M. J., Raimondi, D., Michel, M., and Elofsson, A. (2014). Improved contact predictions using the recognition of protein like contact patterns. *PLoS Comput Biol*, 10(11):e1003889.
- Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7.
- Soding, J. (2005). Protein homology detection by hmm-hmm comparison. *Bioinformatics*, 21(7):951–60.
- Sogin, M. L., Morrison, H. G., Huber, J. A., Mark Welch, D., Huse, S. M., Neal, P. R., Arrieta, J. M., and Herndl, G. J. (2006). Microbial diversity in the deep sea and the underexplored "rare biosphere". *Proc Natl Acad Sci U S A*, 103(32):12115–20.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
- Stein, R. R., Marks, D. S., and Sander, C. (2015). Inferring pairwise interactions from biological data using maximum-entropy probability models. *PLoS Comput Biol*, 11(7):e1004182.
- Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and UniProt, C. (2015). Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–32.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Tang, C. L. and Xiaoou (2014). Surpassing human-level face verification performance on lfw with gaussianface. *arXiv preprint arXiv*, page 1404.3840.
- Tang, Y., Huang, Y. J., Hopf, T. A., Sander, C., Marks, D. S., and Montelione, G. T. (2015). Protein structure determination by combining sparse nmr data with evolutionary couplings. *Nat Methods*, 12(8):751–4.
- Taylor, W. R. and Hatrick, K. (1994). Compensating changes in protein multiple sequence alignments. *Protein Eng*, 7(3):341–8.
- tensorflow (Accessed: 8/1/2018). <https://www.tensorflow.org>.

- Tetchner, S. (2016). *Computational modelling of multidomain proteins with covarying residue pairs*. Thesis.
- Tetchner, S., Kosciolatek, T., and Jones, D. T. (2014). Opportunities and limitations in applying coevolution-derived contacts to protein structure prediction. *Bio-Algorithms and Med-Systems*, 10(4).
- The UniProt, C. (2017). Uniprot: the universal protein knowledgebase. *Nucleic Acids Res*, 45(D1):D158–D169.
- Thompson, J. D., Gibson, T. J., and Higgins, D. G. (2002). Multiple sequence alignment using clustalw and clustalx. *Curr Protoc Bioinformatics*, Chapter 2:Unit 2 3.
- Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustal-w - improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680.
- Thompson, J. N. (1994). *The coevolutionary process*. University of Chicago Press.
- Toth-Petroczy, A., Palmedo, P., Ingraham, J., Hopf, T. A., Berger, B., Sander, C., and Marks, D. S. (2016). Structured states of disordered proteins from genomic sequences. *Cell*, 167(1):158–170 e12.
- Uguzzoni, G., John Lovis, S., Oteri, F., Schug, A., Szurmant, H., and Weigt, M. (2017). Large-scale identification of coevolution signals across homo-oligomeric protein interfaces by direct coupling analysis. *Proc Natl Acad Sci U S A*, 114(13):E2662–E2671.
- Uniprot (Accessed: 8/1/2018). <http://www.uniprot.org>.
- UniProt, C. (2015). Uniprot: a hub for protein information. *Nucleic Acids Res*, 43(Database issue):D204–12.
- Uniref (Accessed: 8/1/2018). <http://www.uniprot.org/help/uniref>.
- Uziela, K. and Wallner, B. (2016). Proq2: estimation of model accuracy implemented in rosetta. *Bioinformatics*, 32(9):1411–3.
- Vogel, C., Bashton, M., Kerrison, N. D., Chothia, C., and Teichmann, S. A. (2004). Structure, function and evolution of multidomain proteins. *Curr Opin Struct Biol*, 14(2):208–16.
- Walsh, I., Bau, D., Martin, A. J., Mooney, C., Vullo, A., and Pollastri, G. (2009). Ab initio and template-based prediction of multi-class distance maps by two-dimensional recursive neural networks. *BMC Struct Biol*, 9:5.

- Wang, G. and Dunbrack, R. L., J. (2003). Pisces: a protein sequence culling server. *Bioinformatics*, 19(12):1589–91.
- Wang, S., Li, W., Zhang, R., Liu, S., and Xu, J. (2016a). Coinfold: a web server for protein contact prediction and contact-assisted protein folding. *Nucleic Acids Res.*
- Wang, S., Li, Z., Yu, Y., and Xu, J. (2017a). Folding membrane proteins by deep transfer learning. *Cell Systems*, 5(3):202–211.e3.
- Wang, S., Peng, J., Ma, J., and Xu, J. (2016b). Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports*, 6:18962.
- Wang, S., Peng, J., Ma, J., and Xu, J. (2016c). Protein secondary structure prediction using deep convolutional neural fields. *Sci Rep*, 6:18962.
- Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. (2017b). Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Comput Biol*, 13(1):e1005324.
- Wang, Z. and Xu, J. (2013). Predicting protein contact map using evolutionary and physical constraints by integer programming. *Bioinformatics*, 29(13):i266–73.
- Webb, B. and Sali, A. (2014). Comparative protein structure modeling using modeller. *Curr Protoc Bioinformatics*, 47:5 6 1–32.
- Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. (2009). Identification of direct residue contacts in protein-protein interaction by message passing. *Proc Natl Acad Sci U S A*, 106(1):67–72.
- Weinreb, C., Riesselman, A. J., Ingraham, J. B., Gross, T., Sander, C., and Marks, D. S. (2016). 3d rna and functional interactions from evolutionary couplings. *Cell*, 165(4):963–75.
- Wikipedia/Dihedral Angle (Accessed: 8/1/2018). https://en.wikipedia.org/wiki/Dihedral_angle.
- Wikipedia/ImageNet (Accessed: 01/12/2018). <https://en.wikipedia.org/wiki/ImageNet>.
- Wikipedia/Protein Structure (Accessed: 8/1/2018). https://simple.wikipedia.org/wiki/Protein_structure#/media/File:Main_protein_structure_levels_en.svg.
- Wikipedia/Ramachandran (Accessed: 8/1/2018). https://en.wikipedia.org/wiki/G._N._Ramachandran.
- Wikipedia/Ramachandran Plot (Accessed: 8/1/2018). https://en.wikipedia.org/wiki/Ramachandran_plot.

- Wu, C. H., Apweiler, R., Bairoch, A., Natale, D. A., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M. J., Mazumder, R., O'Donovan, C., Redaschi, N., and Suzek, B. (2006). The universal protein resource (uniprot): an expanding universe of protein information. *Nucleic Acids Res*, 34(Database issue):D187–91.
- Wu, F. Y. (1982). The potts model. *Reviews of Modern Physics*, 54(1):235–268.
- Wu, S. and Zhang, Y. (2007). Lomets: a local meta-threading-server for protein structure prediction. *Nucleic Acids Res*, 35(10):3375–82.
- Xie, S., Girshick, R., Dollar, P., Tu, Z., and He, K. (2016). Aggregated residual transformations for deep neural networks. *arXiv:1611.05431*.
- Xu, D. and Xu, Y. (2004). Protein databases on the internet. *Curr Protoc Mol Biol*, Chapter 19:Unit 19 4.
- Xu, D. and Zhang, Y. (2013). Ab initio structure prediction for escherichia coli: towards genome-wide protein structure modeling and fold assignment. *Sci Rep*, 3:1895.
- Yang, J., Yan, R., Roy, A., Xu, D., Poisson, J., and Zhang, Y. (2015). The i-tasser suite: protein structure and function prediction. *Nat Methods*, 12(1):7–8.
- Yang, Y., Gao, J., Wang, J., Heffernan, R., Hanson, J., Paliwal, K., and Zhou, Y. (2016). Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Brief Bioinform*.
- Yaseen, A. and Li, Y. (2014). Context-based features enhance protein secondary structure prediction accuracy. *J Chem Inf Model*, 54(3):992–1002.
- Yeang, C. (2007). Detecting coevolution in and among protein domains. *PLOS Computational Biology*.
- Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. *arXiv preprint arXiv*, page 1609.05473.
- Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv:1605.07146*.
- Zhang, J. and Zong, C. (2015). Deep neural networks in machine translation: An overview. *IEEE Intelligent Systems*, 30(5):16–25.
- Zhang, K., Sun, M., Han, X., Yuan, X., Guo, L., and Liu, T. (2017). Residual networks of residual networks: Multilevel residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1.
- Zhang, R. Y. J. S. W. C. Z. (2015). A short review on protein secondary structure prediction methods. *Pattern Recognition in Computational Molecular Biology: Techniques and Approaches*, 1.

- Zhang, Y. and Skolnick, J. (2004a). Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc Natl Acad Sci U S A*, 101(20):7594–9.
- Zhang, Y. and Skolnick, J. (2004b). Scoring function for automated assessment of protein structure template quality. *Proteins*, 57(4):702–10.
- Zhang, Y. and Skolnick, J. (2004c). Spicker: a clustering approach to identify near-native protein folds. *J Comput Chem*, 25(6):865–71.
- Zhang, Y. and Skolnick, J. (2005). Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic Acids Res*, 33(7):2302–9.
- Zhang, Y. and Wu, S. (2009). Protein structure prediction. *Bioinformatics: Tools and Applications*.
- Zhao, F. and Xu, J. (2012). A position-specific distance-dependent statistical potential for protein structure and functional study. *Structure*, 20(6):1118–26.
- Zhou, Y. and Karplus, M. (1999). Interpreting the folding kinetics of helical proteins. *Nature*, 401(6751):400–3.
- Zwanzig, R., Szabo, A., and Bagchi, B. (1992). Levinthal’s paradox. *Proc Natl Acad Sci U S A*, 89(1):20–2.