

**Improving the Performance of Protein Model  
Synthesis from Electron-Density Maps**

Emad Muteb Alharbi

Department of Computer Science  
University of York

Dissertation submitted for the degree of  
*Doctor of Philosophy*

June, 2022

# Abstract

Proteins are large biological molecules and the building blocks of all cells in living organisms. Modelling their structure supports the understanding of their role in key biological processes, including the onset, evolution and cure of diseases. Nevertheless, protein model building is extremely challenging. Although the computational tools for protein model building (e.g., from crystallographic data sets) have improved significantly in recent years, they still perform poorly for protein structures for which only data sets with low resolution and affected by poor phase distributions are available.

This thesis introduces new methods that support and improve model building for such protein structures. We start with a systematic evaluation of all major automated crystallographic model-building pipelines using 1211 protein structures (202 at original resolution and 1009 at truncated resolutions). Using the results of this study as a baseline, we then propose and show the effectiveness of using pairwise pipeline combinations to build better protein models for many crystallographic data sets.

As the performance of individual pipelines and pipeline combinations depends on the input data set, we introduce a predictive machine learning model that recommends pipelines or pipeline combinations suitable for a given data set, helping researchers avoid the time-consuming running of pipelines likely to perform poorly. The model bases its predictions on statistical features calculated from the electron-density map, and is available as a freely accessible web application.

Finally, we introduce a neural network trained to recognise incorrect parts of a protein model during the building process. Developed using large training data sets newly created for this purpose, and integrated into the protein model building software Buccaneer, the neural networks enables Buccaneer to avoid these incorrect parts and to produce protein models with significantly improved completeness and fitting measures to crystallography data.



*In memory of my grandmother,*

*Rashedah Alharbi*

*1949 - 2021*

# Contents

<b>Abstract</b>	<b>1</b>
<b>List of Figures</b>	<b>8</b>
<b>List of Tables</b>	<b>12</b>
<b>Acknowledgements</b>	<b>15</b>
<b>Declaration</b>	<b>16</b>
<b>1 Introduction</b>	<b>18</b>
1.1 Motivation . . . . .	18
1.2 Contributions and thesis structure . . . . .	19
<b>2 Background</b>	<b>23</b>
2.1 Protein structure building . . . . .	23
2.1.1 Protein geometry . . . . .	23
2.1.1.1 Amino acids . . . . .	23
2.1.1.2 Torsion angles . . . . .	24
2.1.1.3 Electron-density map . . . . .	24
2.1.2 Techniques of obtaining three-dimensional structure . . . . .	26
2.1.3 Processing data collected by X-ray crystallography . . . . .	27
2.1.3.1 Asymmetric unit . . . . .	28
2.1.3.2 Miller indices . . . . .	28
2.1.3.3 The phase problem . . . . .	29
2.1.3.4 Representation of electron-density maps . . . . .	30
2.1.3.5 Resolution of electron-density maps . . . . .	30

---

2.1.3.6	Electron-density map modification . . . . .	31
2.2	Machine learning . . . . .	34
2.2.1	Decision trees . . . . .	34
2.2.2	Random forests . . . . .	34
2.2.3	Neural networks . . . . .	35
2.2.3.1	Feedforward neural networks . . . . .	35
2.2.3.2	Recurrent neural networks . . . . .	37
<b>3</b>	<b>Comparison of automated crystallographic model-building pipelines</b>	<b>39</b>
3.1	Abstract . . . . .	39
3.2	Introduction . . . . .	40
3.3	Pipelines and methods . . . . .	41
3.3.1	ARP/wARP . . . . .	41
3.3.2	Buccaneer . . . . .	41
3.3.3	PHENIX AutoBuild . . . . .	42
3.3.4	SHELXE . . . . .	42
3.4	Data sets . . . . .	43
3.5	Method of the comparison . . . . .	44
3.6	Results . . . . .	46
3.6.1	Overview . . . . .	46
3.6.2	Structure completeness . . . . .	48
3.6.3	R-work and R-free . . . . .	51
3.6.4	Structure correlation . . . . .	54
3.6.5	Pipeline execution time . . . . .	56
3.7	Discussion . . . . .	56
3.8	Data and methods . . . . .	59
<b>4</b>	<b>Pairwise running of automated crystallographic model-building pipelines</b>	<b>60</b>
4.1	Abstract . . . . .	60
4.2	Introduction . . . . .	61
4.3	Data sets . . . . .	61
4.4	Method of the pairwise running . . . . .	62
4.5	Results . . . . .	63

---

4.5.1	Overview . . . . .	63
4.5.2	Structure completeness . . . . .	64
4.5.3	R-free . . . . .	71
4.5.4	Structure correlation . . . . .	73
4.6	Discussion . . . . .	74
4.7	Data and methods . . . . .	76
<b>5</b>	<b>Predicting the performance of automated crystallographic model-building pipelines</b>	<b>77</b>
5.1	Abstract . . . . .	77
5.2	Introduction . . . . .	78
5.3	Predictive model . . . . .	80
5.3.1	Data sets . . . . .	80
5.3.2	Crystallographic model-building pipelines . . . . .	81
5.3.3	Protein structure evaluation . . . . .	81
5.3.4	Electron-density map features . . . . .	81
5.3.5	Predictive model training . . . . .	82
5.4	Predictive model evaluation . . . . .	84
5.4.1	Evaluation of crystallography data set features used for model training . . . . .	84
5.4.2	Evaluation of predictive model performance . . . . .	85
5.4.3	Evaluation of recommended pipeline variant . . . . .	90
5.5	Discussion . . . . .	94
5.6	Availability . . . . .	95
<b>6</b>	<b>Identifying incorrect fragments to improve backbone chain tracing using neural network in Buccaneer</b>	<b>97</b>
6.1	Abstract . . . . .	97
6.2	Introduction . . . . .	98
6.3	Method . . . . .	99
6.3.1	Creating the training data sets . . . . .	99
6.3.2	Features of small fragments . . . . .	100
6.3.2.1	Ramachandran angles . . . . .	101

---

6.3.2.2	Log likelihood score . . . . .	101
6.3.2.3	Density score . . . . .	101
6.3.2.4	Root mean square deviation . . . . .	102
6.3.2.5	Small fragment position . . . . .	102
6.3.3	Neural network architecture and training . . . . .	102
6.3.3.1	Data set preparation . . . . .	102
6.3.3.2	Neural network architecture . . . . .	103
6.3.3.3	Neural network training . . . . .	103
6.3.4	Using the neural network in Buccaneer . . . . .	104
6.4	Results . . . . .	106
6.4.1	Evaluation of neural network training . . . . .	106
6.4.2	Feature importance . . . . .	107
6.4.3	Evaluation of using the neural network in Buccaneer . . . . .	108
6.4.3.1	Evaluation of the decision tree . . . . .	109
6.4.3.2	Experimental phasing . . . . .	109
6.4.3.3	MR . . . . .	111
6.4.4	Evaluation of execution times . . . . .	112
6.4.5	Evaluation of using the neural network in Buccaneer running from ModelCraft . . . . .	112
6.5	Discussion . . . . .	113
<b>7</b>	<b>Conclusion</b>	<b>122</b>
7.1	Summary . . . . .	122
7.2	Limitations and future work . . . . .	124
<b>Appendix A Comparison of automated crystallographic model-building pipelines (additional results)</b>		<b>127</b>
A.1	Experimental results for the original data sets used in Buccaneer de- velopment . . . . .	127
A.2	Experimental results for synthetic data sets for the original data sets used in Buccaneer development . . . . .	140
A.3	Original resolutions without the Buccaneer development data sets . . .	153
A.4	Synthetic resolutions without Buccaneer development data sets . . . .	163

A.5	Reproducibility of the comparison experiment . . . . .	175
A.6	PDB codes used in the comparison . . . . .	177
A.7	SHELXE results for using default and optimised solvent fraction for the original data sets without the Buccaneer development data sets . .	178
A.8	Comparison of ARP/wARP run with and without R-free . . . . .	179
<b>Appendix B Pairwise running of automated crystallographic model-building pipelines (additional results)</b>		<b>180</b>
B.1	Experimental results for the original data sets used in Buccaneer de- velopment . . . . .	180
B.2	Experimental results for synthetic data sets for the original data sets used in Buccaneer development . . . . .	199
B.3	Experimental results for the original data sets without the Buccaneer development data sets . . . . .	218
B.4	Experimental results for the synthetic data sets without the Buccaneer development data sets . . . . .	237
B.5	The command line used to run the pipelines . . . . .	257
B.5.1	PHENIX AutoBuild . . . . .	257
B.5.2	ARP/wARP . . . . .	265
B.5.3	Buccaneer . . . . .	266
<b>Appendix C Identifying incorrect fragments to improve backbone chain trac- ing using neural network in Buccaneer (additional results)</b>		<b>268</b>
C.1	Comparison of R-work, R-free and structure correlation between Buc- caneer and Buccaneer with neural network . . . . .	268
<b>Bibliography</b>		<b>270</b>

# List of Figures

1.1	The data sets and model-building pipelines are used in each contributions chapter . . . . .	22
2.1	Amino acids structures . . . . .	25
2.2	Torsion angles of main chain residues . . . . .	26
2.3	Ramachandran plot example . . . . .	26
2.4	Number of solved structures per publication year . . . . .	27
2.5	X-ray crystallography experiment . . . . .	28
2.6	The difference between crystal, unit cell and asymmetric unit . . . . .	29
2.7	Miller indices examples . . . . .	29
2.8	The diffraction of X-ray and Miller indices . . . . .	30
2.9	The phase problem . . . . .	30
2.10	Electron-density map and protein model . . . . .	31
2.11	Examples of electron-density map resolutions . . . . .	32
2.12	Electron-density map modification . . . . .	33
2.13	Decision tree to predict an amino acid type . . . . .	36
2.14	Simple neural network example . . . . .	37
2.15	Sigmoid and tanh activation functions . . . . .	37
2.16	Feedforward neural network example . . . . .	38
3.1	Resolutions of the 202 original data sets . . . . .	44
3.2	Mean completeness for the protein models built for all NO-NCS data sets based on their resolution . . . . .	51
3.3	Mean residues incorrectly built for the protein models built for all NO-NCS data sets . . . . .	51

3.4	Mean completeness for the models built for the original NO-NCS data sets based on their initial map correlation . . . . .	52
3.5	Mean protein model R-work for the NO-NCS data sets . . . . .	54
3.6	Mean protein model R-free for the NO-NCS data sets . . . . .	55
3.7	Mean correlation between built protein model and final deposited protein model for NO-NCS data sets . . . . .	55
3.8	Mean pipeline execution times for the original NO-NCS data sets . . .	56
4.1	Structure completeness comparison for the models generated from the original data sets for the pairwise running . . . . .	67
4.2	Mean completeness for the protein models built for all data sets for the pairwise running . . . . .	69
4.3	Mean completeness for the models built for the original data sets based on their initial map correlation for the pairwise running . . . . .	69
4.4	Mean residues incorrectly built for the protein models built for all data sets for the pairwise running . . . . .	70
4.5	Four structures built by Buccaneer, PHENIX AutoBuild(Parrot) and their combinations, and compared to the deposited structures . . . . .	71
4.6	Comparison of R-free the structures generated from the original data sets for the pairwise running . . . . .	72
4.7	Mean protein model R-free for the data sets for the pairwise running .	73
4.8	Mean correlation between built protein model and final deposited protein model . . . . .	74
5.1	Analysis of the crystallographic model-building pipelines used in 3273 PDB protein-structure research papers published between 2010 and 2020	80
5.2	Ablation studies showing the features importance for the predictive model . . . . .	84
5.3	Prediction error for the ML predictive model and the median predictor for recently deposited and JCSG experimental phasing data sets. . . .	87
5.4	Mean absolute error (MAE) and root mean squared error (RMSE) of structure completeness and R-free/R-work for training and testing for the JCSG experimental phasing data sets and the MR data sets . . . .	88



5.5	Inference time for the predictive model for individual pipelines and pipeline combinations . . . . .	91
5.6	Execution time required to run all the pipeline variants . . . . .	92
5.7	Difference between the best completeness, R-free and R-work achieved by running all the pipeline variants and running the recommended pipeline variant for the JCSG data sets . . . . .	93
5.8	Difference between the best completeness, R-free and R-work achieved by running all the pipeline variants and running the recommended pipeline variant for the MR data sets . . . . .	93
6.1	An example of splitting a fragment into small fragments. The fragment split into three small fragments. . . . .	99
6.2	An example of four small fragments (F) and the distance between them. The matrix shows when two small fragments can be joined when the distance between them less than 4Å. . . . .	102
6.3	Creating the training data sets, the neural network architecture and using the neural network in Buccaneer. . . . .	105
6.4	Difference in loss score and AUC between training and validation data sets across the epochs . . . . .	107
6.5	Difference between the baseline model ( where the values of the features are not shuffled in training and validation) and the model where the feature values are shuffled to find out the features importance. . .	108
6.6	Comparison of structure completeness, structure correlation, R-work and R-free between Buccaneer and the Buccaneer with neural network (Buccaneer(NN)) variants using ten thresholds and the Freedman–Diaconis rule, for the JCSG data sets . . . . .	115
6.7	Comparison of structure completeness between Buccaneer and the Buccaneer(NN) variants using 10 thresholds and the Freedman–Diaconis rule, for the recently deposited experimental phasing data sets . . . . .	116
6.8	A protein structure built by Buccaneer and Buccaneer(NN) compared to the deposited structure. The structure PDB ID is 6HCZ and its resolution is 2.3 Å . . . . .	117

6.9	A protein structure built by Buccaneer and Buccaneer(NN) compared to the deposited structures. The structure PDB ID is 2GNR and its truncated resolution is 3.2 Å . . . . .	118
6.10	Comparison of structure completeness, R-work, R-free and structure correlation between Buccaneer and Buccaneer with neural network (Buccaneer(NN)) for the MR data sets . . . . .	119
6.11	Mean execution time of Buccaneer and Buccaneer(NN) for the JCSG original data sets. The structure sizes are grouped into classes, and the number of data sets in each class is reported under the graph. . . . .	120
6.12	Comparison of the mean structure completeness, R-work and R-free achieved by ModelCraft and i1 with and without neural network . . .	121

# List of Tables

3.1	Pipeline variants used in the comparison. . . . .	47
3.2	Complete and intermediate models produced by the pipelines used in the comparison for the original data sets . . . . .	48
3.3	Complete and intermediate models produced by the pipelines used in the comparison for the synthetic-resolution data sets . . . . .	48
3.4	Structure completeness comparison for the models generated from the original NO-NCS data sets . . . . .	49
3.5	Structure completeness comparison for the models generated from the original NO-NCS data sets with at least 5% higher structure completeness	49
3.6	Comparison of R-work/R-free for the models generated from the original NO-NCS data sets . . . . .	53
3.7	Comparison of R-work/R-free for the models generated from the original NO-NCS data sets with 5% lower . . . . .	53
4.1	Pipeline and pipeline combination identifiers (IDs) used to present the results. . . . .	64
4.2	Complete and intermediate models produced by the pipeline variants used in the pairwise running . . . . .	65
4.3	Mean and standard deviation (SD) for the structure completeness and R-free . . . . .	66
4.4	Structure completeness and R-free comparison for the original and synthetic data sets, indicating how often pairwise running outperforms either of the component pipelines . . . . .	68

5.1	Mean absolute error (MAE) and root mean squared error (RMSE) of structure completeness and R-free/R-work for two experimental phasing data sets and molecular replacement (MR) data sets. . . . .	86
5.2	Mean and standard deviation (SD) of the real and predicted structure evaluation measures for the JCSG experimental phasing data sets . . .	89
5.3	Mean and standard deviation (SD) of the real and predicted structure evaluation measures for the JCSG experimental phasing data sets for SHELXE and its combinations . . . . .	89
5.4	Mean and standard deviation (SD) of the real and predicted structure evaluation measures for the MR data sets . . . . .	90
5.5	Real structure completeness achieved by the pipeline that was used to solve the protein structure when deposited in the PDB and by the pipeline recommended by the predictive model, for the MR data sets.	94
6.1	Features used in training the neural network in addition to the electron-density map resolution . . . . .	101
6.2	Protein structure evaluation indicators; Buccaneer indicators, R-work and R-free. Indicated whether the indicator is better when has a higher or lower value. . . . .	106

# Acknowledgements

Firstly, I would like to thank my supervisors, Professor Radu Calinescu and Professor Kevin Cowtan, for their help, guidance, comments, and always being there when I need help or have a question. They always find time to reply to my emails and give detailed feedback even in their busy times. I enjoyed every moment in this project, not only because it was an interesting field but also because of how they supervised me during the project. My thanks also go to Professor Julie Wilson, my assessor, for her support in the thesis advisory panel meetings and to Professor Randy Read, my external examiner, for his comments which helped me to improve the thesis.

My thanks extend to both TASP and Cowtan's groups for the useful discussions in the seminars and the team meetings. Special thanks go to Dr Paul Bond for providing data sets and scripts to produce data sets. Additionally, I would like to thank the University of Tabuk for funding my PhD studies. I am grateful for Professor Mohammed Alwakeel, who believed in me and guided me in the first step of my academic career. I am thankful to Dr Umar Albalawi from the University of Tabuk for his help in the scholarship administrative support.

I am very grateful to the High-performance Computing team at the University of York, who were very helpful and supportive in solving the problems I faced on the HPC cluster. I also would like to thank the EGI federation for providing computing resources for web application hosting.

I would like to thank my mother Norah for her unconditional support and to my family: Saud, Mohammed, Ahmed, Layla, Feryal and Najla. Finally, I would like to extend my thanks to my grandfather Mutairan and my cousin Abdullah.

# Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

- Emad Alharbi, Paul S. Bond, Radu Calinescu, and Kevin Cowtan. **“Comparison of automated crystallographic model-building pipelines.”** Acta Crystallographica Section D: Structural Biology 75, no. 12 (2019): 1119-1128.
  - Contributors to the research paper: PB created the synthetic data sets. EA designed the study and conducted the comparison, analysed the results and wrote the manuscript. RC and KC supervised the work. All the authors reviewed and edited the manuscript.
- Emad Alharbi, Radu Calinescu, and Kevin Cowtan. **“Pairwise running of automated crystallographic model-building pipelines.”** Acta Crystallographica Section D: Structural Biology 76, no. 9 (2020).
  - Contributors to the research paper: EA designed the study and conducted the pairwise running experiment, analysed the results and wrote the manuscript. RC and KC supervised the work. All the authors reviewed and edited the manuscript.
- Emad Alharbi, Paul Bond, Radu Calinescu, and Kevin Cowtan. **“Predicting the performance of automated crystallographic model-building pipelines.”** Acta Crystallographica Section D: Structural Biology 77, no. 12 (2021).
  - Contributors to the research paper: EA designed the study, implemented the predictive model, analysed the results and wrote the manuscript. PB

created a script to obtain experimental phasing data sets from the Protein data bank. RC and KC supervised the work. All the authors reviewed and edited the manuscript.

# Introduction

## 1.1 Motivation

In the 1950s, the first protein structures were determined and, since then, more than 154,000 protein structures have been solved and deposited in the Protein Data Bank (PDB) [1, 2]. However, the number of solved protein structures is only a small fraction of protein structures that have not been solved yet. A frequently used method for determining a protein structure starts with crystallising the structure and then applying a determination technique, such as X-ray crystallography, to obtain an electron-density map, which is then used to interpret the coordinates of the protein structure atoms. A similar method to X-ray crystallography is cryogenic electron microscopy (cryo-EM), which is useful for the protein structures that are difficult to crystallize as the method is based on freezing the sample rather than the crystallization[3]. The folding of the protein structure can result in a complicated electron-density map that makes building the protein model manually very time-consuming.

The challenges faced during the building of protein structures include the *phase problem*; the reconstruction of the electron-density map needs intensities of waves (which can be measured from the experiment), the amplitudes (square root of the intensities) and the phase (which can not be measured from the experiment and describes the shift between the waves) [4, 5]. The phase problem may be solved by either molecular replacement or experimental phasing methods [6, 7]. These methods lead to electron-density maps with rather different properties: in the case of experimental phasing, the maps usually contain noise due to ambiguity in the experimental phasing, whereas in the molecular replacement case, the errors in the map can arise from bias towards the molecular replacement model. The resolution of the experimental observations, the



quality of experimental phasing or the similarity of the molecular replacement model, and many other features such as ice rings, which arise because the water to freeze to ice in macromolecular crystals as the X-ray crystallography is data collected at cryogenic temperatures [8, 9], may also affect the quality of the data. Each of these factors impacts the building of the protein structure in different ways [10, 11, 12].

To automate the building of the protein structure, several automated pipelines have been developed. These pipelines include ARP/wARP [13, 14, 15, 16, 17], Buccaneer [18, 19], PHENIX AutoBuild [20], SHELXE [21, 22, 23, 24]. Protein structures built using these pipelines can differ in the evaluation measures, and sometimes, the difference can be significant.

Since the early releases of these pipelines, major improvements have been made to enable them to build more complete models. However, they still cannot build complete models in difficult cases, for example, for electron-density maps with low resolutions or poor phases.

Recent advances in machine learning (ML) have enabled the use of ML techniques to further progress protein model building [25]. Machine learning is used at different stages of the process to solve protein structures, including serial crystallography and model building. In an example of the use of machine learning in model building, a neural network was trained to identify incorrect residues in a final model [26]. Moreover, machine learning was used to improve the tracing of the protein structure backbone by finding “good” fragments [27]. However, as many challenges remain to obtain a complete protein structure that requires minimum manual building, further machine learning techniques are required to make the built protein structure models sufficiently accurate.

## **1.2 Contributions and thesis structure**

In this thesis, we first determine a baseline for the current model-building pipelines through systematically evaluating their performance for a large number of crystallography data sets. Moreover, we examine the improvements achieved by running these pipelines in pairwise combinations in order to gain the most from the complementarity of their algorithms. However, the pairwise running method leads to a large number of

pipeline combinations, each with different levels of performance across the data sets. To avoid the need to run all these pipeline combinations and the individual pipelines on each data set, we introduce a machine learning model capable of predicting the performance of the pipelines and their combinations for a given data set. Finally, to alleviate the problem of placing incorrect fragments into protein models, we introduce a neural network trained to identify and remove such fragments during the model building process. The use of this neural network within a new version of the protein model building software tool Buccaneer [18, 19, 28] can significantly improve the protein models built by the tool.

These contributions and structure of the thesis are summarised below.

- **Chapter 2: Background**

The chapter provides background information about the techniques used to obtain models of three-dimensional protein structures, and about the existing protein model-building pipelines. Additionally, the chapter introduces machine learning and neural network concepts and techniques used in later chapters of the thesis.

- **Chapter 3: A performance baseline for protein model-building pipelines (contribution).**

The chapter presents an extensive comparison of protein-model building pipelines ARP/wARP, Buccaneer, Phenix AutoBuild and SHELXE. The four widely used pipelines were run on large number of crystallography data sets that range from easy to challenging and compared based on the structure completeness and R-work/R-free of the protein models they generated for these data sets.

- **Chapter 4: Pairwise running of the protein model-building pipelines (contribution).**

We propose and examine the usefulness of combining these pipelines to improve the built protein structures by running them in pairwise combinations. The chapter presents an evaluation of combining these pipelines based on the structure completeness and R-free.

- **Chapter 5: Predicting the performance of the protein model-building pipelines (contribution).**

Identifying the best pipeline or pipeline combination to use for a protein structure is difficult, as the pipeline performance differs significantly from one protein structure to another. The chapter presents a machine learning model trained to predict the performance of the protein-model building pipelines. We start by analysing the uses of these pipelines and then explain how we trained a machine learning model to predict structure completeness, R-free and R-work they can each achieve for a give crystallography data set. We evaluated the machine learning model based on RMSE, MAE and through comparing its accuracy to that of a zero-R predictive model. The predictive model is freely available as an online tool.

- **Chapter 6: Avoiding the use of incorrect fragments in the protein model (contribution).**

Placing incorrect fragments during the building process leads to wrong residues being sequenced, and therefore to a poor protein model. We introduce a neural network trained to identify incorrect fragments, and show how its use within Buccaneer can help remove such fragments in order to improve backbone tracing. Buccaneer augmented with the neural network produces protein models with significantly improved structure completeness for experimental phasing data sets. The chapter presents the method used to label the data samples used to train the network, the evaluation of the trained neural network, and its use within Buccaneer.

- **Chapter 7: Conclusion**

The chapter summarises the achievements and limitations of the research presented in the thesis, and proposes directions for future work.

Figure 1.1 shows the data sets and the model building pipelines used in each research-contributions chapter, and indicates where the results from a chapter are used in other contributions chapters.

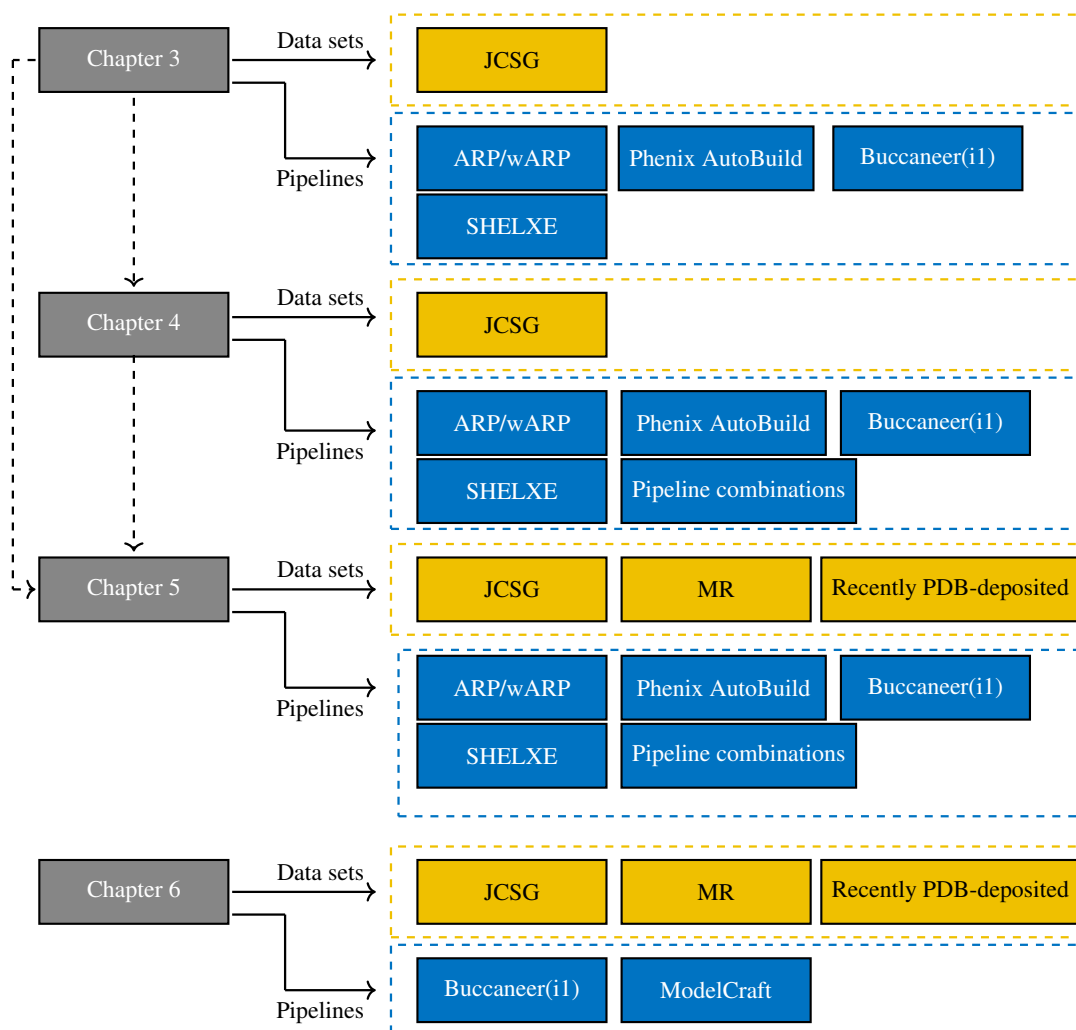


Figure 1.1: Model-building pipelines and data sets (PDB=Protein Data Bank, JCSG=Joint Center for Structural Genomics, MR=molecular replacement) used in each contributions chapter. The dashed arrows indicate where the results from a chapter are used in other chapters.

# Background

## 2.1 Protein structure building

### 2.1.1 Protein geometry

Proteins are macromolecules that perform essential biological functions which depend on their three-dimensional structure. A protein is a chain of amino acids, which are chemical compounds that contain nitrogen, carbon, hydrogen, oxygen and a unique side chain. The next sections introduce key concepts and terminology about amino acids, torsion angles of the amino acids, and the electron-density map.

#### 2.1.1.1 Amino acids

More than 300 amino acids have been identified in nature; however, only twenty types of amino acids are needed to produce common human proteins and most of other proteins [29]. All amino acids contain a carbon atom called  $C^\alpha$  (C alpha) located in the centre of the amino acid,  $NH_2$  and  $COOH$ . However, these amino acids have different chemical properties:

1. An amino acid has a unique side-chain  $R$  with a different number of atoms bonded to the  $C^\alpha$  atom (Figure 2.1). However, some amino acids may have the same number of atoms.
2. Hydrophobicity and hydrophilicity of an amino acid, which means the amino acid interacts to water (hydrophilic) and those repel water (hydrophobic) [30].
3. Chemical bonds of atoms in an amino acid affect its chemical properties even if two amino acids have the same atomic composition, for example, Isoleucine and Leucine (Figure 2.1).

4. An amino acid is either positively, neutral or negatively charged. The attraction between amino acids is affected by their charges as the amino acids with the same charge interact and those with opposite charges repel each other [31].

Each one of these amino acids contains a different number of atoms, giving it a unique shape.

Each amino acid has a unique side-chain  $R$  with a different number of atoms bonded to the  $C^\alpha$  atom (Figure 2.1). Two amino acids are bonded together by the N-terminus side connected to the C-terminus in other amino acids [32]. We refer to amino acids as *residues* in the rest of this thesis.

### 2.1.1.2 Torsion angles

The geometrical structure of the main chain can be described using angles, known as *torsion angles*. The torsion angles describe the rotations between  $N - C^\alpha$ , called *Phi*  $\Phi$ , and between  $C^\alpha - C$ , called *Psi*  $\Psi$  (Figure 2.2). Ramachandran is the physicist who described these angles and designed a plot for exhibiting the angles' correctness [33]. The plot of Ramachandran shows  $\Psi$  on the horizontal axis and  $\Phi$  on the vertical axis, with both scales varying from  $-180$  to  $+180$ . The plot uses dots for representing each torsion angle of the amino acids on the axes for the angles' distribution. Figure 2.3 shows an example of a Ramachandran plot with three regions; favoured, allowed and disallowed regions. Residues in disallowed regions were results of steric hindrance, which is the non-bonded atoms that come close to each other and cause a rise in the energy and repulsions [34].

### 2.1.1.3 Electron-density map

The atoms of the residues are surrounded by electrons moving in orbital motion and creating a "cloud" called *electron-density* around these atoms. As a protein contains a number of these residues and each has its electrons, this results in a *map of electron density* being created. In the absence of the residues' coordinates, this density map can be used to interpret the residues positions. However, obtaining the density map needs special techniques to determine its 3D shape.

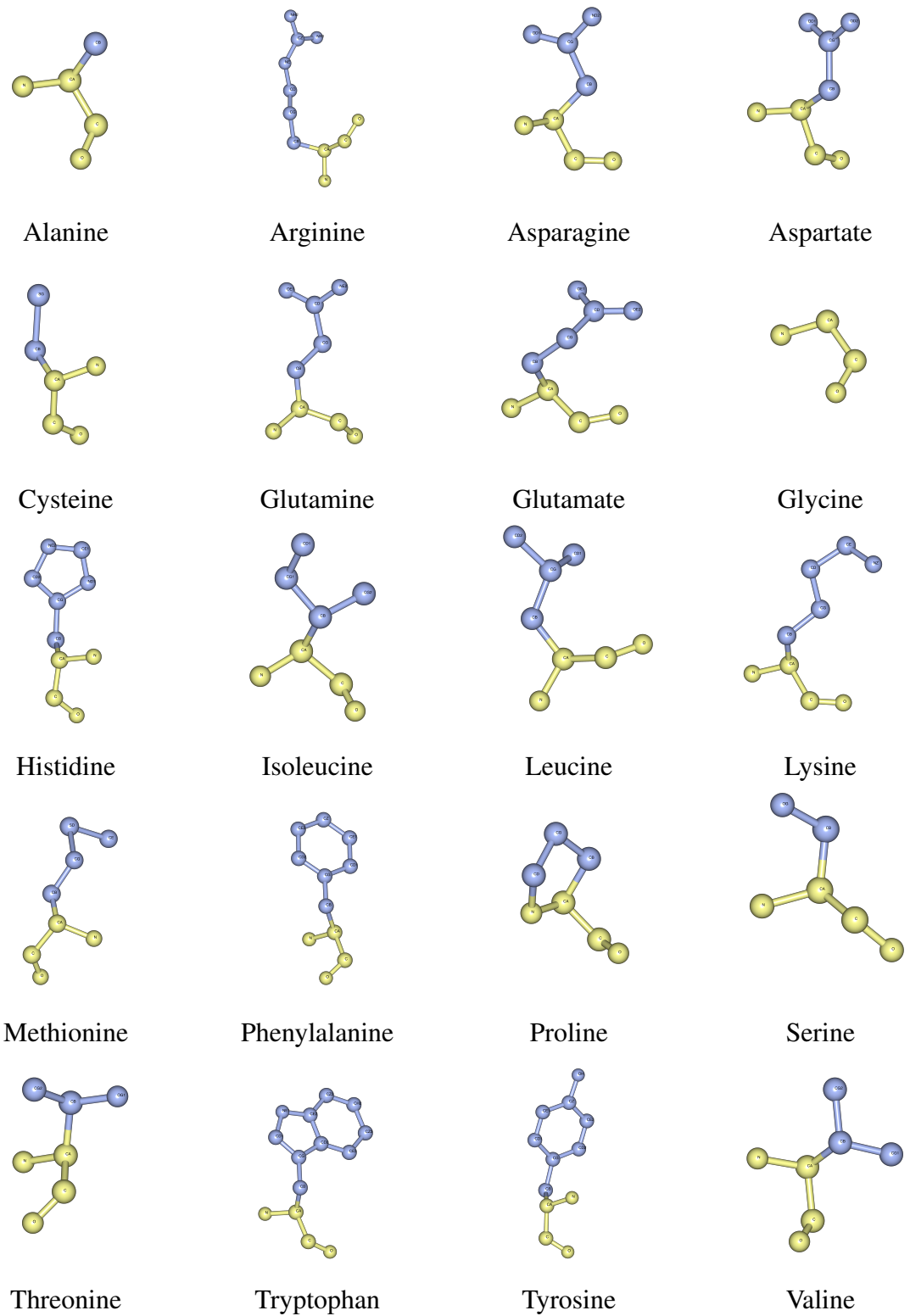


Figure 2.1: The twenty types of amino acids that have been identified in protein structures. The main chain is identical in all of them but they have different side chains, which determine the unique shape of each amino acid.

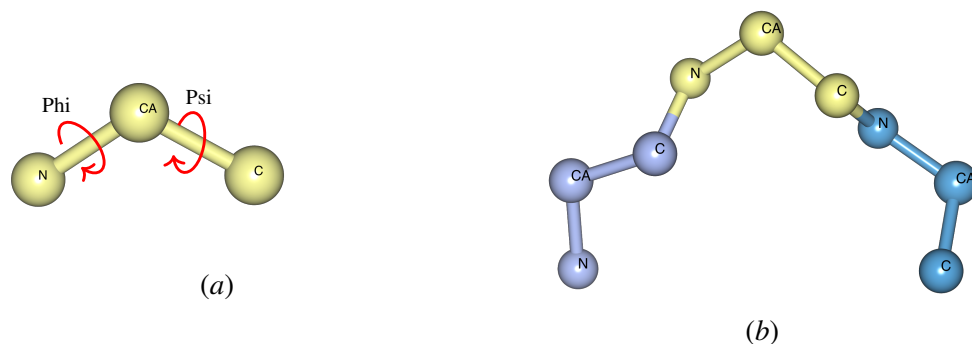


Figure 2.2: (a) The angles between  $N - C^\alpha$  (Phi  $\Phi$ ) and between  $C^\alpha - C$  (Psi  $\Psi$ ). (b) Torsion angles for bonded residues.

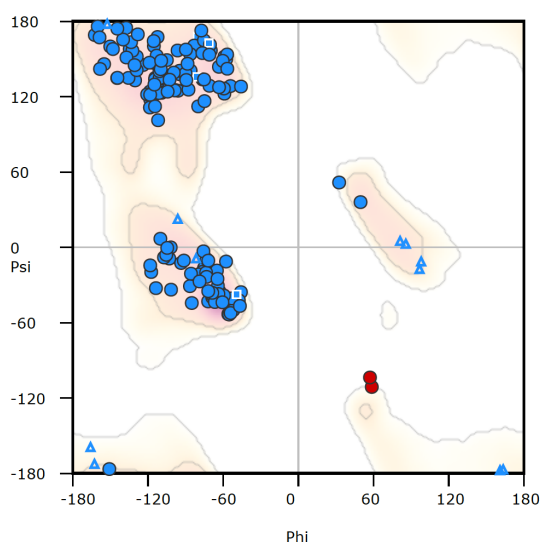


Figure 2.3: An example of Ramachandran's plot. Area in salmon colour shows favoured, allowed in light yellow and white for disallowed regions. Glycine and Proline are shown as triangles and squares, respectively, and other residues are shown as circles. Residues were in disallowed regions shown in red colour.

## 2.1.2 Techniques of obtaining three-dimensional structure

X-ray crystallography [35], nuclear magnetic resonance (NMR) [36], and electron microscopy (cryo-EM)[37, 38, 39] are the most used techniques to solve the protein structures, with the highest use for X-ray crystallography (Figure 2.4). To determine the structure of a protein using X-ray crystallography, a series of steps need to be conducted, starting by crystallizing the relevant molecule, collecting the molecule's diffraction, solving the phase problem, and then fitting the model into the density map.

Crystallisation is a process of organizing atoms or molecules into a regular solid structure. In X-ray crystallography, the crystal obtained from the crystallisation process is centred in the path of X-rays. When X-rays (electromagnetic waves) pass



through the crystal, the electrons scatter the wave with the same wavelength as the incident wave, and the scattered waves register on a photographic plate (Figure 2.5) [35]. The relation between the X-ray wavelength and its reflection is described by Bragg's law:

$$2d\sin\theta = n\lambda, \quad (2.1)$$

where  $d$  is the distance between the crystal planes,  $\lambda$  is the wavelength, and  $n$  is the diffraction order [40]. Bragg's law is used in X-ray crystallography to identify the crystal lattice, which can be described as an ordered array of points.

Once the diffraction spots are registered, they are indexed using Miller indices because each wave diffracted from a plane gives information about the structure of the molecules within the analysed crystal. The next step is to solve the phase problem and calculate the density map for use in fitting the model. The next sections describe each of these steps.

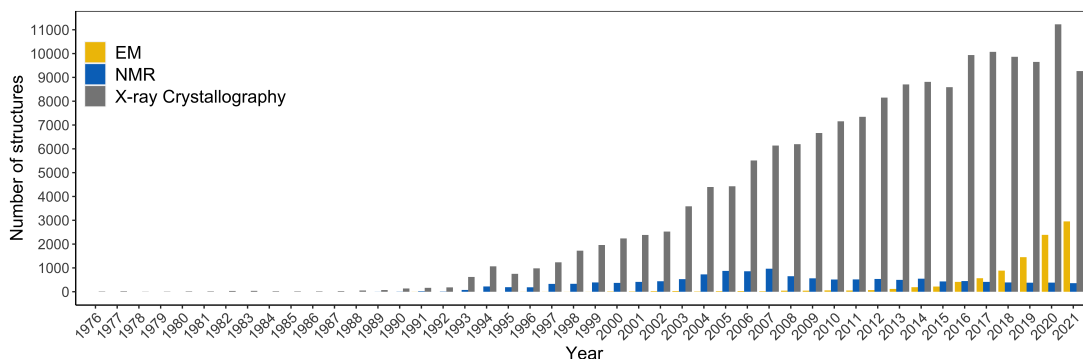


Figure 2.4: Number of solved structures per publication year by determination method as in 2021.

### 2.1.3 Processing data collected by X-ray crystallography

The data collected through X-ray crystallography is used to calculate the electron-density map

$$\rho(x, y, z) = \frac{1}{V} \sum_h \sum_k \sum_l |F(hkl)| \cdot e^{-2\pi i[hx + ky + lz - \phi(hkl)]}, \quad (2.2)$$

where  $\rho(x, y, z)$  is the map coordinate,  $V$  is the volume of the unit cell,  $h, k, l$  represent the Miller indices,  $|F(hkl)|$  structure factor, and  $\phi(hkl)$  gives the phases; however,

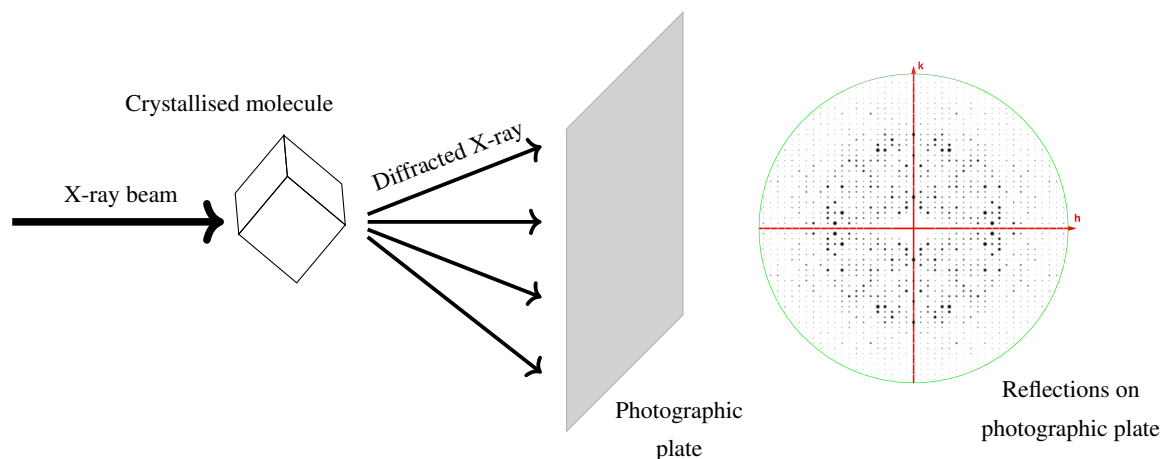


Figure 2.5: X-ray crystallography experiment showing a crystallised molecule and an X-ray passing through the crystal. The diffracted X-ray reaches a photographic plate, and the resulting image is used to calculate an electron-density map that is then used to build a protein model.

these phases cannot be obtained from the X-ray crystallography experiment—an issue known as the *phase problem*. [41].

### 2.1.3.1 Asymmetric unit

The molecule is repeated over the crystal space and solving the structure of one molecule leads to determining the whole crystal structure. The crystal is divided into small parts called *unit cells*; the smallest volume that can be repeated to make the entire crystal [42]. An asymmetric unit is a part of the cell unit that has the identical parts of one molecule or more with no relations in symmetry between them [43]. After solving the molecule structure and finding the coordinates of the atoms from the asymmetric unit, symmetry operations are used to generate the other units' cell contents, which leads to predicting the whole crystal structure. Figure 2.6 shows a crystal and a unit cell represented as mini cubes, as well as an asymmetric unit.

### 2.1.3.2 Miller indices

Miller indices are a group of three numbers used to represent a plane in the crystal:  $h, k, l$ . The crystal is divided into imaginary planes, and those planes are identified by three points recorded on the photographic plate [43]. For example, given a building block with six faces, the position of the top face in Miller indices is  $(0, 0, 1)$ ; however, the last number in the position is 1 because the plate is located on the  $z$  axis and at 0 on

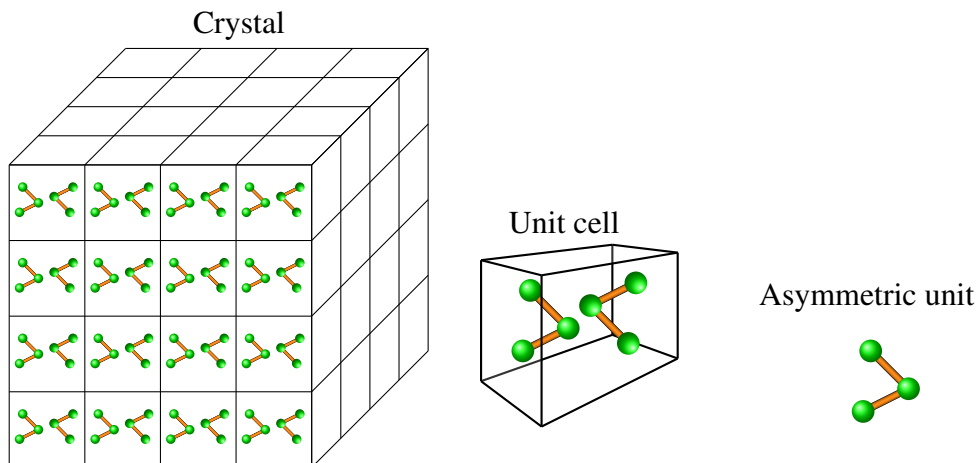


Figure 2.6: A crystal can be represented as mini building blocks where one building block represents a unit cell that may contain multiple copies of the molecule. The asymmetric unit has the identical parts of the molecule.

the  $x$  and  $y$  axis. Figure 2.7 shows different planes and their Miller indices. The Miller indices of a reflection depend on the plane which the X-ray diffracts from. Figure 2.8 shows Miller indices for the reflection on a photographic plate.

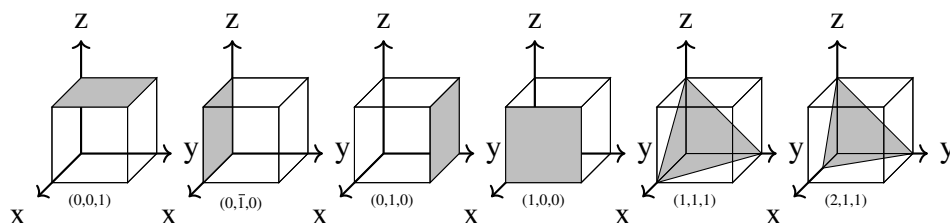


Figure 2.7: Each plane in the cube can be represented by three points,  $h$ ,  $k$ ,  $l$ . In the leftmost cube, the top face of the cube position is  $(0,0,1)$ , with the first two indices corresponding to  $x$  and  $y$ , and 1 being the value of  $z$ . When the points of the plane are located in the middle of an axis such as in the last cube, the value of the axis is divided by 2 or is dependent on the exact value of the axis.

### 2.1.3.3 The phase problem

Phases are required to calculate the electron-density map; however, the phases cannot be determined during the X-ray crystallography experiment. Current equipment is limited in its ability to determine the intensities of the rays from the photographic plate for use in the electron-density map equation. Figure 2.9 shows the missing information, i.e., the phase angle of the diffraction in the crystal. [4, 44]. Two methods are used to solve the phase problem: experimental phasing is when the phases are determined from

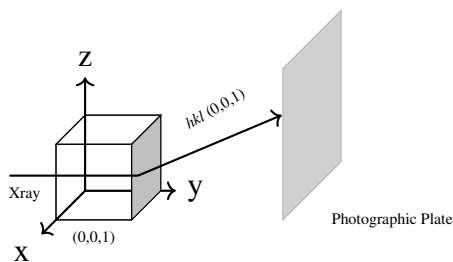


Figure 2.8: The diffraction of the X-ray and Miller indices. The reflection of the X-ray is recorded by Miller indices, which represent a plane in the crystal.

the observed data using features of special atoms, such as those with a large number of electrons, e.g. Dauter and Dauter [45], and molecular replacement (MR) obtains initial phases from a known protein structure that is similar to the protein structure that we want to build, e.g. Evans and McCoy [6].

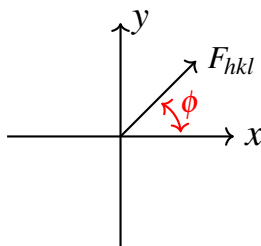


Figure 2.9: Phases  $\phi$  of  $F_{hkl}$  are the missing information from X-ray crystallography experiment.

#### 2.1.3.4 Representation of electron-density maps

The 3D electron-density map visualisation is represented as a mesh in visualisation tools such as, Coot [46] and CCP4MG [47]. This supports the assessment and identification of possible errors in the density map or even in fitting the protein model into the electron-density map (Figure 2.10).

#### 2.1.3.5 Resolution of electron-density maps

The X-ray diffraction spots on the detector correspond to the molecular structure in the crystal, and these diffraction spots are affected by several factors, including the complexity of the molecular structure. The effects of these factors can be negative, leading to poor diffraction and low-quality density maps. The details of the analysed

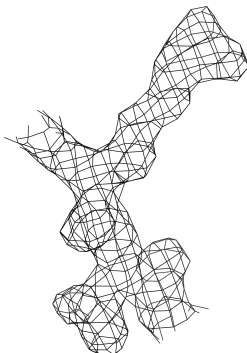


Figure 2.10: A part of electron-density map. The figure was produced by CCP4MG [47] for a part of PDB id 1o6a.

protein are easy to identify in high-resolution electron-density maps, and much harder in the case of low resolution such as in 3 Å and lower [48]. Figure 2.11 shows the increase in the level of difficulty of finding the atoms' positions in a protein model from an easier case when the resolution is high, 0.6 Å or higher, to a challenging case with low resolution, 4Å.

### 2.1.3.6 Electron-density map modification

As described in Section 2.1.3.3, the phases are required to construct the electron-density maps, however, the phase set obtained from the methods of initial phases calculations such as multiple anomalous dispersion (MAD), may not sufficient for protein model building due to the introduction of heavy atoms in the crystal which affect on its order. Therefore, the introduction of heavy atoms might affect the quality of phases and lead to an uninterpretable electron-density map. However, the phases can be improved by knowing the chemical properties about the protein structure that we want to solve and the information obtained from the initial phases. This method is known as Density Modification (DM). It is used to generate improved phases and combine them with initial phases, reducing the noise in the electron-density map and making it more interpretable. The following three approaches are used in DM [49]:

- (i) Solvent flattening is based on determining the regions of solvent (e.g. waters) in the protein structure and creating a mask (e.g., 0 for the solvent parts and 1 for the protein parts) to eliminate the noise from the electron-density map. The

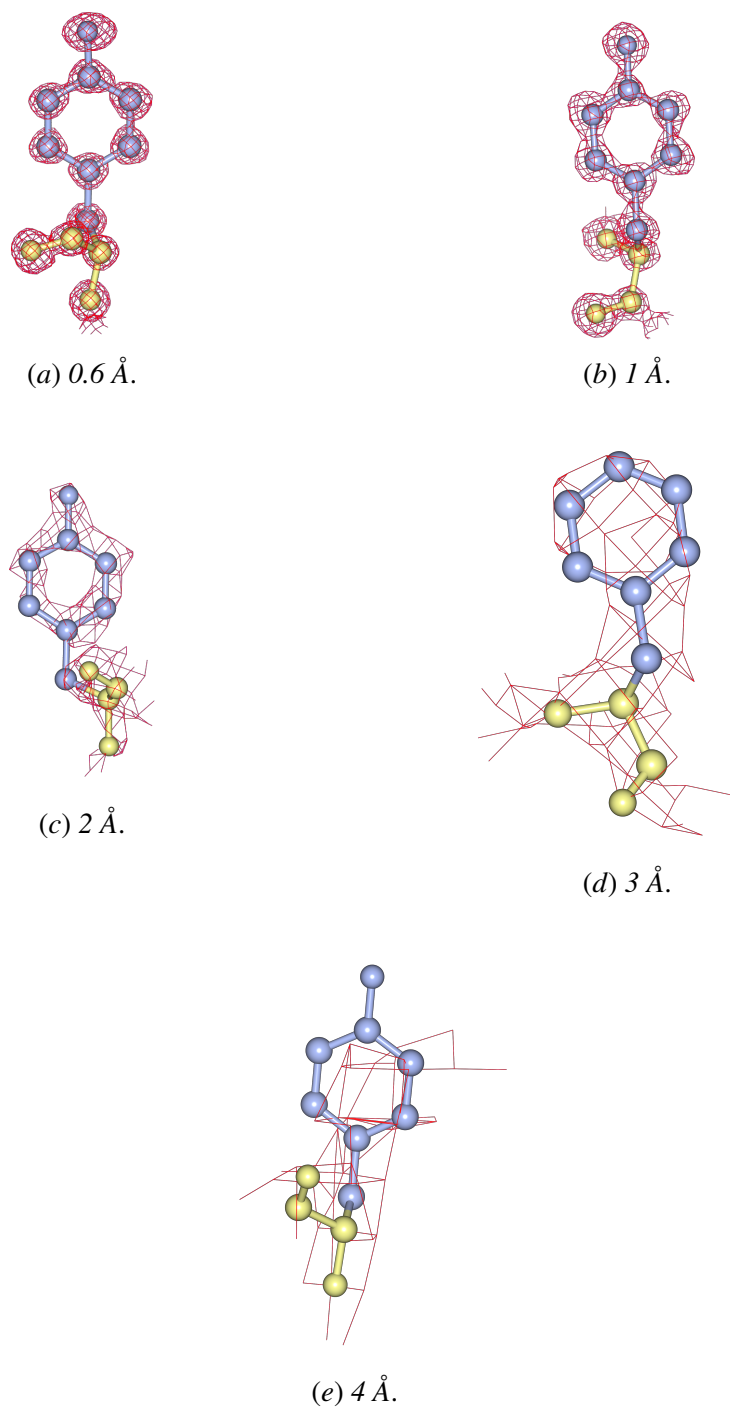


Figure 2.11: Interpretation of electron-density map becomes difficult as its resolution gets worse. (a) A very high resolution electron-density map for which fitting the atoms in their density is simple due to lack of overlaps between the atoms' densities. (b) and (c) Reduced resolutions where the overlaps between the atoms' densities leads to difficulty in placing the atoms in their densities. (d) and (e) Very low resolution maps; such density maps might be misleading for building the correct protein model as they are unhelpful in interpretation of the electron-density.

phases can be calculated *modified phases* and combined with experimental data, and an improvement should be obtained compared to the initial phases.

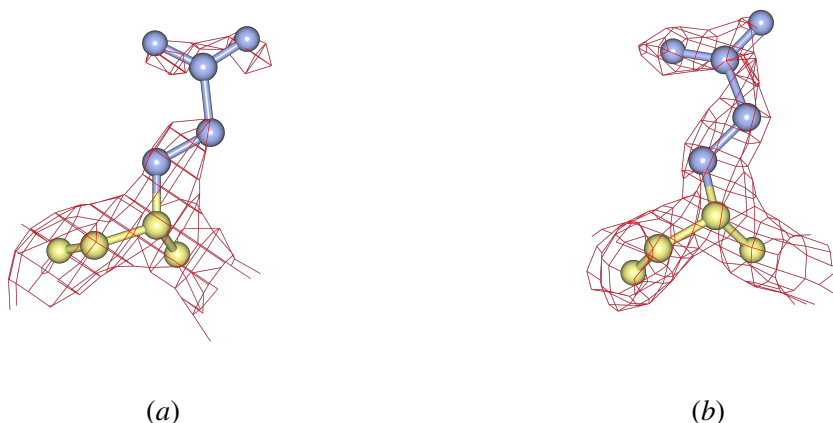


Figure 2.12: Density modification: (a) An electron-density map calculated using initial phases (before density modification). (b) The electron-density map after density modification using Parrot [50].

- (ii) Histogram matching is taken from image processing when two images at grayscale, for example, one image is darker than the other and modify the darker image's histogram to match the brighter image's histogram. Therefore, the darker image might be improved. In electron-density map modification, the histogram of the initial density map is modified to match the histogram of the ideal density map.
- (iii) Noncrystallographic symmetry (NCS) exploits the similarities of electron density between the different regions in the density map for improving the regions with low quality. NCS occurs when the asymmetric unit has multiple copies of a molecular structure with no crystallographic symmetry between them. Discovering NCS in the density map is possible through methods such as the use of heavy atoms (hancs), model building, and molecular replacement (mrncs).

Figure 2.12 shows an electron-density map before DM and after. The density of the side chain is significantly improved after DM, allowing the identification of the positions of the atoms. However, not all electron-density maps can be improved to the level that shows the details of the electron-density map with sufficient clarity for protein model building.

## 2.2 Machine learning

Machine learning has been used to accelerate the protein model building, e.g. using ML to correct the protein model or trace the backbone and, more recently, to build the protein model from its sequence [25, 51]. However, machine learning is an area of artificial intelligence that focuses on the development of methods to learn from past information collected in digital format in order to make predictions from new data [52]. There are many of these learning methods, primarily grouped into two classes: supervised algorithms when the training data is labelled, usually by a domain expert, and unsupervised algorithms when the training data is unlabeled and the learning algorithms discover the relationships between the data set's instances [53].

### 2.2.1 Decision trees

A decision tree is a predictor that takes instance  $x$ , which can be a vector of features, and gives a label  $y$  [54]. The features are the characteristics of the past information that is used by machine learning algorithms to learn relationships between the instance  $x$  and the label  $y$ . Creating decision trees starts by finding a root node and then splitting the tree into branches to add leaf nodes. Selecting the root node is based on splitting measures, such as the Gini Index, which measures uncertainty if a feature is classified incorrectly. The feature with lowest Gini Index is used as root and the process is repeated to split the tree further. Deep decision trees may lead to overfitting when the decision tree performs better on the training data set and worse on the testing data set. However, this problem may be prevented by reducing the number of iterations, and, therefore, the tree size; or by using ensemble methods [54]; however, this may reduce the performance of the decision tree. Figure 2.13 shows a decision tree that we trained to predict the type of an amino acid using its numbers of carbon, oxygen and hydrogen in both the main and side chain as features.

### 2.2.2 Random forests

As described in Section 2.2.1, the over-fitting of decision trees can be reduced by using ensemble methods, which involves training multiple machine learning models to produce more accurate predictions [55]. Random forests were introduced by Breiman



[56] to address the overfitting problem of decision trees through creating multiple decision trees and picking the prediction with the most “votes” from these decision trees. To create a training data set for each decision tree in the random forest, we select a random sample from the training data set to generate a subset of the whole training data set and repeat the process to create multiple subsets (whose sizes may differ). Then, the splitting of the tree is conducted as described in Section 2.2.1[54].

### 2.2.3 Neural networks

A neural network is a type of machine learning inspired by the way in which certain functions are carried out by the human brain [59]. A neural network comprises processing nodes called neurons. Each neuron computes a weighted sum of one or several numerical inputs. The result of this computation is fed into an activation function (Figure 2.14) that computes the neuron output by mapping the weighted sum to a value within a fixed range. Two examples of activation functions are shown in Figure 2.15.

The *training* of a neural network is an optimisation problem in which the neuron weights are adjusted over a sequence of iterations in which a loss function that measures the neural network’s prediction error for a training set of labelled data samples is reduced [60]. Therefore, optimisation methods such as the Adam optimizer [61] are used with neural networks to optimize the parameters of the model.

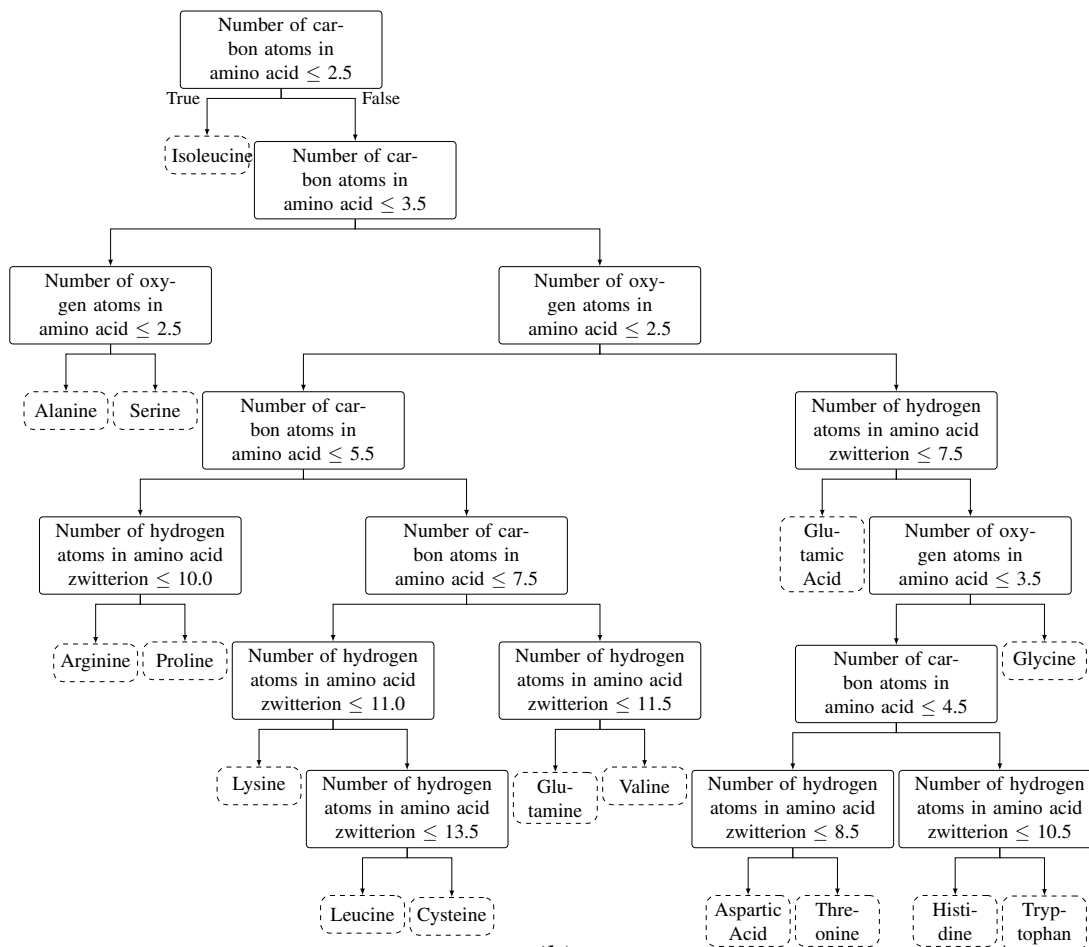
A simple neural network contains three layers; input, hidden and output layer. The layers can have a different number of neurons. In the following sections, we summarise several important types of neural network.

#### 2.2.3.1 Feedforward neural networks

The feedforward model is an essential neural network architecture type where the data are fed from a higher layer to a lower layer with no feedback shared between the layers [62]. Each layer contains a number of neurons and is linked to each neuron in the next layer. Figure 2.16 shows an example of a feedforward neural network with the three layers and different numbers of neurons in each layer.

Amino acid	Number of atoms in amino acid			Amino acid	Number of atoms in amino acid		
	Carbon	Hydrogen	Oxygen		Carbon	Hydrogen	Oxygen
Alanine	3	7	2	Methionine	5	11	2
Cysteine	3	7	2	Asparagine	4	8	3
Aspartic Acid	4	7	4	Proline	5	9	2
Glutamic Acid	5	9	4	Glutamine	5	10	3
Phenylalanine	9	11	2	Arginine	6	14	2
Glycine	2	5	2	Serine	3	7	3
Histidine	6	9	2	Threonine	4	9	3
Isoleucine	6	13	2	Valine	5	11	2
Lysine	6	14	2	Tryptophan	11	12	2
Leucine	6	13	2	Tyrosine	9	11	3

(a)



(b)

Amino acid	Predicted	Amino acid	Predicted	Amino acid	Predicted	Amino acid	Predicted
Alanine	Alanine	Glycine	Glycine	Methionine	Methionine	Serine	Serine
Cysteine	Alanine	Histidine	Histidine	Asparagine	Asparagine	Threonine	Threonine
Aspartic Acid	Aspartic Acid	Isoleucine	Isoleucine	Proline	Proline	Valine	Methionine
Glutamic Acid	Glutamic Acid	Lysine	Arginine	Glutamine	Glutamine	Tryptophan	Tryptophan
Phenylalanine	Phenylalanine	Leucine	Isoleucine	Arginine	Arginine	Tyrosine	Tyrosine

(c)

Figure 2.13: An example of decision tree for predicting the type of an amino acid using its numbers of carbon, oxygen and hydrogen atoms. (a) The training data sets obtained from [57]. (b) The decision tree was trained using scikit-learn [58]. (c) The performance of the decision tree was tested on the same training data sets however, the test data should not be the same as the training data sets for valid machine learning testing. Here, we do not test the decision tree on independent data sets, as this example shows the creation of a decision tree rather than producing a valid machine learning model.

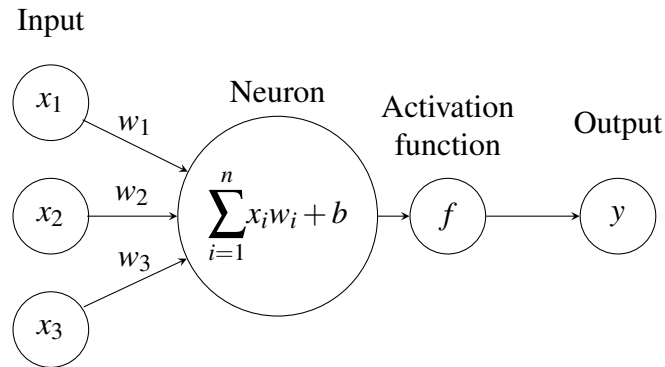


Figure 2.14: Example of simple neural network. Each input neuron  $x$  connected to the next neuron and a weight  $w$  assigned to each connection. The neuron sum the inputs and the weights and add a bias  $b$  to the summation. An activation function decides whether the neuron will be activated or not.

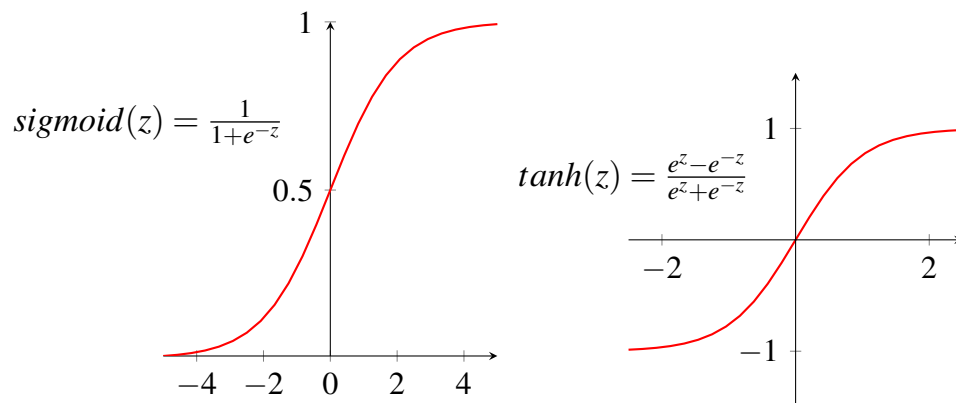


Figure 2.15: Left: Sigmoid shape ranging from 0 to 1. Right: tanh shape ranging from -1 to 1.  $z$  is the output of the neuron.

### 2.2.3.2 Recurrent neural networks

A recurrent neural network (RNN) is an extension of a traditional neural network that is able to process sequential data with different lengths [63]. A feedforward neural network only processes the current input, meaning that it does not remember the previous input because the training data set moves in one direction. Unlike Feedforward neural networks, an RNN considers the current input with the previous one when adjusting the RNN network weight. Classic RNNs have a long-term dependencies learning problem. This occurs when the input is a long sequence of data with dependencies, and the classic RNNs cannot remember the status of the data that was received earlier [64, 65, 66, 67]. In 1995, long short-term memory (LSTM) was introduced by [68] to address this problem. A LSTM layer has a memory that remembers the data shown earlier. In 2015, the attention mechanism was introduced in deep learning which a

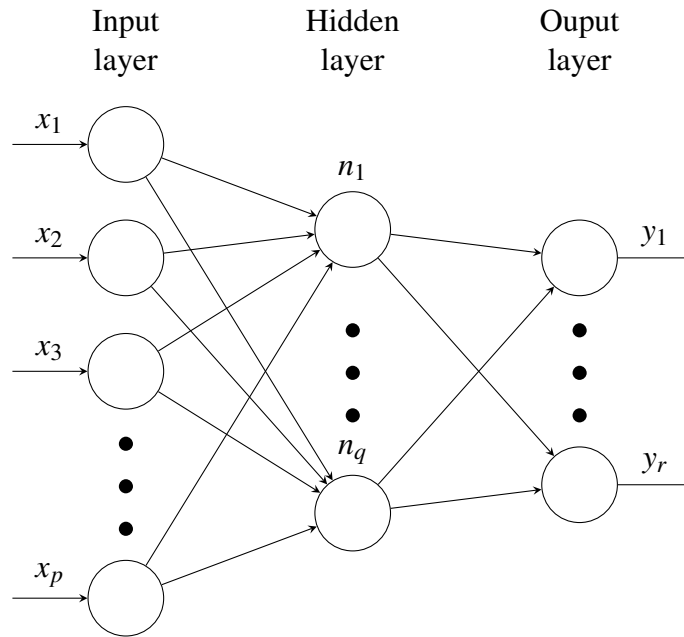


Figure 2.16: An example of feedforward neural network.

neural network gives more importance to some data instances in a sequence, unlike LSTM, which gave the same importance to the data instances [69].

# Comparison of automated crystallographic model-building pipelines

In this chapter, we present a comparison between automated crystallographic model-building pipelines. We ran the pipelines on large data sets and evaluated them based on the structure completeness, R-work/R-free of the protein models they generated and the correlation between generated models and final deposited models. The aim of the work in this chapter was to determine a performance baseline for use in the evaluation of the methods proposed in later chapters of the thesis.

## 3.1 Abstract

A comparison between four protein-building pipelines (ARP/wARP, Buccaneer, PHENIX AutoBuild and SHELXE) was performed using data sets from 202 experimentally phased cases, both with the data as observed and truncated to simulate lower resolutions. All pipelines were run using default parameters. Additionally, an ARP/wARP run was completed using models from Buccaneer. All pipelines achieved nearly complete protein structures and low R-work/R-free at resolutions between 1.2 Å and 1.9 Å, with PHENIX Autobuild and ARP/wARP producing slightly lower R-work. At lower resolutions, Buccaneer leads to significantly more complete models.

## 3.2 Introduction

The automation of protein model building began with the release of ARP/wARP in the late 1990s [13, 14, 15, 17], and has rapidly advanced through the development of additional protein-building pipelines. These pipelines include Buccaneer [18, 19], PHENIX AutoBuild [20], SHELXE [21, 22, 23, 24], and a major new version of ARP/wARP [16]. Judging by the numbers of Web of Science citations across 2017 and 2018, ARP/wARP (286 citations), Buccaneer (304 citations) and PHENIX AutoBuild (217 citations) are all widely used; SHELXE was cited 9548 times within the same time period (with all citation counts being based on the papers listed above).

Complex optimization problems like building protein structures can be tackled using multiple approaches. As such, different protein-building pipelines employ different steps and algorithms, may refine their intermediate structures using difference refinement programs such as REFMAC [70] or phenix.refine [71], and yield different results for the same data. The comparison detailed here sheds light on some of these differences by examining the completeness of protein structures, the R-work/R-free values, and the execution times of ARP/wARP, Buccaneer, PHENIX AutoBuild and SHELXE. Performed for data sets with resolutions ranging from 1.2 Å to 4.0 Å, this comparison provides insights into the strengths and weaknesses of the different pipelines, which may be of use when addressing specific problem data sets, as well as to developers seeking to improve their own algorithms or to build new meta-pipelines which exploit the complementary strengths of the different algorithms.

As scientists are inevitably affected by cognitive biases, including self-serving biases, this study would ideally have been conducted by an independent party, similar to the study of van den Bedem et al. [72]. However, independent researchers often lack the motivation to perform detailed tool comparisons. For us, further development of the Buccaneer methods required a better understanding of their limitations, and thus, we conducted our own comparison. We acknowledge that its results may have been impacted by biases in our study, and we make those sources of bias that we are aware of explicit in the discussion.

## 3.3 Pipelines and methods

### 3.3.1 ARP/wARP

ARP/wARP was the first fully-automated pipeline for building protein models from electron-density maps. Initially limited to high resolutions of better than 2.3 Å [14], ARP/wARP was subsequently extended to 2.7 Å or 2.8 Å [16]. More recent versions have further enlarged the useful range of resolutions [73]. ARP/wARP is integrated with CCP4, and therefore can be used from the CCP4 GUIs. Additionally, ARP/wARP has a web service interface for remote running, which enables access to resources beyond those available on the users' local machines.

The ARP/wARP approach starts by placing free atoms in the electron-density map. Free atoms are atoms that do not have a chemical identity, but are likely to develop one during the model building and refinement. The approach then traces the main protein chain via an algorithm [74] that uses modified depth-first search techniques. Next, ARP/wARP uses a rotamer library and a downhill simplex algorithm to fit the side chains into the map density. Finally, the missing parts of the protein model are completed by matching  $C^\alpha$  segments from known models, and choosing those that best fit the density of the working model. Following the building stage, the model is refined with REFMAC, and the calculated map is used for further ARP/wARP building cycles.

### 3.3.2 Buccaneer

Buccaneer is a command-line protein model building tool developed by Cowtan [18]. Its subsequent integration with the Collaborative Computational Project Number 4's CCP4 software suite [75] provided Buccaneer with a graphical user interface through the CCP4i [76] and CCP4i2 [77] GUIs.

The Buccaneer algorithm is built around a likelihood target function for the identification of likely  $C^\alpha$  positions. This function is used to find a small set of 'seed' residues, and then to grow these seeds into chain fragments using Ramachandran restraints. Overlapping chain fragments are merged, and docked into the sequence on the basis of a further application of the likelihood target function to the identification

of the side chain type [18, 19]. Model building is iterated with refinement in REFMAC [78].

### 3.3.3 PHENIX AutoBuild

PHENIX AutoBuild is a part of the PHENIX software suite for the automated modelling of molecular structures. Using a graphical user interface (GUI) based on the main PHENIX GUI, AutoBuild facilitates the interactive specification of protein-building parameters, with default values automatically provided for most parameters. Additionally, command-line access is available to enable the integration of AutoBuild with other tools.

PHENIX AutoBuild accepts several types of input—experimental phases, an existing model, and a model whose sequence differs by less than 5% from that of the target model—and performs different procedures for each input type. The steps of its fully automated pipeline include density modification, model building and refinement [79, 80, 81]. These AutoBuild steps are not executed sequentially, as the density modification is repeated after refinement, to exploit information from the built model.

Early in the structure determination procedure, AutoBuild scores models using a metric based on their number of residues built, number of residues that match the protein sequence, and number of chains [20]. Later, when their R-work drops below a pre-set value, the models are scored mainly using R-work. The refinement of the built structures is performed using *phenix.refine* [71], a refinement tool from the PHENIX suite.

### 3.3.4 SHELXE

SHELXE is a program for main chain tracing and density modification from experimental phases and molecular replacement [22, 23]. Backbone tracing begins by finding seven residue  $\alpha$ -helices and extending them in both directions whenever possible. The latest version of SHELXE was extended to find up to 14 residues. [24]. Traced chains are then cut at their closest points of contact, and the N-termini and C-termini are joined together. Finally, new estimated phases are calculated from traced residues and combined with the initial phases for use in the next cycle of density modification and tracing [22].



SHELXE scores a built structure using a correlation coefficient (CC) calculated from structure factors from the trace against native data. A CC above 25% for resolution 2.5 Å indicates that SHELXE may have found a correct solution [24].

### 3.4 Data sets

We used 202 real data sets [72] with resolutions between 1.2 Å and 3.2 Å (Figure 3.1), as well as synthetic data sets obtained through simulating each of the original data sets at resolutions of 3.2 Å, 3.4 Å, 3.6 Å, 3.8 Å and 4.0 Å. The 202 data sets used are a subset of the 770 data sets from van den Bedem et al. [72]. A total of 230 structures were available to the authors, of which 229 had one or more data sets from experimental phasing. A single data set, with the highest RMSD of local map RMSD, was chosen for each structure. There is no guarantee that the chosen data set is the same one used for the final deposited structure, but in order to check this, the deposited coordinates were refined against the chosen data set using REFMAC v.5.8.0158 in CCP4 v.7.0.045 [78]. Eleven structures failed due to large differences between cell definitions in the reflection file and deposited model and one structure failed due to a serine residue being labelled as UNK. A further 15 structures were removed as they had very high R-work/R-free after refinement. Five of the deposited structures (2a9v, 2ash, 2awa, 2o5r and 2pnk) have their structural determination method listed as a combination of MAD and molecular replacement and one (2fcl) has only molecular replacement. In these cases the deposited structure may contain some model bias from the original author's search model. This simulation involved inflating the B-factors of the structure factor amplitudes and removing the reflections with resolutions higher than the target resolution. Inflation of B-factors was carried out by first downloading a list of all structures in the PDB, each with a resolution and average B-factor. A linear fit was then performed, which gave a gradient of 32.8Å used to inflate the B-factors by the difference in resolution. This modification resulted in the reduction of the electron-density map resolution to that of the simulated resolution. This process produced 1009 synthetic data sets—five synthetic data sets at the lower resolutions mentioned above for each original data set, except for a single data set in which the original resolution was already 3.2Å. This gave us 1211 data sets in total. The 52 data sets that had previously

been used in the development of Buccaneer<sup>1</sup> were excluded, along with the synthetic data sets obtained from them.

The density of both the original and synthetic data sets was then modified using Parrot [50] for three density modification types: heavy-atom NCS (HA-NCS) determined using S or Se atom positions from the deposited model, molecular replacement NCS (MR-NCS) determined using all atoms of the deposited model, and no NCS (NO-NCS). The three groups of 1211 data sets (i.e. 3633 data sets in total) created in this way were used in the comparison. The PDB codes used in the comparison (provided as supplementary material in Appendix A.6)

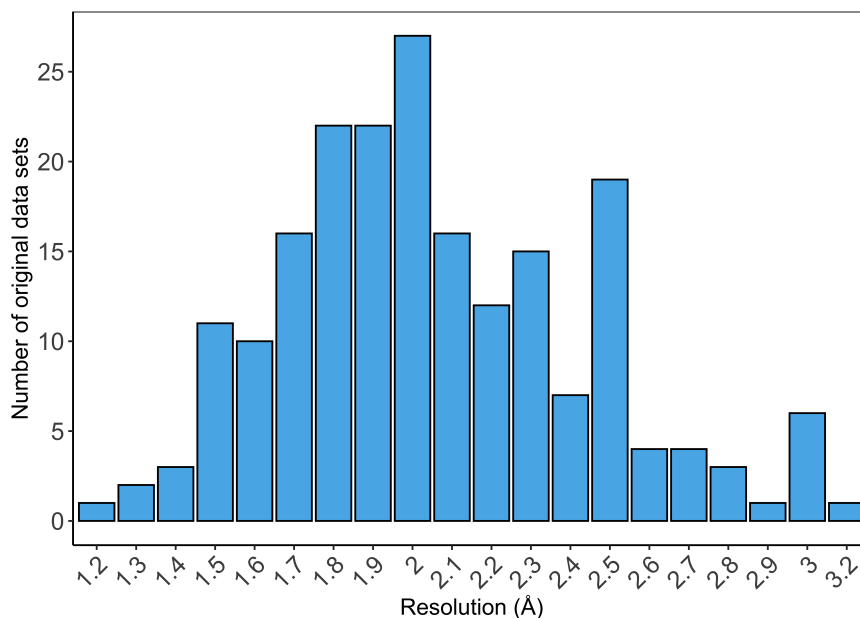


Figure 3.1: Resolutions of the 202 original data sets.

### 3.5 Method of the comparison

A comparison was conducted between the following versions of the four protein-building pipelines described in Sections 3.3.1–3.3.4: PHENIX Autobuild version 1.14, Buccaneer in CCP4i, ARP/wARP 8 and SHELXE version 2019/1. All binary files were obtained from CCP4 7.0.066, and run with the default parameters set by the developers of each pipeline ARP/wARP was run without the R-free flag, in line with the tool’s documentation, and automatically includes a secondary structure building step in cases

<sup>1</sup>These 52 data sets were analysed for a secondary study in which we assessed the efficiency of choosing training data sets for pipeline development (Appendix B.1 and B.2)

where resolution is worse than 2.7Å. PHENIX Autobuild by default builds three models at each step leading to improved results at the cost of computing time. Additionally, the comparison considered several pipeline variants with non-default parameters:

- ARP/wARP with the R-free flag set, and using as initial models the models built by Buccaneer in CCP4i, as one known Buccaneer limitation is its use of fewer model finalization techniques;
- PHENIX AutoBuild with density-modified phases (using Parrot [50]);
- SHELXE with density-modified phases (using Parrot [50]);
- SHELXE (with and without density-modified phases) variants have set -t flag to 20 as higher value is recommended in the tool's documentation;

Table 3.1 shows the short names used for these pipeline variants in the rest of the paper.

Each execution of a pipeline received two inputs: a reflection data file comprising the result of an experimental phasing calculation; and the sequence file of the relevant protein. SHELXE did not receive the sequence file because it is not required. The model building task was then submitted as a job to a 173-node high-performance cluster with 7024 Intel Xeon Gold/Platinum cores, a total memory of 42TB. Each job involved building one protein model, and was stopped if it did not complete within 48 hours. There was no resource sharing between jobs.

Following model building, a 'zero cycle' REFMAC run was used to calculate R-work/R-free (which measure the fit of the protein structure against the observed data, with R-free using only observations which are not used in the refinement calculation – typically 5% of the data [82]), to avoid the confounding effects of different scaling and solvent parameterizations in different refinement programs. REFMAC was run with default parameters. The quality of the starting phases was assessed using on the weighted F-map correlation between the initial map and the phases from the refined deposited model. A *structure completeness* measure was obtained for the final model, by calculating the percentage of residues in the processed deposited model from the Protein Data Bank (PDB) whose  $C\alpha$  atoms have the same residue type as, and coordinates within 1.0 Å of, the corresponding residue in the built model. SHELXE completeness was calculated from only  $C\alpha$  in correct positions within 1.0 Å because

SHELXE only builds the main chain. The correlation between generated and final deposited models was obtained by calculating the F-map correlation using a map from a built model and a map from a final deposited model (we will refer to this measure as *structure correlation* in the rest of the thesis).

A tool was developed to automate the execution of the pipelines and the analysis of their results. To ensure the reproducibility of the study, the execution of all pipeline variants was repeated for a sample of 30 structures. The results (provided as supplementary material in Appendix A) did not vary significantly when the pipelines were rerun with the same inputs. Additionally, a series of tests searching for errors that might have occurred during the running or analysing stages were performed; for example, the running parameters from log files were verified for possible errors in the parameter settings.

Four measures were used to compare the protein models built by different pipelines: structure completeness, R-work/R-free, structure correlation and pipeline execution time. R-work/R-free values were rounded to two decimal places, and completeness was rounded to the nearest whole number.

For both completeness and R-work/R-free, and for each pair of pipelines, we report the percentage of data sets for which one pipeline yields better models than the other; and the percentage of data sets for which one pipeline yields models which are at least 5% better than the models produced by the other pipeline. (Cases where results are equivalent or better by between 1% and 4% are reported in the appendix). The results obtained for the real data sets used in the comparison and for the data sets truncated to simulate lower resolutions are reported separately. For execution time, we report the mean pipeline execution times partitioned into classes based on their structure sizes.

## **3.6 Results**

### **3.6.1 Overview**

The results described here were obtained by comparing the protein structures successfully built by each of the pipeline variants from Table 3.1. For the first 4 pipeline variants from the table, we used all 3633 data sets obtained as described in the previ-

Table 3.1: Pipeline variants used in the comparison.

Short name	Long name
ARP	ARP/wARP.
ARP(B 5I)	ARP/wARP after Buccaneer in CCP4i using the default five iterations.
i1(5I)	Buccaneer in CCP4i using 5 iterations (as set by the pipeline developers).
PHENIX	PHENIX AutoBuild fed by density-unmodified phases.
SHELXE	SHELXE fed by density-unmodified phases.
PHENIX/Parrot	PHENIX AutoBuild runs after Parrot (density-modified phases).
SHELXE/Parrot	SHELXE runs after Parrot (density-modified phases).

ous section. For the PHENIX AutoBuild and SHELXE after Parrot no prior density modification was run and the results were compared to the NO-NCS results from the other pipelines. SHELXE variants were not run on synthetic data sets because this is not recommended, and therefore SHELXE is omitted from synthetic data sets comparison.

All pipeline variants successfully completed the analysis of over 99% of both the original and synthetic data sets. The remaining runs did not complete within 48 hours (a time limit that we set in our experiments), failed due to insufficient memory, or crashed. In all these cases, the pipeline variant was rerun with its memory quota and time limit increased until it either succeeded or a limit of 20GB of allocated memory and 48 hours were reached. As shown in Tables 3.2 and 3.3, only very few runs did not complete (even after this memory increase), and most of these produced intermediate protein models that we used in our comparison. The data sets marked ‘Failed’ in the tables were excluded from the comparison (for all pipeline variants). The numbers of different types of ‘complete’ and ‘intermediate’ models used in the comparison are reported at the bottom of each table.

Including non crystallographic averaging improves the starting phases for structures where NCS is present, but it does not significantly affect the conclusions of this work because the completeness is not significantly affected. Given that the differences between NCS and NO-NCS cases are small, the poorer-phased NO-NCS data sets will be considered for the remainder of the comparison.

Using the correct solvent fraction in SHELXE improves its results, but it does not significantly affect the results when compared to other pipeline variants (results of

using the correct solvent fraction are reported in the Appendix A.7). A default fraction solvent, which is 0.45, is used in the comparison.

Table 3.2: Complete and intermediate models produced by the 7 pipeline variants for the original data sets, where ‘(T)’ and ‘(C)’ denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	HA-NCS			MR-NCS			NO-NCS		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
ARP	201	1(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
ARP(B 5I)	202	0(T) 0(C)	0	201	1(T) 0(C)	0	202	0(T) 0(C)	0
i1(5I)	202	0(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
PHENIX/Parrot	198	2(T) 1(C)	1	200	0(T) 1(C)	1	199	1(T) 1(C)	1
SHELXE/Parrot	202	0(T) 0(C)	0	201	1(T) 0(C)	0	200	2(T) 0(C)	0
PHENIX	-	-	-	-	-	-	199	1(T) 0(C)	2
SHELXE	-	-	-	-	-	-	200	2(T) 0(C)	0

Models used in the comparison: 149 HA-NCS, 149 MR-NCS and 148 NO-NCS.

Table 3.3: Complete and intermediate models produced by the 5 pipeline variants for the synthetic-resolution data sets, where ‘(T)’ and ‘(C)’ denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	HA-NCS			MR-NCS			NO-NCS		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
ARP	1008	1(T) 0(C)	0	1007	2(T) 0(C)	0	1008	1(T) 0(C)	0
ARP(B 5I)	1005	4(T) 0(C)	0	1006	3(T) 0(C)	0	1003	6(T) 0(C)	0
i1(5I)	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0
PHENIX/Parrot	1002	7(T) 0(C)	0	1004	5(T) 0(C)	0	1001	8(T) 0(C)	0
PHENIX	-	-	-	-	-	-	1001	7(T) 0(C)	1

Models used in the comparison: 750 HA-NCS, 750 MR-NCS and 750 NO-NCS.

### 3.6.2 Structure completeness

Tables 3.4 and 3.5 report the percentages of models for which each pipeline variant achieved a structure completeness that is higher and at least 5% higher, respectively, than the other pipeline variants. Note that the two figures associated with a pair of pipeline variants in Table 3.4 do not always add up to 100% because some of the models are generated with the same structure completeness (rounded to the next integer) by the two pipeline variants. For example, the structure completeness of 23% of the ARP models was higher than that of the corresponding ARP(B 5I) models, and 45% of the ARP(B 5I) models had higher structure completeness than that of the ARP models; thus, the remainder 32% of the models built by ARP and ARP(B 5I) had the same structure completeness, after rounding.

As shown in the first of these tables, ARP/wARP built 37% of the data sets better than PHENIX Autobuild, while PHENIX Autobuild did better in 48% of the data sets,

Table 3.4: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP	0	23	33	39	37	68	61
ARP(B 5I)	45	0	40	43	43	76	73
i1(5I)	57	45	0	46	49	77	72
PHENIX/Parrot	49	44	45	0	46	80	77
PHENIX	48	39	41	32	0	78	72
SHELXE	26	15	20	16	16	0	34
SHELXE/Parrot	32	22	24	17	22	57	0

0  80

Table 3.5: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP	0	6	15	11	14	45	40
ARP(B 5I)	24	0	20	16	16	53	53
i1(5I)	28	17	0	16	16	56	48
PHENIX/Parrot	28	20	26	0	14	61	55
PHENIX	28	18	23	7	0	57	51
SHELXE	17	7	11	7	7	0	9
SHELXE/Parrot	21	12	17	5	10	32	0

0  61

which means that 15% of the data sets are equal in their completeness. Buccaneer in CCP4i built more than half of the data sets with higher completeness compared to ARP/wARP. The default 5 cycle Buccaneer runs typically produce less complete models than PHENIX AutoBuild.

Table 3.5 shows the number of cases where one pipeline variant achieved 5% or higher structural completeness than another. By this measure for every pipeline variant there are at least 3% of cases where that pipeline produces a significantly more complete model than another pipeline, however show a similar general pattern to the previous comparison.

Running ARP/wARP after Buccaneer can impact the results. Comparing ARP/wARP after Buccaneer in CCP4i (5 iterations) with ARP/wARP alone showed a 5% improve-

ment in completeness in a quarter of cases, with only a few cases of a comparable decrease in completeness. Using PHENIX AutoBuild after Parrot showed a small benefits of the additional density-modification step; 14% of the data sets were built better, compared with 7% worse.

The comparison of SHELXE with the other pipeline variants shows that over half of the data sets are typically built better by other pipeline variants even when the 5% improvement comparison level is considered. SHELXE built 16% of the data sets better than PHENIX Autobuild, but this number decreased to 7% for the 5% improvement comparison level. SHELXE after Parrot showed some improvements when compared to the other pipeline variants; however, for the 5% improvement comparison level, these variants built over 40% of the data sets better than SHELXE after Parrot.

Figure 3.2 shows the mean structure completeness for different ranges of data set resolutions, across both the original and the synthetic data sets. Expectedly, the pipeline variants achieved the best results at 1.2Å-1.9Å, and the completeness of the models was significantly poorer at 4.0 Å. ARP/wARP dropped rapidly at 3.2 Å (synthetic data sets) and decreased to nearly zero completeness at 4.0 Å. In contrast, for Buccaneer in CCP4i, completeness degrades only slowly as resolution drops below 3.1 Å. PHENIX Autobuild produces the most complete models when using the original data resolution; however, its completeness falls between those of Buccaneer and ARP/wARP for the resolution-truncated data sets. The pipelines were affected by F-map correlation, with lower completeness at an F-map correlation of 0.53 or lower (Figure 3.4).

Figure 3.3 shows the mean number of residues which were built incorrectly, grouped into bins based on the data set resolutions. Achieving high structure completeness leads to the generation of a large number of incorrect residues. For example, Buccaneer in CCP4i built more residues incorrectly than other pipeline variants, e.g. a fraction of 0.50 of the residues were incorrect at 4.0 Å, while PHENIX Autobuild only reached a fraction of 0.20 incorrect residues at the same resolution. ARP/wARP and PHENIX Autobuild built nearly no incorrect residues between 1.2Å-1.9Å.



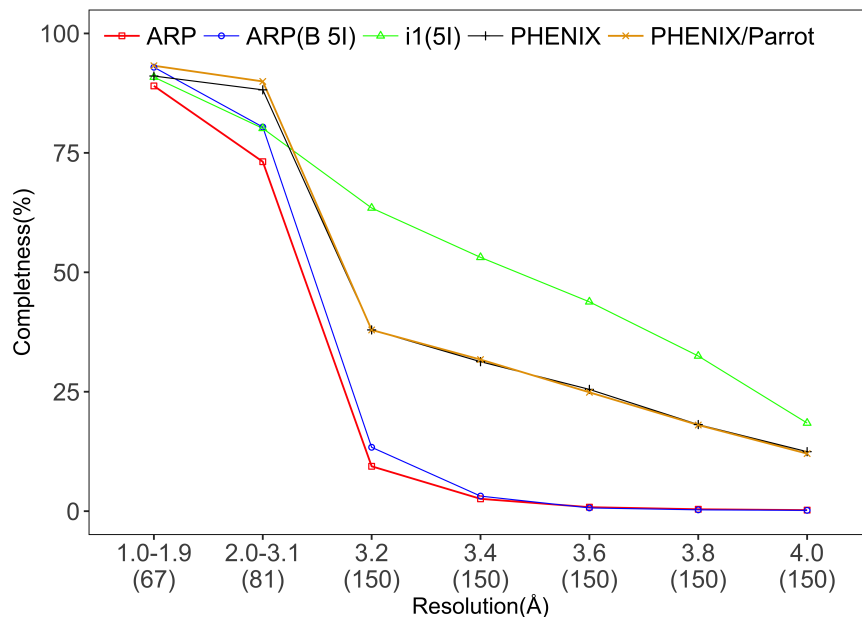


Figure 3.2: Mean completeness for the protein models built for all NO-NCS data sets. The data sets are grouped into bins based on their resolution, with the number of data sets in each bin shown in brackets under the graph.

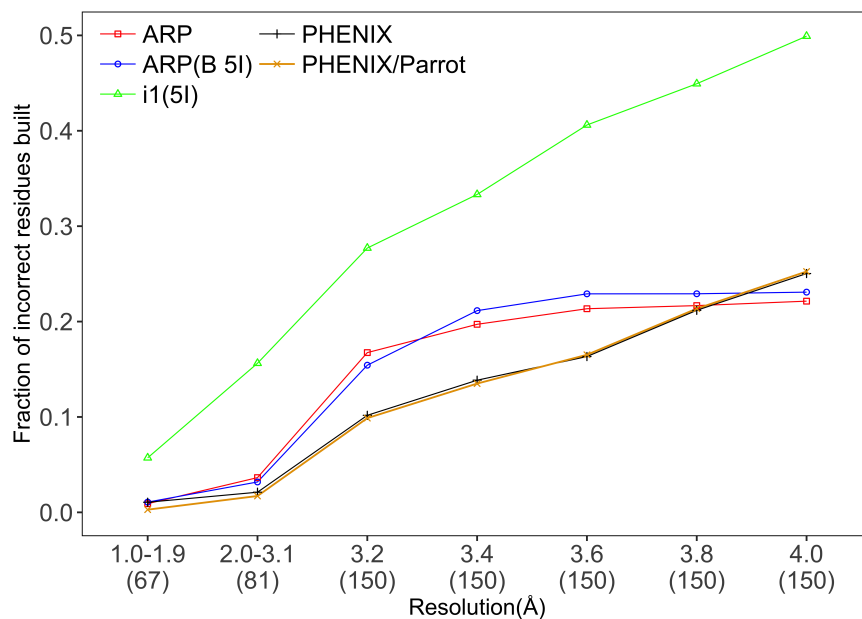


Figure 3.3: Mean residues incorrectly built for the protein models built for all NO-NCS data sets. The data sets are grouped into bins based on their resolution, with the number of data sets in each bin shown in brackets under the graph. The number of residues incorrectly built was normalized by dividing on the number of residues in the deposited model.

### 3.6.3 R-work and R-free

Tables 3.6 and 3.7 show the R-work/R-free results for the pipeline variants at the two levels of comparison (i.e. better and at least 5% better). If R-free was not used, no

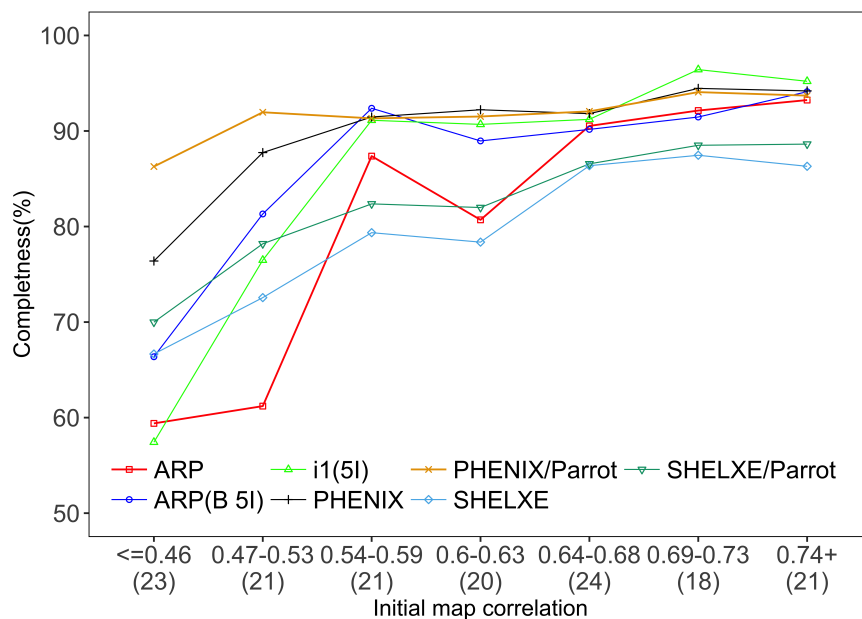


Figure 3.4: Mean completeness for the models built for the original NO-NCS data sets, grouped into bins based on their initial map correlation (F-map correlation); the number of data sets in each bin is reported in brackets under the graph.

results are reported. ARP/wARP and PHENIX AutoBuild obtained results which better explain the X-ray observations than Buccaneer. Buccaneer in CCP4i built less than 10% of the data sets with lower R-work/R-free compared to PHENIX AutoBuild, which built 93% models with lower R-work/R-free than the Buccaneer pipeline. The performance of ARP/wARP and SHELXE can only be compared with the others pipelines in terms of R-work due to not using of R-free, and the results of ARP/wARP were closer to those achieved by PHENIX AutoBuild than to Buccaneer. ARP/wARP built 94% of the models with lower R-work, while Buccaneer only built 5% of the models lower in R-work (Table 3.6). When considering only cases where R-work or R-free change by more than 5% (Table 3.7), there are comparatively few differences between ARP/wARP and PHENIX autobuild, but both outperform the Buccaneer pipeline in a significant proportion of cases. All pipeline variants built at least 97% of the models with lower R-work/R-free compared to SHELXE variants, which built 3% of the models with lower R-work in the best scenario. These results remain almost the same when the 5% improvement comparison level is considered. Using SHELXE after Parrot improved R-work, but it did not significantly improve the results when compared to other pipeline variants.

Figures 3.5 and 3.6 show the R-work and R-free obtained for different resolution

Table 3.6: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP $R$ -work	0	18	94	34	37	100	100
ARP $R$ -free	-	-	-	-	-	-	-
ARP(B 5I) $R$ -work	47	0	99	47	47	100	100
ARP(B 5I) $R$ -free	-	0	76	13	16	-	-
i1(5I) $R$ -work	5	0	0	3	3	97	97
i1(5I) $R$ -free	-	16	0	3	5	-	-
PHENIX/Parrot $R$ -work	47	30	95	0	27	99	99
PHENIX/Parrot $R$ -free	-	74	93	0	31	-	-
PHENIX $R$ -work	43	30	93	22	0	99	99
PHENIX $R$ -free	-	75	93	31	0	-	-
SHELXE $R$ -work	0	0	3	1	1	0	19
SHELXE $R$ -free	-	-	-	-	-	-	-
SHELXE/Parrot $R$ -work	0	0	3	1	1	42	0
SHELXE/Parrot $R$ -free	-	-	-	-	-	-	-



0  100

Table 3.7: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP $R$ -work	0	3	52	5	7	100	100
ARP $R$ -free	-	-	-	-	-	-	-
ARP(B 5I) $R$ -work	5	0	62	6	7	100	100
ARP(B 5I) $R$ -free	-	0	28	1	1	-	-
i1(5I) $R$ -work	0	0	0	0	1	95	94
i1(5I) $R$ -free	-	1	0	0	1	-	-
PHENIX/Parrot $R$ -work	5	3	54	0	2	99	99
PHENIX/Parrot $R$ -free	-	17	57	0	2	-	-
PHENIX $R$ -work	4	2	55	1	0	99	98
PHENIX $R$ -free	-	16	57	1	0	-	-
SHELXE $R$ -work	0	0	1	1	1	0	0
SHELXE $R$ -free	-	-	-	-	-	-	-
SHELXE/Parrot $R$ -work	0	0	1	0	1	1	0
SHELXE/Parrot $R$ -free	-	-	-	-	-	-	-

0  100

ranges. As shown in the tables, PHENIX AutoBuild achieved the best values at 1.2 Å–1.9 Å with the results degrading significantly over at 3.2 Å. The results of Bucca-  
neer degrade more gradually to 4.0 Å. R-free increased in the same manner as R-work.

ARP/wARP produces very good R-work at all resolutions, although the authors caution that overfitting is a problem in the dummy atom model, however, overfitting is likely to happen with other pipelines. Nonetheless, R-free (for the hybrid Buccaneer+ARP/wARP runs, where it is available) is also better than for the other pipelines at lower resolutions, in contrast to the completeness results. This suggests that the dummy atom model has significant predictive power in explaining the X-ray observations, even when it cannot be interpreted in terms of sequenced protein chain.

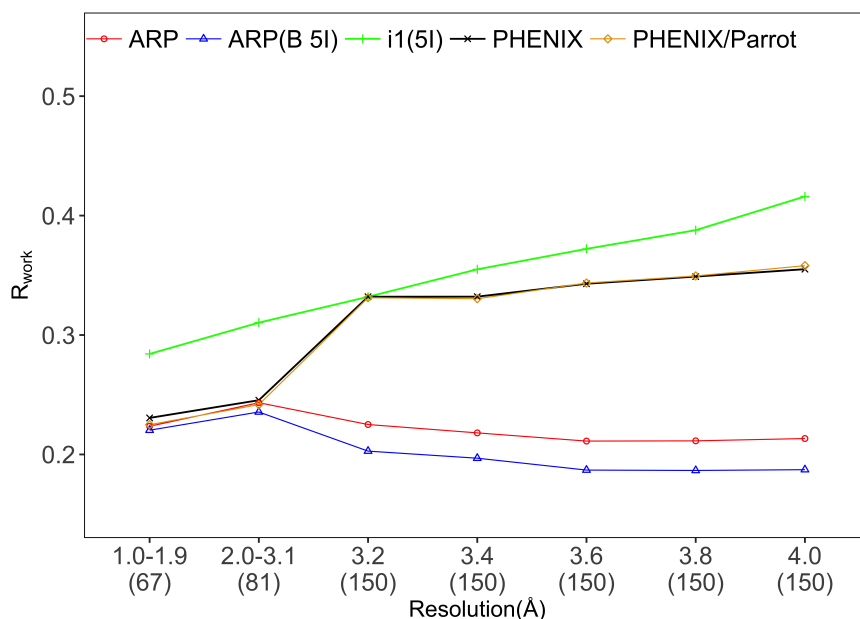


Figure 3.5: Mean protein model R-work for the NO-NCS data sets partitioned into classes based on their resolution. The number of data sets in each class is indicated in brackets under the graph.

### 3.6.4 Structure correlation

Figure 3.7 shows mean correlation between built protein model and final deposited protein model for NO-NCS data sets calculated as described in Section 3.5. At resolution better than 3.2 Å, both PHENIX AutoBuild with and without Parrot showed F-map correlation higher than 0.9, however, PHENIX AutoBuild variants achieved close F-map correlation to Buccaneer at worse resolutions, but they did not fell below 0.8. Structure correlation of the protein structures built by ARP/wARP showed a slightly higher F-map correlation than those built by Buccaneer at resolution better than 2.0 Å and the F-map correlation dropped below 0.6 at resolution worse than 3.1 Å; ARP/wARP on

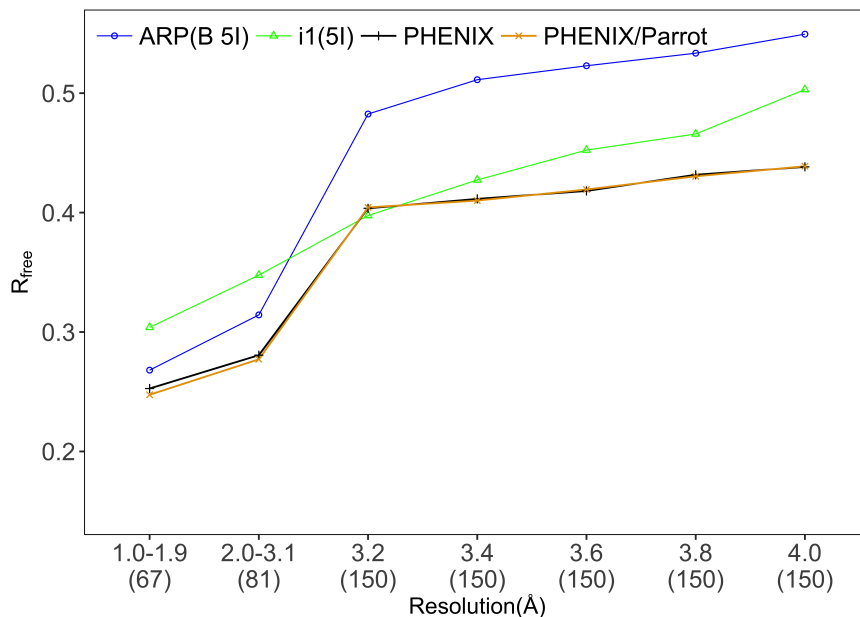


Figure 3.6: Mean protein model R-free for the NO-NCS data sets partitioned into classes based on their resolution. The number of data sets in each class is indicated in brackets under the graph.

its own is better at resolution worse than 3.2 Å compared to running ARP/wARP after Buccaneer.

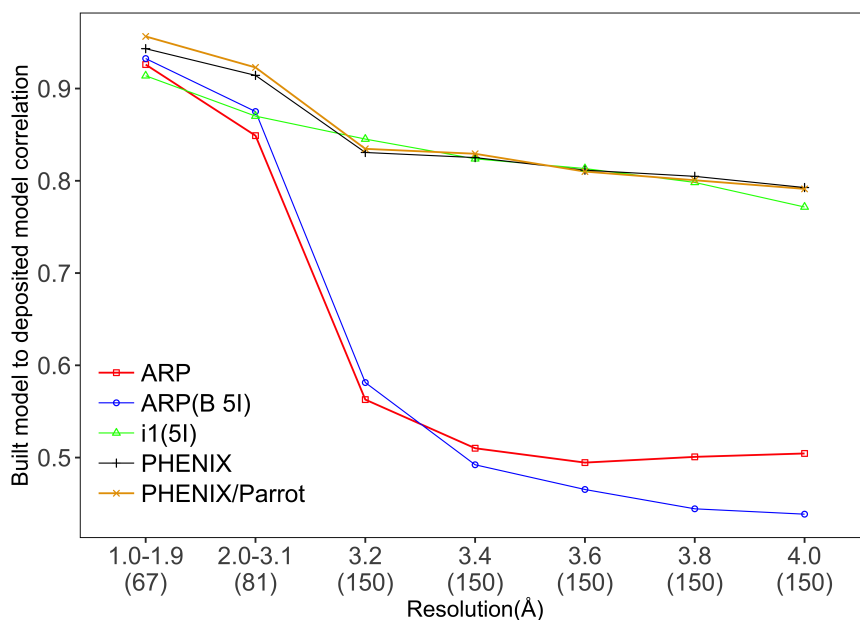


Figure 3.7: Mean correlation between built protein model and final deposited protein model for NO-NCS data sets partitioned into classes based on their resolution. The number of data sets in each class is indicated in brackets under the graph.

### 3.6.5 Pipeline execution time

Figure 3.8 shows mean execution times that the pipeline variants required to build the protein models for the original NO-NCS data sets from our comparison. Buccaneer in CCP4i was the fastest pipelines over all structures sizes. ARP/wARP averaged less than 50 min to build small structure, making it the second fastest pipeline after Buccaneer. Using Buccaneer in CCP4i models as an initial model for ARP/wARP slowed the building of the models compared to the normal run of ARP/wARP, with averages slightly higher than normal ARP/wARP. PHENIX AutoBuild, after Parrot and without Parrot, was the slowest pipeline with averages of around 200 min to build small structures and more than 1600 min for large structures. SHELXE required execution times between those of ARP/wARP and PHENIX AutoBuild, achieving the smallest average when building small structures, but with execution times increased to over 200 min when building large structures.

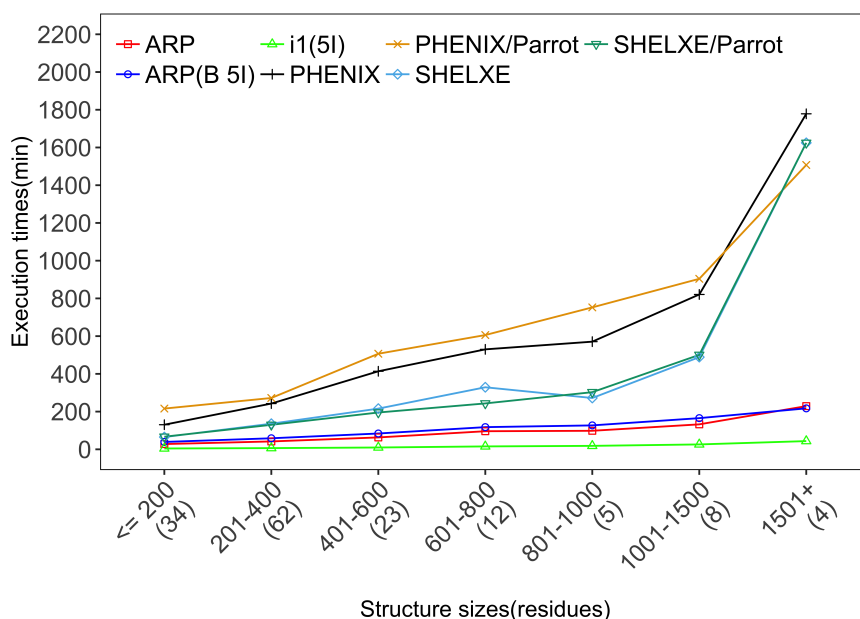


Figure 3.8: Mean pipeline execution times for the original NO-NCS data sets partitioned into classes based on their structure sizes. The number of data sets in each class is indicated in brackets under the graph.

## 3.7 Discussion

Comparisons of the different model building pipelines against a range of observed data sets, both at the original resolution and after simulated resolution reduction, highlight

different strengths and weaknesses of the different software. These may be used to guide users in choosing the most appropriate software for their problem, and developers in the improvement of their software or the construction of hybrid pipelines using multiple tools.

Comparison of the model completeness, as assessed by the fraction of the model alpha carbons built to within 1.0 Å of the correct location and assigned the correct residue type, suggests that at better than 3.1 Å resolution PHENIX Autobuild achieves the most complete models, with Buccaneer and ARP/wARP producing successively less complete models. PHENIX Autobuild was developed mainly against data at better than 3.0 Å resolution [83]. However, the comparison of structure correlation showed that PHENIX Autobuild built models with high correlation to final deposited models at worse resolutions which might be considered to be used as an initial model for further building iterations.

At worse than 3.1 Å resolution, Buccaneer substantially outperforms the other pipelines, with PHENIX Autobuild giving intermediate performance and ARP/wARP only building a small proportion of residues when averaged across many structures. This is consistent with expectations given that the original design criterion for Buccaneer was that it should be more robust against reduced resolution. Running ARP/wARP after Buccaneer leads to results which are worse than Buccaneer, suggesting that the residues successfully sequenced by Buccaneer are not being retained by ARP/wARP.

When comparing model completeness against initial map quality for the original resolution data sets, all the pipelines perform well when the initial phases are good (correlation > 0.64). Best results are obtained using PHENIX Autobuild, especially if after initial phase improvement using Parrot [50]. This suggests that phase improvement in Parrot is in some way complementary to the statistical phase improvement which is incorporated in the PHENIX Autobuild pipeline [20]. SHELXE also showed improved model building when starting from phases improved by Parrot.

When comparing R-work, the conclusions are somewhat different. ARP/wARP produces the lowest R-work across all resolution ranges, and produces dramatically lower R-work at worse than 3.1 Å resolution. PHENIX Autobuild comes close to ARP/wARP at better than 3.2 Å resolution. SHELXE produced the highest R-work because it only built the main chain. Sequence assignment and side chain modelling

are likely to significantly reduce the R-work as long as the chains built by SHELXE do not contain too many tracing errors.

When comparing R-free, a similar pattern emerges, although at worse than 3.1 Å resolution the R-free from the Buccaneer+ARP/wARP pipelines show a more modest gain over the other pipelines. (On the basis of developer recommendations and our tests which provided as supplementary material in Appendix A.8, no free set is used when running ARP/wARP on its own [84]).

The differing conclusions concerning the effectiveness of ARP/wARP from the three metrics are connected with the methodology. The use of dummy atoms in the ARP/wARP calculation allows the observations to be fit very well - and potentially overfit [85], however the portion of the model represented by dummy atoms does not contribute to the completeness score used here. The good R-free values obtained from ARP/wARP show that the dummy atom model has significant explanatory power at lower resolutions even when the dummy atoms cannot be explained in terms of sequenced main chain. This suggests that improved results may be possible either by using ARP/wARP as a preliminary step for another method, or by further development of the methods for interpreting the dummy atom model.

The performance of a model building algorithm is determined by multiple factors - the ability of the method to interpret an initial map, the ability of the pipeline to improve that map in the light of the model build so far, and the amount of finalization (e.g. waters, cis peptides and so on) which is performed by the pipeline. The results presented here suggest that Buccaneer may be the most effective tool for classifying features in the initial map especially at lower resolution, but lacks the finalization tools which are present in ARP/wARP and PHENIX Autobuild, and therefore leads to higher R-work. This suggested the use of ARP/wARP to finalize the Buccaneer model, however the model sequence tends to be lost at lower resolutions, limiting the benefit of this approach. PHENIX Autobuild has however successfully implemented Buccaneer as an optional preliminary step (not tested here).

The model building pipelines show considerable variability in performance from structure to structure, making a-priori recommendation of a single method for a given data set difficult. The speed and ease of use of the model building pipelines mean that users seldom need to try and anticipate which software will be most suitable -



instead most users are likely to use whichever software is most convenient for them. The results presented here may be of use in deciding which pipeline to try next in the case where the first option is unsuccessful. ARP/wARP and PHENIX Autobuild are likely to be better options at better than 3.1 Å resolution, where their advanced model finalization tools lead to lower R-work. As resolution drops below 3.1 Å, Buccaneer is more likely to produce the most complete model, however manual editing to remove wrongly built structure is also required.

Given that the software pipelines perform differently on different problem types, the results of any test will inevitably be biased by the choice of test data. In this case, data sets from the JCSG [72] were used - other JCSG data were also used in the development of Buccaneer, although those data sets were excluded from the results presented here. It is possible that this has led to some element of ‘tuning’ of Buccaneer to work on JCSG-sourced data, although the use of different programs for different structures within the JCSG pipeline may mitigate this. Similarly, the resolution truncation protocol used in for low resolution tests may lead to different results compared to genuine low resolution data sets. In our case, the resolution truncation procedure leads to better phases at low resolution than from a real low resolution data set. Finally, the evaluation criteria also dictate the results; in particular the counting of correctly placed and sequenced alpha carbons appear to penalize ARP/wARP at lower resolutions compared to the results of R-work/R-free comparisons. Which model is more desirable will depend on the needs of the downstream user.

### **3.8 Data and methods**

The comparison tool code, the structures built by the pipelines and logs files and the data used are available at <https://www.doi.org/10.15124/d4cb35df-a42d-4365-b539-9868730d165f>.

# Pairwise running of automated crystallographic model-building pipelines

In this chapter, we examine the usefulness of combining the existing protein model building pipelines to improve the built protein structures by running them in pairwise combinations. To this end, the chapter presents the use of pairwise pipeline combinations to build protein models for the same crystallography data sets as in Chapter 3, and uses structure completeness and R-free to assess the evaluate the resulting protein models.

## 4.1 Abstract

For the last two decades, researchers have worked independently to automate protein model-building, and four widely used software pipelines have been developed for this purpose: ARP/wARP, Buccaneer, PHENIX AutoBuild, and SHELXE. Here, we examine the usefulness of combining these pipelines to improve the built protein structures by running them in pairwise combinations. Our results show that integrating these pipelines can lead to significant improvements in structure completeness and R-free. In particular, running PHENIX Autobuild after Buccaneer improved structure completeness for 29% and 75% of the data sets we examined at original resolution and simulated lower resolution, respectively, compared to running PHENIX Autobuild on its own. In contrast, PHENIX AutoBuild alone produced better structure completeness than the two pipelines combined for only 7% and 3% of these data sets.

## 4.2 Introduction

X-ray crystallography has been used for several decades for the determination of protein structures with RNA/DNA, accounting for 90% of the deposited protein structures in the Protein Data Bank as of 2020 [1, 2]. Multiple steps are required to obtain a protein structure, starting with the crystallization process, obtaining an electron-density map from the diffraction pattern, and building the protein structure. Researchers have investigated ways to automate the building step, and four widely used pipelines have been developed: ARP/wARP [13, 14, 15, 16, 17], Buccaneer [18, 19], PHENIX AutoBuild [20, 86], and SHELXE [21, 22, 23, 24]. RNA/DNA can also be built automatically by PHENIX AutoBuild and other tools. The performances of these pipelines vary depending on electron-density map quality indicators such as resolution and phases. In Chapter 3, we conducted a comparison between these pipelines, and we found that the performance of the pipelines differs from one structure to another, which suggests that there is no best pipeline for all protein structures, although there is often a best pipeline for each protein structure [11].

Researchers have focused on different aspects of the protein-building problem and have developed appropriate methods depending on the coverage of their test data sets. As a result, pipelines tend to perform well when they are run using data sets with similar features to those that were used in developing the pipeline. Having data sets with different features generally makes the pipelines perform poorly. We addressed this matter here by running the pipelines in pairwise combinations, where the first pipeline from the combination built a protein structure as an initial structure for the second pipeline. Using these pairwise pipeline combinations often improved the final protein structure compared to using only one pipeline.

## 4.3 Data sets

We used the original data sets from [72], which have resolutions between 1.9 Å and 3.2 Å, and synthetic data sets obtained by truncating the original data sets to 3.2 Å, 3.4 Å, 3.6 Å, 3.8 Å and 4.0 Å (synthetic-resolutions) as described in Chapter 3. As in our comparison paper, 52 original data sets used in the development of Buccaneer and their

truncated resolutions were omitted from the main results (and are only presented in the supplementary material). This gave us 202 original and 1009 synthetic-resolution data sets initially, and 150 original and 750 synthetic-resolution data sets after omitting the Buccaneer development data sets.

Similarly large data sets of over 1000 structures have recently been used to improve ARP/wARP [87]. However, we were unable to use these data sets because this Chapter builds on Chapter 3, which used the original and synthetic data sets described above.

The density modification was done by Parrot [50]. Phase improvement was performed on the experimental phasing data, but NCS averaging was not used for those structures where NCS was present, with the aim of providing starting data with poorer phases both to test the limits of the model-building algorithms and to better simulate the poorer phases typically associated with lower resolution data sets.

## 4.4 Method of the pairwise running

We ran the same versions of pipelines as in Chapter 3 to compare individual pipelines with combined pipelines and the same high-performance cluster. As in Chapter 3, we allowed a maximum of 48 hours for the building of each structure because that was the highest time limit that the majority of our cluster nodes allowed.

Unlike in Chapter 3, here we tried to achieve the best performance of the pipelines, and to do that we changed the default parameters as necessary. “Rebuild in place” is a feature of PHENIX AutoBuild to improve input structure without adding or removing residues, it is based on removing and rebuilding a small segment of the main chain at a time with maintaining residues type, and it is used by default when the input structure is close to the correct structure [20]. PHENIX AutoBuild is unable to use “rebuild in place” when the initial structure contains unknown residues that cause a mismatch between the input model chains and the model sequence as matching is required to use this feature. This occurred in 13.7% and 3.5% of the structures built by Buccaneer and ARP/wARP, respectively. We forced PHENIX AutoBuild not to use this feature if it failed in the first attempt. An alternative workaround for this scenario is to remove the unknown residues before using the initial structure in Phenix AutoBuild.

SHELXE was not run after other pipelines because it only builds the main chain, while other pipelines build complete structures. However, SHELXE structures were used as input for other pipelines as the initial structure. Additionally, SHELXE structures were only built for the original-resolution data sets, as the synthetic structures fall outside the resolution range recommended for SHELXE.

We considered the same evaluation measures as in Chapter 3 except R-work. The different model parameterizations used by different model building programs lead to overfitting and underestimation of R-work in some cases, so we focus on R-free in this comparison. While the use of a free set is not normally recommended for ARP/wARP, in this experiment we are not primarily interested in individual pipeline performance, so we used a free set for analysis purposes [87]. ARP/wARP does not necessarily set aside the same free reflections as the other pipelines, so the REFMAC evaluation step was changed to use the same free set as that chosen by ARP/wARP when run immediately after ARP/wARP. Dummy atoms were not removed unless ARP/wARP removed them, as they did not significantly affect R-free.

In the next section, we deemed one pipeline or pipeline combination better than another when it produced an improvement of at least 5% in the relevant measure (completeness or R-free); other improvement thresholds are reported in Appendix B. Execution time was not considered here, as this was compared before for individual pipelines in Chapter 3.

## 4.5 Results

### 4.5.1 Overview

We present the results of our comparison using the pipeline and pipeline combination identifiers defined in Table 4.1. Table 4.2 shows the number of “complete”, “intermediate” and “failed” data sets for each of the pipeline variants (i.e., pipelines and pipeline combinations) that we used in our experiments. The data sets were marked as “intermediate” either when the 48-hour time limit was reached while the pipeline was still executing, or when the pipeline stopped/crashed before building the final structure. Data sets for which no structure was built were marked as “failed” and this occurred when the time limit was reached before the pipeline built an intermediate model.

Table 4.1: Pipeline and pipeline combination identifiers (IDs) used to present the results.

ID	Description
A	ARP/wARP
B	Buccaneer in CCP4i using 5 iterations
P	PHENIX AutoBuild
P*	PHENIX AutoBuild with Parrot
S	SHELXE
S*	SHELXE with Parrot
$x \rightarrow y$	Pairwise pipeline combination, with pipeline $y$ executed after pipeline $x$ , e.g., $A \rightarrow P^*$ denotes the pairwise combination in which PHENIX AutoBuild with Parrot is run after ARP/wARP

As shown in Table 4.2, structures were successfully built for most of the data sets; the pipelines only failed to build six data sets (original and synthetic data sets) out of 1211 data sets. After omitting the 52 data sets (used in Buccaneer development, cf. Section 4.3) and the failed data sets, 148 (original) and 746 (synthetic) data sets were used in the analysis, representing 74% of the original and synthetic data sets.

Table 4.3 shows the mean and standard deviation (SD) for the structure completeness and R-free achieved for these data sets by each pipeline variant. The pipelines built structures with high completeness from the original data sets, the majority of which are better than 2.5 Å. The highest mean completeness was 94% with 11% SD (for PHENIX AutoBuild followed by Buccaneer), compared to the lowest mean completeness of 78%, with 33% SD (for SHELXE followed by ARP/wARP). The highest mean completeness dropped to 50% with 30% SD for the synthetic data sets, whose resolution ranges from 3.2 Å to 4.0 Å. From the original data sets, the pipelines built the structures with a mean R-free between 0.26-0.33 and a SD between 0.04-0.10. When building the structures from synthetic data sets, the mean R-free increased to between 0.38-0.52 with SD between 0.05-0.08.

## 4.5.2 Structure completeness

Figure 4.1 shows the structure-completeness results for the original-resolution data sets. Running the pipelines in pairwise combinations shows significant improvements compared to running a single pipeline. For example, both PHENIX AutoBuild post ARP/wARP and Buccaneer post ARP/wARP achieved at least 5% higher structure completeness than ARP/wARP alone for 28% or more of the data sets; in contrast

Table 4.2: Complete and intermediate models produced by the 23 pipeline variants for the original and synthetic-resolution data sets, where ‘(T)’ and ‘(C)’ denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	original			synthetic		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed
A	202	0(T) 0(C)	0	1008	1(T) 0(C)	0
A → P*	201	1(T) 0(C)	0	1007	2(T) 0(C)	0
A → B	202	0(T) 0(C)	0	1009	0(T) 0(C)	0
B	202	0(T) 0(C)	0	1009	0(T) 0(C)	0
B → P*	197	4(T) 0(C)	1	1005	0(T) 0(C)	4
P*	199	1(T) 1(C)	1	1001	8(T) 0(C)	0
P* → A	200	1(T) 0(C)	1	1008	1(T) 0(C)	0
P* → B	201	0(T) 0(C)	1	1009	0(T) 0(C)	0
S*	200	2(T) 0(C)	0	-	-	-
S* → A	202	0(T) 0(C)	0	-	-	-
S* → B	202	0(T) 0(C)	0	-	-	-
S* → P*	196	4(T) 0(C)	2	-	-	-
A → P	199	2(T) 0(C)	1	1009	0(T) 0(C)	0
B → P	200	0(T) 0(C)	2	1003	2(T) 0(C)	4
P	199	1(T) 0(C)	2	1001	7(T) 0(C)	1
P → A	200	0(T) 0(C)	2	1002	6(T) 0(C)	1
P → B	200	0(T) 0(C)	2	1008	0(T) 0(C)	1
S	200	2(T) 0(C)	0	-	-	-
S → A	202	0(T) 0(C)	0	-	-	-
S → B	202	0(T) 0(C)	0	-	-	-
S* → P	197	3(T) 0(C)	2	-	-	-
S → P*	198	2(T) 0(C)	2	-	-	-
S → P	197	3(T) 0(C)	2	-	-	-

Models used in the comparison: 148 original and 746 synthetic .

ARP/wARP on its own was better than the two pipeline combinations for only 6% and 7%, respectively, of the data sets. Similarly, running PHENIX AutoBuild after Buccaneer increased the completeness for 30% of the data sets compared to running Buccaneer on its own, while Buccaneer alone was only better than this pipeline combination for 7% of the data sets.

Running PHENIX AutoBuild in combination with Buccaneer led to higher completeness than using ARP/wARP after or before PHENIX AutoBuild. Using Buccaneer to build an initial structure for PHENIX AutoBuild resulted in completeness improvements (of at least 5%) for 24% of the data sets, compared to only 10% when

Table 4.3: Mean and standard deviation (SD) for the structure completeness and R-free for the original and synthetic data sets. The tables are sorted by structure completeness.

<i>Original data sets</i>					<i>Synthetic data sets</i>				
Pipeline	Completeness		R-free		Pipeline	Completeness		R-free	
	mean	SD	mean	SD		mean	SD	mean	SD
$P^* \rightarrow B$	94	11	0.30	0.04	$P^* \rightarrow B$	50	30	0.43	0.08
$B \rightarrow P^*$	93	8	0.26	0.04	$B \rightarrow P$	49	29	0.38	0.07
$B \rightarrow P$	93	10	0.26	0.04	$P \rightarrow B$	49	30	0.43	0.08
$S \rightarrow P^*$	92	7	0.26	0.04	$B \rightarrow P^*$	48	29	0.38	0.07
$S^* \rightarrow P^*$	92	9	0.26	0.04	B	42	31	0.45	0.08
$S^* \rightarrow P$	92	9	0.26	0.04	$A \rightarrow B$	40	32	0.45	0.09
$S \rightarrow P$	92	9	0.26	0.04	$P^*$	25	16	0.42	0.05
$P^* \rightarrow A$	92	11	0.28	0.04	P	25	16	0.42	0.05
$P \rightarrow B$	92	14	0.31	0.05	$A \rightarrow P$	21	18	0.41	0.08
$P^*$	91	10	0.26	0.04	$A \rightarrow P^*$	20	18	0.41	0.08
P	90	15	0.27	0.05	A	3	9	-	-
$A \rightarrow P$	90	16	0.27	0.06	$P^* \rightarrow A$	2	8	0.51	0.06
$A \rightarrow P^*$	90	17	0.27	0.06	$P \rightarrow A$	2	8	0.52	0.06
$P \rightarrow A$	89	17	0.28	0.06					
$S \rightarrow B$	89	18	0.32	0.06					
$S^* \rightarrow B$	89	18	0.32	0.06					
$A \rightarrow B$	88	22	0.32	0.06					
B	85	23	0.33	0.07					
$S^*$	82	18	-	-					
$S^* \rightarrow A$	81	31	0.30	0.09					
A	80	30	-	-					
S	79	21	-	-					
$S \rightarrow A$	78	33	0.31	0.10					

ARP/wARP was used to build an initial model. These results dropped slightly to 20% and 9%, respectively, when Parrot was used before PHENIX AutoBuild.

It is interesting to consider to what extent the pairwise combination of pipelines produces a better model compared to running both of the component pipelines and picking the best result; this allows us to distinguish between the case where the second pipeline simply conserves the good features of the first and where the pipelines have complementary features which can augment one another. Table 4.4 shows the percentage of the original and synthetic data sets that are built at least 5% higher in structure-completeness by the combined pipelines or either of the two pipelines alone. Running PHENIX AutoBuild alone built the structures with higher completeness compared when ARP/wARP ran before it, 11% and 49% of the original and synthetic



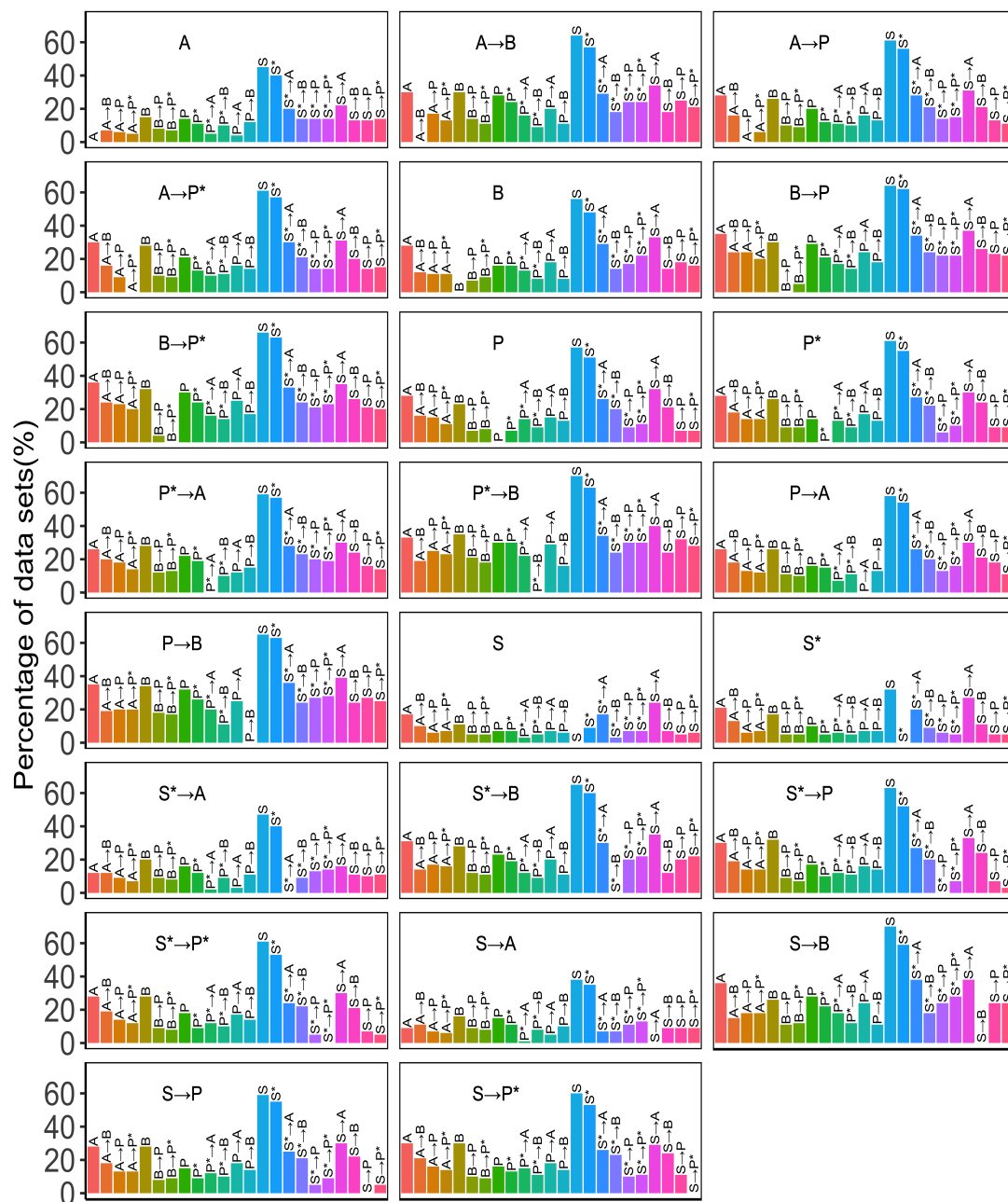


Figure 4.1: Structure completeness comparison for the models generated from the original data sets. Each plot corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of structures that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

data sets, respectively, built with higher completeness by PHENIX AutoBuild alone compared to 8% and 10% of the original and synthetic data sets, respectively when ARP/wARP ran in combination with PHENIX AutoBuild. However, Buccaneer with PHENIX AutoBuild showed greater benefits; only 2% and 11% of Buccaneer models built from the original and synthetic data sets respectively are better in terms of structure-completeness, compared to 14% and 41% of both data sets built with higher

completeness when PHENIX AutoBuild ran after Buccaneer.

Table 4.4: Structure completeness and R-free comparison for the original and synthetic data sets, indicating how often pairwise running outperforms either of the component pipelines. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of the models that either the combined pipeline ( $x \rightarrow y$ ) or the individual pipelines alone ( $x$  or  $y$ ) built at least 5% higher structure completeness and lower R-free.

Pipeline variant	Original						Synthetic					
	Completeness			R-free			Completeness			R-free		
	$x \rightarrow y$	$x$	$y$	$x \rightarrow y$	$x$	$y$	$x \rightarrow y$	$x$	$y$	$x \rightarrow y$	$x$	$y$
$A \rightarrow B$	14	3	8	-	-	-	27	0	33	-	-	-
$A \rightarrow P^*$	6	3	11	-	-	-	12	1	50	-	-	-
$A \rightarrow P$	8	4	11	-	-	-	10	0	49	-	-	-
$B \rightarrow P^*$	9	3	5	3	0	2	40	14	4	30	1	4
$B \rightarrow P$	14	2	2	4	0	3	41	11	2	29	1	4
$P^* \rightarrow A$	6	11	1	-	-	-	1	91	1	-	-	-
$P^* \rightarrow B$	14	3	2	0	29	0	47	7	17	9	23	4
$P \rightarrow A$	6	12	3	-	-	-	0	91	1	-	-	-
$P \rightarrow B$	17	7	3	0	36	0	42	7	18	8	24	5
$S \rightarrow A$	6	11	16	-	-	-	-	-	-	-	-	-
$S \rightarrow B$	22	4	11	-	-	-	-	-	-	-	-	-
$S \rightarrow P^*$	9	4	8	-	-	-	-	-	-	-	-	-
$S \rightarrow P$	13	4	7	-	-	-	-	-	-	-	-	-
$S^* \rightarrow A$	7	13	9	-	-	-	-	-	-	-	-	-
$S^* \rightarrow B$	21	6	11	-	-	-	-	-	-	-	-	-
$S^* \rightarrow P^*$	5	3	7	-	-	-	-	-	-	-	-	-
$S^* \rightarrow P$	12	5	7	-	-	-	-	-	-	-	-	-

Figure 4.2 shows the mean completeness for both original and synthetic data sets. Combined pipelines outperformed individual pipelines at resolution 1.0Å-2.0Å, and Buccaneer post PHENIX AutoBuild with Parrot outperformed the other pipeline variants at resolutions worse than 3.1 Å. PHENIX AutoBuild after Buccaneer obtained close results at resolutions worse than 3.1 Å, and ARP/wARP combined with PHENIX AutoBuild performed poorly at these resolutions.

Figure 4.3 shows how the mean completeness varied with the mean initial map correlation (F-map) for the original data sets. ARP/wARP running after PHENIX AutoBuild with Parrot at an initial map correlation lower than 0.5 led to above 90% completeness compared to running ARP/wARP on its own, which achieved lower than 60% completeness. When initial phases are better, the majority of the pipeline results

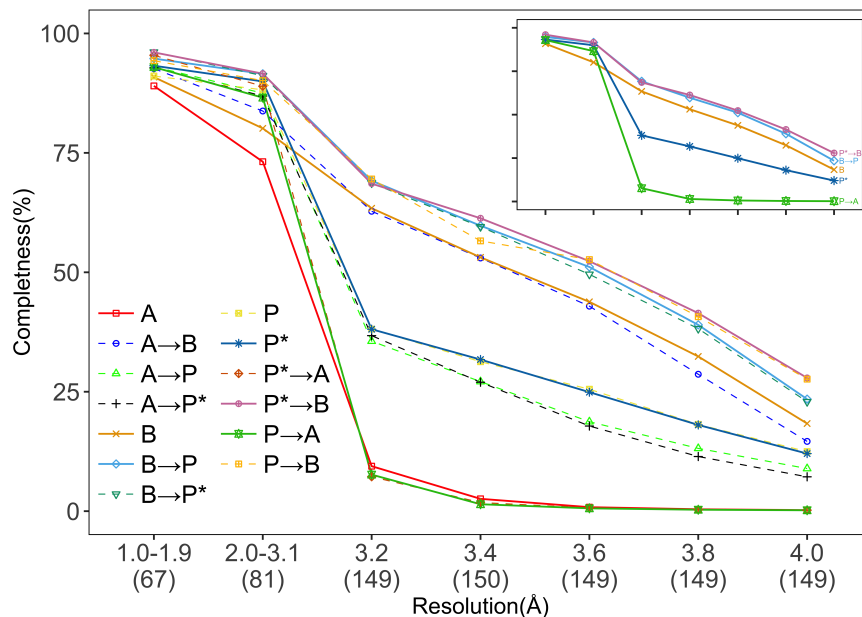


Figure 4.2: Mean completeness for the protein models built for all data sets. The data sets are grouped into bins based on their resolution, with the number of data sets in each bin shown in brackets under the graph. The insets figure indicated the pipelines that achieved the highest, middle and lowest mean completeness across the data sets bins.

reach higher than 90% completeness between 0.7-0.9.

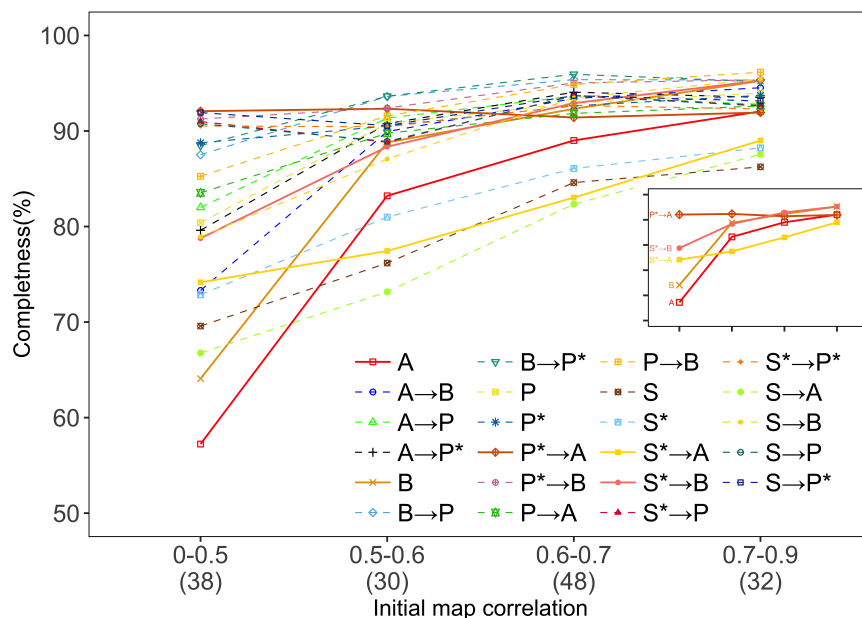


Figure 4.3: Mean completeness for the models built for the original data sets, grouped into bins based on their initial map correlation (F-map correlation); the number of data sets in each bin is reported in brackets under the graph. The insets figure indicated the pipelines that achieved the highest, middle and lowest mean completeness across the data sets bins.

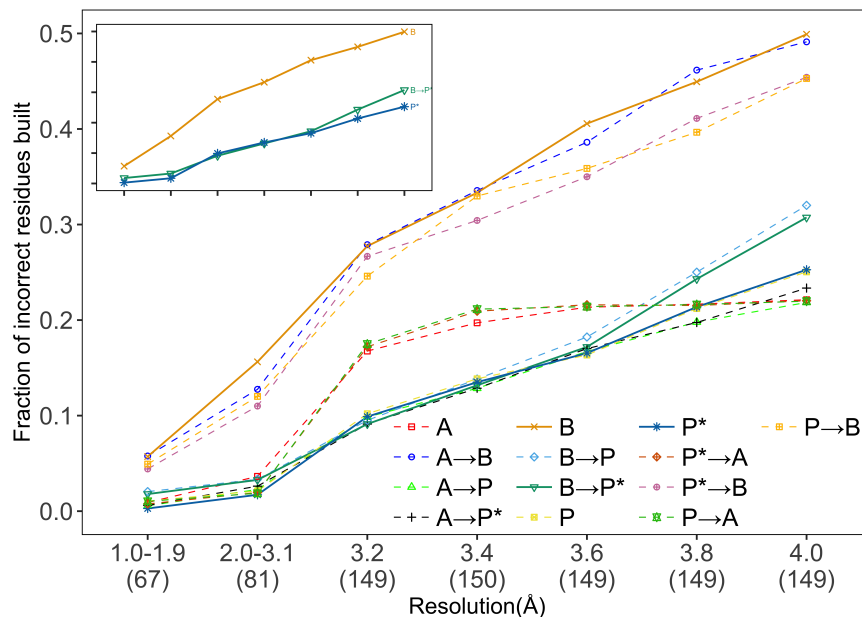


Figure 4.4: Mean residues incorrectly built for the protein models built for all data sets. The data sets are grouped into bins based on their resolution, with the number of data sets in each bin shown in brackets under the graph. The number of residues incorrectly built was normalized through dividing it by the number of residues in the deposited model. The insets figure indicated the pipelines that achieved the highest and lowest mean residues incorrectly built across the data sets bins.

Figure 4.4 shows the fraction of incorrect residues built for both original and synthetic data sets. Compared to other pipelines, a known problem of using Buccaneer is that Buccaneer may build a large number of incorrect residues, which can be 50% of the structure at 4.0 Å. PHENIX AutoBuild outperformed Buccaneer in lowering the number of incorrect residues, and using PHENIX AutoBuild post Buccaneer reduced junk residues to around 30% of the structure at 4.0 Å.

Figure 4.5 provides an illustration of a case for which pairwise running of two pipelines gave substantially better results than either pipeline alone, in this case, the structure 2AWA. The Buccaneer model is substantially incomplete, which some correctly traced fragments but only 8% of the sequence correctly docked. The PHENIX AutoBuild model is more complete, but still only 59% of the sequence is correctly docked. When both pipelines are used, a largely complete model is obtained and correctly sequenced. Running PHENIX AutoBuild with Parrot after Buccaneer built a structure with higher completeness that is 91%.

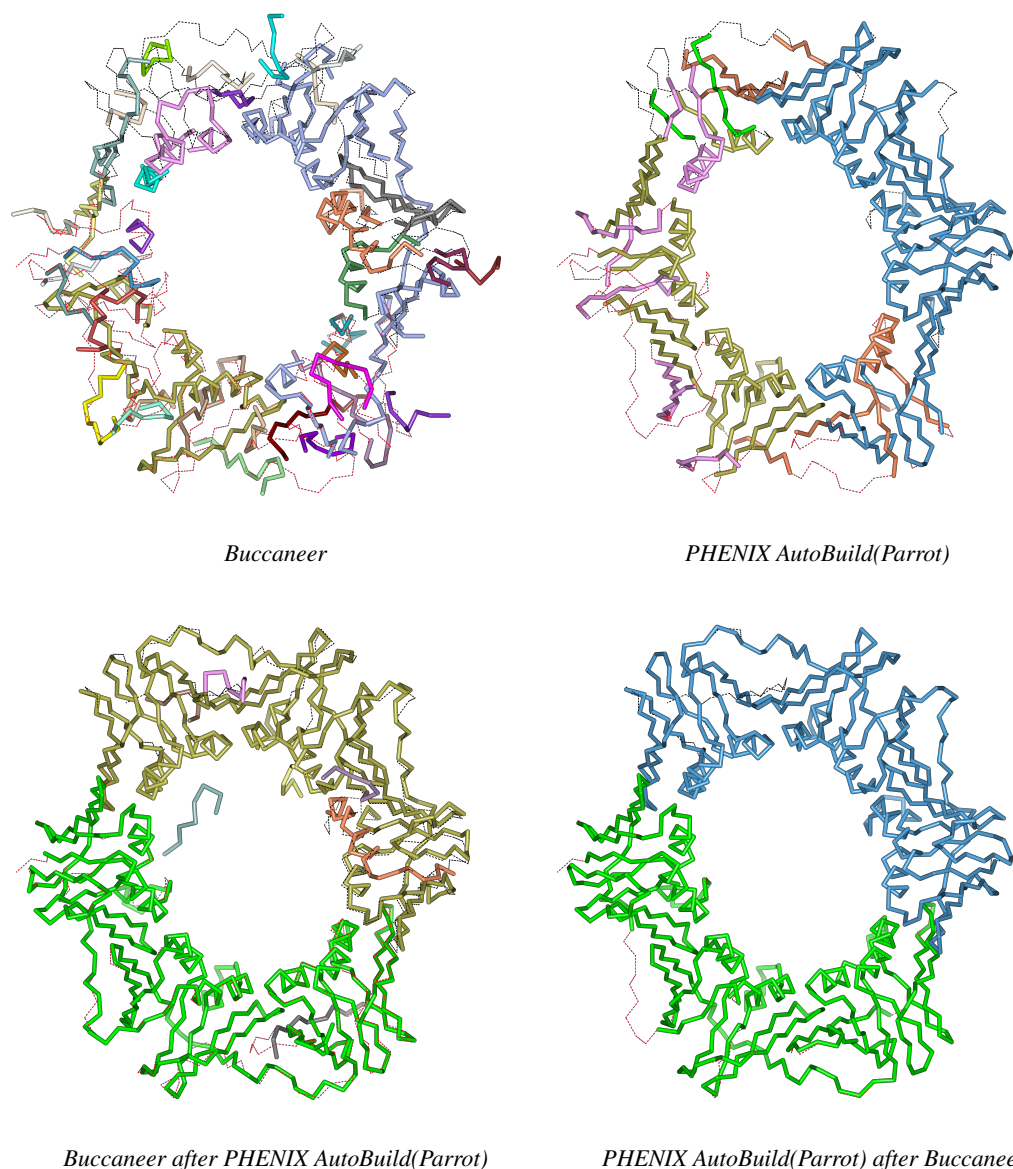


Figure 4.5: Four structures built by Buccaneer, PHENIX AutoBuild(Parrot) and their combinations, and compared to the deposited structures. The chains of deposited structures are coloured in red and black bonds. The PDB ID is 2AWA, and its resolution is 2.7 Å.

### 4.5.3 R-free

Figure 4.6 shows the R-free results for the original-resolution data sets. Similar to the completeness comparison from Section 4.5.2, individual pipelines performed worse than when we used them in combination with other pipelines. Comparing Buccaneer on its own to the combination in which it was followed by PHENIX AutoBuild shows significant improvement by including AutoBuild, as the structures produced for 65% of the data sets decreased (by at least 5%) in R-free when PHENIX AutoBuild ran after

Buccaneer. None of the structures built by Buccaneer on its own was better in R-free than those built by PHENIX AutoBuild after Buccaneer.

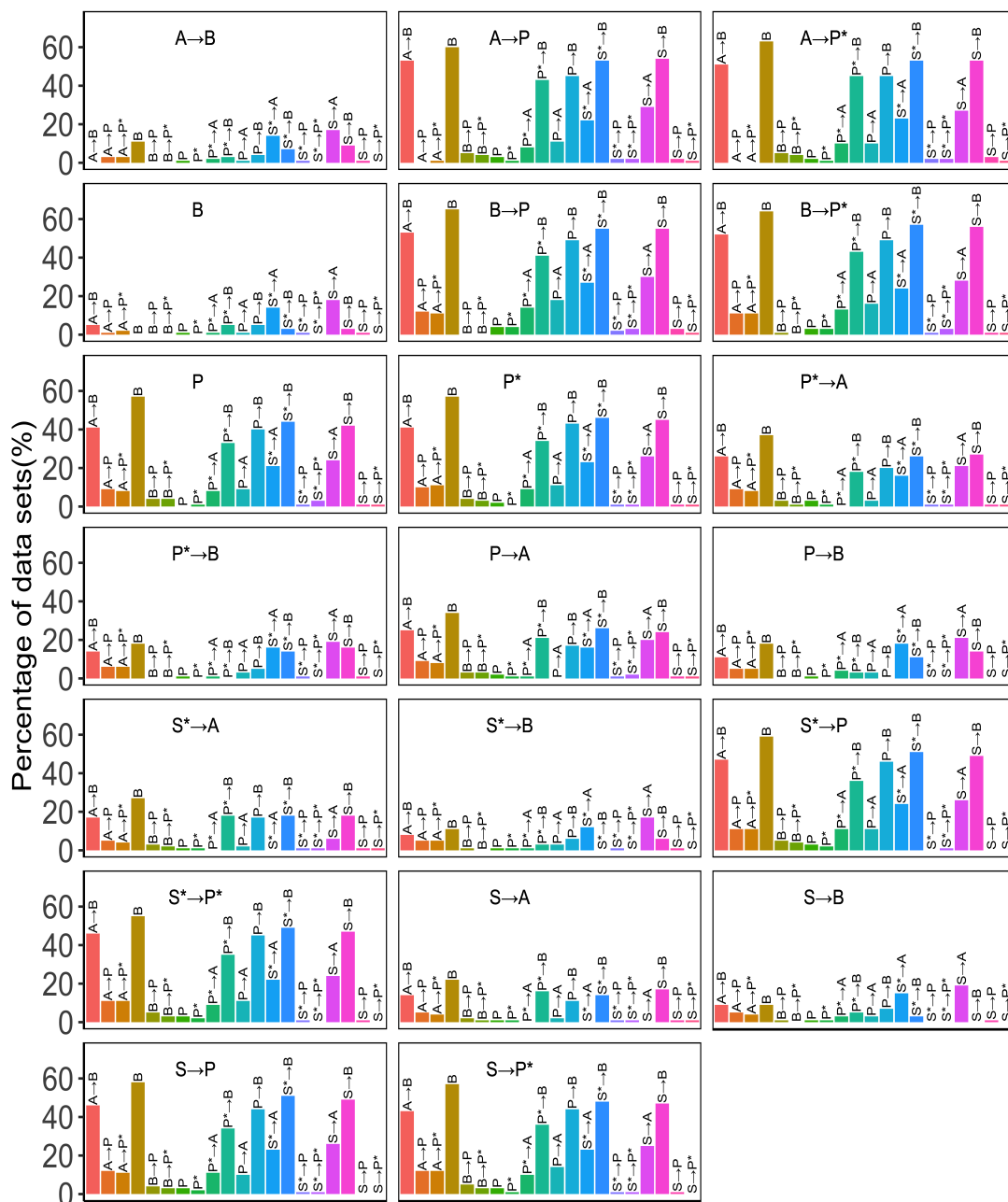


Figure 4.6: Comparison of R-free (rounded to two decimal places) for the structures generated from the original data sets. Each plot shows the percentage of models that a pipeline variant built with R-free at least 5% lower than each other pipeline variant.

Finalising the structures with Buccaneer as the second pipeline of a pipeline combination caused high R-free, while starting with a Buccaneer structure as an initial model for other pipelines was more effective. As shown in Table 4.4, Buccaneer after PHENIX AutoBuild did not improve R-free compared to PHENIX AutoBuild alone

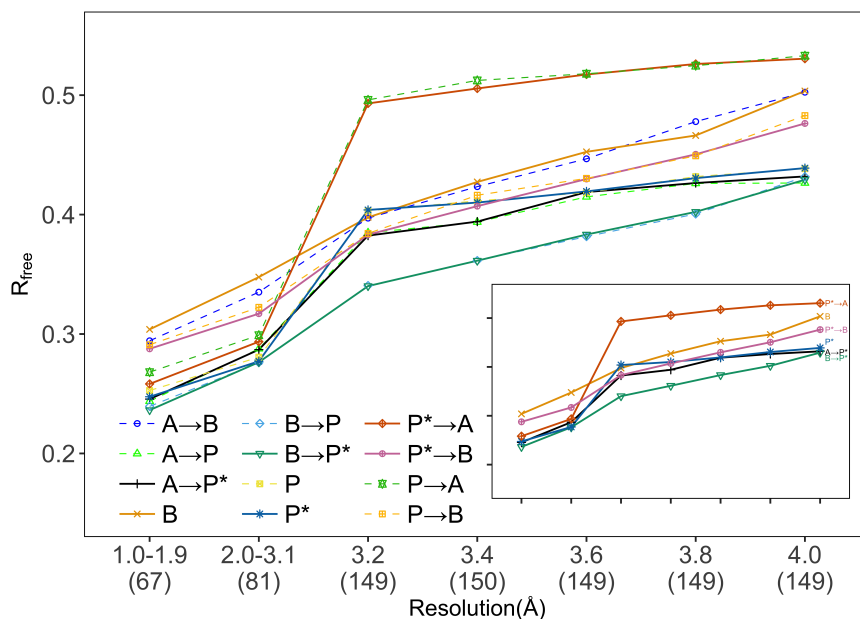


Figure 4.7: Mean protein model R-free for the data sets partitioned into classes based on their resolution. The number of data sets in each class is indicated in brackets under the graph. The inset figure indicated the pipelines that achieved the highest, middle and lowest mean R-free across the data sets bins.

as 36% of the original data sets have lower R-free. Running PHENIX AutoBuild after Buccaneer improved 4% of the original data sets in terms of R-free and no Buccaneer models have lower R-free than the combination. Following PHENIX AutoBuild by ARP/wARP generated better results than using Buccaneer after PHENIX AutoBuild. ARP/wARP built 17% of the data sets with better R-free than Buccaneer, while only 3% were built better by Buccaneer compared to ARP/wARP.

Figure 4.7 shows the mean R-free for the data sets grouped into classes based on their resolution. Running PHENIX AutoBuild with Parrot after ARP/wARP or Buccaneer led to lower R-free at resolutions better than 1.9 Å compared to Buccaneer or ARP/wARP run after PHENIX AutoBuild. The combination of Buccaneer and PHENIX AutoBuild achieved the lowest R-free across all pipeline combinations at resolutions worse than 3.1 Å, while ARP/wARP after PHENIX AutoBuild achieved the highest R-free for the same resolution range.

#### 4.5.4 Structure correlation

Figure 4.8 shows mean correlation between built protein models and final deposited protein models grouped into classes based on their resolution. At resolution better than

3.2 Å, all the pipeline variants achieved structure correlation higher than 0.8, however, only PHENIX AutoBuild with and without Parrot after Buccaneer still achieved the same figure at 4.0 Å. Structure correlation for ARP/wARP after PHENIX AutoBuild variants dropped significantly at 3.2 Å to below 0.6 and below 0.5 at worse than 3.6 Å.

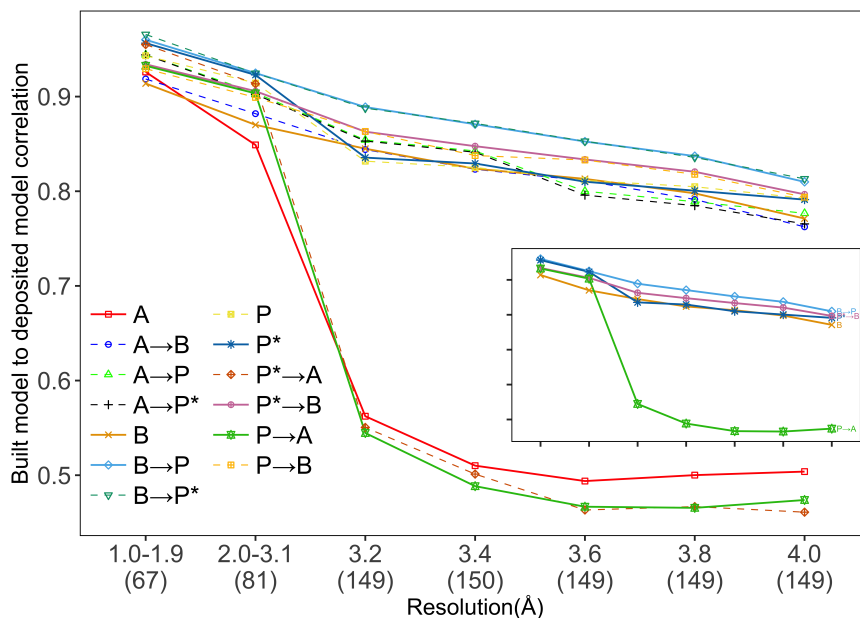


Figure 4.8: Mean correlation between built protein model and final deposited protein model partitioned into classes based on their resolution. The number of data sets in each class is indicated in brackets under the graph. The inset figure indicated the pipelines that achieved the highest, middle and lowest mean structure correlation across the data sets bins.

## 4.6 Discussion

We presented pairwise running of widely used model-building pipelines using original and lower resolution simulation data sets, and we focused on the successful combinations. We have focused on the results of running pipelines in sequence with at most minor adjustments to the pipeline options, however in future it may be possible to produce further improvements by deeper integration of methods from different pipelines.

Combining the pipelines improved the structure built by the first pipeline in most of the data sets. The significance of the improvement depended on the limitations of the first pipeline and the ability of the second pipeline to address these limitations. Running Buccaneer after PHENIX AutoBuild improved the structure completeness at resolutions worse than 3.1 Å, as it is known that PHENIX AutoBuild is more effec-



tive at resolutions better than 3.0 Å. Running the same two pipelines in reverse order yielded better results than either pipeline because PHENIX AutoBuild was able to address poor finalization of the model by Buccaneer, leading to improved R-free.

When we compared the structure completeness on the basis of the initial map correlation, few pipeline combinations performed well when the initial phases were poor. ARP/wARP after PHENIX AutoBuild obtained the best results when PHENIX AutoBuild ran after Parrot. Also, PHENIX AutoBuild after SHELXE, and Buccaneer after PHENIX AutoBuild with Parrot obtained close results. We notice from these combinations that the pipelines that do density modification internally during model-building produced a good structure for others to use as an initial structure. For example, Buccaneer after SHELXE showed better results than Buccaneer alone as SHELXE contributes substantially to phase quality and Buccaneer performance is affected by the quality of the phases.

When comparing R-free, most of the pipeline variants achieved close R-free at resolutions better than 3.1 Å, and PHENIX AutoBuild ran after Buccaneer outperformed the others at resolutions worse than 3.1 Å. ARP/wARP run after PHENIX AutoBuild, and Buccaneer run after ARP/wARP were the worst combinations at resolutions worse than 3.1 Å, as they produced structures with the highest mean R-free values. In line with R-free results, PHENIX AutoBuild ran after Buccaneer built the protein structures close to the deposited structures based on the structure correlation.

The results of our comparison show the usefulness of pipeline combinations instead of running them individually. Pairwise pipeline combinations have the ability to fix errors caused by the first pipeline in the combination. For instance, Buccaneer alone often produced a highly complete structure but with a large number of incorrect residues due to its building method. In contrast, when Buccaneer was followed by PHENIX AutoBuild, the number of incorrect residues significantly reduced because of the ability of PHENIX AutoBuild to fix the structure without adding new residues. The pipelines that do not perform density modification as a part of model-building (e.g., ARP/wARP and Buccaneer) showed the worst results against the initial map correlation (correlation < 0.5). Therefore, combining ARP/wARP and Buccaneer with PHENIX AutoBuild produced a more complete structure than that generated by either ARP/wARP or Buccaneer alone, both when PHENIX AutoBuild was used on its own

or with Parrot. The performance of the pipelines might be biased due to our approach in truncating the data sets to lower resolution, as genuinely low-resolution results from the crystalline disorder as increasing of the disorder drop off the scattering as well as increased phase errors due to lower signal to noise going into the phasing calculation, which these not simulated in the truncation approach , however this was necessary due to the difficulty of obtaining large real data sets.

The decision of which pipeline to start with depends on the quality of the electron-density map. When the initial phases are not good, starting with a pipeline that includes density modification is the most effective approach. However, the decision can change from one structure to another, even if the structure features are very similar. Running all these pipelines variants can be time-consuming, and there is not one individual or combined pipeline that is the best across all resolution ranges. Developers are inevitably influenced by their own interests and by the coverage of their test data sets. Combining features from different model building pipelines improves model building results because in many cases the complementary features of models from different pipelines are preserved. Further efforts to understand the strengths and weaknesses of different tools may allow further improvements through a more systematic approach to combining components from different pipeline. Moreover, further research is required to provide users with clear guidelines for which individual pipeline or combined pipeline is the best depending on their model features, for example, the quality of initial phases, as shown in this chapter and Chapter 3 affect the model compilation.

## 4.7 Data and methods

The structures built by the pairwise pipeline combinations and the associated logs files are available at <https://doi.org/10.15124/4b7c880a-d6b0-471a-a379-d52c4ee947fe>.

# **Predicting the performance of automated crystallographic model-building pipelines**

In this chapter, we present a machine learning model for predicting the performance of crystallographic model-building pipelines. We used the data sets from Chapter 3 in addition to two other data sets to train an ML model that accurately predicts the R-free, R-work and structure completeness of the protein model that each pipeline and pairwise pipeline combination can build from a given crystallography data set.

## **5.1 Abstract**

Proteins are macromolecules that perform essential biological functions which depend on their three-dimensional structure. Determining this structure involves complex laboratory and computational work. For the computational work, multiple software pipelines have been developed to build models of the protein structure from crystallography data. Each of these pipelines performs differently depending on the characteristics of the electron-density map received as input. Identifying the best pipeline to use for a protein structure is difficult, as the pipeline performance differs significantly from one protein structure to another. As such, researchers often select pipelines that do not produce the best possible protein models from the available data.

Here, we introduce a software tool which predicts key quality measures of the protein structures that a range of pipelines would generate if supplied with a given crystallography data set. These measures are crystallographic quality-of-fit indicators

based on included and withheld observations, and structure completeness. Extensive experiments carried out using over 2500 data sets show that our tool yields accurate predictions for both experimental phasing data sets (at resolutions between 1.2Å and 4.0Å) and molecular replacement data sets (at resolutions between 1.0Å and 3.5Å). The tool can therefore provide a recommendation to the user concerning the pipelines that should be run in order to proceed most efficiently to a depositable model.

## 5.2 Introduction

The first protein structures were determined in the 1950s using X-ray crystallography [88]. By 2020, the number of solved protein structures deposited in the Protein Data Bank (PDB) exceeded 154,000 [1, 2]. To enable this progress, researchers have automated the computational work of determining the protein structure from X-ray crystallography data sets. Multiple protein model-building pipelines have been developed within the last three decades: ARP/wARP [13, 14, 15, 17, 16] Buccaneer [18, 19] PHENIX AutoBuild [20, 86] and SHELXE [21, 22, 23, 24]. In recent studies, we showed that the performance of these pipelines differs significantly from one protein structure to another in Chapter 3—which makes selecting a particular pipeline difficult; and that using a pair of pipelines is sometimes the best option [89]—which greatly increases the number of options that crystallographers can choose from.

An important step in building the protein structure involves solving the *phase problem*. The phase problem may be solved by either molecular replacement or experimental phasing methods, e.g. McCoy and Read [7] and Evans and McCoy [6]. These methods lead to electron-density maps with rather different properties: in the case of experimental phasing, the maps usually contain noise due to ambiguity in the experimental phasing, whereas in the molecular replacement case the errors in the map can arise from possible bias towards the molecular replacement model. The resolution of the experimental observations, the quality of experimental phasing or the similarity of the molecular replacement model, and many other features such as ice rings may also affect the quality of the data. Each of these factors impact the performance of different model-building algorithms in different ways [10, 11, 12].

The model building process also contains stochastic elements. The placement of

a first atom or residue in a chain will in turn influence the placement of all subsequent elements, and so substantially different model building results may be obtained from very slight perturbations of the initial conditions. This is addressed in one model building pipeline by building multiple models at each stage of the process [20].

We examined a selection of 3273 research papers cited in PDB to evaluate how crystallographers currently choose which model-building software pipeline to use, by searching for occurrences of the pipeline names in the text of each paper, and excluding papers where the search results were ambiguous or multiple tools were mentioned. The results are plotted against year, journal, and the country of the first author in Figure 5.1. The most striking feature of this analysis is the correlation between the first author's country and the country where each pipeline has been developed, with US researchers more likely to use PHENIX Autobuild, UK researchers more likely to use Buccaneer, and German researchers more likely to use ARP/wARP. While there are practical reasons which might explain this correlation (e.g. access to developers and workshops), it would be surprising if cognitive biases such as the affinity bias [90], to which we are all subject, did not play a role.

To help eliminate this bias, we have developed a software tool that uses a machine learning (ML) model to predict the performance of a wide range of model-building pipelines and pipeline combinations for a given crystallography data set. Our prediction tool serves three purposes:

- To provide users with a more efficient route to a higher-quality depositable structure for their specific data set.
- To challenge users to try different pipelines, and multiple combinations of pipelines, on the basis of likely performance rather than on the basis of familiarity or affinity to the pipeline developers. Given that all pipelines provide very convenient user interfaces, the overhead of trying a new pipeline will cost less than the effort of model completion from a suboptimal starting point.
- To assist future developers in the development of meta-tools which make use of multiple pipelines to further automate the process of structure solution and to obtain more complete models.

To the best of our knowledge, this is the first ML solution that guides the user

in the selection of the model-building pipelines best suited for a given crystallography data set. While a predictive model that employs similar ML techniques was recently proposed in Vollmar et al. [12], that model addresses the complementary problem of predicting the usefulness of collected crystallography data sets.

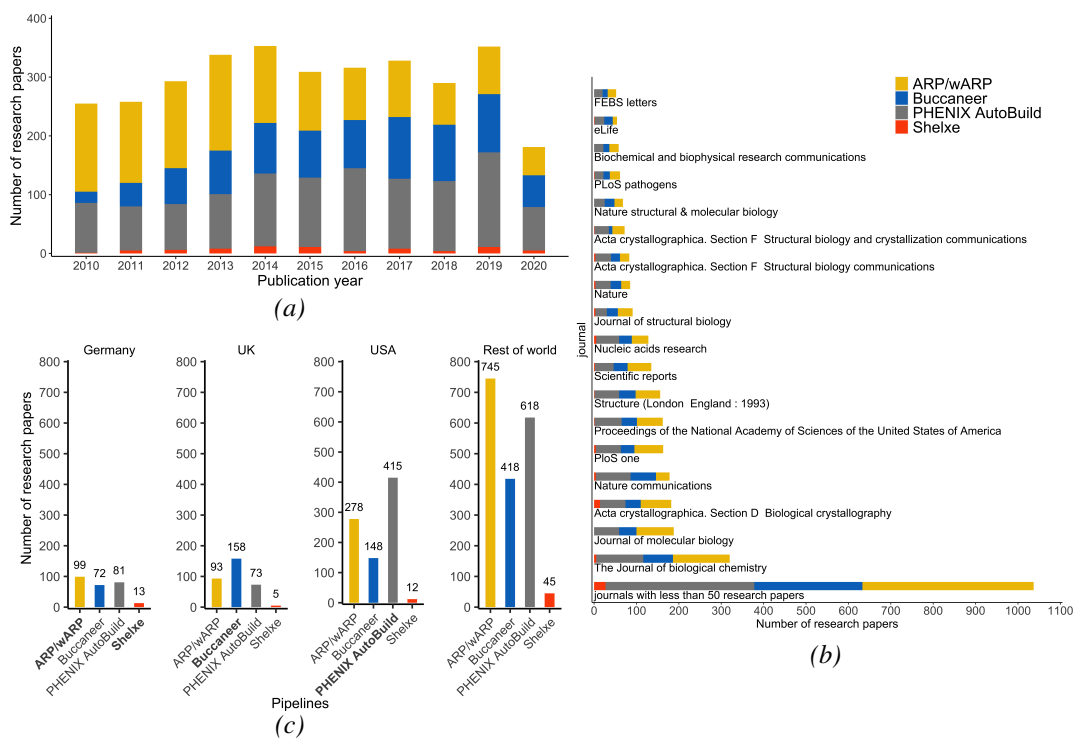


Figure 5.1: Analysis of the crystallographic model-building pipelines used in 3273 PDB protein-structure research papers published between 2010 and 2020. The papers were identified using either their PubMed Identifier or DOI obtained from PDB. We omitted the research papers that used multiple pipelines. We compared the number of uses of each pipeline in its base country, depending on the home country of the first author’s organization. (a) The number of research papers across the publication years for each pipeline. (b) The journals where the research papers were published, where the journals with under 50 research papers are combined into one group. (c) The number of uses for each pipeline in its base country, and across the rest of the world; the pipeline names are shown in bold in their base-country plot.

## 5.3 Predictive model

### 5.3.1 Data sets

We used data sets from three sources to train and evaluate our ML predictive model: 1203 experimental phasing data sets from the Joint Center for Structural Genomics (JCSG) [72, 11] (same as the data sets used in Chapter 3), 32 newer experimental phas-

ing data sets deposited between 2015–2021 and taken from PDB, and 1332 molecular replacement (MR) data sets from [26]. These data sets correspond to two techniques that can be used to build a protein structure; experimental phasing and MR (as explained in Chapter 2).

The resolution of JCSG experimental phasing data sets ranges from 1.2 Å to 4.0 Å, with the low resolution data sets augmented by simulation as in Chapter 3; the phases were solved using SAD/MAD techniques [72]; the resolution of the PDB experimental phasing data sets ranges from 1.1 Å to 5.8 Å; and the resolution of the MR data sets ranges from 1.0 Å to 3.5 Å. Lower resolution data sets have fewer experimental observations, which decreases the performance of the protein-building pipelines.

The way in which we partitioned these data sets into data for the training and data for the evaluation of our ML model is described in Section 5.3.5.

### 5.3.2 Crystallographic model-building pipelines

The four pipeline versions used in our work are PHENIX AutoBuild v.1.14, Buccaneer in CCP4i v.7.0.066, ARP/wARP v.8 and SHELXE v.2019/1. These pipelines were run using the default parameters, both individually and in pairwise combinations where the protein model produced by a first pipeline  $x$  was supplied as input to a second pipeline  $y$ .

### 5.3.3 Protein structure evaluation

We focused on predicting three protein-structure evaluation measures, namely R-free/R-work and structure completeness (as described in Chapter 3).

### 5.3.4 Electron-density map features

We trained our ML prediction model using as input features (i) the resolution of the crystallography data set; and (ii) the following measures of the quality of the electron-density map:

- RMSD—the root-mean-square deviation of the electron-density from the mean of the map;

- Skew—the third moment of the electron-density about the mean, which measures the asymmetry of the electron density histogram [91];
- Maximum density—the highest density of the electron-density map;
- Minimum density—the lowest density of the electron-density map;
- Sequence identity—the sequence identity calculated through superposition of the homologue chain onto the target chain using GESAMT [92, 26].

### 5.3.5 Predictive model training

The individual pipelines were run on all data sets from Section 5.3.1. The pipeline combinations were only run on the experimental phasing data sets, as building protein models from such “raw data” can often be improved by using pipeline combinations [89]. The results of these runs are described in detail in Chapters 3 and 4. The data sets and the protein structures obtained from these runs were used to train and evaluate the predictive ML model as follows:

- 80% of the JCSG experimental phasing data sets, and 80% of the MR data sets were used to train the predictive model;
- the remaining 20% of the JCSG experimental phasing and MR data sets, and all 32 PDB experimental phasing data sets were used to evaluate the trained model.

We used random forests [56] as implemented in Weka framework [93, 94] for the predictive model, as this approach showed the lowest error rate across the ML algorithms that we tested, and that included support vector machine [95] and the RepTree decision tree algorithm. We varied the number of trees in the random forest from 1 to 5000 in geometric sequence, and 1024 was chosen for the final training, as this showed the lowest error rate. The depth of the trees was set to unlimited and bagging [96] was used to reduce the variance. We trained the predictive model using a 173-node high-performance cluster with 7024 Intel Xeon Gold/Platinum cores and a total memory of 42 TB.

A separate regression ML model (random forests model) was trained for each of the 24 pipeline variants (i.e., individual pipelines or pipeline combinations) from



Table 5.1 and for each of the three structure-evaluation measures from Section 5.3.3 relevant to the considered pipeline variant. For instance, R-free is not relevant for ARP/wARP and SHELXE with and without Parrot used on their own, so no ML model was built for these individual pipelines and R-free. We obtained 69 and 10 regression ML models in total for experimental phasing and for MR, respectively. Our predictive model consists of these regression ML models taken together.

We used the root-mean-square error (RMSE) and mean absolute error (MAE) measures to compare the accuracy of our predictive model to that of a “baseline” predictive model. In line with the standard practice for the evaluation of regression models, we used zero-R algorithm as a baseline predictive model [97]. Given a pipeline variant and any evaluation data set, the zero-R algorithm predicts that the R-free/R-work and structure completeness for the structure built by the pipeline would be the same as the median R-free/R-work and structure completeness for the training data sets, respectively.

To evaluate the accuracy of the predictive model for data sets of different resolutions, we partitioned the evaluation data sets into classes based on their resolutions, and we examined the prediction errors for each such class. Finally, to show the time saved by running only the pipeline variant predicted to build the best protein structure for a data set, we compared the execution time of this pipeline to the time required to run all the pipeline variants for that data set.

To quantify the uncertainty of the ML prediction, we calculate prediction intervals using the kernel estimator method from [98]. The width of these intervals reflects the prediction uncertainty. As such, we sort and report the pipelines in increasing prediction interval width order, with the pipelines of similar prediction uncertainty (i.e., with no more than 5% difference in prediction interval width) grouped together.

Finally, we generate a script for each pipeline and pipeline combination, ensuring that the users of our tool can run the individual pipelines and pipeline combinations in the manner used to obtain the training data sets for our ML prediction model. Furthermore, these ready-to-run scripts are customized based on the data provided by the tool users.

## 5.4 Predictive model evaluation

### 5.4.1 Evaluation of crystallography data set features used for model training

We evaluated the importance of the features used to train our predictive model by removing one feature at a time and comparing the accuracy of the model trained without that feature to the accuracy of the predictive model when trained on all the features. Figure 5.2 shows the difference in MAE and RMSE when one feature is removed compared to when all the features are used in the training, for each of the four individual pipelines, with separate MAE and RMSE presented for the JCSG experimental phasing and the MR data sets.

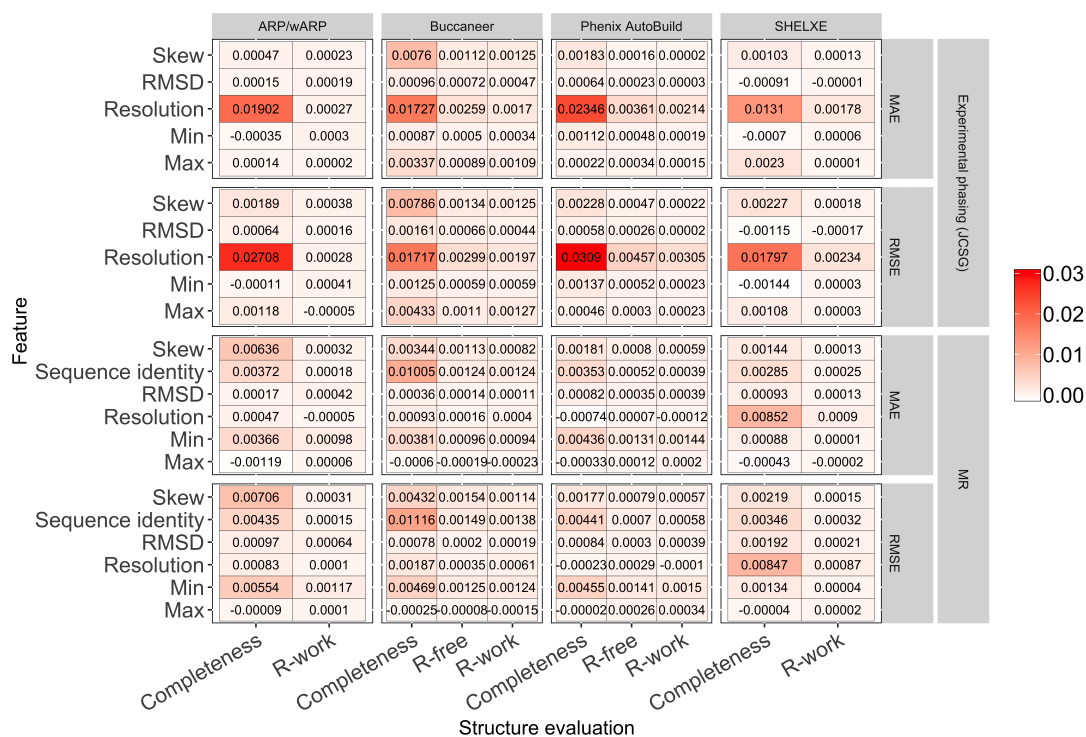


Figure 5.2: Ablation studies showing the difference in mean absolute error (MAE) and root mean squared error (RMSE) between when ML model trained on all features and when one feature is removed at a time. Higher values indicate more important features.

This analysis indicates that Phenix AutoBuild and ARP/wARP are more dependant on the data set resolution than Buccaneer, in line with previous results in Chapter 3. However, Phenix AutoBuild and ARP/wARP are less sensitive to the resolution for

MR data sets compared to experimental phasing data sets. RMSD and skew have different effects on the performance of the pipelines. For example, Buccaneer is affected by these two features more than Phenix AutoBuild for the experimental phasing data set, indicating a greater dependence on the noise level in the starting map. For MR data sets, the sequence identity affected the performance of all pipelines, with the highest effect for Buccaneer.

## 5.4.2 Evaluation of predictive model performance

Table 5.1 shows the MAE and RMSE for both types of data sets (experimental phasing and MR), for each of the three protein structure evaluation measures. For the JCSG experimental phasing data sets, both the MAE (0.04–0.19) and RMSE (0.08–0.26) of predicting the protein structure completeness are higher than the MAE and RMSE for the other measures. These values decreased when predicting R-free/R-work for MAE (0.02–0.06) and RMSE (0.02–0.08). For MR data sets, the MAE of structure completeness increased to 0.15–0.21 and RMSE to 0.20–0.29. The MAE of R-free/R-work was between 0.02–0.07, compared to RMSE, which is between 0.04–0.09.

Different levels of predictability were achieved for different pipeline variants. For the experimental phasing data sets and ARP/wARP after PHENIX AutoBuild, the predictive model achieved the lowest structure completeness MAE (0.04) with close RMSE, which indicates a small number of large error predictions. On the other hand, for MR data sets, the MAE of structure completeness for ARP/wARP and PHENIX AutoBuild ran individually increased to 0.20 and 0.21, respectively. Buccaneer ran individually and after ARP/wARP or PHENIX AutoBuild showed the lowest predictability, with MAE and RMSE values above 0.17.

R-free/R-work are more predictable across all pipeline variants and for both types of data sets, with MAE and RMSE values lower than those achieved for the structure completeness. For the JCSG experimental phasing data sets, the predictive model achieved a low MAE for R-work (0.02–0.03) and only slightly larger MAE for R-free (0.03–0.05) for all the individual pipelines. The MAE obtained for pipeline combinations and R-work ranged between 0.02–0.05, and that for R-free varied between 0.04–0.06. RMSE is slightly higher than MAE for both the individual and the combination pipelines. For the MR data sets, the MAE of R-work is between 0.02–0.06, with

Table 5.1: Mean absolute error (MAE) and root mean squared error (RMSE) of structure completeness and R-free/R-work for two experimental phasing data sets and molecular replacement (MR) data sets. ARP/wARP and SHELXE are not used R-free. For MR data sets, only individual pipelines were run. MAE and RMSE were calculated for the ML predictive model (P), and median predictor (M) used as a baseline (zero-R) model. The values are highlighted in light red, where the ML predictive model has higher MAE or RMSE than the median predictor; otherwise, the values are highlighted in light green.

Pipeline variant	Experimental phasing (JCSG)										Experimental phasing (recently deposited data sets)													
	MAE					RMSE					MAE					RMSE								
	Completeness		R-free		R-work	Completeness		R-free		R-work	Completeness		R-free		R-work	Completeness		R-free		R-work				
	P	M	P	M	P	P	M	P	M	P	P	M	P	M	P	P	M	P	M	P	M			
ARP/wARP	0.06	0.15	-	-	0.03	0.03	0.14	0.34	-	-	0.05	0.05	0.27	0.57	-	-	0.04	0.05	0.38	0.72	-	-	0.06	0.07
ARP/wARP → Buccaneer	0.19	0.3	0.05	0.08	0.05	0.07	0.25	0.34	0.07	0.1	0.06	0.08	0.26	0.42	0.08	0.12	0.07	0.1	0.32	0.44	0.09	0.13	0.09	0.11
ARP/wARP → Phenix AutoBuild(Parrot)	0.11	0.24	0.06	0.07	0.02	0.03	0.15	0.34	0.08	0.09	0.03	0.03	0.16	0.61	0.05	0.1	0.03	0.05	0.21	0.65	0.07	0.12	0.04	0.06
ARP/wARP → Phenix AutoBuild	0.1	0.23	0.06	0.07	0.02	0.03	0.15	0.33	0.07	0.09	0.03	0.03	0.23	0.6	0.07	0.1	0.04	0.05	0.32	0.64	0.09	0.12	0.05	0.05
Buccaneer	0.18	0.3	0.05	0.08	0.05	0.07	0.23	0.33	0.07	0.09	0.06	0.08	0.24	0.4	0.07	0.12	0.07	0.1	0.31	0.42	0.09	0.13	0.09	0.1
Buccaneer → ARP/wARP	0.06	0.17	0.05	0.08	0.02	0.03	0.15	0.37	0.07	0.11	0.03	0.03	0.27	0.62	0.09	0.19	0.04	0.05	0.38	0.75	0.12	0.21	0.05	0.06
Buccaneer → Phenix AutoBuild(Parrot)	0.16	0.28	0.05	0.07	0.03	0.05	0.21	0.32	0.06	0.08	0.04	0.06	0.1	0.33	0.04	0.1	0.04	0.07	0.14	0.35	0.06	0.11	0.05	0.08
Buccaneer → Phenix AutoBuild	0.15	0.28	0.05	0.07	0.04	0.05	0.2	0.31	0.06	0.08	0.05	0.06	0.2	0.33	0.06	0.1	0.05	0.07	0.28	0.35	0.07	0.11	0.07	0.08
Phenix AutoBuild(Parrot)	0.09	0.21	0.03	0.06	0.02	0.04	0.12	0.3	0.05	0.08	0.03	0.06	0.11	0.56	0.04	0.12	0.03	0.09	0.13	0.59	0.05	0.13	0.04	0.1
Phenix AutoBuild(Parrot) → ARP/wARP	0.04	0.16	0.04	0.08	0.02	0.02	0.09	0.37	0.06	0.11	0.03	0.03	0.18	0.71	0.07	0.2	0.03	0.04	0.29	0.79	0.1	0.22	0.05	0.05
Phenix AutoBuild(Parrot) → Buccaneer	0.17	0.27	0.05	0.07	0.05	0.06	0.22	0.32	0.07	0.09	0.06	0.07	0.13	0.31	0.05	0.1	0.04	0.07	0.17	0.33	0.07	0.11	0.06	0.08
Phenix AutoBuild → ARP/wARP	0.04	0.16	0.04	0.07	0.02	0.02	0.08	0.37	0.06	0.11	0.03	0.03	0.21	0.68	0.08	0.19	0.04	0.05	0.31	0.78	0.1	0.21	0.07	0.08
Phenix AutoBuild → Buccaneer	0.18	0.26	0.05	0.07	0.05	0.06	0.23	0.31	0.07	0.09	0.06	0.07	0.15	0.29	0.05	0.1	0.04	0.07	0.19	0.32	0.07	0.11	0.06	0.08
Phenix AutoBuild	0.09	0.21	0.03	0.05	0.03	0.04	0.12	0.3	0.04	0.08	0.03	0.05	0.16	0.55	0.05	0.12	0.05	0.09	0.26	0.58	0.08	0.14	0.09	0.1
SHELXE	0.14	0.18	-	-	0.02	0.03	0.2	0.26	-	-	0.03	0.03	0.18	0.28	-	-	0.03	0.04	0.23	0.4	-	-	0.03	0.06
SHELXE → ARP/wARP	0.17	0.23	0.06	0.08	0.03	0.03	0.26	0.41	0.08	0.11	0.04	0.04	0.22	0.37	0.08	0.12	0.05	0.06	0.32	0.55	0.11	0.17	0.06	0.07
SHELXE → Buccaneer	0.12	0.1	0.04	0.05	0.04	0.04	0.19	0.2	0.06	0.06	0.05	0.06	0.27	0.26	0.07	0.09	0.07	0.08	0.35	0.43	0.1	0.13	0.09	0.11
SHELXE → Phenix AutoBuild(Parrot)	0.06	0.06	0.03	0.03	0.02	0.03	0.08	0.09	0.03	0.04	0.03	0.03	0.13	0.15	0.05	0.06	0.04	0.05	0.19	0.28	0.06	0.08	0.06	0.07
SHELXE → Phenix AutoBuild	0.07	0.07	0.03	0.04	0.02	0.03	0.11	0.12	0.03	0.04	0.03	0.04	0.22	0.18	0.06	0.07	0.06	0.06	0.32	0.34	0.09	0.11	0.09	0.09
SHELXE(Parrot) → ARP/wARP	0.17	0.2	0.06	0.07	0.03	0.03	0.26	0.38	0.07	0.1	0.03	0.03	0.23	0.33	0.08	0.12	0.04	0.05	0.32	0.51	0.11	0.17	0.06	0.07
SHELXE(Parrot) → Buccaneer	0.11	0.11	0.04	0.05	0.03	0.04	0.17	0.21	0.05	0.07	0.05	0.06	0.21	0.26	0.06	0.09	0.05	0.08	0.3	0.42	0.08	0.13	0.07	0.11
SHELXE(Parrot) → Phenix AutoBuild(Parrot)	0.06	0.06	0.03	0.03	0.02	0.03	0.09	0.1	0.03	0.04	0.03	0.03	0.11	0.14	0.05	0.06	0.04	0.04	0.17	0.27	0.07	0.09	0.06	0.07
SHELXE(Parrot) → Phenix AutoBuild	0.06	0.07	0.02	0.03	0.02	0.03	0.11	0.12	0.03	0.04	0.03	0.04	0.21	0.13	0.06	0.05	0.06	0.04	0.29	0.26	0.08	0.08	0.08	0.07
SHELXE(Parrot)	0.11	0.14	-	-	0.02	0.02	0.16	0.21	-	-	0.02	0.03	0.17	0.25	-	-	0.03	0.04	0.22	0.37	-	-	0.03	0.05

## MR

Pipeline variant	MAE						RMSE					
	Completeness		R-free		R-work		Completeness		R-free		R-work	
	P	M	P	M	P	M	P	M	P	M	P	M
ARP/wARP	0.2	0.39	-	-	0.04	0.06	0.29	0.58	-	-	0.05	0.07
Buccaneer	0.15	0.29	0.04	0.07	0.04	0.07	0.2	0.37	0.06	0.11	0.05	0.09
Phenix AutoBuild	0.21	0.28	0.07	0.09	0.06	0.08	0.27	0.35	0.09	0.11	0.08	0.1
SHELXE	0.17	0.36	-	-	0.02	0.04	0.23	0.39	-	-	0.04	0.05

the lowest obtained for SHELXE; and the MAE for R-free is between 0.04-0.07. Finally, the RMSE of R-free/R-work are between 0.06–0.09 and 0.04–0.08, respectively.

Compared to the baseline zero-R predictive model (see Section 5.3.5), our predictive model achieved lower or much lower MAE and RMSE prediction errors for almost all the pipeline variants, types of data sets and protein structure evaluation measures, i.e., for 288 out of the 296 entries from Table 5.1. Notably, the predictions for recently PDB deposited experimental phasing data sets (which we did not use for the training of the predictive model) also have a much lower error for our predictive model than for the zero-R predictive model (Figure 5.3), with the exception of the predictions for SHELXE before Buccaneer and Phenix AutoBuild, for which the zero-R baseline model predictions achieve similar or marginally lower errors.

To evaluate the fitting of our predictive model, Figure 5.4 shows the difference of MAE and RMSE between training and testing for the JCSG experimental phasing

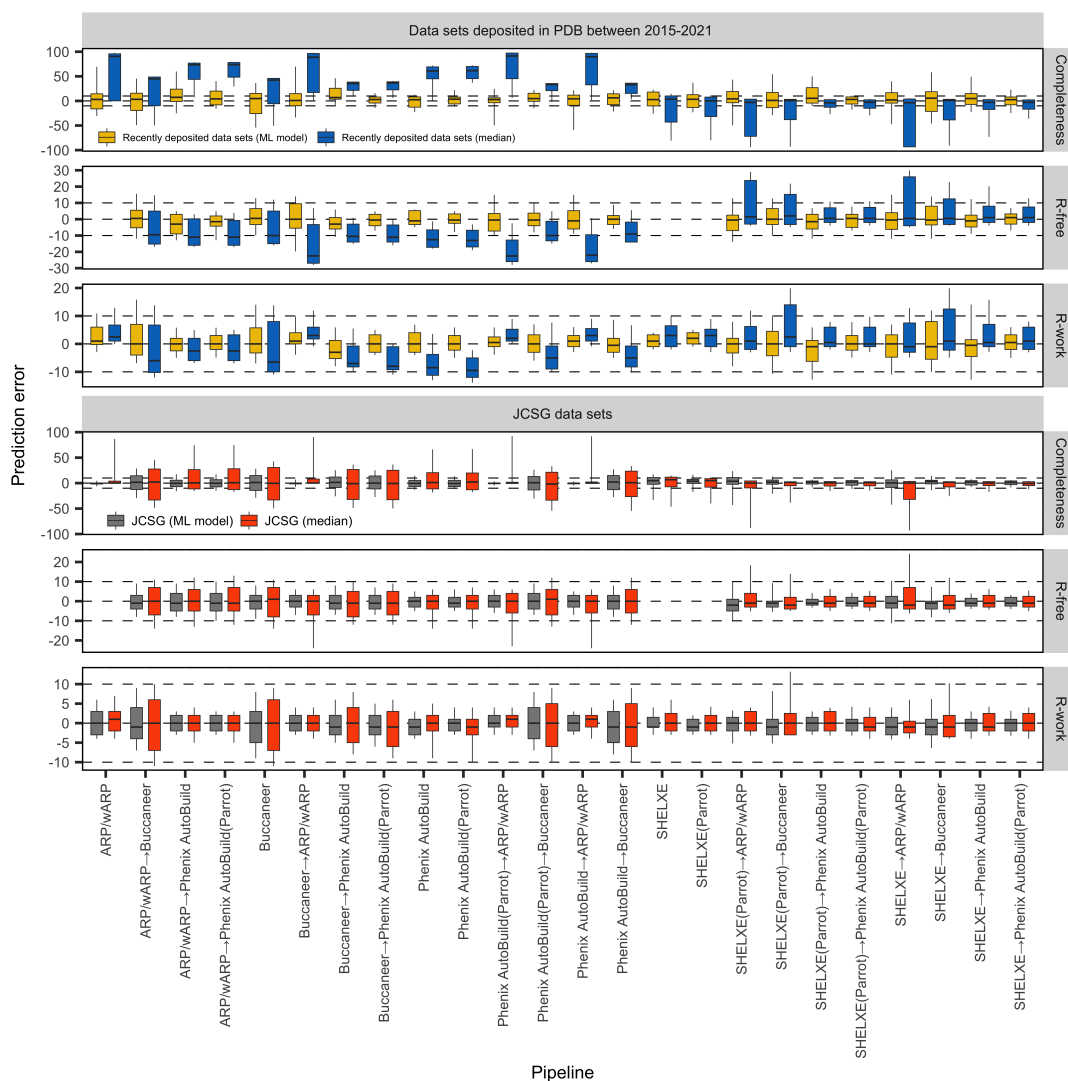


Figure 5.3: Prediction error for the ML predictive model and the median predictor for recently deposited and JCSG experimental phasing data sets.

and the MR data sets. The difference in MAE and RMSE between training and testing data sets for structure completeness is higher than in R-work/R-free for the JCSG experimental phasing and the MR data sets. When comparing the pipelines by structure completeness, Phenix AutoBuild and Buccaneer have the lowest error difference for the JCSG experimental phasing and the MR data sets, respectively. For R-work/R-free, the pipelines have a smaller difference in MAE and RMSE between the training and testing data sets compared to the structure completeness.

To further evaluate the accuracy of our predictive model, we analysed the mean and standard deviation (SD) of the predicted and actual protein structure evaluation measures for the crystallography data sets grouped based on their resolutions. Tables 5.2 and 5.3 show the result of this analysis for JCSG experimental phasing data

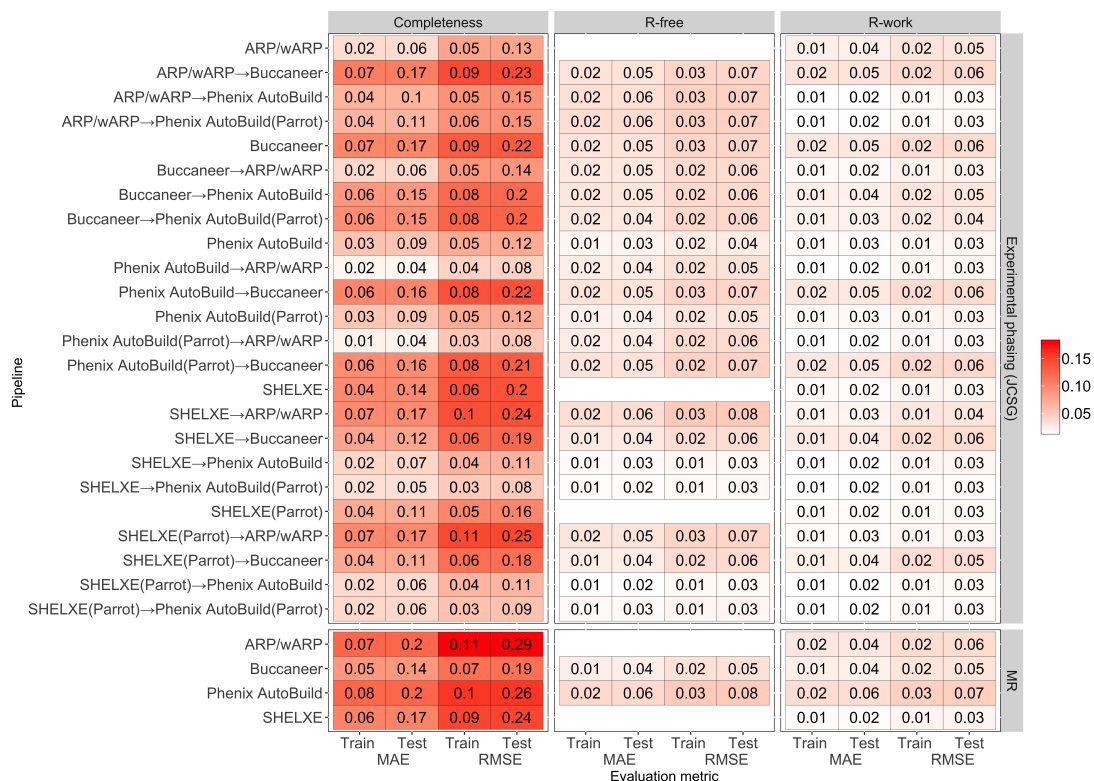


Figure 5.4: Mean absolute error (MAE) and root mean squared error (RMSE) of structure completeness and R-free/R-work for training and testing for the JCSG experimental phasing data sets and the MR data sets. The entries are shaded based on the magnitude of the difference in MAE and RMSE between the training and testing data sets.

sets for the pipeline variants without SHELXE and with SHELXE, respectively. For resolutions between 1.2 Å and 3.1 Å, the predicted and actual mean and SD values are very close for most pipeline variants. The spread of the predicted structure completeness for ARP/wARP run alone, and run after SHELXE has a higher SD compared to the completeness achieved when the pipelines were run in reality. At worse than 3.2 Å, the predicted R-free/R-work have mean and SD values close to the real results, while the predicted structure completeness has a larger difference in the SD and a smaller difference in the mean than the actual results.

Table 5.4 shows the results of the same analysis as above for the MR data sets. The mean of all the predicted structure evaluation measures as well as the SD values for the predicted R-free/R-work are close to the actual results. However, at resolutions better than 3.0 Å, the difference between the SD for the predicted and actual structure completeness is larger than that for R-free/R-work. At resolutions of 3.1 Å or worse, this difference decreased significantly.

Table 5.2: Mean and standard deviation (SD) of the real and predicted structure evaluation measures for the JCSG experimental phasing data sets grouped based on resolution, with the number of data sets in each group shown in brackets. The entries from the table are shaded based on the magnitude of the difference between the real (R) and predicted (P) results.

Pipeline variant	Structure evaluation	Resolution																																															
		1.2 - 3.1(39)						3.2(45)						3.4(41)						3.6(31)						3.8(43)						4.0+(42)																	
		mean			SD			mean			SD			mean			SD			mean			SD			mean			SD			mean			SD														
P	R		P	R		P	R		P	R		P	R		P	R		P	R		P	R		P	R		P	R		P	R		P	R															
ARP/wARP → Buccaneer	Completeness	0.87	0.89	0.15	0.19	0.64	0.6	0.17	0.34	0.56	0.54	0.17	0.29	0.46	0.49	0.14	0.29	0.28	0.34	0.12	0.24	0.15	0.16	0.08	0.18	0.32	0.31	0.04	0.07	0.40	0.41	0.05	0.09	0.42	0.42	0.04	0.07	0.44	0.43	0.04	0.08	0.48	0.47	0.03	0.07	0.50	0.50	0.03	0.06
	R-free	0.29	0.28	0.04	0.06	0.33	0.34	0.05	0.09	0.35	0.35	0.04	0.07	0.37	0.35	0.04	0.07	0.04	0.07	0.40	0.39	0.03	0.07	0.42	0.42	0.02	0.07	0.42	0.42	0.02	0.08	0.43	0.43	0.03	0.09	0.44	0.43	0.03	0.08	0.43	0.43	0.03	0.07	0.42	0.42	0.02	0.05	0.42	0.42
	R-work	0.9	0.91	0.11	0.11	0.35	0.36	0.08	0.22	0.28	0.29	0.06	0.18	0.19	0.17	0.06	0.16	0.11	0.12	0.04	0.11	0.07	0.07	0.03	0.06	0.27	0.26	0.04	0.05	0.39	0.39	0.02	0.07	0.39	0.38	0.03	0.06	0.42	0.42	0.02	0.08	0.43	0.43	0.03	0.09	0.44	0.43	0.03	0.08
ARP/wARP → PHENIX AutoBuild(Parrot)	Completeness	0.23	0.22	0.02	0.03	0.25	0.25	0.01	0.02	0.25	0.24	0.01	0.02	0.27	0.28	0.01	0.02	0.27	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03
	R-free	0.91	0.91	0.09	0.12	0.38	0.35	0.06	0.22	0.28	0.29	0.08	0.16	0.19	0.2	0.06	0.15	0.14	0.14	0.05	0.11	0.09	0.11	0.03	0.07	0.43	0.44	0.02	0.10	0.43	0.43	0.02	0.10	0.43	0.43	0.02	0.10	0.43	0.43	0.02	0.10	0.43	0.43	0.02	0.10				
	R-work	0.22	0.22	0.02	0.03	0.25	0.25	0.01	0.02	0.25	0.24	0.01	0.02	0.27	0.28	0.01	0.03	0.27	0.26	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03	0.26	0.27	0.01	0.03				
ARP/wARP	Completeness	0.83	0.77	0.21	0.32	0.08	0.09	0.07	0.16	0.03	0.03	0.03	0.05	0.01	0.01	0.01	0.03	0.01	0	0.01	0	0.01	0	0.01	0	0.01	0	0.01	0	0.01	0	0.01	0	0.01	0	0.01	0												
	R-free	0.29	0.29	0.06	0.08	0.49	0.49	0.03	0.08	0.51	0.50	0.03	0.06	0.52	0.51	0.02	0.06	0.54	0.54	0.02	0.07	0.55	0.55	0.02	0.04	0.55	0.55	0.02	0.04	0.55	0.55	0.02	0.04	0.55	0.55	0.02	0.04	0.55	0.55	0.02	0.04								
	R-work	0.23	0.23	0.01	0.03	0.20	0.20	0.01	0.03	0.20	0.20	0.01	0.02	0.19	0.19	0.02	0.03	0.19	0.19	0.02	0.03	0.19	0.19	0.02	0.03	0.19	0.19	0.02	0.03	0.19	0.19	0.02	0.03	0.19	0.19	0.02	0.03	0.19	0.19	0.02	0.03								
Buccaneer → ARP/wARP	Completeness	0.91	0.93	0.07	0.09	0.69	0.67	0.12	0.28	0.61	0.62	0.12	0.24	0.52	0.56	0.11	0.27	0.4	0.41	0.13	0.25	0.23	0.2	0.08	0.16	0.13	0.25	0.23	0.2	0.08	0.16	0.13	0.25	0.23	0.2	0.08	0.16												
	R-free	0.26	0.25	0.03	0.04	0.34	0.35	0.03	0.08	0.36	0.35	0.03	0.07	0.38	0.36	0.03	0.06	0.40	0.40	0.03	0.06	0.43	0.43	0.02	0.05	0.43	0.43	0.02	0.05	0.43	0.43	0.02	0.05	0.43	0.43	0.02	0.05												
	R-work	0.23	0.23	0.02	0.03	0.27	0.27	0.02	0.06	0.28	0.27	0.02	0.04	0.31	0.29	0.02	0.05	0.32	0.32	0.02	0.05	0.34	0.34	0.02	0.05	0.34	0.34	0.02	0.05	0.34	0.34	0.02	0.05	0.34	0.34	0.02	0.05												
Buccaneer → PHENIX AutoBuild(Parrot)	Completeness	0.91	0.93	0.07	0.08	0.72	0.67	0.11	0.27	0.6	0.62	0.14	0.23	0.54	0.56	0.14	0.28	0.41	0.43	0.14	0.25	0.23	0.22	0.08	0.16	0.14	0.25	0.23	0.22	0.08	0.16	0.14	0.25	0.23	0.22	0.08	0.16												
	R-free	0.26	0.25	0.03	0.04	0.34	0.34	0.02	0.08	0.37	0.35	0.04	0.06	0.38	0.36	0.03	0.07	0.40	0.40	0.03	0.06	0.43	0.44	0.02	0.04	0.43	0.44	0.02	0.04	0.43	0.44	0.02	0.04	0.43	0.44	0.02	0.04												
	R-work	0.23	0.23	0.02	0.03	0.27	0.27	0.02	0.06	0.28	0.28	0.03	0.05	0.31	0.29	0.02	0.05	0.32	0.32	0.02	0.05	0.34	0.35	0.02	0.05	0.34	0.35	0.02	0.05	0.34	0.35	0.02	0.05	0.34	0.35	0.02	0.05												
Buccaneer → PHENIX AutoBuild	Completeness	0.83	0.86	0.15	0.19	0.63	0.61	0.16	0.32	0.56	0.53	0.15	0.28	0.47	0.48	0.11	0.28	0.33	0.37	0.12	0.26	0.19	0.17	0.08	0.17	0.12	0.26	0.19	0.17	0.08	0.17	0.12	0.26	0.19	0.17														
	R-free	0.33	0.32	0.05	0.07	0.40	0.41	0.04	0.09	0.42	0.43	0.04	0.08	0.45	0.43	0.02	0.07	0.47	0.46	0.03	0.08	0.50	0.50	0.03	0.05	0.50	0.50	0.03	0.05	0.50	0.50	0.03	0.05																
	R-work	0.30	0.29	0.04	0.06	0.34	0.34	0.04	0.09	0.35	0.36	0.04	0.07	0.37	0.36	0.03	0.07	0.39	0.38	0.03	0.07	0.39	0.38	0.03	0.07	0.39	0.38	0.03	0.07	0.39	0.38	0.03	0.07																
PHENIX AutoBuild(Parrot) → ARP/wARP	Completeness	0.9	0.89	0.15	0.15	0.05	0.08	0.05	0.16	0.03	0.02	0.03	0.04	0.01	0.01	0.01	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
	R-free	0.28	0.28	0.04	0.05	0.50	0.49	0.03	0.07	0.50	0.50	0.03	0.05	0.52	0.51	0.02	0.07	0.53	0.53	0.02	0.07	0.53	0.53	0.02	0.07	0.53	0.53	0.02	0.07	0.53	0.53	0.02	0.07																
	R-work	0.22	0.22	0.01	0.03	0.21	0.21	0.01	0.02	0.21	0.20	0.01	0.02	0.19	0.20	0.02	0.02	0.19	0.19	0.01	0.04	0.19	0.21	0.02	0.03	0.19	0.21	0.02	0.03	0.19	0.21	0.02	0.03																
PHENIX AutoBuild(Parrot) → Buccaneer	Completeness	0.92	0.94	0.09	0.09	0.7	0.66	0.13	0.28	0.63	0.61	0.13	0.23	0.54	0.53	0.12	0.28	0.42	0.44	0.13	0.28	0.28	0.29	0.1	0.22	0.13	0.28	0.28	0.29	0.1	0.22	0.13	0.28																
	R-free	0.31	0.30	0.03	0.04	0.38	0.40	0.04	0.08	0.40	0.41	0.03	0.07	0.42	0.42	0.03	0.09	0.45	0.46	0.03	0.08	0.48	0.48	0.03	0.05	0.48	0.48	0.03	0.05	0.48	0.48	0.03	0.05																
	R-work	0.28	0.27	0.03	0.04	0.32	0.33	0.04	0.08	0.33	0.34	0.03	0.06	0.35	0.35	0.03	0.08	0.37	0.36	0.03	0.08	0.39	0.39	0.03	0.05	0.39	0.39	0.03	0.05	0.39	0.39	0.03	0.05																
PHENIX AutoBuild(Parrot)	Completeness	0.91	0.92	0.08	0.07	0.37	0.38	0.06	0.15	0.32	0.32	0.06	0.13	0.26	0.26	0.06	0.13	0.18	0.18	0.04	0.12	0.12	0.12	0.03	0.07	0.12	0.12	0.03	0.07	0.12	0.12	0.03	0.07																
	R-free	0.27	0.26	0.04	0.04	0.41	0.41	0.01	0.04	0.41	0.41	0.01	0.04	0.42	0.42	0.01	0.05	0.43	0.44	0.01	0.06	0.44	0.44	0.02	0.04	0.44	0.44	0.02	0.04	0.44	0.44	0.02	0.04																
	R-work	0.23	0.23	0.02	0.03	0.33	0.33	0.01	0.03	0.33	0.33	0.01	0.03	0.34	0.34	0.01	0.03	0.35	0.35	0.01	0.04	0.36	0.35	0.01	0.04	0.36	0.35	0.01	0.04	0.36	0.35	0.01	0.04																
PHENIX AutoBuild → ARP/wARP	Completeness	0.87	0.89	0.14	0.14	0.07	0.07	0.05	0.15	0.02	0.02	0.02	0.04	0.01	0.01	0.01	0.02	0	0	0	0.01	0	0	0	0.01	0	0	0	0	0	0																		
	R-free	0.29	0.28	0.04	0.06	0.50	0.50	0.03	0.07	0.51	0.51	0.03	0.04	0.52	0.50	0.03	0.06	0.53	0.53	0.02	0.06	0.52	0.53	0.03	0.06	0.52	0.53	0.03	0.06	0.52	0.53	0.03	0.06																
	R-work	0.23	0.22	0.02	0.03	0.21	0.21	0.01	0.03	0.21	0.20	0.01	0.02	0.19	0.19	0.02	0.03	0.19	0.19	0.01	0.04	0.19	0.20	0.02	0.03	0.19	0.20	0.02	0.03	0.19	0.20	0.02	0.03																
PHENIX AutoBuild → Buccaneer	Completeness	0.9	0.93	0.09	0.09	0.7	0.69	0.13	0.26	0.59	0.61	0.13	0.27	0.54	0.56	0.13	0.25	0.42	0.45	0.14	0.27	0.19	0.29	0.11	0.24	0.14	0.27	0.19	0.29	0.11	0.24																		
	R-free	0.31	0.30	0.03	0.05	0.39	0.39	0.04	0.08	0.41	0.40	0.03	0.07	0.43	0.42	0.03	0.07	0.45	0.46	0.03	0.08	0.48	0.48	0.03	0.05	0.48	0.48	0.03	0.05	0.48	0.48	0.03	0.05																
	R-work	0.29	0.27	0.03	0.04	0.32	0.32	0.03	0.07	0.34	0.34	0.03	0.07	0.35	0.34	0.03	0.07	0.37	0.36	0.03	0.07	0.39	0.39	0.03	0.07	0.39	0.39	0.03	0.07	0.39	0.39	0.03	0.07																
PHENIX AutoBuild	Completeness	0.9	0.91	0.07	0.09	0.38	0.39	0.05	0.15	0.32	0.32	0.07	0.12	0.26	0.25																																		



Table 5.4: Mean and standard deviation (SD) of the real and predicted structure evaluation measures for the MR data sets grouped based on resolution, with the number of data sets in each group shown in brackets. The table entries are shaded based on the difference between the real (R) and predicted (P) results.

Pipeline variant	Structure evaluation	Resolution																			
		1.0 - 1.5 (65)		1.6 - 2.0 (65)				2.1 - 2.5 (50)				2.6 - 3.0 (55)		3.1+ (31)							
		mean	SD	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD								
		P	R	P	R	P	R	P	R	P	R	P	R	P	R						
ARP/wARP	Completeness	0.65	0.63	0.31	0.46	0.47	0.51	0.31	0.46	0.32	0.37	0.28	0.43	0.16	0.18	0.18	0.34	0.04	0.02	0.03	0.04
	R-work	0.25	0.25	0.03	0.05	0.27	0.27	0.03	0.06	0.30	0.29	0.05	0.08	0.29	0.29	0.04	0.07	0.31	0.31	0.05	0.08
Buccaneer	Completeness	0.46	0.47	0.26	0.38	0.43	0.45	0.26	0.37	0.33	0.35	0.25	0.32	0.27	0.3	0.19	0.29	0.18	0.15	0.13	0.16
	R-free	0.45	0.44	0.08	0.11	0.46	0.46	0.07	0.10	0.50	0.49	0.06	0.08	0.51	0.51	0.04	0.06	0.53	0.53	0.02	0.04
	R-work	0.42	0.41	0.08	0.10	0.42	0.41	0.07	0.10	0.43	0.43	0.06	0.08	0.43	0.42	0.04	0.06	0.44	0.44	0.02	0.04
PHENIX AutoBuild	Completeness	0.72	0.72	0.13	0.3	0.71	0.74	0.14	0.29	0.58	0.56	0.19	0.38	0.52	0.55	0.18	0.33	0.38	0.42	0.17	0.27
	R-free	0.29	0.29	0.04	0.10	0.32	0.31	0.05	0.10	0.38	0.38	0.06	0.12	0.39	0.39	0.05	0.09	0.43	0.42	0.05	0.08
	R-work	0.27	0.27	0.03	0.09	0.28	0.28	0.05	0.09	0.33	0.33	0.06	0.10	0.32	0.32	0.04	0.08	0.36	0.35	0.05	0.06
SHLXEXE	Completeness	0.84	0.85	0.11	0.25	0.71	0.67	0.15	0.35	0.38	0.34	0.19	0.31	0.2	0.2	0.09	0.18	0.12	0.13	0.07	0.13
	R-work	0.46	0.45	0.01	0.03	0.47	0.48	0.02	0.04	0.51	0.52	0.02	0.06	0.53	0.52	0.01	0.03	0.53	0.52	0.01	0.02
		0.0		0.2		0.0		0.06													

prediction error was classified in the first group for each protein structure in our testing data set. For the JCSG experimental phasing data set, 85%, 94% and 91% of the pipelines with the lowest prediction error were classified in the first group for structure completeness, R-free and R-work, respectively. For the MR data set the percentages were 60%, 69% and 87% respectively.

Figure 5.5 shows the inference time of the predictive model for individual pipelines and pipeline combinations for the JCSG experimental phasing and the MR data sets. The inference time is the total of time to predict the structure completeness and R-free/R-work. The SHLXEXE variants for the JCSG experimental phasing data set and ARP/wARP and Buccaneer for the MR data set have the lowest inference time.

### 5.4.3 Evaluation of recommended pipeline variant

To further evaluate our predictive model, we analysed the potential benefits of using the pipeline variant recommended by the model, i.e., the pipeline variant predicted to achieve the best completeness or R-free/R-work for each of the data sets.

To this end, we first analysed the time savings that can be achieved by using the recommended pipeline variant instead of running all the pipeline variants in order to obtain the best possible structure. Figure 5.6 shows the total execution time when running all the pipeline variants and when only the pipeline recommended by our predictive model was run. The time saved (on the powerful high-performance cluster mentioned in Section 5.3.5) was up to 20 hours for a small protein structure, and up to 60 hours for large structures. When these pipeline variants were ran in parallel



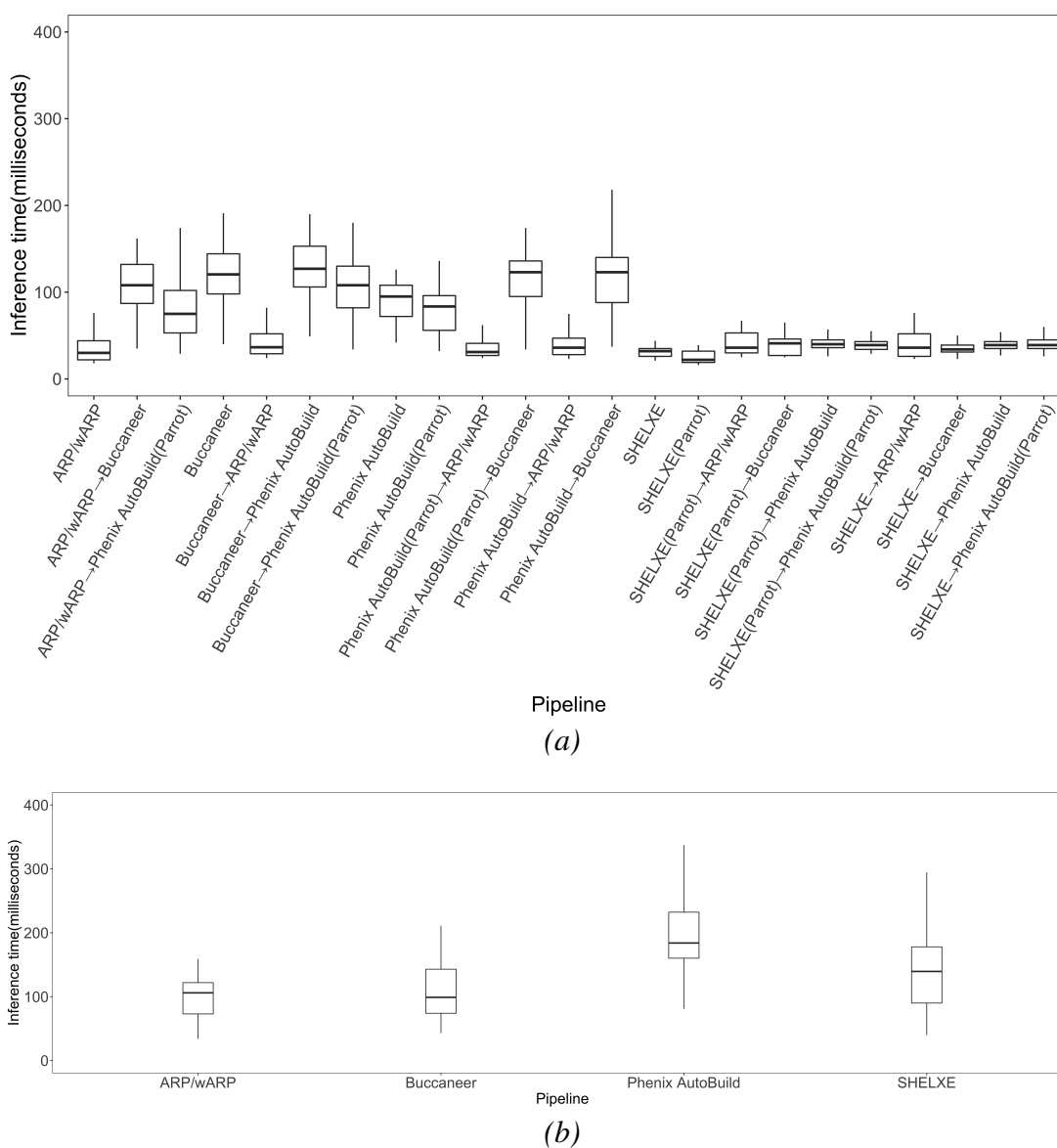


Figure 5.5: Inference time for the predictive model for individual pipelines and pipeline combinations. For each data set of the JCSG experimental phasing and the MR data sets, the inference time is the total time of predicting the structure completeness, R-free, and R-work. (a) Inference time for the JCSG experimental phasing. (b) Inference time for the MR data sets.

on our high-performance cluster, this time saving was reduced; however, running the recommended pipeline still saved up to 30 hours when building large structures.

Next, we analysed how close the completeness and R-free/R-work of the protein structure built by the recommended pipeline variant was to the best completeness and R-free/R-work value achievable by running all the pipeline variants. Figures 5.7 and 5.8 present the results of this analysis for the JCSG experimental phasing and MR data sets, respectively. These results show that the recommended pipeline variant built

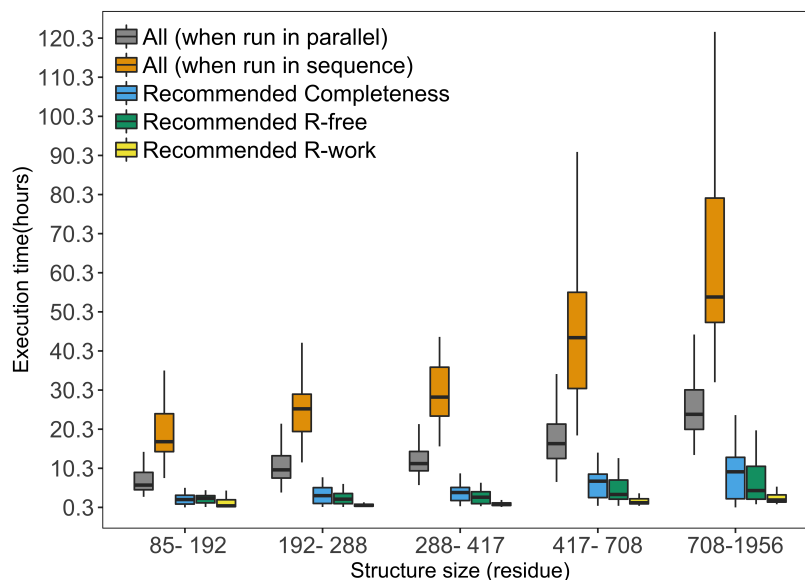


Figure 5.6: Execution time required to run all the pipeline variants (in parallel, and in sequence) versus the execution time required to run the pipeline recommended by the predictive model (for best completeness, best R-free, and best R-work), for the JCSG experimental phasing data sets.

protein structures with completeness, R-free and R-work within only 1% of those of the best pipeline for 32%, 50% and 59% of the JCSG experimental phasing data sets; and 70%, 99% and 71% of the MR data sets, respectively; and within only 5% of the best pipeline for 52%, 78% and 93% of the JCSG experimental phasing data sets; and 83%, 100% and 87% of the MR data sets, respectively.

Finally, for each of the 15 research papers that we could find for our testing MR data sets and that mentioned the pipeline used to build the protein structure, we compared the pipeline used in the paper to the pipeline variant recommended by our predictive model. To ensure a fair comparison, we ran the pipeline used in the paper and the pipeline recommended by our predictive model using the same search model to obtain initial phases for each structure. This search model could not be the same as the one used for the PDB deposited structure, which is unavailable.

Table 5.5 presents the structure completeness achieved by the pipeline that was chosen to solve the protein structure when deposited in the PDB compared to the completeness achieved by our recommended pipeline for each of these MR data sets. As shown in this table, our recommended pipeline achieved better completeness than the other pipeline for 10 out of the 15 protein structures, and identical completeness for three additional structures for which the predictive model recommended the

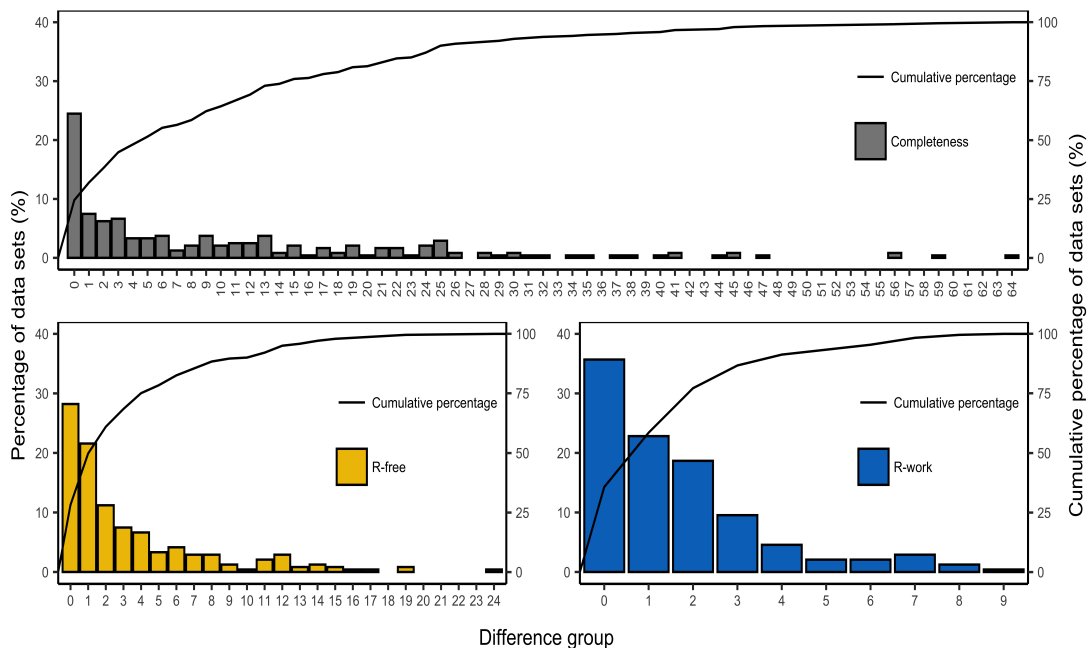


Figure 5.7: Difference between the best completeness, R-free and R-work achieved by running all the pipeline variants and running the recommended pipeline variant for the JCSG experimental phasing data sets. The percentage of the data sets for each difference group is shown on the left side and the cumulative percentage on the right side.

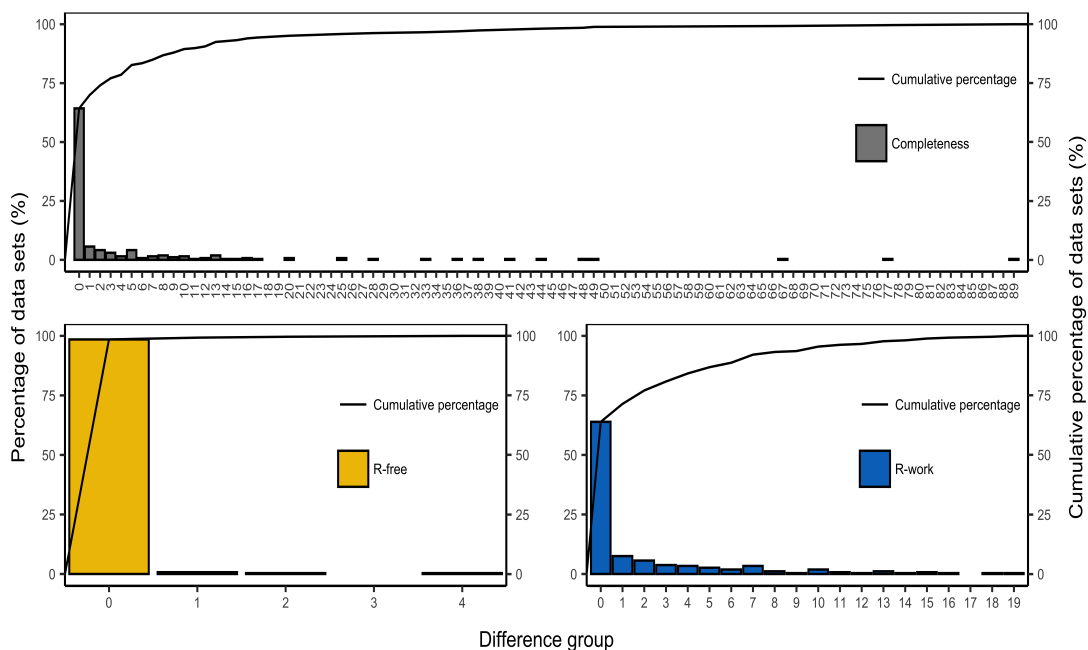


Figure 5.8: Difference between the best completeness, R-free and R-work achieved by running all the pipeline variants and running the recommended pipeline variant for the MR data sets. The percentage of the data sets for each difference group is shown on the left side and the cumulative percentage on the right side.

Table 5.5: Real structure completeness achieved by the pipeline that was used to solve the protein structure when deposited in the PDB and by the pipeline recommended by the predictive model, for the MR data sets.

PDB ID	Used		Recommended		
	Pipeline variant	Real completeness	Pipeline variant	Real completeness	Predicted completeness
4YVO	ARP/wARP	98.32	SHELXE	97.48	91.45
3MOK	ARP/wARP	97.8	ARP/wARP	97.8	93.75
2VMH	ARP/wARP	97.28	SHELXE	97.96	91.57
5N13	ARP/wARP	97.22	SHELXE	100.0	90.07
5WRT	PHENIX AutoBuild	93.58	PHENIX AutoBuild	93.58	60.62
5Y3D	PHENIX AutoBuild	83.03	PHENIX AutoBuild	83.03	70.88
6GP5	ARP/wARP	64.8	PHENIX AutoBuild	88.48	84.55
3TZE	Buccaneer	57.63	PHENIX AutoBuild	85.16	75.43
4D70	Buccaneer	40.62	PHENIX AutoBuild	80.31	83.62
5IXG	PHENIX AutoBuild	33.63	SHELXE	76.59	79.82
2XSJ	Buccaneer	10.22	PHENIX AutoBuild	0.11	44.58
6FDH	ARP/wARP	0.45	SHELXE	91.82	56.39
2CQT	ARP/wARP	0.06	PHENIX AutoBuild	81.2	34.73
5ZWP	ARP/wARP	0	SHELXE	94.69	76.69
3P7Y	ARP/wARP	0	SHELXE	94.21	81.96

same pipeline as the one used to build the PDB structure. The recommended pipeline achieved worse completeness for only two of the 15 proteins structures (with a drop in completeness of under 1% for one of these).

## 5.5 Discussion

We have presented a predictive model of the performance of four widely used protein model-building pipelines and of their pairwise combinations. We have separately trained this predictive model for both experimental phasing and molecular replacement data sets, and for three commonly used structure evaluation measures. With this predictive model, we aim to help users choose the best pipeline for solving their protein structure based on the features of their starting data, to encourage them to use pipelines which may be less familiar to them, and to increase the joint use of multiple pipelines, as doing so is likely to yield a more complete and more refined structure.

The features were calculated in scale dependent measures (the scaling of the data is contingent on the output of the data reduction program used); however, the scale independent measures are more natural in crystallographic contexts. The scale dependent measures were implemented first, yielding almost indistinguishable results. We assume that this is due to the machine learning model effectively factoring out scale internally.

The MAE and RMSE analysis showed that R-free and R-work are more predictable than structure completeness in both experimental phasing and MR data sets. This unpredictability differs between the pipeline variants, suggesting that the electron-density map features have different effects on the pipelines' performance. The predictability of pipelines involving PHENIX Autobuild tends to be higher, which is likely due to the use of multiple models to offset stochastic effects. Both MAE and RMSE for our predictive model are significantly lower than the MAE and RMSE for the training data sets median used by the baseline, zero-R predictive model.

When comparing the individual data sets by using the mean and SD for the real and predicted structure evaluation measures, at a high resolution, which is considered an easier case, the performance of the pipelines is more predictable than at a low resolution. When the data sets become worse in terms of resolution (which typically also means that the phases become worse), the difference in SD between the real and predicted results becomes larger.

The pipeline variant predicted to build the best protein structure frequently produced structures with the same or similar completeness and/or R-free/R-work to the best pipeline variant. Moreover, using the pipeline variant recommended by our predictive model save days of pipeline execution time on high-spec computers, and the time saved increases when the protein structure is larger. Finally, the predictive model can be used to try massive search models in MR cases, enabling the selection of good initial phases [99, 100].

Future work will consider a multi-task method for predicting structure completeness, R-free and R-work, and will combine the ML models into a single model. We envisage that this could lead to more accurate predictions and to better pipeline ranking. Moreover, we will explore additional ML algorithms, e.g. XGBoost [101], as this may improve our predictive model.

## 5.6 Availability

We implemented the predictive model described in the paper as a web application that is publicly available and free to use at <http://www.robin-predictor.org>. The source code for the application is available at <https://doi.org/10.15124/>

ee9d169f-c34b-44f2-8c75-3b68e7cd68a8.

# Identifying incorrect fragments to improve backbone chain tracing using neural network in Buccaneer

In this chapter, we introduce a neural network trained to identify incorrect fragments during the protein model building process. We start with a presentation of the method used to label the training data sets, and of the neural network training process. We then describe how the neural network can be integrated into the protein model building process of the Buccaneer software. Finally, we systematically evaluate the performance of the Buccaneer variant that uses our neural network to avoid the use of incorrect fragments in the protein structures it builds.

## 6.1 Abstract

Tracing the backbone in protein model-building is a critical step, as incorrect tracing leads to poor protein models. Here, we present a neural network trained to identify incorrect fragments and remove them from the model building process in order to improve backbone tracing. Our neural network was tested on experimental phasing data sets from the Joint Center for Structural Genomics (JCSG), recently deposited experimental phasing data sets (from 2015–2021), and molecular replacement data sets. Our experimental results show that using the neural network in the Buccaneer protein model building software can produce significantly more complete protein models than those built using Buccaneer alone. In particular, Buccaneer with the neural network built protein models with completeness at least 5% higher for 25% and 47% of the

original and truncated resolution JCSG experimental phasing data sets, respectively, for 26% of the recently collected experimental phasing data sets, and for 16% of the molecular replacement data sets.

## 6.2 Introduction

A key step in building a protein model is tracing the backbone (i.e., its longest chain). Model-building pipelines such as ARP/wARP [14, 16] and Buccaneer [18] start their model building by finding the protein structure backbone. The procedure used to find the longest chain can yield wrong tracing because of choosing residues that are incorrectly placed. We examined this problem by modifying the growing step of the Buccaneer model building process so that at the long fragments obtained at the end of the stage were each split into their constituent small fragments (i.e., into fragments with three residues) and then removing one small fragment at a time before the next (i.e., tracing) step of the model building. The protein structure built without each of these small fragments was evaluated against the deposited structure. Identifying and removing incorrect small fragments improves the protein structure, because such fragments break up some paths and force the tracing algorithm to change its direction away from the correct trace.

A protein structure may have hundreds or even thousands of small fragments. As such, removing one small fragment at a time to assess its correctness requires a huge amount of computation even for one building cycle. All widely used protein model-building pipelines (ARP/wARP [14, 16], Buccaneer [18, 19, 28], PHENIX AutoBuild [20, 86] and SHELXE [21, 22, 23, 24]) are iterative, which makes using this simple method unfeasible.

To address this problem, we developed a neural network model that identifies incorrect small fragments and can be used to efficiently eliminate them from the protein model building before the backbone tracing step. The neural network predicts the probabilities that these small fragments are incorrect based on fragment features calculated from the electron-density map and the protein model geometry. We show that backbone tracing is significantly improved by eliminating the small fragments whose probabilities are below a certain threshold as those classified as incorrect small frag-



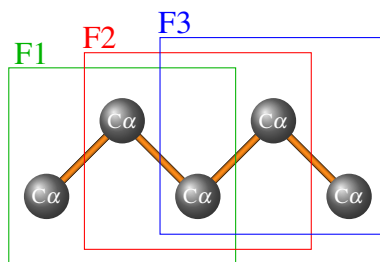


Figure 6.1: An example of splitting a fragment into small fragments. The fragment split into three small fragments.

ments.

## 6.3 Method

### 6.3.1 Creating the training data sets

We used molecular replacement (MR) data sets containing 1351 protein structures [26] to create the training data sets for the neural network. These MR data sets have resolution ranges from 1.0 Å to 3.5 Å. For each protein structure, we ran the finding and growing steps in Buccaneer [18]. The output of the growing step is a set of overlapped fragments that have different lengths. Each fragment was split into three-residues fragments, which we called *small fragments* (Figure 6.1). All these small fragments were saved into a CIF file. The procedure for labelling each of the small fragments as either “correct” or “incorrect” is as follows:

1. Run Buccaneer for one building cycle starting from the joining step, in order to build a protein structure from all the small fragments, and compare the built structure to the deposited structure to compute the structure completeness ((as described in Chapter 3) ). This structure and its completeness provide a baseline for the later steps of our solution.
2. Omit one small fragment at a time and build the protein structure as in step 1.
3. Compare the completeness of the protein structure obtained in step 2 to that of the baseline structure from step 1.
4. If the structure from step 2 has higher structure completeness, label the omitted small fragment as “incorrect”.

5. Repeat steps 2, 3 and 4 for the rest of the small fragments, removing from the model building process all the small fragments identified as “incorrect” in step 4.

As an additional step, we examine whether the small fragments not removed by the procedure above were actually included in the protein structure. There are two reasons why a small fragment may not be included in the structure, and thus its removal would have no impact on the structure completeness. Thus, Buccaneer is not using small fragments that cannot be combined into chains of at least six residues (which is the minimum length set in Buccaneer for tracing) nor appended as a small branch to a long fragment. These small fragments are also labelled as “incorrect”. Finally, we labelled as “correct” all the small fragments not labelled “incorrect” after this additional step.

Using the procedure above, we labelled the small fragments in 1132 protein structures of the MR data sets, producing 822,366 correct and 299,577 incorrect small fragments. A number of protein structures were not used for the following reasons, with the number of omitted protein structures reported in parentheses:

1. Protein structures with more than 2856 small fragments, as this is the highest number of chains that can be saved in a CIF file with a unique ID of two characters (172 protein structures).
2. Protein structures for which no incorrect small fragments were found using our procedure (22 protein structures).
3. Protein structures that had a very large number of small fragments, and the identification of the incorrect small fragments could not be completed within 48 hours, which is the maximum time we allocated for processing each protein structure (25 protein structures).

### 6.3.2 Features of small fragments

Table 6.1 shows the features used in training the neural network in addition to the electron-density map resolution. The following sections describe each of these features.

Table 6.1: Features used in training the neural network in addition to the electron-density map resolution. Mean, SD, highest and lowest were calculated for the features when applicable, and each was used as a separate feature.

Feature	Mean	SD	Highest	Lowest	Categorical values (0 or 1)	Single value
Ramachandran angles in favoured regions					✓	
Ramachandran angles in allowed regions					✓	
Local likelihood score (LLK)	✓	✓	✓	✓		
Density score	✓	✓		✓		
Root mean square deviation (RMSD)						✓
Is a small fragment in the start of a chain?					✓	
Is a small fragment in the middle of a chain?					✓	
Is a small fragment in the end of a chain?					✓	

### 6.3.2.1 Ramachandran angles

A residue is classified in either favoured or allowed regions based on the probability densities of Phi ( $\phi$ ) and Psi ( $\psi$ ) [33]. When the probability densities of ( $\phi$ ) and ( $\psi$ ) is greater than  $0.01 \text{ rad}^{-2}$  or  $0.0005 \text{ rad}^{-2}$ , the residue is classified either in favoured or allowed regions, respectively [102, 18].

### 6.3.2.2 Log likelihood score

The log likelihood score (LLK), also known as the density-likelihood function, is a score of possible C-alpha group positions that reflects the reproducibility of the density features of real C-alpha groups in a simulated electron-density map for a known structure and it can be calculated as follow:

$$\log P(F | \rho) = \sum_x \log P[F | \rho(x)] = \sum_x - \left\{ \frac{[\rho(x) - \rho''(x')]^2}{2\sigma''(x')^2} \right\} + c \quad (6.1)$$

where  $F$  represent the electron density of correct C-alpha group position and orientation, and  $x$  is the coordinate in the observed density map while  $x'$  is the coordinate in the search fragment map rotated and translated to a give position and orientation in the observed map[103, 18].

### 6.3.2.3 Density score

The mean of electron-density for each residue in the small fragment, and the electron-density here is calculated for the only main residue's chain.

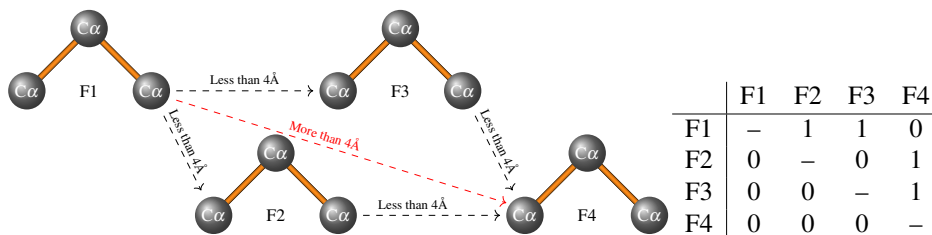


Figure 6.2: An example of four small fragments (F) and the distance between them. The matrix shows when two small fragments can be joined when the distance between them less than  $4\text{\AA}$ .

### 6.3.2.4 Root mean square deviation

We use the root mean square deviation (RMSD) between the small fragment and best matching fragment from the Top 500 well-refined protein structure database [104].

### 6.3.2.5 Small fragment position

Another feature used by the neural network is a categorical measure that distinguishes between small fragments located at the start, middle or end of a chain. We determine the value for this feature for a small fragment by measuring the distance between the fragment and the surrounding small fragments within a  $4\text{\AA}$  radius. Figure 6.2 shows an example of four small fragments and their associated *joining matrix*. The matrix element in row  $i$  and column  $j \neq i$  of this matrix is 1 if the distance between fragments  $F_i$  and  $F_j$  is less than  $4\text{\AA}$ , and fragment  $F_j$  is to the right of  $F_i$  (meaning that  $F_i$  can be followed by  $F_j$  in a chain); otherwise, this matrix element is zero. The small fragments that have zeros in their corresponding columns can only be at the start of a chain, and those with only zeros in their corresponding rows can only be at the end of a chain. All other small fragments are middle fragments.

## 6.3.3 Neural network architecture and training

### 6.3.3.1 Data set preparation

The sets of correct and incorrect small fragments from Section 6.3.1 were split into a training data set (containing 78.97% of the correct, and 79.26% of the incorrect fragments) and a validation data set (containing the remaining fragments). We normalised both the training data set and the validation data set by using z-score normalisation,

which is a standard practice in machine learning. This normalisation ensures that the features used to train and validate the neural network have zero mean and unit standard deviation. To this end, the mean and standard deviation of every feature is calculated for the data set undergoing normalisation, and the value of each data sample feature is adjusted by subtracting from it the mean and dividing the result by the standard deviation.

### 6.3.3.2 Neural network architecture

The input of the neural network model is a  $2849 \times 14$  array. The 2849 rows correspond to the largest number of small fragments across all the protein structures from the training and validation data sets, and the 14 columns correspond to the 14 small-fragment features that we used. The output of the neural network model is a probability of the small fragment being correct, and ranges from 0 to 1. The neural network was implemented using the Keras framework version 2.3.1 [105].

Because the number of small fragments differ between protein structures, the first layer in the neural network model is a masking layer. This layer uses a mask value of -1 for the rows from the input array for which no corresponding small fragment is available for a protein structures, ensuring that the neural network disregards these rows.

The hidden layers contained five long short-term memory (LSTM) layers with 512 neurons in the first hidden layer and reduced in geometric sequence to 32 neurons in the last hidden layer [68]. A sigmoid function was used in the output layer, and binary cross-entropy was used for the loss function [106].

### 6.3.3.3 Neural network training

The training of the neural network was carried out using an NVIDIA Tesla V100 32GB SXM2 GPU server. The maximum number of epochs was set to 1000, with early stopping when the Area Under the Curve (AUC) did not increasing for ten successive epochs. The Adam optimizer [61] was used, and the learning rate was set to 0.005. To evaluate the performance of the neural network model, we used the AUC and loss function.

To evaluate feature importance, we used permutation feature importance [56],

which involves shuffling the values of each feature, evaluating the neural network model obtained for the shuffled feature values, and comparing it to the baseline model (the model where the values of the features are not shuffled). As shuffling the values of the features disconnected the association with the true label, the change from the baseline model in the evaluation metrics showed the feature importance.

### 6.3.4 Using the neural network in Buccaneer

The neural network model weights and biases from Section 6.3.3 were extracted and saved into a CSV file, and C code was then generated for the neural network model by using the Keras2c library [107]. The C code was converted to C++ code for use in Buccaneer. As part of this work, the Keras2c library was extended to support the masking layer. The results from the Keras2c library were validated against the Keras Python framework.

As shown in Figure 6.3, the neural network model is used in the joining step of Buccaneer, after the fragments built by Buccaneer in earlier steps are split into small fragments, and before Buccaneer performs its tracing substep. The role of the neural network is to partition the set of small fragments into a subset of “correct” fragments for use in the tracing substep, and a subset of “incorrect” fragments that are disregarded (i.e., not used for this tracing). To that end, a threshold is applied to the outputs of the neural network, such that small fragments are deemed “correct” if their associated neural-network outputs (i.e., estimate probabilities of being “correct”) are above this threshold. To improve the likelihood of producing a good protein model, multiple thresholds are used to generate a small set of such models, and a decision tree developed by our project is employed to select the best of these models at the end of the Buccaneer model building cycle.

Two mechanisms for determining the thresholds were developed; the first mechanism is to set a fixed number of thresholds (e.g. ten thresholds) to divide the probabilities range into equal intervals. The second mechanism is to use Freedman–Diaconis rule to determine the number of the thresholds based on the probability distribution [108]. A model will be built for each threshold by eliminating the small fragments that have probabilities lower than this threshold. Moreover, we run either one or two Buccaneer confirmation building cycles to estimate how this protein structure will evolve

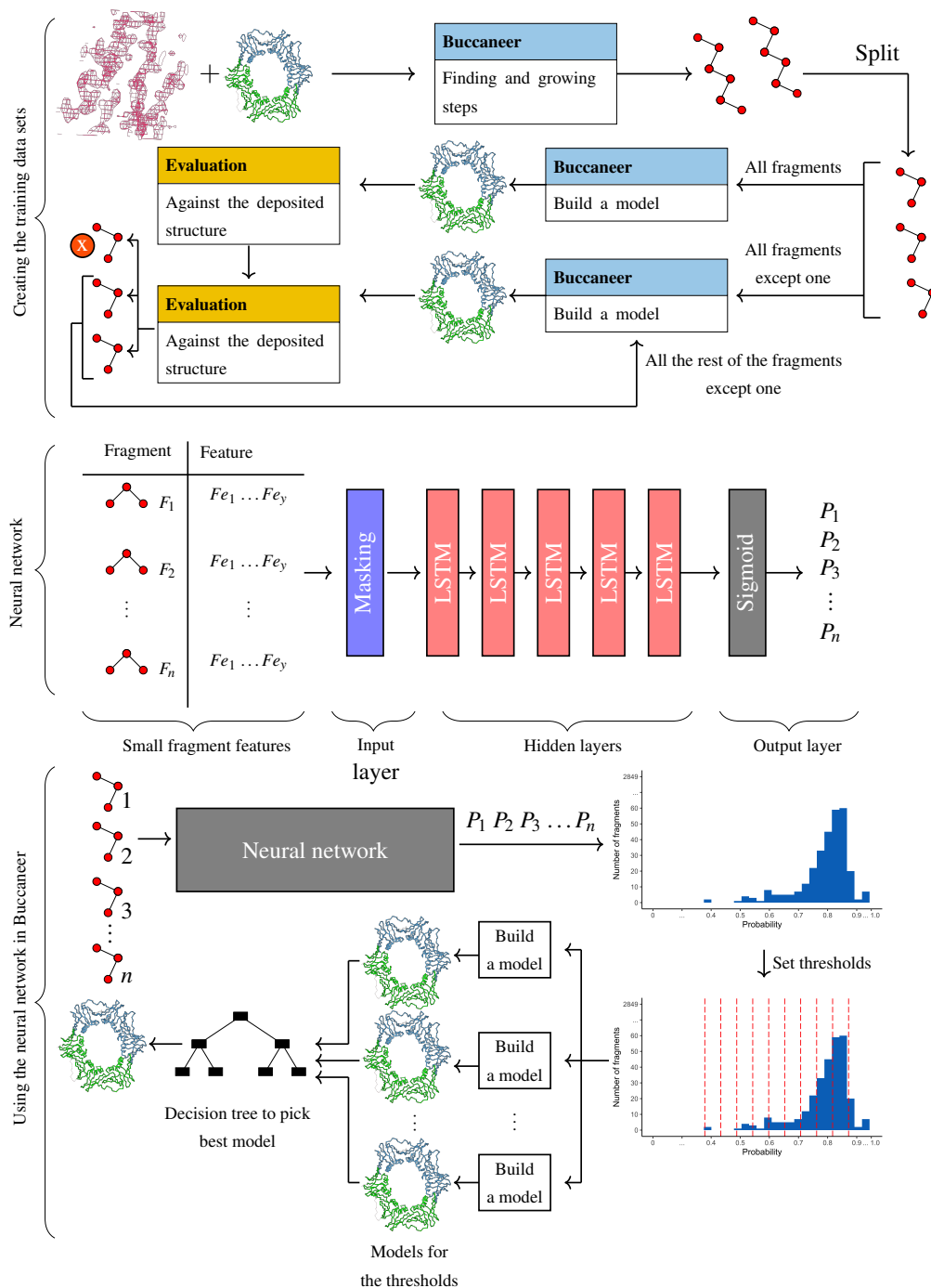


Figure 6.3: Creating the training data sets, the neural network architecture and using the neural network in Buccaneer.

in the next building cycles, and then pick the best model.

A decision tree (RTree) was trained to predict the best indicators to use in picking the best model (from the models built at different thresholds) using Weka framework version 3.8.5 [94]. The training data set for the decision tree was obtained by running Buccaneer using two different seeds with no neural network as using non-default seed

Table 6.2: Protein structure evaluation indicators; Buccaneer indicators, R-work and R-free. Indicated whether the indicator is better when has a higher or lower value.

Indicator	Higher or lower is better
Longest fragment	higher
Number of residues built	lower
Number of fragment	lower
Number of sequenced residues	higher
Number of residues uniquely allocated chain	higher
Completeness by residues	higher
Completeness by chain	higher
R-work	lower
R-free	lower

led to changes in the model. The difference between Buccaneer evaluation indicators, R-work and R-free, were calculated between models built from the same data set (Table 6.2). We deemed that the model is better when the structure completeness is at least 5% higher. The actual difference between the evaluation indicators was replaced by binary labels; “Y” when the indicator is better based on Table 6.2, otherwise “N”. Under-sampling was applied and cross-validation was used to train the decision tree.

The first model of these multiple models will be built from all the fragments, as the first threshold used to partition small fragments into “correct” and “incorrect” is always zero. The number of confirmation building cycles are the remaining of the initial number of the building cycles. For example, if Buccaneer runs on three building cycles, we run two and one confirmation building cycles in the first and second building cycles, respectively; no confirmation building cycle is run in the third building cycle. As our neural network model is limited to 2849 small fragments, Buccaneer will not use the neural network model when the number of fragments exceeds this limit.

## 6.4 Results

### 6.4.1 Evaluation of neural network training

As is common in machine learning, we have tried a wide range of neural network architectures and training hyperparameters in order to obtain a suitable neural network for our framework. For instance, we trained alternative neural networks with six layers and between 1024 and 32 neurons, and we used multiple learning rate for the training process (e.g. 0.001 and 0.005). From all the candidate neural networks we obtained,



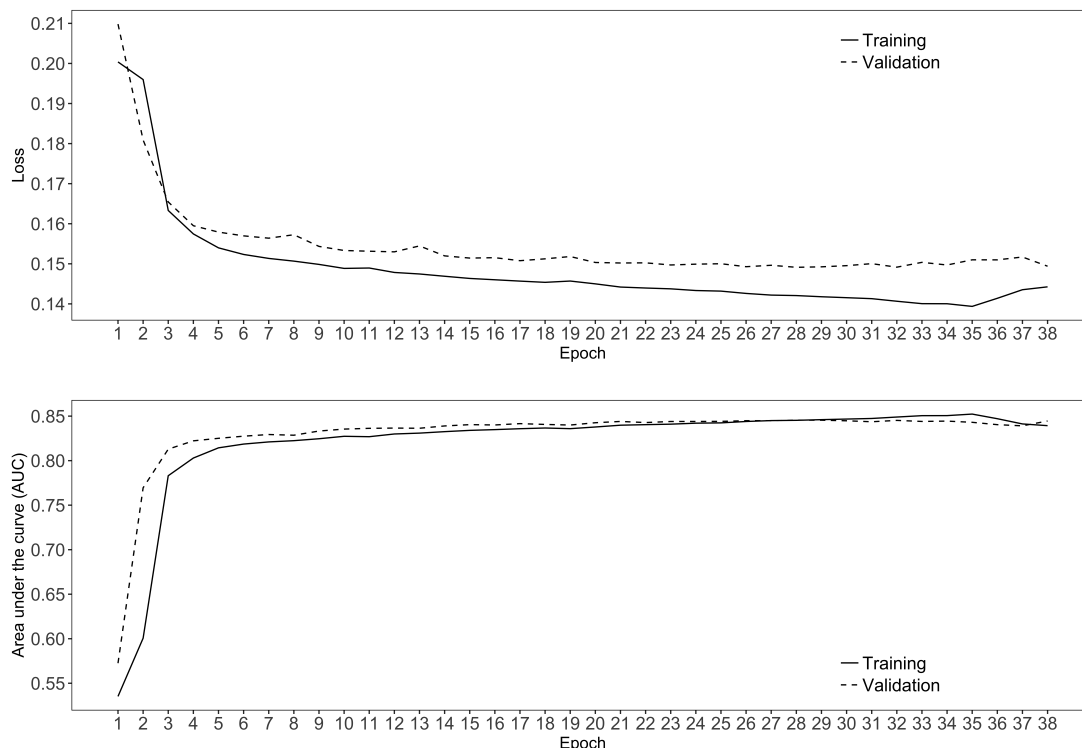


Figure 6.4: Difference in loss score and AUC between training and validation data sets (the data sets used during the model training for frequent evaluation and tuning of the model’s parameters) across the epochs. The best model was obtained from epoch 28 as this has the highest AUC on the validation data sets.

we selected the one that had five layers (the neural network detailed in Section 6.3.3.2).

The training of this neural network was stopped after epoch 38, as the AUC stopped improving at epoch 28. Figure 6.4 shows the AUC and loss score of the training and validation data sets across the 38 epoch. The AUC and loss score improved until epoch 28. Then the neural network model started to be overfitted, as the difference of the loss score between the training and validation data sets became larger. The neural network model from epoch 28 was used as the final model.

## 6.4.2 Feature importance

Figure 6.5 shows the importance of features based on the change in AUC in training and validation data sets. The application of permutation feature importance as described in Section 6.3.3.3 affected the AUC negatively for each of the features, decreasing it with between 0.001 and 0.11; the mean of the LLk score has the highest impact on the model. The positions of the residues dropped the performance of the model by more than 0.03 of AUC. Other features have less impact on the model per-

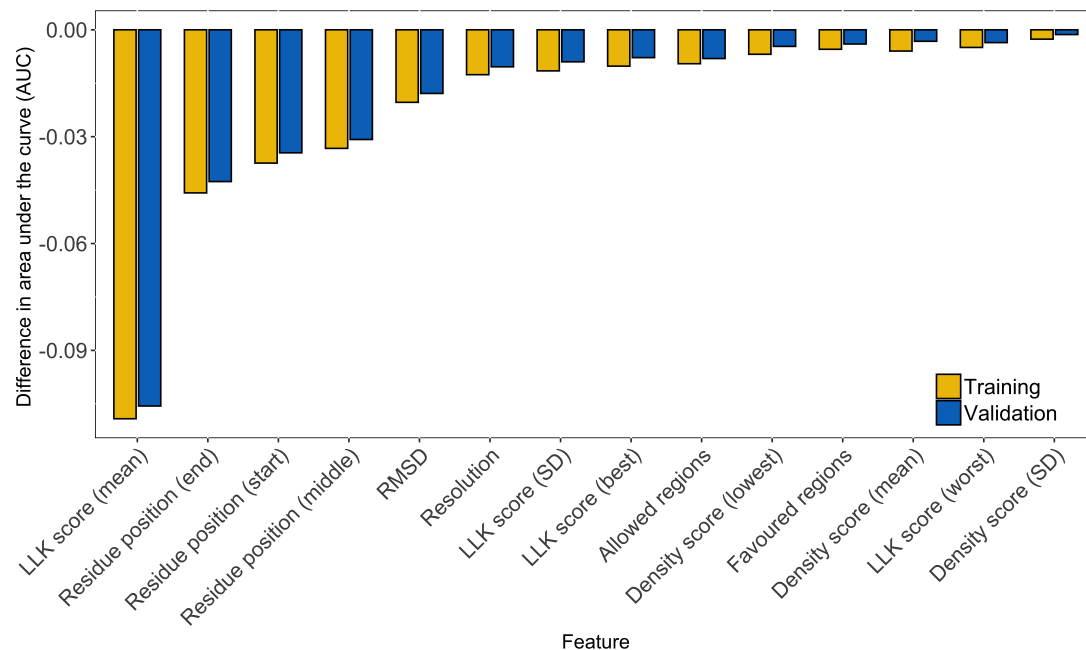


Figure 6.5: Difference between the baseline model ( where the values of the features are not shuffled in training and validation) and the model where the feature values are shuffled to find out the features importance.

formance; the SD of density score has the lowest impact.

Overall, the features using mean have a higher level of importance compared to the SD features for the same characteristics. For example, the mean LLK score and density score have a higher importance than the SD of the same scores. Comparing the regions of Ramachandran angles showed that the model relied on the allowed regions feature rather than the favoured regions.

### 6.4.3 Evaluation of using the neural network in Buccaneer

We assessed the effect of using the neural network in Buccaneer for three data sets:

1. 900 experimental phasing from the Joint Center for Structural Genomics (JCSG) as original and truncated resolutions [72, 11];
2. 205 newer experimental phasing data sets deposited between 2015–2021 and taken from PDB;
3. 219 MR data sets (the remaining of the 1351 MR data sets from Section 6.3.1) that were not used in either the training or validation of the neural network.

The resolution of the JCSG experimental phasing data sets was between 1.2 Å and 4Å, corresponding to 150 and 750 data sets at original and truncated resolution, respectively. Structure completeness, R-work and R-free were considered in this evaluation; we deemed the Buccaneer version augmented with the neural network better when the improvement was at least 5% in the relevant measure.

We ran Buccaneer twice to evaluate the two methods of selecting the threshold for including small fragments in the Buccaneer tracing, as described in Section 6.3.4. We set the maximum number of models to ten by selecting 10 equidistant thresholds and only building a model for those thresholds whose use increased the number of small fragments deemed “correct” compared to the previous threshold.

All the experiments used the Buccaneer v.1.6.11 and i1 pipeline, as implemented in CCP4 v.7.0.045. We will refer to the Buccaneer variant that uses the neural network as ‘Buccaneer(NN)’ in the rest of the chapter.

#### **6.4.3.1 Evaluation of the decision tree**

The decision tree was trained using data obtained through running Buccaneer on JCSG experimental phasing data sets; 562 protein structures were used in training and testing after under-sampling. The trained decision tree predicted the best model has lower R-work and higher uniquely residues allocated to a chain. The precision, recall and F-Measure were 0.781, 0.778 and 0.777, respectively.

#### **6.4.3.2 Experimental phasing**

We summarise below the results obtained for the Buccaneer(NN) variants with small-fragment selection based on both equidistant and Freedman–Diaconis thresholds.

This Buccaneer(NN) variant with equidistant thresholds built 22% and 38% of the protein structures with at least 5% higher completeness than Buccaneer for the JCSG original and truncated resolution data sets, respectively, compared to 1% and 10% of the data sets were built better by Buccaneer without the NN.

For the original resolution, Buccaneer(NN) improved the R-work and R-free of 4% and 5% of the data sets, respectively, and no structure was better built by Buccaneer. At truncated resolutions, 9% and 13% of the protein models were built by Buccaneer(NN) with better R-work and R-free. By comparison, only 4% of the pro-

tein structures were built with better R-free, and none of the structures were built with better R-work by Buccaneer.

Using the Freedman–Diaconis rule to select the threshold led to Buccaneer(NN) building 25% and 47% of the protein models with (at least 5%) higher structures completeness, and 2% and 7% with (at least 5%) lower structures completeness compared to Buccaneer, for the JCSG original and truncated resolution data sets, respectively. R-work and R-free improved as when a fixed number of thresholds was used for the JCSG original resolution. However, for the JCSG truncated resolution data sets, 17% and 20% of the protein structures were built with lower R-work and lower R-free, respectively, and 5% of the structures were built with higher R-free compared to Buccaneer; none of these structures was built with higher R-work.

For experimental phasing data sets recently deposited, 25% and 26% of the protein models were built with higher structure completeness by Buccaneer(NN) using the fixed number of thresholds and the Freedman–Diaconis rule, respectively, and 3% were built with lower structure completeness (by both Buccaneer(NN) variants) compared to Buccaneer. R-work improved in 6% of the data sets for both threshold-selection methods, and R-free in 8% and 6% of the data sets for the fixed number of thresholds and the Freedman–Diaconis rule, respectively; no protein structure built by Buccaneer had better R-work or R-free.

Figures 6.6 and 6.7 show the results of JCSG experimental phasing for structure completeness, R-work, R-free and structure correlation, and the structure completeness for the recently deposited data sets. The R-work, R-free and structure correlation results for the recently deposited data sets are reported in the Appendix C.

For the JCSG experimental phasing data sets, the results show multiple data sets for which the completeness significantly increased from around 20% when no neural network was used to around 70% for Buccaneer(NN) with fixed threshold, and improved even further when the Freedman–Diaconis rule was used. While Buccaneer(NN) did not produce better protein models for a number of data sets, it did improve the majority of the structures by different degrees. These improvements were less significant when Buccaneer(NN) used the Freedman–Diaconis rule.

R-work and R-free show less improvement than completeness, but Buccaneer(NN) did still achieve remarkable improvements in R-free and R-work for several data sets.

For example, Buccaneer(NN) lowered R-free for a number of data sets from over 0.37 to approximately 0.27 (when using the fixed number of thresholds), and from over 0.52 to under 0.32 (when using the Freedman–Diaconis rule).

Structure correlation shows that Buccaneer(NN) built the protein structures slightly closer to the deposited structures and closer when using the Freedman–Diaconis rule. However, fewer protein structures got worse than those when compared based on structure completeness, R-work or R-free.

For the recently deposited data sets, Buccaneer(NN) only produced slight improvements for the protein structures that were already built by Buccaneer with high completeness. However, the structures built with medium completeness by Buccaneer were improved when built by Buccaneer(NN). Only a few protein structures were built with slightly lower completeness by Buccaneer(NN) compared to Buccaneer.

We illustrate the use of Buccaneer(NN) in Figures 6.8 and 6.9, which depict two protein structures built by Buccaneer and by our two Buccaneer(NN) variants. To provide an impartial view, we present both a protein structure whose modelling is improved by Buccaneer(NN) (PDB id 6HCZ, Figure 6.8) and a protein structure that Buccaneer builds with better results (PDB id 2GNR, Figure 6.9). Thus, for PDB id 6HCZ, Buccaneer(NN) using the Freedman–Diaconis rule increased the structure completeness by 42%, while for PDB id 2GNR, the structure completeness decreased by 17% when Buccaneer(NN) was used.

### 6.4.3.3 MR

For the MR data sets, Buccaneer(NN) with a fixed number of thresholds produced protein models with (at least 5%) better completeness, R-work and R-free than Buccaneer for 15%, 4% and 5% of the data sets, respectively. By comparison, Buccaneer built protein structures with better completeness and R-free for only 2% and 1% of the data sets, respectively; no protein structure built by Buccaneer had better R-work than the corresponding structure built by Buccaneer(NN). Using the Freedman–Diaconis rule to select the threshold, 16%, 2% and 3% of the MR data sets were built with better completeness, R-work and R-free, respectively, by Buccaneer(NN), compared to only 2% of the MR data sets built with higher structure completeness by Buccaneer; no structure was built with (at least 5%) worse R-work or R-free by Buccaneer(NN).

Figure 6.10 shows the same result analysis for the MR data sets as in Figure 6.6. The results obtained for individual data sets show multiple significant improvements achieved by Buccaneer(NN); for example, Buccaneer(NN) with a fixed number of thresholds improved the completeness of one protein structure from around 40% to more than 70%, and decreased the R-free of another protein structure from around 0.41 to approximately 0.31. Moreover, structure correlation is improved for some protein structures from around 0.50 to close to 0.70.

#### **6.4.4 Evaluation of execution times**

Figure 6.11 shows the mean Buccaneer and Buccaneer(NN) execution times for JCSG original data sets. We ran both Buccaneer variants using a 173-node high-performance cluster with 7024 Intel Xeon Gold/Platinum cores and a total memory of 42 TB. For small structures, both Buccaneer and Buccaneer(NN) built the structures in less than 50 minutes. However, this execution time increased to around 450 minutes when large structures were built using Buccaneer(NN) with the Freedman–Diaconis rule and to under 150 minutes using the Buccaneer(NN) variant with fixed thresholds. In contrast, the Buccaneer completed the building of the large structures in under 50 minutes.

#### **6.4.5 Evaluation of using the neural network in Buccaneer running from ModelCraft**

ModelCraft [109] is a newly released pipeline using Buccaneer to build a model and perform other steps between the iterative building, such as it improves phases using Sheetbend [110, 111] and Parrot [50], adding water using coot before running Buccaneer [112, 46], pruning chains [26], and build nucleic acids using Nautilus [113].

We ran ModelCraft v.1.0 with Buccaneer(NN) instead of the standard Buccaneer for the JCSG experimental phasing data sets from Section 6.4.3. Figure 6.12 compares the ModelCraft and i1 pipelines ran with and without the neural network, in terms of mean structure completeness, R-work and R-free achieved for these data sets. The i1 and ModelCraft variants that used the neural network (regardless of the threshold selection methods) outperformed the pipelines without the neural network for all three evaluation measures.

## 6.5 Discussion

A new method to improve the backbone tracing step of protein model building software by using a neural network was presented. As no training data sets were available, we created our training data sets and used them in neural network training and validation. Moreover, two experimental phasing data sets were used in the evaluation.

The evaluation of the feature importance in determining correct and incorrect small fragments yielded unexpected results. In particular, the RMSD of the data sets has lower importance than the residue position type. In contrast, the LLK score (used in Buccaneer to decide when fragments stop growing) has the highest importance in discriminating between correct and incorrect small fragments among all the other features in the model building. A comparison between the impact of the mean and SD of the features shows that the mean of a feature has higher importance than its SD.

Optimizing the threshold used to select the small fragments used in the tracing step of the model building is key to achieving good neural network performance. The imbalance of the feature data makes this particularly challenging. In this paper, we addressed the threshold-tuning problem by trying several thresholds obtained both by using a fixed number of equidistant thresholds and the Freedman–Diaconis rule. The evaluation of the two threshold methods shows that the Freedman–Diaconis rule is more effective at worse crystallography data set resolutions as the truncated resolutions data sets; which all of their resolutions are worse than 3.1 Å, improved in their structure completeness more than the original resolutions data sets .

Training a decision tree to predict the best indicators for selecting the best model from a set of models showed that R-work and the residues uniquely allocated to a chain are best at reflecting the improvement in the structure completeness. Running Buccaneer on different seeds than the default one led to changes in the fragments, and therefore to change in the structure completeness. However, 205 newer experimental phasing data sets were used in the evaluation in order to eliminate the potential bias due to using JCSG experimental phasing data sets in the training of the decision tree.

The systematic evaluation shows that completeness, R-work/R-free and structure correlation are significantly improved by Buccaneer(NN). For MR data sets, we noticed that Buccaneer(NN) significantly improved the structures that Buccaneer built

with R-work lower than 0.43. This may suggest that the structures with high R-work (above 0.43) have no or few correct fragments, and therefore the use of the neural network cannot improve them. The problem needs to be addressed by extending the neural network to build correct fragments itself instead of only using those built by Buccaneer. Moreover, our neural network has an input size that can accommodate up to 2849 small fragments. Buccaneer built a larger number of small fragments for some of the MR data sets, which led to the neural network not being used in all or some of the building cycles of these data sets.

Buccaneer(NN) achieved higher levels of improvement in structure completeness than in R-work, R-free and structure correlation. This may be due to our use of structure completeness as an improvement measure when the training data sets for the neural network were created. In future work, this will be addressed by creating training data sets based on the structure completeness, R-work, R-free and structure correlation, and training a new version of the neural network.



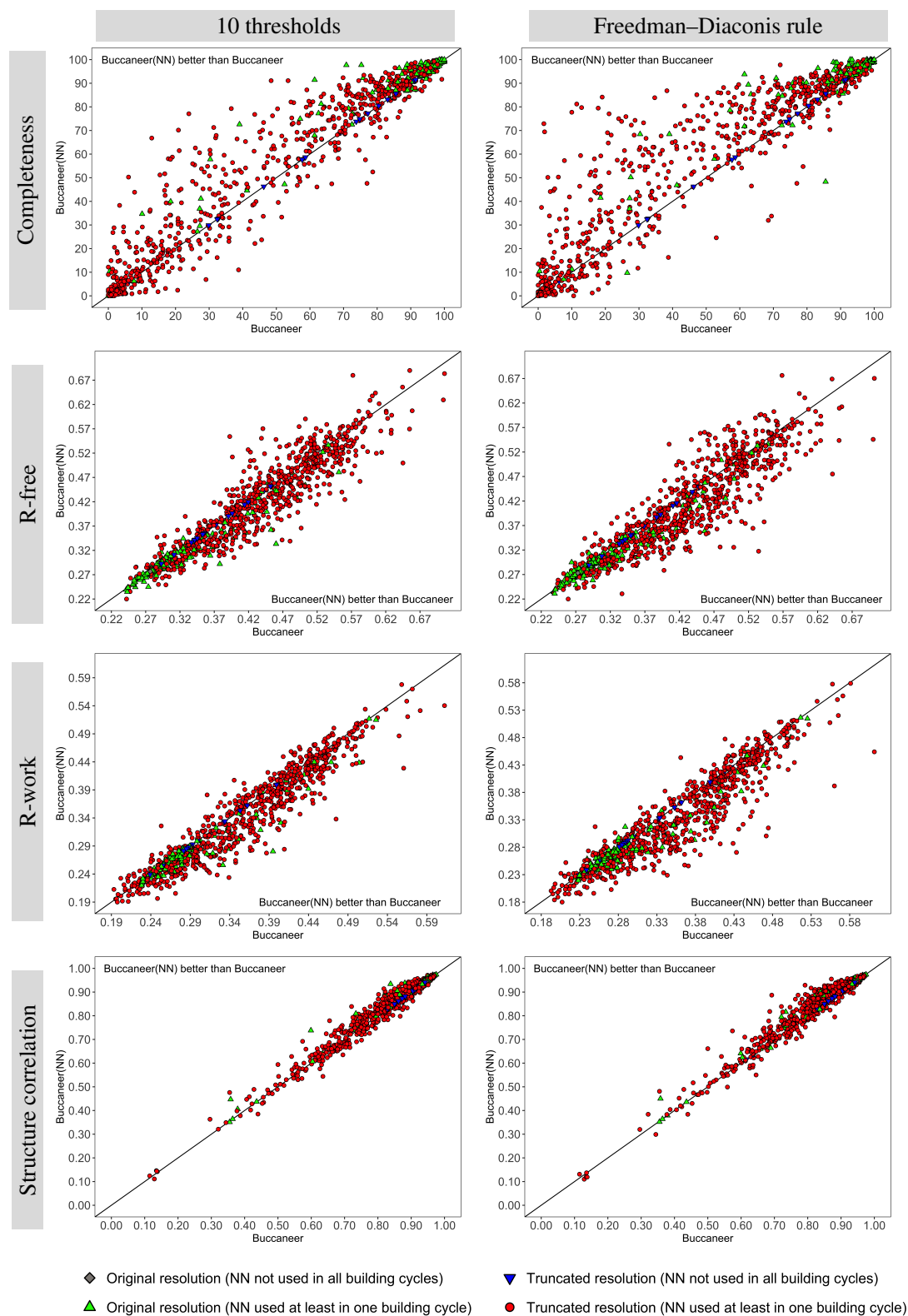
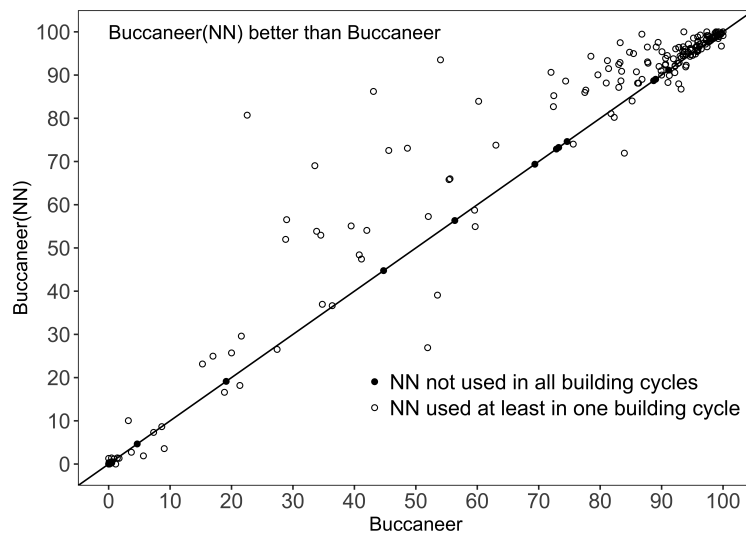
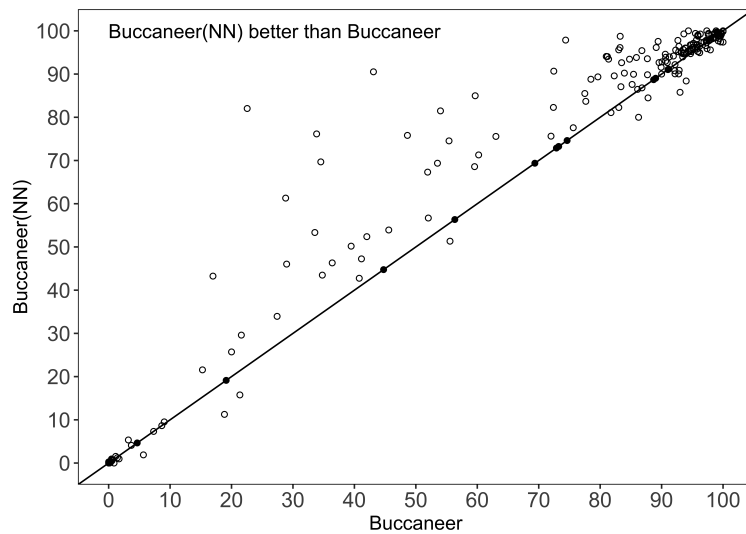


Figure 6.6: Comparison of structure completeness, R-work and R-free between Buccaneer and the Buccaneer with neural network (Buccaneer(NN)) variants using ten thresholds and the Freedman–Diaconis rule, for the JCSG experimental phasing data sets with original and truncated resolutions. The regions where Buccaneer(NN) is better than Buccaneer (either below or above the diagonal) are indicated in the diagrams.



(a)



(b)

Figure 6.7: Comparison of structure completeness between Buccaneer and the Buccaneer(NN) variants for the recently deposited experimental phasing data sets. (a) The Buccaneer(NN) using 10 thresholds. (b) The Buccaneer(NN) using the Freedman–Diaconis rule. The regions where Buccaneer(NN) is better than Buccaneer are indicated in the diagrams.

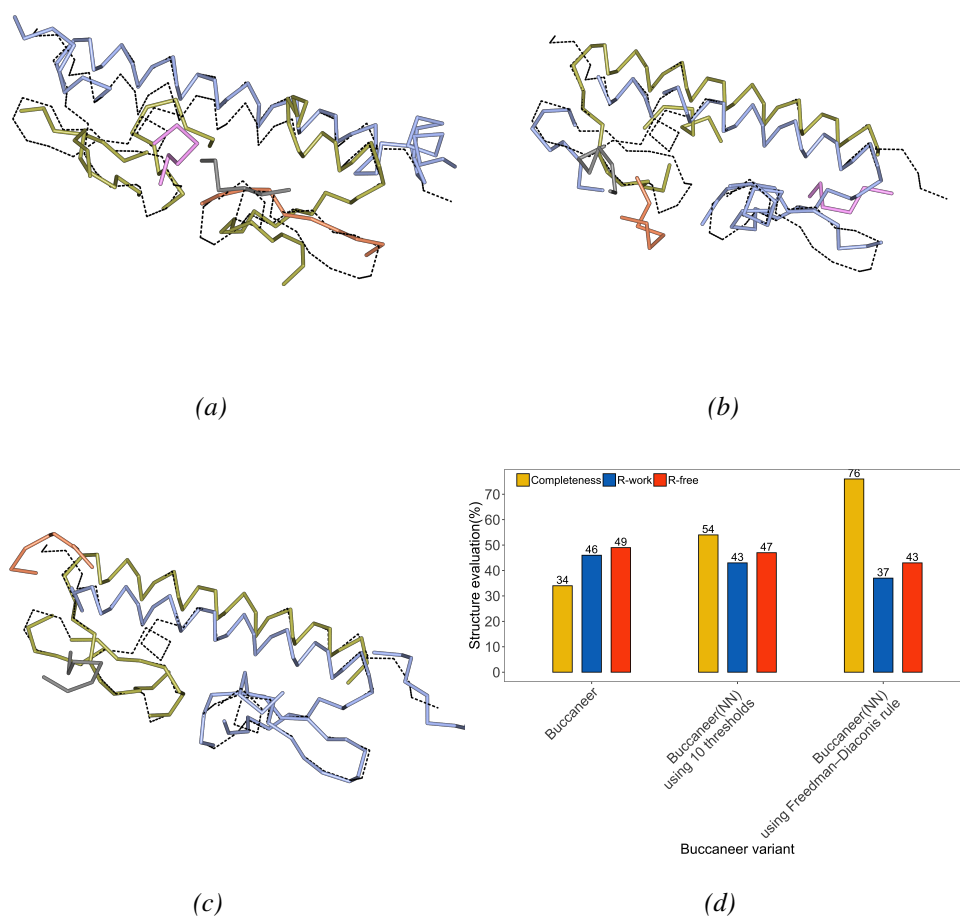


Figure 6.8: A protein structure built by Buccaneer and Buccaneer(NN) compared to the deposited structure, with the chains of the deposited structure depicted in dashed bonds. *a* The structure built by Buccaneer. *b* and *c*. The protein structure built by Buccaneer(NN) using ten thresholds and Freedman–Diaconis rule, respectively; *d* The structure completeness,  $R_{work}$  and  $R_{free}$  achieved by Buccaneer and the two Buccaneer(NN) variants. The structure PDB ID is 6HCZ and its resolution is 2.3 Å.

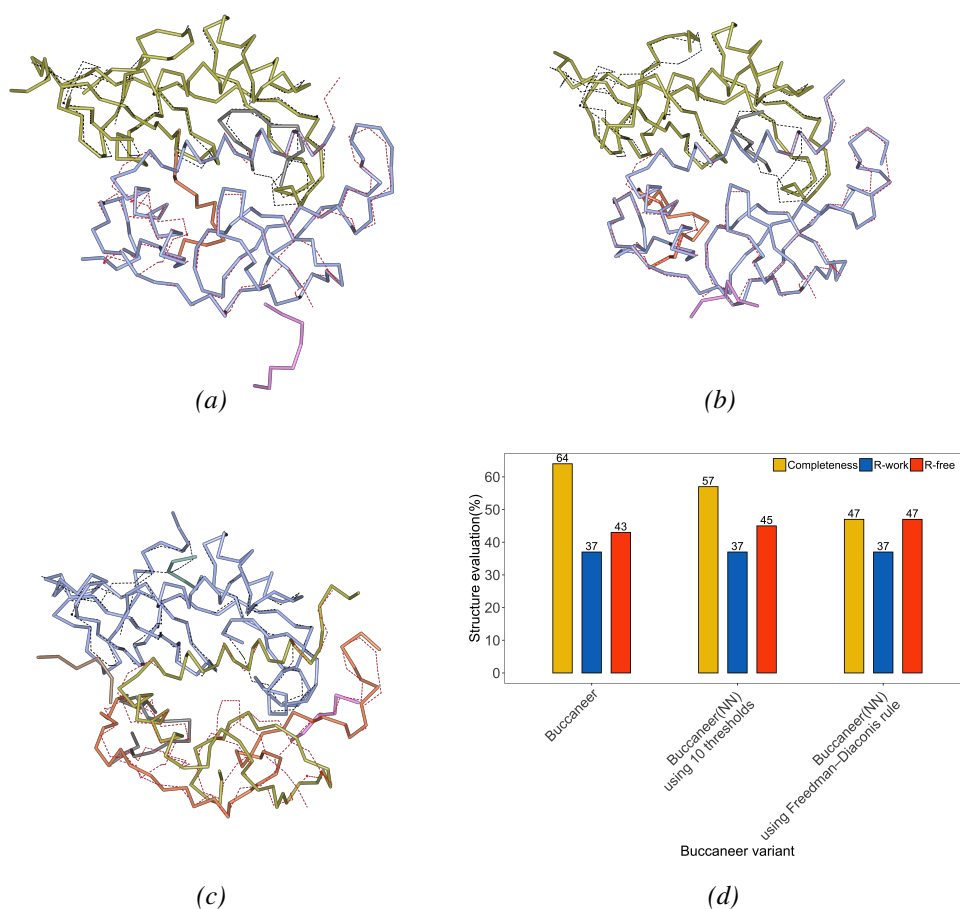


Figure 6.9: A protein structure built by Buccaneer and Buccaneer(NN) compared to the deposited structures. The chains of the deposited structure are in dashed bonds. *a* The structure built by Buccaneer. *b* and *c* The protein structure built by Buccaneer(NN) using ten thresholds and Freedman–Diaconis rule, respectively. *d* The structure completeness,  $R_{work}$  and  $R_{free}$  of the Buccaneer variant. The structure PDB ID is 2GNR and its truncated resolution is 3.2 Å.

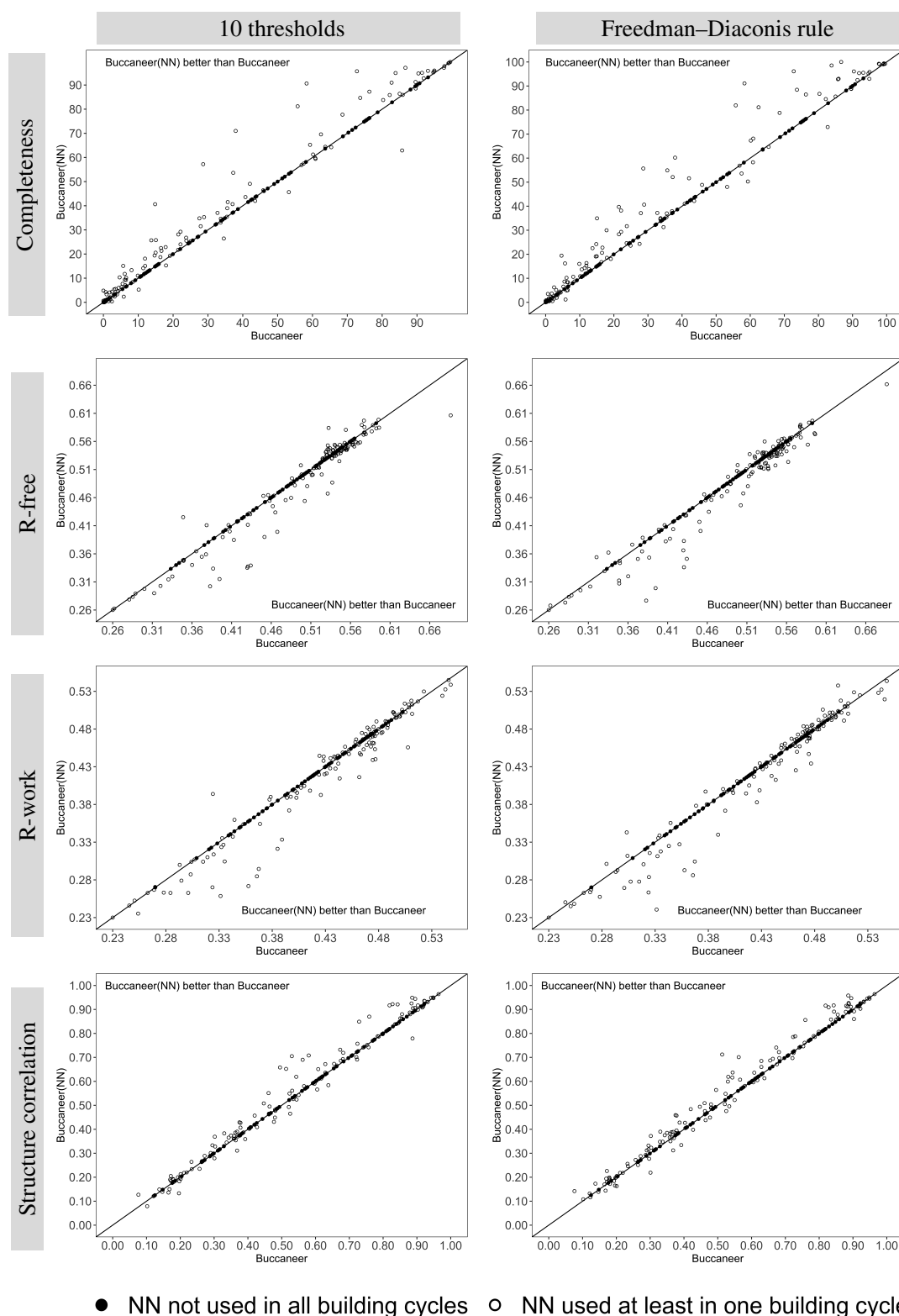


Figure 6.10: Comparison of structure completeness, R-work and R-free between Buccaneer and Buccaneer with neural network (Buccaneer(NN)) using ten thresholds and Freedman-Diaconis rule for the MR data sets. The results where Buccaneer(NN) is better than Buccaneer either below or above the diagonal is indicated in the figures.

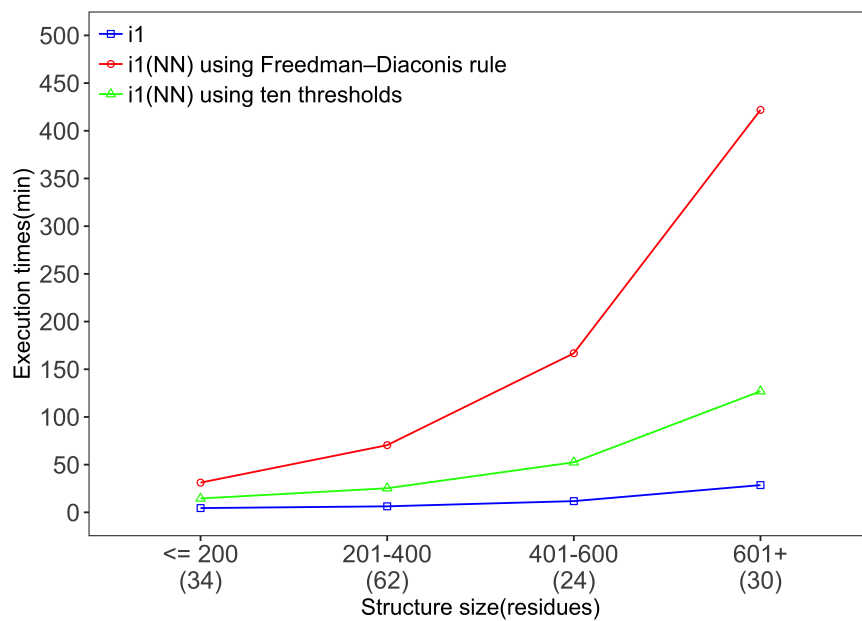


Figure 6.11: Mean execution time of Buccaneer and Buccaneer(NN) for the JCSG original data sets. The structure sizes are grouped into classes, and the number of data sets in each class is reported under the graph.

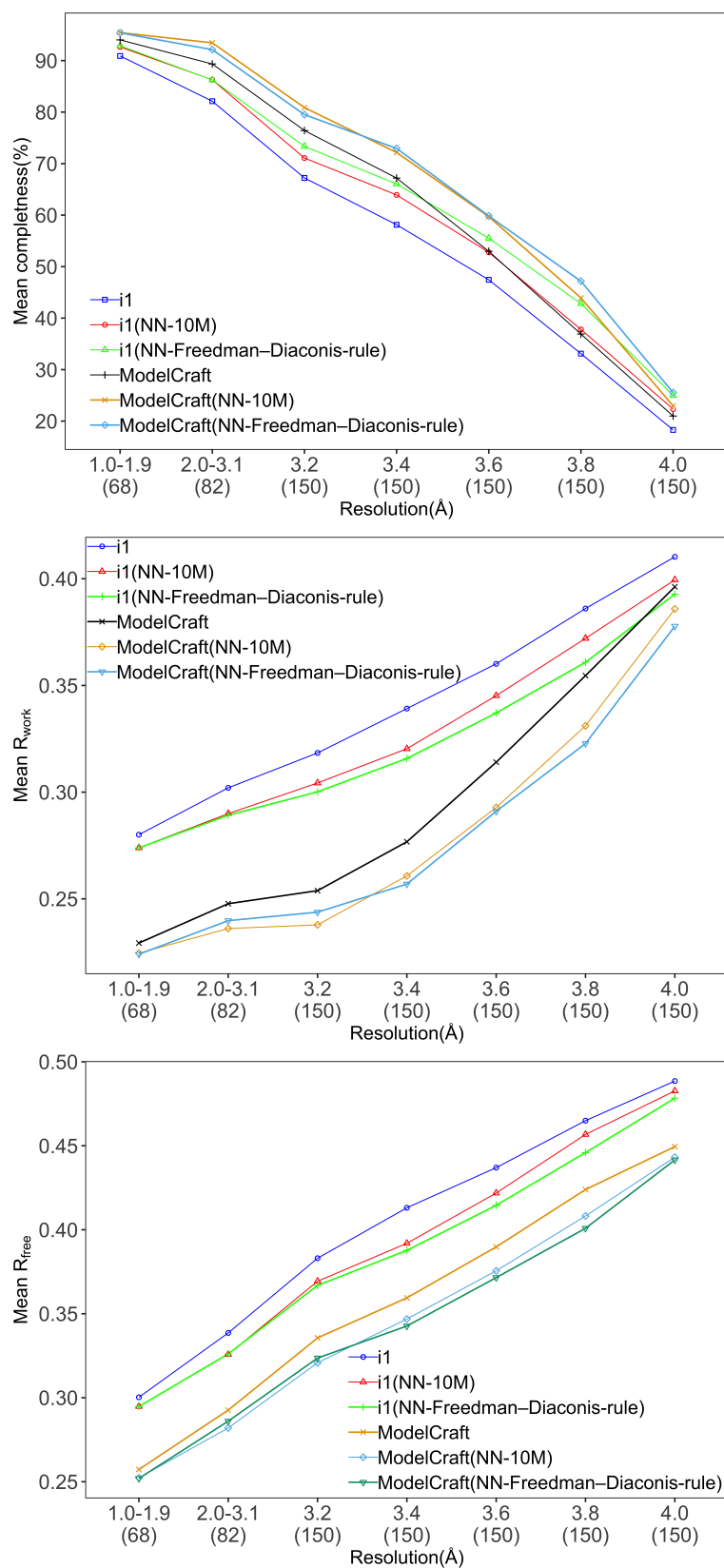


Figure 6.12: Comparison of the mean structure completeness, R-work and R-free achieved by ModelCraft and i1 with and without neural network at different data set resolutions. The number of the data sets at each resolution is reported under the plots.

# Conclusion

## 7.1 Summary

In this thesis, we introduced methods that improve protein model building from crystallography data sets both by using existing software pipelines and by enhancing one of these pipelines with a machine learning component. To evaluate the improvement of our methods, we first conducted a comparison of established protein model building pipelines in order to find a baseline for protein model building pipelines. This comparison provided insights into where these pipelines perform well on the data sets, which led to running the pipelines in pairwise combinations. However, as no single pipeline or pipeline combination can perform well on all crystallography data sets, we introduced a predictive machine learning model that addresses the problem of choosing the “best” pipeline for each data set.

In Chapter 3, we compared the protein model pipelines using JCSG experimental phasing data sets. We started with 202 data sets, and we obtained additional data sets by truncating each JCSG data set to lower resolution, resulting in a total of 1211 data sets (both as original and truncated resolution). For a fair comparison, we ran the pipelines using their default parameters and the same computational resources. We used four measures to evaluate the performance of these pipelines; structure completeness, R-work, R-free and structure correlation. We run REFMAC for zero cycles to obtain comparable refinement statistics across all pipelines to avoid the effect of the different parameterizations used by the pipelines (REFMAC can be run for zero cycles to calculate R-work and R-free using the REFMAC keyword “NCYCLES 0”). We chose these measures for our comparison because R-work and R-free are widely used, and the structure completeness enabled the evaluation of the build model against the



deposited model, and structure correlation to identify the bias of structure completeness. As an additional evaluation, we assessed the execution times of these pipelines for different structure sizes. Our findings suggest that no single pipeline is best across all the data sets. This insight led us to use pairs of pipelines in combination to produce better protein structures.

In Chapter 4, we run the pipelines in pairwise combinations by using the structure built by the first pipeline as an initial structure (i.e., as input) for the second pipeline. We used the same data sets as in Chapter 3; however, as here we were not interested in comparing the pipelines run on their own, we used R-free when running ARP/wARP in combination of with one of the other pipelines.<sup>1</sup> The same evaluation measures as in Chapter 3 were used, except R-work, which was not used because different model-building programs use different model parameterizations, and that may lead to overfitting and the underestimation of R-work. We showed experimentally that the pairwise pipeline combinations can yield a significant improvement over the individual pipeline performance. However, the challenge of finding the best pipeline or pipeline combination for a given crystallography data set is amplified since the number of options is considerably increased by the opportunity to use pipeline combinations in addition to individual pipelines.

In Chapter 5, we address the problem of choosing the best pipeline based for a given crystallography data set. We began by analyzing the research community's use of the pipelines in order to identify the criteria underpinning the researchers' choice of pipelines. This analysis suggests that the researchers were to a great extent influenced by factors relating to their geographical location. This finding led us to train a predictive machine learning model for recommending a pipeline for a data set based on features calculated from the density map. We evaluated the resulting predictive model based on machine learning performance metrics (RMSE and MAE). Moreover, we picked MR data sets and ran all pipelines on these data sets, and then compared the pipeline used by the research team who published the data set to the pipeline recommended by our predictive model; we based this comparison on structure completeness achieved by the two pipelines. Also, we showed the significant time saving achieved by using our recommended pipeline instead of running all the pipelines on the given

---

<sup>1</sup>R-free is not applicable to ARP/wARP alone.

data set. To help researchers build protein models for their data sets using the pipeline or pairwise pipeline combination recommended by our predictive model, we developed software that automatically generates ready-to-run scripts for the processing of the given data set by the recommended individual pipeline or pipeline combination. Finally, we implemented the predictive model as a web application which is deployed on EGI Foundation cloud infrastructure<sup>2</sup> at <http://www.robin-predictor.org> and is free to use.

Last but not least, in Chapter 6 we improved Buccaneer by identifying incorrect fragments using a neural network we specifically developed for this purpose, and removing them to improve backbone tracing. To train the neural network, we first labelled small fragments from training data sets as “correct” and “incorrect” by removing a small fragment at a time and building a protein model, and then comparing that model with the baseline model. The training data sets were used to train a neural network comprising five LSTM layers. This neural network can be used in the joining step of Buccaneer to identify and discard “incorrect” small fragments. Using the default threshold for making this decision is insufficient, as the fragments probabilities produced by the neural network can be skewed. To address the problem of selecting an optimal threshold, we introduced two methods—one involving the use of a fixed number of thresholds, and the other using the Freedman–Diaconis rule. The Buccaneer variant augmented with our neural network built a more complete models for most JCSG data sets. The improvement achieved for MR data sets was less significant, primarily because many of the models generated for these data sets had more small fragments than could be handled by the neural network, meaning that Buccaneer had to bypass the neural network and to employ its standard process for protein building.

## 7.2 Limitations and future work

This thesis presented methods to improve the building of protein models from crystallography data sets. Notwithstanding the benefits of the research contributions summarised in the previous section, they also have limitations that need to be addressed in future research. We examined the usefulness of combining the pipelines; however,

---

<sup>2</sup><https://www.egi.eu>

the pipelines were run using default parameters. Optimizing the pipelines' parameters, e.g., by using search algorithms, may lead to a more complete model being built for a given data set. This optimization of the pipelines' parameters could be carried out with beneficial outcomes on both individual pipelines and pipelines combinations.

To predict the performance of the protein building pipelines for a crystallography data set, we trained an ML model for each pipeline. These ML models could be combined into a single ML model, as this may improve the predictive model's performance. Moreover, additional ML algorithms could be tested, e.g. XGBoost [101] and neural networks. In particular, using density maps features as an input for the neural network may give better estimations of the quality of the phases and, therefore, better predictions for those pipelines that are more dependent on the phases than other features.

For the improvement of protein model building using the neural network, multiple aspects of the research presented in the thesis can be improved further. Firstly, the creation of the training data sets was based on the improvement of the structure completeness, which led to a lower level of improvement for R-work and R-free when the neural network was used in Buccaneer. The training data sets could be improved by evaluating the structures based on the four evaluation measures. Moreover, the training data sets were obtained from one building cycle in Buccaneer, with the fragments labelled based on their correctness in the first cycle. Those fragments may not get the same "correctness" in further cycles. We envisage that running more cycles to label the correctness of the fragments could lead to more accurate labelling.

Removing fragments from the model has a different effect on the structure completeness, as not all the fragments affect the model in the same manner. In this thesis, we coarsely partitioned the fragments into two groups regardless of the degree of their impact on the structure completeness. The fragments could be labelled based on the actual impact on the structure completeness. This could help identify and discard only those fragments with a highly negative impact on the structure building.

The training data sets used in training the neural network are imbalanced, leading to skewed predictions, which makes the default threshold ineffective. We addressed this problem by building multiple models and picking the best. However, this solution could be time-consuming in large structures. An optimal threshold can be found by

running Buccaneer using different thresholds on the training data sets and then setting the threshold that is found effective in the majority of the data sets as a default threshold. This would decrease the number of models to be built and, therefore, speed up the model building.

The data sets that Buccaneer built with low structure completeness and Buccaneer with neural network did not improve them may suggest that no correct fragments were found. These data sets may have poor phases and result in wrong fragments being placed. However, the release of AlphaFold partially solved the problem of the data sets have poor phases, particularly when the predicted model has a high confidence score and use as a search model to obtain good phases; therefore, this is an obvious area for future development for those data sets where AlphaFold models predicted with low confidence [51]. Current density modification algorithms do not use machine learning, and as known, obtaining good phases will lead to building a better protein model as Buccaneer more depends on the phases than other map features.

# Comparison of automated crystallographic model-building pipelines (additional results)

## A.1 Experimental results for the original data sets used in Buccaneer development

Table A.1: Complete and intermediate models produced by the 7 pipeline variants for the 52 original data sets, where (T) and (C) denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	HA-NCS			MR-NCS			NO-NCS		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
ARP	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
ARP(B 5I)	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
iI(5I)	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
PHENIX/Parrot	51	1(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
SHELXE/Parrot	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
PHENIX	-	-	-	-	-	-	52	0(T) 0(C)	0
SHELXE	-	-	-	-	-	-	52	0(T) 0(C)	0

Models used in the comparison: 52 HA-NCS, 52 MR-NCS and 52 NO-NCS.

Table A.2: Structure completeness comparison for the models generated from the 52 original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	29	27	29	69
ARP(B 5I)	48	0	29	37	79
i1(5I)	60	52	0	44	90
PHENIX/Parrot	58	50	44	0	83
SHELXE/Parrot	27	19	8	10	0



Table A.3: Structure completeness comparison for the models generated from the 52 original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	100	23	13	13	4
ARP(B 5I)	23	100	19	13	2
i1(5I)	13	19	100	12	2
PHENIX/Parrot	13	13	12	100	8
SHELXE/Parrot	4	2	2	8	100



Table A.4: Structure completeness comparison for the models generated from the 52 original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	12	13	13	56
ARP(B 5I)	17	0	15	12	63
i1(5I)	37	29	0	21	73
PHENIX/Parrot	31	31	21	0	67
SHELXE/Parrot	17	15	2	8	0



Table A.5: Structure completeness comparison for the models generated from the 52 original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	17	13	15	13
ARP(B 5I)	31	0	13	25	15
i1(5I)	23	23	0	23	17
PHENIX/Parrot	27	19	23	0	15
SHELXE/Parrot	10	4	6	2	0



Table A.6: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	29	96	52	100
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	42	0	96	62	100
ARP(B 5I) <i>R-free</i>	-	0	85	46	-
i1(5I) <i>R-work</i>	2	4	0	0	100
i1(5I) <i>R-free</i>	-	12	0	6	-
PHENIX/Parrot <i>R-work</i>	35	29	98	0	100
PHENIX/Parrot <i>R-free</i>	-	44	90	0	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.7: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	100	29	2	13	0
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	29	100	0	10	0
ARP(B 5I) <i>R-free</i>	-	100	4	10	-
i1(5I) <i>R-work</i>	2	0	100	2	0
i1(5I) <i>R-free</i>	-	4	100	4	-
PHENIX/Parrot <i>R-work</i>	13	10	2	100	0
PHENIX/Parrot <i>R-free</i>	-	10	4	100	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	100
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-





Table A.8: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	2	60	12	100
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	6	0	71	13	100
ARP(B 5I) <i>R-free</i>	-	0	60	21	-
i1(5I) <i>R-work</i>	0	0	0	0	96
i1(5I) <i>R-free</i>	-	6	0	0	-
PHENIX/Parrot <i>R-work</i>	4	0	48	0	100
PHENIX/Parrot <i>R-free</i>	-	13	50	0	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.9: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	27	37	40	0
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	37	0	25	48	0
ARP(B 5I) <i>R-free</i>	-	0	25	25	-
i1(5I) <i>R-work</i>	2	4	0	0	4
i1(5I) <i>R-free</i>	-	6	0	6	-
PHENIX/Parrot <i>R-work</i>	31	29	50	0	0
PHENIX/Parrot <i>R-free</i>	-	31	40	0	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.10: Structure completeness comparison for the models generated from the 52 original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	31	25	37	71
ARP(B 5I)	38	0	23	31	77
i1(5I)	60	60	0	50	94
PHENIX/Parrot	50	54	37	0	87
SHELXE/Parrot	23	21	6	8	0



Table A.11: Structure completeness comparison for the models generated from the 52 original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	100	31	15	13	6
ARP(B 5I)	31	100	17	15	2
i1(5I)	15	17	100	13	0
PHENIX/Parrot	13	15	13	100	6
SHELXE/Parrot	6	2	0	6	100



Table A.12: Structure completeness comparison for the models generated from the 52 original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	10	10	10	60
ARP(B 5I)	15	0	12	12	62
i1(5I)	37	35	0	21	75
PHENIX/Parrot	33	37	19	0	75
SHELXE/Parrot	12	10	2	4	0



Table A.13: Structure completeness comparison for the models generated from the 52 original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	21	15	27	12
ARP(B 5I)	23	0	12	19	15
i1(5I)	23	25	0	29	19
PHENIX/Parrot	17	17	17	0	12
SHELXE/Parrot	12	12	4	4	0



Table A.14: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP $R$ -work	0	27	92	58	100
ARP $R$ -free	-	-	-	-	-
ARP(B 5I) $R$ -work	42	0	94	54	100
ARP(B 5I) $R$ -free	-	0	79	44	-
i1(5I) $R$ -work	4	2	0	2	100
i1(5I) $R$ -free	-	13	0	6	-
PHENIX/Parrot $R$ -work	27	31	92	0	100
PHENIX/Parrot $R$ -free	-	48	87	0	-
SHELXE/Parrot $R$ -work	0	0	0	0	0
SHELXE/Parrot $R$ -free	-	-	-	-	-



Table A.15: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP $R$ -work	100	31	4	15	0
ARP $R$ -free	-	-	-	-	-
ARP(B 5I) $R$ -work	31	100	4	15	0
ARP(B 5I) $R$ -free	-	100	8	8	-
i1(5I) $R$ -work	4	4	100	6	0
i1(5I) $R$ -free	-	8	100	8	-
PHENIX/Parrot $R$ -work	15	15	6	100	0
PHENIX/Parrot $R$ -free	-	8	8	100	-
SHELXE/Parrot $R$ -work	0	0	0	0	100
SHELXE/Parrot $R$ -free	-	-	-	-	-



Table A.16: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	2	54	10	100
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	6	0	52	12	100
ARP(B 5I) <i>R-free</i>	-	0	52	15	-
i1(5I) <i>R-work</i>	0	0	0	0	98
i1(5I) <i>R-free</i>	-	8	0	0	-
PHENIX/Parrot <i>R-work</i>	2	2	38	0	100
PHENIX/Parrot <i>R-free</i>	-	10	40	0	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.17: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	25	38	48	0
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	37	0	42	42	0
ARP(B 5I) <i>R-free</i>	-	0	27	29	-
i1(5I) <i>R-work</i>	4	2	0	2	2
i1(5I) <i>R-free</i>	-	6	0	6	-
PHENIX/Parrot <i>R-work</i>	25	29	54	0	0
PHENIX/Parrot <i>R-free</i>	-	38	46	0	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.18: Structure completeness comparison for the models generated from the 52 original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP	0	23	27	35	40	75	73
ARP(B 5I)	50	0	31	33	38	83	81
i1(5I)	62	56	0	40	46	88	90
PHENIX/Parrot	54	46	50	0	38	79	85
PHENIX	50	42	40	29	0	81	85
SHELXE	21	17	6	10	12	0	37
SHELXE/Parrot	23	15	6	8	10	54	0



Table A.19: Structure completeness comparison for the models generated from the 52 original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP	100	27	12	12	10	4	4
ARP(B 5I)	27	100	13	21	19	0	4
i1(5I)	12	13	100	10	13	6	4
PHENIX/Parrot	12	21	10	100	33	12	8
PHENIX	10	19	13	33	100	8	6
SHELXE	4	0	6	12	8	100	10
SHELXE/Parrot	4	4	4	8	6	10	100



Table A.20: Structure completeness comparison for the models generated from the 52 original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP	0	2	13	12	13	56	60
ARP(B 5I)	21	0	13	12	19	65	69
i1(5I)	33	27	0	21	27	67	77
PHENIX/Parrot	35	31	27	0	13	67	69
PHENIX	35	27	31	8	0	67	71
SHELXE	13	6	0	6	8	0	10
SHELXE/Parrot	13	8	0	4	8	21	0



Table A.21: Structure completeness comparison for the models generated from the 52 original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP	0	21	13	23	27	19	13
ARP(B 5I)	29	0	17	21	19	17	12
i1(5I)	29	29	0	19	19	21	13
PHENIX/Parrot	19	15	23	0	25	12	15
PHENIX	15	15	10	21	0	13	13
SHELXE	8	12	6	4	4	0	27
SHELXE/Parrot	10	8	6	4	2	33	0



Table A.22: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP <i>R-work</i>	0	13	98	56	50	100	100
ARP <i>R-free</i>	-	-	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	50	0	100	63	62	100	100
ARP(B 5I) <i>R-free</i>	-	0	81	10	19	-	-
i1(5I) <i>R-work</i>	0	0	0	0	0	100	100
i1(5I) <i>R-free</i>	-	13	0	2	4	-	-
PHENIX/Parrot <i>R-work</i>	35	23	98	0	33	100	100
PHENIX/Parrot <i>R-free</i>	-	71	94	0	40	-	-
PHENIX <i>R-work</i>	35	27	100	29	0	100	100
PHENIX <i>R-free</i>	-	67	94	25	0	-	-
SHELXE <i>R-work</i>	0	0	0	0	0	0	21
SHELXE <i>R-free</i>	-	-	-	-	-	-	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0	33	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-	-	-



Table A.23: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP <i>R-work</i>	100	37	2	10	15	0	0
ARP <i>R-free</i>	-	-	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	37	100	0	13	12	0	0
ARP(B 5I) <i>R-free</i>	-	100	6	19	13	-	-
i1(5I) <i>R-work</i>	2	0	100	2	0	0	0
i1(5I) <i>R-free</i>	-	6	100	4	2	-	-
PHENIX/Parrot <i>R-work</i>	10	13	2	100	38	0	0
PHENIX/Parrot <i>R-free</i>	-	19	4	100	35	-	-
PHENIX <i>R-work</i>	15	12	0	38	100	0	0
PHENIX <i>R-free</i>	-	13	2	35	100	-	-
SHELXE <i>R-work</i>	0	0	0	0	0	100	46
SHELXE <i>R-free</i>	-	-	-	-	-	-	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0	46	100
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-	-	-





Table A.24: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP <i>R-work</i>	0	4	62	10	12	100	100
ARP <i>R-free</i>	-	-	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	4	0	67	15	13	100	100
ARP(B 5I) <i>R-free</i>	-	0	33	2	4	-	-
i1(5I) <i>R-work</i>	0	0	0	0	0	96	96
i1(5I) <i>R-free</i>	-	8	0	0	0	-	-
PHENIX/Parrot <i>R-work</i>	6	4	56	0	0	100	100
PHENIX/Parrot <i>R-free</i>	-	23	62	0	2	-	-
PHENIX <i>R-work</i>	6	4	56	0	0	100	100
PHENIX <i>R-free</i>	-	19	56	2	0	-	-
SHELXE <i>R-work</i>	0	0	0	0	0	0	0
SHELXE <i>R-free</i>	-	-	-	-	-	-	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-	-	-



Table A.25: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP <i>R-work</i>	0	10	37	46	38	0	0
ARP <i>R-free</i>	-	-	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	46	0	33	48	48	0	0
ARP(B 5I) <i>R-free</i>	-	0	48	8	15	-	-
i1(5I) <i>R-work</i>	0	0	0	0	0	4	4
i1(5I) <i>R-free</i>	-	6	0	2	4	-	-
PHENIX/Parrot <i>R-work</i>	29	19	42	0	33	0	0
PHENIX/Parrot <i>R-free</i>	-	48	33	0	38	-	-
PHENIX <i>R-work</i>	29	23	44	29	0	0	0
PHENIX <i>R-free</i>	-	48	38	23	0	-	-
SHELXE <i>R-work</i>	0	0	0	0	0	0	21
SHELXE <i>R-free</i>	-	-	-	-	-	-	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0	33	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-	-	-



## A.2 Experimental results for synthetic data sets for the original data sets used in Buccaneer development

Table A.26: Complete and intermediate models produced by the 5 pipeline variants for the 52 synthetic data sets, where (T) and (C) denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	HA-NCS			MR-NCS			NO-NCS		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
ARP	258	1(T) 0(C)	0	258	1(T) 0(C)	0	258	1(T) 0(C)	0
ARP(B 5I)	256	3(T) 0(C)	0	258	1(T) 0(C)	0	257	2(T) 0(C)	0
i1(5I)	259	0(T) 0(C)	0	259	0(T) 0(C)	0	259	0(T) 0(C)	0
PHENIX/Parrot	259	0(T) 0(C)	0	259	0(T) 0(C)	0	257	2(T) 0(C)	0
PHENIX	-	-	-	-	-	-	256	2(T) 0(C)	1

Models used in the comparison: 259 HA-NCS, 259 MR-NCS and 258 NO-NCS.

Table A.27: Structure completeness comparison for the models generated from the 52 synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	25	1	4
ARP(B 5I)	25	0	0	4
i1(5I)	97	97	0	88
PHENIX/Parrot	95	95	10	0

0 97

Table A.28: Structure completeness comparison for the models generated from the 52 synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	100	50	3	1
ARP(B 5I)	50	100	3	1
i1(5I)	3	3	100	1
PHENIX/Parrot	1	1	1	100

1  100

Table A.29: Structure completeness comparison for the models generated from the 52 synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	5	0	3
ARP(B 5I)	8	0	0	4
i1(5I)	93	93	0	86
PHENIX/Parrot	93	92	5	0

0  93

Table A.30: Structure completeness comparison for the models generated from the 52 synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	20	1	1
ARP(B 5I)	17	0	0	0
i1(5I)	3	3	0	3
PHENIX/Parrot	2	2	5	0



Table A.31: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	0	24	95	97
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	60	0	100	100
ARP(B 5I) $R$ -free	-	0	40	39
i1(5I) $R$ -work	4	0	0	45
i1(5I) $R$ -free	-	58	0	48
PHENIX/Parrot $R$ -work	2	0	49	0
PHENIX/Parrot $R$ -free	-	60	48	0



Table A.32: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	100	16	1	0
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	16	100	0	0
ARP(B 5I) $R$ -free	-	100	2	0
i1(5I) $R$ -work	1	0	100	7
i1(5I) $R$ -free	-	2	100	4
PHENIX/Parrot $R$ -work	0	0	7	100
PHENIX/Parrot $R$ -free	-	0	4	100



Table A.33: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	0	2	89	96
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	22	0	97	100
ARP(B 5I) $R$ -free	-	0	35	37
i1(5I) $R$ -work	2	0	0	22
i1(5I) $R$ -free	-	47	0	20
PHENIX/Parrot $R$ -work	1	0	28	0
PHENIX/Parrot $R$ -free	-	53	23	0



Table A.34: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP <i>R-work</i>	0	22	6	2
ARP <i>R-free</i>	-	-	-	-
ARP(B 5I) <i>R-work</i>	38	0	3	0
ARP(B 5I) <i>R-free</i>	-	0	5	2
i1(5I) <i>R-work</i>	2	0	0	22
i1(5I) <i>R-free</i>	-	11	0	28
PHENIX/Parrot <i>R-work</i>	1	0	20	0
PHENIX/Parrot <i>R-free</i>	-	7	25	0



Table A.35: Structure completeness comparison for the models generated from the 52 synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	27	1	4
ARP(B 5I)	25	0	0	3
i1(5I)	97	97	0	88
PHENIX/Parrot	95	96	10	0



Table A.36: Structure completeness comparison for the models generated from the 52 synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	100	49	3	1
ARP(B 5I)	49	100	3	1
i1(5I)	3	3	100	1
PHENIX/Parrot	1	1	1	100

1  100

Table A.37: Structure completeness comparison for the models generated from the 52 synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	6	0	3
ARP(B 5I)	8	0	0	3
i1(5I)	93	93	0	85
PHENIX/Parrot	92	92	4	0

0  93

Table A.38: Structure completeness comparison for the models generated from the 52 synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	20	1	1
ARP(B 5I)	16	0	0	0
i1(5I)	4	3	0	4
PHENIX/Parrot	3	4	6	0



Table A.39: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	0	23	95	98
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	63	0	100	100
ARP(B 5I) $R$ -free	-	0	39	38
i1(5I) $R$ -work	4	0	0	46
i1(5I) $R$ -free	-	59	0	48
PHENIX/Parrot $R$ -work	2	0	49	0
PHENIX/Parrot $R$ -free	-	61	45	0





Table A.40: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	100	14	1	0
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	14	100	0	0
ARP(B 5I) $R$ -free	-	100	2	1
i1(5I) $R$ -work	1	0	100	4
i1(5I) $R$ -free	-	2	100	7
PHENIX/Parrot $R$ -work	0	0	4	100
PHENIX/Parrot $R$ -free	-	1	7	100



Table A.41: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	0	3	90	96
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	18	0	96	100
ARP(B 5I) $R$ -free	-	0	35	37
i1(5I) $R$ -work	2	0	0	18
i1(5I) $R$ -free	-	48	0	19
PHENIX/Parrot $R$ -work	1	0	27	0
PHENIX/Parrot $R$ -free	-	54	23	0



Table A.42: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP <i>R-work</i>	0	20	5	2
ARP <i>R-free</i>	-	-	-	-
ARP(B 5I) <i>R-work</i>	45	0	3	0
ARP(B 5I) <i>R-free</i>	-	0	4	1
i1(5I) <i>R-work</i>	2	0	0	28
i1(5I) <i>R-free</i>	-	10	0	30
PHENIX/Parrot <i>R-work</i>	1	0	23	0
PHENIX/Parrot <i>R-free</i>	-	7	22	0



Table A.43: Structure completeness comparison for the models generated from the 52 synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP	0	23	1	4	3
ARP(B 5I)	22	0	0	3	3
i1(5I)	95	96	0	81	82
PHENIX/Parrot	96	96	17	0	45
PHENIX	97	97	16	42	0



Table A.44: Structure completeness comparison for the models generated from the 52 synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP	100	55	3	0	0
ARP(B 5I)	55	100	4	1	0
i1(5I)	3	4	100	2	2
PHENIX/Parrot	0	1	2	100	13
PHENIX	0	0	2	13	100



Table A.45: Structure completeness comparison for the models generated from the 52 synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP	0	4	0	3	3
ARP(B 5I)	8	0	0	3	3
i1(5I)	90	90	0	77	76
PHENIX/Parrot	94	93	9	0	12
PHENIX	93	93	10	14	0



Table A.46: Structure completeness comparison for the models generated from the 52 synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP	0	19	1	1	0
ARP(B 5I)	14	0	0	0	0
i1(5I)	6	6	0	4	7
PHENIX/Parrot	2	2	8	0	34
PHENIX	4	4	6	28	0



Table A.47: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP <i>R-work</i>	0	19	95	97	97
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	67	0	100	100	100
ARP(B 5I) <i>R-free</i>	-	0	8	6	5
i1(5I) <i>R-work</i>	4	0	0	37	37
i1(5I) <i>R-free</i>	-	91	0	36	34
PHENIX/Parrot <i>R-work</i>	2	0	57	0	33
PHENIX/Parrot <i>R-free</i>	-	94	60	0	38
PHENIX <i>R-work</i>	2	0	57	31	0
PHENIX <i>R-free</i>	-	95	61	44	0



Table A.48: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP $R$ -work	100	14	2	0	1
ARP $R$ -free	-	-	-	-	-
ARP(B 5I) $R$ -work	14	100	0	0	0
ARP(B 5I) $R$ -free	-	100	2	0	0
i1(5I) $R$ -work	2	0	100	6	6
i1(5I) $R$ -free	-	2	100	4	5
PHENIX/Parrot $R$ -work	0	0	6	100	36
PHENIX/Parrot $R$ -free	-	0	4	100	19
PHENIX $R$ -work	1	0	6	36	100
PHENIX $R$ -free	-	0	5	19	100



Table A.49: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP $R$ -work	0	2	89	96	95
ARP $R$ -free	-	-	-	-	-
ARP(B 5I) $R$ -work	21	0	97	100	100
ARP(B 5I) $R$ -free	-	0	2	3	3
i1(5I) $R$ -work	2	0	0	18	17
i1(5I) $R$ -free	-	68	0	16	17
PHENIX/Parrot $R$ -work	1	0	43	0	0
PHENIX/Parrot $R$ -free	-	85	34	0	2
PHENIX $R$ -work	1	0	40	0	0
PHENIX $R$ -free	-	89	39	5	0



Table A.50: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP <i>R-work</i>	0	17	6	2	2
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	46	0	2	0	0
ARP(B 5I) <i>R-free</i>	-	0	5	2	2
i1(5I) <i>R-work</i>	2	0	0	19	20
i1(5I) <i>R-free</i>	-	23	0	20	17
PHENIX/Parrot <i>R-work</i>	1	0	14	0	33
PHENIX/Parrot <i>R-free</i>	-	9	25	0	36
PHENIX <i>R-work</i>	1	0	17	31	0
PHENIX <i>R-free</i>	-	5	22	39	0



### A.3 Original resolutions without the Buccaneer development data sets

Table A.51: Structure completeness comparison for the models generated from the original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	27	28	41	64
ARP(B 5I)	47	0	34	41	75
i1(5I)	64	54	0	50	74
PHENIX/Parrot	48	44	40	0	74
SHELXE/Parrot	30	21	20	20	0



Table A.52: Structure completeness comparison for the models generated from the original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	100	26	8	11	5
ARP(B 5I)	26	100	12	15	3
i1(5I)	8	12	100	9	6
PHENIX/Parrot	11	15	9	100	5
SHELXE/Parrot	5	3	6	5	100



Table A.53: Structure completeness comparison for the models generated from the original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	5	13	13	40
ARP(B 5I)	21	0	17	16	47
i1(5I)	28	20	0	21	52
PHENIX/Parrot	27	20	23	0	49
SHELXE/Parrot	18	11	11	8	0



Table A.54: Structure completeness comparison for the models generated from the original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	21	15	28	24
ARP(B 5I)	26	0	16	25	28
i1(5I)	36	34	0	29	22
PHENIX/Parrot	21	24	17	0	26
SHELXE/Parrot	12	10	9	12	0





Table A.55: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	21	93	32	100
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	43	0	95	42	100
ARP(B 5I) <i>R-free</i>	-	0	86	50	-
i1(5I) <i>R-work</i>	5	1	0	3	99
i1(5I) <i>R-free</i>	-	10	0	3	-
PHENIX/Parrot <i>R-work</i>	45	36	95	0	99
PHENIX/Parrot <i>R-free</i>	-	44	95	0	-
SHELXE/Parrot <i>R-work</i>	0	0	1	1	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.56: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	100	36	2	23	0
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	36	100	4	22	0
ARP(B 5I) <i>R-free</i>	-	100	4	7	-
i1(5I) <i>R-work</i>	2	4	100	1	0
i1(5I) <i>R-free</i>	-	4	100	3	-
PHENIX/Parrot <i>R-work</i>	23	22	1	100	0
PHENIX/Parrot <i>R-free</i>	-	7	3	100	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	100
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.57: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	5	47	4	100
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	4	0	49	6	100
ARP(B 5I) <i>R-free</i>	-	0	56	9	-
i1(5I) <i>R-work</i>	0	0	0	0	95
i1(5I) <i>R-free</i>	-	2	0	0	-
PHENIX/Parrot <i>R-work</i>	4	3	50	0	99
PHENIX/Parrot <i>R-free</i>	-	13	50	0	-
SHELXE/Parrot <i>R-work</i>	0	0	1	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.58: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	17	46	28	0
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	39	0	46	36	0
ARP(B 5I) <i>R-free</i>	-	0	30	40	-
i1(5I) <i>R-work</i>	5	1	0	3	4
i1(5I) <i>R-free</i>	-	8	0	3	-
PHENIX/Parrot <i>R-work</i>	41	33	45	0	0
PHENIX/Parrot <i>R-free</i>	-	31	44	0	-
SHELXE/Parrot <i>R-work</i>	0	0	0	1	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.59: Structure completeness comparison for the models generated from the original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	32	33	40	71
ARP(B 5I)	38	0	33	40	76
i1(5I)	57	53	0	46	78
PHENIX/Parrot	44	43	40	0	75
SHELXE/Parrot	24	17	16	17	0



Table A.60: Structure completeness comparison for the models generated from the original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	100	31	10	16	5
ARP(B 5I)	31	100	14	17	7
i1(5I)	10	14	100	14	6
PHENIX/Parrot	16	17	14	100	8
SHELXE/Parrot	5	7	6	8	100



Table A.61: Structure completeness comparison for the models generated from the original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	5	15	16	42
ARP(B 5I)	16	0	15	15	50
i1(5I)	26	19	0	19	54
PHENIX/Parrot	26	19	23	0	52
SHELXE/Parrot	16	8	8	5	0



Table A.62: Structure completeness comparison for the models generated from the original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP	0	27	17	24	29
ARP(B 5I)	21	0	18	25	26
i1(5I)	32	34	0	26	23
PHENIX/Parrot	17	23	17	0	23
SHELXE/Parrot	8	9	8	11	0



Table A.63: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	16	91	34	100
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	40	0	95	47	100
ARP(B 5I) <i>R-free</i>	-	0	88	47	-
i1(5I) <i>R-work</i>	6	1	0	3	99
i1(5I) <i>R-free</i>	-	10	0	4	-
PHENIX/Parrot <i>R-work</i>	46	34	93	0	100
PHENIX/Parrot <i>R-free</i>	-	41	93	0	-
SHELXE/Parrot <i>R-work</i>	0	0	1	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.64: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	100	44	3	20	0
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	44	100	3	19	0
ARP(B 5I) <i>R-free</i>	-	100	2	12	-
i1(5I) <i>R-work</i>	3	3	100	3	0
i1(5I) <i>R-free</i>	-	2	100	3	-
PHENIX/Parrot <i>R-work</i>	20	19	3	100	0
PHENIX/Parrot <i>R-free</i>	-	12	3	100	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	100
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.65: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	3	47	3	100
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	4	0	54	5	100
ARP(B 5I) <i>R-free</i>	-	0	54	11	-
i1(5I) <i>R-work</i>	1	0	0	0	97
i1(5I) <i>R-free</i>	-	2	0	0	-
PHENIX/Parrot <i>R-work</i>	3	3	47	0	100
PHENIX/Parrot <i>R-free</i>	-	10	49	0	-
SHELXE/Parrot <i>R-work</i>	0	0	1	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.66: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	SHELXE/Parrot
ARP <i>R-work</i>	0	13	44	31	0
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	36	0	42	42	0
ARP(B 5I) <i>R-free</i>	-	0	34	36	-
i1(5I) <i>R-work</i>	5	1	0	3	1
i1(5I) <i>R-free</i>	-	8	0	4	-
PHENIX/Parrot <i>R-work</i>	42	30	46	0	0
PHENIX/Parrot <i>R-free</i>	-	31	44	0	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-



Table A.67: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP	100	32	9	12	15	5	7
ARP(B 5I)	32	100	16	13	19	9	5
i1(5I)	9	16	100	9	11	3	4
PHENIX/Parrot	12	13	9	100	22	4	6
PHENIX	15	19	11	22	100	6	7
SHELXE	5	9	3	4	6	100	8
SHELXE/Parrot	7	5	4	6	7	8	100

3 100

Table A.68: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP	0	17	18	27	23	24	21
ARP(B 5I)	21	0	20	28	27	22	20
i1(5I)	30	28	0	30	32	21	24
PHENIX/Parrot	21	24	19	0	32	20	22
PHENIX	20	21	18	25	0	21	21
SHELXE	9	7	9	9	9	0	25
SHELXE/Parrot	11	10	7	11	11	26	0

0 32

Table A.69: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP <i>R-work</i>	100	35	1	20	20	0	0
ARP <i>R-free</i>	-	-	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	35	100	1	24	22	0	0
ARP(B 5I) <i>R-free</i>	-	100	8	14	9	-	-
i1(5I) <i>R-work</i>	1	1	100	3	4	0	0
i1(5I) <i>R-free</i>	-	8	100	4	3	-	-
PHENIX/Parrot <i>R-work</i>	20	24	3	100	51	0	0
PHENIX/Parrot <i>R-free</i>	-	14	4	100	38	-	-
PHENIX <i>R-work</i>	20	22	4	51	100	0	0
PHENIX <i>R-free</i>	-	9	3	38	100	-	-
SHELXE <i>R-work</i>	0	0	0	0	0	100	39
SHELXE <i>R-free</i>	-	-	-	-	-	-	-
SHELXE/Parrot <i>R-work</i>	0	0	0	0	0	39	100
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-	-	-



Table A.70: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot
ARP <i>R-work</i>	0	15	42	29	30	0	0
ARP <i>R-free</i>	-	-	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	41	0	36	41	41	0	0
ARP(B 5I) <i>R-free</i>	-	0	48	12	15	-	-
i1(5I) <i>R-work</i>	5	0	0	3	2	2	3
i1(5I) <i>R-free</i>	-	14	0	3	4	-	-
PHENIX/Parrot <i>R-work</i>	42	27	41	0	25	0	0
PHENIX/Parrot <i>R-free</i>	-	57	36	0	29	-	-
PHENIX <i>R-work</i>	39	28	39	21	0	0	1
PHENIX <i>R-free</i>	-	59	35	30	0	-	-
SHELXE <i>R-work</i>	0	0	1	0	1	0	19
SHELXE <i>R-free</i>	-	-	-	-	-	-	-
SHELXE/Parrot <i>R-work</i>	0	0	1	1	1	41	0
SHELXE/Parrot <i>R-free</i>	-	-	-	-	-	-	-





## A.4 Synthetic resolutions without Buccaneer development data sets

Table A.71: Structure completeness comparison for the models generated from the synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	21	1	2
ARP(B 5I)	21	0	1	3
i1(5I)	93	94	0	75
PHENIX/Parrot	97	96	23	0



Table A.72: Structure completeness comparison for the models generated from the synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	100	58	5	1
ARP(B 5I)	58	100	5	1
i1(5I)	5	5	100	2
PHENIX/Parrot	1	1	2	100



Table A.73: Structure completeness comparison for the models generated from the synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	4	1	2
ARP(B 5I)	7	0	0	3
i1(5I)	84	84	0	70
PHENIX/Parrot	92	91	16	0



Table A.74: Structure completeness comparison for the models generated from the synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	17	1	0
ARP(B 5I)	14	0	0	0
i1(5I)	9	10	0	5
PHENIX/Parrot	5	5	7	0



Table A.75: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	0	27	93	97
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	60	0	99	100
ARP(B 5I) $R$ -free	-	0	48	45
i1(5I) $R$ -work	6	1	0	37
i1(5I) $R$ -free	-	50	0	38
PHENIX/Parrot $R$ -work	2	0	59	0
PHENIX/Parrot $R$ -free	-	54	56	0



Table A.76: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	100	13	1	0
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	13	100	1	0
ARP(B 5I) $R$ -free	-	100	2	2
i1(5I) $R$ -work	1	1	100	4
i1(5I) $R$ -free	-	2	100	6
PHENIX/Parrot $R$ -work	0	0	4	100
PHENIX/Parrot $R$ -free	-	2	6	100



Table A.77: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP <i>R-work</i>	0	5	86	93
ARP <i>R-free</i>	-	-	-	-
ARP(B 5I) <i>R-work</i>	20	0	92	99
ARP(B 5I) <i>R-free</i>	-	0	43	43
i1(5I) <i>R-work</i>	2	0	0	20
i1(5I) <i>R-free</i>	-	42	0	19
PHENIX/Parrot <i>R-work</i>	0	0	38	0
PHENIX/Parrot <i>R-free</i>	-	48	34	0



Table A.78: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP <i>R-work</i>	0	22	8	4
ARP <i>R-free</i>	-	-	-	-
ARP(B 5I) <i>R-work</i>	41	0	7	1
ARP(B 5I) <i>R-free</i>	-	0	4	2
i1(5I) <i>R-work</i>	4	1	0	18
i1(5I) <i>R-free</i>	-	9	0	19
PHENIX/Parrot <i>R-work</i>	2	0	21	0
PHENIX/Parrot <i>R-free</i>	-	6	22	0



Table A.79: Structure completeness comparison for the models generated from the synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	21	1	2
ARP(B 5I)	25	0	0	3
i1(5I)	95	95	0	76
PHENIX/Parrot	97	95	22	0



Table A.80: Structure completeness comparison for the models generated from the synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	100	54	5	1
ARP(B 5I)	54	100	4	1
i1(5I)	5	4	100	2
PHENIX/Parrot	1	1	2	100



Table A.81: Structure completeness comparison for the models generated from the synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	3	0	2
ARP(B 5I)	9	0	0	3
i1(5I)	86	86	0	72
PHENIX/Parrot	92	91	15	0



Table A.82: Structure completeness comparison for the models generated from the synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP	0	19	0	0
ARP(B 5I)	16	0	0	0
i1(5I)	9	9	0	4
PHENIX/Parrot	5	5	7	0



Table A.83: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	0	28	93	98
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	59	0	99	100
ARP(B 5I) $R$ -free	-	0	48	46
i1(5I) $R$ -work	5	1	0	40
i1(5I) $R$ -free	-	51	0	41
PHENIX/Parrot $R$ -work	2	0	56	0
PHENIX/Parrot $R$ -free	-	53	54	0



Table A.84: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP $R$ -work	100	13	2	0
ARP $R$ -free	-	-	-	-
ARP(B 5I) $R$ -work	13	100	0	0
ARP(B 5I) $R$ -free	-	100	1	1
i1(5I) $R$ -work	2	0	100	4
i1(5I) $R$ -free	-	1	100	5
PHENIX/Parrot $R$ -work	0	0	4	100
PHENIX/Parrot $R$ -free	-	1	5	100



Table A.85: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP <i>R-work</i>	0	5	86	93
ARP <i>R-free</i>	-	-	-	-
ARP(B 5I) <i>R-work</i>	19	0	93	99
ARP(B 5I) <i>R-free</i>	-	0	43	43
i1(5I) <i>R-work</i>	2	0	0	21
i1(5I) <i>R-free</i>	-	42	0	21
PHENIX/Parrot <i>R-work</i>	0	0	37	0
PHENIX/Parrot <i>R-free</i>	-	47	33	0



Table A.86: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot
ARP <i>R-work</i>	0	23	7	4
ARP <i>R-free</i>	-	-	-	-
ARP(B 5I) <i>R-work</i>	40	0	6	1
ARP(B 5I) <i>R-free</i>	-	0	5	3
i1(5I) <i>R-work</i>	4	1	0	18
i1(5I) <i>R-free</i>	-	9	0	20
PHENIX/Parrot <i>R-work</i>	2	0	19	0
PHENIX/Parrot <i>R-free</i>	-	6	21	0





Table A.87: Structure completeness comparison for the models generated from the synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP	0	20	1	2	2
ARP(B 5I)	20	0	0	3	3
i1(5I)	94	95	0	68	69
PHENIX/Parrot	97	96	29	0	43
PHENIX	97	96	28	45	0



Table A.88: Structure completeness comparison for the models generated from the synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP	100	60	5	1	1
ARP(B 5I)	60	100	5	1	1
i1(5I)	5	5	100	3	3
PHENIX/Parrot	1	1	3	100	12
PHENIX	1	1	3	12	100



Table A.89: Structure completeness comparison for the models generated from the synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP	0	4	1	2	2
ARP(B 5I)	8	0	0	3	3
i1(5I)	82	82	0	63	63
PHENIX/Parrot	92	92	21	0	15
PHENIX	92	90	21	16	0



Table A.90: Structure completeness comparison for the models generated from the synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP	0	16	1	0	0
ARP(B 5I)	12	0	0	0	1
i1(5I)	12	12	0	5	6
PHENIX/Parrot	5	4	8	0	28
PHENIX	5	6	7	29	0



Table A.91: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP $R$ -work	0	19	95	97	97
ARP $R$ -free	-	-	-	-	-
ARP(B 5I) $R$ -work	67	0	99	100	100
ARP(B 5I) $R$ -free	-	0	13	5	6
i1(5I) $R$ -work	4	0	0	32	31
i1(5I) $R$ -free	-	84	0	34	33
PHENIX/Parrot $R$ -work	3	0	64	0	33
PHENIX/Parrot $R$ -free	-	94	62	0	41
PHENIX $R$ -work	2	0	64	36	0
PHENIX $R$ -free	-	93	63	43	0



Table A.92: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP $R$ -work	100	13	1	0	1
ARP $R$ -free	-	-	-	-	-
ARP(B 5I) $R$ -work	13	100	1	0	0
ARP(B 5I) $R$ -free	-	100	3	1	1
i1(5I) $R$ -work	1	1	100	4	5
i1(5I) $R$ -free	-	3	100	4	4
PHENIX/Parrot $R$ -work	0	0	4	100	31
PHENIX/Parrot $R$ -free	-	1	4	100	17
PHENIX $R$ -work	1	0	5	31	100
PHENIX $R$ -free	-	1	4	17	100



Table A.93: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP <i>R-work</i>	0	3	88	93	93
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	25	0	94	99	100
ARP(B 5I) <i>R-free</i>	-	0	4	2	2
i1(5I) <i>R-work</i>	2	0	0	16	16
i1(5I) <i>R-free</i>	-	66	0	16	15
PHENIX/Parrot <i>R-work</i>	0	0	47	0	1
PHENIX/Parrot <i>R-free</i>	-	84	43	0	6
PHENIX <i>R-work</i>	1	0	47	1	0
PHENIX <i>R-free</i>	-	84	43	7	0



Table A.94: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	ARP	ARP(B 5I)	i1(5I)	PHENIX/Parrot	PHENIX
ARP <i>R-work</i>	0	17	7	4	5
ARP <i>R-free</i>	-	-	-	-	-
ARP(B 5I) <i>R-work</i>	42	0	5	0	0
ARP(B 5I) <i>R-free</i>	-	0	10	3	3
i1(5I) <i>R-work</i>	3	0	0	16	16
i1(5I) <i>R-free</i>	-	18	0	18	18
PHENIX/Parrot <i>R-work</i>	2	0	17	0	32
PHENIX/Parrot <i>R-free</i>	-	10	19	0	34
PHENIX <i>R-work</i>	1	0	17	35	0
PHENIX <i>R-free</i>	-	9	20	36	0



## A.5 Reproducibility of the comparison experiment

The results of this comparison are reproducible, excluding the execution times that the pipeline variants required to build the protein models, which might be affected by certain factors and differ in each run. Tables A.95, A.96, A.97 and A.98 compare the mean of completeness, R-work/R-free and the execution times for original and synthetic. It is clear from these tables that completeness and R-work/R-free can be reproduced, while execution times can vary across different runs, as happens in Phenix Autobuild.

Table A.95: The mean of the three comparative factors, completeness(%), R-work/R-free and the execution times in minutes for the reproducibility experiment for the original NO-NCS data sets.

Pipeline variant	Completeness	R-work/R-free	Execution time
ARP	94	0.24/0.24	32
ARP(B 5I)	93	0.23/0.26	32
i1(5I)	95	0.26/0.29	4
PHENIX	92	0.24/0.26	71
SHELXE	90	0.45/0.44	66
PHENIX/Parrot	93	0.24/0.26	91
SHELXE/Parrot	92	0.44/0.44	59

Table A.96: The mean of the three comparative factors, completeness(%), R-work/R-free and the execution times in minutes for the main experiment for the original NO-NCS data sets.

Pipeline variant	Completeness	R-work/R-free	Execution time
ARP	94	0.24/0.24	28
ARP(B 5I)	93	0.23/0.26	40
i1(5I)	95	0.26/0.29	4
PHENIX	92	0.24/0.26	101
SHELXE	90	0.45/0.44	65
PHENIX/Parrot	93	0.24/0.26	92
SHELXE/Parrot	92	0.44/0.44	65

Table A.97: The mean of the three comparative factors, completeness(%), R-work/R-free and the execution times in minutes for the reproducibility experiment for the synthetic NO-NCS data sets.

Pipeline variant	Completeness	R-work/R-free	Execution time
ARP	2	0.21/0.2	30
ARP(B 5I)	1	0.19/0.4	32
i1(5I)	62	0.32/0.4	5
PHENIX	45	0.29/0.37	49
PHENIX/Parrot	43	0.29/0.38	77

Table A.98: The mean of the three comparative factors, completeness(%), R-work/R-free and the execution times in minutes for the main experiment for the synthetic NO-NCS data sets.

Pipeline variant	Completeness	R-work/R-free	Execution time
ARP	2	0.21/0.2	24
ARP(B 5I)	0	0.19/0.39	45
i1(5I)	63	0.32/0.4	5
PHENIX	45	0.29/0.37	92
PHENIX/Parrot	43	0.29/0.38	95

## A.6 PDB codes used in the comparison

The following PDB codes have been used in the comparison (the omitted data sets are marked with an asterisk): 1o6a\*, 1vjf\*, 1vjn\*, 1vjr\*, 1vjv\*, 1vjx\*, 1vjz\*, 1vk2\*, 1vk3\*, 1vk4\*, 1vk8\*, 1vk9\*, 1vkb\*, 1vkd\*, 1vkh\*, 1vkm\*, 1vkn\*, 1vku\*, 1vky\*, 1vkz\*, 1vl0\*, 1vl4\*, 1vl5\*, 1vl6\*, 1vlc\*, 1vli\*, 1vll\*, 1vlm\*, 1vlo\*, 1vlu\*, 1vm8, 1vme\*, 1vmf\*, 1vmg\*, 1vmi\*, 1vp4\*, 1vp7\*, 1vp8\*, 1vpb\*, 1vpm\*, 1vpy\*, 1vpz\*, 1vqr\*, 1vqs\*, 1vqy\*, 1vqz\*, 1vr0\*, 1vr3\*, 1vr5\*, 1vr8\*, 1vra, 1vrb\*, 1z82\*, 1z85\*, 1zbt, 1zkg, 1zko, 1ztc, 1zy9, 1zyb, 2a2m, 2a3n, 2a6a, 2a6b, 2a9v, 2aam, 2afb, 2aj6, 2aj7, 2ajr, 2aml, 2anu, 2ash, 2avn, 2awa, 2b8m, 2ess, 2etd, 2eth, 2etj, 2ets, 2f4l, 2f4p, 2fcl, 2fea, 2ffj, 2fg0, 2fg9, 2fna, 2fno, 2fqp, 2fur, 2fzt, 2g0t, 2gb5, 2gfg, 2ghr, 2ghs, 2gjj, 2glz, 2gm6, 2gno, 2gnr, 2go7, 2gpj, 2gvh, 2gvk, 2h1q, 2hag, 2hcf, 2hdo, 2hh6, 2hhz, 2hi0, 2hoe, 2hq7, 2hr2, 2hsb, 2hti, 2huh, 2huj, 2hx1, 2hxy, 2hyt, 2i51, 2i5i, 2i8d, 2i9w, 2ia7, 2ich, 2ifx, 2ig6, 2ii1, 2iiu, 2ilb, 2inb, 2isb, 2it9, 2itb, 2nlv, 2nuj, 2nvw, 2nyh, 2o08, 2o1q, 2o2g, 2o2x, 2o3l, 2o5r, 2o62, 2o7t, 2o8q, 2obn, 2obp, 2oc5, 2oc6, 2od4, 2od5, 2od6, 2ogi, 2oh1, 2oh3, 2okc, 2okf, 2ooc, 2ooj, 2op5, 2opk, 2opl, 2ord, 2osd, 2otm, 2ou6, 2ouw, 2owp, 2oyo, 2ozg, 2ozj, 2p10, 2p1a, 2p4g, 2p4o, 2p7i, 2p8j, 2p97, 2pbl, 2pc1, 2pg3, 2pg4, 2pgc, 2pim, 2pke, 2pn1, 2pn2, 2pnk, 2ppv, 2pr7, 2pr, 2prv, 2prx, 2pv4 and 2pw4.

## A.7 SHELXE results for using default and optimised solvent fraction for the original data sets without the Buccaneer development data sets

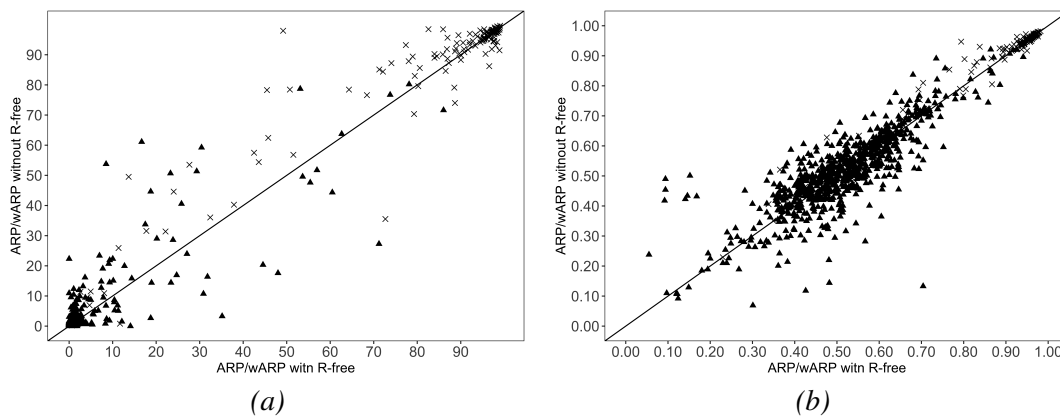
Table A.99: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants. SHELXE were run using the default solvent fraction, which is 0.45 and an optimised solvent fraction (multiple protein models were built using the solvent fraction range from 0 to 1 with an increased step of 0.01 and selected the protein model with the highest correlation coefficient).

Pipeline variant	ARP	ARP(B 5I)	iI(5I)	PHENIX/Parrot	PHENIX	SHELXE	SHELXE/Parrot	SHELXE/Parrot (optimised)	SHELXE (optimised)
ARP	0	6	15	11	14	45	40	42	60
ARP(B 5I)	24	0	20	16	16	53	53	54	74
iI(5I)	28	17	0	16	16	56	48	46	72
PHENIX/Parrot	28	20	26	0	14	61	55	55	72
PHENIX	28	18	23	7	0	57	51	53	72
SHELXE	17	7	11	7	7	0	9	20	51
SHELXE/Parrot	21	12	17	5	10	32	0	27	57
SHELXE/Parrot (optimised)	18	11	14	7	10	23	8	0	50
SHELXE (optimised)	14	6	6	4	4	9	6	0	0

0  74



## A.8 Comparison of ARP/wARP run with and without R-free



× Original resolution ▲ Truncated resolution

Figure A.1: Comparison of ARP/wARP with and without R-free flag for both original and synthetic NO-NCS data sets. Points above the diagonal indicate that ARP/wARP was run without R-free is better than when R-free flag is used. (a) The comparison of structure completeness. (b) The comparison of structure correlation

# Pairwise running of automated crystallographic model-building pipelines (additional results)

## B.1 Experimental results for the original data sets used in Buccaneer development

Table B.1: Complete and intermediate models produced by the 23 pipeline variants for the 52 original data sets, where (T) and (C) denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	HA-NCS			MR-NCS			NO-NCS		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
A	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
A→P*	51	1(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
A→B	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
B	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
B→P*	51	0(T) 0(C)	1	51	0(T) 0(C)	1	50	1(T) 0(C)	1
P*	51	1(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
P*→A	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
P*→B	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
S*	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
S*→A	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
S*→B	52	0(T) 0(C)	0	52	0(T) 0(C)	0	52	0(T) 0(C)	0
S*→P*	52	0(T) 0(C)	0	51	1(T) 0(C)	0	52	0(T) 0(C)	0
A→P	-	-	-	-	-	-	52	0(T) 0(C)	0
B→P	-	-	-	-	-	-	51	0(T) 0(C)	1
P	-	-	-	-	-	-	52	0(T) 0(C)	0
P→A	-	-	-	-	-	-	52	0(T) 0(C)	0
P→B	-	-	-	-	-	-	52	0(T) 0(C)	0
S	-	-	-	-	-	-	52	0(T) 0(C)	0
S→A	-	-	-	-	-	-	52	0(T) 0(C)	0
S→B	-	-	-	-	-	-	52	0(T) 0(C)	0
S*→P	-	-	-	-	-	-	52	0(T) 0(C)	0
S→P*	-	-	-	-	-	-	52	0(T) 0(C)	0
S→P	-	-	-	-	-	-	52	0(T) 0(C)	0

Models used in the comparison: 51 HA-NCS, 51 MR-NCS and 51 NO-NCS.

Table B.2: Structure completeness comparison for the models generated from the 52 original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	18	20	27	24	29	35	14	71	53	29	35
A→B	69	0	67	57	45	69	73	31	92	80	61	71
A→P*	63	22	0	47	33	45	61	14	94	71	39	57
B	59	31	43	0	20	43	49	22	90	69	41	55
B→P*	61	41	53	67	0	63	55	29	94	75	49	75
P*	57	27	29	45	25	0	47	14	82	65	39	45
P*→A	45	20	14	37	27	35	0	12	82	61	27	49
P*→B	80	45	67	71	55	76	76	0	96	80	65	82
S*	25	8	2	8	2	10	14	4	0	41	6	8
S*→A	20	14	12	18	16	29	20	12	55	0	14	29
S*→B	59	25	45	47	31	51	51	22	94	75	0	55
S*→P*	51	25	31	43	16	29	39	18	86	65	35	0



Table B.3: Structure completeness comparison for the models generated from the 52 original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	100	14	18	14	16	14	20	6	4	27	12	14
A→B	14	100	12	12	14	4	8	24	0	6	14	4
A→P*	18	12	100	10	14	25	25	20	4	18	16	12
B	14	12	10	100	14	12	14	8	2	14	12	2
B→P*	16	14	14	14	100	12	18	16	4	10	20	10
P*	14	4	25	12	12	100	18	10	8	6	10	25
P*→A	20	8	25	14	18	18	100	12	4	20	22	12
P*→B	6	24	20	8	16	10	12	100	0	8	14	0
S*	4	0	4	2	4	8	4	0	100	4	0	6
S*→A	27	6	18	14	10	6	20	8	4	100	12	6
S*→B	12	14	16	12	20	10	22	14	0	12	100	10
S*→P*	14	4	12	2	10	25	12	0	6	6	10	100



Table B.4: Structure completeness comparison for the models generated from the 52 original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	10	0	14	10	14	12	10	57	33	6	14
A→B	29	0	18	22	16	29	29	12	75	49	24	33
A→P*	35	14	0	25	12	12	27	10	78	43	18	25
B	35	18	18	0	12	22	25	10	75	43	14	24
B→P*	41	22	25	29	0	25	31	16	75	47	25	33
P*	29	20	14	22	8	0	25	10	69	39	22	16
P*→A	20	14	4	24	8	14	0	6	69	37	12	25
P*→B	37	22	29	27	22	37	37	0	80	49	31	43
S*	16	4	0	2	0	8	8	4	0	35	4	4
S*→A	10	6	0	4	6	8	6	6	41	0	4	14
S*→B	35	10	18	16	10	20	25	12	80	43	0	31
S*→P*	33	16	12	22	10	14	24	10	73	43	16	0



Table B.5: Structure completeness comparison for the models generated from the 52 original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	8	20	14	14	16	24	4	14	20	24	22
A→B	39	0	49	35	29	39	43	20	18	31	37	37
A→P*	27	8	0	22	22	33	33	4	16	27	22	31
B	24	14	25	0	8	22	24	12	16	25	27	31
B→P*	20	20	27	37	0	37	24	14	20	27	24	41
P*	27	8	16	24	18	0	22	4	14	25	18	29
P*→A	25	6	10	14	20	22	0	6	14	24	16	24
P*→B	43	24	37	43	33	39	39	0	16	31	33	39
S*	10	4	2	6	2	2	6	0	0	6	2	4
S*→A	10	8	12	14	10	22	14	6	14	0	10	16
S*→B	24	16	27	31	22	31	25	10	14	31	0	24
S*→P*	18	10	20	22	6	16	16	8	14	22	20	0



Table B.6: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	92	25	96	39	51	31	86	100	49	94	47
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	4	0	2	53	2	6	4	25	98	4	53	4
A→B <sub>R-free</sub>	-	0	12	47	4	6	24	24	-	45	53	4
A→P* <sub>R-work</sub>	57	98	0	98	51	73	51	92	100	71	100	76
A→P* <sub>R-free</sub>	-	86	0	90	43	61	80	82	-	90	92	63
B <sub>R-work</sub>	2	22	0	0	0	0	2	12	100	2	27	2
B <sub>R-free</sub>	-	29	8	0	2	6	14	25	-	43	37	6
B→P* <sub>R-work</sub>	41	86	27	98	0	45	39	90	100	57	96	57
B→P* <sub>R-free</sub>	-	90	43	96	0	47	69	86	-	76	94	53
P* <sub>R-work</sub>	35	86	12	98	18	0	27	88	100	45	96	31
P* <sub>R-free</sub>	-	92	24	90	31	0	67	84	-	82	98	35
P*→A <sub>R-work</sub>	43	96	31	96	45	57	0	94	100	51	98	59
P*→A <sub>R-free</sub>	-	73	6	76	20	18	0	69	-	61	73	16
P*→B <sub>R-work</sub>	8	37	6	61	4	8	4	0	100	18	55	6
P*→B <sub>R-free</sub>	-	41	12	61	8	12	24	0	-	45	55	14
S* <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	27	94	16	92	29	41	20	80	100	0	96	35
S*→A <sub>R-free</sub>	-	51	4	55	8	8	18	51	-	0	53	12
S*→B <sub>R-work</sub>	2	22	0	37	0	2	2	18	100	2	0	0
S*→B <sub>R-free</sub>	-	29	6	45	0	2	20	22	-	43	0	2
S*→P* <sub>R-work</sub>	29	92	10	96	16	22	25	86	100	37	94	0
S*→P* <sub>R-free</sub>	-	94	22	94	22	22	65	82	-	78	94	0



Table B.7: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <i>R-work</i>	100	4	18	2	20	14	25	6	0	24	4	24
A <i>R-free</i>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	4	100	0	25	12	8	0	37	2	2	25	4
A→B <i>R-free</i>	-	100	2	24	6	2	4	35	-	4	18	2
A→P* <i>R-work</i>	18	0	100	2	22	16	18	2	0	14	0	14
A→P* <i>R-free</i>	-	2	100	2	14	16	14	6	-	6	2	16
B <i>R-work</i>	2	25	2	100	2	2	2	27	0	6	35	2
B <i>R-free</i>	-	24	2	100	2	4	10	14	-	2	18	0
B→P* <i>R-work</i>	20	12	22	2	100	37	16	6	0	14	4	27
B→P* <i>R-free</i>	-	6	14	2	100	22	12	6	-	16	6	25
P* <i>R-work</i>	14	8	16	2	37	100	16	4	0	14	2	47
P* <i>R-free</i>	-	2	16	4	22	100	16	4	-	10	0	43
P*→A <i>R-work</i>	25	0	18	2	16	16	100	2	0	29	0	16
P*→A <i>R-free</i>	-	4	14	10	12	16	100	8	-	22	8	20
P*→B <i>R-work</i>	6	37	2	27	6	4	2	100	0	2	27	8
P*→B <i>R-free</i>	-	35	6	14	6	4	8	100	-	4	24	4
S* <i>R-work</i>	0	2	0	0	0	0	0	0	100	0	0	0
S* <i>R-free</i>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <i>R-work</i>	24	2	14	6	14	14	29	2	0	100	2	27
S*→A <i>R-free</i>	-	4	6	2	16	10	22	4	-	100	4	10
S*→B <i>R-work</i>	4	25	0	35	4	2	0	27	0	2	100	6
S*→B <i>R-free</i>	-	18	2	18	6	0	8	24	-	4	100	4
S*→P* <i>R-work</i>	24	4	14	2	27	47	16	8	0	27	6	100
S*→P* <i>R-free</i>	-	2	16	0	25	43	20	4	-	10	4	100



Table B.8: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	53	14	59	10	12	4	41	100	22	55	14
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	0	0	0	12	0	2	0	10	96	2	6	0
A→B <sub>R-free</sub>	-	0	2	10	0	2	8	8	-	33	4	0
A→P* <sub>R-work</sub>	2	57	0	67	0	6	4	49	100	16	71	4
A→P* <sub>R-free</sub>	-	45	0	61	0	4	20	39	-	49	59	0
B <sub>R-work</sub>	0	6	0	0	0	0	0	6	96	2	4	0
B <sub>R-free</sub>	-	8	2	0	0	0	6	4	-	31	4	0
B→P* <sub>R-work</sub>	4	51	4	61	0	4	6	45	100	16	59	2
B→P* <sub>R-free</sub>	-	53	6	63	0	2	22	41	-	39	59	2
P* <sub>R-work</sub>	4	39	4	49	0	0	2	22	100	12	49	0
P* <sub>R-free</sub>	-	39	6	51	0	0	16	27	-	43	49	2
P*→A <sub>R-work</sub>	4	47	12	61	12	14	0	43	100	18	53	14
P*→A <sub>R-free</sub>	-	18	4	31	0	0	0	18	-	31	29	0
P*→B <sub>R-work</sub>	2	16	2	22	0	0	0	0	100	6	20	0
P*→B <sub>R-free</sub>	-	14	6	22	0	0	10	0	-	37	18	0
S* <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	2	35	0	51	0	4	2	29	100	0	49	4
S*→A <sub>R-free</sub>	-	12	0	20	0	0	0	10	-	0	16	0
S*→B <sub>R-work</sub>	0	6	0	4	0	2	0	6	96	2	0	0
S*→B <sub>R-free</sub>	-	4	0	6	0	2	8	6	-	29	0	0
S*→P* <sub>R-work</sub>	4	41	4	53	0	2	2	25	100	12	49	0
S*→P* <sub>R-free</sub>	-	37	8	53	0	4	14	25	-	39	43	0



Table B.9: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	39	12	37	29	39	27	45	0	27	39	33
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	4	0	2	41	2	4	4	16	2	2	47	4
A→B <sub>R-free</sub>	-	0	10	37	4	4	16	16	-	12	49	4
A→P* <sub>R-work</sub>	55	41	0	31	51	67	47	43	0	55	29	73
A→P* <sub>R-free</sub>	-	41	0	29	43	57	61	43	-	41	33	63
B <sub>R-work</sub>	2	16	0	0	0	0	2	6	4	0	24	2
B <sub>R-free</sub>	-	22	6	0	2	6	8	22	-	12	33	6
B→P* <sub>R-work</sub>	37	35	24	37	0	41	33	45	0	41	37	55
B→P* <sub>R-free</sub>	-	37	37	33	0	45	47	45	-	37	35	51
P* <sub>R-work</sub>	31	47	8	49	18	0	25	67	0	33	47	31
P* <sub>R-free</sub>	-	53	18	39	31	0	51	57	-	39	49	33
P*→A <sub>R-work</sub>	39	49	20	35	33	43	0	51	0	33	45	45
P*→A <sub>R-free</sub>	-	55	2	45	20	18	0	51	-	29	43	16
P*→B <sub>R-work</sub>	6	22	4	39	4	8	4	0	0	12	35	6
P*→B <sub>R-free</sub>	-	27	6	39	8	12	14	0	-	8	37	14
S* <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	25	59	16	41	29	37	18	51	0	0	47	31
S*→A <sub>R-free</sub>	-	39	4	35	8	8	18	41	-	0	37	12
S*→B <sub>R-work</sub>	2	16	0	33	0	0	2	12	4	0	0	0
S*→B <sub>R-free</sub>	-	25	6	39	0	0	12	16	-	14	0	2
S*→P* <sub>R-work</sub>	25	51	6	43	16	20	24	61	0	25	45	0
S*→P* <sub>R-free</sub>	-	57	14	41	22	18	51	57	-	39	51	0

0 73



Table B.10: Structure completeness comparison for the models generated from the 52 original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	22	25	25	22	37	27	14	73	51	29	39
A→B	65	0	61	45	37	65	75	31	94	78	57	61
A→P*	51	31	0	41	24	37	49	16	94	67	43	45
B	59	29	47	0	25	49	55	20	94	73	41	61
B→P*	61	43	63	61	0	67	65	35	92	78	55	80
P*	49	29	37	37	25	0	47	16	86	59	39	47
P*→A	43	18	24	25	22	35	0	8	80	55	31	43
P*→B	78	53	65	65	51	69	82	0	96	82	67	80
S*	22	6	2	6	4	8	14	4	0	39	4	8
S*→A	20	16	18	20	12	31	20	10	57	0	14	31
S*→B	59	25	43	39	33	53	49	16	96	71	0	51
S*→P*	47	29	39	35	12	31	45	14	88	59	37	0



Table B.11: Structure completeness comparison for the models generated from the 52 original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	100	14	24	16	18	14	29	8	6	29	12	14
A→B	14	100	8	25	20	6	8	16	0	6	18	10
A→P*	24	8	100	12	14	25	27	20	4	16	14	16
B	16	25	12	100	14	14	20	16	0	8	20	4
B→P*	18	20	14	14	100	8	14	14	4	10	12	8
P*	14	6	25	14	8	100	18	16	6	10	8	22
P*→A	29	8	27	20	14	18	100	10	6	25	20	12
P*→B	8	16	20	16	14	16	10	100	0	8	18	6
S*	6	0	4	0	4	6	6	0	100	4	0	4
S*→A	29	6	16	8	10	10	25	8	4	100	16	10
S*→B	12	18	14	20	12	8	20	18	0	16	100	12
S*→P*	14	10	16	4	8	22	12	6	4	10	12	100



Table B.12: Structure completeness comparison for the models generated from the 52 original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	8	6	10	10	10	8	8	61	35	10	14
A→B	35	0	22	14	12	31	31	8	75	47	18	35
A→P*	31	16	0	18	12	10	27	10	82	49	24	18
B	35	16	20	0	10	22	33	10	76	45	18	27
B→P*	43	24	29	25	0	31	35	10	78	49	25	43
P*	31	16	18	20	10	0	25	12	76	43	22	20
P*→A	16	10	8	16	10	10	0	4	65	39	12	20
P*→B	35	18	24	25	18	29	35	0	82	49	31	39
S*	10	2	0	2	2	4	8	4	0	31	0	6
S*→A	6	4	0	6	6	8	6	6	41	0	6	12
S*→B	37	8	18	12	8	25	27	8	78	45	0	27
S*→P*	31	20	12	20	10	12	24	12	75	45	20	0



Table B.13: Structure completeness comparison for the models generated from the 52 original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	14	20	16	12	27	20	6	12	16	20	25
A→B	29	0	39	31	25	33	43	24	20	31	39	25
A→P*	20	16	0	24	12	27	22	6	12	18	20	27
B	24	14	27	0	16	27	22	10	18	27	24	33
B→P*	18	20	33	35	0	35	29	25	14	29	29	37
P*	18	14	20	18	16	0	22	4	10	16	18	27
P*→A	27	8	16	10	12	25	0	4	16	16	20	24
P*→B	43	35	41	39	33	39	47	0	14	33	35	41
S*	12	4	2	4	2	4	6	0	0	8	4	2
S*→A	14	12	18	14	6	24	14	4	16	0	8	20
S*→B	22	18	25	27	25	27	22	8	18	25	0	24
S*→P*	16	10	27	16	2	20	22	2	14	14	18	0



Table B.14: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <i>R-work</i>	0	88	25	92	39	57	22	84	100	43	90	45
A <i>R-free</i>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	6	0	0	39	4	4	0	27	98	8	49	4
A→B <i>R-free</i>	-	0	4	37	6	4	18	24	-	43	49	8
A→P* <i>R-work</i>	59	98	0	96	55	78	47	92	100	69	100	75
A→P* <i>R-free</i>	-	92	0	92	49	61	82	86	-	90	96	63
B <i>R-work</i>	4	25	0	0	2	2	0	18	100	4	33	6
B <i>R-free</i>	-	33	8	0	6	6	18	25	-	49	43	8
B→P* <i>R-work</i>	49	84	31	96	0	51	39	88	100	55	92	63
B→P* <i>R-free</i>	-	88	33	92	0	51	69	88	-	78	90	63
P* <i>R-work</i>	27	86	10	92	20	0	20	88	100	43	96	29
P* <i>R-free</i>	-	88	16	86	24	0	65	86	-	80	98	39
P*→A <i>R-work</i>	51	98	39	100	43	69	0	96	100	47	100	65
P*→A <i>R-free</i>	-	71	6	71	24	16	0	71	-	61	75	22
P*→B <i>R-work</i>	8	39	6	53	6	6	2	0	100	14	55	6
P*→B <i>R-free</i>	-	43	6	55	8	8	24	0	-	45	57	12
S* <i>R-work</i>	0	0	0	0	0	0	0	0	0	0	0	0
S* <i>R-free</i>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <i>R-work</i>	35	90	16	92	24	39	12	84	100	0	94	39
S*→A <i>R-free</i>	-	49	2	49	10	8	16	49	-	0	51	14
S*→B <i>R-work</i>	4	14	0	27	2	0	0	18	100	4	0	0
S*→B <i>R-free</i>	-	25	2	31	2	0	18	14	-	45	0	4
S*→P* <i>R-work</i>	31	88	10	94	16	27	24	88	100	41	92	0
S*→P* <i>R-free</i>	-	92	18	92	16	27	63	80	-	78	94	0



Table B.15: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	100	6	16	4	12	16	27	8	0	22	6	24
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	6	100	2	35	12	10	2	33	2	2	37	8
A→B <sub>R-free</sub>	-	100	4	29	6	8	12	33	-	8	25	0
A→P* <sub>R-work</sub>	16	2	100	4	14	12	14	2	0	16	0	16
A→P* <sub>R-free</sub>	-	4	100	0	18	24	12	8	-	8	2	20
B <sub>R-work</sub>	4	35	4	100	2	6	0	29	0	4	39	0
B <sub>R-free</sub>	-	29	0	100	2	8	12	20	-	2	25	0
B→P* <sub>R-work</sub>	12	12	14	2	100	29	18	6	0	22	6	22
B→P* <sub>R-free</sub>	-	6	18	2	100	25	8	4	-	12	8	22
P* <sub>R-work</sub>	16	10	12	6	29	100	12	6	0	18	4	43
P* <sub>R-free</sub>	-	8	24	8	25	100	20	6	-	12	2	33
P*→A <sub>R-work</sub>	27	2	14	0	18	12	100	2	0	41	0	12
P*→A <sub>R-free</sub>	-	12	12	12	8	20	100	6	-	24	8	16
P*→B <sub>R-work</sub>	8	33	2	29	6	6	2	100	0	2	27	6
P*→B <sub>R-free</sub>	-	33	8	20	4	6	6	100	-	6	29	8
S* <sub>R-work</sub>	0	2	0	0	0	0	0	0	100	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	22	2	16	4	22	18	41	2	0	100	2	20
S*→A <sub>R-free</sub>	-	8	8	2	12	12	24	6	-	100	4	8
S*→B <sub>R-work</sub>	6	37	0	39	6	4	0	27	0	2	100	8
S*→B <sub>R-free</sub>	-	25	2	25	8	2	8	29	-	4	100	2
S*→P* <sub>R-work</sub>	24	8	16	0	22	43	12	6	0	20	8	100
S*→P* <sub>R-free</sub>	-	0	20	0	22	33	16	8	-	8	2	100



Table B.16: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	47	6	53	8	10	0	43	100	18	55	8
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	0	0	0	10	0	0	0	8	96	2	6	0
A→B <sub>R-free</sub>	-	0	2	4	2	0	4	8	-	29	4	0
A→P* <sub>R-work</sub>	0	61	0	61	4	4	0	55	100	12	69	4
A→P* <sub>R-free</sub>	-	51	0	55	2	4	18	39	-	45	55	0
B <sub>R-work</sub>	0	4	0	0	0	0	0	8	98	2	6	0
B <sub>R-free</sub>	-	8	2	0	2	0	4	6	-	31	10	0
B→P* <sub>R-work</sub>	4	49	6	53	0	4	2	47	100	18	57	2
B→P* <sub>R-free</sub>	-	51	10	45	0	10	24	41	-	45	57	2
P* <sub>R-work</sub>	2	39	4	39	2	0	0	27	100	10	49	0
P* <sub>R-free</sub>	-	35	6	41	2	0	10	31	-	39	49	2
P*→A <sub>R-work</sub>	2	45	10	53	14	14	0	51	100	22	53	14
P*→A <sub>R-free</sub>	-	24	4	24	2	0	0	16	-	29	33	0
P*→B <sub>R-work</sub>	0	16	2	18	2	0	0	0	98	6	16	0
P*→B <sub>R-free</sub>	-	18	4	16	2	0	8	0	-	35	16	0
S* <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	0	37	0	41	4	0	0	33	100	0	49	4
S*→A <sub>R-free</sub>	-	12	0	12	0	0	0	8	-	0	18	0
S*→B <sub>R-work</sub>	0	4	0	6	0	0	0	4	98	2	0	0
S*→B <sub>R-free</sub>	-	4	0	4	2	0	4	4	-	27	0	0
S*→P* <sub>R-work</sub>	2	39	4	37	2	0	0	27	100	12	49	0
S*→P* <sub>R-free</sub>	-	37	4	43	2	0	12	29	-	39	51	0



Table B.17: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	41	20	39	31	47	22	41	0	25	35	37
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	6	0	0	29	4	4	0	20	2	6	43	4
A→B <sub>R-free</sub>	-	0	2	33	4	4	14	16	-	14	45	8
A→P* <sub>R-work</sub>	59	37	0	35	51	75	47	37	0	57	31	71
A→P* <sub>R-free</sub>	-	41	0	37	47	57	65	47	-	45	41	63
B <sub>R-work</sub>	4	22	0	0	2	2	0	10	2	2	27	6
B <sub>R-free</sub>	-	25	6	0	4	6	14	20	-	18	33	8
B→P* <sub>R-work</sub>	45	35	25	43	0	47	37	41	0	37	35	61
B→P* <sub>R-free</sub>	-	37	24	47	0	41	45	47	-	33	33	61
P* <sub>R-work</sub>	25	47	6	53	18	0	20	61	0	33	47	29
P* <sub>R-free</sub>	-	53	10	45	22	0	55	55	-	41	49	37
P*→A <sub>R-work</sub>	49	53	29	47	29	55	0	45	0	25	47	51
P*→A <sub>R-free</sub>	-	47	2	47	22	16	0	55	-	31	41	22
P*→B <sub>R-work</sub>	8	24	4	35	4	6	2	0	2	8	39	6
P*→B <sub>R-free</sub>	-	25	2	39	6	8	16	0	-	10	41	12
S* <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	35	53	16	51	20	39	12	51	0	0	45	35
S*→A <sub>R-free</sub>	-	37	2	37	10	8	16	41	-	0	33	14
S*→B <sub>R-work</sub>	4	10	0	22	2	0	0	14	2	2	0	0
S*→B <sub>R-free</sub>	-	22	2	27	0	0	14	10	-	18	0	4
S*→P* <sub>R-work</sub>	29	49	6	57	14	27	24	61	0	29	43	0
S*→P* <sub>R-free</sub>	-	55	14	49	14	27	51	51	-	39	43	0

0 75

APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.18: Structure completeness comparison for the models generated from the 52 original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B	S	S→A	S→B	S*	S*→A	S*→B	S*→P*	S*→P	S→P*	S→P
A	0	16	22	22	27	31	27	35	31	22	41	37	14	76	71	35	75	51	29	41	37	41	41
A→B	67	0	55	55	55	45	39	61	61	24	57	61	27	84	82	49	86	76	59	59	55	61	57
A→P*	63	35	0	25	55	45	41	43	63	20	49	67	24	92	82	59	94	67	53	59	55	51	53
A→P	59	27	25	0	51	49	39	39	55	20	43	59	22	88	80	57	90	69	51	57	49	59	55
B	61	31	31	37	0	25	25	39	53	22	45	53	10	88	80	41	90	73	45	49	49	47	47
B→P*	53	41	37	43	57	0	24	53	55	31	47	53	22	88	73	57	90	69	53	57	55	57	61
B→P	53	41	45	47	55	25	0	59	57	27	53	57	24	88	73	53	90	67	51	57	59	63	65
P*	53	29	29	33	51	37	31	0	49	18	39	45	14	78	63	53	84	59	49	43	45	45	37
P*→A	45	24	12	20	35	31	27	31	0	20	41	33	10	80	69	41	84	55	31	45	49	47	43
P*→B	73	61	61	63	67	57	61	71	73	0	65	75	37	88	82	65	94	75	69	75	73	76	75
P	49	31	25	31	41	37	31	27	47	18	0	43	10	80	65	47	84	61	45	35	37	37	37
P→A	43	27	22	27	33	31	25	37	41	18	39	0	6	84	65	43	86	59	31	45	41	43	43
P→B	80	57	63	65	75	61	57	71	75	39	78	76	0	96	88	65	98	86	69	78	82	76	82
S	20	14	6	8	6	8	10	10	18	6	12	14	4	0	51	14	37	47	14	8	12	10	18
S→A	8	14	8	10	16	18	16	29	12	12	25	14	6	45	0	18	49	20	8	24	33	33	31
S→B	59	35	29	29	45	29	29	37	49	24	45	49	10	86	73	0	86	69	43	43	47	39	51
S*	22	12	0	2	6	6	10	8	12	6	10	12	2	53	49	12	0	43	12	6	14	12	14
S*→A	22	14	16	16	22	22	20	31	22	18	31	22	6	51	49	24	53	0	18	33	31	33	39
S*→B	57	27	33	33	45	41	37	45	49	22	49	55	22	82	76	47	88	69	0	43	41	43	45
S*→P*	47	31	31	27	41	35	31	35	41	24	51	47	16	88	67	47	86	59	49	0	35	39	45
S*→P	45	29	31	35	43	27	24	41	41	18	43	47	12	82	59	47	80	55	49	45	0	51	45
S→P*	49	31	25	24	41	35	27	35	43	14	37	51	20	82	63	53	82	57	49	27	33	0	33
S→P	49	33	24	31	41	27	24	33	45	16	39	45	14	76	57	43	76	53	47	39	37	41	0



Table B.19: Structure completeness comparison for the models generated from the 52 original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B	S	S→A	S→B	S*	S*→A	S*→B	S*→P*	S*→P	S→P*	S→P	
A	100	18	16	20	12	16	20	12	24	6	10	20	6	4	22	6	4	27	14	12	18	10	10	
A→B	18	100	10	18	14	14	20	10	16	16	12	12	16	2	4	16	2	10	14	10	16	8	10	
A→P*	16	10	100	49	14	18	14	27	25	20	25	12	14	2	10	12	6	18	14	10	14	24	24	
A→P	20	18	49	100	12	8	14	27	25	18	25	14	14	4	10	14	8	16	16	16	16	18	14	
B	12	14	14	12	100	18	20	10	12	12	14	14	16	6	4	14	4	6	10	10	8	12	12	
B→P*	16	14	18	8	18	100	51	10	14	12	16	16	18	4	10	14	4	10	6	8	18	8	12	
B→P	20	20	14	14	20	51	100	10	16	12	16	18	20	2	12	18	0	14	12	12	18	10	12	
P*	12	10	27	27	10	10	10	100	20	12	33	18	16	12	8	10	8	10	6	22	14	20	29	
P*→A	24	16	25	25	12	14	16	20	100	8	12	25	16	2	20	10	4	24	20	14	10	10	12	
P*→B	6	16	20	18	12	12	12	12	8	100	18	8	24	6	6	12	0	8	10	2	10	10	10	
P	10	12	25	25	14	16	16	33	12	18	100	18	12	8	10	8	6	8	6	14	20	25	24	
P→A	20	12	12	14	14	16	18	18	25	8	18	100	18	2	22	8	2	20	14	8	12	6	12	
P→B	6	16	14	14	16	18	20	16	16	24	12	18	100	0	6	25	0	8	10	6	6	4	4	
S	4	2	2	4	6	4	2	12	2	6	8	2	0	100	4	0	10	2	4	4	6	8	6	
S→A	22	4	10	10	4	10	12	8	20	6	10	22	6	4	100	10	2	31	16	10	8	4	12	
S→B	6	16	12	14	14	14	18	10	10	12	8	8	25	0	10	100	2	8	10	10	6	8	6	
S*	4	2	6	8	4	4	4	0	8	4	0	6	2	0	10	2	2	100	4	0	8	6	6	10
S*→A	27	10	18	16	6	10	14	10	24	8	8	20	8	2	31	8	4	100	14	8	14	10	8	
S*→B	14	14	14	16	10	6	12	6	20	10	6	14	10	4	16	10	0	14	100	8	10	8	8	
S*→P*	12	10	10	16	10	8	12	22	14	2	14	8	6	4	10	10	8	8	8	100	20	33	16	
S*→P	18	16	14	16	8	18	18	14	10	10	10	12	6	6	8	6	6	14	10	20	100	16	18	
S→P*	10	8	24	18	12	8	10	20	10	10	25	6	4	8	4	8	6	10	8	33	16	100	25	
S→P	10	10	24	14	12	12	12	29	12	10	24	12	4	6	12	6	10	8	8	16	18	25	100	



APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.20: Structure completeness comparison for the models generated from the 52 original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>s</sup>	S <sup>s</sup> →A	S <sup>s</sup> →B	S <sup>s</sup> →P <sup>s</sup>	S <sup>s</sup> →P	S→P <sup>s</sup>	S→P
A	0	10	0	0	14	14	12	12	8	16	14	8	4	57	43	14	61	37	10	16	18	16	16
A→B	33	0	12	10	20	22	18	25	20	8	29	27	2	63	47	27	69	47	24	24	29	31	27
A→P <sup>s</sup>	33	20	0	8	27	20	20	12	29	16	22	25	8	73	57	29	80	53	22	18	25	20	24
A→P	31	24	10	0	27	24	18	16	27	16	18	27	6	69	57	29	76	49	25	22	24	18	20
B	31	14	12	10	0	18	14	22	25	12	27	20	4	67	53	16	78	43	14	16	18	24	24
B→P <sup>s</sup>	37	25	16	22	29	0	6	22	27	12	27	27	10	67	57	31	76	51	27	31	33	29	31
B→P	33	25	20	24	33	2	0	27	27	16	31	29	8	67	57	31	75	49	27	33	31	31	33
P <sup>s</sup>	33	24	16	16	27	18	16	0	27	10	14	31	8	67	53	33	71	45	27	16	16	12	14
P <sup>s</sup> →A	24	16	4	4	18	16	16	14	0	12	18	12	6	59	47	25	69	45	14	20	20	16	20
P <sup>s</sup> →B	37	25	27	24	31	22	24	35	33	0	37	31	14	67	55	41	73	49	33	31	41	39	43
P	33	24	12	12	31	18	20	8	27	12	0	24	6	67	53	31	73	49	25	14	20	14	22
P→A	22	20	6	4	24	12	16	10	12	12	16	0	6	61	51	25	69	47	20	20	20	16	20
P→B	37	29	25	24	33	29	29	33	27	18	39	35	0	73	63	41	80	55	37	35	41	35	41
S	12	6	2	2	0	2	2	6	6	4	8	2	2	0	37	6	10	35	2	6	8	4	4
S→A	2	8	0	0	6	8	10	8	4	12	6	2	2	24	0	6	33	12	2	8	8	12	14
S→B	29	18	12	12	12	16	10	20	20	12	24	22	4	67	51	0	76	49	14	20	22	24	27
S <sup>s</sup>	12	6	0	0	0	4	4	4	8	4	8	4	2	20	41	6	0	37	2	4	4	2	0
S <sup>s</sup> →A	6	6	0	0	4	10	10	4	10	16	2	2	2	35	25	8	39	0	4	12	12	14	10
S <sup>s</sup> →B	35	16	10	8	20	18	16	20	22	18	25	24	4	65	55	18	75	47	0	16	18	20	25
S <sup>s</sup> →P <sup>s</sup>	33	25	16	18	27	25	24	16	29	16	16	27	8	67	53	27	73	49	24	0	14	8	20
S <sup>s</sup> →P	33	22	14	16	24	20	16	8	25	12	12	29	10	63	47	25	73	47	22	8	0	10	14
S→P <sup>s</sup>	35	24	18	14	29	18	16	6	25	10	12	25	8	61	51	33	71	47	25	12	12	0	14
S→P	31	20	12	14	22	14	14	10	25	10	14	29	6	65	47	31	69	43	24	12	12	10	0



Table B.21: Structure completeness comparison for the models generated from the 52 original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>s</sup>	S <sup>s</sup> →A	S <sup>s</sup> →B	S <sup>s</sup> →P <sup>s</sup>	S <sup>s</sup> →P	S→P <sup>s</sup>	S→P
A	0	6	22	22	14	18	16	24	24	6	27	29	10	20	27	22	14	14	20	25	20	25	25
A→B	33	0	43	45	35	24	22	35	41	16	27	33	25	22	35	22	18	29	35	35	25	29	29
A→P <sup>s</sup>	29	16	0	18	27	25	22	31	33	4	27	41	16	20	25	29	14	14	31	41	29	31	29
A→P	27	4	16	0	24	25	22	24	27	4	25	31	16	20	24	27	14	20	25	35	25	41	35
B	29	18	20	27	0	8	12	18	27	10	18	33	6	22	27	25	12	29	31	33	31	24	24
B→P <sup>s</sup>	16	16	22	22	27	0	18	31	27	20	20	25	12	22	16	25	14	18	25	25	22	27	29
B→P	20	16	25	24	22	24	0	31	29	12	22	27	16	22	16	22	16	18	24	24	27	31	31
P <sup>s</sup>	20	6	14	18	24	20	16	0	22	8	25	14	6	12	10	20	14	14	22	27	29	33	24
P <sup>s</sup> →A	22	8	8	16	18	16	12	18	0	8	24	22	4	22	22	16	16	10	18	25	29	31	24
P <sup>s</sup> →B	35	35	33	39	35	35	37	35	39	0	27	43	24	22	27	24	22	25	35	43	31	37	31
P	16	8	14	20	10	20	12	20	20	6	0	20	4	14	12	16	12	12	20	22	18	24	16
P→A	22	8	16	24	10	20	10	27	29	6	24	0	0	24	14	18	18	12	12	25	22	27	24
P→B	43	27	37	41	41	31	27	37	47	22	39	41	0	24	25	24	18	31	31	43	41	41	41
S	8	8	4	6	6	6	8	4	12	2	4	12	2	0	14	8	27	12	12	2	4	6	14
S→A	6	6	8	10	10	10	6	22	8	0	20	12	4	22	0	12	16	8	6	16	25	22	18
S→B	29	18	18	18	33	14	20	18	29	12	22	27	6	20	22	0	10	20	29	24	25	16	24
S <sup>s</sup>	10	6	0	2	6	2	6	4	4	2	2	8	0	33	8	6	0	6	10	2	10	10	14
S <sup>s</sup> →A	16	8	16	16	18	12	10	22	18	8	16	20	4	16	24	16	14	0	14	22	20	20	29
S <sup>s</sup> →B	22	12	24	25	25	24	22	25	27	4	24	31	18	18	22	29	14	22	0	27	24	24	20
S <sup>s</sup> →P <sup>s</sup>	14	6	16	10	14	10	8	20	12	8	35	20	8	22	14	20	14	10	25	0	22	31	25
S <sup>s</sup> →P	12	8	18	20	20	8	8	33	16	6	31	18	2	20	12	22	8	8	27	37	0	41	31
S→P <sup>s</sup>	14	8	8	10	12	18	12	29	18	4	25	25	12	22	12	20	12	10	24	16	22	0	20
S→P	18	14	12	18	20	14	10	24	20	6	25	16	8	12	10	12	8	10	24	27	25	31	0





APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.22: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>s</sup>	S <sup>s</sup> →A	S <sup>s</sup> →B	S <sup>s</sup> →P <sup>s</sup>	S <sup>s</sup> →P	S→P <sup>s</sup>	S→P
A <sub>R-work</sub>	0	94	27	29	98	47	47	55	22	84	49	22	94	100	51	98	100	41	96	47	51	51	45
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	2	0	0	0	49	6	6	2	2	20	4	0	16	98	16	55	98	12	53	2	4	4	4
A→B <sub>R-free</sub>	-	0	2	0	51	8	6	2	14	25	4	14	16	-	43	53	-	49	51	4	2	2	4
A→P <sup>s</sup> <sub>R-work</sub>	59	100	0	18	100	61	59	71	45	94	69	49	98	100	75	100	100	69	100	73	71	65	71
A→P <sup>s</sup> <sub>R-free</sub>	-	94	0	29	96	51	47	53	84	82	51	82	86	-	90	98	-	90	98	55	57	51	53
A→P <sub>R-work</sub>	57	100	24	0	100	67	61	69	47	94	73	49	98	100	78	100	100	71	100	73	73	73	71
A→P <sub>R-free</sub>	-	92	27	0	98	51	45	51	84	82	59	84	86	-	92	96	-	94	98	57	51	51	57
B <sub>R-work</sub>	0	18	0	0	0	0	0	0	0	20	0	0	6	100	10	41	100	4	33	2	2	2	2
B <sub>R-free</sub>	-	27	2	0	0	2	0	2	10	31	4	10	8	-	45	43	-	47	41	2	4	2	2
B→P <sup>s</sup> <sub>R-work</sub>	37	84	25	20	98	0	16	41	27	86	51	33	84	100	61	96	100	53	96	47	45	47	51
B→P <sup>s</sup> <sub>R-free</sub>	-	88	35	37	96	0	18	31	65	86	45	71	84	-	78	96	-	73	94	43	39	43	39
B→P <sub>R-work</sub>	35	84	22	12	100	18	0	41	29	84	45	33	90	100	59	94	100	53	96	49	39	47	47
B→P <sub>R-free</sub>	-	92	31	37	94	29	0	41	69	82	49	65	90	-	78	96	-	80	96	51	41	47	49
P <sup>s</sup> <sub>R-work</sub>	35	90	12	6	98	27	29	0	24	92	33	24	90	100	53	96	100	47	98	33	35	27	25
P <sup>s</sup> <sub>R-free</sub>	-	96	35	29	94	39	37	0	71	90	39	78	84	-	82	98	-	86	100	39	37	35	35
P <sup>s</sup> →A <sub>R-work</sub>	49	96	35	33	98	55	53	61	0	94	63	25	98	100	55	98	100	47	100	63	59	59	57
P <sup>s</sup> →A <sub>R-free</sub>	-	76	8	10	80	24	18	14	0	67	14	29	63	-	59	84	-	63	76	18	12	16	18
P <sup>s</sup> →B <sub>R-work</sub>	10	41	4	4	57	8	10	6	2	0	6	0	25	100	22	61	100	18	55	6	10	4	6
P <sup>s</sup> →B <sub>R-free</sub>	-	39	10	12	59	10	10	6	20	0	6	22	27	-	49	65	-	47	55	10	8	8	8
P <sub>R-work</sub>	35	96	12	10	100	33	29	29	27	90	0	22	94	100	47	96	100	47	98	29	25	29	27
P <sub>R-free</sub>	-	92	27	31	94	33	31	25	69	88	0	71	86	-	82	94	-	80	92	29	27	29	33
P→A <sub>R-work</sub>	45	96	29	29	100	55	49	59	25	90	51	0	96	100	53	100	100	49	100	57	59	51	51
P→A <sub>R-free</sub>	-	71	12	12	80	18	16	10	31	69	16	0	63	-	61	86	-	65	82	16	16	10	12
P→B <sub>R-work</sub>	4	41	2	2	57	8	6	4	2	31	4	2	0	100	25	65	100	18	57	4	8	6	8
P→B <sub>R-free</sub>	-	47	10	12	65	12	8	2	20	45	6	22	0	-	51	67	-	55	69	6	10	4	10
S <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0
S <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S→A <sub>R-work</sub>	16	82	8	10	88	29	27	33	18	69	27	10	73	100	0	86	100	24	84	37	31	35	33
S→A <sub>R-free</sub>	-	47	2	2	45	12	10	6	18	47	8	18	37	-	0	47	-	31	47	6	4	10	10
S→B <sub>R-work</sub>	2	18	0	0	31	0	0	0	0	16	2	0	8	100	10	0	100	2	25	0	0	0	0
S→B <sub>R-free</sub>	-	27	2	0	31	2	2	0	10	18	4	8	14	-	47	0	-	47	27	0	0	0	0
S <sup>s</sup> <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0
S <sup>s</sup> <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S <sup>s</sup> →A <sub>R-work</sub>	29	84	20	18	92	33	35	37	18	78	35	20	76	100	39	96	100	0	94	35	33	39	35
S <sup>s</sup> →A <sub>R-free</sub>	-	47	2	2	49	12	8	6	22	47	8	18	43	-	45	47	-	0	47	10	6	8	8
S <sup>s</sup> →B <sub>R-work</sub>	2	20	0	0	31	0	0	0	0	22	2	0	6	100	14	41	100	4	0	0	0	2	4
S <sup>s</sup> →B <sub>R-free</sub>	-	29	2	0	41	0	0	0	12	27	2	8	14	-	49	33	-	49	0	2	2	2	4
S <sup>s</sup> →P <sup>s</sup> <sub>R-work</sub>	33	92	12	8	98	33	33	29	24	86	29	22	88	100	49	98	100	43	94	0	24	27	31
S <sup>s</sup> →P <sup>s</sup> <sub>R-free</sub>	-	96	29	25	98	35	31	27	67	86	29	73	82	-	78	98	-	82	96	0	24	27	31
S <sup>s</sup> →P <sub>R-work</sub>	35	90	10	4	96	25	24	29	22	86	31	24	88	100	47	96	100	47	96	33	0	33	31
S <sup>s</sup> →P <sub>R-free</sub>	-	94	24	22	96	37	29	31	71	84	37	76	88	-	82	96	-	86	94	33	0	29	31
S→P <sup>s</sup> <sub>R-work</sub>	37	90	12	10	96	31	33	25	16	88	31	22	92	100	49	98	100	45	94	29	22	0	25
S→P <sup>s</sup> <sub>R-free</sub>	-	94	29	25	94	41	29	29	71	86	35	73	88	-	82	98	-	90	96	35	24	0	31
S→P <sub>R-work</sub>	35	90	10	10	96	24	25	25	22	88	29	20	90	100	47	98	100	43	94	29	20	27	0
S→P <sub>R-free</sub>	-	94	27	29	96	33	27	29	73	88	37	76	88	-	86	98	-	84	96	31	33	33	0



APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.23: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>*</sup>	S <sup>*</sup> →A	S <sup>*</sup> →B	S <sup>*</sup> →P <sup>s</sup>	S <sup>*</sup> →P	S→P <sup>s</sup>	S→P
A <sub>R-work</sub>	100	4	14	14	2	16	18	10	29	6	16	33	2	0	33	0	0	29	2	20	14	12	20
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	4	100	0	0	33	10	10	8	2	39	0	4	43	2	2	27	2	4	27	6	6	6	6
A→B <sub>R-free</sub>	-	100	4	8	22	4	2	2	10	35	4	16	37	-	10	20	-	4	20	0	4	4	2
A→P <sup>s</sup> <sub>R-work</sub>	14	0	100	59	0	14	20	18	20	2	20	22	0	0	18	0	0	12	0	16	20	24	20
A→P <sup>s</sup> <sub>R-free</sub>	-	4	100	43	2	14	22	12	8	8	22	6	4	-	8	0	-	8	0	16	20	20	20
A→P <sub>R-work</sub>	14	0	59	100	0	14	27	25	20	2	18	22	0	0	12	0	0	12	0	20	24	18	20
A→P <sub>R-free</sub>	-	8	43	100	2	12	18	20	6	6	10	4	2	-	6	4	-	4	2	18	27	24	14
B <sub>R-work</sub>	2	33	0	0	100	2	0	2	2	24	0	0	37	0	2	27	0	4	35	0	2	2	2
B <sub>R-free</sub>	-	22	2	2	100	2	6	4	10	10	2	10	27	-	10	25	-	4	18	0	0	4	2
B→P <sup>s</sup> <sub>R-work</sub>	16	10	14	14	2	100	67	31	18	6	16	12	8	0	10	4	0	14	4	20	29	22	25
B→P <sup>s</sup> <sub>R-free</sub>	-	4	14	12	2	100	53	29	12	4	22	12	4	-	10	2	-	16	6	22	24	16	27
B→P <sub>R-work</sub>	18	10	20	27	0	67	100	29	18	6	25	18	4	0	14	6	0	12	4	18	37	20	27
B→P <sub>R-free</sub>	-	4	22	18	6	53	100	22	14	8	20	20	2	-	12	2	-	12	4	18	29	24	24
P <sup>s</sup> <sub>R-work</sub>	10	8	18	25	2	31	29	100	16	2	37	18	6	0	14	4	0	16	2	37	35	47	49
P <sup>s</sup> <sub>R-free</sub>	-	2	12	20	4	29	22	100	16	4	35	12	14	-	12	2	-	8	0	33	31	35	35
P <sup>s</sup> →A <sub>R-work</sub>	29	2	20	20	2	18	18	16	100	4	10	49	0	0	27	2	0	35	0	14	20	25	22
P <sup>s</sup> →A <sub>R-free</sub>	-	10	8	6	10	12	14	16	100	14	18	39	18	-	24	6	-	16	12	16	18	14	10
P <sup>s</sup> →B <sub>R-work</sub>	6	39	2	2	24	6	6	2	4	100	4	10	43	0	10	24	0	4	24	8	4	8	6
P <sup>s</sup> →B <sub>R-free</sub>	-	35	8	6	10	4	8	4	14	100	6	10	27	-	4	18	-	6	18	4	8	6	4
P <sub>R-work</sub>	16	0	20	18	0	16	25	37	10	4	100	27	2	0	25	2	0	18	0	41	43	39	43
P <sub>R-free</sub>	-	4	22	10	2	22	20	35	18	6	100	14	8	-	10	2	-	12	6	41	35	35	29
P→A <sub>R-work</sub>	33	4	22	22	0	12	18	18	49	10	27	100	2	0	37	0	0	31	0	22	18	27	29
P→A <sub>R-free</sub>	-	16	6	4	10	12	20	12	39	10	14	100	16	-	22	6	-	18	10	12	8	18	12
P→B <sub>R-work</sub>	2	43	0	0	37	8	4	6	0	43	2	2	100	0	2	27	0	6	37	8	4	2	2
P→B <sub>R-free</sub>	-	37	4	2	27	4	2	14	18	27	8	16	100	-	12	20	-	2	18	12	2	8	2
S <sub>R-work</sub>	0	2	0	0	0	0	0	0	0	0	0	0	0	100	0	0	47	0	0	0	0	0	0
S <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S→A <sub>R-work</sub>	33	2	18	12	2	10	14	14	27	10	25	37	2	0	100	4	0	37	2	14	22	16	20
S→A <sub>R-free</sub>	-	10	8	6	10	10	12	12	24	4	10	22	12	-	100	6	-	24	4	16	14	8	4
S→B <sub>R-work</sub>	0	27	0	0	27	4	6	4	2	24	2	0	27	0	4	100	0	2	33	2	4	2	2
S→B <sub>R-free</sub>	-	20	0	4	25	2	2	2	6	18	2	6	20	-	6	100	-	6	39	2	4	2	2
S <sup>*</sup> <sub>R-work</sub>	0	2	0	0	0	0	0	0	0	0	0	0	0	47	0	0	100	0	0	0	0	0	0
S <sup>*</sup> <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S <sup>*</sup> →A <sub>R-work</sub>	29	4	12	12	4	14	12	16	35	4	18	31	6	0	37	2	0	100	2	22	20	16	22
S <sup>*</sup> →A <sub>R-free</sub>	-	4	8	4	4	16	12	8	16	6	12	18	2	-	24	6	-	100	4	8	8	2	8
S <sup>*</sup> →B <sub>R-work</sub>	2	27	0	0	35	4	4	2	0	24	0	0	37	0	2	33	0	2	100	6	4	4	2
S <sup>*</sup> →B <sub>R-free</sub>	-	20	0	2	18	6	4	0	12	18	6	10	18	-	4	39	-	4	100	2	4	2	0
S <sup>*</sup> →P <sup>s</sup> <sub>R-work</sub>	20	6	16	20	0	20	18	37	14	8	41	22	8	0	14	2	0	22	6	100	43	43	39
S <sup>*</sup> →P <sup>s</sup> <sub>R-free</sub>	-	0	16	18	0	22	18	33	16	4	41	12	12	-	16	2	-	8	2	100	43	37	37
S <sup>*</sup> →P <sub>R-work</sub>	14	6	20	24	2	29	37	35	20	4	43	18	4	0	22	4	0	20	4	43	100	45	49
S <sup>*</sup> →P <sub>R-free</sub>	-	4	20	27	0	24	29	31	18	8	35	8	2	-	14	4	-	8	4	43	100	47	35
S→P <sup>s</sup> <sub>R-work</sub>	12	6	24	18	2	22	20	47	25	8	39	27	2	0	16	2	0	16	4	43	45	100	47
S→P <sup>s</sup> <sub>R-free</sub>	-	4	20	24	4	16	24	35	14	6	35	18	8	-	8	2	-	2	2	37	47	100	35
S→P <sub>R-work</sub>	20	6	20	20	2	25	27	49	22	6	43	29	2	0	20	2	0	22	2	39	49	47	100
S→P <sub>R-free</sub>	-	2	20	14	2	27	24	35	10	4	29	12	2	-	4	2	-	8	0	37	35	35	100



APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.24: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P <sup>a</sup>	A→P	B	B→P <sup>a</sup>	B→P	P <sup>a</sup>	P <sup>a</sup> →A	P <sup>a</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>a</sup>	S <sup>a</sup> →A	S <sup>a</sup> →B	S <sup>a</sup> →P <sup>a</sup>	S <sup>a</sup> →P	S→P <sup>a</sup>	S→P
A <sub>R-work</sub>	0	51	10	8	61	10	12	10	2	49	12	0	39	100	24	69	100	24	61	12	14	16	16
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	0	0	0	0	8	0	0	0	0	6	0	0	2	94	6	18	94	2	10	0	0	0	0
A→B <sub>R-free</sub>	-	0	0	0	4	0	0	0	4	8	0	0	2	-	37	16	-	33	10	0	0	0	0
A→P <sup>a</sup> <sub>R-work</sub>	4	61	0	2	73	8	8	2	2	53	2	0	49	100	22	78	100	20	78	2	2	2	4
A→P <sup>a</sup> <sub>R-free</sub>	-	51	0	2	63	4	6	2	18	47	2	14	35	-	57	71	-	51	63	0	2	0	4
A→P <sub>R-work</sub>	4	57	2	0	71	8	8	2	2	51	4	0	43	100	20	76	100	22	75	2	2	4	6
A→P <sub>R-free</sub>	-	49	2	0	65	6	4	0	18	45	2	14	33	-	57	67	-	49	61	0	0	0	2
B <sub>R-work</sub>	0	10	0	0	0	0	0	0	0	6	0	0	2	96	2	8	96	2	2	0	0	0	0
B <sub>R-free</sub>	-	8	0	0	0	0	0	0	2	6	0	2	2	-	39	10	-	35	6	0	0	0	0
B→P <sup>a</sup> <sub>R-work</sub>	2	53	4	2	61	0	2	0	2	49	0	0	43	100	18	71	100	20	59	0	2	0	2
B→P <sup>a</sup> <sub>R-free</sub>	-	53	10	10	61	0	2	4	18	45	2	14	41	-	53	67	-	45	59	0	2	0	4
B→P <sub>R-work</sub>	2	57	2	0	63	2	0	0	2	47	0	0	35	100	16	71	100	18	61	0	0	0	0
B→P <sub>R-free</sub>	-	55	4	8	63	2	0	0	18	49	0	14	35	-	53	73	-	47	59	0	0	0	0
P <sup>a</sup> <sub>R-work</sub>	6	45	4	2	57	4	4	0	2	29	0	0	29	100	16	67	100	16	59	0	2	0	4
P <sup>a</sup> <sub>R-free</sub>	-	43	4	4	63	4	6	0	18	35	2	14	27	-	47	57	-	47	61	0	2	0	4
P <sup>a</sup> →A <sub>R-work</sub>	4	49	16	14	61	18	20	16	0	49	16	0	47	100	25	69	100	25	61	16	16	16	16
P <sup>a</sup> →A <sub>R-free</sub>	-	25	4	4	31	0	4	0	0	25	0	0	18	-	41	35	-	39	35	0	2	0	2
P <sup>a</sup> →B <sub>R-work</sub>	2	18	2	2	20	0	2	0	0	0	0	0	6	100	10	24	100	10	25	0	2	0	2
P <sup>a</sup> →B <sub>R-free</sub>	-	16	4	6	24	2	4	0	10	0	0	6	6	-	43	29	-	35	25	0	2	0	2
P <sub>R-work</sub>	6	45	4	2	57	2	4	0	2	35	0	0	31	100	18	63	100	18	59	0	2	2	4
P <sub>R-free</sub>	-	41	6	6	57	4	6	2	12	39	0	14	27	-	45	59	-	45	59	0	4	0	4
P→A <sub>R-work</sub>	6	49	16	14	61	16	18	14	0	47	14	0	41	100	25	69	100	22	63	16	16	18	18
P→A <sub>R-free</sub>	-	25	4	4	29	0	4	0	0	20	2	0	12	-	41	33	-	39	31	0	2	0	4
P→B <sub>R-work</sub>	0	16	0	0	22	0	2	0	2	8	0	0	0	100	8	29	100	8	24	0	2	0	2
P→B <sub>R-free</sub>	-	20	2	2	20	0	2	0	10	10	2	10	0	-	45	29	-	41	25	0	2	0	2
S <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S→A <sub>R-work</sub>	2	39	2	2	47	6	8	4	0	33	4	2	24	100	0	53	100	4	49	4	8	6	10
S→A <sub>R-free</sub>	-	12	0	0	16	2	2	0	0	14	0	2	8	-	0	14	-	2	20	0	2	0	2
S→B <sub>R-work</sub>	0	10	0	0	4	0	0	0	0	4	0	0	2	96	0	0	96	2	4	0	0	0	0
S→B <sub>R-free</sub>	-	10	0	0	4	0	0	0	2	4	0	4	2	-	41	0	-	31	6	0	0	0	0
S <sup>a</sup> <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sup>a</sup> <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S <sup>a</sup> →A <sub>R-work</sub>	0	39	2	0	57	4	6	6	0	33	4	2	27	100	8	63	100	0	51	4	8	8	10
S <sup>a</sup> →A <sub>R-free</sub>	-	12	0	0	16	0	2	0	0	14	2	0	10	-	10	14	-	0	20	0	0	0	2
S <sup>a</sup> →B <sub>R-work</sub>	0	6	0	0	2	0	0	0	0	6	0	0	4	96	2	10	96	2	0	0	0	0	0
S <sup>a</sup> →B <sub>R-free</sub>	-	6	0	0	2	0	0	0	2	6	0	2	2	-	43	8	-	33	0	0	0	0	0
S <sup>a</sup> →P <sup>a</sup> <sub>R-work</sub>	6	45	4	2	59	4	6	2	2	35	0	0	29	100	20	69	100	18	55	0	2	0	4
S <sup>a</sup> →P <sup>a</sup> <sub>R-free</sub>	-	41	8	4	57	6	10	4	12	37	2	14	24	-	45	57	-	45	55	0	4	2	4
S <sup>a</sup> →P <sub>R-work</sub>	4	39	2	0	57	4	2	2	2	35	0	0	27	100	20	67	100	16	53	0	0	0	4
S <sup>a</sup> →P <sub>R-free</sub>	-	37	6	8	55	6	2	2	12	37	0	12	31	-	41	59	-	39	57	0	0	0	4
S→P <sup>a</sup> <sub>R-work</sub>	6	47	4	2	57	2	4	2	2	29	0	0	29	100	20	67	100	18	59	0	2	0	2
S→P <sup>a</sup> <sub>R-free</sub>	-	41	6	10	61	2	4	2	14	43	2	12	29	-	43	59	-	39	57	0	2	0	4
S→P <sub>R-work</sub>	4	39	2	0	55	4	2	0	2	27	0	0	24	100	16	61	100	16	51	0	0	0	0
S→P <sub>R-free</sub>	-	41	4	6	55	4	6	2	12	39	0	12	25	-	43	57	-	41	55	0	2	2	0



APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.25: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>*</sup>	S <sup>*</sup> →A	S <sup>*</sup> →B	S <sup>*</sup> →P <sup>s</sup>	S <sup>*</sup> →P	S→P <sup>s</sup>	S→P
A <sub>R-work</sub>	0	43	18	22	37	37	35	45	20	35	37	22	55	0	27	29	0	18	35	35	37	35	29
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	2	0	0	0	41	6	6	2	2	14	4	0	14	4	10	37	4	10	43	2	4	4	4
A→B <sub>R-free</sub>	-	0	2	0	47	8	6	2	10	18	4	14	14	-	6	37	-	16	41	4	2	2	4
A→P <sup>s</sup> <sub>R-work</sub>	55	39	0	16	27	53	51	69	43	41	67	49	49	0	53	22	0	49	22	71	69	63	67
A→P <sup>s</sup> <sub>R-free</sub>	-	43	0	27	33	47	41	51	67	35	49	69	51	-	33	27	-	39	35	55	55	51	49
A→P <sub>R-work</sub>	53	43	22	0	29	59	53	67	45	43	69	49	55	0	59	24	0	49	25	71	71	69	65
A→P <sub>R-free</sub>	-	43	25	0	33	45	41	51	67	37	57	71	53	-	35	29	-	45	37	57	51	51	55
B <sub>R-work</sub>	0	8	0	0	0	0	0	0	0	17	0	0	4	4	8	33	4	2	31	2	2	2	2
B <sub>R-free</sub>	-	20	2	0	0	2	0	2	8	25	4	8	6	-	6	33	-	12	35	2	4	2	2
B→P <sup>s</sup> <sub>R-work</sub>	35	31	22	18	37	0	14	41	25	37	51	33	41	0	43	25	0	33	37	47	43	47	49
B→P <sup>s</sup> <sub>R-free</sub>	-	35	25	27	35	0	16	27	47	41	43	57	43	-	25	29	-	27	35	43	37	43	35
B→P <sub>R-work</sub>	33	27	20	12	37	16	0	41	27	37	45	33	55	0	43	24	0	35	35	49	39	47	47
B→P <sub>R-free</sub>	-	37	27	29	31	27	0	41	51	33	49	51	55	-	25	24	-	33	37	51	41	47	49
P <sup>s</sup> <sub>R-work</sub>	29	45	8	4	41	24	25	0	22	63	33	24	61	0	37	29	0	31	39	33	33	27	22
P <sup>s</sup> <sub>R-free</sub>	-	53	31	25	31	35	31	0	53	55	37	65	57	-	35	41	-	39	39	39	35	35	31
P <sup>s</sup> →A <sub>R-work</sub>	45	47	20	20	37	37	33	45	0	45	47	25	51	0	29	29	0	22	39	47	43	43	41
P <sup>s</sup> →A <sub>R-free</sub>	-	51	4	6	49	24	14	14	0	41	14	29	45	-	18	49	-	24	41	18	10	16	16
P <sup>s</sup> →B <sub>R-work</sub>	8	24	2	2	37	8	8	6	2	0	6	0	20	0	12	37	0	8	29	6	8	4	4
P <sup>s</sup> →B <sub>R-free</sub>	-	24	6	6	35	8	6	10	0	6	16	22	-	-	6	35	-	12	29	10	6	8	6
P <sub>R-work</sub>	29	51	8	8	43	31	25	29	25	55	0	22	63	0	29	33	0	29	39	29	24	27	24
P <sub>R-free</sub>	-	51	22	25	37	29	25	24	57	49	0	57	59	-	37	35	-	35	33	29	24	29	29
P→A <sub>R-work</sub>	39	47	14	16	39	39	31	45	25	43	37	0	55	0	27	31	0	27	37	41	43	33	33
P→A <sub>R-free</sub>	-	45	8	8	51	18	12	10	31	49	14	0	51	-	20	53	-	25	51	16	14	10	8
P→B <sub>R-work</sub>	4	25	2	2	35	8	4	4	0	24	4	2	0	0	18	35	0	10	33	4	6	6	6
P→B <sub>R-free</sub>	-	27	8	10	45	12	6	2	10	35	4	12	0	-	6	37	-	14	43	6	8	4	8
S <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0
S <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S→A <sub>R-work</sub>	14	43	6	8	41	24	20	29	18	35	24	8	49	0	0	33	0	20	35	33	24	29	24
S→A <sub>R-free</sub>	-	35	2	2	29	10	8	6	18	33	8	16	29	-	0	33	-	29	27	6	2	10	8
S→B <sub>R-work</sub>	2	8	0	0	27	0	0	0	0	12	2	0	6	4	10	0	4	0	22	0	0	0	0
S→B <sub>R-free</sub>	-	18	2	0	27	2	2	0	8	14	4	4	12	-	6	0	-	16	22	0	0	0	0
S <sup>*</sup> <sub>R-work</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0
S <sup>*</sup> <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S <sup>*</sup> →A <sub>R-work</sub>	29	45	18	18	35	29	29	31	18	45	31	18	49	0	31	33	0	0	43	31	25	31	25
S <sup>*</sup> →A <sub>R-free</sub>	-	35	2	2	33	12	6	6	22	33	6	18	33	-	35	33	-	0	27	10	6	8	6
S <sup>*</sup> →B <sub>R-work</sub>	2	14	0	0	29	0	0	0	16	2	0	2	4	4	12	31	4	2	0	0	0	2	4
S <sup>*</sup> →B <sub>R-free</sub>	-	24	2	0	39	0	0	0	10	22	2	6	12	-	6	25	-	16	0	2	2	2	4
S <sup>*</sup> →P <sup>s</sup> <sub>R-work</sub>	27	47	8	6	39	29	27	27	22	51	29	22	59	0	29	29	0	25	39	0	22	27	27
S <sup>*</sup> →P <sup>s</sup> <sub>R-free</sub>	-	55	22	22	41	29	22	24	55	49	27	59	59	-	33	41	-	37	41	0	20	25	27
S <sup>*</sup> →P <sub>R-work</sub>	31	51	8	4	39	22	22	27	20	51	31	24	61	0	27	29	0	31	43	33	0	33	27
S <sup>*</sup> →P <sub>R-free</sub>	-	57	18	14	41	31	27	29	59	47	37	65	57	-	41	37	-	47	37	33	0	29	27
S→P <sup>s</sup> <sub>R-work</sub>	31	43	8	8	39	29	29	24	14	59	31	22	63	0	29	31	0	27	35	29	20	0	24
S→P <sup>s</sup> <sub>R-free</sub>	-	53	24	16	33	39	25	27	57	43	33	61	59	-	39	39	-	51	39	35	22	0	27
S→P <sub>R-work</sub>	31	51	8	10	41	20	24	25	20	61	29	20	67	0	31	37	0	27	43	29	20	27	0
S→P <sub>R-free</sub>	-	53	24	24	41	29	22	27	61	49	37	65	63	-	43	41	-	43	41	31	31	31	0

0  71

## B.2 Experimental results for synthetic data sets for the original data sets used in Buccaneer development

Table B.26: Complete and intermediate models produced by the 23 pipeline variants for the 52 synthetic data sets, where (T) and (C) denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	HA-NCS			MR-NCS			NO-NCS		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
A	258	1(T) 0(C)	0	258	1(T) 0(C)	0	258	1(T) 0(C)	0
A→P*	259	0(T) 0(C)	0	258	0(T) 0(C)	1	259	0(T) 0(C)	0
A→B	259	0(T) 0(C)	0	259	0(T) 0(C)	0	259	0(T) 0(C)	0
B	259	0(T) 0(C)	0	259	0(T) 0(C)	0	259	0(T) 0(C)	0
B→P*	259	0(T) 0(C)	0	259	0(T) 0(C)	0	259	0(T) 0(C)	0
P*	259	0(T) 0(C)	0	259	0(T) 0(C)	0	257	2(T) 0(C)	0
P*→A	259	0(T) 0(C)	0	259	0(T) 0(C)	0	259	0(T) 0(C)	0
P*→B	259	0(T) 0(C)	0	259	0(T) 0(C)	0	259	0(T) 0(C)	0
A→P	-	-	-	-	-	-	259	0(T) 0(C)	0
B→P	-	-	-	-	-	-	258	1(T) 0(C)	0
P	-	-	-	-	-	-	256	2(T) 0(C)	1
P→A	-	-	-	-	-	-	256	2(T) 0(C)	1
P→B	-	-	-	-	-	-	258	0(T) 0(C)	1

Models used in the comparison: 259 HA-NCS, 258 MR-NCS and 258 NO-NCS.

Table B.27: Structure completeness comparison for the models generated from the 52 synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	1	2	1	0	4	35	2
A→B	95	0	90	42	32	86	95	27
A→P*	96	8	0	7	1	26	98	4
B	97	53	92	0	32	88	96	34
B→P*	99	63	97	63	0	94	100	44
P*	95	13	70	10	3	0	98	7
P*→A	15	0	2	0	0	0	0	1
P*→B	97	66	95	62	53	92	98	0



Table B.28: Structure completeness comparison for the models generated from the 52 synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	100	4	2	3	1	1	50	1
A→B	4	100	2	4	5	2	4	7
A→P*	2	2	100	2	2	4	1	1
B	3	4	2	100	6	1	3	3
B→P*	1	5	2	6	100	3	0	3
P*	1	2	4	1	3	100	2	1
P*→A	50	4	1	3	0	2	100	1
P*→B	1	7	1	3	3	1	1	100



Table B.29: Structure completeness comparison for the models generated from the 52 synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	0	1	0	0	3	10	1
A→B	90	0	85	30	22	82	90	20
A→P*	79	2	0	1	1	14	82	1
B	93	37	89	0	19	86	93	22
B→P*	97	47	94	38	0	90	98	32
P*	93	8	41	5	1	0	94	5
P*→A	3	0	0	0	0	0	0	0
P*→B	93	54	90	47	42	90	94	0



Table B.30: Structure completeness comparison for the models generated from the 52 synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	$A \rightarrow B$	$A \rightarrow P^*$	B	$B \rightarrow P^*$	$P^*$	$P^* \rightarrow A$	$P^* \rightarrow B$
A	0	1	1	1	0	1	25	1
$A \rightarrow B$	5	0	5	13	10	3	5	7
$A \rightarrow P^*$	17	6	0	6	0	12	15	3
B	3	16	3	0	12	3	3	12
$B \rightarrow P^*$	2	16	3	24	0	4	2	11
$P^*$	2	4	29	5	3	0	3	2
$P^* \rightarrow A$	11	0	2	0	0	0	0	0
$P^* \rightarrow B$	4	12	5	15	11	2	4	0



Table B.31: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	96	86	95	92	97	29	94
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	3	0	7	41	8	41	1	30
A→B <sub>R-free</sub>	-	0	42	44	11	44	86	33
A→P* <sub>R-work</sub>	9	92	0	91	67	100	2	85
A→P* <sub>R-free</sub>	-	49	0	49	20	53	94	44
B <sub>R-work</sub>	4	48	6	0	3	45	1	34
B <sub>R-free</sub>	-	45	44	0	6	48	87	33
B→P* <sub>R-work</sub>	7	87	23	93	0	92	2	81
B→P* <sub>R-free</sub>	-	85	76	90	0	85	96	80
P* <sub>R-work</sub>	2	51	0	49	3	0	0	41
P* <sub>R-free</sub>	-	49	45	48	10	0	93	36
P*→A <sub>R-work</sub>	53	99	97	98	97	100	0	98
P*→A <sub>R-free</sub>	-	11	4	12	4	6	0	10
P*→B <sub>R-work</sub>	6	60	11	58	13	53	2	0
P*→B <sub>R-free</sub>	-	59	50	57	16	54	89	0





Table B.32: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <i>R-work</i>	100	1	4	1	1	0	18	0
A <i>R-free</i>	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	1	100	2	11	4	7	0	10
A→B <i>R-free</i>	-	100	8	11	4	7	3	8
A→P* <i>R-work</i>	4	2	100	3	10	0	2	4
A→P* <i>R-free</i>	-	8	100	7	4	2	2	6
B <i>R-work</i>	1	11	3	100	4	7	0	8
B <i>R-free</i>	-	11	7	100	4	4	1	10
B→P* <i>R-work</i>	1	4	10	4	100	5	1	7
B→P* <i>R-free</i>	-	4	4	4	100	4	0	4
P* <i>R-work</i>	0	7	0	7	5	100	0	6
P* <i>R-free</i>	-	7	2	4	4	100	1	10
P*→A <i>R-work</i>	18	0	2	0	1	0	100	0
P*→A <i>R-free</i>	-	3	2	1	0	1	100	2
P*→B <i>R-work</i>	0	10	4	8	7	6	0	100
P*→B <i>R-free</i>	-	8	6	10	4	10	2	100



Table B.33: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	89	58	89	73	96	2	85
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	2	0	2	10	0	20	0	8
A→B <sub>R-free</sub>	-	0	19	16	3	24	75	12
A→P* <sub>R-work</sub>	4	73	0	73	34	91	0	63
A→P* <sub>R-free</sub>	-	28	0	31	10	33	78	22
B <sub>R-work</sub>	2	14	0	0	0	22	0	7
B <sub>R-free</sub>	-	16	21	0	2	20	73	9
B→P* <sub>R-work</sub>	3	63	2	60	0	54	0	49
B→P* <sub>R-free</sub>	-	56	49	58	0	55	93	47
P* <sub>R-work</sub>	1	29	0	28	0	0	0	20
P* <sub>R-free</sub>	-	25	21	23	2	0	86	16
P*→A <sub>R-work</sub>	15	93	67	95	82	100	0	92
P*→A <sub>R-free</sub>	-	5	2	6	2	3	0	4
P*→B <sub>R-work</sub>	3	25	1	15	3	29	0	0
P*→B <sub>R-free</sub>	-	26	28	22	5	27	78	0



Table B.34: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	7	29	6	18	2	27	9
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	1	0	5	31	8	21	1	22
A→B <sub>R-free</sub>	-	0	23	28	7	21	12	21
A→P* <sub>R-work</sub>	5	19	0	18	33	9	2	22
A→P* <sub>R-free</sub>	-	22	0	18	10	20	17	21
B <sub>R-work</sub>	2	35	6	0	2	22	1	28
B <sub>R-free</sub>	-	29	23	0	5	28	14	24
B→P* <sub>R-work</sub>	4	25	20	33	0	37	2	31
B→P* <sub>R-free</sub>	-	29	27	32	0	30	3	33
P* <sub>R-work</sub>	1	22	0	20	3	0	0	20
P* <sub>R-free</sub>	-	23	24	25	8	0	7	20
P*→A <sub>R-work</sub>	38	6	30	3	15	0	0	7
P*→A <sub>R-free</sub>	-	6	2	6	2	3	0	5
P*→B <sub>R-work</sub>	3	35	10	42	10	24	2	0
P*→B <sub>R-free</sub>	-	33	22	34	11	27	10	0



Table B.35: Structure completeness comparison for the models generated from the 52 synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	0	2	1	0	4	36	2
A→B	95	0	91	44	33	86	95	30
A→P*	96	8	0	7	1	23	98	5
B	97	52	91	0	33	88	96	35
B→P*	99	64	97	63	0	96	100	46
P*	95	13	72	10	2	0	98	7
P*→A	13	0	2	0	0	0	0	1
P*→B	97	64	95	61	50	93	97	0



Table B.36: Structure completeness comparison for the models generated from the 52 synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	100	5	2	3	1	1	51	2
A→B	5	100	2	4	4	1	4	6
A→P*	2	2	100	2	2	5	1	1
B	3	4	2	100	5	1	3	3
B→P*	1	4	2	5	100	2	0	5
P*	1	1	5	1	2	100	2	1
P*→A	51	4	1	3	0	2	100	2
P*→B	2	6	1	3	5	1	2	100



Table B.37: Structure completeness comparison for the models generated from the 52 synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	0	1	0	0	3	10	1
A→B	90	0	85	32	20	84	91	22
A→P*	78	1	0	1	1	13	81	1
B	93	38	88	0	17	84	93	24
B→P*	97	47	95	46	0	91	98	37
P*	92	7	43	4	1	0	93	3
P*→A	3	0	0	0	0	0	0	0
P*→B	93	54	91	47	41	91	93	0



Table B.38: Structure completeness comparison for the models generated from the 52 synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	0	1	1	0	1	26	1
A→B	5	0	5	12	12	2	5	8
A→P*	18	7	0	6	0	10	16	3
B	4	14	3	0	16	4	3	11
B→P*	2	16	2	17	0	5	2	9
P*	3	6	29	6	1	0	5	3
P*→A	10	0	2	0	0	0	0	1
P*→B	4	10	4	15	8	2	5	0



Table B.39: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <i>R-work</i>	0	96	87	95	90	98	28	94
A <i>R-free</i>	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	3	0	8	43	9	38	1	30
A→B <i>R-free</i>	-	0	42	42	10	41	86	34
A→P* <i>R-work</i>	9	91	0	91	70	100	1	87
A→P* <i>R-free</i>	-	50	0	51	22	53	93	43
B <i>R-work</i>	4	44	7	0	3	46	1	34
B <i>R-free</i>	-	47	45	0	5	48	86	38
B→P* <i>R-work</i>	9	86	21	95	0	92	2	81
B→P* <i>R-free</i>	-	85	75	91	0	86	97	82
P* <i>R-work</i>	2	55	0	50	3	0	0	41
P* <i>R-free</i>	-	51	44	45	10	0	93	36
P*→A <i>R-work</i>	57	99	97	99	97	100	0	98
P*→A <i>R-free</i>	-	11	6	11	3	5	0	11
P*→B <i>R-work</i>	6	59	9	55	14	51	2	0
P*→B <i>R-free</i>	-	58	51	54	13	50	88	0



Table B.40: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <i>R-work</i>	100	1	5	1	1	0	16	0
A <i>R-free</i>	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	1	100	1	13	6	7	0	11
A→B <i>R-free</i>	-	100	7	10	5	8	3	8
A→P* <i>R-work</i>	5	1	100	2	9	0	2	4
A→P* <i>R-free</i>	-	7	100	4	3	3	1	6
B <i>R-work</i>	1	13	2	100	2	4	0	11
B <i>R-free</i>	-	10	4	100	3	7	3	8
B→P* <i>R-work</i>	1	6	9	2	100	4	1	5
B→P* <i>R-free</i>	-	5	3	3	100	5	0	5
P* <i>R-work</i>	0	7	0	4	4	100	0	9
P* <i>R-free</i>	-	8	3	7	5	100	1	14
P*→A <i>R-work</i>	16	0	2	0	1	0	100	1
P*→A <i>R-free</i>	-	3	1	3	0	1	100	1
P*→B <i>R-work</i>	0	11	4	11	5	9	1	100
P*→B <i>R-free</i>	-	8	6	8	5	14	1	100



Table B.41: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	88	58	91	72	96	3	85
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	2	0	1	13	0	19	0	10
A→B <sub>R-free</sub>	-	0	17	14	2	20	75	10
A→P* <sub>R-work</sub>	4	75	0	74	32	90	0	64
A→P* <sub>R-free</sub>	-	29	0	31	8	33	77	23
B <sub>R-work</sub>	2	16	0	0	0	18	0	7
B <sub>R-free</sub>	-	18	22	0	1	18	74	8
B→P* <sub>R-work</sub>	3	64	4	62	0	55	0	51
B→P* <sub>R-free</sub>	-	60	50	58	0	58	93	51
P* <sub>R-work</sub>	1	26	0	27	0	0	0	20
P* <sub>R-free</sub>	-	24	22	23	2	0	86	14
P*→A <sub>R-work</sub>	12	93	69	95	80	100	0	90
P*→A <sub>R-free</sub>	-	5	3	5	2	3	0	4
P*→B <sub>R-work</sub>	3	25	2	16	2	28	0	0
P*→B <sub>R-free</sub>	-	26	27	23	5	24	80	0





Table B.42: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	7	29	4	17	2	25	9
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	1	0	7	30	8	19	1	21
A→B <sub>R-free</sub>	-	0	25	28	8	21	11	25
A→P* <sub>R-work</sub>	5	16	0	17	38	9	1	23
A→P* <sub>R-free</sub>	-	22	0	20	14	21	16	20
B <sub>R-work</sub>	2	28	6	0	3	28	1	26
B <sub>R-free</sub>	-	29	23	0	4	30	12	30
B→P* <sub>R-work</sub>	6	22	17	33	0	37	2	30
B→P* <sub>R-free</sub>	-	25	25	33	0	28	4	31
P* <sub>R-work</sub>	1	29	0	23	3	0	0	21
P* <sub>R-free</sub>	-	27	21	22	8	0	7	22
P*→A <sub>R-work</sub>	44	6	28	3	17	0	0	7
P*→A <sub>R-free</sub>	-	6	3	6	2	2	0	7
P*→B <sub>R-work</sub>	3	34	8	39	12	23	2	0
P*→B <sub>R-free</sub>	-	32	24	31	9	26	8	0



Table B.43: Structure completeness comparison for the models generated from the 52 synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A	0	0	2	3	1	0	0	4	33	2	3	32	0
A→B	94	0	85	84	41	28	25	77	95	22	77	95	24
A→P*	97	13	0	40	11	2	2	24	98	4	21	98	3
A→P	95	12	53	0	9	3	0	24	96	5	22	96	4
B	95	53	88	88	0	26	22	81	96	31	82	96	27
B→P*	99	68	97	97	69	0	35	93	100	44	93	100	41
B→P	99	71	98	98	72	51	0	94	100	48	95	100	42
P*	96	22	72	70	17	5	3	0	98	10	45	97	8
P*→A	13	0	2	2	0	0	0	0	0	1	0	18	0
P*→B	97	72	94	93	65	52	48	89	98	0	88	98	44
P	97	21	72	74	16	5	3	42	99	10	0	98	11
P→A	14	1	1	2	1	0	0	2	19	0	2	0	0
P→B	98	73	95	96	69	57	54	90	99	52	88	99	0



Table B.44: Structure completeness comparison for the models generated from the 52 synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A	100	5	2	2	3	1	0	0	54	1	0	54	2
A→B	5	100	2	3	6	4	4	1	5	5	2	4	2
A→P*	2	2	100	7	1	2	0	4	0	2	8	1	2
A→P	2	3	7	100	3	0	2	6	2	2	4	2	0
B	3	6	1	3	100	5	6	2	4	4	2	3	3
B→P*	1	4	2	0	5	100	14	3	0	3	3	0	2
B→P	0	4	0	2	6	14	100	3	0	3	2	0	4
P*	0	1	4	6	2	3	3	100	2	2	13	1	2
P*→A	54	5	0	2	4	0	0	2	100	1	1	63	1
P*→B	1	5	2	2	4	3	3	2	1	100	2	1	5
P	0	2	8	4	2	3	2	13	1	2	100	0	1
P→A	54	4	1	2	3	0	0	1	63	1	0	100	1
P→B	2	2	2	0	3	2	4	2	1	5	1	1	100



Table B.45: Structure completeness comparison for the models generated from the 52 synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A	0	0	1	2	0	0	0	3	9	1	3	8	0
A→B	86	0	78	76	28	18	16	72	86	16	73	85	14
A→P*	74	4	0	17	2	1	0	14	77	1	12	78	1
A→P	86	7	24	0	3	1	0	10	86	2	11	86	2
B	90	33	83	81	0	14	9	77	90	20	76	90	15
B→P*	97	53	95	91	49	0	10	85	98	34	86	97	28
B→P	98	55	94	93	50	15	0	87	99	36	87	99	28
P*	94	14	52	41	9	1	0	0	94	5	12	95	5
P*→A	4	0	0	2	0	0	0	0	0	0	0	2	0
P*→B	93	57	88	88	50	40	37	86	94	0	84	93	30
P	93	15	50	43	10	2	1	14	95	5	0	94	5
P→A	4	0	1	1	0	0	0	2	4	0	2	0	0
P→B	93	59	90	91	52	44	43	86	94	37	86	93	0



Table B.46: Structure completeness comparison for the models generated from the 52 synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A	0	0	1	0	1	0	0	1	24	1	0	24	0
A→B	9	0	7	9	13	10	9	5	9	7	4	10	11
A→P*	22	9	0	22	9	1	1	10	21	3	9	20	2
A→P	9	6	29	0	5	2	0	13	10	3	11	10	2
B	6	20	5	7	0	12	13	4	5	11	7	5	12
B→P*	2	14	2	6	20	0	26	7	2	10	7	3	12
B→P	1	17	4	5	23	36	0	7	1	12	8	1	14
P*	2	9	21	29	8	4	3	0	4	4	34	2	3
P*→A	9	0	2	0	0	0	0	0	0	0	0	16	0
P*→B	4	15	6	5	15	12	11	3	4	0	4	5	14
P	4	6	22	31	6	3	2	28	3	5	0	4	6
P→A	9	1	0	2	1	0	0	0	15	0	0	0	0
P→B	5	14	5	5	18	13	11	3	5	15	2	6	0



Table B.47: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A <sub>R-work</sub>	0	96	87	86	95	90	90	97	30	94	97	32	94
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	3	0	5	5	41	7	8	33	1	24	34	0	24
A→B <sub>R-free</sub>	-	0	33	33	42	8	9	35	79	29	34	80	28
A→P* <sub>R-work</sub>	9	95	0	31	93	74	73	100	1	89	100	2	90
A→P* <sub>R-free</sub>	-	59	0	43	57	24	28	52	94	50	51	89	48
A→P <sub>R-work</sub>	10	93	38	0	93	76	75	100	2	89	100	2	89
A→P <sub>R-free</sub>	-	60	43	0	61	26	27	52	93	50	50	90	50
B <sub>R-work</sub>	4	45	5	5	0	2	2	37	1	30	37	1	26
B <sub>R-free</sub>	-	43	34	35	0	5	4	36	82	31	34	82	31
B→P* <sub>R-work</sub>	10	90	19	16	97	0	35	86	2	82	85	3	83
B→P* <sub>R-free</sub>	-	89	72	71	93	0	41	81	96	84	83	95	83
B→P <sub>R-work</sub>	9	90	19	18	97	33	0	84	2	83	83	2	83
B→P <sub>R-free</sub>	-	88	69	70	93	43	0	78	96	82	74	95	81
P* <sub>R-work</sub>	2	61	0	0	57	7	9	0	0	47	33	0	48
P* <sub>R-free</sub>	-	59	45	44	60	12	17	0	93	47	38	93	48
P*→A <sub>R-work</sub>	52	99	97	97	99	96	97	100	0	98	100	34	98
P*→A <sub>R-free</sub>	-	16	5	6	15	4	3	6	0	10	5	43	10
P*→B <sub>R-work</sub>	6	65	7	9	58	12	13	45	2	0	45	2	44
P*→B <sub>R-free</sub>	-	64	45	46	61	13	14	43	88	0	43	84	43
P <sub>R-work</sub>	2	62	0	0	57	4	9	31	0	48	0	0	48
P <sub>R-free</sub>	-	59	45	43	61	13	19	44	95	50	0	94	50
P→A <sub>R-work</sub>	53	99	97	97	98	97	97	100	38	98	100	0	99
P→A <sub>R-free</sub>	-	16	8	7	15	4	5	6	41	12	5	0	10
P→B <sub>R-work</sub>	5	67	8	7	63	12	11	46	1	44	45	1	0
P→B <sub>R-free</sub>	-	63	47	45	62	12	12	48	87	50	46	88	0



Table B.48: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A <sub>R-work</sub>	100	0	4	3	2	1	2	0	17	0	1	15	1
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	0	100	0	2	14	3	3	5	0	11	3	1	9
A→B <sub>R-free</sub>	-	100	8	8	15	3	3	7	4	7	7	4	9
A→P* <sub>R-work</sub>	4	0	100	31	2	7	8	0	2	4	0	1	3
A→P* <sub>R-free</sub>	-	8	100	13	9	3	3	3	2	5	4	3	5
A→P <sub>R-work</sub>	3	2	31	100	2	8	7	0	2	2	0	2	4
A→P <sub>R-free</sub>	-	8	13	100	4	3	3	3	2	4	7	2	5
B <sub>R-work</sub>	2	14	2	2	100	2	1	6	0	12	6	0	10
B <sub>R-free</sub>	-	15	9	4	100	3	3	4	3	8	5	3	7
B→P* <sub>R-work</sub>	1	3	7	8	2	100	32	7	2	6	11	1	5
B→P* <sub>R-free</sub>	-	3	3	3	3	100	16	7	0	2	4	1	5
B→P <sub>R-work</sub>	2	3	8	7	1	32	100	7	2	4	8	2	7
B→P <sub>R-free</sub>	-	3	3	3	3	16	100	5	1	4	8	1	7
P* <sub>R-work</sub>	0	5	0	0	6	7	7	100	0	7	36	0	6
P* <sub>R-free</sub>	-	7	3	3	4	7	5	100	1	9	19	1	4
P*→A <sub>R-work</sub>	17	0	2	2	0	2	2	0	100	0	0	28	0
P*→A <sub>R-free</sub>	-	4	2	2	3	0	1	1	100	2	0	16	3
P*→B <sub>R-work</sub>	0	11	4	2	12	6	4	7	0	100	7	0	12
P*→B <sub>R-free</sub>	-	7	5	4	8	2	4	9	2	100	7	3	7
P <sub>R-work</sub>	1	3	0	0	6	11	8	36	0	7	100	0	7
P <sub>R-free</sub>	-	7	4	7	5	4	8	19	0	7	100	0	5
P→A <sub>R-work</sub>	15	1	1	2	0	1	2	0	28	0	0	100	0
P→A <sub>R-free</sub>	-	4	3	2	3	1	1	1	16	3	0	100	2
P→B <sub>R-work</sub>	1	9	3	4	10	5	7	6	0	12	7	0	100
P→B <sub>R-free</sub>	-	9	5	5	7	5	7	4	3	7	5	2	100



Table B.49: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A <sub>R-work</sub>	0	90	60	58	89	74	74	96	3	85	95	2	86
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	2	0	1	0	9	0	1	16	0	7	16	0	6
A→B <sub>R-free</sub>	-	0	12	12	12	2	1	17	66	10	16	62	8
A→P* <sub>R-work</sub>	4	81	0	1	77	41	41	91	0	68	91	1	68
A→P* <sub>R-free</sub>	-	36	0	5	37	14	12	34	75	29	31	76	28
A→P <sub>R-work</sub>	4	81	1	0	77	43	43	90	0	67	90	1	71
A→P <sub>R-free</sub>	-	34	6	0	37	12	13	31	77	28	32	79	26
B <sub>R-work</sub>	2	13	0	1	0	0	0	18	0	5	17	0	6
B <sub>R-free</sub>	-	16	16	14	0	1	1	16	67	9	17	63	9
B→P* <sub>R-work</sub>	3	69	3	4	66	0	0	45	0	53	45	0	54
B→P* <sub>R-free</sub>	-	68	47	49	68	0	3	47	92	52	45	92	48
B→P <sub>R-work</sub>	2	70	2	3	66	0	0	45	0	52	48	0	52
B→P <sub>R-free</sub>	-	69	51	50	67	2	0	45	91	52	46	91	46
P* <sub>R-work</sub>	1	40	0	0	43	0	1	0	0	29	0	0	26
P* <sub>R-free</sub>	-	39	24	21	34	2	3	0	87	23	2	87	22
P*→A <sub>R-work</sub>	13	95	67	69	96	83	84	100	0	92	100	3	92
P*→A <sub>R-free</sub>	-	5	3	3	7	2	2	3	0	4	4	8	4
P*→B <sub>R-work</sub>	3	27	1	1	18	2	3	22	0	0	22	0	10
P*→B <sub>R-free</sub>	-	27	23	22	22	5	4	20	74	0	21	74	13
P <sub>R-work</sub>	1	40	0	0	40	0	0	0	0	28	0	0	26
P <sub>R-free</sub>	-	40	25	25	39	3	4	5	89	28	0	88	24
P→A <sub>R-work</sub>	13	94	69	67	96	83	83	100	3	92	100	0	92
P→A <sub>R-free</sub>	-	5	2	2	7	2	2	5	8	6	4	0	5
P→B <sub>R-work</sub>	3	26	1	2	25	1	2	23	0	14	22	0	0
P→B <sub>R-free</sub>	-	29	22	22	29	2	2	22	75	14	20	74	0



Table B.50: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the 52 synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A <sub>R-work</sub>	0	6	27	28	6	15	15	2	28	9	2	30	8
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	1	0	4	5	32	6	7	18	1	17	19	0	18
A→B <sub>R-free</sub>	-	0	21	21	30	6	7	18	13	19	18	17	20
A→P* <sub>R-work</sub>	5	14	0	31	17	33	32	9	1	21	9	1	22
A→P* <sub>R-free</sub>	-	22	0	38	20	11	16	18	19	21	20	13	20
A→P <sub>R-work</sub>	7	12	37	0	16	33	31	10	2	21	10	1	18
A→P <sub>R-free</sub>	-	26	38	0	24	14	14	22	16	21	18	12	25
B <sub>R-work</sub>	2	31	4	4	0	1	2	19	1	25	20	1	21
B <sub>R-free</sub>	-	28	18	21	0	3	3	20	16	21	17	19	22
B→P* <sub>R-work</sub>	7	22	16	12	31	0	35	41	2	29	40	2	28
B→P* <sub>R-free</sub>	-	21	25	22	25	0	37	34	4	32	37	2	35
B→P <sub>R-work</sub>	6	20	17	15	30	33	0	39	1	31	35	2	31
B→P <sub>R-free</sub>	-	20	18	20	26	41	0	32	4	30	28	4	35
P* <sub>R-work</sub>	1	21	0	0	14	7	8	0	0	18	33	0	22
P* <sub>R-free</sub>	-	20	21	23	25	10	14	0	6	24	36	7	26
P*→A <sub>R-work</sub>	40	4	30	28	3	13	13	0	0	6	0	31	7
P*→A <sub>R-free</sub>	-	12	2	3	9	2	2	2	0	6	2	35	6
P*→B <sub>R-work</sub>	3	38	6	9	40	9	10	23	2	0	22	1	34
P*→B <sub>R-free</sub>	-	37	22	24	39	8	10	23	13	0	22	11	31
P <sub>R-work</sub>	1	22	0	0	17	4	9	31	0	21	0	0	22
P <sub>R-free</sub>	-	19	20	18	22	10	15	39	5	22	0	7	26
P→A <sub>R-work</sub>	41	5	28	29	3	14	14	0	34	6	0	0	7
P→A <sub>R-free</sub>	-	11	6	5	8	2	2	1	34	6	2	0	6
P→B <sub>R-work</sub>	2	41	7	5	38	11	9	22	1	30	22	1	0
P→B <sub>R-free</sub>	-	33	26	23	32	10	10	26	12	36	26	14	0



## B.3 Experimental results for the original data sets without the Buccaneer development data sets

Table B.51: Complete and intermediate models produced by the 23 pipeline variants for the original data sets, where ‘(T)’ and ‘(C)’ denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	HA-NCS			MR-NCS			NO-NCS		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
A	201	1(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
A→P*	196	3(T) 0(C)	3	197	2(T) 0(C)	3	201	1(T) 0(C)	0
A→B	202	0(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
B	202	0(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
B→P*	200	0(T) 0(C)	2	197	3(T) 0(C)	2	197	4(T) 0(C)	1
P*	198	2(T) 1(C)	1	200	0(T) 1(C)	1	199	1(T) 1(C)	1
P*→A	201	0(T) 0(C)	1	201	0(T) 0(C)	1	200	1(T) 0(C)	1
P*→B	201	0(T) 0(C)	1	201	0(T) 0(C)	1	201	0(T) 0(C)	1
S*	202	0(T) 0(C)	0	201	1(T) 0(C)	0	200	2(T) 0(C)	0
S*→A	202	0(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
S*→B	202	0(T) 0(C)	0	202	0(T) 0(C)	0	202	0(T) 0(C)	0
S*→P*	198	2(T) 0(C)	2	197	3(T) 0(C)	2	196	4(T) 0(C)	2
A→P	-	-	-	-	-	-	199	2(T) 0(C)	1
B→P	-	-	-	-	-	-	200	0(T) 0(C)	2
P	-	-	-	-	-	-	199	1(T) 0(C)	2
P→A	-	-	-	-	-	-	200	0(T) 0(C)	2
P→B	-	-	-	-	-	-	200	0(T) 0(C)	2
S	-	-	-	-	-	-	200	2(T) 0(C)	0
S→A	-	-	-	-	-	-	202	0(T) 0(C)	0
S→B	-	-	-	-	-	-	202	0(T) 0(C)	0
S*→P	-	-	-	-	-	-	197	3(T) 0(C)	2
S→P*	-	-	-	-	-	-	198	2(T) 0(C)	2
S→P	-	-	-	-	-	-	197	3(T) 0(C)	2

Models used in the comparison: 147 HA-NCS, 147 MR-NCS and 148 NO-NCS.



Table B.52: Structure completeness comparison for the models generated from the original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	15	25	29	18	41	28	13	65	43	20	38
A→B	78	0	63	61	41	66	68	33	81	77	53	65
A→P*	53	22	0	45	22	45	41	19	76	52	31	44
B	63	17	44	0	16	50	49	18	73	62	34	49
B→P*	71	35	61	57	0	65	67	35	87	69	48	67
P*	48	27	39	41	21	0	39	16	75	51	31	38
P*→A	46	24	34	40	24	49	0	18	77	47	30	45
P*→B	78	45	73	63	49	76	73	0	87	77	56	72
S*	31	16	20	20	9	20	17	10	0	33	12	20
S*→A	32	14	24	29	19	37	23	15	62	0	23	36
S*→B	70	27	59	47	32	60	59	23	84	67	0	59
S*→P*	50	23	32	40	20	38	35	15	72	49	30	0



Table B.53: Structure completeness comparison for the models generated from the original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	100	7	22	8	11	12	27	9	5	25	10	12
A→B	7	100	16	22	24	7	7	22	3	9	20	12
A→P*	22	16	100	12	17	16	24	8	5	24	10	24
B	8	22	12	100	27	9	11	19	6	9	19	11
B→P*	11	24	17	27	100	14	9	16	4	12	20	13
P*	12	7	16	9	14	100	12	8	5	12	10	24
P*→A	27	7	24	11	9	12	100	9	6	30	11	20
P*→B	9	22	8	19	16	8	9	100	3	8	21	13
S*	5	3	5	6	4	5	6	3	100	5	3	8
S*→A	25	9	24	9	12	12	30	8	5	100	10	15
S*→B	10	20	10	19	20	10	11	21	3	10	100	11
S*→P*	12	12	24	11	13	24	20	13	8	15	11	100



Table B.54: Structure completeness comparison for the models generated from the original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	8	7	13	7	14	5	7	41	24	12	13
A→B	33	0	18	25	11	25	24	8	60	35	14	30
A→P*	28	14	0	22	4	14	14	8	52	31	16	15
B	28	11	14	0	7	22	17	5	52	32	10	22
B→P*	35	20	20	27	0	29	24	10	64	37	20	24
P*	27	15	12	23	3	0	15	7	49	29	18	12
P*→A	21	17	14	24	10	17	0	8	52	30	17	16
P*→B	39	18	26	29	16	33	29	0	65	38	20	35
S*	18	12	6	12	3	8	5	3	0	22	9	4
S*→A	10	10	7	14	6	14	1	6	34	0	8	13
S*→B	33	12	19	25	10	25	22	6	59	35	0	24
S*→P*	27	18	12	25	4	10	16	7	51	31	17	0



Table B.55: Structure completeness comparison for the models generated from the original HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	7	18	16	12	27	22	6	24	19	8	25
A→B	45	0	44	35	30	41	44	25	21	41	39	35
A→P*	25	8	0	23	18	31	27	11	24	22	15	29
B	35	6	30	0	10	29	32	13	22	30	24	27
B→P*	35	16	41	30	0	37	44	25	23	33	28	43
P*	20	12	27	18	18	0	24	9	26	22	13	26
P*→A	24	7	20	16	14	32	0	10	25	17	13	29
P*→B	39	27	47	33	33	43	44	0	22	39	36	37
S*	12	4	14	9	5	12	12	7	0	11	3	16
S*→A	22	4	16	16	13	24	22	9	28	0	15	23
S*→B	37	14	39	22	22	35	37	17	25	32	0	35
S*→P*	22	5	20	15	16	28	19	7	21	18	13	0



Table B.56: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	86	24	93	24	33	20	86	100	33	91	33
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	9	0	3	51	2	2	2	26	98	5	46	3
A→B <sub>R-free</sub>	-	0	5	52	4	1	15	31	-	28	46	4
A→P* <sub>R-work</sub>	64	94	0	98	34	56	54	89	100	61	95	56
A→P* <sub>R-free</sub>	-	90	0	92	33	48	77	81	-	84	88	50
B <sub>R-work</sub>	5	16	1	0	1	3	1	11	99	3	27	2
B <sub>R-free</sub>	-	22	5	0	2	3	11	18	-	25	28	3
B→P* <sub>R-work</sub>	62	96	35	99	0	56	54	96	100	61	97	59
B→P* <sub>R-free</sub>	-	93	40	95	0	48	84	95	-	85	94	52
P* <sub>R-work</sub>	44	93	16	95	19	0	31	93	99	40	95	25
P* <sub>R-free</sub>	-	95	31	95	24	0	73	93	-	75	95	33
P*→A <sub>R-work</sub>	43	94	29	97	27	44	0	93	100	35	95	42
P*→A <sub>R-free</sub>	-	75	17	84	12	16	0	72	-	48	79	12
P*→B <sub>R-work</sub>	10	30	5	56	3	3	3	0	99	10	45	5
P*→B <sub>R-free</sub>	-	39	12	58	3	2	18	0	-	29	45	6
S* <sub>R-work</sub>	0	2	0	1	0	1	0	1	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	34	90	23	95	19	38	16	86	100	0	93	37
S*→A <sub>R-free</sub>	-	63	11	68	9	12	21	63	-	0	64	10
S*→B <sub>R-work</sub>	6	20	2	41	2	3	2	15	100	4	0	2
S*→B <sub>R-free</sub>	-	22	7	47	3	3	12	20	-	29	0	4
S*→P* <sub>R-work</sub>	46	94	15	95	16	26	34	91	100	41	94	0
S*→P* <sub>R-free</sub>	-	90	28	93	21	31	73	91	-	79	95	0



Table B.57: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	100	5	12	2	14	23	37	3	0	33	3	20
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	5	100	3	33	2	5	4	44	0	5	33	3
A→B <sub>R-free</sub>	-	100	5	25	3	4	10	29	-	10	32	5
A→P* <sub>R-work</sub>	12	3	100	1	31	29	17	6	0	16	3	29
A→P* <sub>R-free</sub>	-	5	100	3	27	21	6	7	-	5	4	22
B <sub>R-work</sub>	2	33	1	100	0	1	3	33	0	2	33	3
B <sub>R-free</sub>	-	25	3	100	3	3	5	24	-	7	25	3
B→P* <sub>R-work</sub>	14	2	31	0	100	25	18	1	0	20	1	25
B→P* <sub>R-free</sub>	-	3	27	3	100	28	5	3	-	6	3	27
P* <sub>R-work</sub>	23	5	29	1	25	100	25	4	0	22	3	49
P* <sub>R-free</sub>	-	4	21	3	28	100	12	5	-	13	3	37
P*→A <sub>R-work</sub>	37	4	17	3	18	25	100	4	0	48	3	24
P*→A <sub>R-free</sub>	-	10	6	5	5	12	100	10	-	31	9	14
P*→B <sub>R-work</sub>	3	44	6	33	1	4	4	100	0	4	40	4
P*→B <sub>R-free</sub>	-	29	7	24	3	5	10	100	-	8	35	3
S* <sub>R-work</sub>	0	0	0	0	0	0	0	0	100	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	33	5	16	2	20	22	48	4	0	100	3	22
S*→A <sub>R-free</sub>	-	10	5	7	6	13	31	8	-	100	7	12
S*→B <sub>R-work</sub>	3	33	3	33	1	3	3	40	0	3	100	4
S*→B <sub>R-free</sub>	-	32	4	25	3	3	9	35	-	7	100	1
S*→P* <sub>R-work</sub>	20	3	29	3	25	49	24	4	0	22	4	100
S*→P* <sub>R-free</sub>	-	5	22	3	27	37	14	3	-	12	1	100



Table B.58: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	38	6	48	2	4	1	27	100	6	39	3
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	1	0	0	7	0	0	0	3	94	1	6	0
A→B <sub>R-free</sub>	-	0	3	8	1	0	2	3	-	15	7	0
A→P* <sub>R-work</sub>	4	53	0	65	2	3	2	42	100	5	59	0
A→P* <sub>R-free</sub>	-	48	0	59	2	1	10	39	-	25	52	0
B <sub>R-work</sub>	0	4	0	0	0	0	0	3	95	1	5	0
B <sub>R-free</sub>	-	4	2	0	0	0	2	4	-	14	5	0
B→P* <sub>R-work</sub>	5	52	5	63	0	3	2	41	100	7	53	0
B→P* <sub>R-free</sub>	-	54	8	62	0	3	12	40	-	27	55	1
P* <sub>R-work</sub>	4	39	3	50	1	0	0	27	99	6	41	0
P* <sub>R-free</sub>	-	40	9	50	1	0	12	28	-	25	40	0
P*→A <sub>R-work</sub>	6	39	8	53	3	5	0	32	100	7	42	5
P*→A <sub>R-free</sub>	-	22	7	31	1	1	0	17	-	16	26	0
P*→B <sub>R-work</sub>	1	12	1	14	0	0	0	0	99	3	13	0
P*→B <sub>R-free</sub>	-	12	5	14	1	0	2	0	-	16	14	0
S* <sub>R-work</sub>	0	1	0	1	0	0	0	1	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	6	38	5	50	2	3	3	29	100	0	42	1
S*→A <sub>R-free</sub>	-	16	4	22	1	1	0	14	-	0	19	0
S*→B <sub>R-work</sub>	1	7	1	7	0	1	0	2	99	0	0	0
S*→B <sub>R-free</sub>	-	7	5	9	1	1	1	2	-	12	0	0
S*→P* <sub>R-work</sub>	5	41	4	52	1	1	1	28	100	7	41	0
S*→P* <sub>R-free</sub>	-	44	9	49	1	1	9	30	-	24	44	0



Table B.59: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	48	18	45	22	29	18	60	0	27	52	30
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	8	0	3	44	2	2	2	23	4	4	40	3
A→B <sub>R-free</sub>	-	0	3	44	3	1	13	29	-	13	39	4
A→P* <sub>R-work</sub>	60	41	0	33	32	53	52	47	0	56	36	56
A→P* <sub>R-free</sub>	-	42	0	33	31	47	67	42	-	59	36	50
B <sub>R-work</sub>	5	12	1	0	1	3	1	7	4	3	21	2
B <sub>R-free</sub>	-	18	3	0	2	3	9	14	-	12	23	3
B→P* <sub>R-work</sub>	56	44	31	36	0	53	52	55	0	54	44	59
B→P* <sub>R-free</sub>	-	39	32	33	0	46	72	54	-	58	39	52
P* <sub>R-work</sub>	40	53	12	46	18	0	31	65	0	34	53	25
P* <sub>R-free</sub>	-	54	22	45	23	0	61	65	-	50	54	33
P*→A <sub>R-work</sub>	37	55	21	44	24	38	0	61	0	29	53	37
P*→A <sub>R-free</sub>	-	52	10	52	11	14	0	55	-	32	53	12
P*→B <sub>R-work</sub>	9	18	4	42	3	3	3	0	0	7	32	5
P*→B <sub>R-free</sub>	-	28	6	44	2	2	16	0	-	13	31	6
S* <sub>R-work</sub>	0	1	0	0	0	1	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	28	52	18	45	17	35	14	58	0	0	50	35
S*→A <sub>R-free</sub>	-	46	7	46	8	12	21	49	-	0	45	10
S*→B <sub>R-work</sub>	5	14	1	33	2	2	2	13	1	4	0	2
S*→B <sub>R-free</sub>	-	16	3	38	2	2	11	18	-	17	0	4
S*→P* <sub>R-work</sub>	41	53	11	42	15	24	33	63	0	33	52	0
S*→P* <sub>R-free</sub>	-	47	19	44	20	29	65	61	-	55	50	0

0 72

Table B.60: Structure completeness comparison for the models generated from the original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	14	24	33	23	41	31	18	72	50	25	45
A→B	76	0	67	64	45	70	67	35	86	80	52	70
A→P*	48	17	0	46	24	48	38	21	79	54	34	54
B	56	21	45	0	20	46	50	21	78	61	35	55
B→P*	63	34	59	54	0	63	64	32	87	68	50	71
P*	43	21	31	40	20	0	40	19	76	50	33	43
P*→A	43	23	31	40	24	46	0	21	80	51	32	48
P*→B	72	41	70	59	48	76	69	0	86	74	51	78
S*	23	12	16	16	10	17	16	12	0	28	13	18
S*→A	24	13	19	30	20	35	17	18	63	0	25	37
S*→B	60	26	51	46	33	59	54	27	86	67	0	61
S*→P*	41	20	29	35	17	31	33	12	73	46	29	0



Table B.61: Structure completeness comparison for the models generated from the original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	100	11	28	10	14	16	27	10	5	26	15	14
A→B	11	100	16	15	21	9	10	24	2	7	22	10
A→P*	28	16	100	9	16	20	31	9	5	27	15	16
B	10	15	9	100	25	14	10	20	6	10	19	10
B→P*	14	21	16	25	100	17	12	20	3	12	18	12
P*	16	9	20	14	17	100	14	5	7	16	7	27
P*→A	27	10	31	10	12	14	100	10	4	32	14	18
P*→B	10	24	9	20	20	5	10	100	3	7	22	10
S*	5	2	5	6	3	7	4	3	100	9	1	8
S*→A	26	7	27	10	12	16	32	7	9	100	7	16
S*→B	15	22	15	19	18	7	14	22	1	7	100	10
S*→P*	14	10	16	10	12	27	18	10	8	16	10	100



Table B.62: Structure completeness comparison for the models generated from the original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	8	6	16	7	16	7	10	43	20	14	14
A→B	29	0	18	25	13	22	19	10	63	31	18	27
A→P*	24	12	0	19	8	13	16	12	54	29	18	14
B	26	14	15	0	8	20	19	9	55	35	12	21
B→P*	30	18	20	25	0	27	20	14	63	36	21	22
P*	26	13	12	23	5	0	15	9	52	28	19	11
P*→A	15	13	14	21	10	17	0	11	53	24	19	16
P*→B	30	17	23	27	15	28	22	0	60	35	20	33
S*	16	9	7	8	4	5	5	5	0	20	9	5
S*→A	8	7	6	14	6	14	1	9	38	0	10	13
S*→B	27	12	17	20	11	18	18	11	61	31	0	22
S*→P*	22	14	11	22	5	7	15	10	51	24	18	0



Table B.63: Structure completeness comparison for the models generated from the original MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A	0	5	18	18	16	24	23	8	29	31	11	31
A→B	47	0	49	39	32	48	48	25	24	49	35	43
A→P*	24	5	0	27	16	35	22	10	25	25	16	40
B	31	7	30	0	12	26	31	12	22	25	24	34
B→P*	33	16	39	29	0	36	44	18	24	32	29	48
P*	17	8	20	17	15	0	25	10	24	22	14	32
P*→A	28	10	18	19	15	29	0	10	27	27	13	32
P*→B	42	24	47	33	33	48	46	0	26	39	31	45
S*	7	3	10	8	5	12	11	6	0	8	4	14
S*→A	16	5	13	16	14	21	16	10	25	0	15	24
S*→B	33	14	34	25	22	41	35	16	24	36	0	39
S*→P*	19	6	18	12	12	23	18	3	22	22	12	0





Table B.64: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	88	21	91	27	35	18	90	100	31	93	37
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	7	0	3	52	2	2	2	32	99	5	45	3
A→B <sub>R-free</sub>	-	0	6	54	5	3	16	35	-	30	48	5
A→P* <sub>R-work</sub>	64	93	0	97	37	61	56	89	100	61	93	59
A→P* <sub>R-free</sub>	-	89	0	91	36	54	78	80	-	88	88	55
B <sub>R-work</sub>	6	18	3	0	1	3	1	17	99	5	25	3
B <sub>R-free</sub>	-	22	7	0	2	4	11	24	-	24	30	4
B→P* <sub>R-work</sub>	59	96	31	99	0	54	52	95	100	57	97	61
B→P* <sub>R-free</sub>	-	89	37	95	0	50	77	93	-	84	93	53
P* <sub>R-work</sub>	45	93	18	93	20	0	33	94	100	40	95	26
P* <sub>R-free</sub>	-	93	23	93	29	0	70	95	-	76	96	30
P*→A <sub>R-work</sub>	40	95	27	95	30	42	0	94	100	35	95	45
P*→A <sub>R-free</sub>	-	74	15	85	12	12	0	73	-	50	80	16
P*→B <sub>R-work</sub>	6	28	6	51	3	3	4	0	100	8	39	5
P*→B <sub>R-free</sub>	-	31	13	54	4	2	18	0	-	27	44	6
S* <sub>R-work</sub>	0	1	0	1	0	0	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	33	90	21	93	24	38	16	91	100	0	92	41
S*→A <sub>R-free</sub>	-	65	8	68	11	9	16	67	-	0	69	13
S*→B <sub>R-work</sub>	5	19	3	39	1	2	3	21	100	4	0	2
S*→B <sub>R-free</sub>	-	20	8	43	4	2	12	27	-	28	0	5
S*→P* <sub>R-work</sub>	42	94	14	94	17	20	31	90	100	37	95	0
S*→P* <sub>R-free</sub>	-	90	22	93	25	32	72	90	-	78	94	0



Table B.65: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <i>R-work</i>	100	5	15	3	15	20	41	3	0	36	1	20
A <i>R-free</i>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	5	100	4	30	2	5	3	40	0	5	36	3
A→B <i>R-free</i>	-	100	5	23	5	3	10	35	-	5	32	5
A→P* <i>R-work</i>	15	4	100	1	33	22	17	5	0	18	4	28
A→P* <i>R-free</i>	-	5	100	1	27	23	7	7	-	4	3	23
B <i>R-work</i>	3	30	1	100	0	3	4	32	0	3	36	3
B <i>R-free</i>	-	23	1	100	3	3	4	22	-	7	27	3
B→P* <i>R-work</i>	15	2	33	0	100	27	18	2	0	19	2	22
B→P* <i>R-free</i>	-	5	27	3	100	21	12	3	-	5	3	22
P* <i>R-work</i>	20	5	22	3	27	100	25	3	0	22	3	54
P* <i>R-free</i>	-	3	23	3	21	100	18	3	-	15	2	38
P*→A <i>R-work</i>	41	3	17	4	18	25	100	2	0	48	2	24
P*→A <i>R-free</i>	-	10	7	4	12	18	100	9	-	34	8	12
P*→B <i>R-work</i>	3	40	5	32	2	3	2	100	0	1	40	5
P*→B <i>R-free</i>	-	35	7	22	3	3	9	100	-	6	29	4
S* <i>R-work</i>	0	0	0	0	0	0	0	0	100	0	0	0
S* <i>R-free</i>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <i>R-work</i>	36	5	18	3	19	22	48	1	0	100	4	22
S*→A <i>R-free</i>	-	5	4	7	5	15	34	6	-	100	3	9
S*→B <i>R-work</i>	1	36	4	36	2	3	2	40	0	4	100	3
S*→B <i>R-free</i>	-	32	3	27	3	2	8	29	-	3	100	1
S*→P* <i>R-work</i>	20	3	28	3	22	54	24	5	0	22	3	100
S*→P* <i>R-free</i>	-	5	23	3	22	38	12	4	-	9	1	100



Table B.66: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	34	6	48	3	3	2	31	100	5	41	4
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	0	0	0	10	1	0	0	6	95	1	8	0
A→B <sub>R-free</sub>	-	0	3	12	1	0	2	6	-	16	9	0
A→P* <sub>R-work</sub>	2	50	0	65	2	1	2	48	100	3	56	0
A→P* <sub>R-free</sub>	-	44	0	59	3	1	10	43	-	24	54	1
B <sub>R-work</sub>	1	5	1	0	0	0	1	7	97	1	5	0
B <sub>R-free</sub>	-	5	3	0	1	0	3	6	-	14	3	0
B→P* <sub>R-work</sub>	5	50	5	62	0	2	2	45	100	5	56	0
B→P* <sub>R-free</sub>	-	49	9	61	0	2	12	46	-	29	56	1
P* <sub>R-work</sub>	3	35	4	48	1	0	1	31	100	5	42	0
P* <sub>R-free</sub>	-	37	9	49	1	0	10	31	-	26	44	1
P*→A <sub>R-work</sub>	5	36	8	54	3	4	0	38	100	7	47	5
P*→A <sub>R-free</sub>	-	18	7	28	1	1	0	18	-	15	26	1
P*→B <sub>R-work</sub>	1	12	1	14	0	0	0	0	98	2	12	0
P*→B <sub>R-free</sub>	-	11	7	14	1	0	2	0	-	15	13	0
S* <sub>R-work</sub>	0	0	0	1	0	0	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	5	34	4	49	1	1	3	33	100	0	42	1
S*→A <sub>R-free</sub>	-	10	4	20	1	0	0	17	-	0	20	0
S*→B <sub>R-work</sub>	1	7	1	7	0	0	1	5	99	0	0	0
S*→B <sub>R-free</sub>	-	5	5	7	0	0	2	5	-	11	0	0
S*→P* <sub>R-work</sub>	4	34	5	48	1	1	2	33	100	5	43	0
S*→P* <sub>R-free</sub>	-	38	10	48	1	1	8	34	-	24	46	0



Table B.67: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B	S*	S*→A	S*→B	S*→P*
A <sub>R-work</sub>	0	54	15	44	23	31	16	59	0	25	52	33
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	7	0	3	41	1	2	2	26	3	4	37	3
A→B <sub>R-free</sub>	-	0	3	43	5	3	14	29	-	14	39	5
A→P* <sub>R-work</sub>	62	44	0	31	35	59	54	41	0	58	36	59
A→P* <sub>R-free</sub>	-	46	0	32	33	53	69	37	-	64	34	54
B <sub>R-work</sub>	5	14	2	0	1	3	1	10	1	4	20	3
B <sub>R-free</sub>	-	18	5	0	1	4	8	18	-	10	27	4
B→P* <sub>R-work</sub>	54	46	26	37	0	52	50	50	0	52	40	61
B→P* <sub>R-free</sub>	-	40	29	34	0	48	65	46	-	54	37	52
P* <sub>R-work</sub>	41	57	14	46	19	0	31	63	0	35	53	26
P* <sub>R-free</sub>	-	56	14	44	29	0	60	63	-	50	52	29
P*→A <sub>R-work</sub>	35	59	19	41	27	38	0	56	0	29	48	40
P*→A <sub>R-free</sub>	-	56	7	57	11	12	0	56	-	35	54	15
P*→B <sub>R-work</sub>	5	16	5	37	3	3	4	0	2	6	27	5
P*→B <sub>R-free</sub>	-	20	6	41	3	2	16	0	-	12	31	6
S* <sub>R-work</sub>	0	1	0	0	0	0	0	0	0	0	0	0
S* <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-
S*→A <sub>R-work</sub>	29	56	17	44	23	37	13	59	0	0	50	39
S*→A <sub>R-free</sub>	-	55	4	48	10	9	16	50	-	0	48	13
S*→B <sub>R-work</sub>	5	12	2	31	1	2	2	16	1	4	0	2
S*→B <sub>R-free</sub>	-	15	3	36	4	2	10	22	-	17	0	5
S*→P* <sub>R-work</sub>	38	60	9	46	16	19	29	58	0	33	52	0
S*→P* <sub>R-free</sub>	-	52	12	45	24	31	64	56	-	54	48	0

0 69

Table B.68: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>*</sup>	S <sup>*</sup> →A	S <sup>*</sup> →B	S <sup>*</sup> →P <sup>s</sup>	S <sup>*</sup> →P	S→P <sup>s</sup>	S→P
A	0	18	26	29	33	21	20	39	26	18	37	31	21	68	42	24	61	41	26	39	39	38	40
A→B	74	0	59	63	64	41	41	66	60	30	68	65	36	85	74	43	80	71	48	64	64	57	62
A→P <sup>s</sup>	52	30	0	26	51	24	24	45	37	22	49	39	28	81	57	32	78	50	39	46	50	49	47
A→P	49	28	17	0	45	25	26	43	36	20	47	39	24	78	54	30	76	50	36	43	45	44	45
B	57	22	40	45	0	20	21	46	45	18	49	51	24	77	59	28	72	57	33	51	44	47	49
B→P <sup>s</sup>	68	43	59	58	59	0	24	65	59	32	65	59	41	86	66	49	82	68	48	66	60	64	62
B→P	70	42	59	59	55	22	0	64	59	34	67	59	40	87	66	48	84	70	46	59	63	64	63
P <sup>s</sup>	49	28	37	37	45	23	26	0	36	19	46	43	25	80	48	34	77	49	35	45	41	36	41
P <sup>s</sup> →A	49	30	35	37	44	29	30	47	0	22	50	32	27	82	54	34	78	45	36	49	49	48	54
P <sup>s</sup> →B	76	51	70	70	68	49	49	72	67	0	76	71	47	87	76	54	85	74	57	72	72	70	75
P	48	25	34	34	41	18	19	32	37	18	0	37	20	78	47	32	72	49	29	34	30	31	34
P→A	45	27	31	30	41	24	28	43	26	22	45	0	24	81	49	31	76	45	32	48	44	46	49
P→B	72	41	67	69	61	44	44	64	61	36	74	68	0	85	76	50	82	72	50	68	67	61	68
S	26	12	11	14	20	11	10	16	10	12	16	13	13	0	32	11	34	24	9	15	15	11	16
S→A	30	18	28	28	33	22	25	33	24	12	39	30	16	64	0	21	60	25	20	36	35	32	36
S→B	68	36	55	58	55	34	35	59	56	26	61	57	34	89	74	0	84	68	43	61	57	59	59
S <sup>*</sup>	32	16	17	17	24	12	12	17	17	12	22	18	14	57	34	15	0	32	14	19	16	14	20
S <sup>*</sup> →A	31	19	27	29	35	22	24	36	24	17	41	27	20	68	41	24	64	0	26	41	36	38	39
S <sup>*</sup> →B	64	32	50	50	55	34	35	53	53	22	61	55	28	89	67	38	81	68	0	52	53	55	58
S <sup>*</sup> →P <sup>s</sup>	49	28	35	36	43	24	24	34	31	17	41	38	22	80	47	32	74	48	36	0	34	37	37
S <sup>*</sup> →P	49	28	36	36	46	26	28	35	35	18	45	42	23	82	47	35	76	49	39	34	0	33	39
S→P <sup>s</sup>	52	29	36	39	45	23	23	34	34	20	47	43	24	84	54	34	81	51	34	40	39	0	39
S→P	49	28	34	35	46	23	24	33	34	20	47	39	22	80	52	32	76	50	32	39	36	34	0

0 89

Table B.69: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>*</sup>	S <sup>*</sup> →A	S <sup>*</sup> →B	S <sup>*</sup> →P <sup>s</sup>	S <sup>*</sup> →P	S→P <sup>s</sup>	S→P
A	100	7	22	22	9	11	11	12	25	6	15	24	7	5	28	7	7	28	9	12	12	10	11
A→B	7	100	11	9	14	17	17	6	9	19	7	8	22	3	7	21	4	10	20	8	8	14	9
A→P <sup>s</sup>	22	11	100	57	9	16	17	18	28	8	18	30	5	7	15	13	5	23	11	19	14	15	20
A→P	22	9	57	100	11	17	14	20	26	11	19	31	7	7	18	11	7	21	14	21	18	18	20
B	9	14	9	11	100	21	24	9	11	14	11	9	16	3	8	17	4	7	11	7	10	9	5
B→P <sup>s</sup>	11	17	16	17	21	100	54	12	12	19	18	17	15	2	11	18	5	9	18	11	14	14	15
B→P	11	17	17	14	24	54	100	10	11	18	14	13	16	3	9	17	4	7	19	16	9	14	13
P <sup>s</sup>	12	6	18	20	9	12	10	100	17	9	22	14	11	4	19	7	6	15	11	22	24	30	26
P <sup>s</sup> →A	25	9	28	26	11	12	11	17	100	11	13	42	12	7	22	10	5	32	11	20	16	18	12
P <sup>s</sup> →B	6	19	8	11	14	19	18	9	11	100	7	7	17	1	11	20	3	9	21	11	9	10	5
P	15	7	18	19	11	18	14	22	13	7	100	18	7	6	15	7	7	10	10	25	26	22	18
P→A	24	8	30	31	9	17	13	14	42	7	18	100	7	6	21	12	5	28	13	14	14	11	13
P→B	7	22	5	7	16	15	16	11	12	17	7	7	100	2	8	16	4	8	22	10	10	14	10
S	5	3	7	7	3	2	3	4	7	1	6	6	2	100	3	1	8	7	1	5	3	5	4
S→A	28	7	15	18	8	11	9	19	22	11	15	21	8	3	100	5	5	34	13	17	18	14	11
S→B	7	21	13	11	17	18	17	7	10	20	7	12	16	1	5	100	1	8	20	6	7	7	9
S <sup>*</sup>	7	4	5	7	4	5	4	6	5	3	7	5	4	8	5	1	100	5	5	7	8	5	4
S <sup>*</sup> →A	28	10	23	21	7	9	7	15	32	9	10	28	8	7	34	8	5	100	7	11	15	11	11
S <sup>*</sup> →B	9	20	11	14	11	18	19	11	11	21	10	13	22	1	13	20	5	7	100	12	9	11	9
S <sup>*</sup> →P <sup>s</sup>	12	8	19	21	7	11	16	22	20	11	25	14	10	5	17	6	7	11	12	100	32	23	24
S <sup>*</sup> →P	12	8	14	18	10	14	9	24	16	9	26	14	10	3	18	7	8	15	9	32	100	28	25
S→P <sup>s</sup>	10	14	15	18	9	14	14	30	18	10	22	11	14	5	14	7	5	11	11	23	28	100	27
S→P	11	9	20	20	5	15	13	26	12	5	18	13	10	4	11	9	4	11	9	24	25	27	100

1 100

Table B.70: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>s</sup>	S <sup>s</sup> →A	S <sup>s</sup> →B	S <sup>s</sup> →P <sup>s</sup>	S <sup>s</sup> →P	S→P <sup>s</sup>	S→P
A	0	7	5	6	15	7	8	11	5	10	14	4	12	45	22	13	40	20	14	14	14	14	13
A→B	30	0	13	17	30	11	14	24	16	9	28	20	11	64	34	18	57	29	18	24	24	21	25
A→P <sup>s</sup>	30	16	0	9	28	9	10	13	10	11	21	16	14	61	31	20	57	30	21	14	14	15	14
A→P	28	16	6	0	26	9	10	12	11	10	20	16	13	61	31	21	56	28	21	15	14	11	13
B	28	12	11	11	0	9	7	16	13	8	16	18	8	56	33	14	48	29	14	22	17	16	18
B→P <sup>s</sup>	36	24	20	23	32	0	4	24	16	14	30	25	17	66	35	26	63	33	24	23	21	20	21
B→P	35	24	20	24	30	5	0	21	17	14	29	24	18	64	37	26	62	34	24	22	22	22	23
P <sup>s</sup>	28	18	14	14	26	9	9	0	13	9	14	17	13	61	30	24	55	27	22	10	6	9	9
P <sup>s</sup> →A	26	20	14	18	28	13	12	19	0	10	22	12	15	59	30	24	57	28	23	19	20	14	16
P <sup>s</sup> →B	33	19	23	25	35	18	21	30	22	0	30	29	16	70	40	24	63	34	24	30	30	28	32
P	28	16	11	15	23	8	7	7	14	9	0	15	13	57	32	21	51	26	20	11	9	7	7
P→A	26	18	12	13	26	10	11	15	7	11	16	0	13	58	30	21	54	26	20	16	13	11	18
P→B	35	19	20	20	34	17	18	26	20	11	32	25	0	65	39	24	63	36	24	28	27	25	27
S	17	10	7	6	11	5	5	7	3	5	7	7	6	0	24	7	9	17	3	7	7	6	5
S→A	9	11	6	7	16	8	9	11	1	8	15	5	10	38	0	9	35	7	7	13	11	9	9
S→B	36	15	18	18	26	12	11	22	18	12	28	24	11	70	38	0	59	38	18	28	24	24	24
S <sup>s</sup>	21	13	7	6	17	5	5	5	6	5	10	7	7	32	27	11	0	20	9	5	6	5	5
S <sup>s</sup> →A	12	12	7	9	20	8	9	12	2	9	16	3	11	47	16	11	40	0	9	14	13	11	10
S <sup>s</sup> →B	31	14	16	17	28	11	12	19	12	9	23	20	11	65	35	12	60	30	0	22	20	22	20
S <sup>s</sup> →P <sup>s</sup>	28	19	12	14	28	8	9	9	12	10	18	17	14	61	30	21	53	24	22	0	5	5	7
S <sup>s</sup> →P	30	19	14	14	32	7	9	10	12	11	17	16	14	63	33	24	52	27	20	7	0	3	7
S→P <sup>s</sup>	30	21	14	16	30	9	10	13	15	11	16	18	14	60	29	24	53	26	23	11	10	0	11
S→P	28	18	13	13	28	9	8	9	12	10	15	18	14	59	30	22	55	25	21	9	5	5	0



Table B.71: Structure completeness comparison for the models generated from the original NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>s</sup>	S <sup>s</sup> →A	S <sup>s</sup> →B	S <sup>s</sup> →P <sup>s</sup>	S <sup>s</sup> →P	S→P <sup>s</sup>	S→P
A	0	11	22	23	18	14	11	27	21	8	23	27	9	24	20	11	21	21	13	25	26	24	27
A→B	45	0	46	46	34	29	28	42	44	21	40	45	25	21	41	24	23	42	30	40	41	36	37
A→P <sup>s</sup>	22	14	0	18	23	16	14	32	27	11	28	23	14	20	26	12	21	20	18	32	36	34	33
A→P	20	12	11	0	18	16	16	31	26	9	26	24	11	18	23	9	20	22	15	28	32	32	32
B	30	10	29	33	0	11	14	30	32	10	32	33	16	21	26	14	24	28	19	29	27	30	31
B→P <sup>s</sup>	32	19	39	35	27	0	20	41	43	19	35	34	24	20	31	22	20	35	24	43	39	44	41
B→P	34	18	40	36	26	18	0	43	42	20	38	35	22	23	28	22	22	36	22	37	41	42	40
P <sup>s</sup>	21	10	23	24	19	14	17	0	24	9	32	26	12	20	18	10	22	22	13	34	34	27	32
P <sup>s</sup> →A	22	11	21	19	16	16	18	28	0	12	28	20	12	23	24	10	20	16	13	30	29	34	38
P <sup>s</sup> →B	43	32	47	45	33	30	28	41	45	0	46	42	31	18	36	30	22	40	33	42	43	43	43
P	20	9	22	20	18	9	12	25	24	9	0	22	7	21	15	11	21	23	9	23	21	24	28
P→A	18	9	19	17	15	14	18	28	19	10	29	0	11	23	19	10	22	19	13	32	31	34	30
P→B	36	22	47	49	26	27	26	39	41	26	41	43	0	20	36	26	20	36	26	41	40	36	41
S	9	2	5	8	9	7	5	9	7	7	9	5	7	0	8	3	25	7	6	8	7	5	11
S→A	21	7	22	22	17	14	16	22	24	4	24	25	6	26	0	11	25	18	13	23	24	23	27
S→B	32	22	36	40	30	22	24	37	39	14	32	33	22	19	36	0	26	30	25	34	34	35	34
S <sup>s</sup>	11	3	10	11	7	7	7	11	11	7	11	11	6	26	7	4	0	11	5	14	9	9	14
S <sup>s</sup> →A	19	7	20	20	15	14	15	24	22	8	26	24	9	22	26	13	24	0	17	27	23	26	28
S <sup>s</sup> →B	33	19	34	33	28	24	23	34	41	14	38	35	16	24	32	26	21	37	0	30	32	33	39
S <sup>s</sup> →P <sup>s</sup>	22	9	23	22	14	16	16	25	19	7	22	21	7	20	18	11	22	24	14	0	29	32	30
S <sup>s</sup> →P	19	9	23	22	14	18	20	25	23	7	28	26	9	19	14	11	24	22	18	27	0	30	32
S→P <sup>s</sup>	22	8	22	22	15	14	13	21	20	9	31	25	10	24	25	10	28	25	11	29	29	0	28
S→P	20	10	21	22	18	14	16	24	22	9	32	21	9	21	22	9	21	25	11	30	31	29	0



APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.72: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>s</sup>	S <sup>s</sup> →A	S <sup>s</sup> →B	S <sup>s</sup> →P	S <sup>s</sup> →P <sup>s</sup>	S <sup>s</sup> →P	S <sup>s</sup> →P
A <sub>R-work</sub>	0	90	24	24	94	27	26	34	16	88	37	18	88	100	32	90	100	30	93	34	31	29	32
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	6	0	2	3	58	5	3	3	2	29	3	3	32	97	5	43	97	4	45	5	4	3	3
A→B <sub>R-free</sub>	-	0	5	7	59	5	4	2	11	32	5	14	30	-	30	41	-	25	46	5	4	3	3
A→P <sup>s</sup> <sub>R-work</sub>	64	95	0	12	99	32	34	56	51	91	62	57	93	100	64	93	100	62	95	55	57	54	56
A→P <sup>s</sup> <sub>R-free</sub>	-	89	0	26	93	33	36	51	75	79	53	76	84	-	89	86	-	86	88	50	52	51	54
A→P <sub>R-work</sub>	64	95	14	0	99	33	33	58	50	91	64	57	93	100	63	94	100	63	95	57	59	55	57
A→P <sub>R-free</sub>	-	89	23	0	93	35	36	52	78	83	52	77	86	-	89	86	-	85	89	51	53	53	54
B <sub>R-work</sub>	5	17	1	1	0	2	2	3	1	11	3	1	18	97	1	23	97	1	26	3	2	2	1
B <sub>R-free</sub>	-	20	5	5	0	3	3	3	10	19	5	12	14	-	28	24	-	23	29	3	2	3	3
B→P <sup>s</sup> <sub>R-work</sub>	61	94	33	32	97	0	14	52	46	94	55	52	93	100	59	93	100	58	95	55	48	50	50
B→P <sup>s</sup> <sub>R-free</sub>	-	93	43	39	95	0	27	50	78	94	50	81	91	-	84	92	-	80	93	51	45	46	49
B→P <sub>R-work</sub>	61	94	36	32	97	17	0	54	46	95	57	51	94	99	58	93	99	59	95	51	49	49	51
B→P <sub>R-free</sub>	-	92	47	41	95	30	0	47	77	93	54	80	91	-	82	91	-	80	93	51	47	47	51
P <sup>s</sup> <sub>R-work</sub>	47	93	20	16	95	24	22	0	30	93	27	32	95	99	44	93	99	41	95	20	24	19	19
P <sup>s</sup> <sub>R-free</sub>	-	95	30	28	93	28	24	0	70	93	31	70	93	-	82	93	-	78	95	26	25	24	24
P <sup>s</sup> →A <sub>R-work</sub>	48	94	34	31	99	39	36	49	0	95	51	25	96	100	44	96	100	41	97	47	51	43	47
P <sup>s</sup> →A <sub>R-free</sub>	-	78	18	18	86	16	18	20	0	75	19	35	78	-	53	80	-	51	82	16	11	12	13
P <sup>s</sup> →B <sub>R-work</sub>	8	31	7	6	61	5	4	4	1	0	5	4	32	99	12	45	99	9	47	7	4	3	3
P <sup>s</sup> →B <sub>R-free</sub>	-	35	16	14	64	4	4	3	14	0	7	18	34	-	30	41	-	26	49	6	3	3	3
P <sub>R-work</sub>	43	93	14	14	93	21	19	22	29	89	0	30	93	99	43	92	99	39	93	17	17	16	16
P <sub>R-free</sub>	-	91	30	29	93	27	26	31	72	88	0	72	91	-	80	90	-	78	92	27	26	26	24
P→A <sub>R-work</sub>	47	92	27	26	96	33	32	44	21	93	47	0	94	100	42	95	100	39	97	41	46	41	45
P→A <sub>R-free</sub>	-	74	15	14	80	15	15	13	28	72	14	0	72	-	44	74	-	47	80	14	12	11	14
P→B <sub>R-work</sub>	5	34	5	5	59	5	3	2	2	23	2	2	0	99	9	40	99	5	43	5	3	2	3
P→B <sub>R-free</sub>	-	39	12	11	59	5	5	3	13	33	5	18	0	-	34	46	-	27	48	7	4	5	3
S <sub>R-work</sub>	0	2	0	0	3	0	1	1	0	1	1	0	1	0	0	0	19	0	0	0	0	0	0
S <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S→A <sub>R-work</sub>	32	94	21	19	98	24	25	36	16	85	41	20	85	100	0	93	100	20	95	34	34	32	31
S→A <sub>R-free</sub>	-	61	8	7	64	11	14	9	22	64	13	22	61	-	0	59	-	26	61	9	8	9	9
S→B <sub>R-work</sub>	6	24	5	4	47	5	5	1	1	19	5	2	26	100	2	0	100	2	34	3	2	3	3
S→B <sub>R-free</sub>	-	28	11	10	50	6	7	5	11	25	7	18	24	-	34	0	-	28	38	5	5	3	4
S <sup>s</sup> <sub>R-work</sub>	0	2	0	0	3	0	1	1	0	1	1	0	1	42	0	0	0	0	0	0	0	0	0
S <sup>s</sup> <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S <sup>s</sup> →A <sub>R-work</sub>	35	92	22	21	95	26	26	36	15	89	39	17	89	100	30	93	100	0	95	35	37	34	36
S <sup>s</sup> →A <sub>R-free</sub>	-	69	11	10	71	12	12	11	19	69	12	23	66	-	36	69	-	0	68	11	9	9	10
S <sup>s</sup> →B <sub>R-work</sub>	4	23	2	2	45	4	3	3	2	18	4	3	20	100	1	25	100	2	0	4	2	3	2
S <sup>s</sup> →B <sub>R-free</sub>	-	24	7	7	52	5	5	3	10	24	5	13	24	-	30	31	-	24	0	5	4	3	3
S <sup>s</sup> →P <sup>s</sup> <sub>R-work</sub>	48	93	18	14	94	24	18	24	32	91	30	34	92	100	42	94	100	42	93	0	19	18	18
S <sup>s</sup> →P <sup>s</sup> <sub>R-free</sub>	-	90	32	29	93	26	26	29	74	90	36	72	89	-	85	91	-	80	93	0	27	22	27
S <sup>s</sup> →P <sub>R-work</sub>	47	93	20	18	95	20	18	24	32	92	28	34	93	100	43	92	100	41	93	16	0	17	16
S <sup>s</sup> →P <sub>R-free</sub>	-	93	32	27	94	26	26	31	69	91	34	74	93	-	81	92	-	80	93	28	0	26	25
S→P <sup>s</sup> <sub>R-work</sub>	48	93	22	20	96	24	20	24	31	93	34	33	93	100	45	94	100	42	95	22	27	0	20
S→P <sup>s</sup> <sub>R-free</sub>	-	93	30	28	95	27	26	34	76	92	36	76	95	-	83	92	-	80	94	28	30	0	33
S→P <sub>R-work</sub>	46	93	20	18	96	26	21	25	30	93	31	35	94	100	41	93	100	42	94	20	22	16	0
S→P <sub>R-free</sub>	-	93	31	29	95	32	27	32	70	92	38	74	93	-	84	94	-	78	95	27	24	27	0



APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.73: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>*</sup>	S <sup>*</sup> →A	S <sup>*</sup> →B	S <sup>*</sup> →P <sup>s</sup>	S <sup>*</sup> →P	S→P <sup>s</sup>	S→P
A <sub>R-work</sub>	100	4	12	12	1	11	13	20	36	4	20	35	7	0	36	4	0	35	3	18	22	23	22
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	4	100	3	1	25	1	3	4	4	40	3	5	34	1	1	32	1	4	32	1	3	3	3
A→B <sub>R-free</sub>	-	100	6	4	21	3	4	3	10	33	4	12	31	-	9	30	-	6	30	5	3	3	3
A→P <sup>s</sup> <sub>R-work</sub>	12	3	100	74	1	34	30	24	15	3	24	16	2	0	16	2	0	16	3	28	24	24	24
A→P <sup>s</sup> <sub>R-free</sub>	-	6	100	51	2	24	18	20	7	5	18	9	4	-	3	3	-	3	5	18	16	18	15
A→P <sub>R-work</sub>	12	1	74	100	0	34	35	26	19	3	22	18	2	0	18	2	0	16	3	29	23	26	25
A→P <sub>R-free</sub>	-	4	51	100	3	26	23	20	4	3	19	9	3	-	4	3	-	5	4	20	20	18	17
B <sub>R-work</sub>	1	25	1	0	100	1	1	3	0	28	4	3	24	0	1	30	0	3	28	3	3	2	3
B <sub>R-free</sub>	-	21	2	3	100	3	3	4	3	18	3	8	26	-	8	26	-	6	19	3	4	2	2
B→P <sup>s</sup> <sub>R-work</sub>	11	1	34	34	1	100	69	24	16	1	24	15	3	0	17	2	0	16	1	22	32	26	24
B→P <sup>s</sup> <sub>R-free</sub>	-	3	24	26	3	100	43	22	5	2	23	4	3	-	5	2	-	7	3	24	28	27	18
B→P <sub>R-work</sub>	13	3	30	35	1	69	100	24	18	1	24	18	3	0	17	3	0	14	1	30	33	31	28
B→P <sub>R-free</sub>	-	4	18	23	3	43	100	29	5	3	20	5	5	-	4	2	-	7	2	22	27	27	22
P <sup>s</sup> <sub>R-work</sub>	20	4	24	26	3	24	24	100	21	3	51	24	3	0	20	5	0	24	3	55	53	57	56
P <sup>s</sup> <sub>R-free</sub>	-	3	20	20	4	22	29	100	10	5	38	18	3	-	9	3	-	10	3	45	44	43	44
P <sup>s</sup> →A <sub>R-work</sub>	36	4	15	19	0	16	18	21	100	4	20	54	2	0	40	3	0	45	1	20	17	26	23
P <sup>s</sup> →A <sub>R-free</sub>	-	10	7	4	3	5	5	10	100	11	9	37	9	-	26	8	-	30	8	10	20	11	17
P <sup>s</sup> →B <sub>R-work</sub>	4	40	3	3	28	1	1	3	4	100	6	3	45	0	3	36	0	2	34	2	4	4	3
P <sup>s</sup> →B <sub>R-free</sub>	-	33	5	3	18	2	3	5	11	100	5	10	32	-	5	34	-	5	27	4	6	5	5
P <sub>R-work</sub>	20	3	24	22	4	24	24	51	20	6	100	23	5	0	16	3	0	22	3	53	55	50	53
P <sub>R-free</sub>	-	4	18	19	3	23	20	38	9	5	100	14	4	-	7	3	-	9	3	36	40	39	38
P→A <sub>R-work</sub>	35	5	16	18	3	15	18	24	54	3	23	100	4	0	38	3	0	44	1	24	20	26	20
P→A <sub>R-free</sub>	-	12	9	9	8	4	5	18	37	10	14	100	9	-	34	9	-	30	7	15	14	13	12
P→B <sub>R-work</sub>	7	34	2	2	24	3	3	3	2	45	5	4	100	0	5	34	0	6	38	3	5	5	3
P→B <sub>R-free</sub>	-	31	4	3	26	3	5	3	9	32	4	9	100	-	5	30	-	7	28	3	3	0	3
S <sub>R-work</sub>	0	1	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	39	0	0	0	0	0
S <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S→A <sub>R-work</sub>	36	1	16	18	1	17	17	20	40	3	16	38	5	0	100	5	0	49	3	24	24	24	28
S→A <sub>R-free</sub>	-	9	3	4	8	5	4	9	26	5	7	34	5	-	100	7	-	39	9	5	11	8	7
S→B <sub>R-work</sub>	4	32	2	2	30	2	3	5	3	36	3	3	34	0	5	100	0	5	41	3	6	3	4
S→B <sub>R-free</sub>	-	30	3	3	26	2	2	3	8	34	3	9	30	-	7	100	-	3	31	3	3	5	2
S <sup>*</sup> <sub>R-work</sub>	0	1	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	100	0	0	0	0	0
S <sup>*</sup> <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S <sup>*</sup> →A <sub>R-work</sub>	35	4	16	16	3	16	14	24	45	2	22	44	6	0	49	5	0	100	3	23	22	24	22
S <sup>*</sup> →A <sub>R-free</sub>	-	6	3	5	6	7	7	10	30	5	9	30	7	-	39	3	-	100	7	9	11	10	12
S <sup>*</sup> →B <sub>R-work</sub>	3	32	3	3	28	1	1	3	1	34	3	1	38	0	3	41	0	3	100	3	5	3	4
S <sup>*</sup> →B <sub>R-free</sub>	-	30	5	4	19	3	2	3	8	27	3	7	28	-	9	31	-	7	100	1	3	3	3
S <sup>*</sup> →P <sup>s</sup> <sub>R-work</sub>	18	1	28	29	3	22	30	55	20	2	53	24	3	0	24	3	0	23	3	100	65	60	62
S <sup>*</sup> →P <sup>s</sup> <sub>R-free</sub>	-	5	18	20	3	24	22	45	10	4	36	15	3	-	5	3	-	9	1	100	45	50	46
S <sup>*</sup> →P <sub>R-work</sub>	22	3	24	23	3	32	33	53	17	4	55	20	5	0	24	6	0	22	5	65	100	56	62
S <sup>*</sup> →P <sub>R-free</sub>	-	3	16	20	4	28	27	44	20	6	40	14	3	-	11	3	-	11	3	45	100	44	51
S→P <sup>s</sup> <sub>R-work</sub>	23	3	24	26	2	26	31	57	26	4	50	26	5	0	24	3	0	24	3	60	56	100	64
S→P <sup>s</sup> <sub>R-free</sub>	-	3	18	18	2	27	27	43	11	5	39	13	0	-	8	5	-	10	3	50	44	100	40
S→P <sub>R-work</sub>	22	3	24	25	3	24	28	56	23	3	53	20	3	0	28	4	0	22	4	62	62	64	100
S→P <sub>R-free</sub>	-	3	15	17	2	18	22	44	17	5	38	12	3	-	7	2	-	12	3	46	51	40	100





APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.74: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P <sup>s</sup>	A→P	B	B→P <sup>s</sup>	B→P	P <sup>s</sup>	P <sup>s</sup> →A	P <sup>s</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>*</sup>	S <sup>*</sup> →A	S <sup>*</sup> →B	S <sup>*</sup> →P <sup>s</sup>	S <sup>*</sup> →P	S→P <sup>s</sup>	S→P
A <sub>R-work</sub>	0	51	18	18	42	24	22	29	16	59	30	15	53	0	22	45	0	23	51	30	28	27	30
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	6	0	2	3	48	5	3	3	2	24	3	3	28	3	5	33	3	3	39	5	3	3	3
A→B <sub>R-free</sub>	-	0	3	4	47	5	4	2	9	29	4	13	26	-	13	32	-	11	39	5	3	3	3
A→P <sup>s</sup> <sub>R-work</sub>	58	38	0	12	32	29	30	53	48	44	59	53	39	0	54	36	0	57	35	53	55	53	54
A→P <sup>s</sup> <sub>R-free</sub>	-	37	0	26	30	29	31	49	65	34	51	66	39	-	61	33	-	64	34	48	50	50	51
A→P <sub>R-work</sub>	57	36	14	0	32	30	29	55	47	44	61	53	39	0	53	37	0	56	38	55	57	54	55
A→P <sub>R-free</sub>	-	36	22	0	32	31	32	51	70	41	49	66	42	-	59	32	-	64	36	49	51	52	52
B <sub>R-work</sub>	5	11	1	1	0	2	2	3	1	7	2	1	14	2	1	20	3	1	23	3	1	2	1
B <sub>R-free</sub>	-	15	3	3	0	3	3	3	9	14	4	11	9	-	9	21	-	9	26	3	1	3	2
B→P <sup>s</sup> <sub>R-work</sub>	56	41	28	29	35	0	14	49	44	52	50	48	41	0	49	37	0	53	39	53	46	49	49
B→P <sup>s</sup> <sub>R-free</sub>	-	41	31	27	30	0	26	47	66	51	47	66	42	-	57	36	-	57	36	48	44	45	48
B→P <sub>R-work</sub>	55	42	32	28	32	17	0	51	43	52	54	47	43	0	49	36	0	54	38	49	47	48	50
B→P <sub>R-free</sub>	-	39	35	28	30	30	0	43	63	52	50	61	42	-	52	36	-	53	37	49	45	47	49
P <sup>s</sup> <sub>R-work</sub>	42	54	16	13	41	22	19	0	28	59	25	29	53	0	34	49	0	34	49	19	23	18	18
P <sup>s</sup> <sub>R-free</sub>	-	53	19	18	36	25	20	0	61	58	29	58	51	-	56	48	-	55	49	25	24	23	23
P <sup>s</sup> →A <sub>R-work</sub>	42	47	24	23	39	33	31	43	0	53	43	20	49	0	30	48	0	31	46	40	45	39	42
P <sup>s</sup> →A <sub>R-free</sub>	-	53	10	9	49	15	16	18	0	57	16	32	57	-	32	53	-	35	55	15	11	11	12
P <sup>s</sup> →B <sub>R-work</sub>	7	17	6	5	43	5	4	4	1	0	4	3	28	1	9	28	1	7	34	6	3	3	3
P <sup>s</sup> →B <sub>R-free</sub>	-	21	9	7	46	4	4	3	14	0	5	16	29	-	11	26	-	10	35	6	3	3	2
P <sub>R-work</sub>	39	55	11	13	39	18	16	21	28	55	0	29	53	0	36	49	1	35	48	14	16	16	15
P <sub>R-free</sub>	-	50	22	20	35	23	22	30	64	55	0	63	51	-	56	48	-	57	48	24	25	25	23
P→A <sub>R-work</sub>	43	47	18	18	38	28	26	39	19	56	41	0	52	0	29	48	0	30	49	34	41	37	41
P→A <sub>R-free</sub>	-	49	7	5	45	12	12	27	51	12	0	55	-	-	24	50	-	32	54	11	11	10	14
P→B <sub>R-work</sub>	5	22	5	5	39	5	3	2	2	19	1	2	0	2	8	26	2	4	30	5	3	2	3
P→B <sub>R-free</sub>	-	27	7	5	41	5	5	3	9	30	5	16	0	-	13	32	-	9	37	7	4	5	3
S <sub>R-work</sub>	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	19	0	0	0	0	0
S <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S→A <sub>R-work</sub>	27	53	16	15	47	22	22	36	14	55	39	17	50	0	0	47	0	20	53	33	32	30	30
S→A <sub>R-free</sub>	-	47	4	3	42	9	11	9	22	49	11	20	50	-	0	43	-	26	47	9	7	8	9
S→B <sub>R-work</sub>	6	15	4	3	38	5	4	1	1	14	3	2	22	2	2	0	3	1	32	3	2	3	3
S→B <sub>R-free</sub>	-	19	7	5	41	6	6	4	9	20	5	15	18	-	15	0	-	14	35	5	5	3	3
S <sup>*</sup> <sub>R-work</sub>	0	1	0	0	1	0	1	1	0	0	1	0	0	0	41	0	0	0	0	0	0	0	0
S <sup>*</sup> <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S <sup>*</sup> →A <sub>R-work</sub>	31	50	18	18	43	23	23	34	14	59	35	13	52	0	28	47	0	0	48	32	36	32	35
S <sup>*</sup> →A <sub>R-free</sub>	-	52	7	5	44	10	9	11	19	51	11	21	49	-	30	51	-	0	51	11	8	9	9
S <sup>*</sup> →B <sub>R-work</sub>	4	15	1	1	34	4	3	2	2	15	3	3	14	1	1	18	1	2	0	4	1	3	1
S <sup>*</sup> →B <sub>R-free</sub>	-	16	3	3	41	5	5	2	9	22	3	10	18	-	13	25	-	12	0	5	3	3	2
S <sup>*</sup> →P <sup>s</sup> <sub>R-work</sub>	43	51	13	11	39	21	15	22	30	56	27	31	47	0	33	48	0	35	44	0	18	18	18
S <sup>*</sup> →P <sup>s</sup> <sub>R-free</sub>	-	44	22	18	38	22	22	27	64	55	34	61	45	-	61	45	-	57	44	0	26	22	26
S <sup>*</sup> →P <sub>R-work</sub>	43	53	15	14	40	18	15	22	31	57	25	32	47	0	33	46	0	35	45	16	0	17	16
S <sup>*</sup> →P <sub>R-free</sub>	-	45	22	16	35	22	22	29	58	55	32	63	47	-	55	43	-	55	41	28	0	26	25
S→P <sup>s</sup> <sub>R-work</sub>	44	53	16	16	40	21	17	22	30	59	31	30	50	0	35	48	0	36	47	20	26	0	19
S→P <sup>s</sup> <sub>R-free</sub>	-	51	18	16	37	24	20	32	66	55	33	62	51	-	58	45	-	57	46	27	29	0	32
S→P <sub>R-work</sub>	42	53	14	14	43	23	18	23	29	58	28	32	51	0	32	48	0	36	47	19	22	16	0
S→P <sub>R-free</sub>	-	47	20	17	37	30	23	30	59	57	35	64	49	-	58	45	-	55	44	26	24	27	0



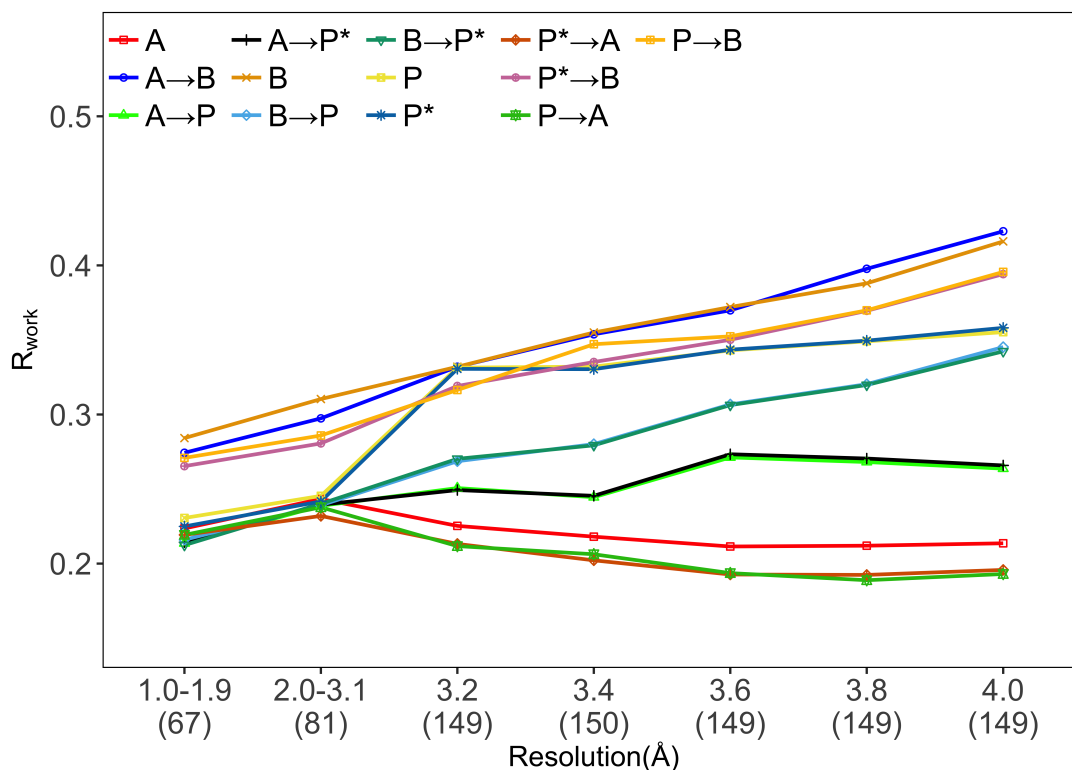
APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

Table B.75: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the original NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→P <sup>a</sup>	A→P	A→B	B	B→P <sup>a</sup>	B→P	P <sup>a</sup>	P <sup>a</sup> →A	P <sup>a</sup> →B	P	P→A	P→B	S	S→A	S→B	S <sup>a</sup>	S <sup>a</sup> →A	S <sup>a</sup> →B	S <sup>a</sup> →P <sup>a</sup>	S <sup>a</sup> →P	S→P <sup>a</sup>	S→P
A <sub>R-work</sub>	0	7	6	39	52	3	5	5	1	28	7	3	34	100	10	45	100	7	43	4	3	2	2
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A→P <sup>a</sup> <sub>R-work</sub>	5	0	0	57	67	3	4	3	3	47	3	3	53	100	9	57	100	5	59	2	2	1	2
A→P <sup>a</sup> <sub>R-free</sub>	-	0	0	51	63	4	5	1	10	45	2	10	45	-	27	53	-	23	53	2	2	1	3
A→P <sub>R-work</sub>	6	1	0	59	67	3	4	3	3	47	2	3	54	100	9	57	100	7	57	2	2	1	1
A→P <sub>R-free</sub>	-	1	0	53	60	4	5	1	8	43	3	11	45	-	29	54	-	22	53	2	2	1	2
A→B <sub>R-work</sub>	0	0	0	0	10	0	0	0	0	5	1	0	4	95	0	10	94	1	7	0	1	0	1
A→B <sub>R-free</sub>	-	3	3	0	11	0	0	0	2	3	1	1	4	-	17	9	-	14	7	0	1	0	1
B <sub>R-work</sub>	0	0	0	5	0	0	0	0	0	4	1	0	3	95	0	3	94	1	3	0	1	0	1
B <sub>R-free</sub>	-	2	1	5	0	0	0	0	1	5	1	1	5	-	18	3	-	14	3	0	1	0	1
B→P <sup>a</sup> <sub>R-work</sub>	5	5	3	53	62	0	1	3	2	42	5	4	52	100	9	55	100	5	56	2	2	1	1
B→P <sup>a</sup> <sub>R-free</sub>	-	11	11	52	64	0	1	3	13	43	3	16	49	-	28	56	-	24	57	3	1	1	1
B→P <sub>R-work</sub>	6	4	3	52	66	0	0	3	3	43	3	4	51	99	9	56	99	5	57	2	3	1	1
B→P <sub>R-free</sub>	-	11	12	53	65	0	0	4	14	41	4	18	49	-	30	55	-	27	55	3	2	1	3
P <sup>a</sup> <sub>R-work</sub>	5	4	3	39	54	3	3	0	1	34	2	3	42	99	9	44	99	6	45	1	1	1	1
P <sup>a</sup> <sub>R-free</sub>	-	11	10	41	57	3	4	0	9	34	2	11	43	-	26	45	-	23	46	1	1	1	1
P <sup>a</sup> →A <sub>R-work</sub>	6	10	8	47	59	5	5	6	0	42	8	5	47	100	14	48	100	9	51	7	5	5	5
P <sup>a</sup> →A <sub>R-free</sub>	-	8	9	26	37	1	3	1	0	18	3	3	20	-	21	27	-	16	26	1	1	1	1
P <sup>a</sup> →B <sub>R-work</sub>	1	1	1	14	18	0	0	0	0	0	1	1	4	99	3	16	99	2	14	1	1	0	1
P <sup>a</sup> →B <sub>R-free</sub>	-	6	6	14	18	0	0	0	1	0	1	3	5	-	19	16	-	16	14	0	1	0	1
P <sub>R-work</sub>	4	3	1	38	55	3	3	1	1	34	0	1	40	99	7	43	98	4	45	3	1	1	1
P <sub>R-free</sub>	-	8	9	41	57	4	4	1	8	33	0	9	40	-	24	42	-	21	44	3	1	1	1
P→A <sub>R-work</sub>	4	9	7	45	58	5	6	5	2	36	6	0	42	100	13	47	100	9	47	7	5	3	3
P→A <sub>R-free</sub>	-	8	9	25	34	3	3	1	1	21	2	0	17	-	20	24	-	16	26	2	1	1	1
P→B <sub>R-work</sub>	0	0	0	12	20	0	0	0	0	4	1	0	0	97	1	14	97	1	12	0	0	0	0
P→B <sub>R-free</sub>	-	5	5	11	18	0	0	0	4	3	1	3	0	-	21	14	-	18	11	0	0	0	0
S <sub>R-work</sub>	0	0	0	1	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
S <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S→A <sub>R-work</sub>	5	5	4	41	51	3	3	1	3	30	2	3	35	100	0	46	100	1	43	1	1	1	1
S→A <sub>R-free</sub>	-	4	5	14	22	1	2	1	0	16	1	2	11	-	0	17	-	0	14	1	1	1	1
S→B <sub>R-work</sub>	0	1	1	9	9	0	1	1	0	5	1	0	5	98	0	0	97	1	3	0	0	0	0
S→B <sub>R-free</sub>	-	4	5	9	9	0	1	1	3	5	1	3	7	-	19	0	-	15	3	0	0	0	1
S <sup>a</sup> <sub>R-work</sub>	0	0	0	1	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0
S <sup>a</sup> <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S <sup>a</sup> →A <sub>R-work</sub>	4	4	3	42	53	3	3	2	1	30	3	4	36	100	3	45	100	0	47	3	1	1	1
S <sup>a</sup> →A <sub>R-free</sub>	-	4	5	17	27	2	3	1	0	18	1	2	17	-	6	18	-	0	18	1	1	1	1
S <sup>a</sup> →B <sub>R-work</sub>	0	1	1	8	11	0	1	1	0	3	1	0	5	99	0	7	99	0	0	0	1	0	1
S <sup>a</sup> →B <sub>R-free</sub>	-	5	5	8	11	0	1	1	1	3	1	3	6	-	17	6	-	12	0	0	1	0	1
S <sup>a</sup> →P <sup>a</sup> <sub>R-work</sub>	5	5	3	42	55	3	3	2	3	35	3	3	45	100	9	46	100	7	49	0	1	0	1
S <sup>a</sup> →P <sup>a</sup> <sub>R-free</sub>	-	11	11	46	55	3	5	2	9	35	3	11	45	-	24	47	-	22	49	0	1	0	1
S <sup>a</sup> →P <sub>R-work</sub>	5	5	3	41	55	2	3	2	1	34	3	3	46	100	9	46	100	6	49	1	0	0	0
S <sup>a</sup> →P <sub>R-free</sub>	-	11	11	47	59	4	5	2	11	36	3	11	46	-	26	49	-	24	51	1	0	0	0
S→P <sup>a</sup> <sub>R-work</sub>	4	5	4	41	56	3	3	2	1	33	3	3	43	100	9	46	100	6	48	2	1	0	1
S→P <sup>a</sup> <sub>R-free</sub>	-	12	12	43	57	3	5	1	10	36	3	14	44	-	25	47	-	23	48	1	1	0	1
S→P <sub>R-work</sub>	4	6	5	41	53	3	3	2	1	35	3	3	43	100	9	45	100	6	47	1	0	0	0
S→P <sub>R-free</sub>	-	11	12	46	58	3	4	2	11	34	3	10	44	-	26	49	-	23	51	1	1	0	0



Figure B.1: Mean protein model R-work for the NO-NCS data sets partitioned into classes based on their resolution. The number of data sets in each class is indicated in brackets under the graph.



## B.4 Experimental results for the synthetic data sets without the Buccaneer development data sets

Table B.76: Complete and intermediate models produced by the 23 pipeline variants for the synthetic data sets, where ‘(T)’ and ‘(C)’ denote intermediate models produced by pipeline executions that timed out and crashed, respectively.

Pipeline variant	HA-NCS			MR-NCS			NO-NCS		
	Complete	Intermediate	Failed	Complete	Intermediate	Failed	Complete	Intermediate	Failed
A	1008	1(T) 0(C)	0	1007	2(T) 0(C)	0	1008	1(T) 0(C)	0
A→P*	1006	2(T) 0(C)	1	1006	2(T) 0(C)	1	1007	2(T) 0(C)	0
A→B	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0
B	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0
B→P*	1003	1(T) 0(C)	5	1004	0(T) 0(C)	5	1005	0(T) 0(C)	4
P*	1002	7(T) 0(C)	0	1004	5(T) 0(C)	0	1001	8(T) 0(C)	0
P*→A	1008	1(T) 0(C)	0	1009	0(T) 0(C)	0	1008	1(T) 0(C)	0
P*→B	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0	1009	0(T) 0(C)	0
A→P	-	-	-	-	-	-	1009	0(T) 0(C)	0
B→P	-	-	-	-	-	-	1003	2(T) 0(C)	4
P	-	-	-	-	-	-	1001	7(T) 0(C)	1
P→A	-	-	-	-	-	-	1002	6(T) 0(C)	1
P→B	-	-	-	-	-	-	1008	0(T) 0(C)	1

Models used in the comparison: 744 HA-NCS, 745 MR-NCS and 746 NO-NCS.

Table B.77: Structure completeness comparison for the models generated from the synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	1	1	1	0	2	28	1
A→B	93	0	80	41	29	72	93	25
A→P*	95	15	0	12	3	23	96	6
B	93	50	83	0	25	75	94	30
B→P*	98	66	95	69	0	86	99	45
P*	97	26	71	23	10	0	97	13
P*→A	15	1	1	1	0	2	0	0
P*→B	96	70	92	63	50	84	97	0



Table B.78: Structure completeness comparison for the models generated from the synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	100	6	4	5	1	1	57	3
A→B	6	100	5	9	5	3	6	5
A→P*	4	5	100	5	2	6	3	2
B	5	9	5	100	6	2	5	7
B→P*	1	5	2	6	100	4	1	5
P*	1	3	6	2	4	100	1	3
P*→A	57	6	3	5	1	1	100	3
P*→B	3	5	2	7	5	3	3	100



Table B.79: Structure completeness comparison for the models generated from the synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	0	0	1	0	2	7	0
A→B	82	0	73	28	22	68	83	17
A→P*	77	6	0	5	1	14	78	2
B	84	36	77	0	14	69	85	20
B→P*	94	52	88	48	0	79	95	32
P*	92	18	47	16	4	0	93	7
P*→A	2	0	0	0	0	1	0	0
P*→B	92	55	87	50	39	79	92	0



Table B.80: Structure completeness comparison for the models generated from the synthetic HA-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	1	1	1	0	0	21	0
A→B	11	0	7	13	8	3	10	9
A→P*	19	9	0	8	2	9	18	3
B	9	14	6	0	11	5	9	10
B→P*	4	13	6	21	0	8	4	13
P*	5	8	24	7	6	0	4	6
P*→A	13	1	0	1	0	0	0	0
P*→B	5	15	5	14	11	5	5	0



Table B.81: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	94	83	93	86	97	28	91
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	5	0	8	36	7	34	2	22
A→B <sub>R-free</sub>	-	0	36	42	10	36	79	29
A→P* <sub>R-work</sub>	13	89	0	89	71	99	3	85
A→P* <sub>R-free</sub>	-	59	0	56	29	55	90	49
B <sub>R-work</sub>	6	51	9	0	4	37	1	29
B <sub>R-free</sub>	-	48	38	0	5	38	80	34
B→P* <sub>R-work</sub>	11	90	23	92	0	82	4	80
B→P* <sub>R-free</sub>	-	87	67	91	0	76	97	77
P* <sub>R-work</sub>	2	62	0	59	11	0	0	49
P* <sub>R-free</sub>	-	59	40	56	18	0	94	47
P*→A <sub>R-work</sub>	60	97	95	98	93	100	0	97
P*→A <sub>R-free</sub>	-	16	7	16	2	5	0	10
P*→B <sub>R-work</sub>	8	65	12	60	14	45	2	0
P*→B <sub>R-free</sub>	-	63	46	58	19	47	87	0



Table B.82: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <i>R-work</i>	100	1	4	1	3	0	12	1
A <i>R-free</i>	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	1	100	3	13	3	5	1	13
A→B <i>R-free</i>	-	100	5	9	4	5	4	8
A→P* <i>R-work</i>	4	3	100	2	6	0	2	3
A→P* <i>R-free</i>	-	5	100	6	4	5	3	6
B <i>R-work</i>	1	13	2	100	4	4	1	11
B <i>R-free</i>	-	9	6	100	4	6	4	8
B→P* <i>R-work</i>	3	3	6	4	100	7	3	6
B→P* <i>R-free</i>	-	4	4	4	100	6	1	4
P* <i>R-work</i>	0	5	0	4	7	100	0	6
P* <i>R-free</i>	-	5	5	6	6	100	1	7
P*→A <i>R-work</i>	12	1	2	1	3	0	100	1
P*→A <i>R-free</i>	-	4	3	4	1	1	100	2
P*→B <i>R-work</i>	1	13	3	11	6	6	1	100
P*→B <i>R-free</i>	-	8	6	8	4	7	2	100



Table B.83: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <i>R-work</i>	0	85	53	85	71	93	3	82
A <i>R-free</i>	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	2	0	1	9	1	17	0	7
A→B <i>R-free</i>	-	0	17	15	2	18	63	9
A→P* <i>R-work</i>	6	75	0	72	39	85	0	67
A→P* <i>R-free</i>	-	38	0	36	16	32	78	29
B <i>R-work</i>	2	13	1	0	0	20	0	8
B <i>R-free</i>	-	17	19	0	1	19	66	12
B→P* <i>R-work</i>	4	70	4	67	0	45	0	51
B→P* <i>R-free</i>	-	62	47	63	0	47	90	49
P* <i>R-work</i>	0	41	0	38	0	0	0	27
P* <i>R-free</i>	-	37	22	34	4	0	84	24
P*→A <i>R-work</i>	19	91	69	91	81	99	0	89
P*→A <i>R-free</i>	-	6	2	7	1	2	0	3
P*→B <i>R-work</i>	2	26	2	23	2	25	0	0
P*→B <i>R-free</i>	-	28	26	26	5	25	73	0





Table B.84: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic HA-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	9	29	8	14	4	25	9
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	3	0	7	27	6	16	2	16
A→B <sub>R-free</sub>	-	0	19	28	7	18	17	20
A→P* <sub>R-work</sub>	7	14	0	17	32	14	3	19
A→P* <sub>R-free</sub>	-	21	0	20	13	23	12	19
B <sub>R-work</sub>	4	37	8	0	4	17	1	21
B <sub>R-free</sub>	-	31	19	0	4	19	14	22
B→P* <sub>R-work</sub>	7	20	19	25	0	37	3	30
B→P* <sub>R-free</sub>	-	25	20	28	0	29	6	28
P* <sub>R-work</sub>	2	21	0	20	10	0	0	21
P* <sub>R-free</sub>	-	22	18	22	15	0	10	23
P*→A <sub>R-work</sub>	40	7	27	7	13	1	0	8
P*→A <sub>R-free</sub>	-	10	5	9	1	3	0	7
P*→B <sub>R-work</sub>	6	39	10	38	12	20	2	0
P*→B <sub>R-free</sub>	-	35	20	32	14	22	14	0



Table B.85: Structure completeness comparison for the models generated from the synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	1	1	1	0	2	30	1
A→B	94	0	81	43	32	74	94	28
A→P*	96	15	0	12	3	24	96	6
B	95	48	85	0	24	76	95	33
B→P*	99	65	95	69	0	86	99	48
P*	97	24	69	22	10	0	97	13
P*→A	16	1	1	0	0	2	0	1
P*→B	97	67	91	61	48	84	97	0



Table B.86: Structure completeness comparison for the models generated from the synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	100	6	3	5	1	1	54	2
A→B	6	100	4	9	4	2	6	5
A→P*	3	4	100	4	1	7	3	2
B	5	9	4	100	7	2	4	6
B→P*	1	4	1	7	100	4	1	4
P*	1	2	7	2	4	100	1	2
P*→A	54	6	3	4	1	1	100	2
P*→B	2	5	2	6	4	2	2	100



Table B.87: Structure completeness comparison for the models generated from the synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	0	0	0	0	2	7	0
A→B	83	0	75	31	23	71	83	20
A→P*	78	7	0	5	1	14	79	3
B	86	34	78	0	13	72	86	22
B→P*	94	51	89	46	0	79	95	33
P*	92	17	46	15	4	0	93	8
P*→A	3	0	0	0	0	1	0	0
P*→B	92	52	86	48	39	81	92	0



Table B.88: Structure completeness comparison for the models generated from the synthetic MR-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A	0	1	0	0	0	0	24	0
A→B	11	0	6	12	9	4	11	9
A→P*	18	8	0	6	2	10	17	4
B	9	14	6	0	11	4	9	11
B→P*	5	13	6	23	0	8	4	15
P*	5	7	23	7	6	0	4	5
P*→A	13	1	0	0	0	0	0	1
P*→B	5	15	5	13	10	4	5	0



Table B.89: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with lower R-work or R-free than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	93	83	93	85	98	30	92
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	6	0	8	41	7	37	2	27
A→B <sub>R-free</sub>	-	0	35	47	10	39	79	34
A→P* <sub>R-work</sub>	13	88	0	88	68	99	3	83
A→P* <sub>R-free</sub>	-	58	0	56	28	55	90	49
B <sub>R-work</sub>	5	46	9	0	4	40	1	32
B <sub>R-free</sub>	-	44	38	0	6	40	81	34
B→P* <sub>R-work</sub>	11	90	25	92	0	83	4	81
B→P* <sub>R-free</sub>	-	86	68	90	0	74	96	77
P* <sub>R-work</sub>	2	57	0	56	11	0	0	49
P* <sub>R-free</sub>	-	57	40	54	19	0	95	44
P*→A <sub>R-work</sub>	59	97	94	97	94	100	0	97
P*→A <sub>R-free</sub>	-	17	7	15	3	4	0	11
P*→B <sub>R-work</sub>	7	60	13	59	14	45	1	0
P*→B <sub>R-free</sub>	-	59	47	58	17	46	86	0



Table B.90: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <i>R-work</i>	100	1	4	2	4	0	12	2
A <i>R-free</i>	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	1	100	3	13	4	6	1	13
A→B <i>R-free</i>	-	100	6	9	4	4	4	8
A→P* <i>R-work</i>	4	3	100	3	7	1	3	4
A→P* <i>R-free</i>	-	6	100	6	4	5	2	5
B <i>R-work</i>	2	13	3	100	3	4	1	10
B <i>R-free</i>	-	9	6	100	4	5	4	8
B→P* <i>R-work</i>	4	4	7	3	100	7	2	5
B→P* <i>R-free</i>	-	4	4	4	100	7	1	5
P* <i>R-work</i>	0	6	1	4	7	100	0	6
P* <i>R-free</i>	-	4	5	5	7	100	1	9
P*→A <i>R-work</i>	12	1	3	1	2	0	100	1
P*→A <i>R-free</i>	-	4	2	4	1	1	100	3
P*→B <i>R-work</i>	2	13	4	10	5	6	1	100
P*→B <i>R-free</i>	-	8	5	8	5	9	3	100



Table B.91: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	84	51	86	68	93	4	81
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	2	0	1	10	1	19	0	9
A→B <sub>R-free</sub>	-	0	18	16	3	20	65	11
A→P* <sub>R-work</sub>	5	73	0	71	39	85	0	65
A→P* <sub>R-free</sub>	-	36	0	36	15	32	81	28
B <sub>R-work</sub>	2	12	1	0	0	21	0	9
B <sub>R-free</sub>	-	18	20	0	1	21	66	11
B→P* <sub>R-work</sub>	3	66	4	64	0	48	0	51
B→P* <sub>R-free</sub>	-	59	48	60	0	47	90	46
P* <sub>R-work</sub>	0	38	0	37	1	0	0	26
P* <sub>R-free</sub>	-	32	21	33	4	0	85	22
P*→A <sub>R-work</sub>	19	90	68	91	79	99	0	88
P*→A <sub>R-free</sub>	-	7	2	6	1	1	0	3
P*→B <sub>R-work</sub>	2	24	2	21	2	26	0	0
P*→B <sub>R-free</sub>	-	28	26	26	4	26	74	0

0  99

Table B.92: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic MR-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	B	B→P*	P*	P*→A	P*→B
A <sub>R-work</sub>	0	9	32	8	17	4	26	11
A <sub>R-free</sub>	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	3	0	7	31	6	18	2	18
A→B <sub>R-free</sub>	-	0	17	31	8	19	14	23
A→P* <sub>R-work</sub>	8	15	0	17	29	14	3	18
A→P* <sub>R-free</sub>	-	22	0	20	12	23	10	21
B <sub>R-work</sub>	4	33	8	0	4	18	1	23
B <sub>R-free</sub>	-	26	18	0	5	19	15	23
B→P* <sub>R-work</sub>	8	24	21	29	0	34	4	30
B→P* <sub>R-free</sub>	-	27	21	30	0	27	6	32
P* <sub>R-work</sub>	2	19	0	19	10	0	0	23
P* <sub>R-free</sub>	-	25	18	21	15	0	10	22
P*→A <sub>R-work</sub>	40	7	27	6	15	1	0	10
P*→A <sub>R-free</sub>	-	10	5	9	2	3	0	8
P*→B <sub>R-work</sub>	4	36	11	37	11	19	1	0
P*→B <sub>R-free</sub>	-	31	21	32	13	21	13	0



Table B.93: Structure completeness comparison for the models generated from the synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A	0	1	1	1	1	0	0	2	27	0	2	25	0
A→B	92	0	74	71	40	27	24	65	92	24	65	93	25
A→P*	95	20	0	40	16	3	3	22	96	6	22	96	7
A→P	96	25	50	0	20	4	4	20	96	8	21	96	8
B	94	50	79	76	0	23	20	68	94	27	69	95	29
B→P*	99	70	95	94	72	0	34	83	99	45	84	99	48
B→P	99	71	96	94	73	49	0	85	99	45	86	99	50
P*	97	32	72	73	29	12	11	0	97	16	43	98	16
P*→A	16	1	1	0	1	0	0	2	0	0	2	19	1
P*→B	97	71	91	88	68	51	50	81	97	0	81	98	50
P	97	33	74	74	28	11	10	46	97	16	0	97	17
P→A	14	0	1	1	1	0	0	1	18	0	1	0	0
P→B	97	69	89	88	65	47	46	81	97	45	81	97	0



Table B.94: Structure completeness comparison for the models generated from the synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage(rounded to the nearest integer) of models that the pipeline variant built with equal structure completeness to each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A	100	7	4	3	5	1	1	1	57	3	1	61	3
A→B	7	100	6	4	10	4	4	3	6	5	2	7	6
A→P*	4	6	100	10	5	2	2	6	4	3	4	3	4
A→P	3	4	10	100	4	2	2	7	3	4	5	3	4
B	5	10	5	4	100	6	6	3	5	6	3	5	6
B→P*	1	4	2	2	6	100	17	5	1	4	5	1	5
B→P	1	4	2	2	6	17	100	4	1	4	4	1	4
P*	1	3	6	7	3	5	4	100	1	3	12	1	3
P*→A	57	6	4	3	5	1	1	1	100	2	1	63	2
P*→B	3	5	3	4	6	4	4	3	2	100	3	2	5
P	1	2	4	5	3	5	4	12	1	3	100	1	2
P→A	61	7	3	3	5	1	1	1	63	2	1	100	3
P→B	3	6	4	4	6	5	4	3	2	5	2	3	100





Table B.95: Structure completeness comparison for the models generated from the synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with at least 5% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A	0	0	1	0	1	0	0	2	6	0	2	6	0
A→B	79	0	67	65	27	18	17	61	79	16	61	80	18
A→P*	75	9	0	21	7	1	1	13	76	2	11	76	2
A→P	84	13	26	0	10	1	1	11	84	3	11	84	4
B	82	33	71	70	0	14	11	63	82	18	63	82	19
B→P*	94	55	88	85	53	0	11	74	95	31	75	94	34
B→P	94	59	88	86	55	16	0	76	94	33	75	94	36
P*	92	24	50	47	21	4	3	0	93	9	15	93	9
P*→A	2	0	0	0	0	0	0	1	0	0	2	3	0
P*→B	90	56	84	82	54	40	38	77	91	0	76	91	37
P	92	24	49	49	21	4	3	16	92	9	0	93	9
P→A	2	0	0	1	0	0	0	1	2	0	1	0	0
P→B	90	55	83	81	49	36	35	76	90	33	75	90	0



Table B.96: Structure completeness comparison for the models generated from the synthetic NO-NCS data sets. Each row corresponds to a pipeline variant, and shows the percentage (rounded to the nearest integer) of models that the pipeline variant built with between 1% and 4% higher structure completeness than each of the other pipeline variants.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A	0	1	0	0	1	0	0	0	21	0	0	18	0
A→B	13	0	8	5	13	9	8	3	13	8	5	13	7
A→P*	20	11	0	19	9	2	2	9	20	4	11	20	4
A→P	12	12	24	0	9	3	3	9	12	5	10	12	5
B	12	17	8	7	0	8	9	5	12	8	6	13	10
B→P*	5	14	7	9	18	0	23	9	4	14	9	4	14
B→P	5	13	8	8	18	33	0	9	5	13	11	5	14
P*	5	8	22	26	8	8	8	0	5	8	27	4	7
P*→A	14	1	0	0	1	0	0	0	0	0	0	16	1
P*→B	6	15	7	6	14	11	13	4	7	0	5	7	13
P	5	9	25	25	8	7	7	29	5	7	0	5	9
P→A	12	0	1	0	1	0	0	0	16	0	0	0	0
P→B	7	14	6	7	16	11	11	5	7	12	5	7	0



Table B.97: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A <i>R-work</i>	0	94	82	82	95	87	87	97	29	92	97	28	93
A <i>R-free</i>	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <i>R-work</i>	5	0	7	7	40	6	6	30	1	22	29	2	26
A→B <i>R-free</i>	-	0	31	30	46	8	7	32	75	31	31	78	31
A→P* <i>R-work</i>	13	89	0	34	91	73	74	99	3	86	99	4	87
A→P* <i>R-free</i>	-	63	0	42	62	30	29	54	89	53	52	92	55
A→P <i>R-work</i>	14	91	39	0	91	74	75	99	5	87	99	5	87
A→P <i>R-free</i>	-	64	44	0	63	31	30	55	91	55	53	92	55
B <i>R-work</i>	4	46	7	7	0	3	3	32	1	26	31	2	29
B <i>R-free</i>	-	43	30	31	0	4	4	34	75	32	33	77	31
B→P* <i>R-work</i>	10	92	21	21	93	0	34	79	3	81	79	4	82
B→P* <i>R-free</i>	-	88	66	64	92	0	41	73	95	80	73	97	82
B→P <i>R-work</i>	10	91	21	20	94	32	0	77	4	83	77	4	82
B→P <i>R-free</i>	-	90	66	65	92	42	0	71	96	80	73	97	82
P* <i>R-work</i>	3	66	0	1	64	12	14	0	0	54	33	0	56
P* <i>R-free</i>	-	65	41	41	62	21	21	0	94	53	41	95	52
P*→A <i>R-work</i>	59	98	94	93	98	95	95	100	0	97	100	38	98
P*→A <i>R-free</i>	-	21	7	6	21	3	3	5	0	14	6	47	14
P*→B <i>R-work</i>	6	66	11	9	65	13	13	40	2	0	39	2	48
P*→B <i>R-free</i>	-	62	42	40	62	16	15	40	83	0	40	84	49
P <i>R-work</i>	2	67	0	0	64	13	14	36	0	56	0	0	55
P <i>R-free</i>	-	65	41	41	63	21	21	43	93	53	0	94	53
P→A <i>R-work</i>	59	98	94	94	98	95	95	100	38	98	100	0	97
P→A <i>R-free</i>	-	18	6	6	19	2	2	4	41	12	5	0	13
P→B <i>R-work</i>	5	64	10	10	60	14	12	40	1	41	40	1	0
P→B <i>R-free</i>	-	61	40	39	61	14	13	41	82	44	40	83	0



Table B.98: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with equal R-work or R-free to each other pipeline variant.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A <sub>R-work</sub>	100	1	5	4	1	3	3	0	12	1	1	13	2
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	1	100	3	2	14	3	3	4	1	12	4	1	10
A→B <sub>R-free</sub>	-	100	6	6	11	3	3	3	5	7	4	4	8
A→P* <sub>R-work</sub>	5	3	100	27	2	5	6	0	2	3	0	2	3
A→P* <sub>R-free</sub>	-	6	100	14	8	4	5	5	3	5	7	2	5
A→P <sub>R-work</sub>	4	2	27	100	2	5	5	0	2	4	0	1	3
A→P <sub>R-free</sub>	-	6	14	100	6	5	5	4	3	5	6	1	5
B <sub>R-work</sub>	1	14	2	2	100	3	3	4	1	9	5	1	11
B <sub>R-free</sub>	-	11	8	6	100	4	3	4	4	7	4	4	8
B→P* <sub>R-work</sub>	3	3	5	5	3	100	34	9	1	5	8	1	5
B→P* <sub>R-free</sub>	-	3	4	5	4	100	17	7	2	4	6	1	5
B→P <sub>R-work</sub>	3	3	6	5	3	34	100	9	1	4	9	1	6
B→P <sub>R-free</sub>	-	3	5	5	3	17	100	9	1	5	7	0	5
P* <sub>R-work</sub>	0	4	0	0	4	9	9	100	0	5	31	0	4
P* <sub>R-free</sub>	-	3	5	4	4	7	9	100	1	7	16	1	8
P*→A <sub>R-work</sub>	12	1	2	2	1	1	1	0	100	1	0	24	1
P*→A <sub>R-free</sub>	-	5	3	3	4	2	1	1	100	2	2	12	4
P*→B <sub>R-work</sub>	1	12	3	4	9	5	4	5	1	100	5	1	11
P*→B <sub>R-free</sub>	-	7	5	5	7	4	5	7	2	100	7	3	7
P <sub>R-work</sub>	1	4	0	0	5	8	9	31	0	5	100	0	5
P <sub>R-free</sub>	-	4	7	6	4	6	7	16	2	7	100	1	6
P→A <sub>R-work</sub>	13	1	2	1	1	1	1	0	24	1	0	100	1
P→A <sub>R-free</sub>	-	4	2	1	4	1	0	1	12	3	1	100	4
P→B <sub>R-work</sub>	2	10	3	3	11	5	6	4	1	11	5	1	100
P→B <sub>R-free</sub>	-	8	5	5	8	5	5	8	4	7	6	4	100



Table B.99: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free between 1% and 4% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A <sub>R-work</sub>	0	8	28	29	7	14	15	4	26	9	5	24	9
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	3	0	6	6	31	5	5	14	1	16	13	1	18
A→B <sub>R-free</sub>	-	0	17	15	32	6	5	16	16	22	16	21	20
A→P* <sub>R-work</sub>	8	12	0	34	14	29	29	15	3	17	16	4	16
A→P* <sub>R-free</sub>	-	21	0	35	21	14	13	22	12	19	19	14	18
A→P <sub>R-work</sub>	9	14	38	0	16	27	29	11	4	17	13	4	16
A→P <sub>R-free</sub>	-	21	34	0	21	13	14	23	12	18	20	12	18
B <sub>R-work</sub>	3	34	6	6	0	3	3	16	1	19	16	1	22
B <sub>R-free</sub>	-	26	15	17	0	3	3	18	16	22	18	17	20
B→P* <sub>R-work</sub>	7	20	18	18	22	0	33	39	3	27	40	3	24
B→P* <sub>R-free</sub>	-	22	21	20	23	0	36	31	7	30	31	8	30
B→P <sub>R-work</sub>	6	19	17	17	24	32	0	36	3	28	37	3	25
B→P <sub>R-free</sub>	-	22	21	21	25	36	0	27	7	29	30	7	29
P* <sub>R-work</sub>	2	18	0	1	17	11	13	0	0	22	32	0	20
P* <sub>R-free</sub>	-	23	17	19	19	16	15	0	10	24	34	10	19
P*→A <sub>R-work</sub>	40	7	26	26	6	12	13	1	0	7	1	33	7
P*→A <sub>R-free</sub>	-	12	5	5	13	2	2	3	0	10	4	35	9
P*→B <sub>R-work</sub>	4	41	8	8	41	10	10	18	2	0	17	1	34
P*→B <sub>R-free</sub>	-	33	19	18	33	11	12	18	14	0	18	14	31
P <sub>R-work</sub>	1	18	0	0	17	12	13	35	0	23	0	0	20
P <sub>R-free</sub>	-	22	19	20	20	16	16	36	9	24	0	10	22
P→A <sub>R-work</sub>	38	7	27	26	5	13	14	1	34	8	1	0	7
P→A <sub>R-free</sub>	-	11	3	5	13	1	2	3	32	8	4	0	8
P→B <sub>R-work</sub>	2	38	8	8	36	11	10	20	1	29	21	1	0
P→B <sub>R-free</sub>	-	33	18	18	34	11	10	22	15	29	21	15	0



Table B.100: Comparison of R-work/R-free (rounded to two decimal places) for the models generated from the synthetic NO-NCS data sets. Each row shows the percentage of models that a pipeline variant built with R-work or R-free at least 5% lower than each other pipeline variant.

Pipeline variant	A	A→B	A→P*	A→P	B	B→P*	B→P	P*	P*→A	P*→B	P	P→A	P→B
A <sub>R-work</sub>	0	86	54	53	88	73	72	93	4	84	92	4	84
A <sub>R-free</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-
A→B <sub>R-work</sub>	2	0	1	1	9	1	1	16	0	7	16	0	8
A→B <sub>R-free</sub>	-	0	14	15	15	2	2	15	59	9	16	57	11
A→P* <sub>R-work</sub>	5	77	0	0	77	44	44	85	1	69	83	1	71
A→P* <sub>R-free</sub>	-	42	0	7	41	16	16	32	77	34	33	78	37
A→P <sub>R-work</sub>	5	77	1	0	76	47	45	88	1	70	86	1	71
A→P <sub>R-free</sub>	-	43	10	0	42	18	16	33	79	36	34	80	38
B <sub>R-work</sub>	2	12	1	1	0	0	0	16	0	6	16	0	8
B <sub>R-free</sub>	-	16	15	14	0	1	1	16	59	10	15	60	11
B→P* <sub>R-work</sub>	3	71	3	3	72	0	1	40	0	54	40	1	58
B→P* <sub>R-free</sub>	-	67	44	44	68	0	5	42	88	50	41	89	52
B→P <sub>R-work</sub>	4	72	4	3	70	1	0	41	0	55	40	1	57
B→P <sub>R-free</sub>	-	68	45	43	67	5	0	43	89	51	43	90	53
P* <sub>R-work</sub>	0	48	0	0	47	1	1	0	0	32	1	0	36
P* <sub>R-free</sub>	-	42	23	22	43	4	5	0	84	29	6	84	33
P*→A <sub>R-work</sub>	19	91	68	67	92	83	82	99	0	90	99	5	90
P*→A <sub>R-free</sub>	-	9	2	1	8	1	1	2	0	5	2	12	5
P*→B <sub>R-work</sub>	2	25	2	2	24	3	3	23	0	0	22	0	14
P*→B <sub>R-free</sub>	-	29	23	22	29	5	4	22	69	0	22	70	18
P <sub>R-work</sub>	1	49	0	0	47	0	1	1	0	33	0	0	34
P <sub>R-free</sub>	-	43	22	21	43	5	5	7	84	29	0	84	31
P→A <sub>R-work</sub>	21	91	67	68	92	82	81	99	5	90	99	0	91
P→A <sub>R-free</sub>	-	7	3	2	6	1	1	1	9	4	1	0	5
P→B <sub>R-work</sub>	3	26	2	1	24	2	2	21	0	12	19	0	0
P→B <sub>R-free</sub>	-	28	22	21	27	3	3	19	67	15	19	69	0



## B.5 The command line used to run the pipelines

### B.5.1 PHENIX AutoBuild

The following command line was used to build data set ID 1O6A (resolution 1.9 Å) and the initial model from Buccaneer.

```
phenix.autobuild \
data=PDBID.mtz \
seq_file=PDBID.fasta \
input_labels='FP SIGFP PHIB FOM HLA HLB HLC HLD FreeR_flag' clean_up=True \
(The following three parameters are used when run PHENIX AutoBuild after Parrot )
input_map_file=PDBID.mtz \
map_file_is_density_modified=True \
input_map_labels='FP hltfom.Phi_fom.phi hltfom.Phi_fom.fom'\
model=Buccaneer/PDBID.pdb
```

```
autobuild {
  data = "PDBID.mtz"
  model = "Buccaneer/PDBID.pdb"
  seq_file = "PDBID.fasta"
  map_file = Auto
  refinement_file = Auto
  hires_file = Auto
  crystal_info {
    unit_cell = None
    space_group = None
    solvent_fraction = None
    chain_type = *Auto PROTEIN DNA RNA
    resolution = 0
    dmax = 500
    overall_resolution = 0
    sequence = None
  }
  input_files {
    input_labels = FP SIGFP PHIB FOM HLA HLB HLC HLD FreeR_flag
    input_hires_labels = None
    input_map_labels = FP hltfom.Phi_fom.phi hltfom.Phi_fom.fom
    input_refinement_labels = None
    input_ha_file = None
    force_input_ha = False
    include_ha_in_model = True
    cif_def_file_list = None
    input_lig_file_list = None
    keep_input_ligands = True
    keep_input_waters = False
    keep_pdb_atoms = True
  }
}
```

```

remove_residues_on_special_positions = False
refine_eff_file_list = None
map_file_is_density_modified = True
map_file_fom = None
use_constant_input_map = False
use_map_file_as_hklstart = None
use_map_in_resolve_with_model = False
identity_from_remark = True
input_data_type = None
}
aniso {
  remove_aniso = True
  b_iso = None
  max_b_iso = 40
  target_b_ratio = 10
}
decision_making {
  acceptable_r = 0.25
  r_switch = 0.4
  semi_acceptable_r = 0.3
  reject_weak = False
  min_weak_z = 0.2
  min_cc_res_rebuild = 0.4
  min_seq_identity_percent = 50
  dist_close = None
  dist_close_overlap = 1.5
  loop_cc_min = 0.4
  group_ca_length = 4
  group_length = 2
  include_molprobity = False
  ok_molp_score = None
  scale_molp_score = None
}
density_modification {
  add_classic_denmod = None
  skip_classic_if_worse_fom = True
  skip_ncs_in_add_classic = True
  thorough_denmod = *Auto True False
  hl = False
  mask_type = *histograms probability wang classic
  mask_from_pdb = None
  mask_type_extreme_dm = histograms probability *wang classic
  mask_cycles_extreme_dm = 1
  minor_cycles_extreme_dm = 4
  wang_radius_extreme_dm = 20
  precondition = False
  minimum_ncs_cc = 0.3
  extreme_dm = False
  fom_for_extreme_dm_rebuild = 0.1

```



```

fom_for_extreme_dm = 0.35
rad_mask_from_pdb = 2
modify_outside_delta_solvent = 0.05
modify_outside_model = False
truncate_ha_sites_in_resolve = *Auto True False
rad_mask = None
s_step = None
res_start = None
map_dmin_start = None
map_dmin_incr = 0.25
use_resolve_fragments = True
use_resolve_pattern = True
use_hl_anom_in_denmod = False
use_hl_anom_in_denmod_with_model = False
mask_as_mtz = False
protein_output_mask_file = None
ncs_output_mask_file = None
omit_output_mask_file = None
}
maps {
  maps_only = False
  n_xyz_list = None
}
model_building {
  build_type = *RESOLVE RESOLVE_AND_BUCCANEER
  allow_negative_residues = False
  highest_resno = None
  semet = False
  use_met_in_align = *Auto True False
  base_model = None
  consider_main_chain_list = None
  dist_connect_max_helices = None
  edit_pdb = True
  helices_strands_only = False
  resolution_helices_strands = 3.1
  helices_strands_start = False
  cc_helix_min = None
  cc_strand_min = None
  loop_lib = False
  standard_loops = True
  trace_loops = False
  refine_trace_loops = True
  density_of_points = None
  max_density_of_points = None
  cutout_model_radius = None
  max_cutout_model_radius = 20
  padding = 1
  max_span = 30
  max_overlap = None

```

```

min_overlap = None
include_input_model = True
input_compare_file = None
merge_models = False
morph = False
morph_main = False
dist_cut_base = 3
morph_cycles = 2
morph_rad = 7
n_ca_enough_helices = None
delta_phi = 20
offsets_list = 53 7 23
all_maps_in_rebuild = False
ps_in_rebuild = False
use_ncs_in_ps = False
remove_outlier_segments_z_cut = 3
refine = True
refine_final_model_vs_orig_data = True
reference_model = None
resolution_build = None
restart_cycle_after_morph = 5
retrace_before_build = False
reuse_chain_prev_cycle = True
richardson_rotamers = *Auto True False
rms_random_frag = None
rms_random_loop = None
start_chains_list = None
trace_as_lig = False
track_libs = False
two_fofc_denmod_in_rebuild = False
rebuild_from_fragments = False
two_fofc_in_rebuild = False
refine_map_coeff_labels = "2FOFCWT PH2FOFCWT"
filled_2fofc_maps = True
map_phasing = False
use_any_side = True
truncate_missing_side_chains = None
use_cc_in_combine_extend = False
sort_hetatms = False
map_to_object = None
}
multiple_models {
combine_only = False
multiple_models = False
multiple_models_first = 1
multiple_models_group_number = 5
multiple_models_last = 20
multiple_models_number = 20
multiple_models_starting = True

```

```

multiple_models_starting_resolution = 4
place_waters_in_combine = None
}
ncs {
  find_ncs = *Auto True False
  input_ncs_file = None
  ncs_copies = None
  ncs_refine_coord_sigma_from_rmsd = False
  ncs_refine_coord_sigma_from_rmsd_ratio = 1
  no_merge_ncs_copies = False
  optimize_ncs = True
  use_ncs_in_build = True
  ncs_in_refinement = *torsion cartesian None
}
omit {
  composite_omit_type = *None simple_omit refine_omit sa_omit \
                        iterative_build_omit
  n_box_target = None
  n_cycle_image_min = 3
  n_cycle_rebuild_omit = 10
  offset_boundary = 2
  omit_boundary = 2
  omit_box_start = 0
  omit_box_end = 0
  omit_box_pdb_list = None
  omit_chain_list = None
  omit_offset_list = 0 0 0 0 0 0
  omit_on_rebuild = False
  omit_selection = None
  omit_region_specification = *composite_omit omit_around_pdb \
                              omit_selection
  omit_res_start_list = None
  omit_res_end_list = None
}
rebuild_in_place {
  min_seq_identity_percent_rebuild_in_place = 95
  n_cycle_rebuild_in_place = None
  n_rebuild_in_place = 1
  rebuild_chain_list = None
  rebuild_in_place = *Auto True False
  rebuild_near_chain = None
  rebuild_near_dist = 7.5
  rebuild_near_res = None
  rebuild_res_end_list = None
  rebuild_res_start_list = None
  rebuild_side_chains = False
  redo_side_chains = True
  replace_existing = True
  delete_bad_residues_only = False
}

```

```

touch_up = False
touch_up_extra_residues = None
worst_percent_res_rebuild = 2
smooth_range = None
smooth_minimum_length = None
}
refinement {
  refine_b = True
  refine_se_occ = True
  skip_clash_guard = True
  correct_special_position_tolerance = None
  use_mhhl = True
  generate_hl_if_missing = False
  place_waters = True
  refinement_resolution = 0
  ordered_solvent_low_resolution = None
  link_distance_cutoff = 3
  r_free_flags_fraction = 0.1
  r_free_flags_max_free = 2000
  r_free_flags_use_lattice_symmetry = True
  r_free_flags_lattice_symmetry_max_delta = 5
  allow_overlapping = None
  fix_ligand_occupancy = None
  remove_outlier_segments = True
  twin_law = None
  max_occ = None
  refine_before_rebuild = True
  refine_with_ncs = True
  refine_xyz = True
  s_annealing = False
  skip_hexdigest = False
  use_hl_anom_in_refinement = False
  use_hl_if_present = True
}
thoroughness {
  build_outside = True
  connect = True
  extensive_build = False
  fit_loops = True
  insert_helices = True
  n_cycle_build = None
  n_cycle_build_max = 6
  n_cycle_build_min = 1
  n_cycle_rebuild_max = 15
  n_cycle_rebuild_min = 1
  n_mini = 10
  n_random_frag = 0
  n_random_loop = 3
  n_try_rebuild = 2

```

```

ncycle_refine = 3
number_of_models = None
number_of_parallel_models = 0
skip_combine_extend = False
fully_skip_combine_extend = False
thorough_loop_fit = True
}
general {
  coot_name = "coot"
  i_ran_seed = 72432
  raise_sorry = False
  background = True
  check_wait_time = 1
  max_wait_time = 1
  wait_between_submit_time = 1
  cache_resolve_libs = True
  resolve_size = "12"
  check_run_command = False
  run_command = "sh "
  queue_commands = None
  condor_universe = "vanilla"
  add_double_quotes_in_condor = True
  condor = None
  last_process_is_local = True
  skip_r_factor = False
  test_flag_value = Auto
  skip_xtriage = False
  base_path = None
  temp_dir = None
  clean_up = True
  print_citations = True
  solution_output_pickle_file = None
  job_title = None
  top_output_dir = None
  wizard_directory_number = None
  verbose = False
  extra_verbose = False
  debug = False
  require_nonzero = True
  remove_path_word_list = None
  fill = False
  res_fill = None
  check_only = False
  keep_files = "overall_best*" "AutoBuild_run_*.log"
  after_autosol = False
  nbatch = 3
  nproc = 1
  quick = False
  resolve_command_list = None

```

```

    resolve_pattern_command_list = None
    ignore_errors_in_subprocess = False
    send_notification = False
    notify_email = None
}
special_keywords {
    write_run_directory_to_file = None
}
run_control {
    coot = None
    ignore_blanks = None
    stop = None
    display_facts = None
    display_summary = None
    carry_on = None
    run = None
    copy_run = None
    display_runs = None
    delete_runs = None
    display_labels = None
    dry_run = False
    params_only = False
    display_all = False
}
non_user_parameters {
    gui_output_dir = None
    background_map = None
    boundary_background_map = None
    extend_try_list = True
    force_combine_extend = False
    model_list = None
    oasis_cnos = None
    offset_boundary_background_map = None
    skip_refine = False
    sg = None
    input_data_file = None
    input_map_file = "PDBID.mtz"
    input_refinement_file = Auto
    input_pdb_file = None
    input_seq_file = Auto
    super_quick = None
    require_test_set = False
}
}

```

## B.5.2 ARP/wARP

The following command line was used to build data set ID 2ASH (resolution 2.1 Å) and the initial model from PHENIX Autobuild without Parrot.

```

set albe = 0
set arpipc =
set arpwarpcdir = temp_tracing
set bcut1 = 2.0
set bcut2 = 2.0
set bcut3 = 2.0
set CCP4I_DEFFILE = UNDEFINED
set cell = '170.109 99.745 124.866 90.000 123.929 90.000'
set cgr = 1
set compareto =
set damp = '1.0 1.0'
set datafile = PDBID.mtz
set dipcut1 = 0.035
set dipcut2 = 0.010
set dipvali = 1
set emmode = 0
set fakedata = '0 0 0'
set fbest =
set flatten = 0
set fom = hltofom.Phi_fom.fom
set fp = FP
set freebuild = 0
set freelabin = 'FREE=FreeR_flag'
set freeloops = 0
set fsig = 3.2
set heavyin =
set hmainpostfit = 1
set is_semet = 0
set JOB_ID = PDBID
set keepdata = SOFTWARE_DEVELOPERS
set keepjunk = 0
set loops = 1
set modeccp4i = WARPNTACEMODEL
set modelin = /PHENIXAutobuild/PDBID.pdb
set models = 1
set multit = 5
set ncsextension = 1
set ncsrestraints = 1
set ncsr_local = 1
set nnuc = 0
set parfile = PDBID/arp_warp_classic.par
set phaselabin =
set phaseref =
set phibest = hltofom.Phi_fom.phi

```

```

set PROJECT = COMMAND_LINE_SUBMISSION
set protsize = 0
set rand1 = 0
set rand2 = 0
set rand3 = 0
set randshift1 = 0.5
set randshift2 = 0.5
set randshift3 = 0.5
set randtimes = 0
set refmax = MLKF
set remote = 0
set remoteemail =
set resol = '103.605 2.100'
set restraints = 1
set restrcyc = 50
set restrref = 5
set ridgerestraints = 0
set rrcyc = 1
set rsig = 1.0
set sad = 0
set sadcard =
set scaleopt = 'SIMPLE LSSC ANIS'
set scalml = 'SCAL MLSC WORK'
set seqin = PDBID.fasta
set side = 1
set sidemethod = SEQQY
set sigfp = SIGFP
set skip = 0
set solvent = 1
set solventc = 1.0000
set sym = 5
set twin = 0
set upmore = 1
set version = 8.0
set warpbin = /ccp4/7.0.066/arp_warp_8.0/bin/bin-x86_64-Linux
set weightv =
set wilsonb = 44.02
set wmat = AUTO
set WORKDIR = /PDBID
set xyzlim = '0.00000 0.50000 0.00000 1.00000 0.00000 0.50000'

```

### B.5.3 Buccaneer

The following command line was used to build data set ID 1O6A (resolution 1.9 Å) and the initial model from PHENIX AutoBuild.

```

mtzin PDBID.mtz
seqin PDBID.fasta

```



APPENDIX B. PAIRWISE RUNNING OF AUTOMATED CRYSTALLOGRAPHIC MODEL-BUILDING PIPELINES (ADDITIONAL RESULTS)

---

```
colin-fo FP,SIGFP
colin-hl parrot.ABCD.A, parrot.ABCD.B, parrot.ABCD.C, parrot.ABCD.D
colin-free FreeR_flag
buccaneer-anisotropy-correction
buccaneer-fast
buccaneer-keyword verbose 5
cycles 5
pdbin PHENIXAutoBuild/PDBID.pdb
----- DEFAULT PARAMETERS -----
title buccaneer auto-build
pdbout buccaneer.pdb
buccaneer-new-residue-name UNK
buccaneer-resolution 2.0
buccaneer-1st-cycles 3
buccaneer-1st-correlation-mode false
buccaneer-1st-sequence-reliability 0.95
buccaneer-nth-cycles 2
buccaneer-nth-correlation-mode true
buccaneer-nth-sequence-reliability 0.95
refmac-twin false
refmac-mlhl true
prefix buccaneer/
-----
```

**Identifying incorrect fragments to  
improve backbone chain tracing using  
neural network in Buccaneer  
(additional results)**

**C.1 Comparison of R-work, R-free and structure  
correlation between Buccaneer and Buccaneer  
with neural network**

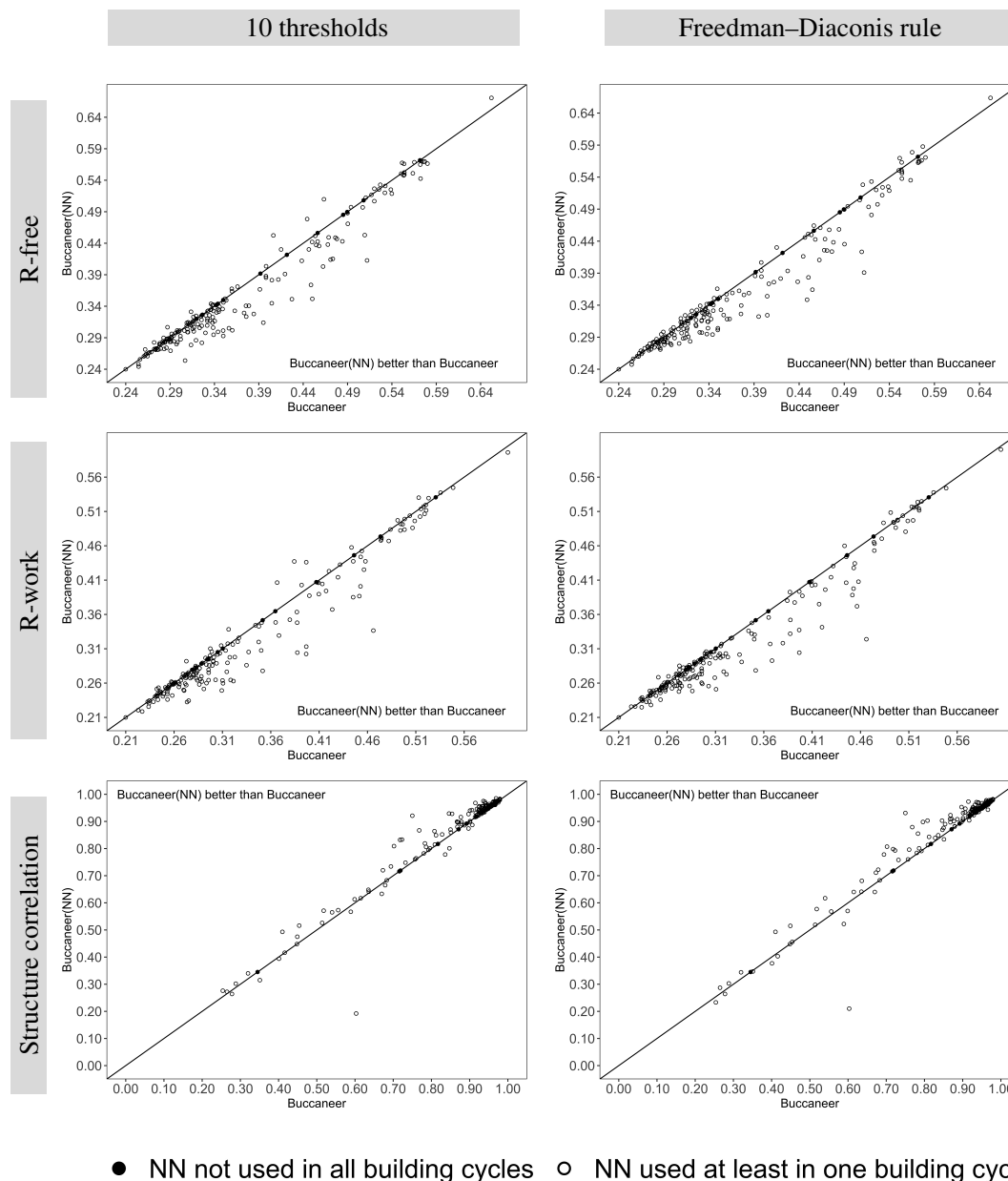


Figure C.1: Comparison of R-work, R-free and structure correlation between Buccaneer and Buccaneer with neural network (Buccaneer(NN)) using ten thresholds and FreedmanDiaconis rule for the recently deposited experimental phasing data sets. The results where Buccaneer(NN) is better than Buccaneer either below or above the diagonal is indicated in the figures.

# Bibliography

- [1] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235. URL <https://doi.org/10.1093/nar/28.1.235>.
- [2] RCSB. PDB data distribution by experimental method and molecular type, February 2020. URL <https://www.rcsb.org/stats/summary>.
- [3] Yifan Cheng. Single-particle cryo-EM at crystallographic resolution. *Cell*, 161(3):450–457, 2015.
- [4] Garry Taylor. The phase problem. *Acta Crystallographica Section D: Biological Crystallography*, 59(11):1881–1890, 2003.
- [5] Aleksandar Bijelic and Annette Rompel. Polyoxometalates: more than a phasing tool in protein crystallography. *ChemTexts*, 4(3):1–27, 2018.
- [6] Philip Evans and Airlie McCoy. An introduction to molecular replacement. *Acta Crystallographica Section D: Biological Crystallography*, 64(1):1–10, 2008.
- [7] Airlie J McCoy and Randy J Read. Experimental phasing: best practice and pitfalls. *Acta Crystallographica Section D: Biological Crystallography*, 66(4): 458–469, 2010.
- [8] Elspeth F Garman and Robin Leslie Owen. Cryocooling and radiation damage in macromolecular crystallography. *Acta Crystallographica Section D: Biological Crystallography*, 62(1):32–47, 2006.
- [9] Andrea Thorn, James Parkhurst, Paul Emsley, Robert A Nicholls, Melanie Vollmar, Gwyndaf Evans, and Garib N Murshudov. AUSPEX: a graphical tool for

- x-ray diffraction data analysis. *Acta Crystallographica Section D: Structural Biology*, 73(9):729–737, 2017.
- [10] Richard J Morris, Petrus H Zwart, Serge Cohen, Francisco J Fernandez, Mattheos Kakaris, Olga Kirillova, Clemens Vornrhein, Anastassis Perrakis, and Victor S Lamzin. Breaking good resolutions with ARP/wARP. *Journal of synchrotron radiation*, 11(1):56–59, 2004.
- [11] Emad Alharbi, Paul S Bond, Radu Calinescu, and Kevin Cowtan. Comparison of automated crystallographic model-building pipelines. *Acta Crystallographica Section D: Structural Biology*, 75(12), 2019.
- [12] Melanie Vollmar, James M Parkhurst, Dominic Jaques, Arnaud Baslé, Garib N Murshudov, David G Waterman, and Gwyndaf Evans. The predictive power of data-processing statistics. *IUCrJ*, 7(2), 2020.
- [13] Victor S Lamzin and Keith S Wilson. Automated refinement of protein models. *Acta Crystallographica Section D: Biological Crystallography*, 49(1):129–147, 1993.
- [14] Anastassis Perrakis, Richard Morris, and Victor S Lamzin. Automated protein model building combined with iterative structure refinement. *Nature Structural and Molecular Biology*, 6(5):458, 1999.
- [15] Richard J Morris, Anastassis Perrakis, and Victor S Lamzin. ARPwARP and automatic interpretation of protein electron density maps. In *Methods in enzymology*, volume 374, pages 229–244. Elsevier, 2003.
- [16] Gerrit Langer, Serge X Cohen, Victor S Lamzin, and Anastassis Perrakis. Automated macromolecular model building for X-ray crystallography using ARP/wARP version 7. *Nature protocols*, 3(7):1171, 2008.
- [17] Gerrit G Langer, Saul Hazledine, Tim Wiegels, Ciaran Carolan, and Victor S Lamzin. Visual automated macromolecular model building. *Acta Crystallographica Section D: Biological Crystallography*, 69(4):635–641, 2013.

- [18] Kevin Cowtan. The buccaneer software for automated model building. 1. Tracing protein chains. *Acta Crystallographica Section D: Biological Crystallography*, 62(9):1002–1011, 2006.
- [19] Kevin Cowtan. Fitting molecular fragments into electron density. *Acta Crystallographica Section D: Biological Crystallography*, 64(1):83–89, 2008.
- [20] Thomas C Terwilliger, Ralf W Grosse-Kunstleve, Pavel V Afonine, Nigel W Moriarty, Peter H Zwart, L-W Hung, Randy J Read, and Paul D Adams. Iterative model building, structure refinement and density modification with the PHENIX Autobuild wizard. *Acta Crystallographica Section D: Biological Crystallography*, 64(1):61–69, 2008.
- [21] George M Sheldrick. A short history of SHELX. *Acta Crystallographica Section A: Foundations of Crystallography*, 64(1):112–122, 2008.
- [22] George M Sheldrick. Experimental phasing with SHELXC/D/E: combining chain tracing with density modification. *Acta Crystallographica Section D: Biological Crystallography*, 66(4):479–485, 2010.
- [23] Andrea Thorn and George M Sheldrick. Extending molecular-replacement solutions with SHELXE. *Acta Crystallographica Section D: Biological Crystallography*, 69(11):2251–2256, 2013.
- [24] Isabel Usón and George M Sheldrick. An introduction to experimental phasing of macromolecules illustrated by SHELX; new autotracing features. *Acta Crystallographica Section D: Structural Biology*, 74(2):106–116, 2018.
- [25] Melanie Vollmar and Gwyndaf Evans. Machine learning applications in macromolecular X-ray crystallography. *Crystallography Reviews*, pages 1–48, 2021.
- [26] PS Bond, KS Wilson, and KD Cowtan. Predicting protein model correctness in Coot using machine learning. *Acta Crystallographica Section D: Structural Biology*, 76(8), 2020.
- [27] Grzegorz Chojnowski, Joana Pereira, and Victor S Lamzin. Sequence assignment for low-resolution modelling of protein crystal structures. *Acta Crystallographica Section D: Structural Biology*, 75(8):753–763, 2019.

- [28] Soon Wen Hoh, Tom Burnley, and Kevin Cowtan. Current approaches for automated model building into cryo-EM maps using buccaneer with CCP-EM. *Acta Crystallographica Section D: Structural Biology*, 76(6):531–541, 2020.
- [29] Guoyao Wu. Amino acids: metabolism, functions, and nutrition. *Amino acids*, 37(1):1–17, 2009.
- [30] Kallol M Biswas, Daniel R DeVido, and John G Dorsey. Evaluation of methods for measuring amino acid hydrophobicities and interactions. *Journal of Chromatography A*, 1000(1-2):637–655, 2003.
- [31] JC Biro. Amino acid size, charge, hydrophathy indices and matrices for protein structure analysis. *Theoretical Biology and Medical Modelling*, 3(1):1–12, 2006.
- [32] H-D Belitz, Werner Grosch, and Peter Schieberle. Amino acids, peptides, proteins. In *Food chemistry*, pages 8–91. Springer, 2004.
- [33] G.N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7(1):95–99, 1963. ISSN 0022-2836. doi: [https://doi.org/10.1016/S0022-2836\(63\)80023-6](https://doi.org/10.1016/S0022-2836(63)80023-6). URL <https://www.sciencedirect.com/science/article/pii/S0022283663800236>.
- [34] André Mann. 15 - conformational restriction and/or steric hindrance in medicinal chemistry. In Camille G. Wermuth, editor, *The Practice of Medicinal Chemistry (Second Edition)*, pages 233–250. Academic Press, London, second edition edition, 2003. ISBN 978-0-12-744481-9. doi: <https://doi.org/10.1016/B978-012744481-9/50019-2>. URL <https://www.sciencedirect.com/science/article/pii/B9780127444819500192>.
- [35] Jan Drenth. *Principles of protein X-ray crystallography*. Springer Science & Business Media, 2007.
- [36] Kurt Wüthrich. The way to NMR structures of proteins. *Nature structural biology*, 8(11):923–925, 2001.

- [37] Richard Henderson and P Nigel T Unwin. Three-dimensional model of purple membrane obtained by electron microscopy. *Nature*, 257(5521):28–32, 1975.
- [38] Richard Henderson, Joyce M Baldwin, Thomas A Ceska, Friedrich Zemlin, Erich Beckmann, and Kenneth H Downing. Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy. *Journal of molecular biology*, 213(4):899–929, 1990.
- [39] Edward H Egelman. The current revolution in cryo-EM. *Biophysical journal*, 110(5):1008–1012, 2016.
- [40] William Henry Bragg and William Lawrence Bragg. The reflection of X-rays by crystals. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 88(605):428–438, 1913.
- [41] Eleanor Dodson. Introduction to molecular replacement: a time perspective. *Acta Crystallographica Section D: Structural Biology*, 2021.
- [42] MS Smyth and JHJ Martin. X-ray crystallography. *Molecular Pathology*, 53(1): 8, 2000.
- [43] Alexander McPherson. *Introduction to macromolecular crystallography*. John Wiley & Sons, 2011.
- [44] Garry L Taylor. Introduction to phasing. *Acta Crystallographica Section D: Biological Crystallography*, 66(4):325–338, 2010.
- [45] Mirosława Dauter and Zbigniew Dauter. Many ways to derivatize macromolecules and their crystals for phasing. In *Protein Crystallography*, pages 349–356. Springer, 2017.
- [46] Paul Emsley, Bernhart Lohkamp, William G Scott, and Kevin Cowtan. Features and development of Coot. *Acta Crystallographica Section D: Biological Crystallography*, 66(4):486–501, 2010.
- [47] S McNicholas, E Potterton, KS Wilson, and MEM Noble. Presenting your structures: the CCP4mg molecular-graphics software. *Acta Crystallographica Section D: Biological Crystallography*, 67(4):386–394, 2011.



- [48] Pdb101: Learn: Guide to understanding pdb data: Resolution. URL <https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/resolution>. Accessed: 2018-07-14.
- [49] Kevin D Cowtan and Kam YJ Zhang. Density modification for macromolecular phase improvement. *Progress in biophysics and molecular biology*, 72(3): 245–270, 1999.
- [50] Kevin Cowtan. Recent developments in classical density modification. *Acta Crystallographica Section D: Biological Crystallography*, 66(4):470–478, 2010.
- [51] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
- [52] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- [53] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *ACM Computing Surveys (CSUR)*, 53(2):1–33, 2020.
- [54] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [55] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.
- [56] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [57] Biological Magnetic Resonance Data Bank. Number of atoms in amino acid. [https://bmr.io/ref\\_info/aadata.dat](https://bmr.io/ref_info/aadata.dat), 2021. Online; accessed 30 September 2021.
- [58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn:

- Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.
- [59] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.
- [60] Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, 50(8):3668–3681, 2019.
- [61] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [62] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [63] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and SS Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.
- [64] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.
- [65] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.
- [66] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.
- [67] Anton Maximilian Schaefer, Steffen Udluft, and Hans-Georg Zimmermann. Learning long-term dependencies with recurrent neural networks. *Neurocomputing*, 71(13-15):2481–2488, 2008.

- [68] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [69] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [70] Garib N. Murshudov, Pavol Skubák, Andrey A. Lebedev, Navraj S. Pannu, Roberto A. Steiner, Robert A. Nicholls, Martyn D. Winn, Fei Long, and Alexei A. Vagin. *REFMAC5* for the refinement of macromolecular crystal structures. *Acta Crystallographica Section D*, 67(4):355–367, Apr 2011. doi: 10.1107/S0907444911001314. URL <https://doi.org/10.1107/S0907444911001314>.
- [71] Pavel V Afonine, Ralf W Grosse-Kunstleve, Nathaniel Echols, Jeffrey J Headd, Nigel W Moriarty, Marat Mustyakimov, Thomas C Terwilliger, Alexandre Urzhumtsev, Peter H Zwart, and Paul D Adams. Towards automated crystallographic structure refinement with phenix.refine. *Acta Crystallographica Section D: Biological Crystallography*, 68(4):352–367, 2012.
- [72] Henry van den Bedem, Guenter Wolf, Qingping Xu, and Ashley M Deacon. Distributed structure determination at the JCSG. *Acta Crystallographica Section D*, 67(4):368–375, 2011.
- [73] Grzegorz Chojnowski. Methods underlying extension of MR solutions in ARP/wARP, 2019. presentation at the CCP4 Study Weekend.
- [74] Richard J Morris, Anastassis Perrakis, and Victor S Lamzin. ARP/wARP’s model-building algorithms. I. The main chain. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):968–975, 2002.
- [75] Martyn D. Winn, Charles C. Ballard, Kevin D. Cowtan, Eleanor J. Dodson, Paul Emsley, Phil R. Evans, Ronan M. Keegan, Eugene B. Krissinel, Andrew G. W. Leslie, Airlie McCoy, Stuart J. McNicholas, Garib N. Murshudov, Navraj S. Pannu, Elizabeth A. Potterton, Harold R. Powell, Randy J. Read, Alexei Vagin, and Keith S. Wilson. Overview of the CCP4 suite and

- current developments. *Acta Crystallographica Section D*, 67(4):235–242, Apr 2011. doi: 10.1107/S0907444910045749. URL <https://doi.org/10.1107/S0907444910045749>.
- [76] Elizabeth Potterton, Peter Briggs, Maria Turkenburg, and Eleanor Dodson. A graphical user interface to the CCP4 program suite. *Acta Crystallographica Section D: Biological Crystallography*, 59(7):1131–1137, 2003.
- [77] Liz Potterton, Jon Agirre, Charles Ballard, Kevin Cowtan, Eleanor Dodson, Phil R Evans, Huw T Jenkins, Ronan Keegan, Eugene Krissinel, Kyle Stevenson, et al. CCP4i2: the new graphical user interface to the CCP4 program suite. *Acta Crystallographica Section D: Structural Biology*, 74(2):68–84, 2018.
- [78] Garib N Murshudov, Pavol Skubák, Andrey A Lebedev, Navraj S Pannu, Roberto A Steiner, Robert A Nicholls, Martyn D Winn, Fei Long, and Alexei A Vagin. REFMAC5 for the refinement of macromolecular crystal structures. *Acta Crystallographica Section D: Biological Crystallography*, 67(4):355–367, 2011.
- [79] Thomas C Terwilliger. Maximum-likelihood density modification. *Acta Crystallographica Section D: Biological Crystallography*, 56(8):965–972, 2000.
- [80] Thomas C Terwilliger. Statistical density modification with non-crystallographic symmetry. *Acta Crystallographica Section D: Biological Crystallography*, 58(12):2082–2086, 2002.
- [81] Thomas C Terwilliger. Automated main-chain model building by template matching and iterative fragment extension. *Acta Crystallographica Section D: Biological Crystallography*, 59(1):38–44, 2003.
- [82] Axel T Brünger. Free R value: A novel statistical quantity for assessing the accuracy of crystal structures. *Nature*, 355(6359):472–475, 1992.
- [83] Gábor Bunkóczi, Airlie J McCoy, Nathaniel Echols, Ralf W Grosse-Kunstleve, Paul D Adams, James M Holton, Randy J Read, and Thomas C Terwilliger. Macromolecular X-ray structure determination using weak, single-wavelength anomalous data. *Nature Methods*, 12(2):127, 2015.

- [84] ARP/wARP. ARP/wARP user guide, October 2019. URL <https://web.archive.org/web/20191002123116/https://www.embl-hamburg.de/ARP/Manual/UserGuide8.0.html>.
- [85] Serge X Cohen, Marouane Ben Jelloul, Fei Long, Alexei Vagin, Puck Knipscheer, Joyce Lebbink, Titia K Sixma, Victor S Lamzin, Garib N Murshudov, and Anastassis Perrakis. ARP/wARP and molecular replacement: the next generation. *Acta Crystallographica Section D: Biological Crystallography*, 64(1): 49–60, 2008.
- [86] Dorothee Liebschner, Pavel V Afonine, Matthew L Baker, Gábor Bunkóczi, Vincent B Chen, Tristan I Croll, Bradley Hintze, L-W Hung, Swati Jain, Airlie J McCoy, et al. Macromolecular structure determination using X-rays, neutrons and electrons: recent developments in Phenix. *Acta Crystallographica Section D: Structural Biology*, 75(10):861–877, 2019.
- [87] Grzegorz Chojnowski, Koushik Choudhury, Philipp Heuser, Egor Sobolev, Joana Pereira, Umut Oezugurel, and Victor S. Lamzin. The use of local structural similarity of distant homologues for crystallographic model building from a molecular-replacement solution. *Acta Crystallographica Section D*, 76(3):248–260, Mar 2020. doi: 10.1107/S2059798320000455. URL <https://doi.org/10.1107/S2059798320000455>.
- [88] John C Kendrew, G Bodo, Howard M Dintzis, RG Parrish, Harold Wyckoff, and David C Phillips. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, 181(4610):662–666, 1958.
- [89] E Alharbi, R Calinescu, and K Cowtan. Pairwise running of automated crystallographic model-building pipelines. *Acta Crystallographica Section D: Structural Biology*, 76(9), 2020.
- [90] Blake E Ashforth and Fred Mael. Social identity theory and the organization. *Academy of management review*, 14(1):20–39, 1989.
- [91] Thomas C Terwilliger, Paul D Adams, Randy J Read, Airlie J McCoy, Nigel W Moriarty, Ralf W Grosse-Kunstleve, Pavel V Afonine, Peter H Zwart, and L-W Hung. Decision-making in structure solution using bayesian estimates of

- map quality: the PHENIX Autosol wizard. *Acta Crystallographica Section D: Biological Crystallography*, 65(6):582–601, 2009.
- [92] Evgeny Krissinel. Enhanced fold recognition using efficient short fragment clustering. *Journal of molecular biochemistry*, 1(2):76, 2012.
- [93] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [94] Frank Eibe, Mark A Hall, and Ian H Witten. The WEKA workbench. online appendix for data mining: practical machine learning tools and techniques. In *Morgan Kaufmann*. 2016.
- [95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [96] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- [97] Rishabh Choudhary and Hemant Kumar Gianey. Comprehensive review on supervised machine learning algorithms. In *2017 International Conference on Machine Learning and Data Science (MLDS)*, pages 37–43. IEEE, 2017.
- [98] Eibe Frank and Remco R Bouckaert. Conditional density estimation with class probability estimators. In *Asian Conference on Machine Learning*, pages 65–81. Springer, 2009.
- [99] Jaclyn Bibby, Ronan M Keegan, Olga Mayans, Martyn D Winn, and Daniel J Rigden. AMPLE: a cluster-and-truncate approach to solve the crystal structures of small proteins using rapidly computed ab initio models. *Acta Crystallographica Section D: Biological Crystallography*, 68(12):1622–1631, 2012.
- [100] Adam J Simpkin, Felix Simkovic, Jens MH Thomas, Martin Savko, Andrey Lebedev, Ville Uski, Charles Ballard, Marcin Wojdyr, Rui Wu, Ruslan Sanishvili, et al. SIMBAD: a sequence-independent molecular-replacement pipeline. *Acta Crystallographica Section D: Structural Biology*, 74(7):595–605, 2018.

- [101] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [102] K Cowtan. IUCr comput. *Commun. Newsl*, 2:4–9, 2003.
- [103] Kevin Cowtan. Fast fourier feature recognition. *Acta Crystallographica Section D: Biological Crystallography*, 57(10):1435–1444, 2001.
- [104] Simon C Lovell, Ian W Davis, W Bryan Arendall III, Paul IW De Bakker, J Michael Word, Michael G Prisant, Jane S Richardson, and David C Richardson. Structure validation by  $C\alpha$  geometry:  $\phi$ ,  $\psi$  and  $C\beta$  deviation. *Proteins: Structure, Function, and Bioinformatics*, 50(3):437–450, 2003.
- [105] François Chollet et al. Keras. <https://keras.io>, 2015.
- [106] Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pages 195–201. Springer, 1995.
- [107] Rory Conlin, Keith Erickson, Joseph Abbate, and Egemen Kolemen. Keras2c: A library for converting Keras neural networks to real-time compatible C. *Engineering Applications of Artificial Intelligence*, 100:104182, 2021.
- [108] David Freedman and Persi Diaconis. On the histogram as a density estimator: L 2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(4):453–476, 1981.
- [109] Paul S Bond. *Next generation software for placing atoms into electron density maps*. PhD thesis, University of York, 2021.
- [110] Kevin Cowtan, Jon Agirre, and Stephen Metcalfe. Shift fields: A new approach to refinement using non-atomic parametrizations. In *ACTA CRYSTALLOGRAPHICA A-FOUNDATION AND ADVANCES*, volume 74, pages E151–E151. INT UNION CRYSTALLOGRAPHY 2 ABBEY SQ, CHESTER, CH1 2HU, ENGLAND, 2018.

- [111] K Cowtan, S Metcalfe, and Paul Bond. Shift-field refinement of macromolecular atomic models. *Acta Crystallographica Section D: Structural Biology*, 76(12), 2020.
- [112] Paul Emsley and Kevin Cowtan. Coot: model-building tools for molecular graphics. *Acta Crystallographica Section D: Biological Crystallography*, 60(12):2126–2132, 2004.
- [113] Kevin Cowtan. Automated nucleic acid chain tracing in real time. *IUCrJ*, 1(6): 387–392, 2014.