Bioinformatics Tool and Web Server Development

Focusing on Structural Bioinformatics Applications

A thesis submitted in fulfilment of the requirement for the degree

of

DOCTOR OF PHILOSOPHY
IN BIOINFORMATICS

of

RHODES UNIVERSITY, SOUTH AFRICA

Research Unit in Bioinformatics (RUBi)

Department of Biochemistry and Microbiology

Faculty of Science

by

Margaret Nabatanzi

September 2022

ABSTRACT

This thesis is divided into two main sections: Part 1 describes the design, and evaluation of the accuracy of a new web server – PRotein Interactive MOdeling (PRIMO-Complexes) for modeling protein complexes and biological assemblies. The second part describes the development of bioinformatics tools to predict HIV-1 drug resistance and support bioinformatics research and education.

Recent technological advances have resulted in a tremendous increase in the number of sequences and protein structures deposited in the Universal Protein Resource Knowledgebase (UniProtKB) and the Protein Data Bank (PDB). However, the number of sequences has increased at a higher rate compared with the experimentally solved multimeric protein structures. This is partly due to advances in high-throughput sequencing technology. To fill this protein sequence-structure gap, computational approaches have been developed to predict protein structures from available sequences. Computational approaches include template-based and *ab initio* modeling with the former being the most reliable. Template-based modeling process can be achieved using either standalone software or automated modeling web servers. However, using standalone software requires familiarity with command-line interfaces as well as utilising other intermediate programs which could be daunting to novice users. To alleviate some of these problems, the modeling process has been automated, however, it still has numerous challenges.

To date, only a few web servers that support multimeric protein modeling have been developed and even these provide little, if any user involvement in the process. To address some of these issues, a new web server – PRIMO-Complexes – was developed to model protein complexes and biological assemblies. The existing PRIMO web server could only model monomeric proteins. Part 1 of this thesis provides a detailed account of the development and evaluation of

PRIMO-Complexes. The rationale for developing this new web server was based on the understanding that most proteins function as protein multimers and often the ligand-binding sites, and enzyme active sites are located at the protein-protein interfaces. It, therefore, necessitated developing capabilities for modeling multimeric proteins.

PRIMO-Complexes web server was developed using the Waterfall system development life cycle model, is based on the Django web framework and makes use of high-performance computing resources to execute jobs. The accuracy of the algorithms embedded in PRIMO-Complexes was evaluated and the results were promising. Additionally, PRIMO-Complexes performs comparatively well in relation to other web servers that offer multimeric protein modeling. Another unique feature of PRIMO-Complexes is its interactivity. The webserver was developed with capabilities for allowing users to model multimeric proteins with an appreciable degree of control over the process.

In the second part of the thesis several other bioinformatics tools are described, for example, a webserver for predicting HIV-1 drug resistance, the RUBi protein model repository, and a bioinformatics web portal for education and research resources. RUBi protein model repository stores verified theoretical models built using various modeling approaches. This enables users to easily access models to reproduce and/or further the research. This is described in chapter 5.

Chapter 6 describes the design and development of the Human Immunodeficiency type 1 Resistance Predictor (HIV-1 ResPredictor), a web application that employs artificial neural networks (ANN) to predict drug resistance in patients infected with HIV-1 subtype B. The ANNs and subtype classifiers performed well making this web application potentially useful to both clinicians and researchers in this era of personalised medicine.

Finally, chapter 7 describes a bioinformatics education web portal that equips students with information on how to use bioinformatics online resources. Being aware of these resources is

not enough without a deeper understanding and guidance on how to apply bioinformatics methods to solve practical problems. This web portal was aimed at familiarising students with the basic terminology and approaches in structural bioinformatics. Students will potentially gain skills to conduct real-life bioinformatics research to obtain biological insights.

DECLARATION

I declare that this thesis is my own unaided work unless otherwise stated. It is being submitted for the degree of Doctor of Philosophy in Bioinformatics in the Faculty of Science of Rhodes University. It has not been submitted before for any degree or examination in any other university.

MARGARET NABATANZI

DATED THIS 13th DAY OF SEPTEMBER 2022

DEDICATION

I dedicate this work to my late parents Mr. and Mrs. Samuel Mutyaba Semakula without whose support this achievement would not have been possible – you are forever in my heart. May the Almighty God bless you abundantly.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the following people:

My supervisor Professor Özlem Taştan Bishop for her guidance, and unending encouragement during the completion of this work. Her support, belief in me and unyielding commitment have inspired me to be a better person, and to achieve my goals.

The RUBi members both present and those who left, for their suggestions and discussions to better my work. My sincere gratitude goes to Dr Magambo Phillip Kimuda, Dr Caroline Ross, and Dr Olivier Sheik Amamuddy for their insightful discussions during the early stages of this work. I would also like to thank Michael Glenister, and Thulani Tshabalala for the help rendered to me when the RUBi servers and clusters were acting up. My sincere appreciation goes to Dr Dorothy Wavinya Nyamai for her relentless effort, support, and encouragement throughout this research journey and the writing period. I will forever be grateful.

Personal acknowledgement

My sincere gratitude goes to my friends and family, this academic journey wouldn't have been possible without your support, prayers, and constant guidance during the writing of this thesis.

Lastly and most importantly, I give all the glory to God for He has seen me through every step and has accorded me the strength, wisdom and understanding in this journey.

Funding acknowledgement

This work is supported by the National Institutes of Health Common Fund under grant number U41HG006941 to H3ABioNet.

TABLE OF CONTENTS

ABSTRACT ..... **i**
DECLARATION ..... **iv**
DEDICATION ..... **v**
ACKNOWLEDGEMENTS ..... **vi**
TABLE OF CONTENTS ..... **vii**
LIST OF FIGURES ..... **xiii**
LIST OF TABLES ..... **xv**
RESEARCH OUTPUTS ..... **xvi**
LIST OF ABBREVIATIONS ..... **xvii**
THESIS OVERVIEW ..... **xix**
PART I: MODELING OF PROTEIN COMPLEXES AND BIOLOGICAL ASSEMBLIES **1**
CHAPTER ONE **1**
LITERATURE REVIEW: PROTEIN STRUCTURE AND PREDICTION **1**
1 Introduction **1**
1.1 Proteins **1**
1.2 Protein structure **1**
1.2.1 Protein primary structure 2
1.2.2 Protein secondary structure 3
1.2.3 Protein tertiary structure 4
1.2.4 Protein quaternary structure 4
1.3 Protein function **5**
1.4 Methods for predicting protein structure **5**
1.4.1 Experimental techniques for predicting protein structures 5
1.4.1.1 Macromolecular crystallography 6
1.4.1.2 Nuclear magnetic resonance spectroscopy 8
1.4.1.3 Electron microscopy 9
1.4.1.4 Storage of experimentally determined protein structures 10
1.4.2 Computational techniques for predicting protein structures 12
1.4.2.1 Template-based modeling 13
1.4.2.2 Template-free modeling 14
1.4.2.3 Hybrid approaches to modeling protein structures 15
1.5 Assessment of protein structure prediction **16**

1.5.1 Critical Assessment of Protein Structure Prediction (CASP) 16
CHAPTER TWO 18
A REVIEW OF HOMOLOGY MODELING PROCEDURES AND TOOLS CURRENTLY IN USE 18
Chapter overview 18
2.1 Homology modeling steps 18
2.1.1 Template identification 19
2.2.2 Target-template sequence alignment 21
2.2.3 Model building 22
2.2.4 Model evaluation 22
2.3 Alignment programs 23
2.3.1 Clustal programs 23
2.3.2 MUSCLE program 25
2.3.3 MAFFT program 25
2.3.4 T-COFFEE program 26
2.3.5 PROMALS3D program 27
2.4 Model evaluation programs 27
2.4.1 VERIFY3D 27
2.4.2 ProSA 28
2.4.3 QMEAN 28
2.4.4 z-DOPE 29
2.4.5 PROCHECK 29
2.5 Existing automated protein modeling tools 30
2.5.1 I-TASSER 30
2.5.2 SWISS-MODEL 31
2.5.3 Rosetta 31
2.5.4 PRotein Interactive MOdeling (PRIMO) 31
2.6 Research motivation 32
2.7 Research aims and objectives 33
CHAPTER THREE 34
DEVELOPMENT OF PRIMO-COMPLEXES: A WEB SERVER FOR MODELING MULTIMERIC PROTEINS AND BIOLOGICAL ASSEMBLIES 34
Chapter overview 34
3.1 Implementation overview 34
3.1.1 Software development process model 36

3.2 Requirement definition 36
3.3 System and software design 37
3.3.1 System architecture for PRIMO-Complexes 38
3.3.2 Software architecture for PRIMO-Complexes 40
3.3.2.1 PRIMO-Complexes Django web server design 41
3.3.2.1.1 WebUi app 41
3.3.2.1.2 Users app 42
3.3.2.1.3 Evaluation app 42
3.3.2.1.4 Impi app 43
3.4 Implementation and unit testing 43
3.4.1 Web technologies and software used in PRIMO-Complexes development 44
3.4.1.1 Django web framework 44
3.4.1.1.1 Model 44
3.4.1.1.2 View 44
3.4.1.1.3 Template 45
3.4.1.2 Django REST framework 45
3.4.1.2.1 HTTP methods 45
3.4.1.3 MySQL database 46
3.4.1.4 KnockoutJS 46
3.4.1.4 Asynchronous JavaScript and XML (AJAX) 47
3.4.2 PRIMO-Complexes modeling algorithm 47
3.4.2.1 Template identification 48
3.4.2.2 Target-template sequence alignment 49
3.4.2.3 Protein modeling and evaluation 50
3.4.2.4 Building large macromolecules 51
3.4.3 Molecular and sequence visualization 51
3.4.3.1 PV-MSA plugin wrapper 51
3.4.3.2 MSA plugin 52
3.4.3.3 NGL viewer 53
3.4.4 Multimeric protein modeling Python scripts in PRIMO-Complexes 53
3.4.4.1 Python scripts functionality and modifications made 54
3.4.4.2 New scripts to specifically support protein multimeric modeling 58
3.5 Integration and system testing 59
3.6 Operation and maintenance 60
3.7 Results and discussion 60

3.7.1 Web server location and access 60
3.7.2 Description of the final system architecture and functionalities 60
3.7.2.1 PRIMO-Complexes software and system architecture 61
3.7.3 Job Input 63
3.7.4 Job handling – process and outputs 65
3.7.4.1 Template identification 65
3.7.4.2 Target-template sequence alignment 69
3.7.4.3 Protein modeling and evaluation 72
3.7.4.4 Building large macromolecules 77
3.8 Maintenance of PRIMO-Complexes 78
3.9 Comparison between the first PRIMO and PRIMO-Complexes 79
3.10 Strengths of PRIMO-Complexes web server 80
3.11 Limitations of PRIMO-Complexes web server 81
3.12 Conclusion and recommendations 81
CHAPTER FOUR 82
EVALUATION OF THE ACCURACY OF PRIMO-COMPLEXES TO MODEL MULTIMERIC PROTEINS 82
Chapter overview 82
Overview of the evaluation procedures 82
4.1 Testing of multimeric protein modeling scripts embedded in PRIMO-Complexes 83
4.1.1 Dataset generation and template identification 84
4.1.2 Clustering templates 87
4.1.3 Target-template sequence alignment and model generation 87
4.1.4 Normalizing and filtering protein models 88
4.2 Evaluation of the protein models 90
4.3 Additional assessment and evaluation performed 91
4.3.1 Remodeling target proteins and assessing PDB structures 91
4.3.2 Testing model refinement parameters 91
4.3.2.1 Evaluation of model refinement results 92
4.4 Continuous assessment of the pipeline 92
4.5 Case studies 93
4.6 Results and discussion 94
4.6.1 Protein modeling and filtering models 94
4.6.2 Evaluation of models using z-DOPE score 98

4.6.3 Evaluation of protein models using RMSD and other additional metric parameters 101
4.6.4 Model refinement results 107
4.6.5 Case studies 110
4.7 Conclusion 117
PART II: SIDE PROJECTS – DEVELOPMENT OF OTHER BIOINFORMATICS TOOLS AND WEB SERVERS 118
CHAPTER FIVE 118
RUBI PROTEIN MODEL REPOSITORY FOR ANNOTATED 3D PROTEIN STRUCTURES 118
Chapter overview 118
5.1 Introduction 118
5.2 Research aim and objectives 119
5.3 Implementation details 119
5.3.1 Web interface 119
5.3.2 Repository content 119
5.4 Results 120
5.4.1 Web server design and content 120
5.5 Examples of models in the repository 122
5.5.1 *Plasmodium* 1-deoxy-D-xylulose 5-phosphate reductoisomerase (DXR) 122
5.5.2 Glycoside Hydrolase 1 enzymes from *Bacillus licheniformis* 123
5.5.3 HIV protease 124
5.5.4 GTP CycloHydrolase 1 (GCH1) 125
5.5.5 Aminoacyl tRNA synthetases (aaRSs) 125
5.5.6 *Plasmodial* Transketolases 126
5.5.7 Heat shock proteins 128
5.5.8 Auxiliary Activity family 9 (AA9) 129
5.5.9 *Plasmodial* proteases 129
5.6 Maintenance 130
5.7 Conclusion 131
CHAPTER SIX 132
HIV-1 RESPREDICTOR: A WEB APPLICATION EMPLOYING ARTIFICIAL NEURAL NETWORKS TO PREDICT ANTIRETROVIRAL DRUG RESISTANCE IN PATIENTS INFECTED WITH HIV-1 SUBTYPE B 132
Chapter overview 132
6.1 Introduction 133

6.2 Research motivation **135**
6.3 Research aim and objectives **135**
6.4 Methodology **136**
6.4.1 Implementation 136
6.4.2 Translation of ANN models to Python 136
6.4.3 Determination of levels of agreement between MATLAB and translated Python scripts 138
6.4.4 Development of hidden Markov models to classify HIV-1 subtypes 138
6.4.5 Determination of a cut-off for subtyping sequences 140
6.4.6 Performance testing of HIV-1 ResPredictor with existing subtyping tools 140
6.5 Results and discussion **141**
6.5.1 Initial page and input formatting 142
6.5.2 Performance of HIV-1 subtype classifiers 144
6.5.3 Results page 145
6.5.4 Comparison of subtyping tools performance results 146
6.6 Conclusion **151**
CHAPTER SEVEN **153**
DEVELOPMENT OF A BIOINFORMATICS EDUCATION WEB PORTAL **153**
Chapter overview **153**
7.1 Introduction **153**
7.2 Motivation **154**
7.3 Research aim and objectives **154**
7.4 Implementation **155**
7.5 Results **155**
7.6 Conclusion **160**
CONCLUSIONS, FUTURE WORK, AND REFERENCES **161**
8.1 Conclusions **161**
8.2 Future work **162**
Supplementary data **163**
Chapter 5 supplementary data **163**
References **167**

LIST OF FIGURES

Fig 1.1. Protein structure organization levels. 2
Fig 1.2. A dehydration reaction to form a peptide bond between two amino acids with R and R_1 side chains. 3
Fig 1.3. Number of PDB structures released annually. 11
Fig 2.1. A schematic representation of template-based protein structure modeling. 19
Fig 3.1. A representation of the system development life cycle followed to develop PRIMO-Complexes 35
Fig 3.2. The diagram shows the Waterfall model phases used to implement the PRIMO-Complexes web server. 36
Fig 3.3. System architecture of PRIMO-Complexes – system components. 39
Fig 3.4. System architecture of PRIMO-Complexes – sub-systems. 40
Fig 3.5. Software architecture of PRIMO-Complexes. 41
Fig 3.6. A flowchart shows multimeric protein modeling algorithm incorporated by PRIMO-Complexes 48
Fig 3.7. The PRIMO-Complexes class diagram for backend scripts. 57
Fig 3.8. PRIMO-Complexes Django web server software architecture. 62
Fig 3.9. Execution of jobs submitted to PRIMO-Complexes web server. 63
Fig 3.10. PRIMO-Complexes home page. 65
Fig 3.11. Template identification results page for a homomultimeric protein. 67
Fig 3.12. Template identification results page for a heteromultimeric protein. 68
Fig 3.13. Template identification results page for a heteromultimeric protein - viral capsid. 69
Fig 3.14. Target-template alignment results page for a homomultimeric protein 70
Fig 3.15. Target-template alignment results page for a heteromultimeric protein. 71
Fig 3.16. Target-template alignment results page for a heteromultimeric protein - viral capsid 72
Fig 3.17. Protein modeling results page for a homomultimeric protein 74
Fig 3.18. Protein model evaluation results page for a homomultimeric protein model (model004). 75
Fig 3.19. Protein modeling results page for a heteromultimeric protein 76
Fig 3.20. Protein modeling results page for a heteromultimeric protein - viral capsid. 77
Fig 3.21. Protein building for large macromolecule for a heteromultimeric protein - viral capsid. 78
Fig 4. 1. A workflow diagram for testing the multimeric protein modeling Python scripts of PRIMO-Complexes 84
Fig 4.2. A workflow diagram showing steps followed to filter protein models. 89
Fig 4.3. Final homomultimeric targets that remained after performing the filtering steps. 95
Fig 4.4. Final heteromultimeric targets that remained after performing the filtering steps. 96
Fig 4.5. Box plots showing the average target-template sequence identities for each target-template combination per sequence identity bin 97
Fig 4.6. The z-DOPE score results for testing the multimeric protein modeling Python scripts for the four oligomeric states used in this work. 99
Fig 4.7. Evaluation of the multimeric protein modeling Python scripts for the homodimeric dataset.. 103

Fig 4.8. Evaluation of the multimeric protein modeling Python scripts for the homotetrameric dataset. 104
Fig 4.10. Evaluation of the multimeric protein modeling Python scripts for the heterotetrameric dataset. 106
Fig 4.11. Evaluation of homomultimeric models using model refinement options..... 108
Fig 4.12. Evaluation of heteromultimeric models using model refinement options..... 109
Fig 4.13. Cartoon representation of GTP cyclohydrolase I (GTP_CH-I) protein models..... 111
Fig 4.14. Cartoon representation of superoxide dismutase (hSod1) protein models. 112
Fig 4.15. Cartoon representation of hemoglobin subunit alpha (alpha-globin) protein models 113
Fig 5.1. A single page web interface for RUBi protein homology models repository 121
Fig 5.2. Protein models web page. 122
Fig 6.1. HIV-1 ResPredictor web interface. 142
Fig 6.2. Workflow of the HIV-1 ResPredictor web application. 143
Fig 6.3. Bland Altman plots. 144
Fig 6.4. Resistance prediction results of reverse transcriptase sequence against reverse transcriptase inhibitors 146
Fig 7.1. Home page for the bioinformatics web portal 156
Fig 7.2. Study protocols page showing the overall overview of the bioinformatics protocols. 157
Fig 7.3. Study protocols page showing information on how to use NCBI and BLAST..... 158
Fig 7.4. Study protocols page showing some of the information in the NCBI and BLAST protocol. 159

LIST OF TABLES

Table 3.1. Comparison between PRIMO and PRIMO-Complexes 79
Table 4.1. Summary of the first twenty protein oligomeric states for biological assemblies available in the PDB as of June 2020. 85
Table 4.2. Protein model quality evaluation results for modeling GTP cyclohydrolase I (GTP-CH-I) protein using PRIMO-Complexes and other modeling servers..... 114
Table 4.3. Protein model quality evaluation results for modeling superoxide dismutase (hSod1) protein using PRIMO-Complexes and other modeling servers. 115
Table 4.4. Protein model quality evaluation results for modeling hemoglobin subunit alpha (alpha-globin) protein using PRIMO-Complexes and other modeling servers. 116
Table 6.1. Summary of the important parameters and values obtained in classifying the subtype of protease and reverse transcriptase sequences. 145
Table 6.2. Comparison of performance of HIV-1 ResPredictor with existing automated subtyping tools: classification of protease and reverse transcriptase sequences 147
Table 6.3. Usability comparison between HIV-1 ResPredictor and other HIV resistance prediction servers 151

RESEARCH OUTPUTS

Conference oral presentations

Margaret Nabatanzi and Özlem Taştan Bishop. "**PRIMO: PRotein Interactive MOdeling pipeline - protein complex modeling**". H3Africa 13th Consortium Meeting, Tunis, Tunisia, 2019.

Margaret Nabatanzi and Özlem Taştan Bishop. "**PRIMO: PRotein Interactive MOdeling pipeline - protein complex modeling**". H3ABioNet AGM, Cape Town, South Africa, 2018.

Margaret Nabatanzi and Özlem Taştan Bishop. "**PRIMO: PRotein Interactive MOdeling pipeline – How to use PRIMO demonstrating with case studies**". H3ABioNet Online AGM, South Africa, 2021.

LIST OF ABBREVIATIONS

Abbreviation	Description
3D	Three-dimensional
AA9	Auxiliary Activity family 9
aaRSs	Aminoacyl tRNA synthetases
ABD	Anticodon binding domain
AF2	AlphaFold2
AIDS	Acquired immunodeficiency syndrome
AJAX	Asynchronous JavaScript and XML
ANNs	Artificial neural networks
API	Application Programming Interface
ART	Antiretroviral therapy
AUC	Area under the curve
BLAST	Basic Local Alignment Search Tool
BMRB	BioMagResBank
CAMEO	Continuous Automated Model EvaluatiOn
CAPRI	Critical assessment of prediction of interactions
CASP	Critical assessment of protein structure prediction
CAZy	Carbohydrate-Active Enzyme
cryo-EM	Cryogenic electron microscopy
CSS	Cascading Style Sheets
CTD	C-terminal domain
DHNP	Dihydroneopterin triphosphate
DOPE	Discrete Optimized Protein Energy
DXR	*Plasmodium* 1-deoxy-D-xylulose 5-phosphate reductoisomerase
EM	Electron microscopy
EMDB	Electron Microscopy Data Bank
ET	Electron Tomography
FFT	Fast Fourier transform
FROSTY	freezing rotational diffusion of protein solutions at low temperature and high viscosity
GCH1	GTP CycloHydrolase 1
GDT-HA	Global Distance Test – High Accuracy
GH	Glycoside hydrolases
GPCRdb	G protein-coupled receptor database
HIV-1 ResPredictor	HIV-1 Resistance Predictor
HIV-1	Human Immunodeficiency Virus type 1
HMM	Hidden Markov model
HPC	High Performance Computing
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HUMA	Human Mutation Analysis
I-TASSER	Iterative threading assembly refinement
JMS	Job Management System
JSON	JavaScript Object Notation
KO	Knockout
lDDT	Local Distance Difference Test
MAFFT	Multiple Alignment using Fast Fourier Transform

MEP	2-C-methyl-D-erythritol-4-phosphate
mmCIF	Macromolecular crystallographic information framework
MQAPs	Model Quality Assessment Programs
MSAs	Multiple sequence alignments
MUSCLE	Multiple Sequence Comparison by Log-Expectation
MVC	Model View Controller
MVT	Model View Template
MVVM	Model-View-View-Model
MX	Macromolecular crystallography
MySQL	My Structured Query Language
NCBI	National Center for Biotechnology Information
NJ	Neighbour-Joining method
NMR	Nuclear magnetic resonance spectroscopy
NNRTIs	Non-Nucleoside Reverse Transcriptase Inhibitors
NOE	Nuclear Overhauser effect
NRTIs	Nucleoside Reverse Transcriptase Inhibitors
ORM	Object relational mapper
PBDx	PDB data exchange dictionary
PDB	Protein Data Bank
PDBe	Protein Data Bank in Europe
PDBj	Protein Data Bank Japan
PIs	Protease Inhibitors
PPA	Profile-profile alignment
PR	Protease
PRIMO	PRotein Interactive Modeling
PROMALS3D	Profile Multiple Alignment with Local Structures and 3D constraints
ProSA	Protein Structure Analysis
PSI-BLAST	Position-Specific Iterated BLAST
PV	Protein viewer
QMEAN	Qualitative Model Energy ANalysis
RCSB	US Research Collaboratory for Structural Bioinformatics Protein Data Bank
RIA	Rich internet app
RMSD	Root mean square deviation
RT	Reverse transcriptase
RUBi	Research Unit in Bioinformatics
SDLC	System development life cycle
SFLD	Structure Function Linkage Database
SMTL	SWISS-MODEL Template Library
SQL	Server query language
T-COFFEE	Tree-based Consistency Objective Function for alignment Evaluation
TEM	Transmission electron microscope
ThDP	Thiamine diphosphate
TM	Template Modeling
TROSY	Transverse relaxation optimized spectroscopy
UniProtKB	Universal Protein Resource Knowledgebase
UPGMA	Unweighted Pair Grouping Method with Arithmetic-mean
wwPDB	Worldwide Protein Data Bank
XML	Extensible Markup Language

THESIS OVERVIEW

The aim of this body of work was to develop: 1) a web server for modeling multimeric proteins, 2) a repository for protein models, 3) a web server for predicting HIV-1 drug resistance, and 4) a bioinformatics education web portal.

This thesis is made up of seven chapters divided into 2 main parts. Research work conducted in this thesis is discussed in the first two parts, while the last part consists of the conclusion, supplementary data, and references.

Part 1 is an introduction about proteins, the development of PRIMO pipeline and evaluation of the accuracy of this web server to model protein complexes and biological assemblies. This part consists of chapters 1-4. Chapter 1 introduces protein structure and function, protein structure prediction. Chapter covers specifics of homology modeling and aim of the project together with objectives of part 1. Chapter 3 covers the design and development of PRIMO-Complexes that models multimeric proteins. Chapter 4 covers performance evaluation of multimeric protein modeling Python scripts of PRIMO-Complexes.

Part 2 covers other web servers that are vital in bioinformatics research, and it is made up of chapters 5-7. In chapter 5, the RUBi protein model repository is described. This repository stores annotated 3D protein structures from our group for easy research reproducibility. Chapter 6 describes HIV-1 ResPredictor web server, a web application that employs artificial neural networks to predict antiretroviral drug resistance in patients infected with HIV-1 subtype B. Chapter 7 follows this up by describing the bioinformatics education web portal embedded with protocols to familiarise students with the basic terminology and approaches in structural bioinformatics.

Both parts of this thesis are united under a common theme of tool development for bioinformatics research and analysis.

PART I: MODELING OF PROTEIN COMPLEXES AND BIOLOGICAL ASSEMBLIES

CHAPTER ONE

LITERATURE REVIEW: PROTEIN STRUCTURE AND PREDICTION

1 Introduction

1.1 Proteins

Proteins are polymers consisting of various amino acids joined by peptide bonds to form polypeptide chains. An amino acid is a molecule that contains amine and carboxyl functional groups as well as a side chain unique to each amino acid [1]. Proteins are made from twenty different standard amino acids some with varying chemical properties. Amino acid sequences are held together by different bonds and folded into different three-dimensional (3D) structures. Proteins are the functional molecules in living cells and account for 50% of their dry mass [1].

Proteins are grouped into three main types according to their shape and solubility. They include globular, fibrous, and membrane proteins. Globular proteins have a spherical structure. They have hydrophobic amino acid side chains in the interior and hydrophilic side chains exposed to the surface [2]. Fibrous proteins have regular linear structures and play a role in matrix formation in the cell. Membrane proteins are characterized by hydrophobic amino acid side chains exposed to the outside and play a vital role in cellular transport. Fibrous and membrane proteins are water-insoluble [3]. Proteins fold into various structures according to the amino acid composition which is also correlated with their biological function [4].

1.2 Protein structure

Proteins are organized into four different structural levels including primary, secondary, tertiary, and quaternary structures (Fig 1.1).

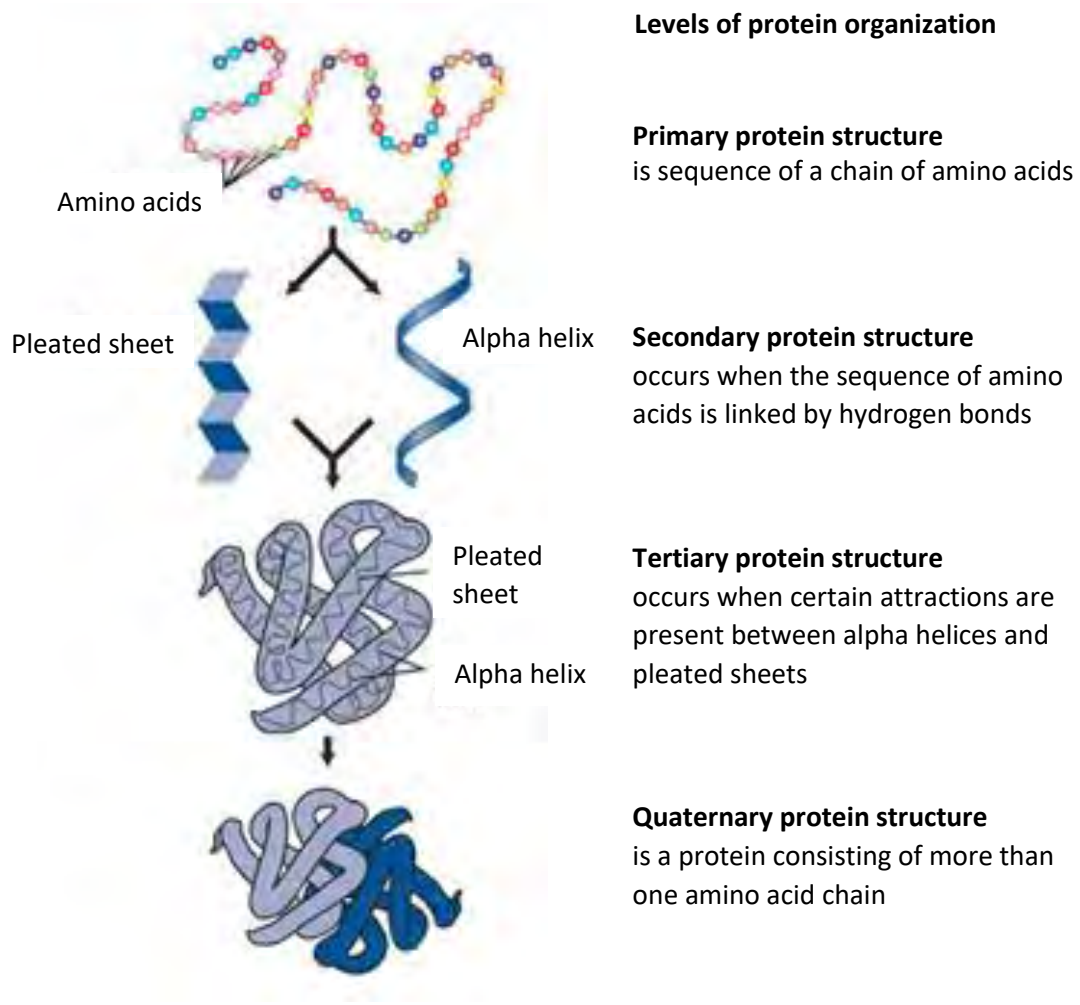


Fig 1.1. Protein structure organization levels. The primary structure is defined as the sequence of amino acid residues. The secondary structure comprises the α-helices or β-sheets conformations which interact to form the spatial arrangement of a protein resulting in the tertiary structure. Proteins can interact to form quaternary structures. Image adapted from [5].

1.2.1 Protein primary structure

The primary structure of a protein describes the specific linear sequence of amino acids that are linked together by peptide bonds to form a polypeptide chain [6]. Every protein is made up of unique sequences of amino acids which define its final 3D structure. The polypeptide chain comprises an end with a free α–amino group referred to as the N-terminus whereas the other end has a free α–carboxyl group known as the C-terminus. Each amino acid consists of a carboxylic acid group (-COOH) on one side, an amine group ($-NH_2$) on the other side and a unique residual group [1]. The carboxyl acid group bonds with an amine group on an adjacent amino acid to form a peptide bond (Fig 1.2). This sequence forms the backbone of the

polypeptide. The side chain defines characteristics, including size, polarity, and pH specific to each amino acid in the polypeptide chain. The appearance of the secondary, tertiary, and quaternary structure is determined by the position of amino acids in the polypeptide chain.

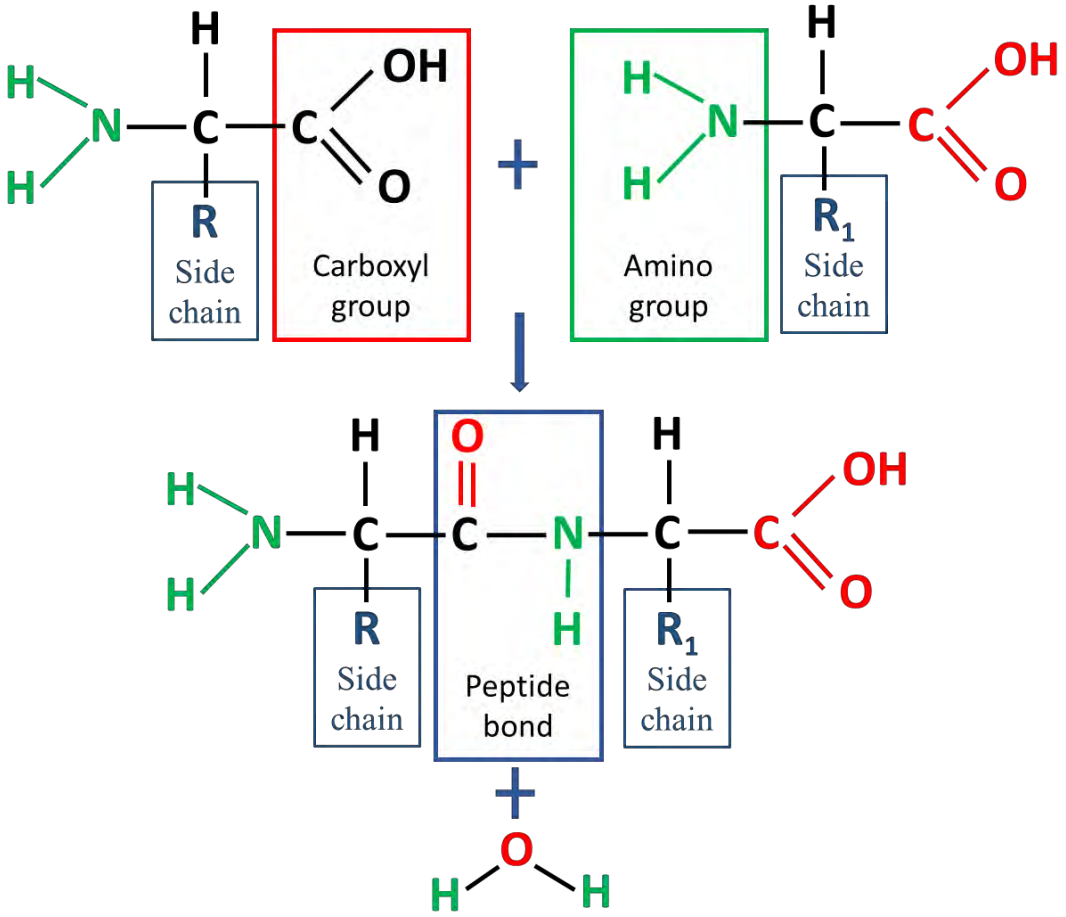


Fig 1.2. A dehydration reaction to form a peptide bond between two amino acids with R and R_1 side chains.

1.2.2 Protein secondary structure

The linear polypeptide chain twists and folds into regular patterns to form the secondary structure. There are mainly two types of secondary structures: alpha helix (α-helix), and beta-pleated sheet (β-sheets). The alpha helix is formed when the polypeptide chain folds into a rod-like structure with the backbone on the inside and the side chain on the outside [7]. The alpha helix is either left or right-handed. Right-handed alpha helices are the most common types found in biological systems. Alpha helices are found in nearly all globular, some fibrous, and

membrane proteins. The alpha helices are characterized by backbone dihedral angles (ψ, φ) around -60° and -45°.

The structure is held together by different types of bonds. The bond in alpha-helices is formed between amino acids of the same polypeptide chain whereas, for beta-sheets, bonds are formed between either the same or different polypeptide chains [8]. The beta sheets can form parallel chains through the interaction of adjacent chains facing the same direction or antiparallel chains which fold back and forth.

1.2.3 Protein tertiary structure

A protein tertiary structure is the 3D structure of the protein that is formed by interactions between the side chains of different amino acids in the polypeptide. A protein's tertiary structure is made up of several bonds including hydrophilic and hydrophobic interactions, disulphide, hydrogen, and ionic bonds [9]. Hydrophobic interactions play a key role in the folding of the polypeptide chain which usually occurs when amino acids with nonpolar side chains cluster at the core of the protein [10]. Weak Van der Waals forces, hydrogen, ionic and disulphide bonds stabilize the protein. The final shape of the tertiary structure is determined by the properties of amino acids forming the polypeptide chain [9].

1.2.4 Protein quaternary structure

Some proteins are made up of more than one polypeptide chain, resulting in a quaternary structure. This structure is formed through the combination of multiple (two or more) subunits, which are held together by noncovalent interactions [8]. These interactions include disulphide bridges, hydrogen bonds, and salt bridges. The subunits in protein complexes can either be identical (homomeric proteins) or different (heteromeric proteins) [11].

1.3 Protein function

The shape of proteins, often referred to as its structure, plays an important role in determining the biological functions in proteins. Atoms in protein molecules define the spatial arrangement or conformation of the protein thus establishing the biological function [2]. Theoretical, experimental, and computational studies indicate that proteins are not rigid but have internal collective atomic motions that modulate their biological function [12,13]. The motion of macromolecules, specifically proteins, is the crucial link between structure and function. Changes in conformation are associated with protein function. Protein motions are involved in numerous biological functions including enzyme catalysis [14], cellular locomotion and regulation of activity [15]. Knowing the structure of the protein is of utmost importance as it enables researchers to predict its function and to design compounds such as drugs that can be used to manipulate or influence its characteristics and/or function.

1.4 Methods for predicting protein structure

Several techniques exist for predicting the 3D structures of proteins. These can be broadly grouped into experimental and computational techniques.

1.4.1 Experimental techniques for predicting protein structures

Experimentally determined 3D structures are of great importance since understanding their structure helps in the elucidation of their molecular function, the physical origin of protein folding, and stability. Furthermore, experimental structures form the basis of computational approaches.

3D macromolecular structures were initially solved using crystallography. Currently, several techniques are available for determining 3D macromolecular structures including X-ray crystallography, nuclear magnetic resonance, electron microscopy, powder diffraction and

fibre diffraction. The main protein structure prediction techniques are discussed in the subsequent sections.

1.4.1.1 Macromolecular crystallography

Macromolecular crystallography (MX) is the most preferred technique for determining 3D structures of proteins and biological macromolecules. This technique is used to obtain the 3D molecular structure from a crystal at near-atomic resolution. The first molecular structure solved using X-ray crystallography was myoglobin in 1958 [16] followed shortly by a discovery that the folding of helices in myoglobin was similar to that of the subunits of haemoglobin [17].

The quality of the protein crystals is the limiting factor for application of protein crystals in structure determination. A purified sample at high resolution is induced out of a solution to form crystals under the right conditions or precipitation is allowed to occur [18]. Protein crystal growth is affected by several factors including varying the initial protein concentration, precipitating agent concentration, pH, temperature and presence or absence of co-factors, co-solvents and impurities [19,20]. Initially, crystallization experiments were performed on a trial-and-error basis since the aim was to consider all the variables above to yield high-quality crystals. However, this process may result to either none, precipitates, microcrystals, or a few very tiny crystals generated. The size of the crystals can be improved using various methods including seeding, altering protein concentration, and temperature. The resultant crystals are required to be least 0.1 mm in the longest dimension for a substantial volume of the crystal lattice to be exposed to the X-ray beam. The crystal is then mounted into an X-ray beam which is generated from accelerating electrons in a synchrotron storage ring or from electrons striking a copper anode [18].

X-ray diffraction analysis is performed after obtaining the desired crystal size and ascertaining the presence of the macromolecular subunit of interest in the crystals. The distance between the crystal and the detector is calculated and adjusted up to a maximum resolution of 1.5 – 3.0 Å before the collection of diffraction spots. The diffraction image becomes weak at high resolution thus a compromise has to be made between reduced quality of diffraction and increased resolution [18]. In addition, the unit cell dimensions, crystal system and space group have to be determined during diffraction analysis. A unit cell is the smallest repeating portion of the crystal lattice that shows the 3D pattern of the crystal. It helps in establishing the spacing of spots on the diffraction image. The crystal system is defined by the crystal shape whereas the space group is specified by symmetry of the diffraction pattern. Data processing is carried out on the first diffraction image. At first, data processing involves accurately determining the crystal system, unit cell dimensions and the crystal orientation in the beam. Auto-indexing and intensity measurement is then performed on each spot on the image using a program such as DENZO[21]. The phase angle cannot be determined directly resulting in a problem. This phase problem can be solved by isomorphous method or molecular replacement method [18].

The structure factors can be calculated using the fast Fourier transform method [22] according to the amplitudes and phases, which results in an electron density map. This map forms the 3D contours from which the protein structure is built. In case the quality of the density map is low, it can be improved by refinement and then building of the model is performed. The output structure is saved in a Protein Data Bank (PDB) format and the quality is expressed as an R factor [18].

Production of diffraction quality crystals suitable for high resolution data collection limits determination of accurate macromolecular structures. Protein or macromolecular crystallography process can be automated to increase the possibility of structure determination in the shortest time possible and improve the quality of the models [23–26]. The automation

process sometimes reduces X-ray structure quality and introduces errors as human intuition and reasoning is not considered. The curators of PDB database have introduced validation procedures to deposit structures over the years [27].

1.4.1.2 Nuclear magnetic resonance spectroscopy

Nuclear magnetic resonance spectroscopy (NMR) was implemented as a technique to determine macromolecules in the early 1980s [28]. It is now a well-established structure prediction method that is able to give high-quality 3D structures at atomic resolution. NMR spectroscopy is a spectroscopic technique used for structural studies of proteins both in solution and solid state. Solution- and solid-state NMR spectroscopy are highly effective in characterizing supramolecular protein assemblies [29,30].

Several technical advances in the field of NMR including cryogenic probe technology, increase in magnetic field strength, reduced dimensionality triple resonance NMR, minimal data collection strategies, and automated data analysis lead to high throughput protein structure determination [31]. There are four principal components in the NMR method: the nuclear Overhauser effect (NOE), sequence-specific assignment of several NMR peaks, computational tools, and multidimensional NMR techniques [32].

NMR has several applications including structure determination of proteins with disordered regions [33–36], structural characterization of multi-domain proteins in the absence of crystal packing forces [37], and structure determination of proteins without corresponding X-ray crystal structures. NMR allows characterization of dynamic regions in the proteins and the process requires less data collection time compared with X-ray crystallography given that NMR samples are prepared in solution. However, use of NMR for structural prediction has several downsides including poor sensitivity, can only be used for macromolecules with molecular size limit of about 30 kDa, and the protein sample stability is limited [31].

Improvements to alleviate these challenges are underway. For example, methods to overcome the size limitation of solution NMR spectroscopy have been developed. These methods used for study of large proteins or complexes include methyl transverse relaxation optimized spectroscopy (TROSY) [38], and freezing rotational diffusion of protein solutions at low temperature and high viscosity (FROSTY), or sedimented solute NMR spectroscopy [30,39].

1.4.1.3 Electron microscopy

Although X-ray crystallography and NMR have led to elucidation of an enormous number of structural atomic models, the processes are quite challenging. The electron microscopy (EM) method was thus developed to study non-living materials and to provide images of macromolecules as detailed as those provided by X-ray crystallography [40–42]. The first structure to be solved by EM was reported in 1975 after two-dimensional crystals were obtained from the protein but were unsuitable for X-ray diffraction [43]. A major breakthrough of EM technology was achieved in the 1980s when liquid ethane was used for preparation of samples hence the term ‘cryo’ [44].

Cryogenic electron microscopy (cryo-EM) technique was introduced to foster the acceleration process of protein structure prediction [45]. This technique uses a photograph of frozen biological molecules to determine protein structure [46]. Moreover, the technique is limited by some challenges including low signal-to-noise ratio (poor contrast of the image) and image quality degradation owing to movement of the electron beam [47,48]. Advances in cryo-EM have resulted in the production of high-quality structures with high resolutions since protein suspensions are frozen on a transmission electron microscope (TEM) support grid. The most important step is the formation of the vitreous ice layer which preserves the target in a near-native state. Cryo-EM approach uses low-electron dose conditions to reduce radiation damage

of frozen protein samples before capturing of the images. Moreover, automation of this process has led to the production of high-resolution structures [45,49].

1.4.1.4 Storage of experimentally determined protein structures

One of the major achievements in structural biology was the development of the Protein Data Bank (PDB) database in 1971 at Brookhaven National Laboratory [50]. PDB is an archive that provides free access to experimentally determined macromolecules. The number of deposited structures was initially seven and has drastically increased due to advanced technology in all aspects including data sharing, MX, and NMR techniques. Increased influx of structural data driven by the need to understand the biological functions of macromolecules, and the initiative of structural genomics has led to improved strategies for data acquisition, validation, organization, and distribution [51].

PDB database is jointly managed by the Worldwide Protein Data Bank (wwPDB) consortium [52,53]. The members of the consortium include the US Research Collaboratory for Structural Bioinformatics Protein Data Bank (RCSB PDB; *rcsb.org*) [51], Protein Data Bank Japan (PDBj; *pdbj.org*) [54], Protein Data Bank in Europe (PDBe; *pdbe.org*) [55], and BioMagResBank (BMRB; *www.bmrb.wisc.edu*) [56]. RCSB PDB is the core archive that houses 3D structural models of proteins, DNA/RNA, experimental data, metadata from macromolecular crystallography, and macromolecular complexes with metals and small molecules. BioMagResBank (BMRB) core archive houses related metadata from NMR whereas another partnership with the Electron Microscopy Data Bank (EMDB; *emdb-empiar.org*), houses related metadata from EM and Electron Tomography (ET) [57].

As of 11th February 2022, there were 186,934 protein structures in the PDB (https://www.rcsb.org/stats/summary) [58]. Most of the 3D structures deposited are determined by MX (87.5%), whereas the rest are remainder determined by NMR (7.3%), EM (4.8%), and

other techniques (0.1%). The overall trend in the amount of data determined experimentally was consistent but in 2016, EM surpassed NMR as the second most popular technique (Fig 1.3).

Despite the growing number of high-quality structures deposited in the PDB [51], some have errors and present with inconsistencies [59–62]. These errors in predicted structural conformation can result in inaccurate findings in structure-based processes that rely on this process for example drug design and data-mining studies. Efforts to validate and perform quality control of macromolecular structure models are underway [57,63].

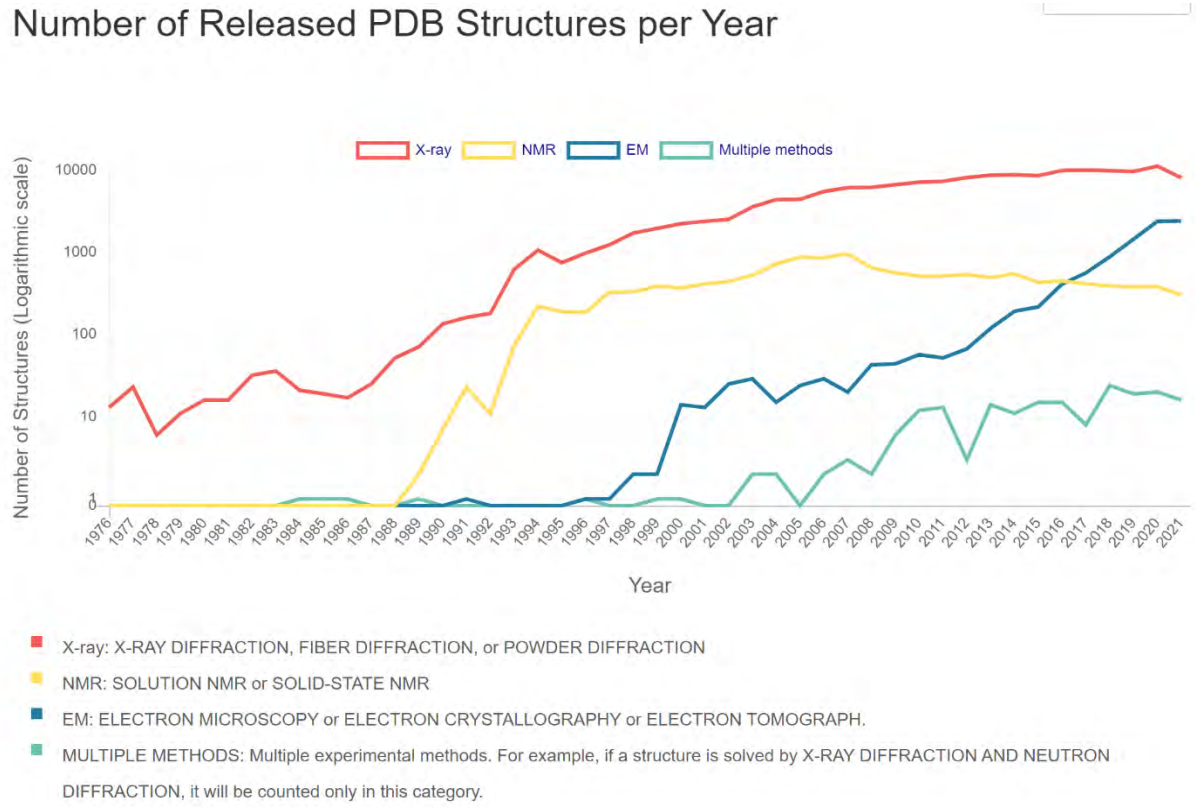


Fig 1.3. Number of PDB structures released annually. Structures released per year are coloured by experimental technique (X-ray – red, NMR – yellow, EM – blue, Multiple methods – green). The highest number of structures in the PDB are those determined by X-ray followed by NMR, EM, and the use of multiple methods. EM technique overtook the NMR technique in 2016 becoming the second most common protein structure prediction technique. Image sourced from (https://www.rcsb.org/stats/summary) [58] on 05/11/2021.

The 3D macromolecules are stored as flat files including the PDB file format [64], the PDB data exchange dictionary or macromolecular crystallographic information framework

(PBDx/mmCIF) format [65,66], and the PDBML/XML format [67]. The size and complexity of macromolecules have increased over the years leading to limited use of legacy PDB file formats. This limitation was addressed by embracing a new format called extensible PDBx/mmCIF data framework though the worldwide Protein Data Bank (wwPDB) presents a PDB file format for user convenience [57,63].

The ever-growing archive, size, and complexity of the deposited structures have posed challenges to the wwPDB consortium. These challenges have been addressed by developing and continuously improving the OneDep deposition/validation/biocuration system [68]. The wwPDB is developing strategies to enable depositors to make corrections to the PDB files, improve validation reports and provide the official digital object identifier (DOI) for each PDB structure [57,63].

1.4.2 Computational techniques for predicting protein structures

Although advances in experimental protein structure prediction techniques have increased the number of protein 3D structures deposited in the PDB [51], these are still few in comparison to the number of protein sequences deposited in Universal Protein Resource Knowledgebase (UniProtKB) [69]. To shorten this gap, computational approaches to modeling protein structures were developed and have been in use since the end of the 20th century [70].

Computational approaches use the amino acid sequences to predict the 3D protein structures. In 1973, Anfinsen demonstrated that all the information needed by the polypeptide chain to fold into a 3D structure is encoded in its amino acid sequence [70]. Physically, the amino acid sequence determines the protein's basic molecular composition, whereas the native structure corresponds to the most stable (lowest free energy) conformation. Approaches used to computationally model protein structures are broadly classified into two: 1) template-based

modeling and 2) template-free modeling approaches. Some composite approaches, however, combine the two strategies.

1.4.2.1 Template-based modeling

Template-based modeling (TBM) involves the use of a structure as an evolutionarily related experimental protein to generate an unknown 3D structure based on a protein sequence. This technique depends on the principle that the structure of a protein is more conserved compared with its amino acid sequence [71,72]. Illergård and colleagues reported that the protein structure is ten times more conserved relative to its sequence [72]. In addition, protein evolution studies on sequence-based inference of homology indicate that proteins have similar folds [73] with very few exceptions [74]. TBM consists of four sequential steps include: 1) searching for suitable templates (known structures) related to the sequence (target), 2) aligning the target sequence to the template structure, 3) modeling the target sequence using spatial restraints from templates concurrently with loop refinement, and 4) evaluating the model [75]. These steps are repeated if the quality of the model is not satisfactory, until no improvement in the model is attained. Each of these steps can be done using different programs, techniques, and web servers [76]. This technique of modeling protein structures is also known as homology or comparative modeling.

Template-based modeling has a wide array of applications including structure-based drug design, identification of active and binding site [77], insight into binding mechanisms [78], modeling of substrate specificity [79], analysis of mutations [80,81], and protein-protein docking simulations [82] among others [83]. Despite the advantages of using template-based modeling, the poor choice of templates (known 3D protein structures), inaccurate alignments of target (unknown protein structure) sequences to template sequences particularly low sequence identity and incorrect loop folding results into inaccurate structures [84].

The most reliable and successful protein structure prediction method is template-based modeling [85,86]. The method is, however, less effective when there are no close enough homologs. Several template-based protein structure modeling web servers have been developed. These ease the cumbersome process of installing and using standalone modeling programs. They include PRIMO [87], SWISS-MODEL [88], and HHpred [89] among others. This method is discussed in detail in Chapter 2.

1.4.2.2 Template-free modeling

Template-free modeling techniques are particularly helpful when no structural analogs to the target protein exist in the PDB [51], or when the target protein is partially covered by the template structure. Occasionally, when template-based modeling is used, some regions – including insertions and loops of the target sequence are not modeled since they are not represented in the template structure. In such cases, template-free methods become the best option to add those missing regions to build complete protein models. This structure prediction built from scratch using the amino acid sequence to predict the most stable protein spatial arrangement with the lowest free energy. It is assumed that a protein sequence folds to a native conformation that is near the global free-energy minimum [90].

These approaches are broadly classified according to the methodologies employed in modeling. They include physics-based approaches, fragment-based approaches, secondary structural elements-based approaches, and deep learning-based approaches. To date, the fragment-based approach is the most accurate strategy for template-free protein structure prediction [91]. This approach uses short, contiguous backbone fragments usually 3-15 residues in length extracted from proteins with known structures to build models [92,93]. Energy functions and coarse-grained molecular representations used in fragment assembly do not accurately select native-like models and the atomistic details required by some applications such as structure-based

drug design are not included in the model. Simulation approaches to move the first model closer to the native structure are used. This process is called model refinement and it involves an accurate energy function along with a strategy to explore the conformation space. The most successful conformation strategy applied is the molecular dynamics simulation [94].

Some of the major challenges faced by the *ab initio* method are the huge conformational space to be searched given the dynamic nature of proteins, and the low accuracy of the protein models [95]. Servers for free template modeling have been developed, and include Rosetta [96–98], Phyre2 [99], and I-TASSER [100], among others. In CASP13, template-free modeling showed the greatest improvement in model accuracy. This progress was driven using deep learning techniques to predict 3D contacts as well as inter-residue distances owing to availability of an adequate known number of sequences for the protein family [101]. Whereas template-free modeling improved in CASP13, it was not the case in CASP14. In CASP14 the models built using homologous structure information were slightly more accurate [102].

1.4.2.3 Hybrid approaches to modeling protein structures

Advancements in computational protein structure prediction techniques now allow for the amalgamation of template-based, template-free approaches and artificial intelligence to predict of protein structures. These approaches include Bhageerath [103,104] and AlphaFold [105]. As of now AlphaFold is the most recent development in protein structure prediction. AlphaFold is a computational approach whose accuracy has been deemed close to that of experimental structure accuracy [105]. It uses neural networks and training procedures based on evolutionary, geometric, and physical constraints of protein structures to predict single protein structures. AlphaFold had the highest accuracy compared to the other participating methods in CASP13 [101] and CASP14 [102]. This group participated as AlphaFold and AlphaFold2 (AF2) in CASP13 and CASP14 respectively. Whereas AlphaFold2 had the highest accurate

models in CASP14, the other research groups performed better than AlphaFold in CASP13. This was because AlphaFold2 did not submit server models [102]. The AF2 neural network models were adapted to predict multimeric protein complex structures. This method was referred to as AF2Complex [106]. It does not require paired multiple sequence alignments (MSAs) [106].

1.5 Assessment of protein structure prediction

After performing protein structure prediction, it is important to assess the accuracy of these structures. Users of these predicted models need to evaluate them and select the best for their studies. One of the means that provide an independent mechanism to assess protein structure prediction techniques is Critical Assessment of Protein Structure Prediction (CASP).

1.5.1 Critical Assessment of Protein Structure Prediction (CASP)

Critical assessment of protein structure prediction (CASP) [101,107–118] is a platform that tests and assesses the current methods of modeling protein structure from the amino acid sequence to establish their capabilities, progress, and specific bottlenecks. The first assessment meeting was held in 1994 and since then, these experiments take place biannually. The core principle is *double-blinded testing* whereby the participants are unaware of (blinded to) the solutions to the modeling challenges, and their identity is unknown to the assessors. Information on targets whose structures are yet to be solved is collected from the experimental community and shared with the prediction community through the prediction center (https://predictioncenter.org/). Participants submit their predictions before the experimental data is released. The models are then evaluated by independent assessors using a range of numerical criteria comparing them to newly determined experimental structures. Different domains are tested, including template-based modeling, template-free modeling, contact

prediction, and refinement protocols. The performance of all the participating groups is summarized and reported at the CASP meeting.

CASP assesses the progress that has been made and reveals the areas where future efforts should be focused. A total of fourteen experiments have been performed so far with the last one conducted in 2020. The third last experiment (CASP12) resulted in substantial progress mainly in template-based modeling, contact prediction, template-free modeling, and estimating the model accuracy [110].

CHAPTER TWO

A REVIEW OF HOMOLOGY MODELING PROCEDURES AND TOOLS CURRENTLY IN USE

Chapter overview

Given the widening gap between known protein sequences and their corresponding predicted structures, there is interest in shortening this gap to harness the benefits of knowing the structural conformation of proteins. Computational approaches now play a pivotal role in addressing this problem. One of such approaches is template-based modeling also known as homology modeling. Template-based modeling is one of the most effective and widely used computational approaches to computationally predict multimeric proteins. The approach is used in cases where protein sequence(s) with unknown protein structures have homologues with known protein structures. Template-based modeling is either performed using human-expert guided standalone protein modeling engines such as MODELLER [119], or automated modeling servers such as PRIMO [87], SWISS-MODEL [88], HHpred [89], Phyre2 [99], and I-TASSER [100]. Although standalone protein modeling engines are still widely used by experienced bioinformaticians, their protocols are not easy to follow for novice users.

In this chapter, the homology modeling approach is reviewed in detail because part 1 of this work focuses on the development of web server that relies on homology modeling to build multimeric proteins for unknown protein structures.

2.1 Homology modeling steps

The homology modeling process involves four sequential steps (Fig 2.1) including template identification, target-template alignment, model building and model evaluation [75].

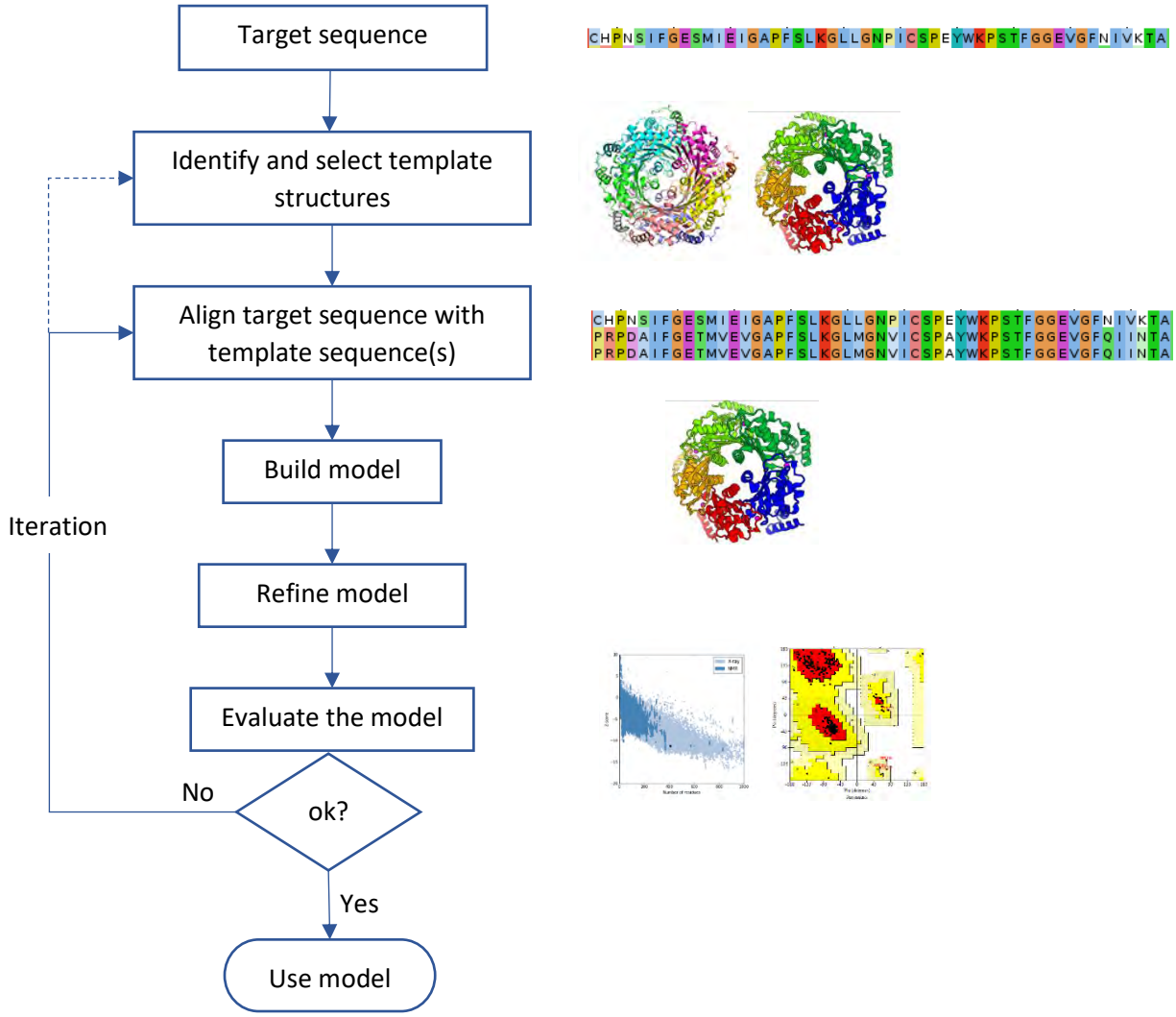


Fig 2.1. A schematic representation of template-based protein structure modeling. Once the target protein sequence with an unknown structure is provided, the most suitable homologous 3D protein structure is chosen as the template for modeling. Alignment of the target sequence to the template sequence(s) is performed, which is used together with the template structure to build the model. The model is refined and evaluated. Iteratively realignment and rebuilding of the model are performed where necessary. The protein structure and alignment images were prepared using the protein viewer (PV) (https://biasmv.github.io/pv).

2.1.1 Template identification

The most critical step in modeling is the selection of templates [75]. Templates are known protein structures that are homologous to the protein with an unknown structure (target). Templates are retrieved from the PDB [51] using the target sequence as the query sequence to identify known homologous protein structures. Searching for templates can be based on either sequence identity between the target and template or physicochemical properties of amino

acids such as solvent accessibility and secondary structure. Homolog search can be done using a single target sequence (fast-all (FASTA) [120], Basic Local Alignment Search Tool (BLAST) [121]) or a profile formed from multiple sequences (Position-Specific Iterated BLAST (PSI-BLAST) [122], HMMER [123], HHpred [89]). These programs are more effective if the target and template proteins share at least 40% sequence identity [124] which is true but with a few exceptions [75]. Sequence identity is defined as the percentage of residues that are identical between paired aligned sequences. Below the 'safe zone' (sequence identity of 35% or higher can be safely regarded as having close homology), is the twilight zone which describes a threshold (between 20 – 35% sequence identity) at which it is unclear whether two proteins are homologous [124].

The Basic Local Alignment Search Tool (BLAST) [121] is used to identify potential templates. It uses substitution matrices to positively score conserved replacements and penalise gaps and poorly substituted residues during sequence comparison. BLAST uses heuristics to assess local similarity using a word-based approach. Fixed short segments (words) from the query sequence are aligned against the sequences in the database, and segment pairs are scored. If the segment pair score is greater or equal to a certain set threshold, it is extended until it falls below the threshold. As a result, the maximal segment pair (MSP) is calculated to give a measure of local similarity of any pair of sequences. MSP is the highest scoring region of two identical length segments of two sequences. This enables BLAST to quickly search and return results from a huge database. BLAST is faster than FASTA but only generates gap-free local alignments hence the release of a gapped BLAST version and the Position-Specific Iterated BLAST (PSI-BLAST) in 1997 [122]. The gapped BLAST algorithm increased the speed of the initial database search since only one ungapped alignment was considered. PSI-BLAST uses the significant alignments from BLAST to construct a position-specific score matrix then uses the matrix to search the database.

Introduction of PSI-BLAST [122] and hidden Markov models (HMMs) [125–127] significantly improved efficacy of template search [128]. These methods use the target sequence to create a profile-based alignment from homologous sequences with known protein structures. In addition, profile-profile methods were introduced where a profile created from the target sequence is compared with a profile created from all known structures thus greatly improving sensitivity of the search and detection of remote homologs [129,130]. Among the programs that use HMMs is HHpred. HHpred is used for remote protein homology detection and structure prediction [89]. It uses pairwise alignment of profile hidden Markov models (HMM) hence maximizing sensitivity while ensuring increased speed and enhanced alignment quality. The HMM is compared against all the HMMs in precalculated databases which also include secondary structure information from either PSIPRED [131] or DSSP [132]. The database search is done by HHsearch which uses HMM-HMM comparison employing position-specific gap penalties [133].

2.2.2 Target-template sequence alignment

After the template(s) are identified, sequences of the target and template proteins are realigned as a next step, since the alignment generated during template search might not be optimal for template-based modeling [76]. The target sequence is aligned with the template sequence using refined alignment algorithms embedded in programs which include; Clustal [134], Multiple Sequence Comparison by Log-Expectation (MUSCLE) [135], Multiple Alignment using Fast Fourier Transform (MAFFT) [136], Tree-based Consistency Objective Function for alignment Evaluation (T-COFFEE) [137] and Profile Multiple Alignment with Local Structures and 3D constraints (PROMALS3D) [138]. These programs align sequences either globally (across the entire length) or locally (specific regions) with some incorporating structural information in the alignment. More details on these programs are given in Section 2.3.

Closely related proteins with over 40% sequence identity have more accurate alignments compared with those in the twilight zone [124]. Alignments with low sequence identity (below 35%) might lead to misalignment errors and might have several gaps [124]. The alignment step is crucial in template-based modeling process since any inaccuracies affect the model quality. The alignment can be manually edited, if necessary to improve its quality. Sequence alignment is not only used in modeling protein structures but also lays a foundation for modern bioinformatic studies such as enabling biologists to study conserved regions and reconstruct phylogenetic relationships through evolution [139].

2.2.3 Model building

This step follows the target-template alignment step and uses several methods to construct a 3D model for the target protein. Model building methods include modeling by rigid-body assembly [140,141], by segment matching [142–144] and by satisfaction of spatial restraints [145–148]. These methods produce models of relatively similar accuracy. Notably, factors such as template selection and alignment accuracy that significantly affect model accuracy [76] especially if the sequence identity to templates is less than 35% [124]. Some of the programs that implement these methods include the SEGMOD program [144] that models protein structures by segment matching, the MODELLER program [119] that models protein structures by satisfaction of spatial restraints and the COMPOSER program [149], as well the NEST [150] program for modeling of protein structures through assembly of rigid bodies. These modeling programs have similar performance except for MODELLER which is more effective compared with the other programs [151].

2.2.4 Model evaluation

The final model should be assessed for the correctness of the overall structure, errors over localized regions and stereochemical properties [60]. Assessment of models is mainly

performed using structural comparison methods and/or methods with no reference to the structure. Structure comparison methods use two known protein structures to assess the model's accuracy. These methods include the RMSD [152] and the global distance test (GDT) [153]. The most commonly used method is the RMSD metric which determines the mean distance between corresponding atoms after two structures have been superimposed [154].

Model assessment can also be performed without reference to the structure and these methods include statistical potentials [155,156] and physics-based energy calculations [157]. The two methods estimate energy of the model(s). Statistical potential calculations are based on residue-residue contact frequencies among known protein structures in the PDB [51]. The most popular statistical potential calculation methods include z-DOPE [158] and ProSA [159]. Physics-based calculations are performed using a molecular mechanics force field which aims to capture interatomic physical interactions responsible for protein stability in solution [160]. The alignment and modeling steps are repeated, if necessary, until the model quality cannot be further improved. More details on these programs are given in Section 2.4.

2.3 Alignment programs

Several alignment programs have been developed to ease the burden of sequence alignment some of which are discussed below. Multiple sequence alignments (MSAs) form the foundation for most data analyses including structural and functional analysis. MSAs play an important role in elucidating molecular evolution by describing the relationship between nucleotide or protein sequences. Sequence alignments can be pairwise or multiple. These programs can perform either MSAs or pairwise sequence alignments.

2.3.1 Clustal programs

A series of Clustal programs have been developed over the years [134] with the first program reported in 1988 [161]. These Clustal series provide robust and portable programs that rapidly

perform accurate biologically relevant alignments [134]. Clustal tools perform global alignment of protein and nucleic sequences as well as construction of phylogenetic trees. Multiple pairwise alignments are built progressively following the branching order of a phylogenetic tree. This process is more effective when the sequences are closely related because very important information is captured by the time very divergent sequences are added to the alignment. In case all the sequences are highly divergent (< 25 – 30% sequence identity), then the progressive approach becomes less efficient [162]. Guide trees were initially constructed from multiple alignments using the Unweighted Pair Grouping Method with Arithmetic-mean (UPGMA) algorithm developed by Sneath and Sokal [161]. Subsequently, the Neighbour-Joining method (NJ) [163] was introduced for a robust and accurate alignment owing to optimised branch length and weight calculations. The choice of weight matrix varies depending on the alignment stage to down-weight very similar sequences and up-weight sequences that are divergent. Position-specific gap penalties introduce new gaps in loop regions rather than regions that form secondary structure elements to improve accuracy of the alignment [162].

The UPGMA algorithm for constructing guide trees was further improved as an alternative to the NJ algorithm. UPGMA is extremely fast on large datasets, however, it clusters long branches together in case of differences in evolutionary rates. In addition, an iterative alignment option was introduced to increase alignment accuracy. A sequence is removed from the alignment at every iteration step and the alignment is realigned as the weighted sum of pairs (WSP) score is adjusted. The alignment is retained when the WSP score reduces. Iteration can be done either at each step in the progressive alignment or at the final alignment [164]. Currently, the latest version of the Clustal series is the Clustal Omega program used to align large numbers of sequences. It uses mBED algorithm [165] and HHalign [166] method to generate guide trees and accurate alignments, respectively [167].

2.3.2 MUSCLE program

Multiple Sequence Comparison by Log-Expectation (MUSCLE) uses pairwise profile alignment to generate a progressive alignment which is used during the refinement stage [135]. Two progressive alignments are generated, one from unaligned pair of sequences using *k*mer distance and the second from aligned sequences using the Kimura distance. A *k*mer also known as a word or *k*-tuple is a subsequence of length *k* that occurs in related sequences. Matrices produced from these distances are used in the UPGMA [161] method to construct binary trees with the Kimura distance tree being more accurate. The Kimura distance tree is refined until an improved SP score is attained for the generated new multiple alignments [135]. MUSCLE program produces multiple sequence alignments that are on average as accurate as those generated by best current alignment programs [135,168].

2.3.3 MAFFT program

Multiple Alignment using Fast Fourier Transform (MAFFT) program was developed to perform rapid calculation of large scale MSAs to reduce the computational time with comparable accuracy [169]. This tool is used for multiple sequence alignment based on the fast Fourier transform (FFT) algorithm. This algorithm converts amino acid sequence to a sequence of vectors including the volume and polarity between each amino acid pair. The FFT algorithm used in MAFFT is 100 times faster compared with algorithms used in other MSA programs. The progressive method and iterative refinement method are also implemented in MAFFT program with a few modifications. A guide tree is constructed using UPGMA method [161] and sequences are progressively aligned following this guide tree [169].

Following the initial implementation of MAFFT, new features have been added including the PartTree algorithm [170] for improved scalability of progressive alignment and the Four-way consistency objective function for improved accuracy of the structural alignment of ncRNAs

[171]. Further improvements were incorporated in MAFFT in 2013 including addition of unaligned sequences to an existing MSA as a backbone, parallel processing and adjusting the direction of deoxyribonucleic acid (DNA) sequences [136].

The MAFFT-DASH (Database of Aligned Structural Homologs) web server was developed in 2019. This incorporates structural alignments in MAFFT MSA. MAFFT-DASH has a higher performance compared with the standard MAFFT tool when aligning remote homologs with a 10-20% improvement. Notably, tools that include structural information in the alignment have higher accuracy relative to tools that use purely sequence-based methods [172].

2.3.4 T-COFFEE program

The Tree-based Consistency Objective Function for alignment Evaluation (T-COFFEE) program combines both global and local multiple alignments with a progressive alignment strategy using the NJ method [163] to construct a guide tree [137]. It computes two primary libraries for each pair of sequences using Needleman & Wunsch [173] and the other using Lalign [174] for global and local pairwise alignment, respectively. The pairwise sequences in each library are weighted according to the sequence identity. The alignments are then combined into one library using dynamic programming [137].

Quality of MSAs generated by T-COFFEE program was improved by incorporating structural information in the alignment of protein sequences and this version is called 3D-Coffee [175]. In 3D-Coffee, structures associated with each sequence to be used in the MSA were manually added by the user making it cumbersome. Therefore, the Expresso version was developed which rapidly and automatically identifies suitable structural templates using BLAST search against the PDB database [176].

2.3.5 PROMALS3D program

Profile Multiple Alignment with Local Structures and 3D constraints (PROMALS3D) program uses sequence and structure-based information to align sequences [138]. It uses a progressive method named PROMALS [177] to cluster and align similar sequences to select representatives. PROMALS3D uses PSI-BLAST [122] to search for additional homologs and PSIPRED [131] to predict secondary structure to create a hidden Markov model (HMM) [123] of profile-profile alignment. Structural and sequence constraints are derived from profiles of sequence representatives which are later combined to generate the multiple sequence alignment. PROMALS3D produces high-quality alignments of distantly related sequences [138].

2.4 Model evaluation programs

Assessing the quality of protein models is very vital in protein structure prediction as the accuracy of models impacts the areas where they are applied such as structure-based drug design. Protein structures determined either experimentally or computationally are prone to errors [59]. Determination of these errors is computationally expensive thus indirect methods are used to assess the accuracy of some parts or the whole model by either checking the stereochemistry or geometrical properties independent of the experimental data [178,179].

Several programs have been developed to assess the correctness of experimental and computationally designed structures. Some of these programs are described below.

2.4.1 VERIFY3D

Verify3D program assesses compatibility of the protein model to its amino acid sequence using a 3D profile [180,181]. Every residue position in the sequence of the protein model is represented by its environment and a row of 20 numbers (representing amino acids) making up the profile. The environment is made up of three features including the buried area of the

residue, the side-chain area covered by polar atoms and the local secondary structure. The 3D profile score is computed as a sum of all residue positions of the 3D-1D scores for each amino acid protein sequence. An improperly modeled segment in the protein model is identified by assessing the profile score along with a sliding window scan of 21 residues as shown by low scoring regions in a profile window plot [181].

2.4.2 ProSA

Protein Structure Analysis (ProSA) uses Boltzmann's principle [182] to extract force fields from known 3D structures in the PDB. These force fields are extracted in the form of the potential of the mean force field which is then used to determine incorrect protein structures with high energies/z-scores [60,182].

ProSA-web is an interactive web tool used to detect potential errors in 3D structures of proteins. ProSA-web displays two plots one showing the overall quality of scores of the input protein structure in comparison with the scores for all the available experimentally determined protein structures in the PDB [51]. The second plot shows the local quality energies scores for the input structure [159].

2.4.3 QMEAN

Qualitative Model Energy ANalysis (QMEAN) [183] uses approaches/methods which rely on scoring functions based on: i) analysis of single models based on evolutionary or physiochemical properties and ii) information from an ensemble of models for a given sequence. The user chooses either QMEAN [184] which calculates both global and local quality estimates to select the best model and also highlights the problematic incorrect regions or MEANclust [185] which estimates the quality and local conformations of multiple models.

A quality score is calculated for each model and these models are ranked or assessed according to the calculated quality scores ranging from 0 to 1. QMEANclust uses the best model scored by QMEAN to perform the cluster analysis [183]. A model quality score known as the QMEAN Z-score was introduced in 2011 [186]. This quality score method relies on the QMEAN function [183,184] to determine the quality of protein models as well as to detect errors in experimental structures. QMEAN Z-score can be calculated for both single protein chains and biological assemblies whereby low-quality models have strongly negative values [186].

2.4.4 z-DOPE

z-DOPE is a normalized Discrete Optimized Protein Energy (DOPE). DOPE is a statistical potential that is dependent on atomic distance and is calculated from a native protein structure [158]. It is based on an improved physical reference state and takes into account the finite and spherical shape of the native protein structure [158,187]. Negative z-DOPE scores indicate better models.

2.4.5 PROCHECK

PROCHECK runs five programs to compute and assess the stereochemical properties of the protein model and validates them against the ideal values obtained from the PDB [51]. The program checks stereochemistry of the protein structure including those in existence, those in the process of being solved and those modeled from known structures. Plots generated by PROCHECK give an overall assessment of the structure quality highlighting problematic regions. Structures are usually checked for bad contacts, inspected graphically, and through use of Ramachandran plot to show residues that lie in the ‘disallowed’ regions (sterically disallowed regions for all amino acids except glycine) [188].

2.5 Existing automated protein modeling tools

Several fully or partially automated template-based modeling web servers have been developed to circumvent the cumbersome processes of manually installing software or using standalone programs. Some of these tools use *ab initio* modeling methods, others use comparative modeling whereas others use a combination of the two modeling approaches. Modeling servers include PRIMO [87], SWISS-MODEL [88], HHpred [89], Phyre2 [99], and I-TASSER [100]. Some of these servers are used for building monomeric proteins whereas others are applied in building multimeric proteins. Web servers used for building multimeric proteins are discussed below. Although PRIMO web server models protein monomers, it is discussed below since it was the basis of this work.

2.5.1 I-TASSER

The iterative threading assembly refinement (I-TASSER) [189] server uses an iterative hierarchical protein structure modeling approach called threading assembly refinement (TASSER) [190]. I-TASSER algorithm was improved through introduction of a new knowledge-based force field using neural network hydrophobic potential, a new simple profile-profile alignment (PPA) threading approach, and a two-step iterative refinement approach [189].

I-TASSER uses consensus constraints from templates for modeling of proteins. The I-TASSER server was tested using CASP7 [191] experiments and it was better than the original version of TASSER. All the different I-TASSER versions have been tested using CASP 7 to CASP 11 data [192]. The findings show no clear improvement of the LOMETS [193] threading programs over PSI-BLAST [122]; however, they exhibit progress in structural refinement. This improvement is attributed to integration of template-based modeling and physics-based *ab initio* folding simulations [192].

In the CASP13 experiment, the "Zhang-Server" pipeline was renamed to contact-guided I-TASSER (C-I-TASSER). This version has an iterative multiple sequence alignment (MSA) program which improved accuracy of contact-map prediction, extended NeBcon [194] capability, and led to a novel contact potential term [195].

2.5.2 SWISS-MODEL

SWISS-MODEL server is an automated server for protein structure homology modeling. It has been in use for over 20 years [196]. This server has a user-friendly web interface that allows users to search for templates against the SWISS-MODEL Template Library (SMTL). This library collects information derived from experimental structures curated in the PDB [51]. Models are mainly generated using ProMod-II [197]. Notably, MODELLER [119] is used if the model is not satisfactory and their quality is assessed using the local composite scoring function, QMEAN [186]. SWISS-MODEL server was extended to model oligomeric structures with evolutionary ligands and the accuracy of the models is evaluated by the Continuous Automated Model EvaluatiOn (CAMEO) project [198].

2.5.3 Rosetta

The Rosetta tool [96,199] was initially developed for *ab initio* protein structure prediction and protein folding [200]. It was later expanded to offer molecular docking, protein design, template-based modeling, and determination of protein structures from experimental NMR data [201]. Rosetta uses Monte Carlo simulation method to assemble fragments into protein-like structures. This server performed better relative to other *ab initio* protein structure prediction servers in CASP11 [202].

2.5.4 PRotein Interactive MOdeling (PRIMO)

Between 2016 and 2017 our research group within the Research Unit on Bioinformatics at Rhodes University in South Africa (RUBi) developed PRIMO - PRotein Interactive

Modeling (PRIMO) to perform modeling of monomeric proteins [87]. PRIMO provides a user-friendly interface that allows users to alter parameters during the modeling process. It also provides the functionality to model ligands and ions into the target proteins identified from template PDB files. The PRIMO modeling process involves three steps including template identification and selection, target-template sequence alignment and protein modeling and evaluation. PRIMO is registered with (Continuous Automated Model EvaluatiOn) project for continuous evaluation [87].

2.6 Research motivation

The number of protein structures and sequences deposited in the PDB [51] and UniProtKB [69], respectively, keeps growing exponentially. Currently, close to 190 million protein sequences have been deposited in the UniProtKB database [69] and there are approximately 180 thousand structures in the PDB [51]. The increase in protein structure prediction is not at par with high-throughput sequencing technology. Consequently, there is a significant sequence-structure gap.

Limitations of experimental structure determination techniques necessitated the development of computational approaches to predict protein structures to fill the gap. The most reliable computational protein structure prediction approach is template-based modeling [85,86]. Template-based modeling can be performed using either human-expert guided standalone protein modeling engines or automated modeling servers. Although standalone protein modeling engines are still widely used by experienced bioinformaticians, they present challenges to novice users. Automated modeling servers are being developed to address some of these challenges. Most of these web servers in existence, however, provide limited user involvement during the modeling process with a few supporting multimeric protein modeling.

Information about the whole protein complex structure is important to understand its functionality since most essential interactions happen at the interfaces between one or more proteins [203,204]. Ligand binding sites and enzyme active sites are mainly located at the protein-protein interfaces hence the need to study the oligomeric protein structure. The web servers with the functionality to model protein complexes have limited user involvement for the different modeling stages and some take long days to model proteins for even simple jobs. This has necessitated the development of PRIMO-Complexes, to complement the first PRIMO's functionalities to cater to the research community whose studies deal with protein-protein interactions.

2.7 Research aims and objectives

The overall goal of part one of this project was to extend the functionality of the PRIMO pipeline [87] to model multimeric proteins and biological assemblies. This extension is referred to as *PRIMO-Complexes*, meaning "PRIMO with the functionality to model protein complexes and biological assemblies." To achieve this aim, the work was divided into the following specific objectives:

1. To develop algorithms for identifying templates from the Protein Data Bank (PDB) for modeling multimeric proteins.
2. To develop a mechanism for aligning target-template sequences of multimeric protein structures.
3. To develop algorithms for predicting (modeling) multimeric protein structures from PDB templates.
4. To develop an interactive user interface for inputting protein sequences and visualising 3-Dimensional multimeric protein modeling results.
5. To evaluate the accuracy of this pipeline to model protein complexes.

CHAPTER THREE

DEVELOPMENT OF PRIMO-COMPLEXES: A WEB SERVER FOR MODELING MULTIMERIC PROTEINS AND BIOLOGICAL ASSEMBLIES

Chapter overview

Mostly, protein monomers have been modeled, however most proteins function by interacting with other proteins and assemble into stable protein complexes. There is need to know the structures of protein complexes to understand how they function. Modeling of these proteins is either performed using standalone software or automated web servers. Users with little or no experience in using bioinformatics software find it hard to install and use this software hence resorting to automated web servers. Various protein modeling servers have been developed; however, they have some limitations such as limited user involvement in the modeling process and long job processing times. This chapter presents the design and implementation of algorithms for modeling protein complexes and biological assemblies with an intuitive user interface. This web server is termed as PRIMO-Complexes, an extension to PRIMO which initially designed to model only monomers. The web application can be freely accessed at https://primo-oligo.rubi.ru.ac.za/.

The chapter provides detailed information about the description of the design, algorithms implemented, features and implementation of the web server.

3.1 Implementation overview

We developed PRIMO-Complexes as a new platform for modeling multimeric proteins, with linkages to the first PRIMO web server, which models only monomers. To ensure smooth interoperability and function across both platforms, we developed PRIMO-Complexes using technologies similar to the first PRIMO. The systems development lifecycle (SDLC) approach was used to develop PRIMO-Complexes, covering seven stages in the process [205]: 1)

planning, 2) requirements analysis, 3) design and prototyping, 4) software development, 5) software testing and evaluation, 6) integration and implementation, and 7) operations and maintenance. During the planning stage, the objective of developing PRIMO-Complexes was defined and the scope of the existing PRIMO web server was analysed. The specific details of requirements for PRIMO-Complexes were also determined. These included technologies, programming languages to be used during development and user needs. The system and software architectures were outlined as well as the user interface design for PRIMO-Complexes. After defining the design, the web server was then developed, tested, and continuously maintained.

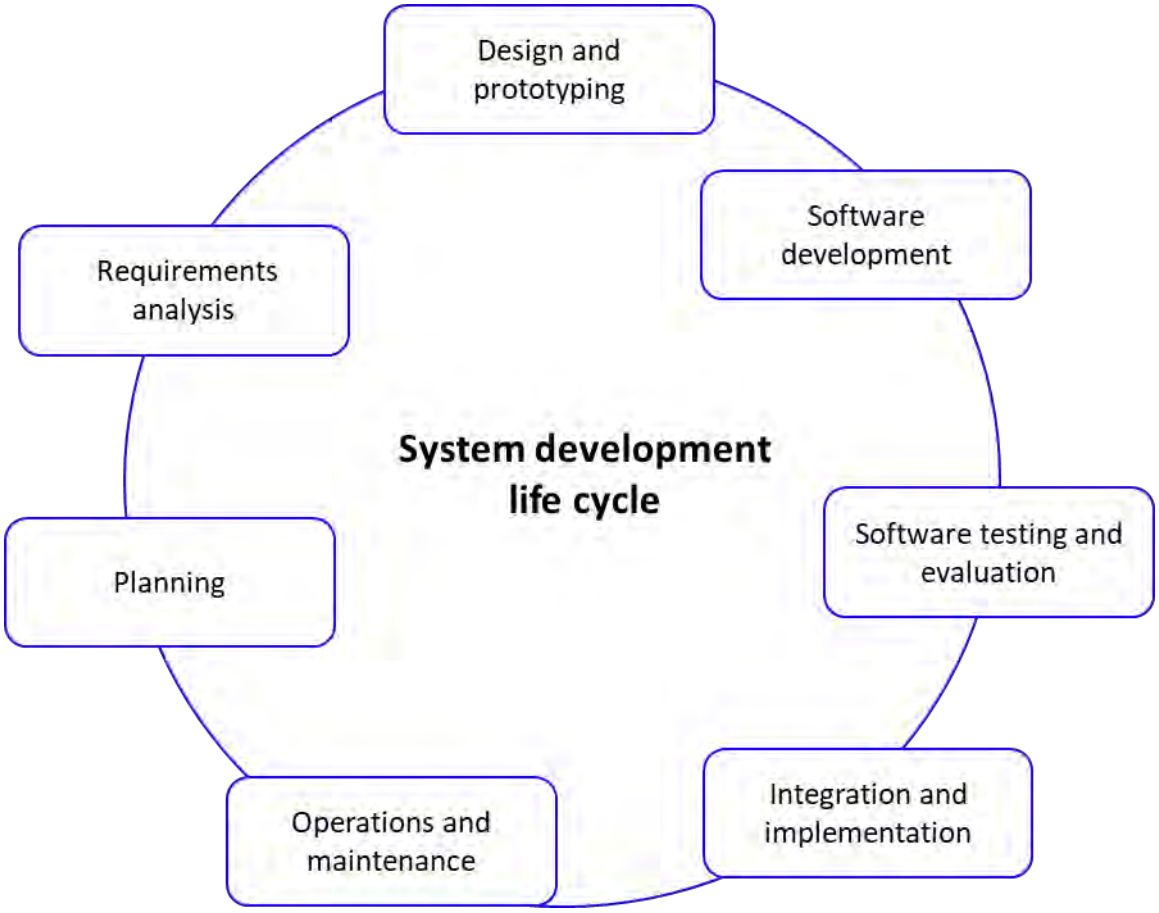


Fig 3.1. A representation of the system development life cycle followed to develop PRIMO-Complexes. This process involves seven stages which guide in project management, system deployment of the final product and later maintenance. Adapted from [206].

3.1.1 Software development process model

Among the system development process models, the Waterfall model was the most appropriate to use for our needs [207]. The functionality of the PRIMO-Complexes web server to model multimeric proteins required a sequential approach since the stages involved can be developed as separate phases but in a linear order (Fig 3.2). The stages to accomplish multimeric modeling were also well defined from the start and unlikely to change during the process.

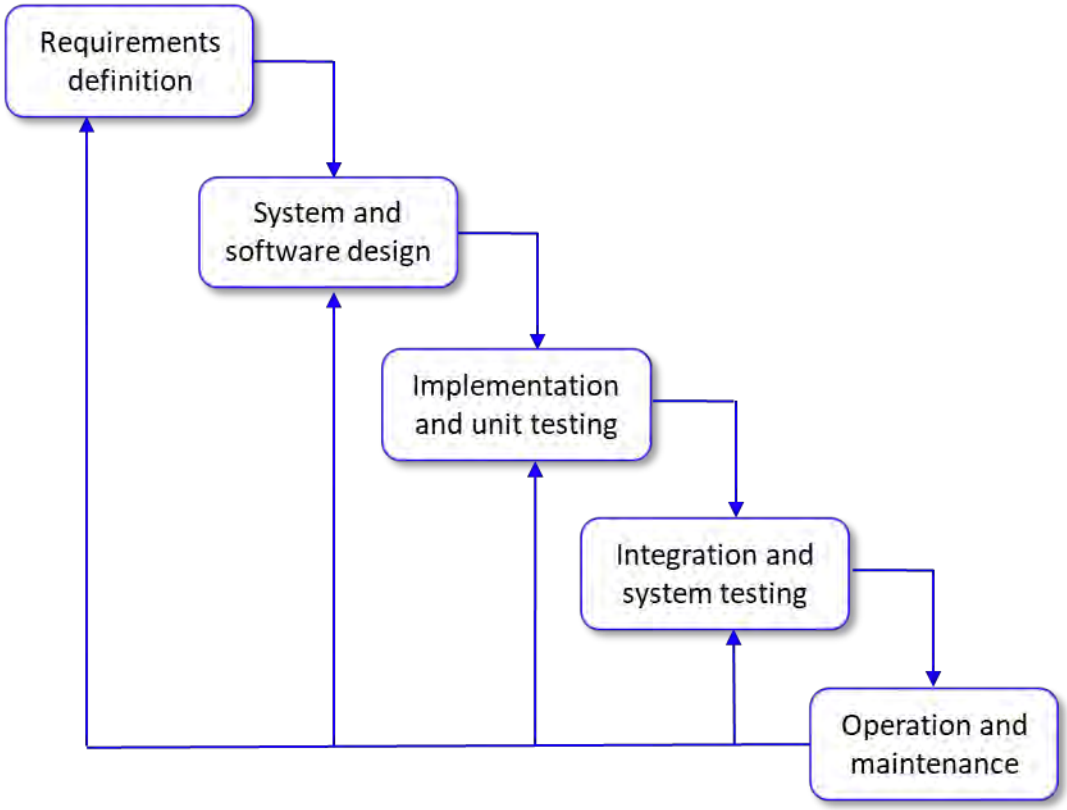


Fig 3.2. The diagram shows the Waterfall model phases used to implement the PRIMO-Complexes web server. Each computing node offers different computing power. Adapted from [207].

3.2 Requirement definition

An assessment of the initial PRIMO web server was carried out. This web server was customised to deal with monomeric proteins from the start to the end of the process. The architecture of the web server was evaluated, and this showed that there was no possibility of just extending or embedding the functionality of modeling multimeric proteins. The web server

was tailored to process and display monomeric protein results. Since PRIMO builds protein monomers and the need was to develop a system that models multimeric proteins, the following approaches were considered. The user interface, new algorithms to build multimeric proteins, the organization of how the results are to be displayed and workflow of the modeling process necessitated the development of a separate web server.

The backend and frontend of the web server were also assessed to ascertain the coordination in performing protein modeling. The scripts were studied to understand how they functioned and connected with the rest of the system. From this activity, scripts were evaluated to assess which ones to reuse, not use and those to adapt and transform to support modeling of protein multimers. The assessment showed that protein modeling steps to be followed would be the same however to model multimeric proteins, the architecture, and organisation needed to be different.

Evaluation of the existing web servers that model multimeric proteins was done. These servers included SWISS-MODEL [88], Rosetta [96–98] and I-TASSER [100]. This was because there was need to establish how these servers function, their limitations and how to make this new system better. It was discovered that these servers had limited user involvement for example choosing ligands to model, choosing the most suitable template, and editing target-template alignment. Software and development technologies needed to develop PRIMO-Complexes such as Django, Django Representational State Transfer (REST) framework were identified.

3.3 System and software design

The PRIMO-Complexes architecture can be described in two aspects: the system architecture and the software architecture. The system architecture (Fig 3.3 and Fig 3.4) indicates how the web server communicates with the High-Performance Computing JMS [208] platform while the software architecture (Fig 3.5 and 3.6) represents the actual design of the web server.

3.3.1 System architecture for PRIMO-Complexes

The system is organized and deployed this way to ensure the efficient execution of each module. Each computing node offers different computing power. Based on this architecture, the PRIMO-Complexes components are deployed such that the most compute-intensive tasks being executed on the HPC cluster and the least, on the user's personal computer (browser).

This architecture eliminates the need to only access PRIMO-Complexes through high processor-powered devices. This is possible because the system code is refactored in such a way that the browser client program can be handled by the most prevalent personal computing nodes in use currently.

Execution of a single modeling job on PRIMO-Complexes requires the deployment of several computing nodes with an HPC cluster at its core (Fig 3.3). As illustrated in Fig 3.3 and Fig 3.4, the system is organised in a way that jobs are executed by compute nodes on JMS while the fewer manipulations are run by the PRIMO-Complexes web server and web interface (browser). At the heart of PRIMO-Complexes is the Job Management System (JMS) [208]. This handles all scripts for the PRIMO-Complexes modeling pipeline stages, exposing them as services through a RESTful API. The HPC cluster runs the job after it is submitted through the PRIMO-Complexes web server to JMS, whereas JMS monitors execution of the job. Once the job is complete, JMS notifies the PRIMO-Complexes web server which returns the results to the user interface (Fig 3.3).

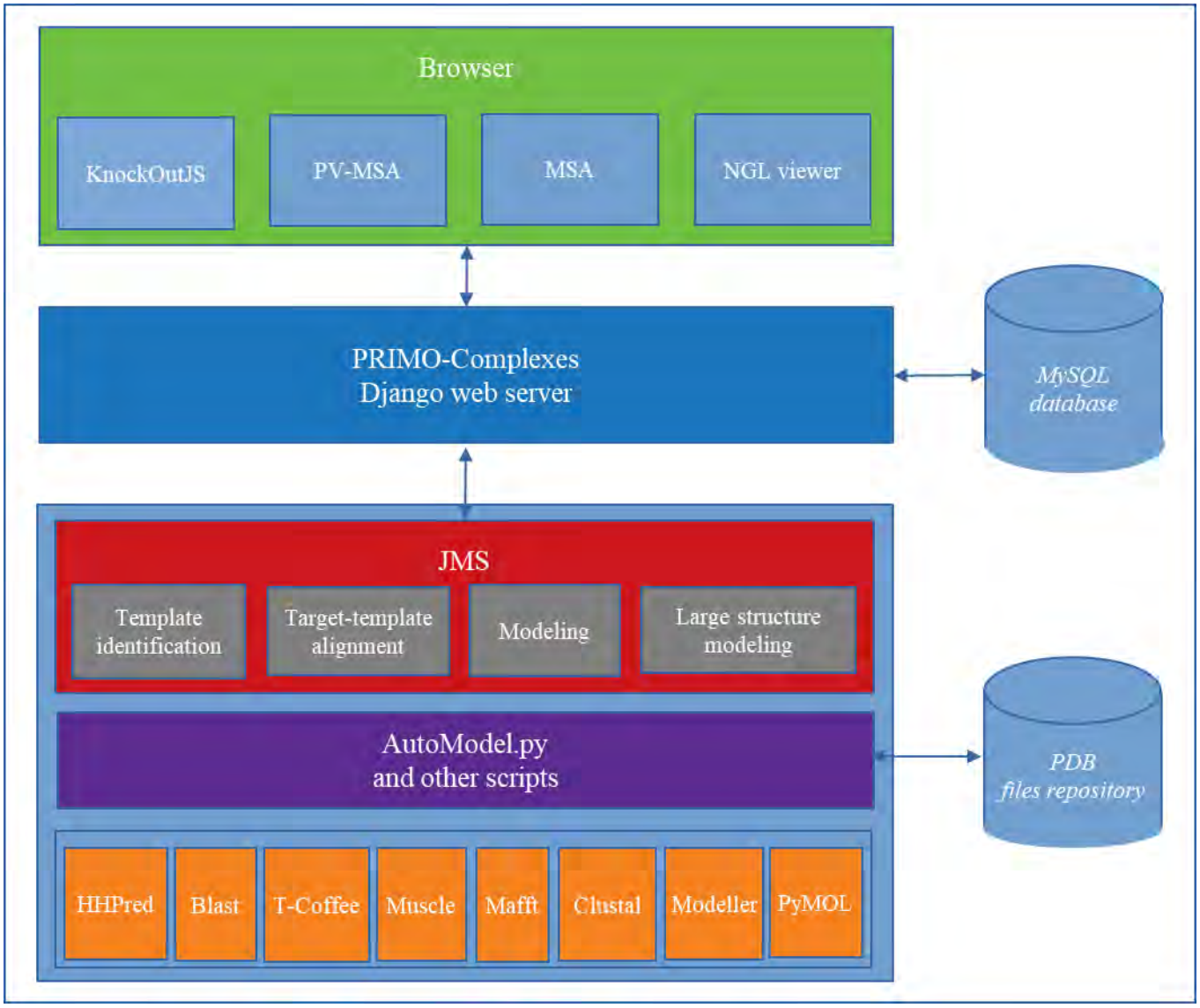


Fig 3.3. System architecture of PRIMO-Complexes – system components. Each computing node in JMS offers different computing power. Jobs are submitted through the PRIMO web server to the Job Management System (JMS).

The PRIMO-Complexes system organisation allows efficient execution of each module since each JMS node offers different computing power. Based on this architecture in Fig 3.4, the PRIMO-Complexes components are deployed such that the most compute-intensive tasks are executed on the HPC cluster and the least, on the user's personal computer (browser). This architecture eliminates the need to only access PRIMO-Complexes through high processor-powered devices.

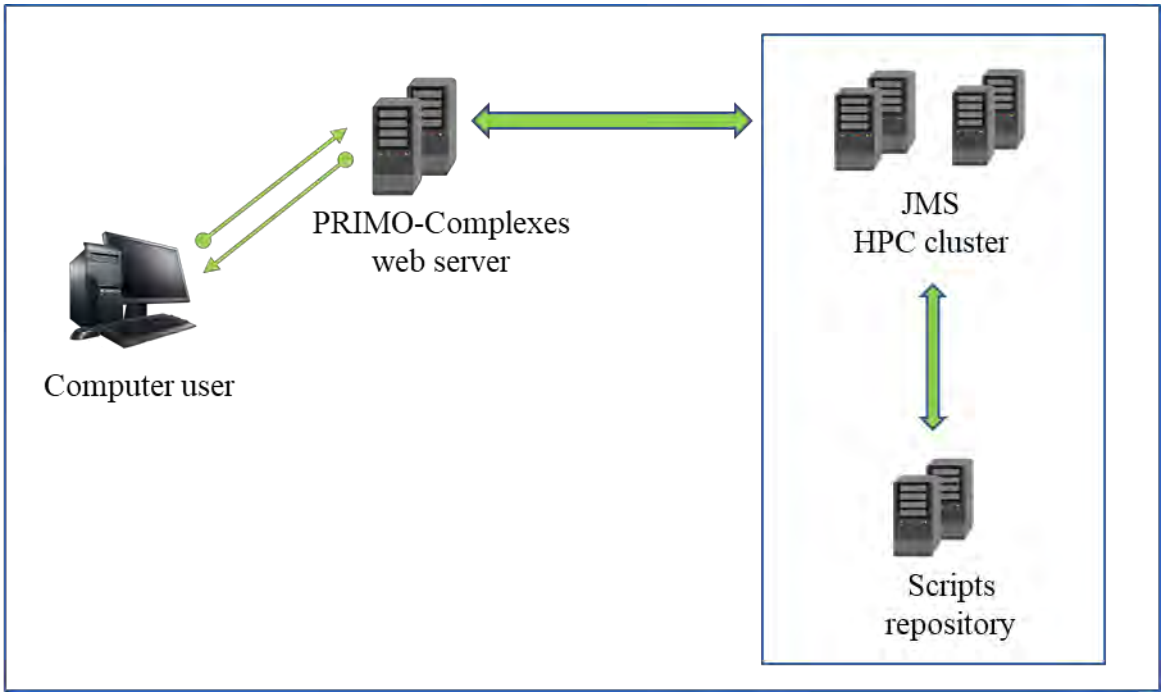


Fig 3.4. System architecture of PRIMO-Complexes – sub-systems. Several computing nodes are required to execute a single job on PRIMO-Complexes.

3.3.2 Software architecture for PRIMO-Complexes

The PRIMO-Complexes web interface acts as the link between users and the multimeric protein modeling scripts embedded in JMS. When a job is created, an associated file structure is also set up on the PRIMO-Complexes server (Fig 3.5). This directory is named using the system-generated Job_ID and holds sub-directories for every stage of job execution. Each of these folders holds data relevant to its corresponding stage on the PRIMO-Complexes modeling pipeline. The output generated at every stage is stored in these folders. The output includes desired generated files and error logs in case of failure. At the end of each subsequent stage, these output files are propagated to the next stage to act as input.

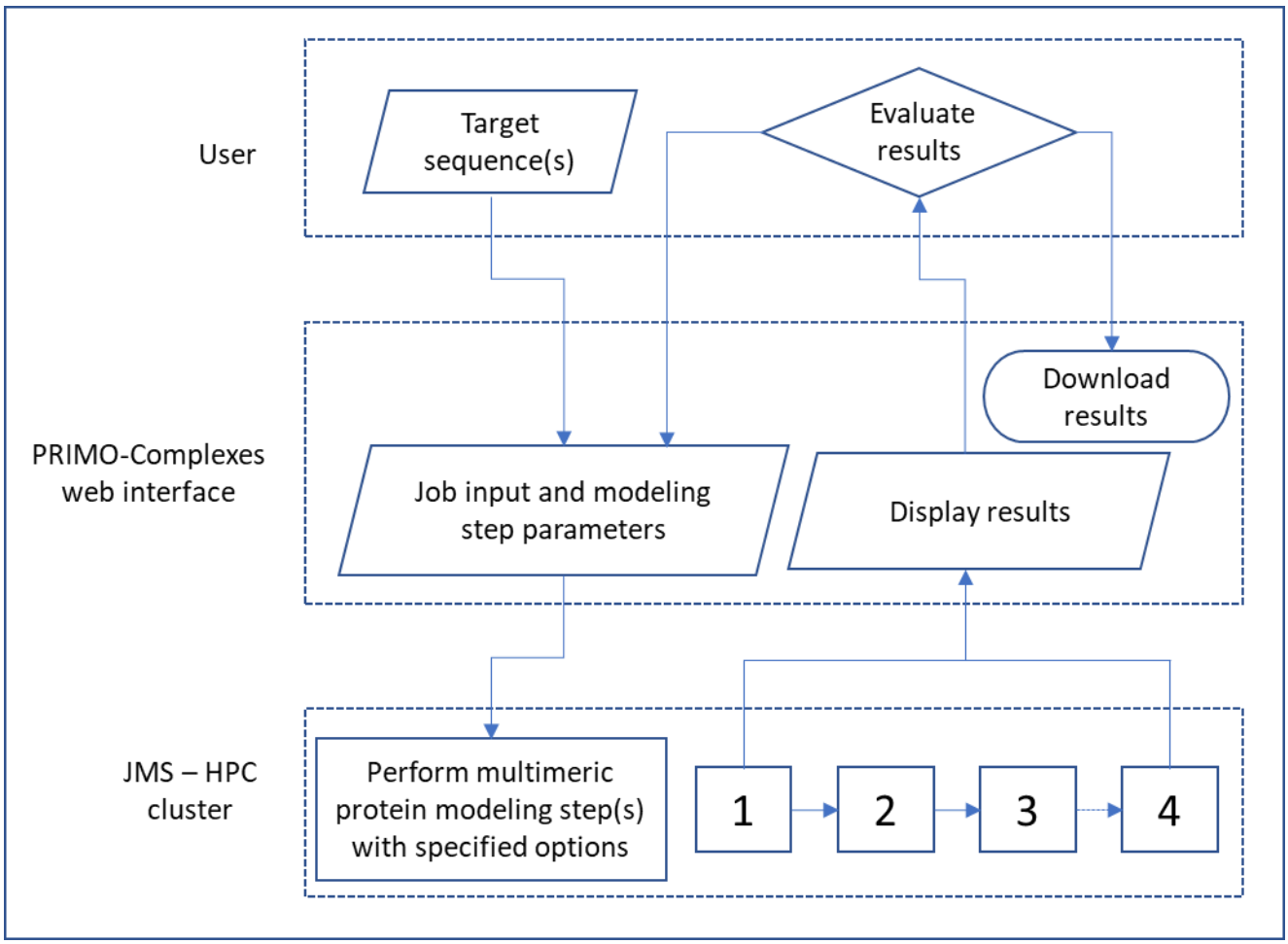


Fig 3.5. Software architecture of PRIMO-Complexes. This diagram shows how a potential user would interact with PRIMO-Complexes web interface. Once data is input, the interface was designed to communicate with the Python scripts on JMS. The scripts on JMS were developed to perform homology modeling of protein complexes and biological assemblies indicated as 1) template identification, 2) target-template alignment, 3) protein modeling and evaluation, 4) Building large macromolecules.

3.3.2.1 PRIMO-Complexes Django web server design

The PRIMO-Complexes Django web server design was designed to contain four applications, including the Impi, the WebUi, the Users and the Evaluation app. The Impi app is the largest application handling most of the burden of calls on the server.

3.3.2.1.1 WebUi app

The WebUi app was used to create the user interface on the PRIMO-Complexes Django web server. This interface is implemented in form of a web application to be accessed through a web browser. The PRIMO-Complexes Django web server loads the entire web application as

a single page – index.html, unlike other web applications that consist of several pages that are loaded as a user interacts with the application. In PRIMO-Complexes, this page was designed to comprise several view fragments/window frames that, are initially hidden from the user and are brought into view selectively as the application changes state. For example, an error dialog view fragment will be set to be visible to the user to show an error if it occurs.

These embedded view fragments, however, do not contain data other than static state representation data. For example, the error dialog view fragment is designed not have the actual error message, yet it may contain style code and an icon for depicting an error message. Each time an error is to be displayed to the user, this same view code is activated and populated with the appropriate error message. This design approach eliminates the need for repetitive downloading of static view code from the server which saving the user time and bandwidth by reducing load time while increasing efficiency.

State synchronization between the server and client after loading the view markup, style, and JavaScript code on the browser is what remains. This is done through calls to the RESTful APIs using AJAX.

3.3.2.1.2 Users app

We created the Users app to handle user account-related functionality such as authentication, account creation, password recovery, and logout in PRIMO-Complexes. The Users app exposes its functionality through a RESTful API implemented in the views.py script. The user account data are presented to a MySQL database using Django's ORM.

3.3.2.1.3 Evaluation app

We included an Evaluation app which handles model evaluation after the modeling process is completed. It consists of a RESTful API exposing only a single endpoint function procheck. This function takes the job_id, model_name and number as arguments and uses PROCHECK

to evaluate the given model. PROCHECK evaluates the stereochemical quality of a protein structure, producing several PostScript plots displaying the overall and residue-by-residue geometry [188].

3.3.2.1.4 Impi app

The Impi app in PRIMO-Complexes handles requests pertinent to the core modeling tasks. These requests are handled through a RESTful API that accepts and returns data without any associated presentation or styling code such as HTML or JavaScript. These endpoint functions rely on routines factored out of the views script into a helpers.py script. This is done to reduce congestion on the views.py file and as a result, promotes reusability and maintainability of the application. Impi manages all jobs on the PRIMO server. For instance, it creates new jobs and manages job stages from template identification through alignment stage to the modeling stage and/or complex biounit modeling stage.

Job creation involves instantiating a new job object and populating its attributes with data about the new Job. This job object is then presented to the database through Django's ORM. Job object attributes.

3.4 Implementation and unit testing

The PRIMO-Complexes' web interface was designed as a single page, managed by Django web framework [209] and Representational State Transfer (REST) framework [210]. The web interface communicates with JMS tools via Asynchronous JavaScript and XML (AJAX) calls and the web page content is updated using knockoutJS library. After development of every stage in the PRIMO-Complexes web server, the functionality was tested to ascertain the output and behaviour of the system. Below, I briefly describe the technologies deployed in the development of PRIMO-Complexes and tests performed at every stage.

3.4.1 Web technologies and software used in PRIMO-Complexes development

3.4.1.1 Django web framework

Django [211] is a Python web framework developed to enable rapid web application development. It is built on the "Model View Template" (MVT) architectural paradigm which, is similar to the more common Model View Controller (MVC) architecture and allows separation of concerns in the web application design [209].

In the development of PRIMO-Complexes, Django was used to reduce on the web development time focusing on the application development. Since Django is secure and exceedingly scalable, it minimised the chances of making common security mistakes and provide support for changes during application development. Applications designed based on MVT architectural paradigm are easy to maintain and refactor in case the need arises owing to loose coupling. The MVT components are discussed below.

3.4.1.1.1 Model

Data structuring is handled by the model in Django through the use of classes that are descendants of *django.db.models.Model*. In the PRIMO-Complexes web server, a MySQL database was used, and the subsequent models (layout of database) were created. The models were represented by Python classes and each class was mapped to a single database table as well as each attribute mapped to a single database table field. Since Django was used, the application was automatically given a database-access API (Application Programming Interface) thus performing CRUD (Create, Read, Update and Delete) operations.

3.4.1.1.2 View

The view is equivalent to the controller in MVC and holds application logic. It is placed between the template and the model. In PRIMO-Complexes, the view processes user requests,

queries the model for data and passes the data to the appropriate template scripts for presentation to the user.

3.4.1.1.3 Template

This is equivalent to the view in MVC nomenclature and consists of mainly HTML, JavaScript, and CSS scripts. The template component was used to design and develop the web interface of PRIMO-Complexes. The template is used for data presentation in PRIMO-Complexes web server where the user interacts with interacts with it on the browser.

3.4.1.2 Django REST framework

The Django REST framework [210] is a toolkit for developing APIs that conform to the Representational State Transfer (REST) [212] architectural style constraints. In PRIMO-Complexes, the REST architectural style was used to allow communication between the backend and World Wide Web (WWW) through Hypertext Transfer Protocol (HTTP). This framework was used because Django has an Object Relational Mapper (ORM) which permits database interaction in a Pythonic way.

3.4.1.2.1 HTTP methods

To provide users with access to the PRIMO-Complexes data, HTTP methods including GET, POST, PUT and DELETE were used. Furthermore, the specification of HTTP allows extension in a way that does not break already existing infrastructure [213].

GET: The GET method is used for retrieving information (in the form of an entity) identified by the Request-URI. The GET request does not contain any payload except for request parameters encoded in the URL itself.

POST: The POST method is used to query the origin server to accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.

PUT: The PUT method sends a request for the enclosed entity to be stored under the supplied Request-URI.

DELETE: The DELETE method sends a request for the origin server to delete the resource identified by the Request-URI.

Although the REST specification allows for state representation in several formats including JSON, XLT, HTML, or plain text, in PRIMO-Complexes, JSON is predominantly used since Python objects cannot be sent over the network.

3.4.1.3 MySQL database

During PRIMO-Complexes development, My Structured Query Language (MySQL) database was used. MySQL is an open-source relational database management system [214] and runs as a service on a separate process accessible through a specified port, and the default port is 3306. Django supports MySQL in its object-relational mapper (ORM). Using MySQL database in PRIMO-Complexes enables, one to create, update, read and delete data from the database through a Python interface without the need to write server query language (SQL) code.

3.4.1.4 KnockoutJS

KnockoutJS is a JavaScript library based on Model-View-ViewModel (MVVM) pattern and it is used for developing rich and responsive web user interfaces. The user interface of PRIMO-Complexes was built using Knockout (KO) [215] because supports development of user interfaces that require updates on only specific parts of the interface when an underlying data model changes. View widgets are bound to data models using the KO JavaScript library such

that when the data models are updated, they cause the widgets to update automatically reflecting the change. Using Knockout significantly aided the development of PRIMO-Complexes since the web server involves waiting on long executing tasks whose statuses have to be reflected on the user interface seamlessly without the need to update the entire web page. Knockout handled this process very well. The underlying data models were updated using AJAX.

3.4.1.4 Asynchronous JavaScript and XML (AJAX)

Asynchronous JavaScript and XML (AJAX) was used in the development of PRIMO-Complexes web server to enable asynchronous interaction between a server and client (web browser). AJAX also eliminates the need for continuous browser refreshing to display new data on a web page. This is done by enabling calls to a web server after a page has already been loaded to asynchronously fetch as well as send data to and from the web server.

3.4.2 PRIMO-Complexes modeling algorithm

The PRIMO-Complexes modeling algorithms are presented in Fig 3.6. The required input for this web server is an amino acid sequence(s) in FASTA format. If the target protein is a heteromultimer (consists of different protein chains), amino acid sequences for each subunit must be specified. The subsequent process is divided into either three or four steps depending on the size of the target protein. These steps are: 1) template identification, 2) target-template sequence alignment, 3) protein modeling and evaluation or 4) building large macromolecules. Each step allows user intervention, parallel job processing and altering modeling parameters.

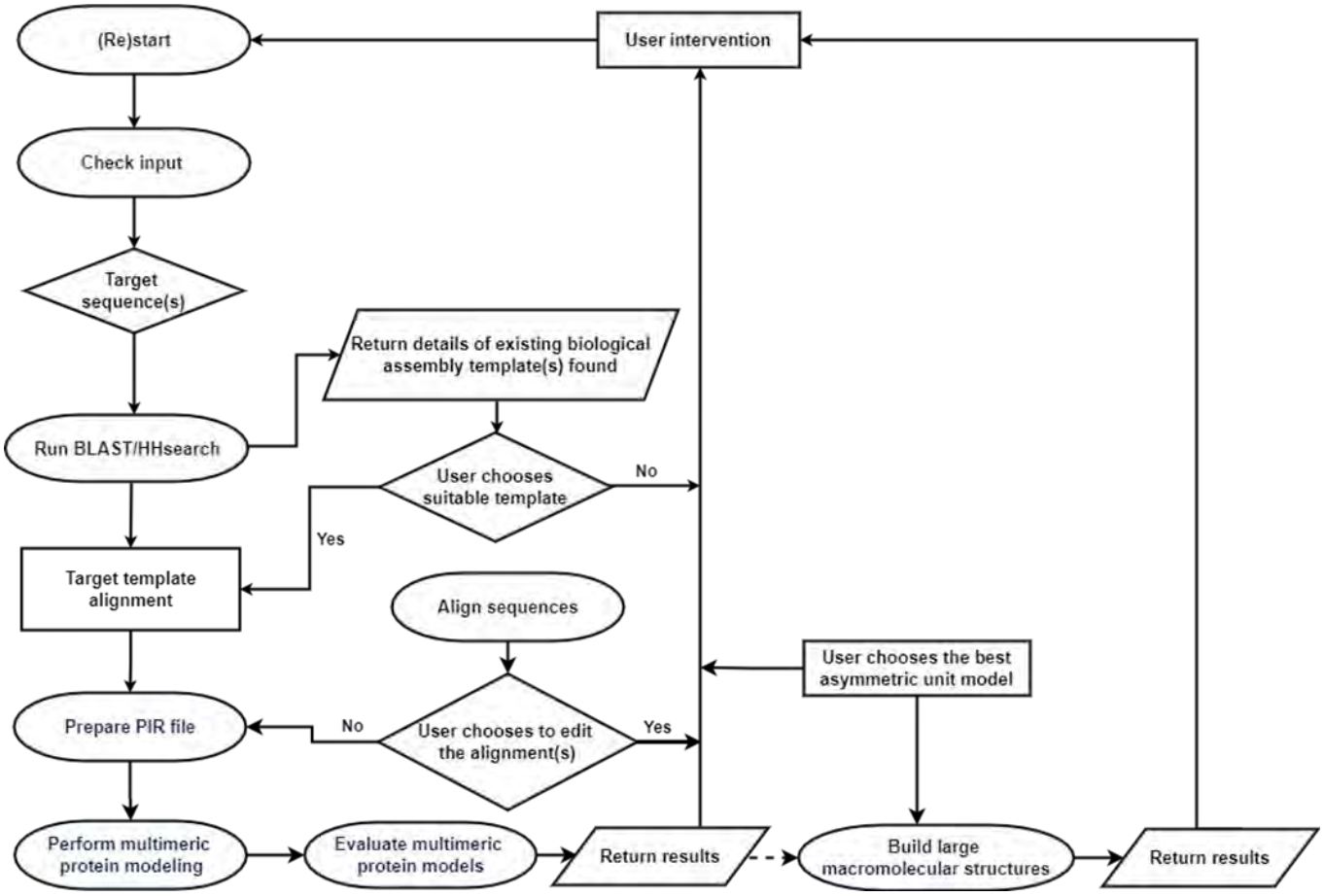


Fig 3.6. A flowchart shows multimeric protein modeling algorithm incorporated by PRIMO-Complexes. The steps involved in multimeric protein modeling are broken down into tasks and processes performed by either the user or PRIMO-Complexes. The major steps are template identification and selection (run BLAST/HHsearch), target-template alignment, protein modeling and evaluation (perform multimeric protein modeling, evaluate multimeric protein models), and building large macromolecules.

3.4.2.1 Template identification

Homologous template structures from PDB are determined at this stage based on a given target sequence file. This step is achieved by a script that was written and named get_templates_multimer.py. This sequence file is encoded in FASTA format and may contain one or many sequences. The sequences are checked for duplicates before they are processed. A target sequence FASTA file containing several sequences is split into different FASTA files, each for every sequence. Structures for templates are then iteratively generated for each single target sequence. Two template identification methods are availed as options to be used to identify templates at this stage, including BLAST and HHsearch. Asymmetric unit structure

templates that make up the intersection are determined as well as their corresponding biological assembly structures. Homomultimeric structures that are represented more than once using differing chain names are removed. Biological assembly information is also determined including multiple chains, biological assembly identification (ID), oligomeric state, number of existing biological assemblies in the PDB and description.

3.4.2.2 Target-template sequence alignment

The biological assembly PDB file selected from the template identification is parsed to extract all homologous sequences to the target protein. In the case of repeated identical proteins in the biological assembly PDB file, the chain identifiers (IDs) are renamed before aligning sequences. These renamed unique chain identifiers are then used when performing alignment for clarity at this stage and the modeling stage. The new unique chain IDs with their corresponding PDB information is written to a new PDB file to be used during the modeling stage.

Missing residues and non-standard amino acids are included in the alignment as the "X" character. An algorithm to compact the multiple sequence alignment into a single paired alignment for identical sequences is included in the alignment stage. This was done to remove the redundancy of having more paired alignments. Non-identical sequences are presented as separate paired alignments. The target-template alignment script (align_tempTargetMultimer.py) provides the user with an option to align sequences using either Clustal-Omega [167], MUSCLE [135], MAFFT [136] or T-COFFEE [137] program. Currently, the PRIMO-Complexes server allows protein modeling using one protein structure template. Therefore, multiple sequence alignment is only permitted for homomultimeric protein sequences which at times have differing residues.

3.4.2.3 Protein modeling and evaluation

This step uses the target-template sequence alignments and the modified biological assembly template structures from the previous step to generate a PIR file for each chain. All the PIR files are integrated to form a single PIR file. Alignment sequences are pre-processed to conform to the PIR format. Characters that are not recognised by MODELLER [119] in sequences are replaced including missing residue characters ("X") with gap characters ("-") and modified characters with period (".") characters. Each target-template alignment pair in the PIR file is trimmed so that the target sequence corresponds to the template sequence at each end. Biological assembly PDB files are updated with ordered chain IDs as well as ligand information for the given chains. This way only ligands selected by the user for the desired chains are added. Each template sequence in the PIR file is compared to its corresponding extracted sequence from the biological assembly PDB template. If the user selected ligands and ions at the template identification step to be modeled into desired chains, ligand information is added to the integrated PIR file. The integrated PIR file is updated with start chain, start residue, end chain and end residue for sequence and template segment. The ligands that are included in the new updated biological assembly PDB file in each chain are identified and their positions become the end residues for the template segments in the integrated PIR file. The target sequences corresponding to those template segments are then updated with gap characters to match the length of the template sequences. These gap characters in every target-template pair are replaced with period characters to match the ligand positions in the desired template chains.

After preparing the PIR file, a modeling script is created and then run by MODELLER. Generated models are then evaluated using normalized DOPE function (z-DOPE score) [158] in MODELLER, and the PROCHECK tool [188]. If the selected biological assembly template has more than 62 chains, then the modeling step uses the asymmetric unit PDB template to

perform modeling of the smallest structural unit (protomer). This process involves proceeding to the next stage.

3.4.2.4 Building large macromolecules

This step is automatically initialized after the modeling stage in case the protein subunits in the biological assembly template used for modeling exceed 62 chains. This step utilizes the user-selected best model and the biological assembly template selected at the template identification stage. A script that uses PyMOL software [216] is used to construct the structures of macromolecules. This script creates copies of the best protein model equivalent to the number of protomers in the PDB file of the biological assembly and superimposes them onto the respective protomers of the biological assembly template then saves the generated structure.

3.4.3 Molecular and sequence visualization

Visualisation is a key step during structure analysis. The Protein Viewer - Multiple Sequence Alignment viewer (PV-MSA) plugin was embedded to visualize templates and modeled protein structures. Additionally, an MSA plugin was incorporated to visualize sequence alignments and an NGL viewer to render large macromolecules. The PV-MSA plugin initially linked the PV and MSA plugins and was customised for monomeric proteins [87]. PV-MSA and MSA plugins were extended in the current study to cater for multimeric protein visualization.

3.4.3.1 PV-MSA plugin wrapper

PV-MSA is a JavaScript wrapper around the Protein Viewer (PV) [217] and BioJs Multiple Sequence Alignment (MSA) viewer [218]. The PV and MSA viewer are based on JavaScript. The PV [217] is used for visualization of protein structures whereas BioJs MSA viewer is used to visualize and analyze multiple sequence alignment datasets [218]. These viewers are equally important in bioinformatics, but they are less versatile. The need to visualize multiple sequence

alignment with their corresponding structures led to the development of a PV-MSA wrapper by David Brown combining their functionality [87]. It leverages some of the features of PV and MSA to offer a single viewer, capable of rendering a protein structure, together with an MSA alignment for the template structure sequence.

Although the initial implementation of PV-MSA featured support for only monomers, extensions have been included in the present study to provide support for more complex structures. Notably, the current version has a separate alignment for each template structure chain sequence to its corresponding target sequence. The resultant viewer interlinks PV and MSA in such a way that once a residue of a given template structure chain sequence is selected in the PV viewer, the same residue in the corresponding MSA alignment is highlighted and vice-versa. In addition, the PV-MSA provides a feature to display and hide protein complex structures with their corresponding paired chain alignments. Paired sequence alignments are displayed only for matching hit chains from the template structures

The PV-MSA plugin provides support for different rendering styles such as cartoon, balls and sticks, lines, trace, spheres, points, and tubes as well as spin and zoom capabilities.

3.4.3.2 MSA plugin

PRIMO-Complexes uses the MSA plugin [218] described above for displaying sequence alignments generated by embedded alignment methods including Clustal-Omega [167], MUSCLE [135], MAFFT [136] or T-COFFEE [137]. The sequence alignment stage of the PRIMO-Complexes pipeline features distinct sequence alignments rendered for every target-template sequence pair being modeled. In addition, the MSA plugin in the PRIMO version for modeling protein multimers has the functionality to cluster identical chains, and present paired alignments for different chains to the user. These alignments can further be analysed, edited and/or exported before proceeding to the next stage of modeling.

3.4.3.3 NGL viewer

NGL viewer is a web application built with JavaScript and WebGL for visualization of proteins and other molecular structures [219]. NGL viewer was embedded into PRIMO-Complexes by writing a single JavaScript file containing API methods that are used to create a stage on which large macromolecular structures are loaded. Zoom, spin controls and display styles such as lines, points, licorice, cartoon, ribbon, balls and sticks, tube and trace were added to this API to aid manipulation of the rendered structure. PRIMO-Complexes incorporates both PV and NGL viewers for protein structure rendering in its design.

3.4.4 Multimeric protein modeling Python scripts in PRIMO-Complexes

The main functionality of the PRIMO-Complexes web server is executed by four PRIMO-Complexes JMS tools including template identification, target-template alignment, modeling, and complex structure modeling which rely on PRIMO-Complexes Python scripts (discussed below). Two approaches were used to develop Python scripts for modeling protein complexes and biological assemblies under PRIMO-Complexes: Firstly, completely new scripts were written specifically for modeling multimeric proteins. Examples of these scripts include rmsd_complex.py, rename_biounit_chains.py, ligands_complex.py, and determine_bioassemblies_complex.py. Secondly, and where reasonable, some scripts were adapted from the first PRIMO web server that models protein monomers [87] and transformed them by incorporating capabilities to model protein complexes and biological assemblies. Examples of these include Automdel.py, calculate_resolution.py, HHPred_obj.py, and RowanFunctions.py. Lastly, some scripts were assessed and needed to be reused as is while others were not needed. Examples of reused scripts include Align_PDBs.py, HHPred_obj.py, PIR_file.py, rowanPDB_parser.py, ss_add.py, Template.py, and template_fetch_functions.py. Scripts that were never used include add_pdb_chain.py, ligand_prep.py, and find_lig_pdbs.py.

Details of the functionality of these scripts and transformation made in adapted scripts are described below.

3.4.4.1 Python scripts functionality and modifications made

Auto_model.py is the main script in the PRIMO-Complexes Python scripts repository. These scripts are invoked through Auto_model.py by PRIMO-Complexes JMS tools. The scripts work interdependently to process chunks of data at various stages of the PRIMO-Complexes modeling web server thus generating a model for the provided target sequence(s).

The Auto_model.py script was adapted, and modifications made include changes on how to fetch the target sequence, addition of a check to ensure that the listed PDB file is an asymmetric PDB file with the extension '.pdb' leaving out the biological assembly PDB file. These modifications were made because the target sequence file names were stored with a '.fasta' extension and the PDB file check was included to ensure that the listed file is an asymmetric PDB file. The Auto_model.py script contains a single class named Auto_model. This class contains seven methods including the constructor of the class itself, get_templates, align_sequences, create_pir, create_model_script, run_model_script and evaluate_models (Fig 3.7). In addition, this class, directly and indirectly, depends on other classes including PDB_parser, PIR_file, FASTA_file, Alignment, MODELLER_script and HHpred. The HHPred_obj.py script was modified to update the obsolete.dat file that lists all obsolete PDB structures in the PDB repository.

The Auto_model class' get_templates method is invoked with one of the two supported template identification methods as the sole argument. This method acts on data initially specified at object construction time. The constructor specifies the target_sequence file as a mandatory argument whereas templates, alignment, and pdb_file_location are specified as optional arguments. PDB files were downloaded from the RCSB PDB repository and all the

scripts that accessed these files were updated. These scripts include calculate_resolution.py, template_fetch_functions.py, RowanFunctions.py, and rowanPDB_parser.py. The calculate_resolution.py script was adapted and modified to include PDB structures determined by electron microscopy technique which were ignored in PRIMO. A check for "EXPDTA ELECTRON MICROSCOPY" was added as well as returning resolution values for these structures. This check was included because some multimeric proteins are determined using this technique. This script was also failing for some PDB structures which did not have the resolution section in the PDB file ending with the term "ANGSTROMS". This was solved by including an exception to return the resolution values.

The get_templates method generates templates by invoking HHblitz and HHsearch through an instance of the HHpred class. This happens when HHpred is specified as the template identification method of choice, otherwise, BLAST is executed when BLAST is specified. BLAST is invoked through a function called run_and_parse_blast_PDB, specified in the template_fetch_functions.py script. HHsuite and BLAST databases were and are periodically updated. The template_fetch_functions.py script was adapted and updated with links to the new downloaded obsolete.dat file, new downloaded PDB files and updated BLAST data. The RowanFunctions.py script was adapted and modified since it contains methods used by other scripts to process data including the template_fetch_functions.py, rmsd_complex.py, and rowanPDB_parser.py. This script was modified by including a method to call http://files.rcsb.org/download/ web page in case http://www.ebi.ac.uk/pdbe-srv/view/files/ web page fails to return the desired PDB data. An option is provided to fetch a compressed (.gz) PDB file from the locally downloaded compressed PDB files repository in case the PDB file is not found in the locally downloaded decompressed PDB files repository.

The align_sequences method directly depends on the Alignment class. This class is not an alignment itself; it rather provides methods that do sequence alignment using any of the

alignment programs specified in the PRIMO-Complexes modeling pipeline. Methods specified in the Alignment class include align_MAFFT, align_muscle, align_Clustal and align_t_coffee (Fig 3.7). These methods are all named appropriately to reflect the alignment programs they use to perform alignments.

These alignment methods are executed several times in the case of multimers to generate sequence alignments for each of the sequences contained in the target FASTA file with their corresponding template sequences. These target-template sequence alignments are used as input in the create_pir method of the Auto_model class. This method performs several operations on the alignment sequences such as inclusion of gap characters (‘-’) in case of missing residues and period characters (‘.’) for modified residues in the alignment sequences before including them in the PIR file.

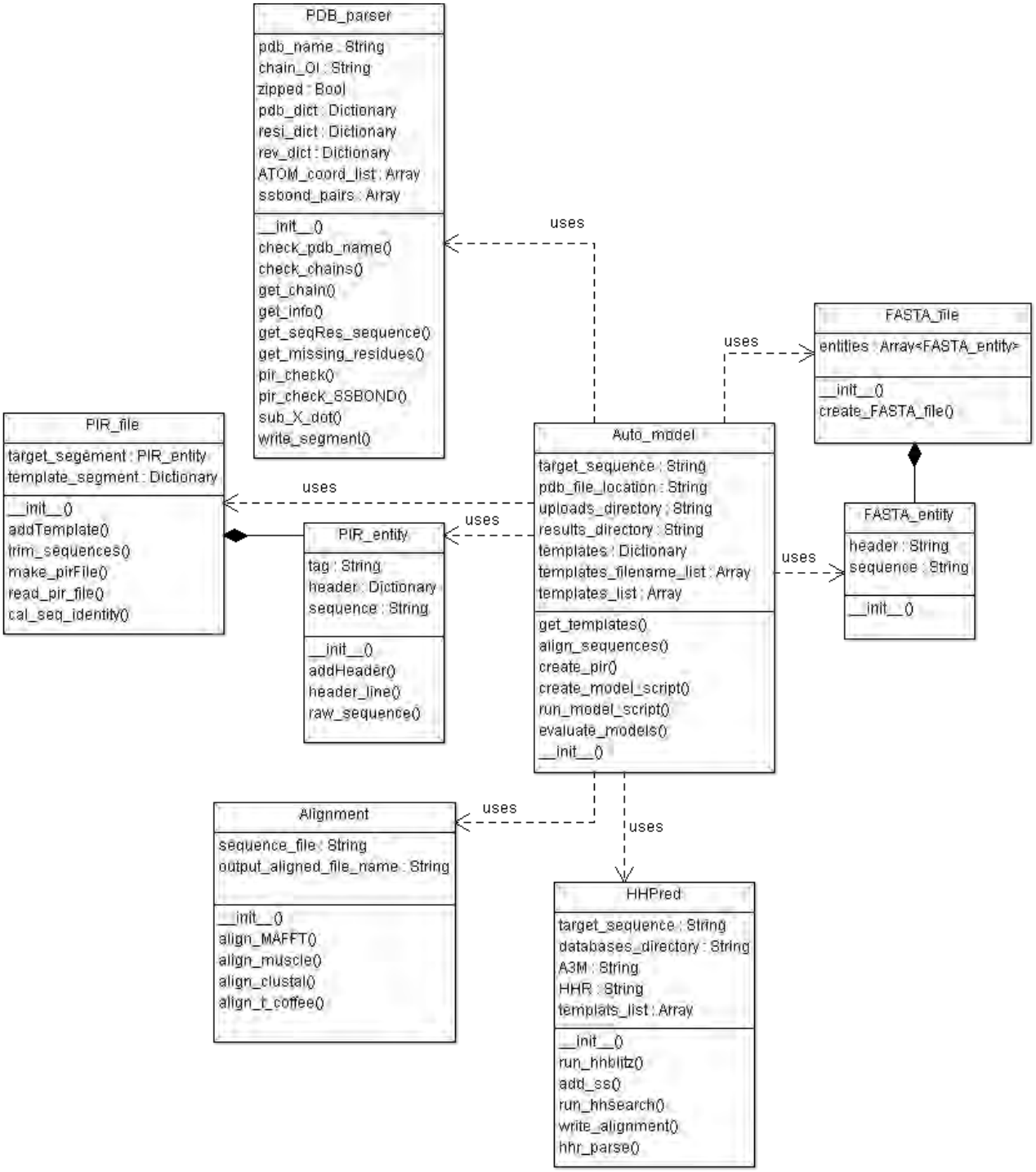


Fig 3.7. The PRIMO-Complexes class diagram for backend scripts. All the scripts are centred and run by Auto_model.py as the main script. The methods in other scripts were designed to be accessed directly or indirectly by the main script to perform the protein modeling.

In the current study, the capabilities of the PIR file creation process that was in the first PRIMO were significantly enriched to allow for complex structure modeling under PRIMO-Complexes. PIR files are generated for each of the target-template sequence alignments. Notably, MODELLER scripts require that all alignment information be specified in a single PIR file, thus these files are integrated to form an integral.pir file. This integral.pir file contains all alignment pairs together with start and end residue information for each of the template sequences as specified by the MODELLER program documentation. The resultant integral.pir file is then used in the create_model_script method where the MODELLER_script class is used to generate a script executable by the MODELLER program [119]. This script is then executed through the run_model_script function call. Evaluate_models method is used to evaluate models after they are generated by MODELLER [119]. This step involves use of *asses_ga341* and *asses_normalized_dope* methods of MODELLER's complete_pdb object to evaluate the models.

3.4.4.2 New scripts to specifically support protein multimeric modeling

Scripts were written that entirely necessitated the modeling of protein complexes and biological assemblies and these include determine_bioassemblies_complex.py, ligands_complex.py, rename_biounit_chains.py, and rmsd_complex.py.

The determine_bioassemblies_complex.py script generates all information about existing biological assemblies including chains, how the biological assembly was determined, and oligomeric state. It also filters the biological assembly template results to return only those associated with the target chain(s). The ligand_complex.py script was written to read ligand information from the biological assembly template PDB files. There was also a requirement of distinguishing the chain identifiers in case the biological assembly PDB file had repeating model segments hence writing a script named rename_biounit_chains.py was needed to sort

that issue. This script renames chain IDs up to 62 chains in case they are repeated to avoid confusing MODELLER when modeling as well as writing and updating the PDB files with the renamed unique chain IDs. There was also need to calculate the RMSD values for the multimeric proteins hence writing a script named rmsd_complex.py to solve this. This script handles several chain model files and prepares a variable to hold data specific to every chain of interest including pdb_atoms, pdb_chain, and model_chain. It then calculates RMSD values.

3.5 Integration and system testing

After writing every script, they were tested to ascertain the output and detect bugs which were mitigated expeditiously. The scripts were embedded in the web server to catch logical errors which cause features to behave incorrectly. The system was rigorously tested by running a several protein sequences from either homomultimers or heteromultimers. These were tested using different parameters including template identification methods, alignment options, number of models to be produced and refinement levels. The output was assessed to be sure it is the correct result expected since some bugs do not make the system to crash. Functional errors were also tested to handle exceptions in the system. The system was tested with the wrong input sequences to check the error handling issues.

This project was presented to the people in our laboratory and research community during monthly meetings and academic conferences for insightful discussions. These discussions included modifications to be made, user interface preferences, and expected release time-frames. System testing was performed by people in RUBi group which involved the alpha testing done during the early stages of development specifically after every step in the multimeric protein modeling process. System quality including performance efficiency, security, and reliability were evaluated. The ability of the system to protect user information and data was assessed by ensuring that every user registers and logins to access their accounts showing all the previous and current job running. The performance efficiency and reliability of

the system was tested by loading several model processing jobs to assess its response time and successful runs without experiencing timeouts. After the development, a pilot version of PRIMO-Complexes web server was tested to assess if its performance conforms to the user requirements in the actual environment. Furthermore, the accuracy of the PRIMO-Complexes web server was evaluated to assess how well it models multimeric proteins. More details on this evaluation are given in chapter 4.

3.6 Operation and maintenance

The system was organised in a way that permits modification, and extension to the code in order to ease maintainability in future. Errors reported by the end users are or will be implemented after deployment to mitigate new issues and unforeseen bugs that arise.

3.7 Results and discussion

In this section, results obtained using the methods discussed above regarding the development and functionality of PRIMO-Complexes web server are described and discussed.

3.7.1 Web server location and access

PRIMO-Complexes web server is freely available to the public as long as they register on the MODELLER website to get an access key to the MODELLER program. PRIMO-Complexes web server can be accessed at https://primo-oligo.rubi.ru.ac.za/.

3.7.2 Description of the final system architecture and functionalities

PRIMO-Complexes is powered by several technologies including the Django web framework [209], and Django REST Framework [210]. The backend scripts were written in Python and presented as four distinct tools on JMS including template identification, target-template alignment, modeling, and complex structure modeling.

3.7.2.1 PRIMO-Complexes software and system architecture

PRIMO-Complexes web server was developed using Python as the main programming language. The software stack of this web server comprises a rich internet app (RIA) developed primarily with JavaScript, a Django web server, and a collection of scripts that are accessed through an HPC Job Management System (JMS) as shown in Fig 3.8.

PRIMO-Complexes RIA was designed conceptually to be a single page that loads once at start-up and its data is refreshed as required. This design was achieved by incorporating various other static code consisting of mainly HTML, JavaScript, and CSS into a single file known as index.html. HTML provides the structure; CSS provides the style information and JavaScript provides the application logic. The application state transfer between the server and client is handled through the RESTful APIs developed on the Django server and AJAX. The client application leverages the power of Knockout, adhering to its Model-View-View-Model (MVVM) design paradigm. Once data is loaded through an AJAX call, only the model data is updated, this change in the model is automatically propagated through knockout's observer mechanism to the view widgets seen by the user on the browser (Fig 3.8).

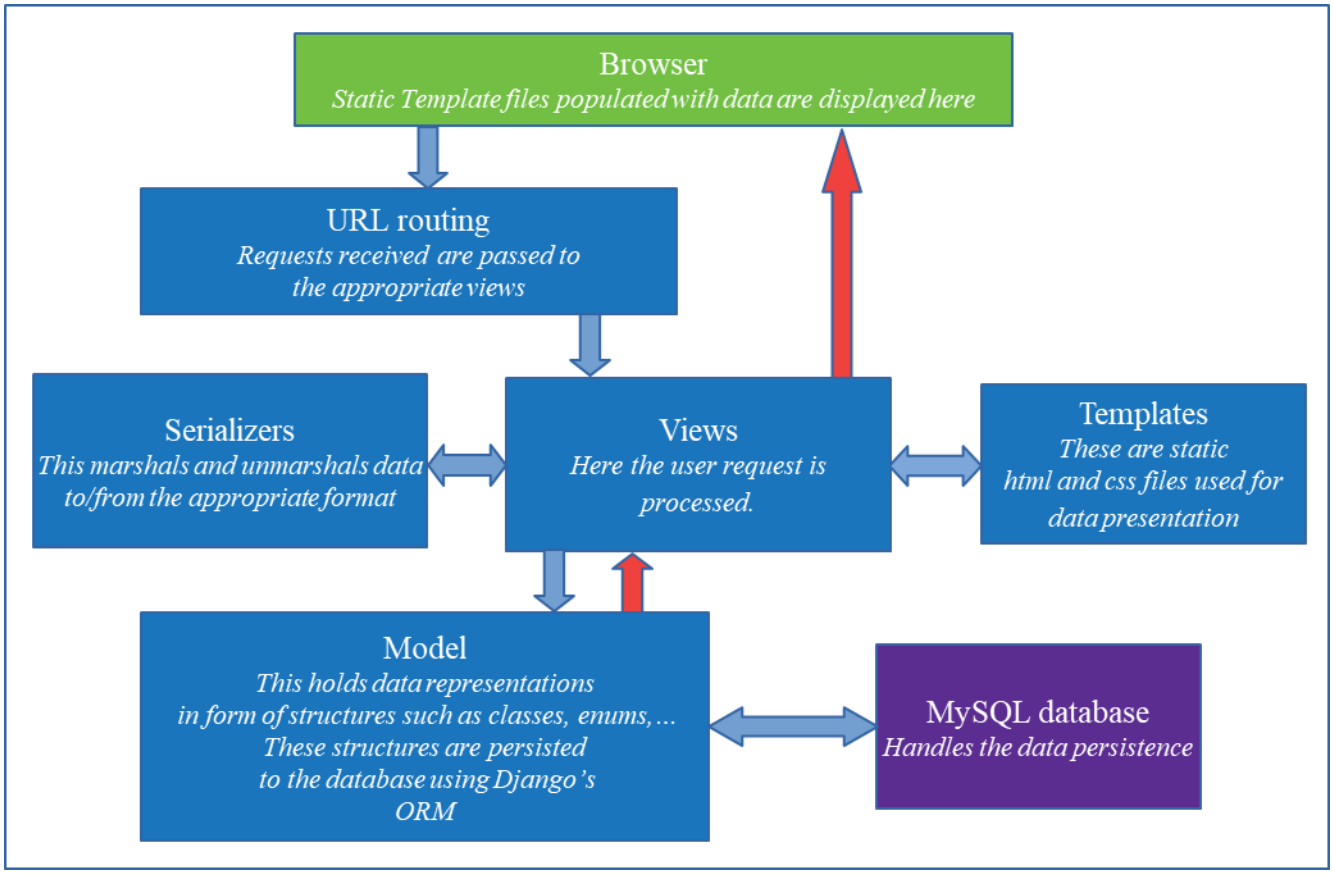


Fig 3.8. PRIMO-Complexes Django web server software architecture. The Django web server consists of 5 components (URL routing, Serializers, Views, Templates, and Model module) and the MySQL database used for data storage.

The process by which jobs are submitted from PRIMO-Complexes to JMS is illustrated in Fig 3.9. Once jobs are submitted to the web interface by the user, parameters are processed and sent to the PRIMO-Complexes web server. PRIMO-Complexes then compiles the parameters into a request it sends to JMS. JMS sends the request together with its authentication details to the HPC cluster. JMS monitors the job execution and notifies PRIMO-Complexes once the job is complete. PRIMO-Complexes web server then sends the results back to the web interface.

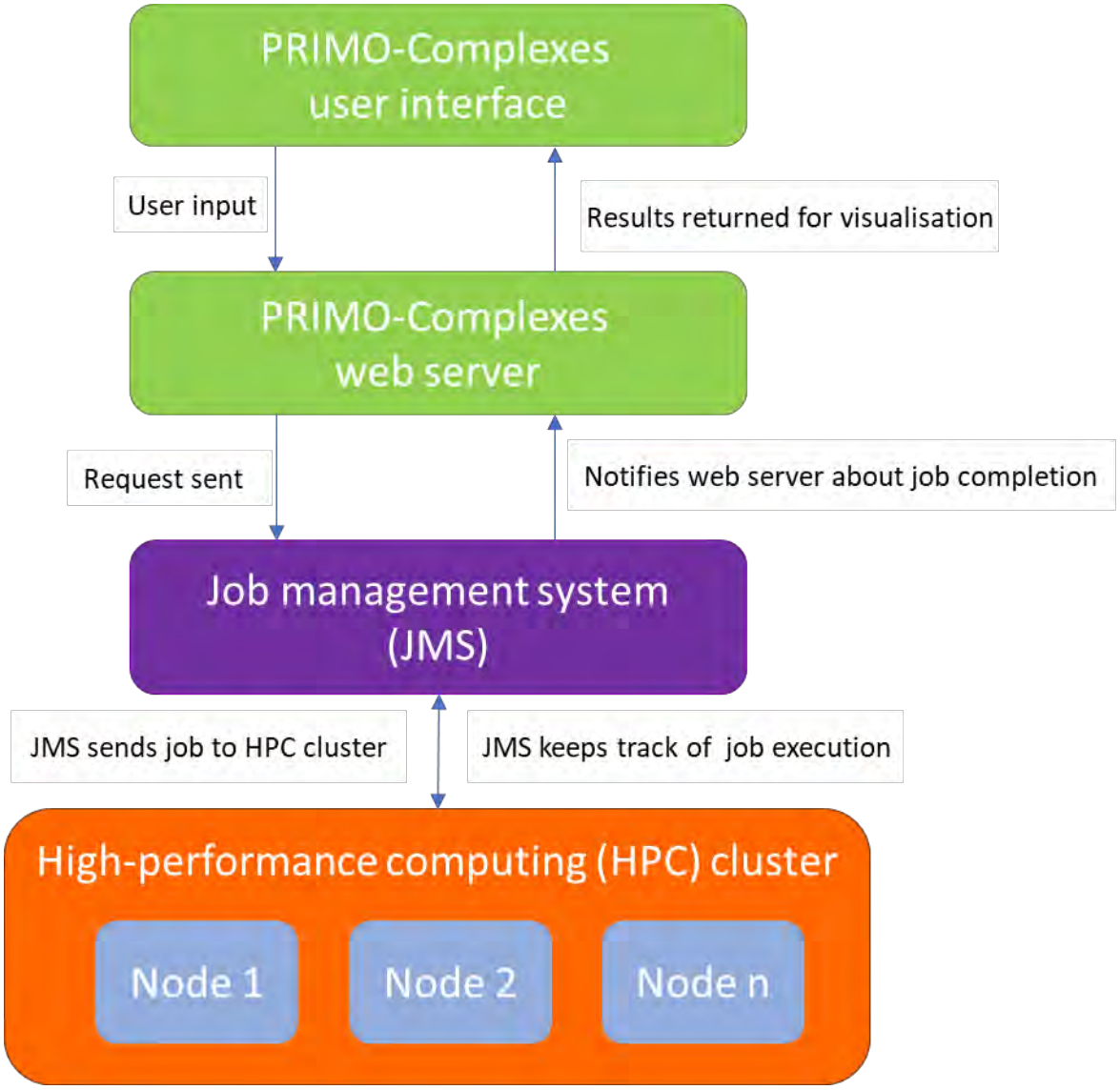


Fig 3.9. Execution of jobs submitted to PRIMO-Complexes web server. The figure shows the process of submitting jobs via JMS from the PRIMO-Complexes web server.

3.7.3 Job Input

The PRIMO-Complexes front-end is a single web page. A single page negates the need to reload each page every time a job is completed. The first page allows users to provide details about their job and optional input regarding all the modeling stages (Fig 3.10). PRIMO-Complexes web server is run sequentially providing user flexibility between steps.

Users are required to provide information to PRIMO-Complexes to necessitate the modeling process as described below. Users must provide a MODELLER key since this pipeline uses MODELLER [119]. A protein amino acid sequence(s) (one-letter sequence(s) or FASTA

format) can be specified directly or by uploading a file to PRIMO-Complexes. Sample sequences are provided to the user which they can use to familiarise with or test the web server. Optional input is also provided by PRIMO-Complexes. This optional input allows users to adjust the template identification, alignment and modeling step parameters. Users can choose to identify templates either using protein BLAST [122] or HHsearch [133]. Moreover, they can select one of the four alignment options provided, including Clustal-Omega [167], MAFFT [136], MUSCLE [135], and T-Coffee [137]. Furthermore, modeling parameters can be specified before the process begins. All the input details for each stage are submitted to JMS [208], a cluster hosted by the RUBi group. Users are then directed through each step of the protein modeling pipeline. If no optional input is specified, then default parameters are used for modeling.

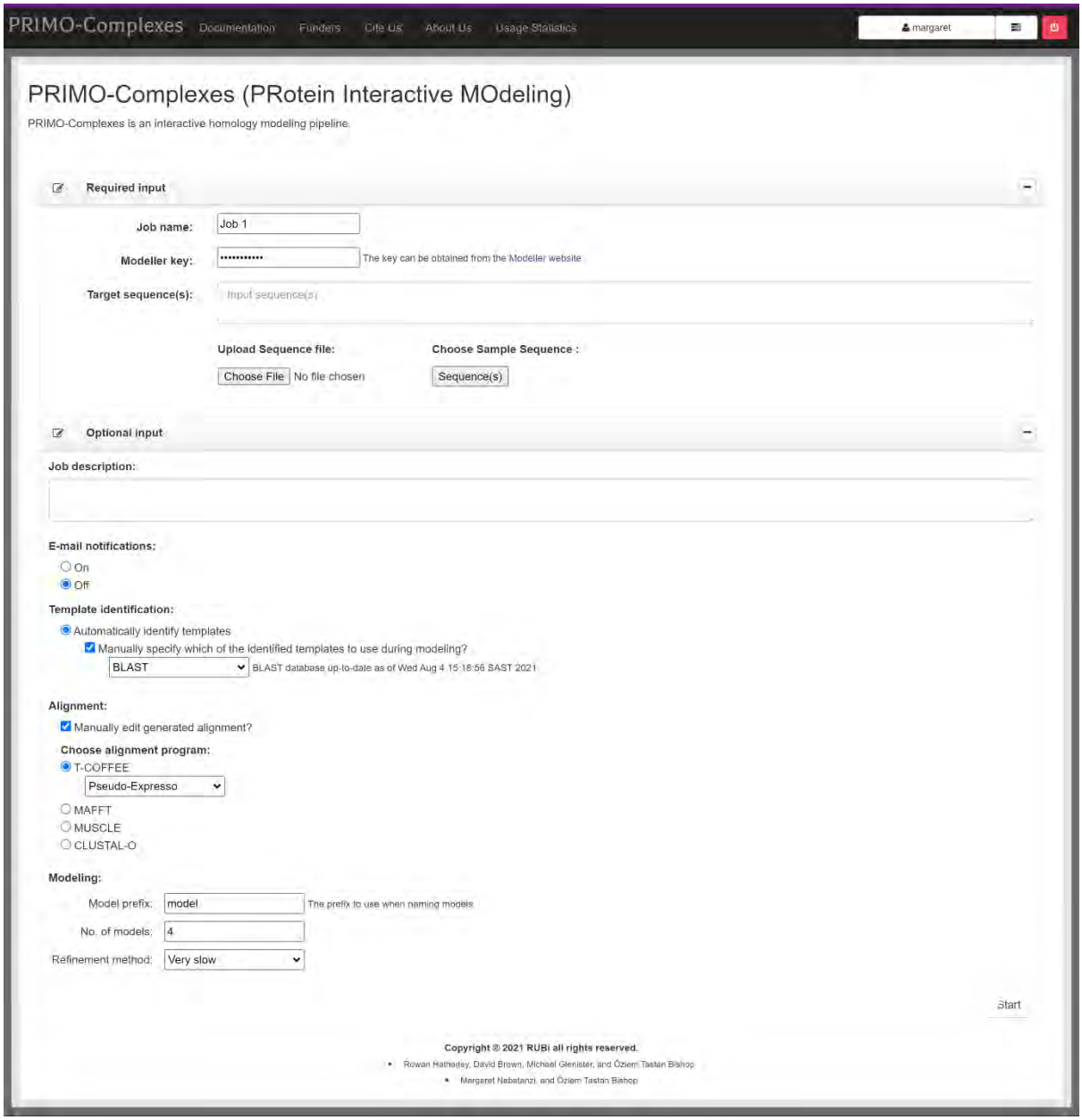


Fig 3.10. PRIMO-Complexes home page. This page contains the login/sign up icon, required input section, and optional input section.

3.7.4 Job handling – process and outputs

3.7.4.1 Template identification

Suitable templates are displayed in a tabular form, sorted according to the BLAST [122] or HHsearch [133] algorithm ranking (Fig 3.11, Fig 3.12, and Fig 3.13). Information including oligomeric state, sequence identities and coverage for each protein chain is displayed for a quick overview of the templates. In cases where more than one functional state exists for a

given template, these states are displayed in a drop-down menu. The user selects a template depending on the intended application and the functional state of the model. For instance, a model can be modeled in complex with a ligand or its apo form. In addition to the tabular form, an interactive 3D viewer (PV) displays the structure of the selected template. A corresponding alignment of only the hit chains is displayed for the selected protein structure to guide the user when assessing the best template based on the query coverage.

PRIMO-Complexes provides the functionality to model biological assemblies in case they exist for both homomultimeric and heteromeric proteins. An asymmetric unit is modeled if there is no existing biological assembly. This page also provides the user with a navigation feature at the top of the page to navigate the different steps of each modeling job. Users have access to the previous jobs through the job history, run new jobs and alter parameters in parallel during the modeling process.

Fig 3.11. Template identification results page for a homomultimeric protein. This figure presents the results page after searching for homologous templates to GTP cyclohydrolase 1 protein. A) The results page displays information on all homologous templates and a PV-MSA viewer is presented on the right side. B) The drop-down arrow displays a table for other existing biological assemblies. C) The ligand dialog window displays all ligands and ions present in the template structure.

The user can visualize and select ligands and/or ions for each chain in the protein complex template structure to be modeled in the target protein complex. A drop-down is provided in case there exists more than one biological assembly with information for each assembly. A continue button is activated after the user selects a template structure for use in modeling to continue to the next stage of the modeling process.

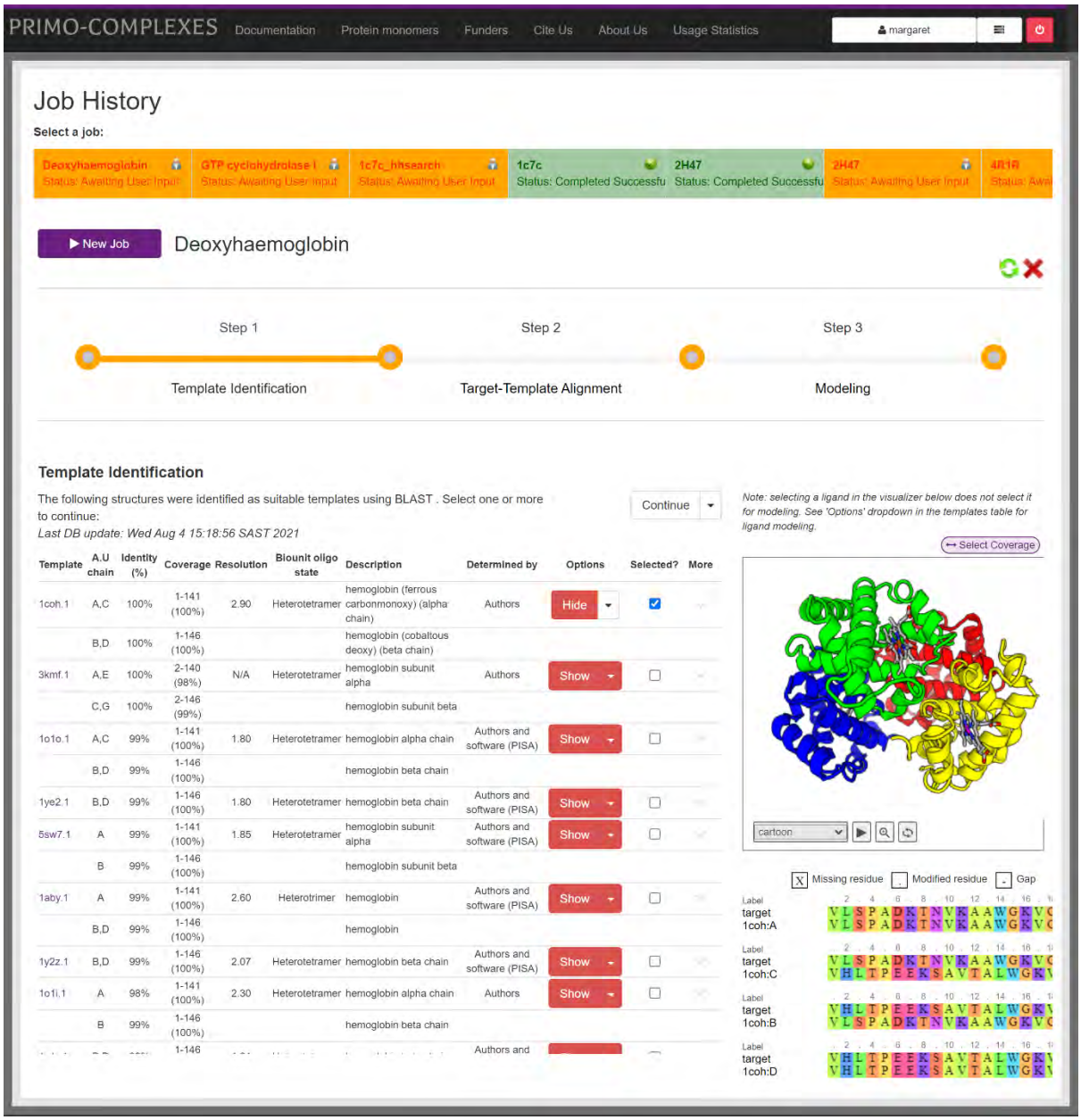


Fig 3.12. Template identification results page for a heteromultimeric protein. The results page after searching for homologous templates to deoxyhaemoglobin protein is presented.

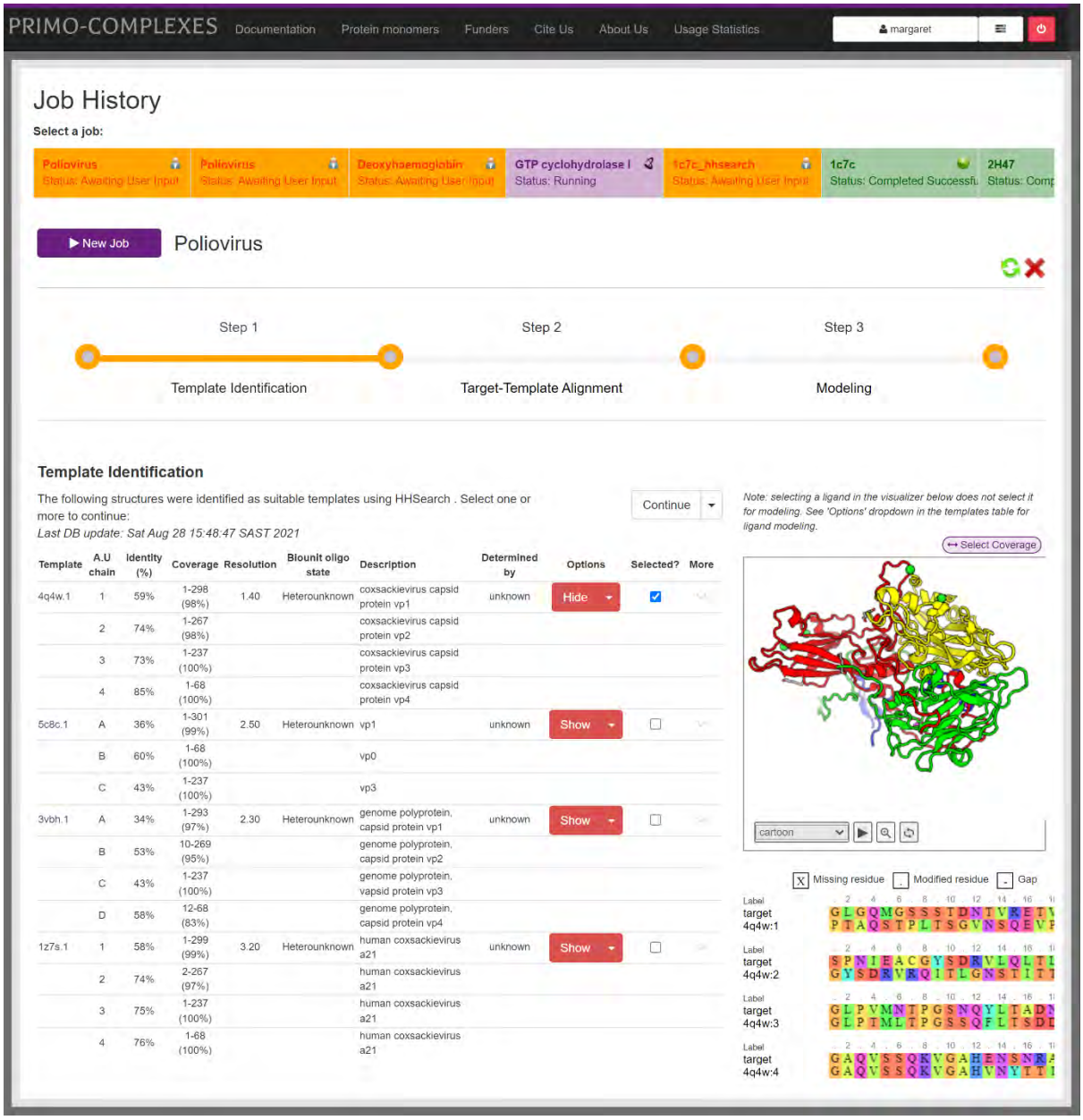


Fig 3.13. Template identification results page for a heteromultimeric protein - viral capsid. The results page after searching for homologous templates to a poliovirus protein (large macromolecules) is displayed.

3.7.4.2 Target-template sequence alignment

Sequences for hit chains are extracted from templates and aligned to their homologous target chains according to the alignment option chosen by the user (Fig 2.10, Fig 2.11, Fig 2.12). The alignment options include Clustal-Omega [167], MUSCLE [135], MAFFT [136] and T-COFFEE [137]. The chains are renamed before use in the alignment in case the biological assembly template does not have unique chain identifiers. An integrated alignment viewer

(MSA plugin) was customised in PRIMO-Complexes to accommodate multimeric proteins. The MSA plugin allows users to edit and export the alignment. Each alignment pair is edited separately for a heteromultimeric target protein. Amino acids and gaps ('-') can be added to the alignment if they are valid. The sequence alignment can only be trimmed at the edges and then validated against the original sequences.

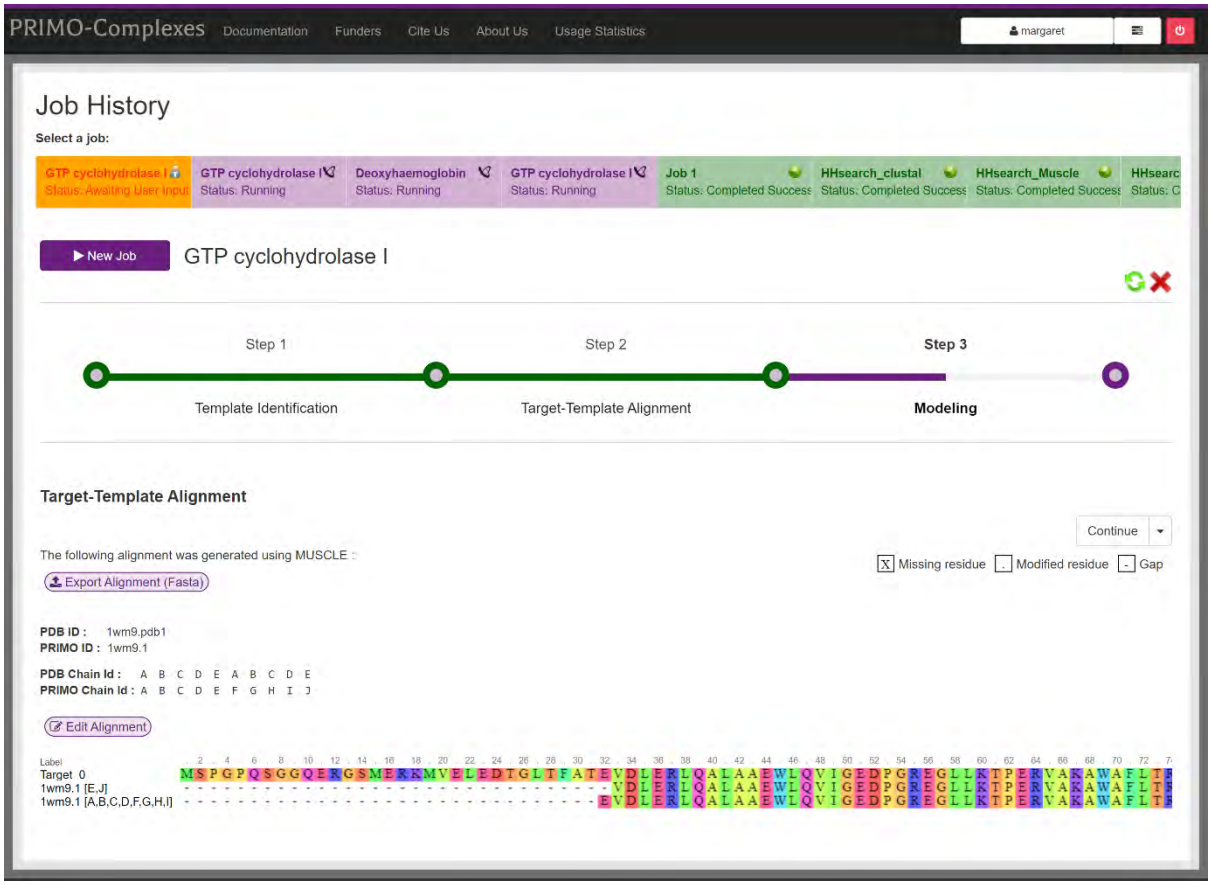


Fig 3.14. Target-template alignment results page for a homomultimeric protein. The figure presents the results page after the alignment of the target sequence with the template sequences for the selected template (template ID: 1wm9).

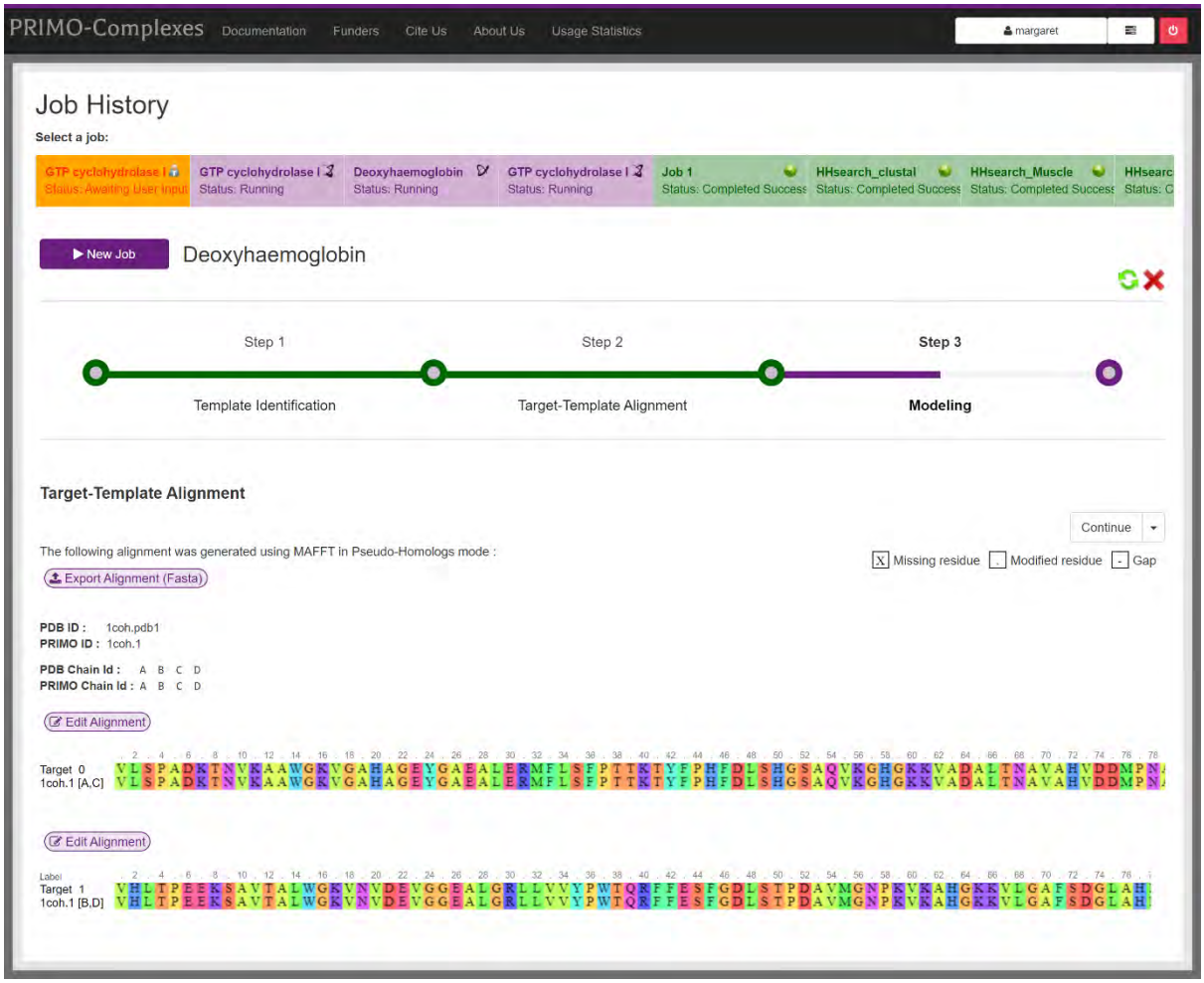


Fig 3.15. Target-template alignment results page for a heteromultimeric protein. The figure shows the results page after aligning the target sequence with the template sequences for the selected template (template ID: 1coh).

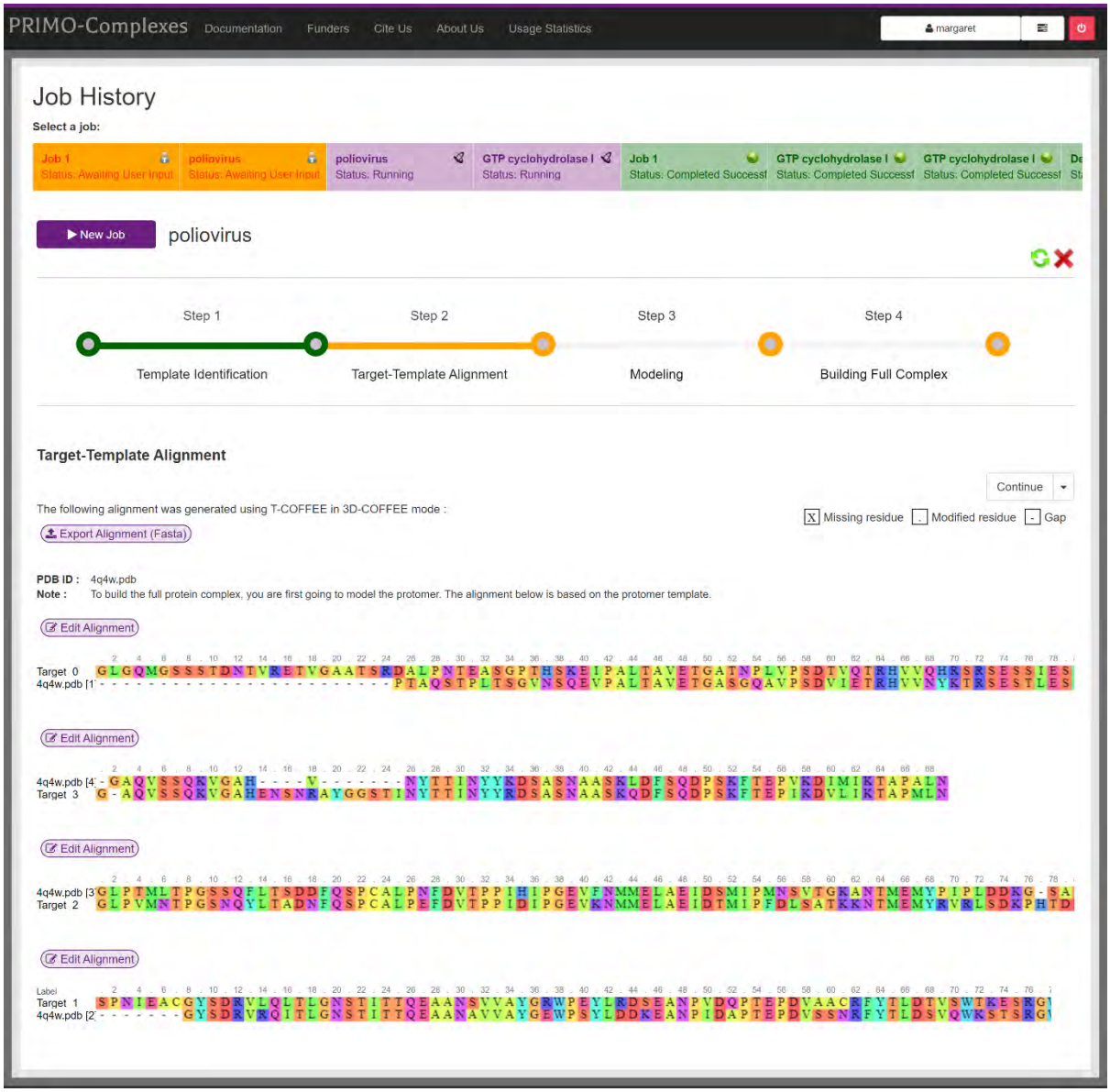


Fig 3.16. Target-template alignment results page for a heteromultimeric protein - viral capsid. The results page after aligning the target sequence with the template sequences for the selected template (template ID: 4q4w). A fourth step is added for building a large macromolecule.

3.7.4.3 Protein modeling and evaluation

A PIR file used for modeling is generated from the alignment step. The target protein is modeled using MODELLER [119] with the parameters specified in the template identification step. Quality of the top models are assessed by z-DOPE score and results are displayed in tabular form. Structures and alignment of these models can be visualised using PV-MSA viewer (Fig 2.13, Fig 2.15, Fig 2.16). Each model can be evaluated further using other

evaluation programs (such as ProSA [159], Verify3D [181], and QMEAN [186]) which are provided as links once the dropdown menu is selected. Models are assessed using PROCHECK tool [188] provided in the dropdown menu for each model (Fig 2.14).

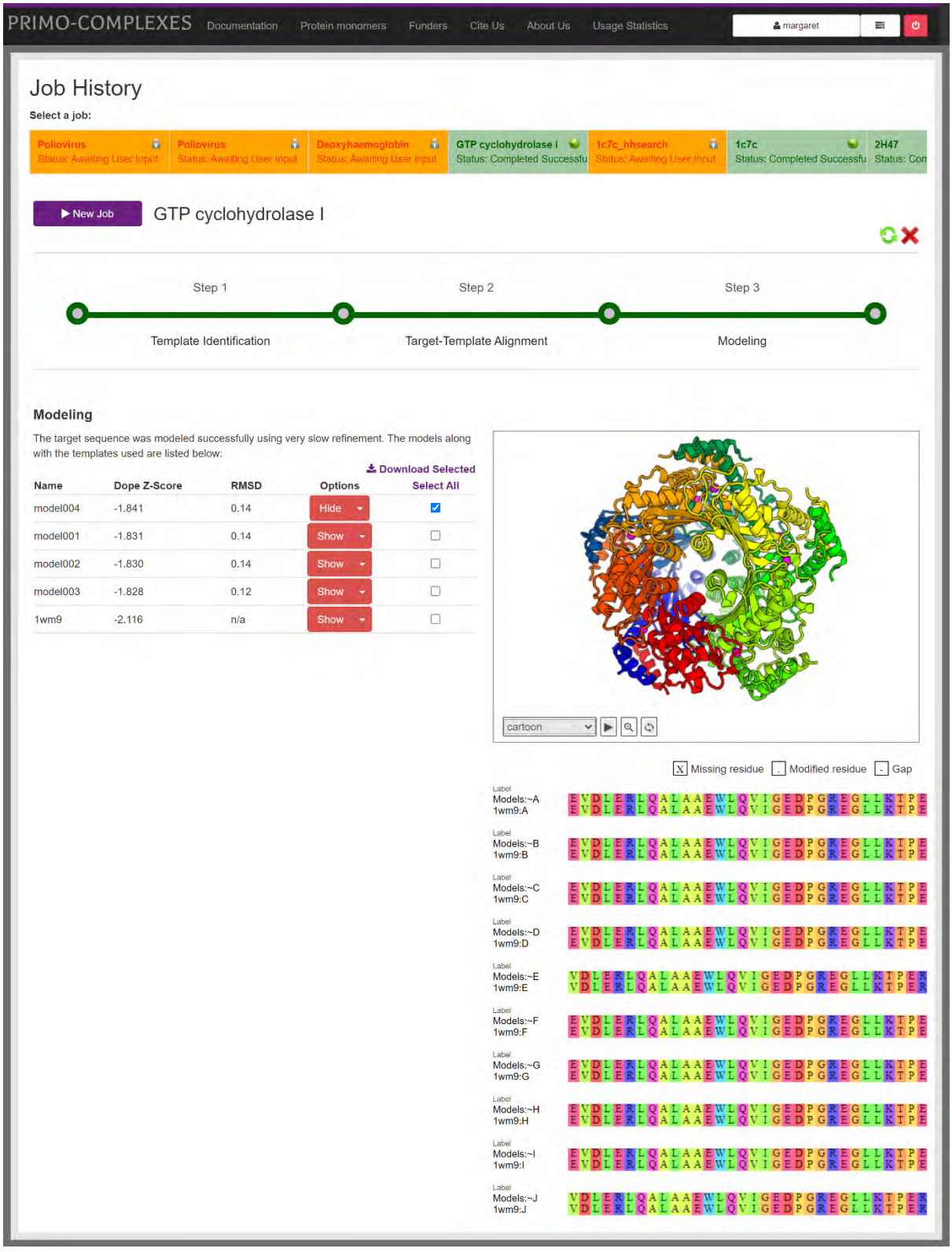


Fig 3.17. Protein modeling results page for a homomultimeric protein. The results for modeling the target protein from the selected template (template ID: 1wm9) are presented.

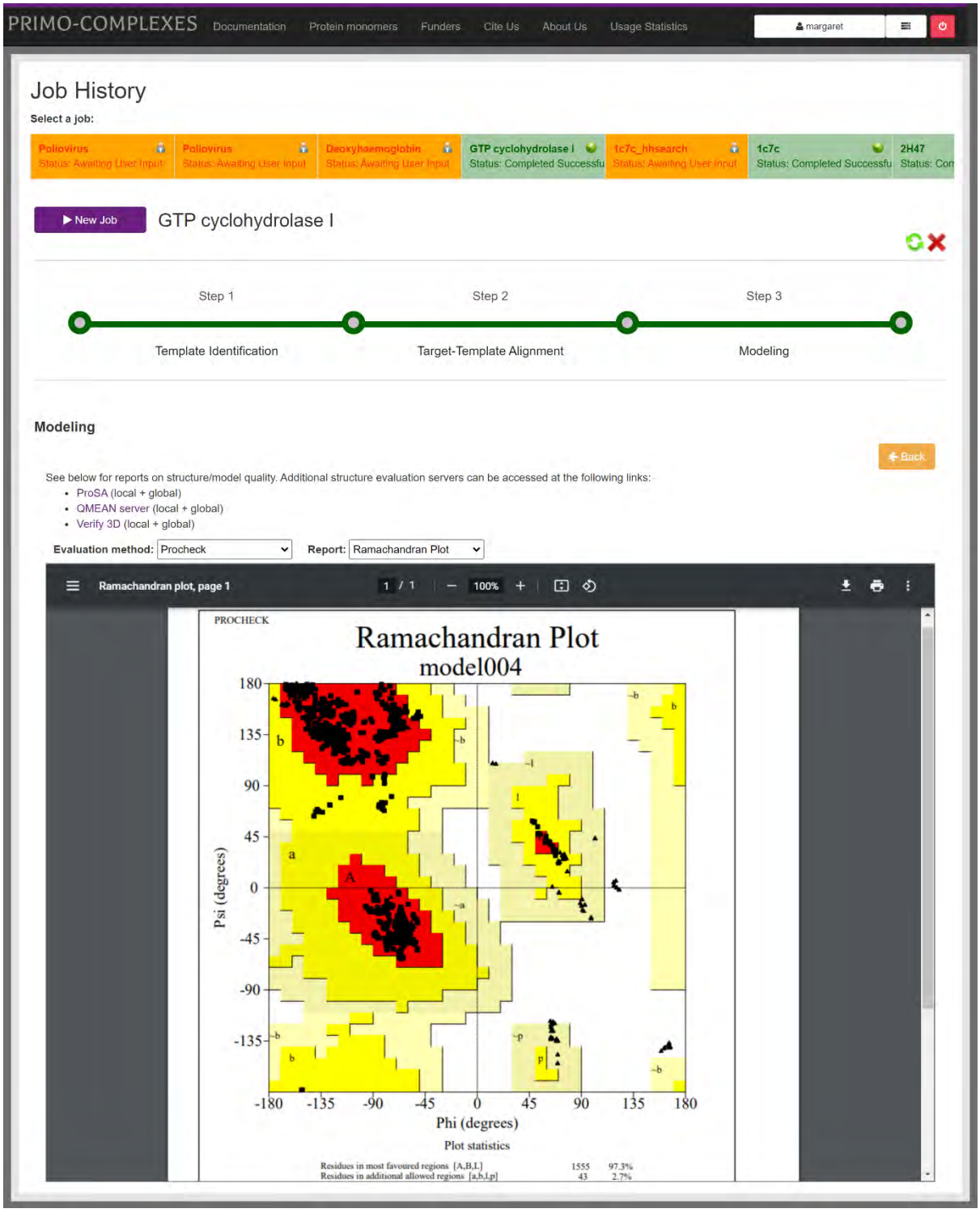


Fig 3.18. Protein model evaluation results page for a homomultimeric protein model (model004). The protein model evaluation results for model004 using PROCHECK tool are presented.

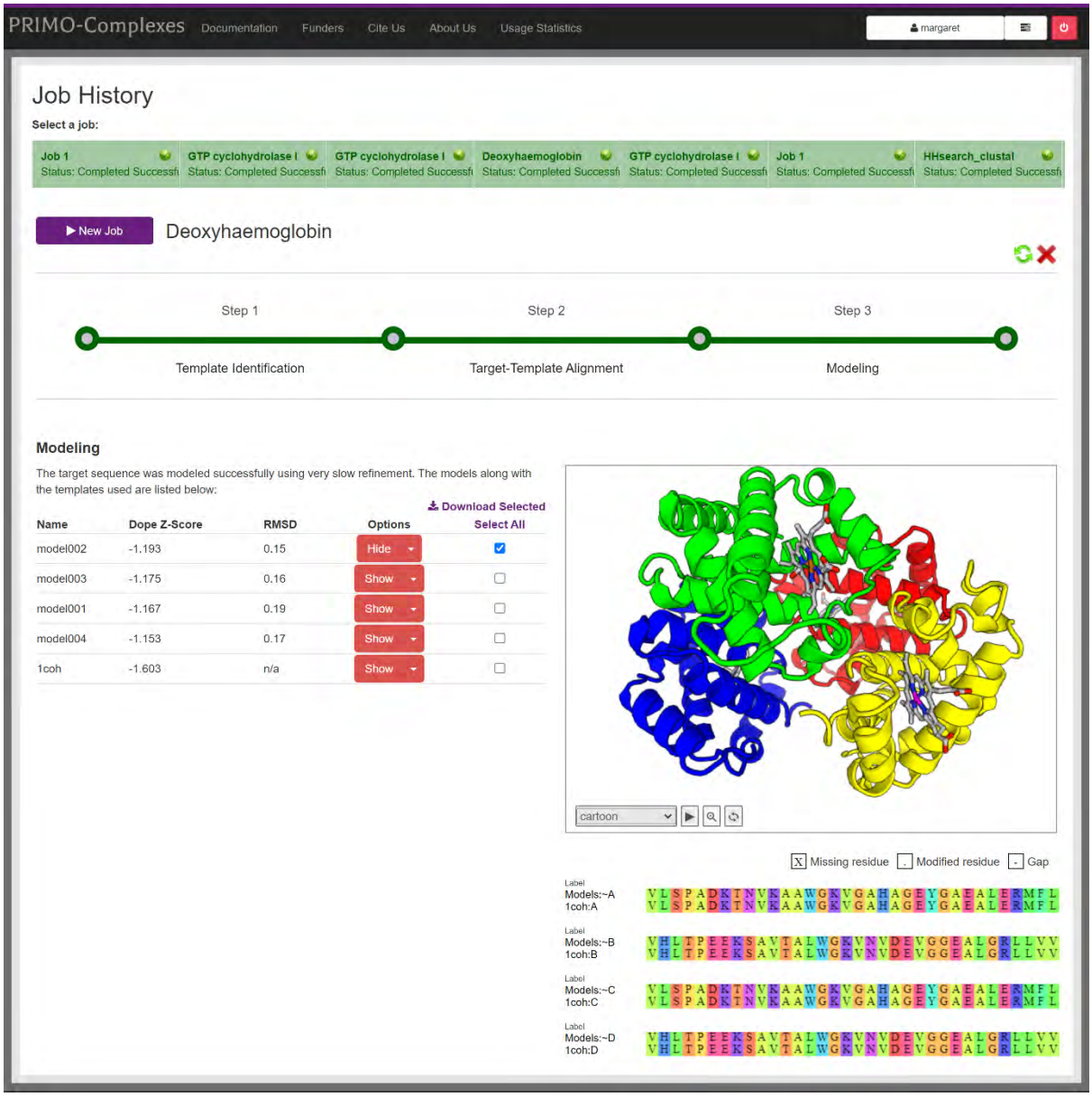


Fig 3.19. Protein modeling results page for a heteromultimeric protein. The results for modeling the target protein from the selected template (template ID: 1coh) are presented.

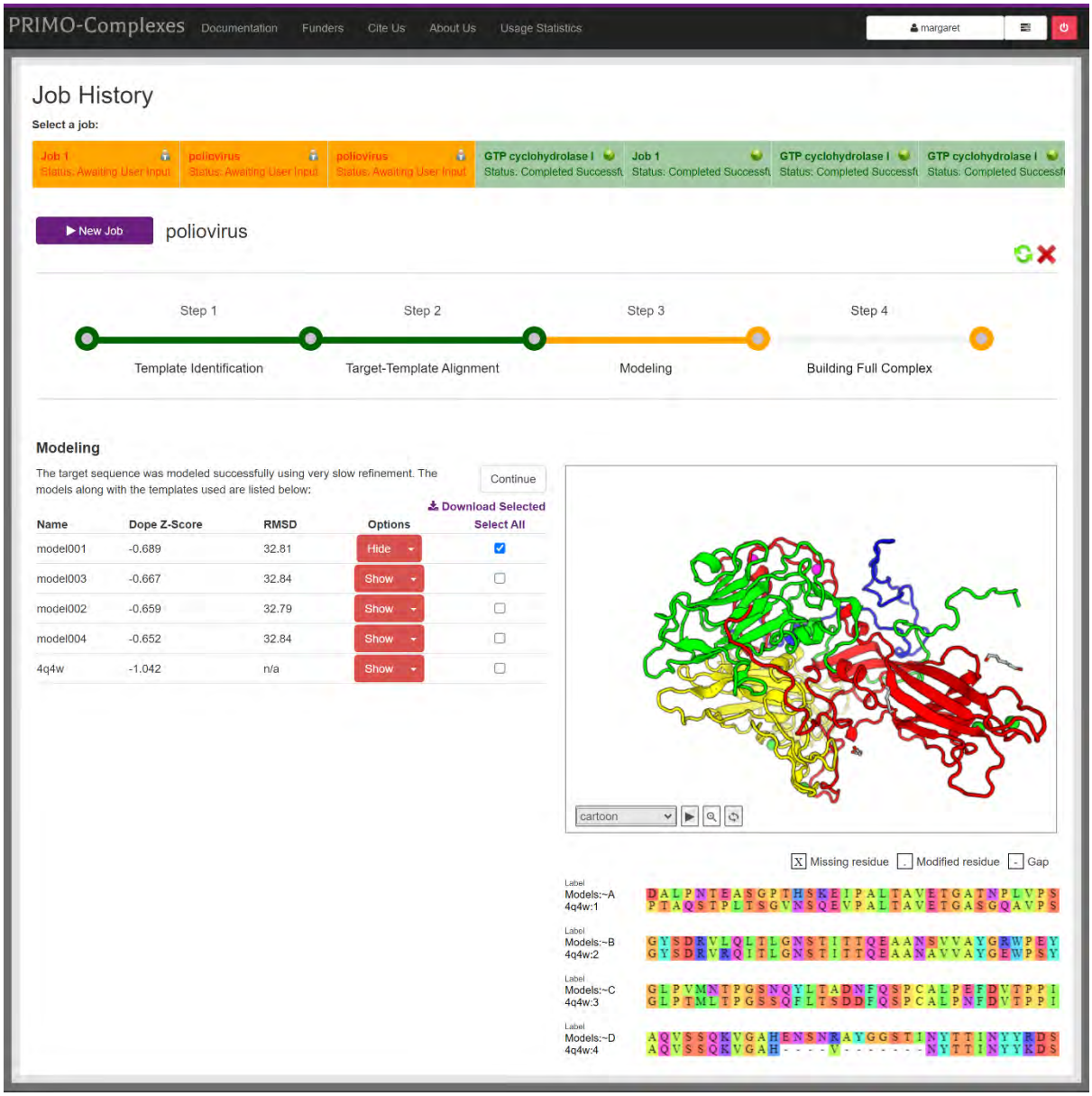


Fig 3.20. Protein modeling results page for a heteromultimeric protein - viral capsid. The results for modeling the target protein from the selected template (template ID: 4q4w) are presented in the figure.

3.7.4.4 Building large macromolecules

This step utilizes the best-selected model and the biological assembly template selected at the modeling stage and template identification step. This step uses a script that uses PyMOL [216] software to construct structures of the large macromolecule. This script superimposes copies of the best protein model on the respective protomers of the biological assembly template. The structures at this stage are displayed using NGL viewer (Fig 2.17).

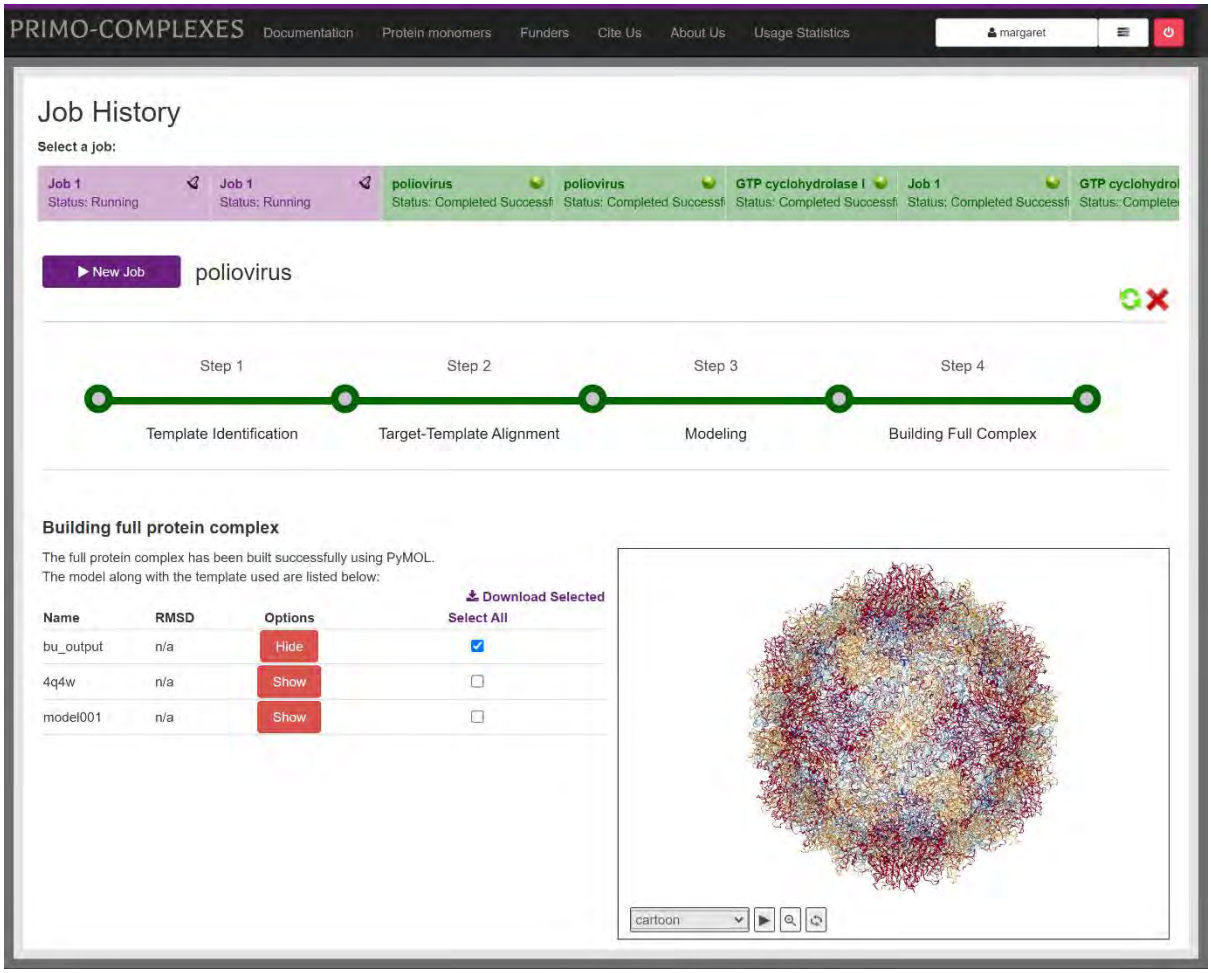


Fig 3.21. Protein building for large macromolecule for a heteromultimeric protein - viral capsid. The results for modeling full protein complex using the best-selected model (model001) are displayed.

3.8 Maintenance of PRIMO-Complexes

PRIMO-Complexes relies on data from the PDB, BLAST and HHsearch which needs to be regularly updated. This pipeline is or will be regularly updated to cater for the never-ending challenges on protein structure prediction including varying PDB file formats, obsolete data support and changing APIs.

Additional features including multiple template modeling, uploading preferred biological assembly template PDB files and specifying chain IDs will be added in future to make the process of protein complex modeling more intuitive.

3.9 Comparison between the first PRIMO and PRIMO-Complexes

This section highlights the differences between the first PRIMO and PRIMO-Complexes as shown in Table 3.1 below.

Table 3.1. Comparison between PRIMO and PRIMO-Complexes

Comparison features	**PRIMO**	**PRIMO-Complexes**
Functionality	Builds protein monomers	Builds protein complexes and biological assemblies
Required Input sequence	Single chain sequence	• Single chain sequence for homomultimeric proteins • two or more chain sequences for heteromultimeric proteins
Job handling process	Three homology modeling stages (template identification, target-template sequence alignment and protein modeling and evaluation)	Four homology modeling stages (template identification, target-template sequence alignment, protein modeling and evaluation and building large macromolecules)
Web interface • Visualisation • Job history panel • Results	• MSA viewer customised for protein monomers • PV-MSA viewer customised to display a single protein structure as well as a single alignment pair • Job history panel is vertically positioned on the left of the results section • Template identification results table contains asymmetric unit information such as PDB ID, chain, identity, coverage, and resolution	• MSA viewer customised for multimeric proteins • PV-MSA viewer customised to display a multimeric protein structure as well as several alignment pairs for every matching chain • Job history panel is horizontally positioned on top of the results section • Template identification results table contains either biological assembly if available or asymmetric unit

		information such as biological assembly PDB ID, asymmetric unit chains, chain identities, chain coverage, resolution, biounit oligomeric state, description and how they are determined

3.10 Strengths of PRIMO-Complexes web server

The PRIMO-Complexes web server allows users to interact and be involved in the modeling process. Various protein structure prediction web servers that model protein complexes have been reported previously; however, PRIMO-Complexes is unique in that it gives the user control over the modeling process. Notably, PRIMO-Complexes allows users to alter modeling parameters, navigate the different modeling stages and rerun jobs. Moreover, unlike other existing web servers, it has functionality for performing protein-ligand complex modeling by allowing users to choose ligands for each chain.

Given that this web server was developed using Django, it is believed that it will be secure as well as flexible in case of any changes or further development. PRIMO-Complexes web server is also not liable to full system failure since the main tools which perform multimeric protein modeling are stored and executed on a separate standalone HPC cluster. The PRIMO-Complexes web interface was developed as a single page application, with one index page loaded once at start-up where its data is refreshed as required. Different sections with information are loaded as the user interacts with the website after each stage is completed.

3.11 Limitations of PRIMO-Complexes web server

PRIMO-Complexes has some limitations including the fact that it only models using a single template and uses PDB file formats as templates. PRIMO-Complexes at the moment does not support multiple template modeling given the complexity involved when using more than one multimeric protein. Time constraints and the need to first develop the standard modeling process which can be extended later are also reasons not to include multiple template modeling. To overcome the limitations of legacy PDB file format, wwPDB resorted to using PDBx/mmCIF format. The PDBx/mmCIF format supports large structures, new and hybrid experimental structure prediction methods. PRIMO-Complexes cannot model using the PDBx/mmCIF format, yet some structures can only be represented in this format hence leaving out these structures when identifying templates.

3.12 Conclusion and recommendations

We successfully developed a new web server – PRIMO-Complexes – for modeling protein complexes and biological assemblies. The new web server is complementary to the first PRIMO web server that models monomeric proteins.

By using the Django web framework, we ensured that PRIMO-Complexes uses a modular architecture that is efficient and easy to maintain. Each component logically executes the shared functionality such as user requests, routing the URL requests, configuring data formats, and facilitating the data access process. The algorithms we incorporated in the PRIMO-Complexes webserver provide smooth functionality from the entry of sequences through modeling and visualisation of multimeric proteins.

CHAPTER FOUR

EVALUATION OF THE ACCURACY OF PRIMO-COMPLEXES TO MODEL MULTIMERIC PROTEINS

Chapter overview

Following completion of the components of the PRIMO-Complexes web server, we needed to evaluate its performance. This chapter focuses on describing the process and results from testing the accuracy and performance of the modeling algorithms used to develop the functionality of the PRIMO-Complexes web server.

Overview of the evaluation procedures

The process of evaluation involved the following steps: 1) generation of a dataset of target multimeric protein structures; 2) identification of templates using HHpred; 3) selection of protein templates and clustering them into bins; 4) perform sequence alignment; 5) model multimeric proteins; 6) Filter and evaluate models. A dataset consisting of PDB files from the Protein Data Bank was modeled through all stages of this pipeline including template identification, target-template sequence alignment, and protein modeling. In this evaluation the fourth stage of the PRIMO-Complexes web server was not included in the testing. This is because it involves a different modeling approach. Stages and parameters ran include HHsearch for template identification, four alignment options (MAFFT, MUSCLE, Clustal-O, 3D-Coffee), and modeling using slow, fast and no refinement. Finally, all the generated models were evaluated using the z-DOPE score, Root Mean Squared Deviation (RMSD), Global Distance Test – High Accuracy (GDT - HA) score, Template Modeling (TM) score, and Local Distance Difference Test (lDDT) score.

.

4.1 Testing of multimeric protein modeling scripts embedded in PRIMO-Complexes

Benchmark tests were done to assess the performance of multimeric protein modeling scripts as well as to troubleshoot errors in JMS tools that execute the core functionality of the PRIMO-Complexes. The JMS tools include template identification, target-template alignment, modeling, and complex structure modeling. The multimeric protein modeling Python scripts were automatically run without user intervention. Python scripts were written to bypass frontend languages such as JavaScript and NodeJS to fetch and move inputs from one step to another through the testing pipeline. Testing was performed using protein complexes as the target proteins. Fig 4.1 summarizes the steps used to perform the benchmark tests.

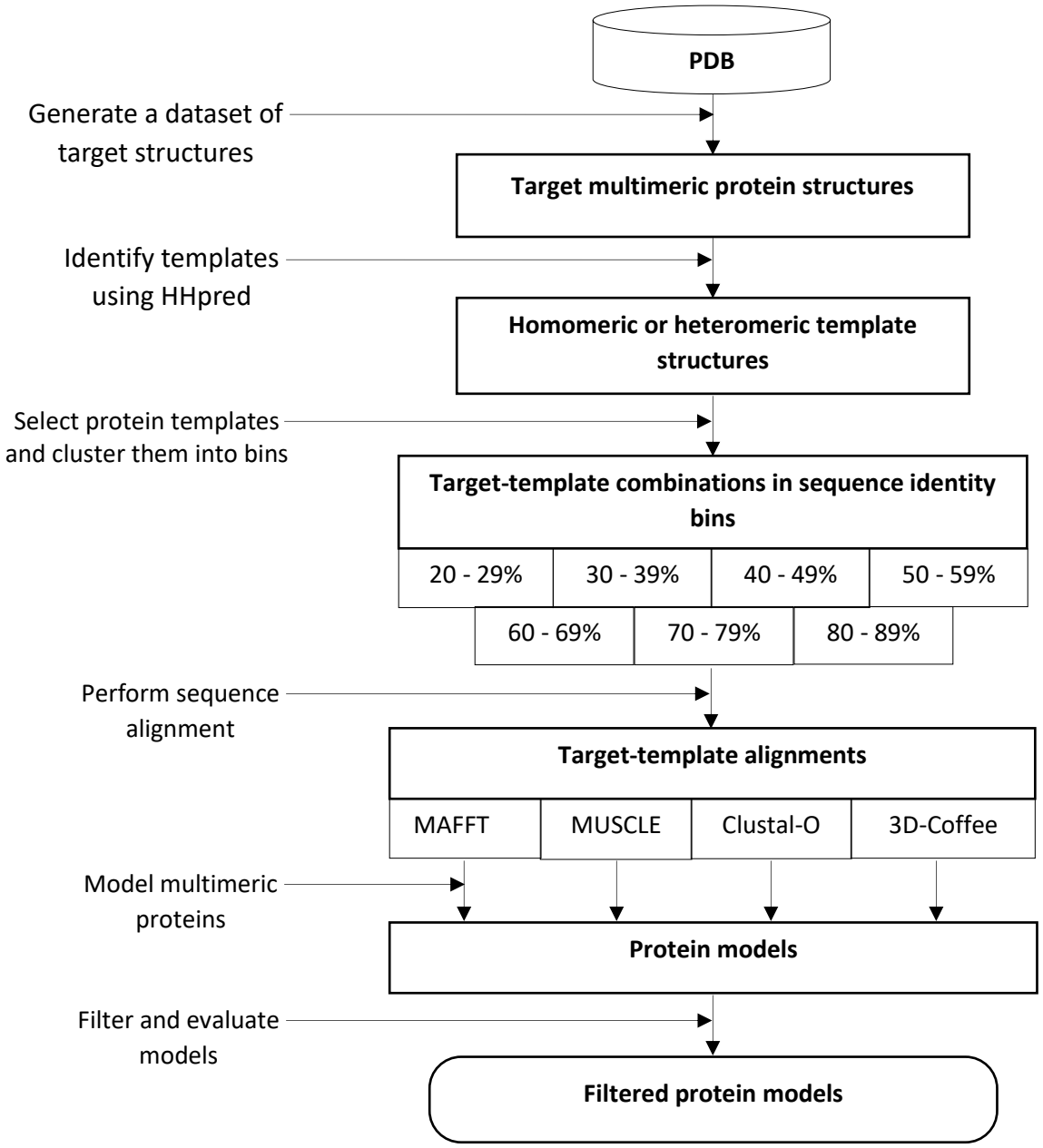


Fig 4. 1. A workflow diagram for testing the multimeric protein modeling Python scripts of PRIMO-Complexes. Overview of the steps followed when modeling known multimeric protein targets from the PDB.

4.1.1 Dataset generation and template identification

Target protein complexes with known protein structures were randomly fetched from the PDB database [51] to be used in the testing process of backend scripts. The dataset consisted of homomeric and heteromeric protein complexes. As of the year 2020, the total number of structures deposited in the PDB was 172,988 and of those 151,612 were protein-only structures

[58]. At the time of testing (June 2020), only protein structures with an existing biological assembly were considered (Table 4.1) before finally deciding on the oligomeric states to consider in this work. The total number of deposited biological assemblies in the PDB were 55,832 homomeric proteins, 19,805 heteromeric proteins and 71,443 monomers.

Table 4.1. Summary of the first twenty protein oligomeric states for biological assemblies available in the PDB as of June 2020.

Oligomeric state	Homomultimers	Heteromultimers
Dimer (2)	35,189	9,773
Trimer (3)	4,364	2,876
Tetramer (4)	9,698	3,116
Pentamer (5)	761	283
Hexamer (6)	2,752	1,441
Heptamer (7)	129	127
Octamer (8)	813	428
Nonamer (9)	34	174
Decamer (10)	337	90
Undecamer (11)	25	33
Dodecamer (12)	619	536
Tridecamer (13)	14	71
Tetradecamer (14)	124	84
Pentadecamer (15)	22	41
Hexadecamer (16)	82	108
Heptadecamer (17)	1	15
Octadecamer (18)	57	85
Nonadecamer (19)	2	11
Eicosamer (20)	22	65

Two sets of target protein complexes were used including homomultimers and heteromultimers. From the table above, it was decided that dimers and tetramers be used since the data was large enough to randomly sample from. Oligomers consisting of only protein macromolecules were downloaded and used in this evaluation (Table 4.1). Several filters were employed to aid the cleaning of the downloaded PDB protein structures. This was done to eliminate:

1. Proteins with chains whose length is less than 20 residues
2. Protein-DNA bound complexes
3. Proteins whose entries are divided between multiple coordinate files due to the limitations of the PDB file format for example TAR files containing a collection of minimal PDB files
4. Proteins in the format of mmCIF, and PDBx
5. Proteins marked as hetero-oligomers due to interactions with short peptides (below 20 amino acids) or antibodies
6. Proteins with existing biological assemblies

A Python script was written to parse sequences from randomly chosen PDB files in every chosen oligomeric state. HHsearch option was run to identify homologous multimeric proteins to these sequences. The HHsearch method was used because it is good at detecting remotely related protein structures [133]. Template structures were returned including biological assemblies that correspond to asymmetric unit template structures. Asymmetric unit PDB files are returned in case no biological assembly was found. An intermediate Python script was used to select only templates with the desired oligomeric state (homodimers, homotetramers, heterodimers, and heterotetramers) for the dataset being run. Since some proteins exist in more

than one functional state, the first biological assembly was selected if it matched the desired oligomeric state otherwise the next biological assembly was selected if it satisfied the criteria.

4.1.2 Clustering templates

The average sequence identity was recorded for each target-template combination for homomeric and heteromeric proteins during the template identification step. In this work, target-template sequence identities from 20% to 89% were considered. A total of seven sequence identity bins in intervals of 10 were created and used. Target-template combinations were binned according to these sequence identities. The binning process was repeated for homodimers, homotetramers heterodimers, and heterotetramers with 1285, 794, 450, and 150 different protein multimeric structures per bin respectively. Total target-template combination protein structures in all the seven bins included 8,995 homodimers, 5,558 homotetramers, 3,150 heterodimers, and 1,050 heterotetramers.

4.1.3 Target-template sequence alignment and model generation

Sequences for each target-template combination entry in each bin were aligned using four different programs embedded in PRIMO-Complexes. These include Clustal-Omega (Clustal-O) [167], MUSCLE [135], MAFFT [136] and 3D-Coffee [175]. Some Python scripts that mimic how these alignment programs work were written by Rowan Hatherley in the first version of PRIMO that models monomeric proteins [87]. A script was written to mimic how the MAFFT program runs MAFFT-homologs using a local version of protein BLAST [121]. It retrieves 50 closely related sequences to both the target and template sequences. Another script to mimic the Expresso [176] option in T-Coffee alignment program was also written. Expresso runs 3D-Coffee [175] which incorporates structural information in the alignment. Normally Expresso runs BLAST to identify homologous protein structures to be used by 3D-Coffee. However, in this work, Expresso uses template structures generated at the template

identification step excluding the target PDB. Alignments for each target-template combination entry were used by the multimeric protein modeling script to generate models. This script uses MODELLER [119] to build protein models and in this evaluation 10 models were generated per run using the very slow refinement option. As explained in detail in chapter 3, new scripts were written to perform multimeric modeling, and these include four tools run and executed by JMS namely template identification, target-template alignment, modeling, and complex structure modeling. These JMS tools rely on other PRIMO-Complexes scripts stored in the web server. Some of these Python scripts were completely written from scratch, others were adapted and modified while some were reused. Details of these scripts are described in chapter 3 above.

4.1.4 Normalizing and filtering protein models

Each target-template combination had differing lengths when modeled using the different alignment programs discussed in section 4.1.3 above. These differing lengths were due to sequence trimming during the PIR file preparation step. These trimmings are done at each end of each sequence to ensure that the targets being modeled have corresponding template sections to be modeled from. Models were normalized before being filtered. Model normalization involved trimming PDB files in each modeling set (target-template combination) to the longest common segment across similar chain identifiers.

Models went through a series of filtering steps as shown in Fig 4.2. After performing target-template alignment using four different alignment programs as described in section 4.1.3 above, some models fell outside their assigned sequence identity bins. The average sequence identity was calculated for both homomeric and heteromeric proteins. Whereas homomeric proteins have identical sequences, some chains may have mutated or missing residues thus the need to calculate the average for all sequence identities. Sequence identities were calculated from PIR files used during the modeling step. Initially, clustering each target-template

combination into different sequence identity bins depended on the target-template alignment generated during the template identification step. Only target-template combinations whose alignment for all the four alignment programs fell in the same bin were retained.

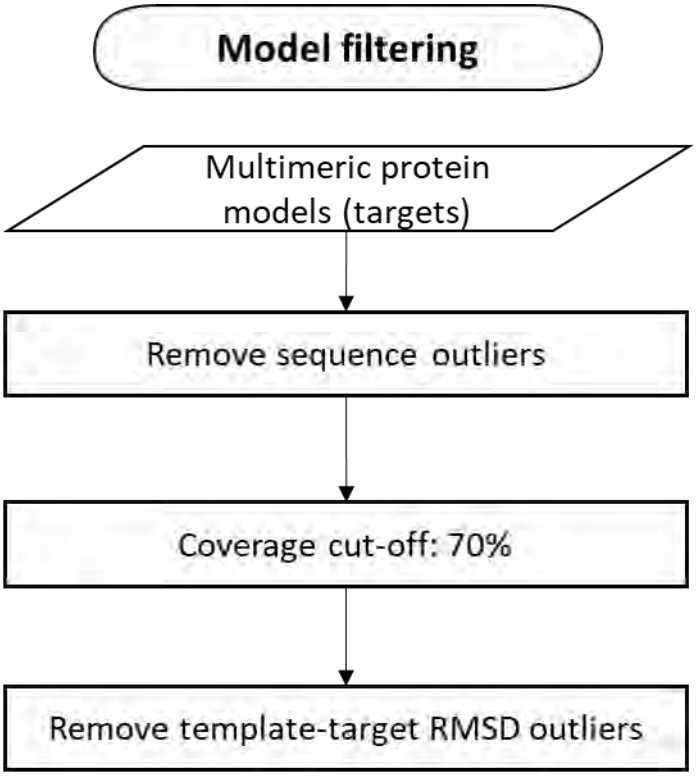


Fig 4.2. A workflow diagram showing steps followed to filter protein models. Filtering was performed before assessing models for each oligomeric state

The average target coverage for each target-template combination was also calculated. The target-template combination was included if the average coverage was at least 70% of the target sequences for all four alignment options. Finally, the calculation of the Root Mean Squared Deviation (RMSD) between each target and template PDB file was performed using BioPython to remove outliers in each sequence identity bin. This was done due to conformational changes between the target and template protein complexes where RMSD could not be used to assess the modeling accuracy.

4.2 Evaluation of the protein models

Evaluation of models was performed to assess their accuracy using different evaluation metrics. Key outcomes measures used in the evaluation:

1. z-DOPE score
2. Root Mean Squared Deviation (RMSD)
3. Local Distance Difference Test (lDDT) score
4. Template Modeling (TM) score
5. Global Distance Test – High Accuracy (GDT - HA) score

A z-DOPE score was calculated for each model for all alignment options in every sequence identity bin to generate the top model for each target-template combination. The top model and target PDB structure were then compared using distance-based measures including RMSD [152], Local Distance Difference Test (lDDT) score [220], Template Modeling (TM) score [221], and Global Distance Test – High Accuracy (GDT - HA) score [153]. Among these distance-based measures, RMSD, TM-score and GDT-HA score are global measures whereas lDDT score is a local measure.

RMSD is the most commonly used metric and measures the mean distance between corresponding atomic coordinates for two superposed structures [152]. RMSD values range from 0 to ∞. This measure is associated with some limitations including dependence on the protein length and is dominated by loops. To solve these problems, the TM-score was introduced as another similarity score.

TM-score is a metric measure that assesses similarity of the topological arrangement of protein structures. The TM-score value lies between (0,1) where 1 indicates that two structures are a perfect match [221].

Another global measure is the GDT score that quantifies the number of atoms in the model that can be superposed with corresponding structure atoms within stipulated thresholds. The GDT-HA score ranges from 0% to 100%. In this study, the GDT-HA score calculated is an average of conserved distances over thresholds of 0.5, 1, 2 and 4 Å [153].

lDDT is a local measure that assesses how well the local atomic interactions in a reference protein structure are reproduced in the predicted structure being evaluated [220]. The lDDT score ranges from 0 to 1 but in this work, the scores were converted to percentages (0 to 100) after the calculation. RMSD, TM-score and GDT-HA score calculations rely on the superposition technique to measure similarity whereas lDDT score is superposition-free.

Before calculating the lDDT, TM, and GDT-HA score, all the chains for each best model and target protein complex were sequentially renumbered from the first to the last chain in the PDB file. The GDT-HA and TM-scores were calculated using TM-score software (http://zhanglab.ccmb.med.umich.edu/TM-score/) and the lDDT score was calculated using OpenStructure/lDDT software (http://swissmodel.expasy.org/lDDT).

4.3 Additional assessment and evaluation performed

4.3.1 Remodeling target proteins and assessing PDB structures

Each target protein was remodeled using its very PDB structure as a template. This was done to represent ideal modeling conditions and get an idea of the error produced by MODELLER [119]. These models were then evaluated by calculating z-DOPE scores. The target and template PDB structures were also evaluated by calculating z-DOPE scores.

4.3.2 Testing model refinement parameters

MODELLER [119] provides other refinement options apart from very slow refinement. Performance tests were done on these model refinement options including none and fast refinement. Final datasets after filtering and evaluating models were used to perform model

refinement tests. The refinement level in the modeling script was altered to either fast refinement or no refinement (none). The modeling script used similar MAFFT PIR files that were generated during the alignment step in benchmark tests to perform modeling. Models generated from none, and fast refinement options were evaluated together with those initially modeled with very slow refinement.

4.3.2.1 Evaluation of model refinement results

A z-DOPE score was calculated for each model for all the alignment options in every sequence identity bin to generate the best model for each target-template combination. The best model and target PDB structure were then compared to generate the RMSD values.

4.4 Continuous assessment of the pipeline

The Continuous Automated Model EvaluatiOn (CAMEO) project provides an independent evaluation of structures generated by protein structure prediction web servers. This evaluation is done based on the criteria established by the protein structure prediction community [198].

To integrate PRIMO-Complexes with CAMEO [198], an endpoint was created in the jobs view script of the PRIMO-Complexes Django web server [87]. This endpoint handles requests from the CAMEO [198] server, it un-marshals the request payload and encodes data in a format that is consumable by the PRIMO-Complexes modeling web server. This endpoint was further extended to handle targets constituting multiple sequences. PRIMO-Complexes web server was designed to allow CAMEO perform testing in four different versions including combinations between template identification options (BLAST, HHsearch) and sequence alignment (3D-Coffee, Clustal-Omega) programs. The setup in PRIMO-Complexes was done and completed and will be registered with CAMEO to begin continuous assessment.

PRIMO-Complexes requires that every job handled be associated with a user account, as such, a dummy user account was created to represent CAMEO requests. Once a request is received

on the CAMEO endpoint, this user data is loaded and then associated with the newly created job object before passing it to the pipeline. CAMEO jobs are handled only in automatic mode; here no input is required from the user as stage transitions are performed. Once the final models are generated, the CAMEO system is contacted through the email provided in the request payload.

4.5 Case studies

To demonstrate the performance of PRIMO-Complexes when compared to other modeling web servers – SWISS-MODEL [88], and Robetta [201], three case studies were performed. These included modeling of: GTP cyclohydrolase I (GTP-CH-I) protein from *Escherichia coli* (accession number: P0A6T5) as a multimeric protein (biological assembly) with ligands; superoxide dismutase (hSod1) protein from *Homo sapiens* (accession number: P00441) with ligands; and hemoglobin subunit alpha (alpha-globin) from *Homo sapiens* (accession number: P69905). The SWISS-MODEL web server was run without user intervention and allowed it to select the best templates. The Robetta web server was run using the option to run only comparative modeling. The characteristics of these servers were described in chapter 2.

For GTP-CH-I, two modeling sets were chosen from PRIMO-Complexes; 1) Using biological assembly template 1wm9 (homodecamer), which is in complex with zinc ion for chains A, B, C, D and E; 2) Biological assembly template 1wur (homodecamer) with its inhibitor and metal binding. Chains A, C and D are in complex with 8-oxo-2'-deoxyguanosine-5'-triphosphate (PDB ID: 8DG) whereas chains A, B, C, D and E have zinc ion binding. Both were aligned using 3D-Coffee without edits to the alignments.

For hSod1, two modeling sets were chosen for PRIMO-Complexes; 1) Using biological assembly template 2xjk (monomer), which is in complex with a copper (II) ion and zinc ions. This was aligned using the MUSCLE program and no further intervention; 2) asymmetric unit

template 1dsw (monomer), which is in complex with copper (II) ion and zinc ion. This was aligned using the 3D-Coffee program and no further intervention.

For alpha-globin, one modeling set was chosen, using biological assembly template 1aby (heterotrimer), which is in complex with cyanide ion and protoporphyrin IX containing Fe. This was aligned using the 3D-Coffee program and no further intervention. All models were assessed using ProSA [159], and Verify3D [181], QMEAN [186], PROCHECK [188] and z-DOPE score [158].

4.6 Results and discussion

4.6.1 Protein modeling and filtering models

To evaluate algorithms used in PRIMO-Complexes, a study was performed to model protein complexes with known structures from the PDB [51]. Templates with sequence identities ranging between 20% to 89% were selected along with four different alignment approaches. After protein modeling, these models were filtered as described in section 4.1.4. The final set included 4,464 dimeric (Fig 4.3A) and 2,940 tetrameric (Fig 4.3B) modeled targets for homomeric proteins (Fig 4.3) whereas 858 dimeric (Fig 4.4A) and 244 tetrameric (Fig 4.4B) modeled targets were identified for heteromeric proteins (Fig 4.4). Given that 10 models were generated for each target-template combination entry (target), a total of 178,560 (dimers) and 117,600 (tetramers) models for homomeric proteins as well as 34,320 (dimers) and 9,760 (tetramers) models for heteromeric proteins were evaluated.

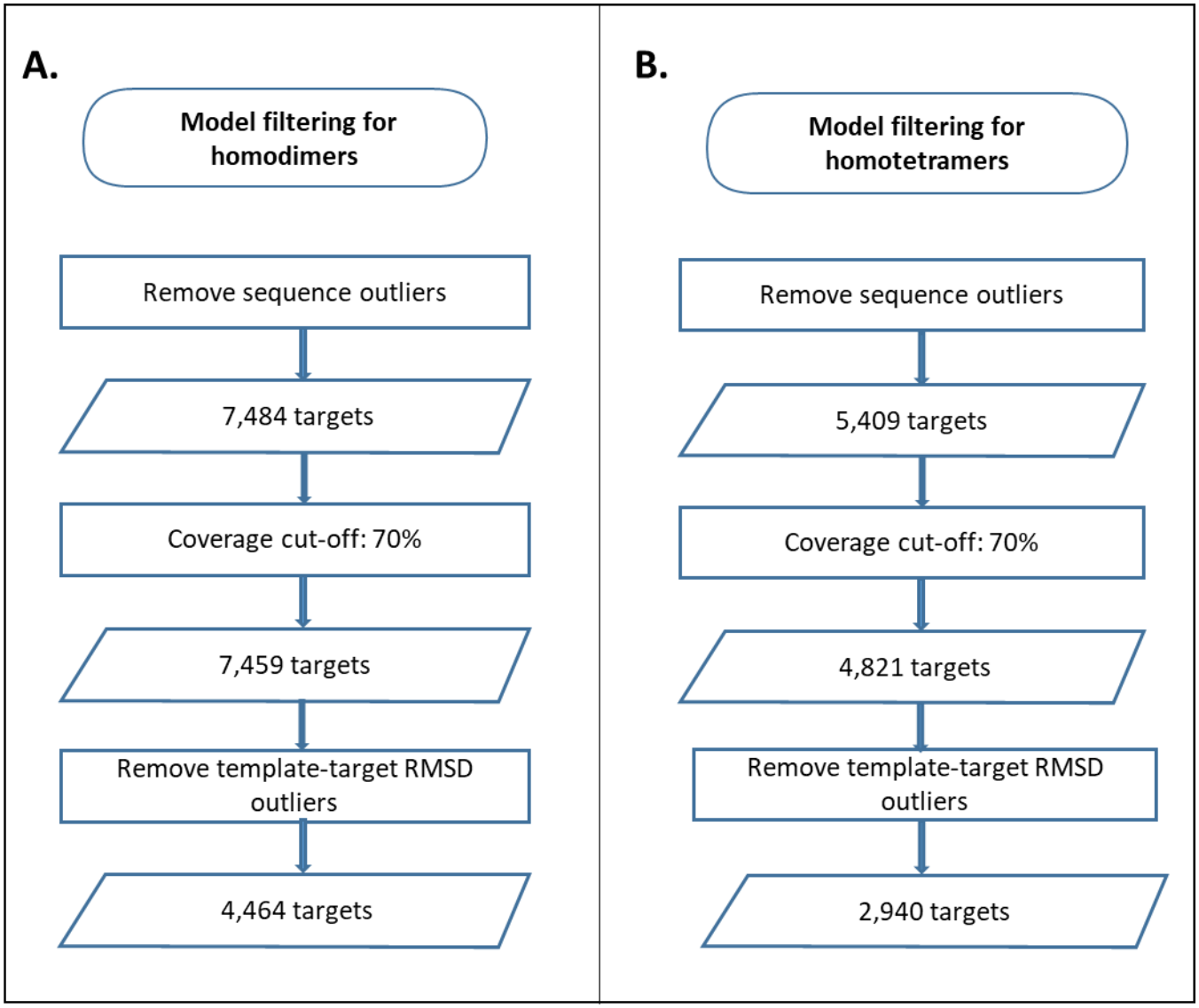


Fig 4.3. Final homomultimeric targets that remained after performing the filtering steps. A) The homodimer dataset reached 4,464 final targets. B) The homotetramer dataset reached 2,940 final targets.

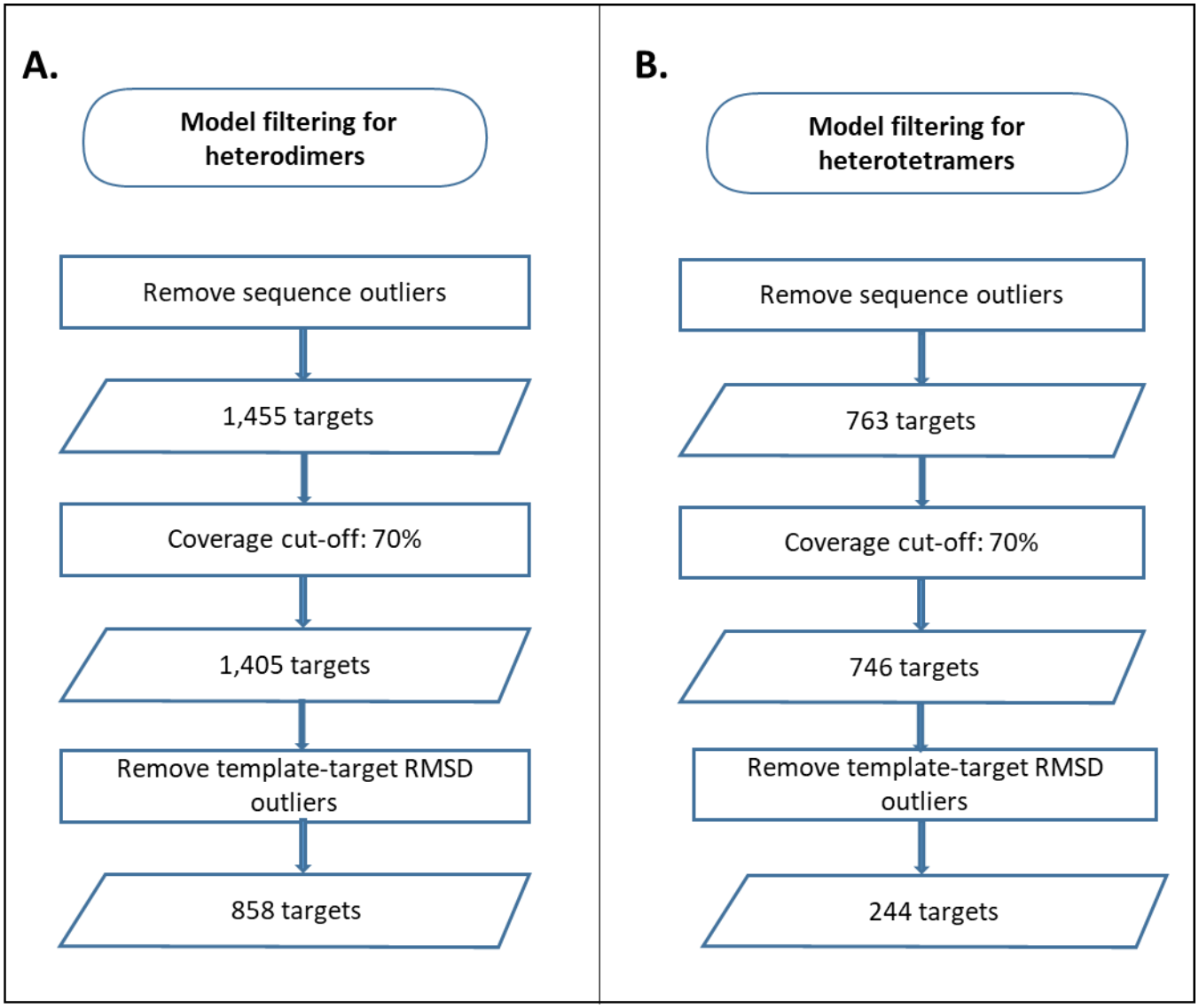


Fig 4.4. Final heteromultimeric targets that remained after performing the filtering steps. A) The heterodimer dataset reached 858 final targets. B) The heterotetramer dataset reached 244 final targets.

After target-template alignment was performed using four different alignment programs as described in the methods section, some models fell outside their assigned sequence identity bins (Fig 4.5). This realignment of the target and template sequences produced different results thus retaining only target-template combinations whose alignment for all the four alignment programs fell in the same bin.

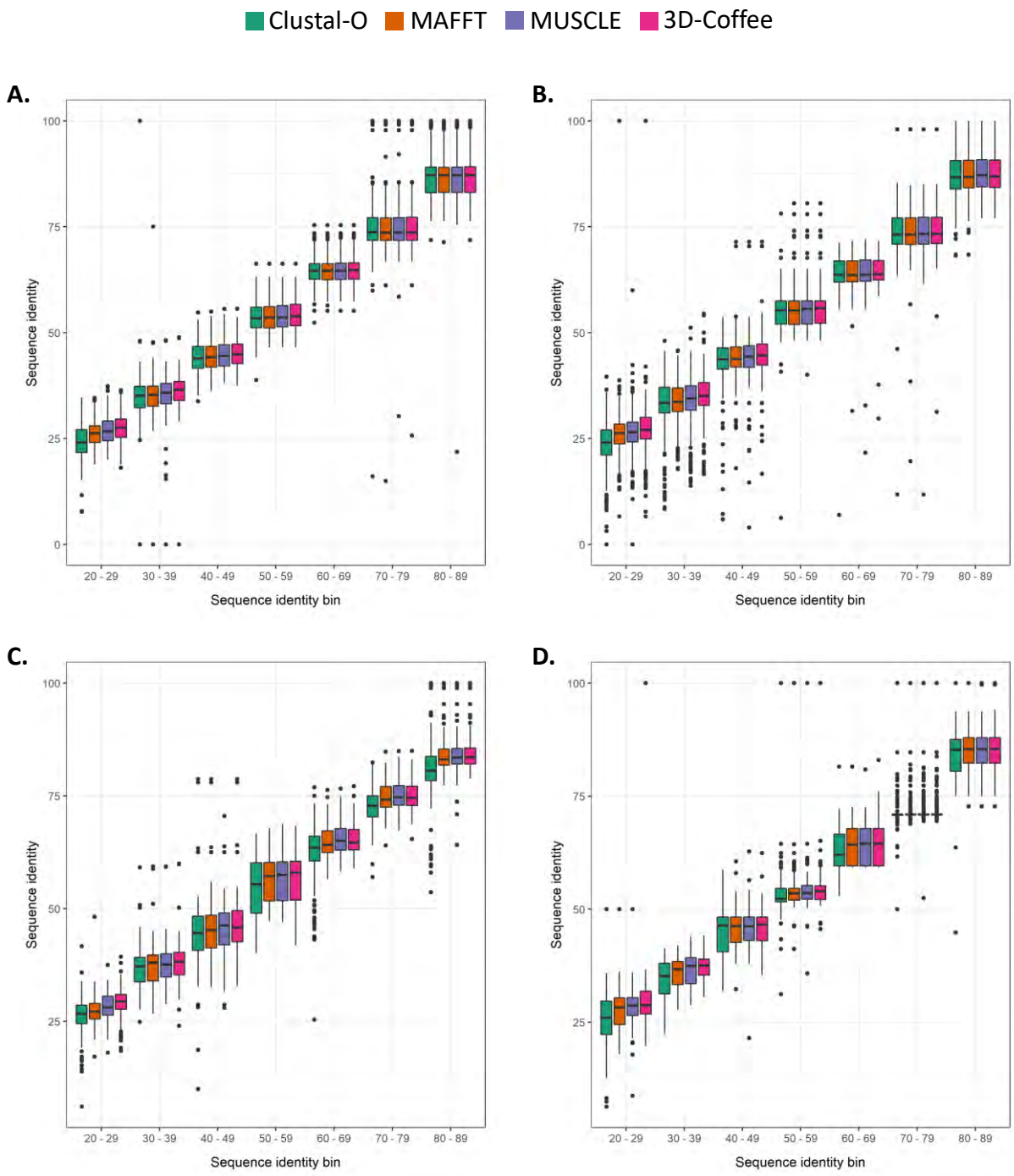


Fig 4.5. Box plots showing the average target-template sequence identities for each target-template combination per sequence identity bin. Sequence identity for each generated model was calculated for four oligomeric states: A) homodimers B) homotetramers C) heterodimers and D) heterotetramers. The average sequence identities were calculated from the PIR files used during the modeling step according to their respective sequence identity bins and alignment program. Sequence identity outliers are shown for every alignment in its respective bin before they were filtered.

4.6.2 Evaluation of models using z-DOPE score

The quality of these models was assessed using MODELLER's z-DOPE score, results from which are shown in Fig 4.6. When using the z-DOPE score to evaluate the quality of models, a score of −1.0 and below is desired as these models are considered native-like [222]. After testing, models from 50 – 69% (Fig 4.6A), 40 – 49% (Fig 4.6B), 60 – 69% (Fig 4.6C), and 40 – 49% (Fig 4.6D) bins and above for homodimers, homotetramers, heterodimers and heterotetramers dataset respectively were on average below this cut-off. The homotetramers and heterotetramers dataset had at least 40% sequence identity, indicating that these protein complexes had similar structures [75]. However, this is not always the case since some proteins with high sequence identities have different structures and functions [223].

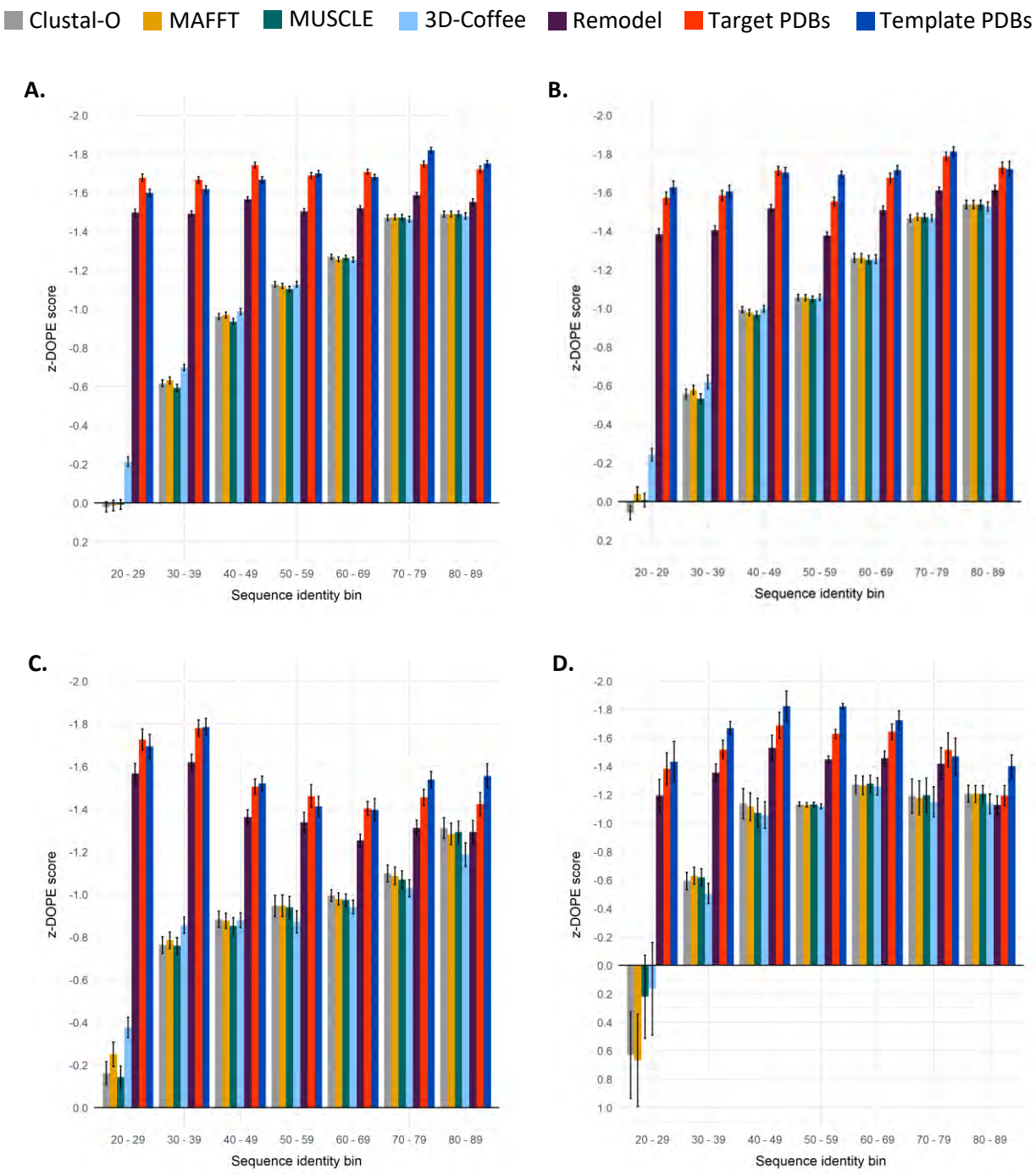


Fig 4.6. The z-DOPE score results for testing the multimeric protein modeling Python scripts for the four oligomeric states used in this work. Target-template combinations were divided into bins based on their average sequence identities. The 'Remodel' data represent the target protein complexes remodeled using their structures as template structures. The 'Target PDBs' and 'Template PDBs' data represent the quality (z-DOPE score) of the target and template protein complexes used. Data shown above are for (A) homodimers, (B) homotetramers, (C) heterodimers, and (D) heterotetramers.

Generally, models for bins below 40% sequence identity for all oligomeric states were of low-quality; however, the different alignment programs performed differently. For all the oligomeric states tested, 3D-Coffee outperformed the other alignment programs for the 20 – 29% and 30 – 39% bins except for the heterotetramer dataset where this happened for only the 20 – 29% bin. This was expected since 3D-Coffee incorporates structural information in the sequence alignment thus improving alignment quality [138]. For bins with high sequence identities (>40%), improvements were less prominent since structural information is most valuable for improving alignments of remote sequences.

PDB structures of the targets and templates were included in z-DOPE score calculations to evaluate the quality of the structures that were used during modeling. On average, models with sequence identities equal to or above 70% showed better results compared with the quality of target and template structures used for all the oligomeric states.

Additionally, each target protein was remodeled using its very structure as the template to represent ideal conditions i.e., 100% sequence identity. None of the models matched the quality of remodeled target protein complexes except the heterotetramer dataset. For the homomultimeric dataset, models in bins with sequence identities from 70% and above on average matched the quality of remodeled targets. Heterodimeric models in the bins with sequence identities from 80% and above on average matched the quality of the remodeled targets. On the other hand, heterotetrameric models in the bins with sequence identities from 80% and above showed similar quality as the remodeled targets and target structures.

Assessment of multimeric protein modeling Python scripts was done by modeling protein targets from the PDB [51] so that the models can be compared to their corresponding known structures. This was done by evaluating RMSD and other additional metric parameters.

4.6.3 Evaluation of protein models using RMSD and other additional metric parameters

For all the oligomeric states, protein models with low sequence identities had high RMSD values except for models generated after aligning using the 3D-Coffee option for homodimers (Fig 4.7A), homotetramers (Fig 4.8A), and heterodimers (Fig 4.9A). For bins in the 60 – 69% (homodimer), 50 – 59% (homotetramer and heterotetramer) range and above, had RMSD values within 2 Å of the target PDBs except for the heterodimer dataset whose RMSD values were within 2.5 Å of the target PDBs for the 60 – 69% range and above. In all the oligomeric states, a similar trend was observed for the RMSD values and those for the z-DOPE score assessment. Models with lower z-DOPE scores were found to correspond to lower RMSD values [224] as well as high sequence identity [225]. Using RMSD as a quality assessment measurement had some limitations since some target and template PDB structures were present in different conformations leading to very high RMSD values. This was solved by removing outliers from each bin in each oligomeric state dataset before evaluation of these models.

However, due to limitations of using RMSD as the only evaluation criterion, GDT-HA score [153], lDDT score [220], and TM-score [221] were considered. RMSD underestimates the accuracy of models if some loop regions are inaccurate [226] which makes it more sensitive than the GDT-HA score. The GDT-HA is a more stringent version of GDT, but all the oligomeric states scored above 50%. The heteromultimers scores were better than those for the homomultimers with bins from 30 – 39% sequence identity and upwards equal to or above 50% GDT-HA score. For the homomultimers, dimers scored 50% from the 40 – 49% bin and above compared to the tetramers which only scored the same for the 80 – 89% bin.

Since lDDT is a superposition-free metric, it is insensitive to domain movements making it a better option compared with methods that calculate global scores [220]. For all the oligomeric states, lDDT scores (Fig 4.7C, Fig 4.8C, Fig 4.9C, and Fig 4.10C) were higher than the GDT-HA scores (Fig 4.7D, Fig 4.8D, Fig 4.9D, and Fig 4.10D). GDT-HA scores were above 70% for the bins equal to or above 60 – 69% for the homodimers (Fig 4.7D) and heterodimers (Fig 4.9D).

TM-score is also a global score; however, it is independent of the protein size for related structure pairs unlike GDT scores and RMSD calculation. This makes TM-score a more sensitive measure compared with GDT scores [221]. This is also depicted in the results (Fig 4.7B, Fig 4.8B, Fig 4.9B, and Fig 4.10B) for all the oligomeric states. Overall, TM-scores were above 0.5 except for the 20 – 29% bin for homotetramers (Fig 4.8B) and heterotetramers (Fig 4.10B). This indicates that the protein complex pairs had the same structural topology [227]. Generally, RMSD values for all the oligomeric states for the bin 20 – 29% were high but with reasonable TM-scores. This is because the average RMSD calculation is affected by the size of the proteins with big proteins having high RMSD values [154].

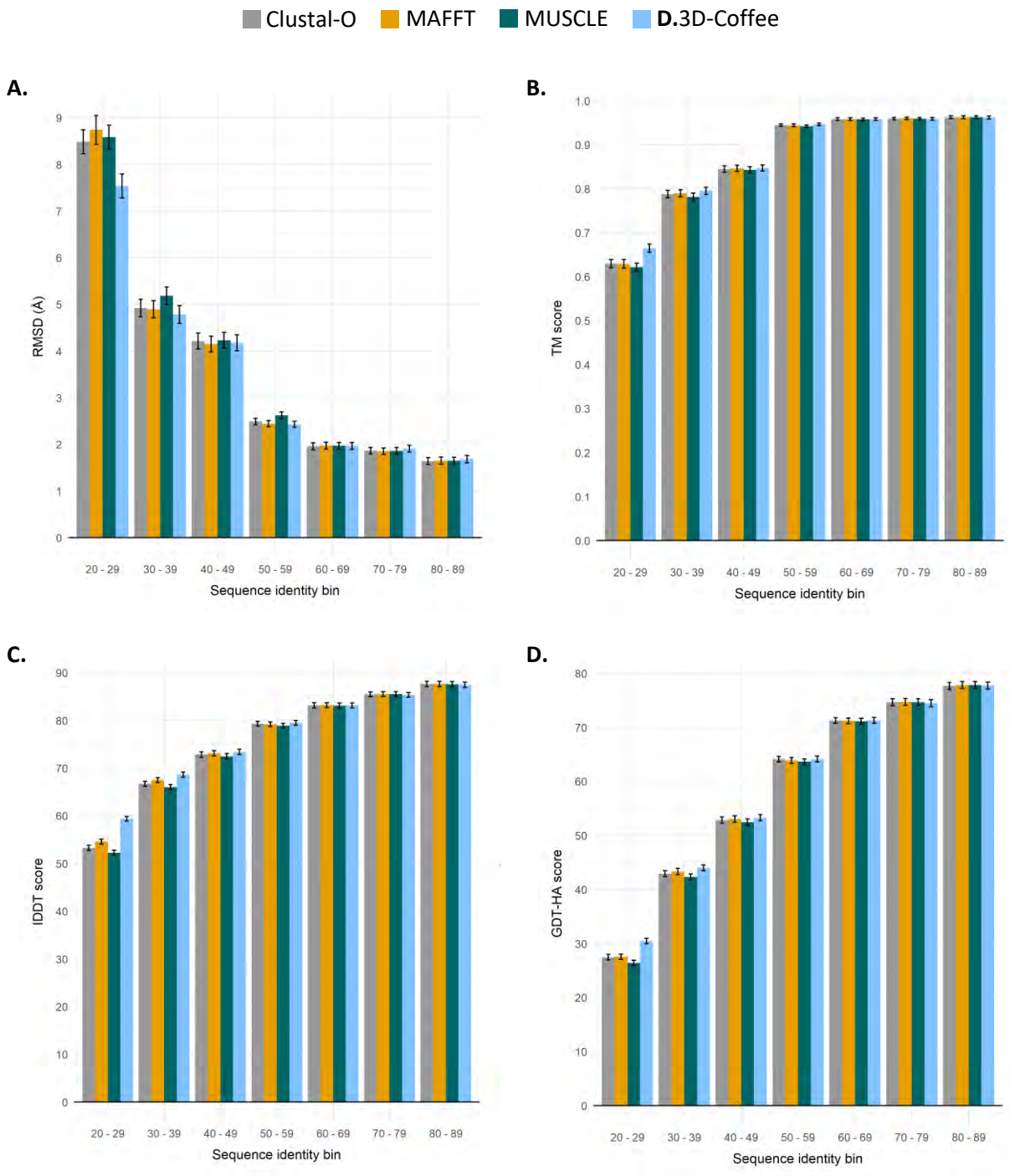


Fig 4.7. Evaluation of the multimeric protein modeling Python scripts for the homodimeric dataset. Results shown are for homodimeric protein models for each alignment program in different sequence identity bins. This demonstrates: (A) the average RMSD, (B) TM-score, (C) lDDT score, and (D) GDT-HA score.

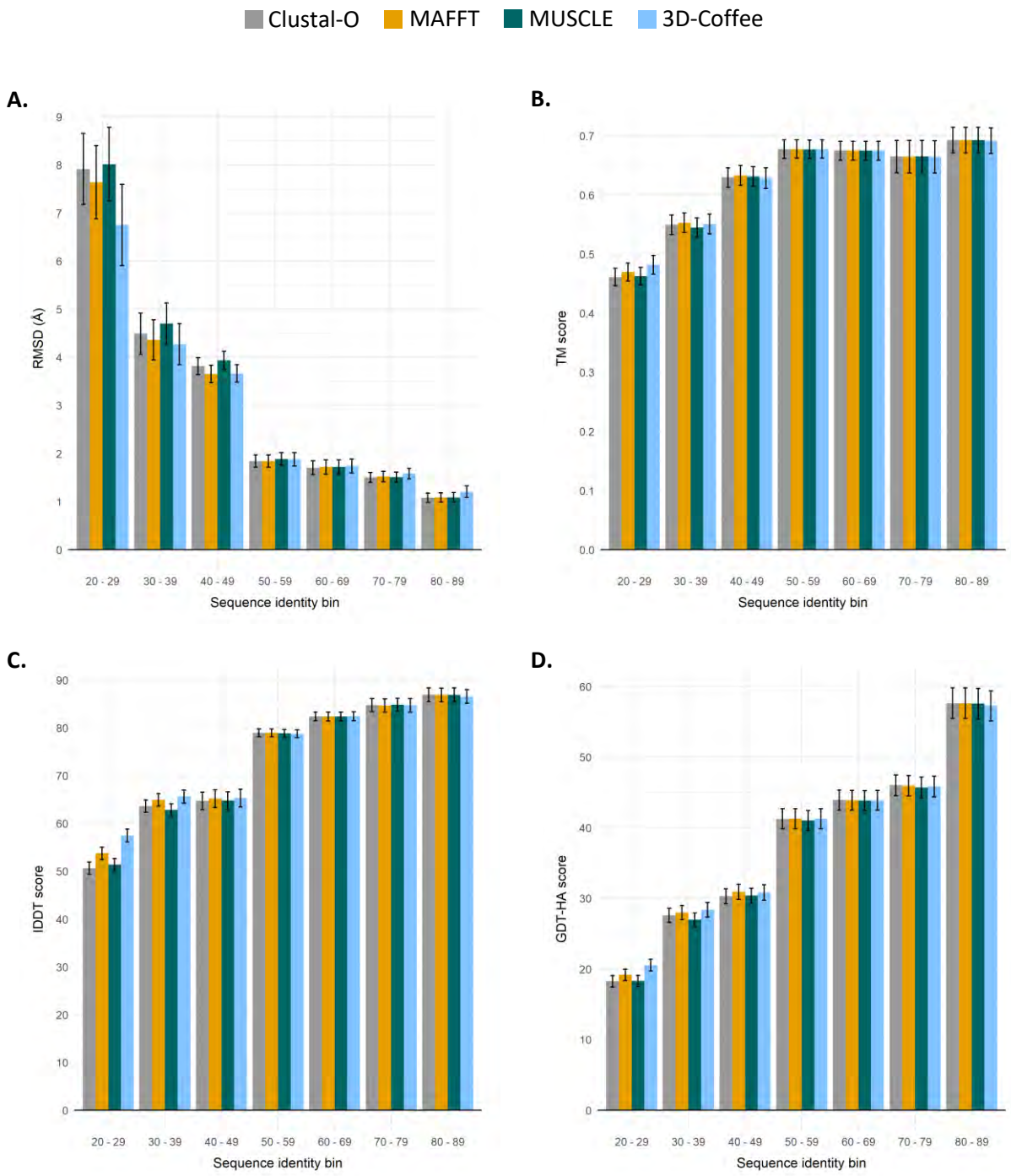


Fig 4.8. Evaluation of the multimeric protein modeling Python scripts for the homotetrameric dataset. Results shown are for homotetrameric protein models for each alignment program in different sequence identity bins. This demonstrates: (A) the average RMSD, (B) TM-score, (C) lDDT score, and (D) GDT-HA score.

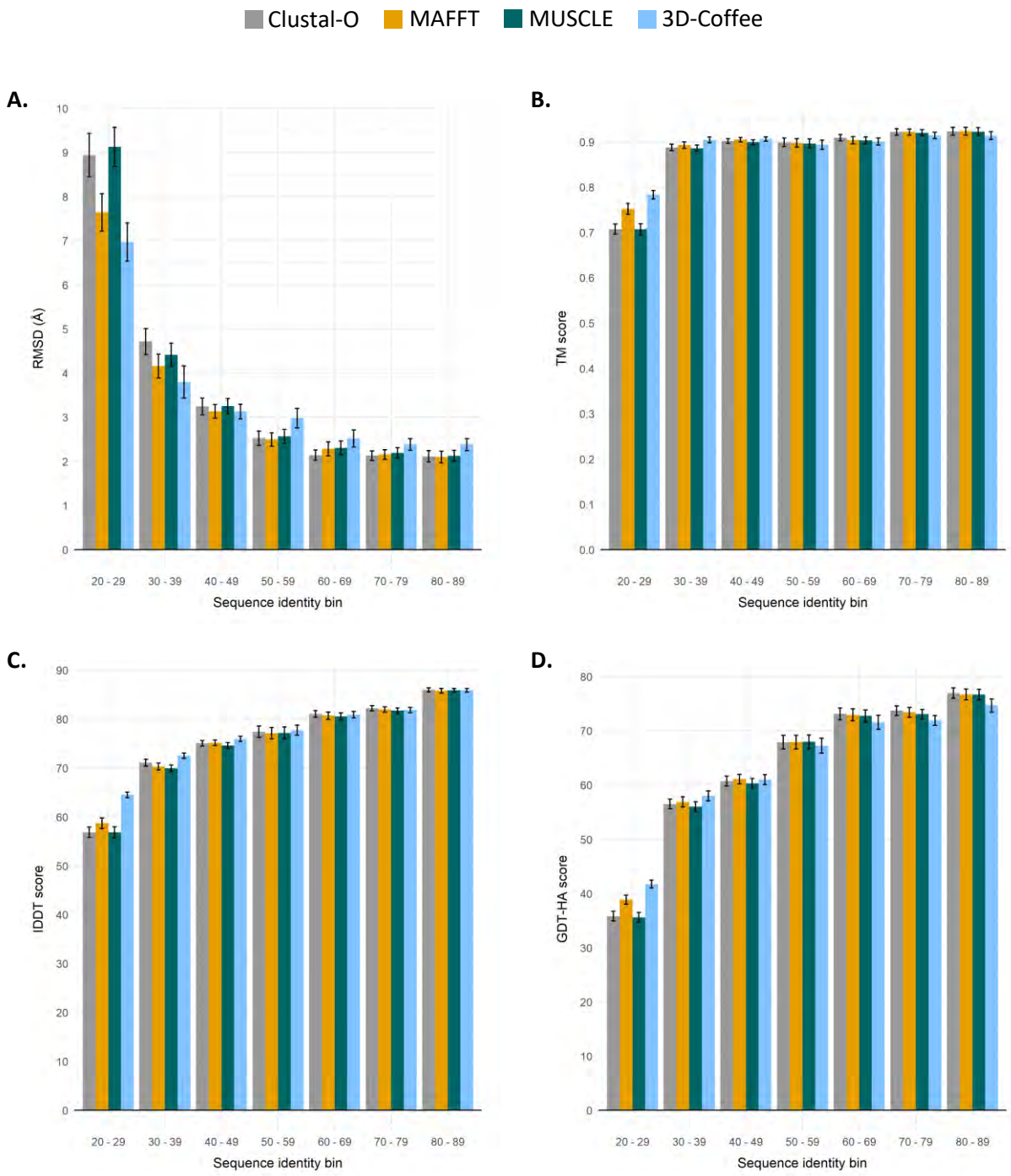


Fig 4.9. Evaluation of the multimeric protein modeling Python scripts for the heterodimeric dataset. Results shown are for heterodimeric protein models for each alignment program in different sequence identity bins. This demonstrates: (A) the average RMSD, (B) TM-score, (C) lDDT score, and (D) GDT-HA score.

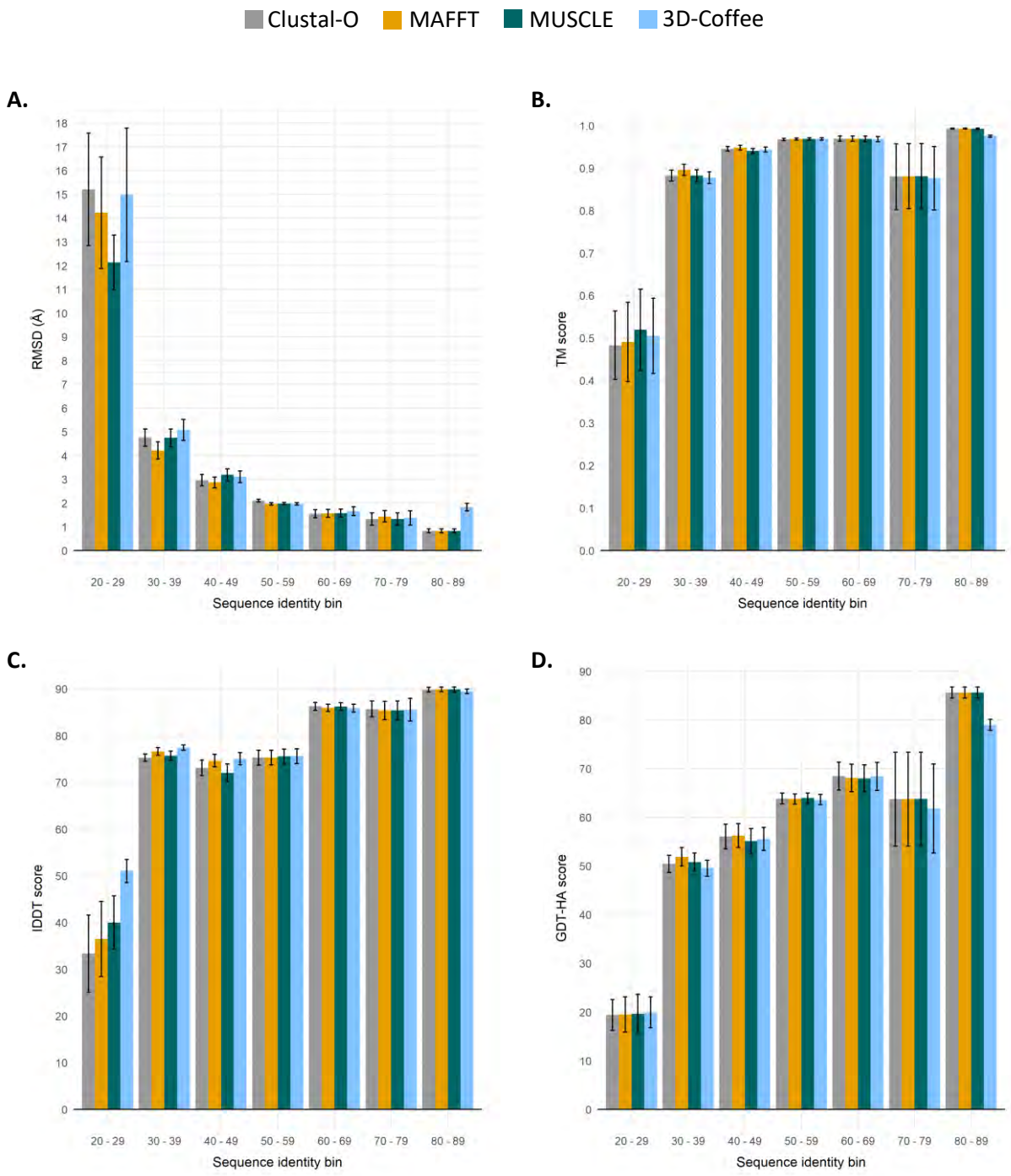


Fig 4.10. Evaluation of the multimeric protein modeling Python scripts for the heterotetrameric dataset. Results shown are for heterotetrameric protein models for each alignment program in different sequence identity bins. This demonstrates: (A) the average RMSD, (B) TM-score, (C) lDDT score, and (D) GDT-HA score.

4.6.4 Model refinement results

To assess the effect of using different refinement options for MODELLER [119], more tests were performed for homomultimeric and heteromultimeric proteins. During benchmark tests, very slow refinement parameter was used to model protein targets. Generated data was complemented with results from modeling the same oligomeric state dataset using other refinement levels including none and fast refinement (Fig 4.11 and Fig 4.12).

A significant improvement was observed for the z-DOPE scores when using fast refinement and no refinement option (Fig 4.11B, Fig 4.12A, and Fig 4.12B) except for homodimers (Fig 4.11A). A slight improvement for z-DOPE scores was also observed between using very slow refinement and fast refinement option with the very slow refinement option performing better. The very slow refinement option showed no significant difference from using no refinement for the homodimers (Fig 4.11A).

Interestingly, the RMSD metric showed consistent results with the z-DOPE scores. Using fast and very slow refinement parameters resulted in lower RMSD values compared with the no refinement option (Fig 4.11D, Fig 4.12C and Fig 4.12D) except for the homodimers (Fig 4.11C). Sequence identities below 50%, 70%, and 40% showed differences for homotetramers (Fig 4.11D), heterodimers (Fig 4.12C) and heterotetramers (Fig 4.12D), respectively when using the three model refinement options.

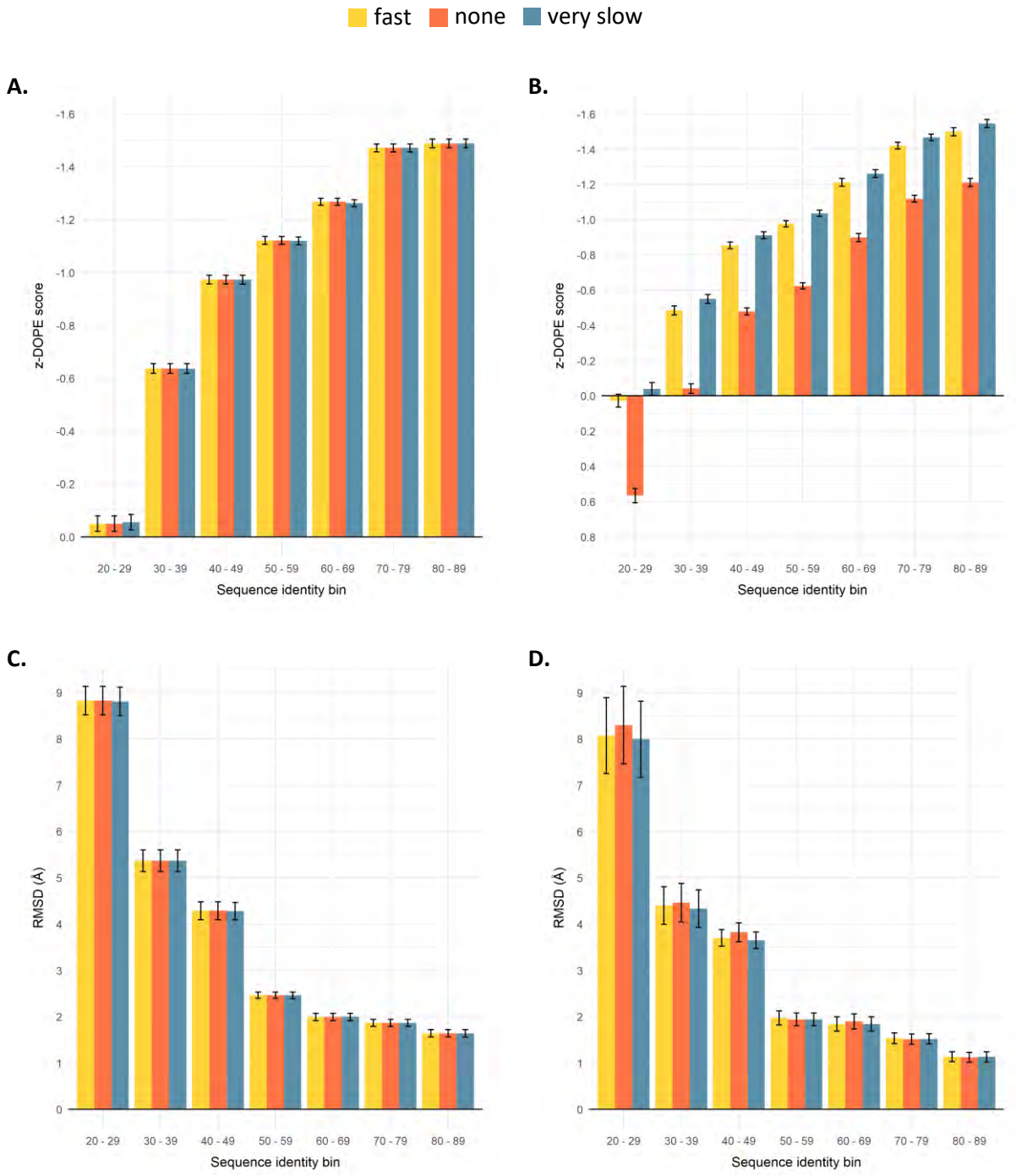


Fig 4.11. Evaluation of homomultimeric models using model refinement options. Results shown are for the z-DOPE score results and RMSD values using fast, none and very slow refinement options in MODELLER. These include the z-DOPE score (A), (B) and RMSD values (C), (D) for dimers and tetramers, respectively.

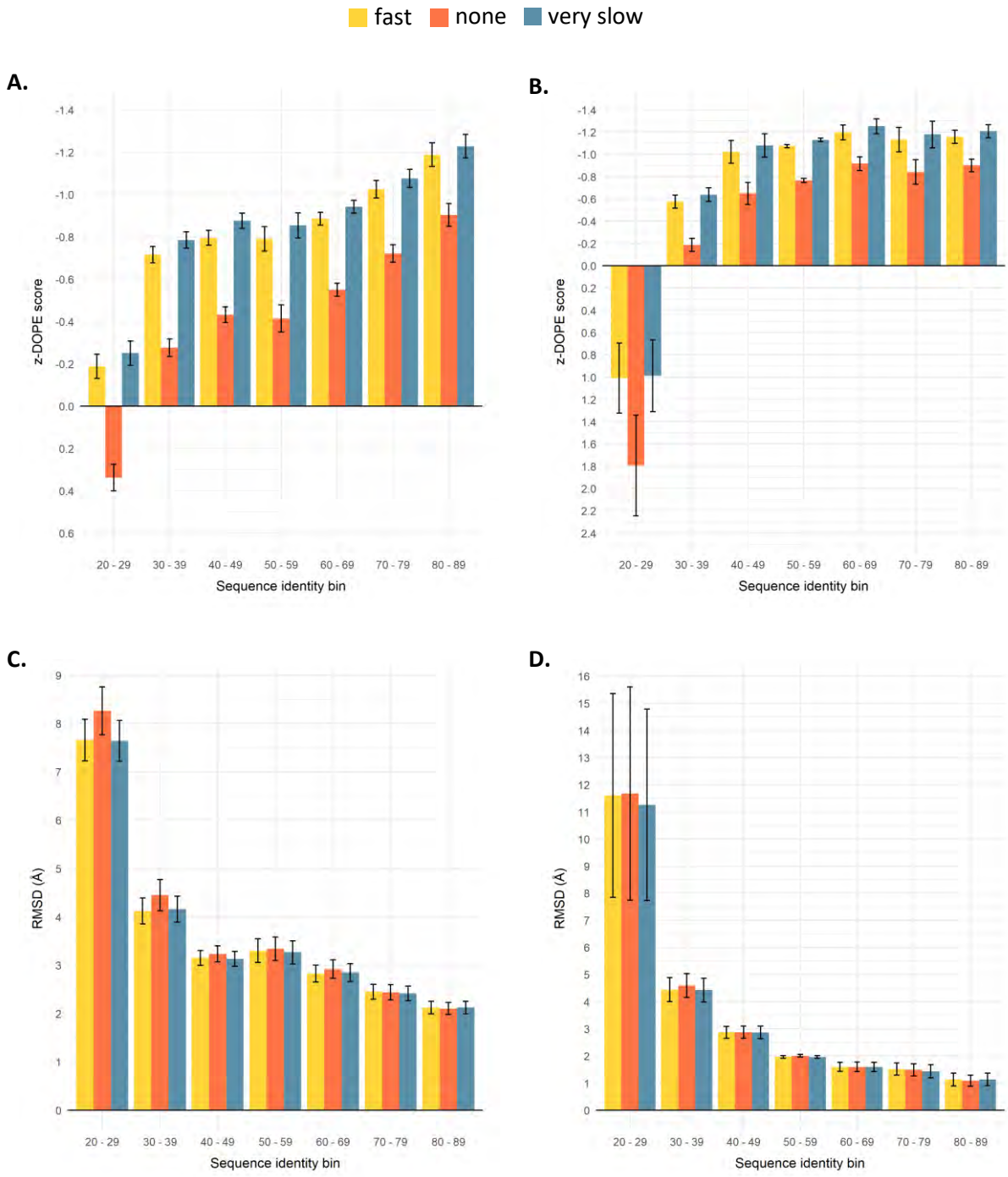


Fig 4.12. Evaluation of heteromultimeric models using model refinement options. Results shown are for the z-DOPE score results and RMSD values using fast, none and very slow refinement options in MODELLER. These include the z-DOPE score (A), (B) and RMSD values (C), (D) for dimers and tetramers, respectively.

4.6.5 Case studies

The assessment of protein models was performed using model evaluation programs including ProSA [159], and Verify3D [181], QMEAN [186], PROCHECK [188] and z-DOPE score [158].

Modeling GTP_CH-I. The GTP_CH-I protein has various biological assembly templates with good sequence identities and coverage. GTP_CH-I can be modeled from various biological assembly states with different conformations from different organisms and homodecamers topped the list of templates. The generated models are shown in Fig 4.13. This showcases some of the features of PRIMO-Complexes namely protein viewer and more information displayed by the drop-down arrow. The protein viewer allows users to select and view different biological assemblies in a similar way when using SWISS-MODEL. The active drop-down arrows on some template rows display information about other existing biological assemblies for that same template.

The evaluation results from the GTP_CH-I case study are summarized in Table 4.2. Different biological assembly templates were used to model the GTP_CH-I protein but both templates were homodecamers. This demonstrates the need to model proteins using different templates and alignment programs, which PRIMO-Complexes web server is designed to accomplish. In this case study, other online servers were used to model GTP_CH-I. These servers were automatically run and only SWISS-MODEL provided an option to select biological assembly templates and their corresponding information. Robetta server provided no option beyond the initial input screen. All servers included inhibitors in models except the Robetta server. Overall, the results show that all the protein models were of good quality. PRIMO-Complexes scored more favourably than the other servers.

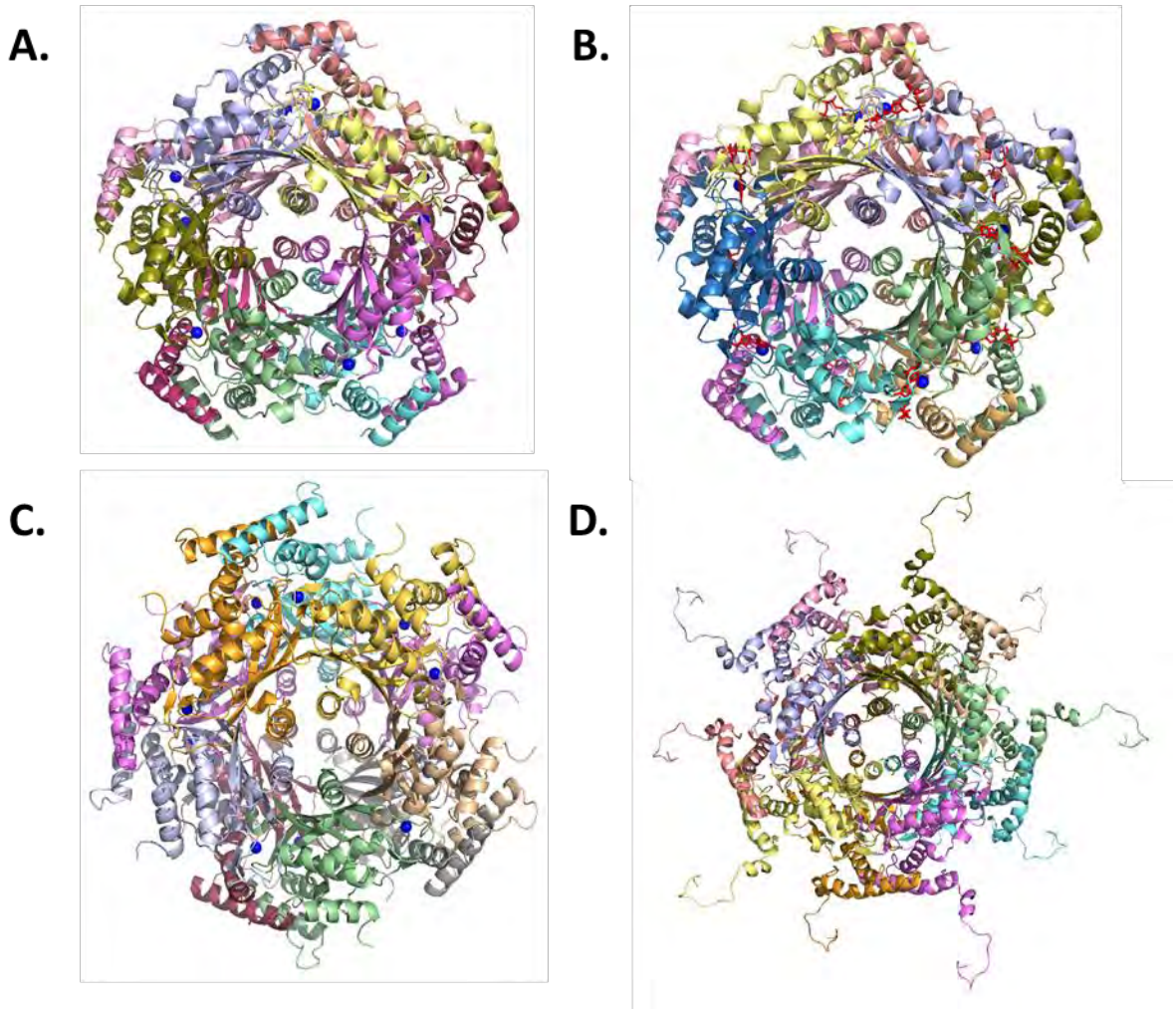


Fig 4.13. Cartoon representation of GTP cyclohydrolase I (GTP_CH-I) protein models. Protein models generated using; A) PRIMO-Complexes using T-Coffee alignment, B) PRIMO-Complexes using MUSCLE alignment. C) SWISS-MODEL, and D) Robetta are shown. PRIMO-Complexes and SWISS-MODEL included ligands and/or ions in the models except the Robetta web server.

Modeling hSod1. This protein was used to show that these servers are capable of modeling protein monomers in case they are predicted to function as single proteins. For PRIMO-Complexes, hSod1 was modeled using an asymmetric unit and biological assembly template. This illustrates the PRIMO-Complexes feature of returning and modeling from asymmetric units in case no biological assemblies exist in the PDB. SWISS-MODEL and Robetta servers were used to model hSod1. PRIMO-Complexes and SWISS_MODEL included inhibitors in the models whereas Robetta server did not as shown in Fig 4.14. All proteins were modeled as monomers and of good quality as shown in Table 4.3.

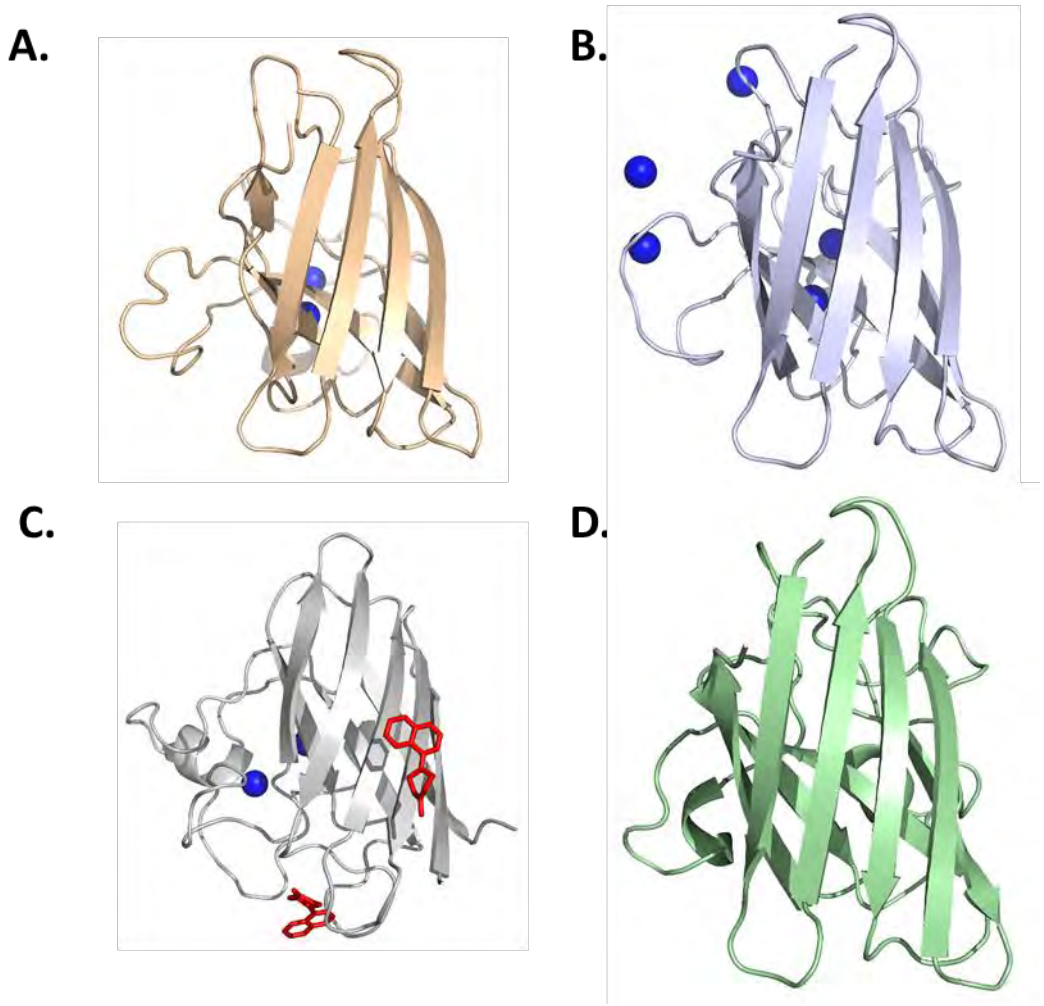


Fig 4.14. Cartoon representation of superoxide dismutase (hSod1) protein models. These protein models generated using; A) PRIMO-Complexes using T-Coffee alignment, B) PRIMO-Complexes using MUSCLE alignment. C) SWISS-MODEL, and D) Robetta are shown. All the web servers generated monomeric protein models.

Modeling alpha-globin. With the exception of Robetta, all other modeling servers returned heterotrimers. Robetta returned a heterodimer, as the predicted biological assembly. In terms of ligand modeling, PRIMO-Complexes and SWISS-MODEL identified ligands in their respective biological assembly templates and included them in the generated models whereas Robetta server did not include ligands as shown in Fig 4.15. All the servers produced good quality models regardless of the different predicted oligomeric states.

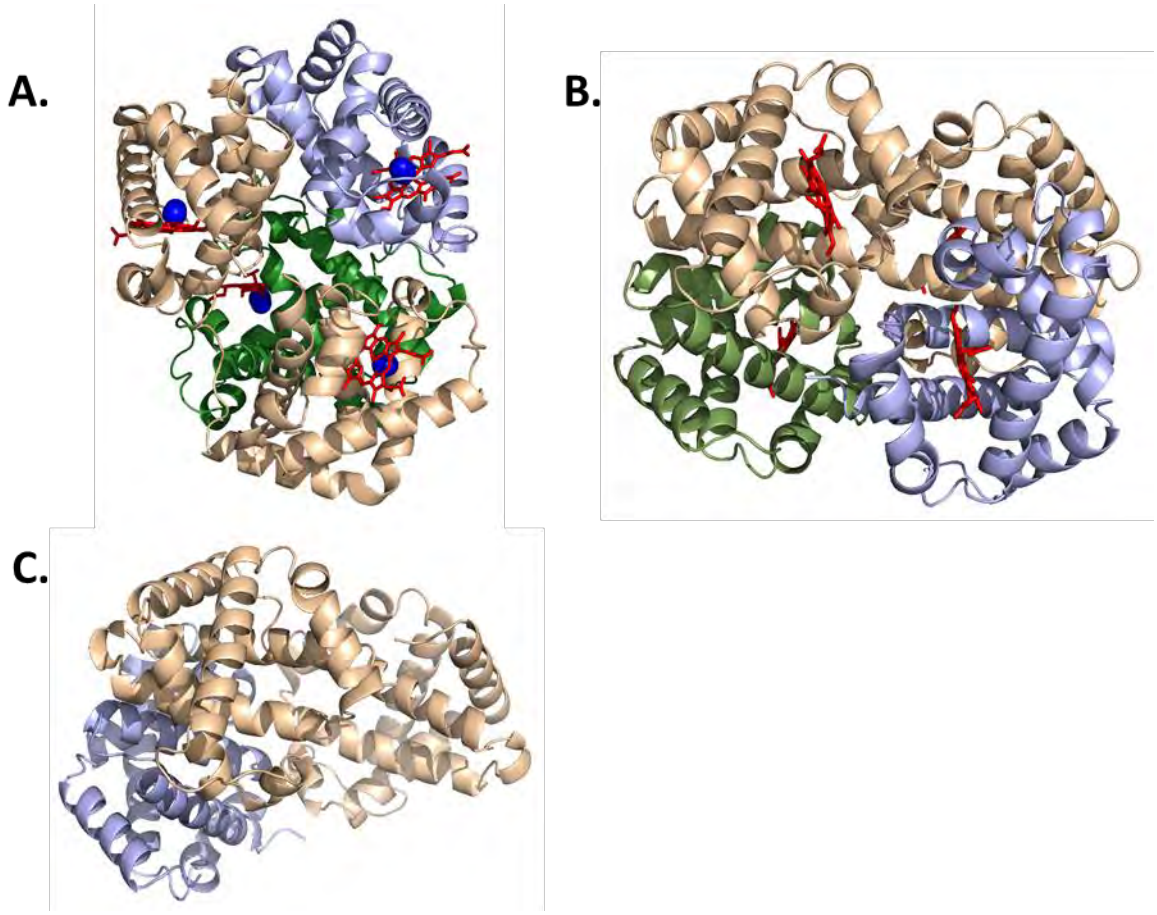


Fig 4.15. Cartoon representation of hemoglobin subunit alpha (alpha-globin) protein models. Protein models generated using; A) PRIMO-Complexes using MAFFT alignment, B) SWISS-MODEL, and C) Robetta are shown. PRIMO-Complexes and SWISS-MODEL modeled trimers and included ligands and/or ions in the models with the except of the Robetta web server.

Apart from PRIMO-Complexes), none of the servers provided options to specify ligands to be included when modeling. For PRIMO-Complexes, specific inhibitor molecules in each chain were selected from each template to be modeled with the protein. These three case studies were not meant to be a comprehensive assessment of PRIMO-Complexes compared to other protein multimeric modeling servers, but it was encouraging to see that PRIMO-Complexes performed relatively well against other servers for most of the evaluation tools used (Table 4.2, Table 4.3, and Table 4.4).

Table 4.2. Protein model quality evaluation results for modeling GTP cyclohydrolase I (GTP-CH-I) protein using PRIMO-Complexes and other modeling servers. GTP-CH-I protein was modeled, and the models were evaluated. Protein models for each server are shown along with the quality scores generated by ProSA, Verify3D, QMEAN, PROCHECK and z-DOPE score. The predicted oligomeric state modeled for each protein is also shown. The PROCHECK results are divided as follows: Fav – Residues in most favoured regions; Add – Residues in additional allowed regions; Gen – Residues in generously allowed regions; Dis – Residues in disallowed regions.

GTP-CH-I									
	ProSA	**Verify3D**	**QMEAN**	**PROCHECK**				**MODELLER**	
Model	**Z-score**	**% Residues with 3D-1D score >= 0.2**	**QMEAN4**	**Fav.**	**Add.**	**Gen.**	**Dis.**	**z-DOPE score**	**Oligomeric state modeled**
PRIMO-Complexes_3DC	-7.21	86.85%	0.57	97.3%	2.7%	0.0%	0.0%	-1.84	homodecamer
PRIMO-Complexes_muscle	-7.21	86.04%	0.24	96.5%	3.3%	0.2%	0.0%	-1.51	homodecamer
SWISS-MODEL	-6.86	69.28%	-1.49	90.0%	8.2%	1.3%	0.5%	-1.75	homodecamer
Robetta	-6.42	80.68%	0.86	94.8%	5.2%	0.0%	0.0%	-1.75	homodecamer

Table 4.3. Protein model quality evaluation results for modeling superoxide dismutase (hSod1) protein using PRIMO-Complexes and other modeling servers. hSod1 protein was modeled, and the models were evaluated. Protein models for each server are shown along with the quality scores generated by ProSA, Verify3D, QMEAN, PROCHECK and z-DOPE score. The predicted oligomeric state modeled for each protein is also shown. The PROCHECK results are divided as follows: Fav – Residues in most favoured regions; Add – Residues in additional allowed regions; Gen – Residues in generously allowed regions; Dis – Residues in disallowed regions.

hSod1									
	ProSA	**Verify3D**	**QMEAN**	**PROCHECK**				**MODELLER**	
Model	**Z-score**	**% Residues with 3D-1D score >= 0.2**	**QMEAN4**	**Fav.**	**Add.**	**Gen.**	**Dis.**	**z-DOPE score**	**Oligomeric state modeled**
PRIMO-Complexes_3DC_asymmetric unit template used	-4.89	97.39%	-1.47	82.0%	16.4%	1.6%	0.0%	-1.41	monomer
PRIMO-Complexes_muscle	-5.54	100.00%	0.78	91.8%	8.2%	0.0%	0.0%	-1.95	monomer
SWISS-MODEL	-5.81	100.00%	1.22	90.2%	9.8%	0.0%	0.0%	-2.11	monomer
Robetta	-5.39	99.35%	1.98	90.2%	9.8%	0.0%	0.0%	-2.18	monomer

Table 4.4. Protein model quality evaluation results for modeling hemoglobin subunit alpha (alpha-globin) protein using PRIMO-Complexes and other modeling servers. Alpha-globin protein was modeled, and the models were evaluated. Protein models for each server are shown along with the quality scores generated by ProSA, Verify3D, QMEAN, PROCHECK and z-DOPE score. The predicted oligomeric state modeled for each protein is also shown. The PROCHECK results are divided as follows: Fav – Residues in most favoured regions; Add – Residues in additional allowed regions; Gen – Residues in generously allowed regions; Dis – Residues in disallowed regions.

alpha-globin									
	ProSA	**Verify3D**	**QMEAN**	**PROCHECK**				**MODELLER**	
Model	**Z-score**	**% Residues with 3D-1D score >= 0.2**	**QMEAN4**	**Fav.**	**Add.**	**Gen.**	**Dis.**	**z-DOPE score**	**Oligomeric state modeled**
PRIMO-Complexes_MAFFT	-8.03	99.65%	-1.69	94.2%	5.6%	0.0%	0.2%	-0.94	heterotrimer
SWISS-MODEL	-7.93	97.74%	-0.72	93.4%	6.4%	0.0%	0.2%	-1.61	heterotrimer
Robetta	-7.88	100.00%	0.11	92.0%	7.7%	0.3%	0.0%	-1.65	heterodimer

4.7 Conclusion

We have successfully developed and evaluated the functionality of PRIMO-Complexes–a new webserver for modeling protein complexes. The results from the benchmark tests are generally promising with low z-DOPE scores for sequence identities above 40% for all the oligomeric states. The other additional metric parameters calculated also showed that multimeric protein modeling algorithms embedded in PRIMO-Complexes can generate high-quality protein complexes. The automated features of PRIMO-Complexes to model protein complexes will be continuously evaluated by CAMEO in future.

PART II: SIDE PROJECTS – DEVELOPMENT OF OTHER BIOINFORMATICS TOOLS AND WEB SERVERS

CHAPTER FIVE

RUBI PROTEIN MODEL REPOSITORY FOR ANNOTATED 3D PROTEIN STRUCTURES

Chapter overview

In this work, a web-based repository was developed to provide researchers with access to protein models generated in our research group – RUBi. In this chapter, the RUBi protein model repository which houses protein models is discussed in detail. This web server can be freely accessed via a web server (https://rhpr.rubi.ru.ac.za/).

5.1 Introduction

Computational protein structure prediction methods [228,229] are currently being used to bridge the gap between and known sequences and their structures. Web servers have been developed to aid the process of protein structure prediction generating enormous protein models. The increased production of these theoretical and pre-computed models has necessitated the development of databases to store them. These repositories serve as starting points to assist researchers in effortlessly exploring the 3D protein space while working on various projects. Repositories including protein model portal [198], Swiss-Model repository [230], Genome3D [231], ModBase [232], and G protein-coupled receptor database (GPCRdb) [233] contain protein models generated using various methods. The annotated 3D protein models in these databases are automictically generated. Some are based on sequences in UniProtKB [234], and /or Structure Function Linkage Database (SFLD) [235] while others are user-specific. In contrast to the other databases which store computational models, GPCRdb [233] is used to store and analyse experimentally derived data from PDB crystal structures [51] and manually annotated single-point mutations.

5.2 Research aim and objectives

The aim of this project is to provide users with access to theoretical verified protein models to ensure the reproducibility of the work.

Specific objectives for this work:

1. To develop a web server to provide access to protein models with their corresponding accuracy information
2. To incorporate an NGL viewer for users to visualise the protein models

5.3 Implementation details

5.3.1 Web interface

The server side of RUBi protein model repository was developed using a Django framework (https://www.djangoproject.com). Graphical aspects of the web application were implemented using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) with a bootstrap framework, JavaScript using AJAX calls to the server and NGL viewer for 3D structure visualization (https://github.com/arose/ngl/).

5.3.2 Repository content

The RUBi protein model repository includes models that have been created using MODELLER [119] and in-house Python scripts. The process of homology modeling involves all the steps as implemented by automated modeling pipelines. Software or tools used at each modeling step include: BLAST [122] and HHpred [133,236] to search for suitable templates (protein structures) in the PDB [51]; Clustal [134], Muscle [135], MAFFT [136], T-Coffee [137], and Promals3D [138] for multiple sequence alignment, MODELLER [119] for modeling and loop refinement; ANOLEA [237], PROCHECK [188], ProSA [159], QMEAN [186], and Verify3D [181] for model quality assessment. This is a standalone repository that will expand as new

protein models are provided by the authors before or after they have published their findings. When better templates become available, these models will not be updated since this repository is not connected to a fully automated homology modeling pipeline. Currently, the repository contains mainly 111 model entries with 10,000 modelled single nucleotide variants.

Homology models were created using well-proven modeling methods (refer to the case study section) and most have been published whereas others are yet to be published. All models are research-driven, tailored to specific projects within the RUBi research group. These protein models include GTP CycloHydrolase 1 (GCH1) generated by Afrah Khairalla; Heat shock proteins generated by Arnold Amusengeri; *Plasmodium* 1-deoxy-D-xylulose 5-phosphate reductoisomerase (DXR) generated by Bakary N'tji Diallo; Aminoacyl tRNA synthetases (aaRSs) generated by Dorothy Nyamai; *Plasmodial* proteases generated by Musyoka Mutemi Thomas; Human Immunodeficiency Virus (HIV) protease generated by Olivier Sheik Amamuddy; *Plasmodial* Transketolases generated by Rita Afriyie Boateng; Auxiliary Activity family 9 (AA9) generated by Vuyani Moses; and Glycoside Hydrolase 1 enzymes from *Bacillus licheniformis* generated by Wayde Veldman.

5.4 Results

5.4.1 Web server design and content

We designed the webserver to store protein models. The webserver can be accessed through https://rhpr.rubi.ru.ac.za/

It has a single-page interactive user interface. The main page of the repository is accessed via a summary table that is displayed to the user (Fig 5.2). This table contains all the available models with their corresponding z-DOPE scores [158], validation scores with a few selected validation programs, and a link to the article in case it is published. This also allows the user to search, download, and view the 3D model structures. An in-page protein structure

visualization was incorporated using the NGL web viewer plugin [219]. The models can be downloaded as flat files in PDB format.

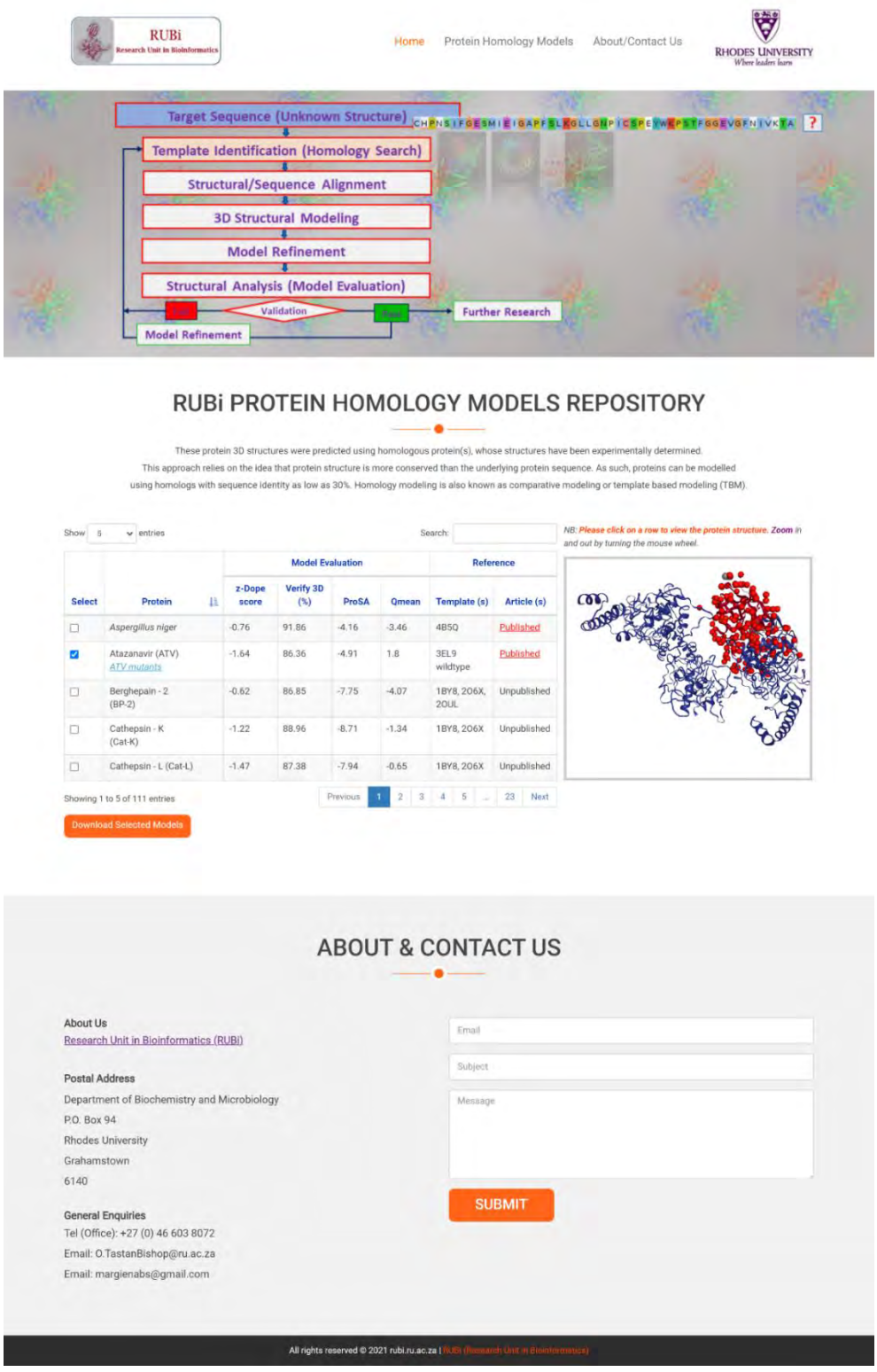


Fig 5.1. A single page web interface for RUBi protein homology models repository. The pages change as you scroll back and forth.

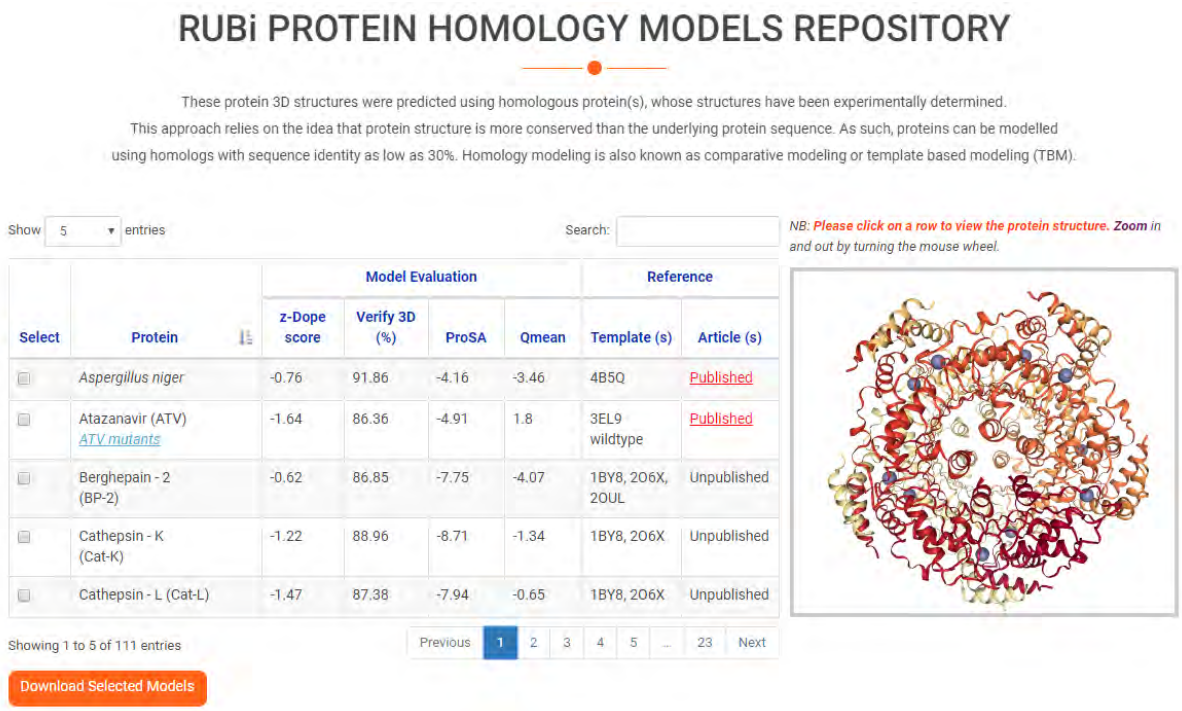


Fig 5.2. Protein models web page. It shows the protein name, z-Dope score, quality assessment values, templates and reference to the article if published. On the right is the 3D structure visualization plugin showing one of the protein models in the repository and selected models are downloaded.

5.5 Examples of models in the repository

5.5.1 *Plasmodium* 1-deoxy-D-xylulose 5-phosphate reductoisomerase (DXR)

The models are of *Plasmodium* 1-deoxy-D-xylulose 5-phosphate reductoisomerase (DXR) in closed and open conformations for the different *Plasmodium* sequences (*P. falciparum, P. malariae, P. vivax, P. ovale, P. knowlesi, P. berghei, P. yoelii yoelii* and *P. chaubadi*). DXR is a class B dehydrogenase that catalyses the second step of the 2-C-methyl-D-erythritol-4-phosphate (MEP) pathway by converting DOXP to MEP by isomerization and followed by NADPH reduction [238].

PfDXR is a homodimer in a V shape with a molecular mass of approximately 47 kDa. Each monomer contains an NADPH molecule and a divalent metal ion (Mg 2+, Co 2+ or Mn 2+) required for the catalytic activity of the enzyme [239]. Each monomer has two large domains, a linker region, and a small C-terminal domain. The two large domains are separated by a cleft

containing a deep pocket. A flexible loop region (residues 291 to 299) is inserted in the catalytic domain. Comparative studies between different DXR structures revealed three conformations: the open form with the loop opened (no substrate/inhibitor), the open form with the flexible loop closed (with substrate/inhibitor, prepared by soaking), and the configuration with the flexible loop covering the active site (with substrate/inhibitor, prepared by co-crystallization) [240].

MODELLER [119] version 9.19 was used for homology modeling. DXR monomer models were developed for each *Plasmodium* specie. HHPred was used for template identification: 5JAZ and 1K5H for the protein in open and closed conformations respectively. MODELLER's ALIGN2D command was used for target-template alignment. A hundred models were generated for each protein using very slow refinement. The models were produced while maintaining the template ligands at their positions to maintain the binding site geometry and environment reasonable similar to the template.

MODELLER (DOPE Z-score), QMEAN [186], PROCHEK [188], ProQ3D [241] and DFIRE [242] were used for model evaluation. The models were first filtered by the DOPE Z-score. The best five models for each protein per DOPE Z-score were selected and assessed using the available QMEAN API. The best QMEAN Z-score were finally selected for each specie. Models were assessed by using their template as a reference for comparison.

5.5.2 Glycoside Hydrolase 1 enzymes from *Bacillus licheniformis*

Glycoside hydrolases (GH) are enzymes that break down polysaccharides that are available in starch and lignocellulose, which makes up most of the content of biomass [243]. The enzymes catalyze the hydrolysis of glycosidic bonds in these polysaccharides [244]. GH1's have a wide array of functions - the Carbohydrate-Active Enzyme (CAZy) database reveals that there are twenty-one different enzymatic activities in the GH1 family, including 6-P-β-glucosidases

(EC 3.2.1.86), β-glucosidases (EC 3.2.1.21), and 6-P-β-galactosidases (EC 3.2.1.85) [245]. GH1 enzymes cleave their substrates under retention of configuration at the anomeric carbon atom, employing a double-displacement mechanism of catalysis [246]. Two conserved catalytic glutamate residues play a role in this mechanism. GH1 enzymes consist of successive (β/α) motifs that form a conserved $(\beta/\alpha)_8$-barrel core connected by short loops [247].

Using the National Center for Biotechnology Information (NCBI) database, the full Bacillus licheniformis Glycoside Hydrolase 1 (BlGH1) sequences with accession numbers AAU41434.1, AAU39684.1 and AAU42981.1 were retrieved. The HHpred web server [236] was used to search for homologous structures. To determine homology and the possibility of use as a template, unpublished solved BlGH1 crystal structures from our collaborator were aligned to the BlGH1 sequences to be modelled - also using HHpred [236]. The PDB ID of the templates selected for each of the BlGH1 sequences are as follows: AAU41434.1 – 2O9P and 5NAV; AAU39684.1 – 3QOM, 2XHY as well as unpublished structures from our collaborator (sequences AAU43012.1 and AAU43027.1); AAU42981.1 – 3QOM, 5NAV, 2O9P and 3W53. Each target sequence was aligned with the templates utilizing PROMALS3D [138]. The GH1 enzymes were modelled using MODELLER [119]. One hundred models were generated, after which the top 3 models were selected based on their z-DOPE score [158] and further evaluated using ProSA [159], QMEAN [186] and Verify3D [181] for accurate model validation. The best model was then chosen based on the combination of these results.

5.5.3 HIV protease

HIV protease Proteolytically cleaves HIV Pol and GagPol polyproteins to aid the process of viral maturation [248]. Comprises two 99 residue-long monomers, which form a substrate binding pocket at their interface [248]. Catalytic activity is mediated by an ASP25 coming from each monomer [248]. Flaps control the opening and closing of the binding cavity [249].

MODELLER was used with very slow refinement for modeling the variants using a high resolution (<1.55 Å) crystal structure as a template (PDB accession: 3EL9). Interfacial water was retained from each template crystal structure by selecting water residues at an intersecting distance of 3 Å between both chains of the dimer. Model quality was assessed using z-DOPE scores [158].

5.5.4 GTP CycloHydrolase 1 (GCH1)

GTP CycloHydrolase 1 (GCH1) is the first and rate limiting enzyme of the *de novo* folate synthesis pathway in bacteria, protozoa, fungi, plants and lower eukaryote [250]. The enzyme is responsible for the conversion of Guanosine-5'-triphosphate to the pteridine moiety dihydroneopterin triphosphate (DHNP) [251]. GCH1 is homo-decameric enzyme with ten metal containing active sites. The active sites are located in a deep pocket formed at the interface of every adjacent three monomers. GCH1 is a member of the Tunnelling-fold structural superfamily which is characterized by a central barrel formed by sequential anti-parallel β-strands flanked by α- helices on each side [252].

The malaria parasite *Plasmodium. falciparum* GCH1 3D structure was built using MODELLER v.9.16 [119]. The *Thermus thermophilus* HB8 crystal structure (PDB ID: 1WUR) was selected as a template structure for the model building. 100 models were generated with slow refinement applied to the modeling process. The resultant models were then ranked based on the calculated z-DOPE score. The top three models with the lowest z-DOPE scores were selected for structure validation via PROCHECK [188], ProSA [159], and QMEAN [186].

5.5.5 Aminoacyl tRNA synthetases (aaRSs)

Aminoacyl tRNA synthetases (aaRSs) are ubiquitous enzymes that catalyse ligation of amino acids to their cognate tRNA during protein translation [253,254]. These enzymes are grouped into two classes based on the mode of tRNA binding and the architecture of the catalytic

domain (CD) [254–256]. There is a minimum of 20 aaRSs enzymes each for charging the 20 amino acids [255,257]. Generally, aaRS have four domains, the N-terminal domain (NTD), the CD, the anticodon binding domain (ABD) and the C-terminal domain (CTD) [256]. In *Plasmodium falciparum,* some aaRSs enzymes are expressed in two copies, one residing in the cytoplasm and the other copy is targeted in the apicoplast. In this work, we only modelled the cytosolic enzymes.

The 3D structures of *P. falciparum* arginyl tRNA synthetase (ArgRS), lysyl tRNA synthetase (LysRS), methionyl tRNA synthetase (MetRS), tryptophanyl tRNA synthetase (TrpRS) and prolyl tRNA synthetase (ProRS) were calculated using MODELLER v9.15 [119]. Templates were identified using PRIMO [87] and HHpred [236] web servers. For PfArgRS-5JLD [255,258]; PfLysRS-4DPG [255,259]; PfTrpRS-1R6U [260] and 4J76 [261]; for PfProRS-4WI1 [262] and 4HVC [263]; for MetRS-4DLP [255,264], the crystal structure of *Brucella melitensis* MetRS was used. A hundred models were calculated for each protein and the top three models with the lowest z-DOPE score were selected for model quality assessment. Verify3D [181], PROSA [159] and QMEAN [186] model validation tools were used to assess the model quality and the model with the best scores was selected.

5.5.6 *Plasmodial* Transketolases

Plasmodial transketolase catalysis the reverse transfer of 2-carbon ketol group from ketose phosphate donor to an aldose phosphate acceptor. This activity produces NADPH and ribose-5-phosphate important for parasite survival. The structure of the *plasmodial* transketolase consists of three domains. The N-terminal or the pyrophosphate (PP-) terminal domain consists of residues ~ 3-322 representing almost half of the subunits and is important for thiamine diphosphate (ThDP) binding. The middle and N-terminal have been reported to be involved in the subunit's interactions and the binding and recognition of ThDP cofactor. The Middle-

terminal domain residues 323 to 538 make up the pyrimidine (Pyr) domain. It consists of parallel beta-sheets which interact with the ThDP. This region is noted for the binding of substrates and ThDP. The C- terminal consists of ~ 150 residues and five stranded mixed beta-sheet. The function of this terminal is not known; however, it is believed to regulate enzyme activities, stereochemical control of the sugar substrate and a regulatory binding site of transketolase.

P. falciparum, P. vivax, P. ovale, P. malariae and *P. knowlesi* sequences were used as query sequences to search homologs structures (templates) using HHpred [265] and PRIMO [87] web servers to retrieve suitable template (s) using the default parameters. The best template with high resolution, good PDB validation matrices, high sequence identity to the query sequences, good coverage of the structure to the query sequences, completeness of the protein structure and E-values of 0 or close to 0 and organism with known information were selected for further validation. Saccharomyces cerevisiae (PDB id: ITRK) [266] was used as the best template for the modeling. Target-template multiple sequence alignment was performed using PROMALS3D [138] tool which takes into consideration the structural information of the proteins during alignment. Aligned output was selected and aligned sequences were manually inspected were applicable. The three input files: PDB atom files of the template protein structure, alignment file of the target and template including ThDP and calcium cofactors and the MODELLER script file that instructs MODELELR 100 refined homodimer 3D structures of each transketolase were generated using MODELLER version 9.18 based on the input sequence alignment and selected template. The cofactors ThDP and Ca^{2+} were included in modeling. Very slow refinement level was used which was performed by MODELLER by default. After the models were generated, a MODELLER Python script was used to generate each of the model's DOPE scores to rank the models on the basics of their energy levels. Top three models with the lowest DOPE evaluated scores were selected form the rest of the models

for further evaluation analysis. The top three models each were evaluated using PROCHECK [188], VERIFY3D [181], QMEAN [186]. One best model with a consensus between the quality validation tools.

5.5.7 Heat shock proteins

Heat shock proteins, in general, are highly conserved molecular chaperones that facilitate correct protein folding, regulate translocation and exposure of misfolded proteins to degradation machineries. PfHsp70-1 and PfHsp70-x, expressed during the critical asexual blood developmental stage of *P. falciparum*, promote the parasites' adaptation and survival in the human body. Normal human cells ubiquitously express Hsc70, and in limited amounts Hsp72. Hsc70 is thought to maintain routine intracellular proteostatic functions. Expression of Hsp72 is often induced by stress stimuli, and it is thought to discourage protein aggregation.

Structurally, Hsp70's are made up of two main domains: a nucleotide binding domain (NBD), and a substrate binding domain (SBD), connected by a conserved interdomain linker. The NBD and SBD are involved in the binding and release of nucleotides (ATP/ADP) and peptide substrate respectively. The protein adopts two major conformations during its functional cycle: An open conformation (ATP bound) and a closed conformation (ADP and substrate bound). Ligand binding/release events induce cross-domain allosteric signals which regulate conformational transitions.

Comparative modeling was implemented in generating all the models. The method used to model Hsp72, Hsc70 and PfHsp70-1, PfHsp70-x structures can be found in our previously published work [267].

5.5.8 Auxiliary Activity family 9 (AA9)

The Auxiliary Activity family 9 (AA9) enzymes are a group of that interact with crystalline cellulose resulting in cleavage of the glucose residues. This reaction involves abstraction of a carbon atom which is subsequently replaced by a hydrogen atom through successive electron rearrangements. The cleavage of the cellulose chain results in the destabilization of the cellulose crystal which makes it susceptible to the action of cellulose [268].

AA9 enzymes are a single monomeric enzyme that have an active site region that can be found close to the central part of their characteristic flat face. This flat face is believed to directly interact with cellulose resulting in cleavage. A copper ion located at the active site is responsible for catalysing the observed cleavage. This copper is chelated by two of the N-terminal histidine and a nitrogen from another histidine side chain resulting in a configuration called the histidine brace [269]. Moses et al details how the modeling of the AA9 protein was performed [270].

5.5.9 *Plasmodial* proteases

During the erythrocytic blood phase, *Plasmodium falciparum* utilises papain-like Clan CA cysteine proteases to degrade host hemoglobin in order to obtain nutrients required for their growth and replication [271–273]. Falcipain 2 and 3 (FP-2 and FP-3) are known to be the key hemoglobinases and are validated drug targets [274,275]. Other Plasmodium species also express highly homologous to FP-2 and FP-3. These include vivapains (vivapain 2 [VP-2] and vivapain 3 [VP-3]), knowlesipains (knowlesipain 2 [KP-2] and knowlesipain 3 [KP-3]), berghepain 2 [BP-2], chabaupain 2 [CP-2] and yoelipain 2 [YP-2] from *Plasmodium vivax*, *Plasmodium knowlesi*, *Plasmodium berghei*, *Plasmodium chabaudi* and *Plasmodium yoelii*, respectively [276–279]. Together with human cathepsins (Cat. K, Cat. L and Cat. S), these proteins share a common structural feature as well as catalytic mechanism. These proteins are

synthesized as zymogens with their substrate processing activity being tightly regulated in space and time through an occluding prodomain segment which blocks the active site making it inaccessible [280]. Auto-splicing of the prodomain occurs under low pH conditions through the disruption of important residue interactions leading to the activation of the enzymes.

Despite sharing a common structural fold, the *plasmodial* proteases have unusual features compared such as longer prodomains and specific inserts in the catalytic domain-a "nose" (~ 17 amino acids) and an "arm" (~ 14 amino acids). As with other Clan CA family of enzymes, they are characterised by a highly conserved catalytic triad consisting of Cys-His-Asn centrally located in a trench-like cleft at the junction between left (L) and right (R) domains [281]. The L domain is mainly alpha helical and the R domain fold into a β-barrel. Extra residues around these catalytic triad centres also play important roles during the substrate hydrolysis process, and are grouped into four subsites namely S1, S2, S3 and S1' [282]. In spite of similarities between the two protein groups, our previous study revealed key subsite residue composition differences which could be exploited for inhibitor design targeting only the *plasmodial* proteins [283]. The prodomain sections have two highly conserved motifs viz. ERFNIN and GNFD which mediate the inhibition of the catalytic domains as well as maintaining the prodomain structural fold [284].

A detailed homology modeling for both the prodomain-catalytic complex as well as the catalytic domains (*plasmodial* proteases without any crystal structure in the PDB can be found in our previous articles [283,285].

5.6 Maintenance

The 3D protein models in the repository will continue to increase as RUBi researchers continue to deposit modeled structures. Additionally, other members of the research community will also be able to deposit modeled structures through the web server administrators. The accuracy

of these models depends on the availability the templates that are used at the time of modeling. All models were assessed using at least one online tool to ascertain the quality.

5.7 Conclusion

We successfully developed the RUBi protein model repository – an online repository for models of proteins. This repository is only the starting point for providing access to protein models for the public and researchers interested in conducting further research and analysis. In the future, this repository will expand as protein models are added by users. The models will be curated to ascertain their accuracy before being uploaded to the repository. The repository will further be connected to automated homology modeling pipelines – PRIMO and PRIMO-Complexes described in Part 1 of this thesis [87].

CHAPTER SIX

HIV-1 RESPREDICTOR: A WEB APPLICATION EMPLOYING ARTIFICIAL NEURAL NETWORKS TO PREDICT ANTIRETROVIRAL DRUG RESISTANCE IN PATIENTS INFECTED WITH HIV-1 SUBTYPE B

Chapter overview

Human Immunodeficiency Virus type 1 (HIV-1) drug resistance is a big problem for HIV care as it results in the failure of many antiretroviral therapies [286]. Unfortunately, it is difficult, if at all possible, to tell beforehand which patients will develop drug resistance. Monitoring the development of drug resistance is a cumbersome and expensive process that involves complex tests that are sometimes not readily available, especially in many low-income settings [287]. Normally, clinicians use their intuition and clinical acumen to judge drug resistance in patients and switch treatments. This approach does not rule out other potential causes of drug resistance and is ineffective at pinpointing the specific drug in the regimen that the HIV virus might be resistant to [288]. Switching drugs is also not the optimal solution as this may lead to cross-resistance over time.

Predicting drug resistance to HIV-1 viruses before initiation of treatment has improved the care of patients. Recent advances in bioinformatics have improved the accuracy in predicting the potential HIV drug resistance but the procedures have some limitations [289]. Several approaches have been developed to detect drug resistance and these include structural-based, sequence-based, and a combination of both structural and sequence-based methods. It is important to share valuable methods for predicting drug resistance with the research community.

This project was aimed at developing a web-based application for predicting drug resistance among patients with HIV-1 subtype B. This web application was named HIV-1 ResPredictor (HIV-1 **Res**istance **Predictor**). This application uses artificial neural networks (ANNs) to predict drug resistance for HIV-1 subtype B [290]. Since these ANNs were trained on HIV-1 subtype B sequences, subtype classifiers were trained using hidden Markov models (HMMs) to distinguish subtype B sequences from the other subtypes to generate appropriate results.

In this chapter, the focus will be placed on discussing the new additions performed to better this web-based application. The work described here builds on my master's degree work [291]. The following tasks were added: performing the Bland-Altman analysis to analyse the level of agreement between Python-generated models and MATLAB-based ANN models, expanding nucleotide sequences in case they have ambiguous characters before being translated to amino acid sequences, testing the performance of HIV-1 ResPredictor in comparison with similar commonly used subtyping tools and assessing the usability comparison between HIV-1 ResPredictor and other HIV resistance prediction servers. This application returns a summary report of the prediction results which can either be downloaded as a portable document format (PDF) file or received via email. The web application can be freely accessed at https://hiv1respredictor.rubi.ru.ac.za/.

6.1 Introduction

Over the last two decades, there has been a decline in acquired immunodeficiency syndrome (AIDS)-related mortality. This progress is due to a global roll-out of HIV testing and treatment witnessed by 27.5 million people living with HIV receiving antiretroviral therapy (ART) globally in 2020 [292]. Despite the progress, HIV continues to be a major global health crisis. In the year 2020, people living with HIV were approximately 37.7 million with 1.5 million newly infected people and 680,000 deaths from AIDS-related causes globally [293]. Furthermore, there are still challenges in the fight against HIV including prevention of new

infections, diagnosis and expansion of access to cost-effective treatment, the emergence of drug-resistant HIV strains and the existence of comorbidities together with premature ageing in infected patients [294–296].

With regard to drug resistance, there is an increasing number of people with HIV resistant strains, especially among patients with difficulties adhering to prescribed antiretroviral therapy (ART) [288]. Consequently, health workers often need to switch treatments early to second and third-line regimens. This is associated with undesirable side effects, costs and additional challenges to adherence [297,298]. When using conventional clinical examination and standard laboratory tests, it is often difficult to predict which patients will become resistant to drugs and challenging to determine the drug types or combinations to which the virus will become resistant [288]. Monitoring drug resistance development requires sophisticated tests [299], which may not be readily available, particularly in resource-limited settings [287].

HIV viruses develop resistance in multiple ways, including mutations in the genes coding for various proteins, particularly enzymes like protease and reverse transcriptase that regulate replication [300]. Since the discovery of the drug resistance problem in the 1990s, several mechanisms have been put in place to understand how and why it occurs [301]. Researchers have come up with ways to solve this problem and one way is to develop computational approaches for predicting drug resistance [289]. Some of these approaches are structural-based, others are sequence-based while some use a combination of both structural and sequence-based methods [299].

While these approaches have improved the accuracy and alleviated the problem of predicting drug resistance, they are not without challenges. In the past years, several interpretation web applications have been developed, among which HIVdb [302], REGA [303], and the Agence Nationale de Recherche sur le SIDA (ANRS) [304] are more regularly used [304]. Some of

these applications predict genotypic resistance, while others predict phenotypic resistance as well as viral outcome analysis [305]. Some of these applications use specific HIV subtypes hence the need to utilise classification tools that provide support for genetic classification. Examples include Stanford HIV-Seq program [302], the Los Alamos Recombinant Identification Program (RIP) [306], the National Center for Biotechnology Information (NCBI) Genotyping Program [307], REGA [308], COntext-based Modeling for Expeditious Typing (COMET) [309] and the Jumping Profile Hidden Markov Model (jpHMM) [310].

6.2 Research motivation

Since there is rapid growth in the interest of genomics and personalised medicine, it is important to harness the functionality of easy-to-use systems that analyse enormous amounts of sequence data to assess the drug response and disease-related risks of a particular individual. Utilizing the simplicity, user-friendliness and short turnaround time of the systems, researchers and clinicians can make guided decisions including better treatment strategies and monitoring the emergence of drug resistance.

6.3 Research aim and objectives

The aim of this work was to develop a web-based application for predicting HIV-1 subtype B drug resistance to eighteen antiretroviral (ARV) drugs. Whereas there are different HIV-1 subtypes, this application focuses on subtype B ARV drugs thus the need to also develop HIV-1 subtype classifiers using hidden Markov models.

To fulfil this aim, the main objectives included:

1. To develop a mechanism for classifying HIV-1 protease and reverse transcriptase amino acid sequences

2. To translate the ANNs scripts for predicting HIV-1 drug resistance from MATLAB to similar task-performing scripts in Python

3. To develop a web application that predicts drug resistance in patients infected with HIV-1 subtype B

6.4 Methodology

6.4.1 Implementation

HIV-1 ResPredictor was developed using the Django framework whereas the drug resistance models incorporated into the web application were re-implemented in Python programming language. The Python open-sourced library NumPy was used for the matrices and vectors sourced from MATLAB ANN models [290]. The HMM models for subtype classification were also incorporated into this web application [291]. Implementation details of the HIV-1 ResPredictor web application were described in my previous project [291].

6.4.2 Translation of ANN models to Python

The drug resistance prediction models are for three antiretroviral drug classes comprising of eighteen drugs: Protease Inhibitors (PIs) (Atazanavir, Darunavir, Fosamprenavir, Indinavir, Lopinavir, Nelfinavir, Saquinavir, Tipranavir), Nucleoside Reverse Transcriptase Inhibitors (NRTIs) (Abacavir, Didanosine, Lamivudine, Stavudine, Tenofovir, Zidovudine) and Non-Nucleoside Reverse Transcriptase Inhibitors (NNRTIs) (Efavirenz, Etravirine, Nevirapine, Rilpivirine). These models [290] were trained using regression ANN on publicly available data from the Stanford HIV drug resistance database (HIVdb) [311].

The output of ANN models comprises fold resistance scores of the various ARVs, and these scores are translated into a classification (susceptible, intermediate, or resistant) by applying drug-based cut-off values taken from Stanford HIVdb [312]. For consistency among all the

classification classes, the fold resistance scores are normalized and returned alongside the unnormalized score in the result summary report. The normalized values are all in the range 0 to 100, with 40 the cut-off between susceptible and intermediate, and 60 the cut-off between intermediate and resistant. The mapping between fold resistance and normalized fold resistance scores is piecewise linear. For a particular drug, if f is the fold resistance score, with f_{si} and f_{ir} the cut-off values between susceptible and intermediate, and between intermediate and susceptible, respectively; and f_m is the maximum value of f across the ANN training data then the normalized score n is

$$
\begin{aligned}
n &= 0, \qquad (f \le 0) \\
n &= 40\frac{f}{f_{si}}, \qquad (0 < f \le f_{si}) \\
n &= 40 + 20\frac{f - f_{si}}{f_{ir} - f_{si}}, \qquad (f_{si} < f \le f_{ir}) \\
n &= 60 + 40\frac{f - f_{ir}}{f_m - f_{ir}}, \qquad (f_{ir} < f \le f_m) \\
n &= 100, \qquad (f_m < f).
\end{aligned} \tag{1}
$$

Although the values of f in the training data are all in the range 0 to f_m, it may happen that the ANN when applied to a new sequence gives a value outside this range; Eq. (1) allows for this possibility. The accuracy of the resistance prediction ANN models was evaluated using the regression (the coefficient of determination – R_2) method (see [290]). These ANN drug resistance prediction models initially trained in MATLAB were adapted and re-implemented in Python before being incorporated into the web application. This is because Python is open source, has a large user support community, and is more readily available. MATLAB, on the other hand, is proprietary.

6.4.3 Determination of levels of agreement between MATLAB and translated Python scripts

A Bland-Altman analysis [313,314] was used to assess the level of agreement between the Python-generated models and the ANN models generated previously in MATLAB. Drug resistance prediction was done for ten sequences for the protease (PR) and reverse transcriptase (RT) enzymes for each drug prediction model both in MATLAB and Python.

Considering that the ANN resistance prediction models were trained and tested on subtype B data, subtype classifiers were also incorporated into the HIV-1 ResPredictor application.

6.4.4 Development of hidden Markov models to classify HIV-1 subtypes

These subtype classifiers were trained using HMMs on the same datasets used to train ANNs [290] with a few more sequences added to the PR dataset from NCBI [315] to improve its performance and accuracy. RT and PR models were trained using 267,398 and 119,206 subtype B sequences respectively. HMMs can be used to solve three types of statistical problems (evaluation, decoding or uncovering and the estimation problem) [316–318]. The machine learning problem being solved is the estimation or learning problem given that we had to find the most suitable HMM $\lambda = (A, B, \pi)$ that maximizes the probability of obtaining the observation sequences of the training set.

Generally, an HMM consists of the following.

1. A hidden Markov chain, i.e.
    - a set of hidden states. In this case, these are the positions of the amino acids, coded as numbers 1 to 99 for the PR and 1 to 240 for the RT.
    - a transition probability matrix A, the entry a_{ij} giving the probability for a transition from hidden state i to hidden state j. In this case, the left end

of the sequence is considered the starting point, where $a_{ij}=1$ is for two consecutive hidden states and $a_{ij}=0$ otherwise.

- an initial distribution π for the hidden states. In this case, the left end of the sequence is considered the starting point, where $\pi(1) = 1$ and all other π's are zero.

2. An observed process, i.e.

    - an alphabet of observed states. In our case, these are the amino acids in the sequences, coded with the one-letter code.

    - an emission probability matrix B, giving for each hidden state j the probability of emission of the observed states. In our case, these emission probabilities were estimated from the training set as the relative frequencies of the amino acids at each position.

Thus, from our training set, the HMM is fixed. Given a new observed sequence, the probability of this observation under the model (A, B, π) can be calculated efficiently using the forward algorithm [317,318] as encoded in the MATLAB function *hmmdecode* [319]. In this case, however, there were no insertions, deletions, or gaps in the sequences, this was simplified by calculating the product of the emission probabilities for each amino acid in the sequence. The more complicated algorithm [318] was used for possible adaptation to aligned sequences with insertions at a later stage. The probability of observing a new sequence was obtained using MATLAB's *hmmdecode* function [319] in the form *[PSTATES, logpseq] = hmmdecode (sequence, TPM, EPM)*, with the logarithm returned as *logpseq*. Python scripts were written that re-implemented this function in our application and they were tested to cross-check whether the same logarithm probability was returned as for the MATLAB function (to machine precision).

6.4.5 Determination of a cut-off for subtyping sequences

The thresholds for classifying the sequences as either subtype B or non-B subtypes were determined using receiver operating characteristics (ROC) analysis [320,321]. The log probabilities for both HIV-1 subtype B and non-B subtype sequences were generated using similar datasets that were used to train the subtype classifier models and sequences from NCBI [315] respectively. The non-B subtype sequences comprised all group M amino acid sequences for both PR and RT excluding circulating recombinant forms (CRFs) and unique recombinant forms (URFs) that were mixed with subtype B to avoid contamination. These sequences were truncated to 99 and 240 residues for PR and RT respectively to conform to the format of the sequence alignments used during HMM training and drug resistance prediction ANN models [290]. The non-B subtype dataset comprised 20,499 and 12,551 sequences for PR and RT virus respectively. Given that our data is highly unbalanced with skewed sample distributions, the area under the curve (AUC) was more favourable to use [320] compared to precision-recall estimates of accuracy. An R script was used to calculate the AUC, threshold, and the corresponding confusion matrix. This was done in three subtype categories i.e., B vs A, B vs C, B vs all non-B for both PR and RT. All non-B stands for subtype A and C combined.

6.4.6 Performance testing of HIV-1 ResPredictor with existing subtyping tools

The performance of our subtype classification models was compared with similar commonly used automated subtyping tools: a similarity-based tool (Stanford HIV drug resistance database (HIVdb) [322]); phylogenetics based tools (subtype classification using evolutionary algorithms (SCUEAL) [323] and Rega HIV subtyping tool (REGA v3) [324]); and a partial matching compression prediction algorithm (context-based modeling for expeditious typing (COMET)) [309]. This was done using a total of 4000 HIV-1 nucleotide PR and RT polymerase (*pol*) sequences retrieved from NCBI [315], with 1000 sequences in each subtype category i.e., subtype B and all non-B subtypes for both PR and RT. The nucleotide sequences were used to

allow for comparison with other existing subtyping tools, some of which only accept sequences in this format. *Pol* sequences were also used because our server and some other tools like SCUEAL only subtype sequences from this gene. The PR and RT sequences were truncated to 297 and 720 base pairs (bp) to conform to our server-accepted length. For evaluation purposes, the sensitivity (Eq. 2), specificity (Eq. 3), positive predictive value (PPV; Eq. 4), negative predictive value (NPV; Eq. 5) and accuracy (Eq. 6) were calculated for each tool.

$$\text{Sensitivity} = TP/[TP + FN] \quad (2)$$

$$\text{Specificity} = TN/[TN + FP] \quad (3)$$

$$\text{Positive predictive value (PPV)} = TP/[TP + FP] \quad (4)$$

$$\text{Negative predictive value (NPV)} = TN/[TN + FN] \quad (5)$$

$$\text{Accuracy} = [TP + TN]/[TP + FN + TN + FP] \quad (6)$$

Where TP, TN, FN, and FP are the true positive, true negative, false negative and false positive respectively.

6.5 Results and discussion

The web application, HIV-1 ResPredictor (Fig 5.1) was developed to be a user-friendly system that can be used by both novice and expert users. HIV-1 ResPredictor provides drug resistance prediction for HIV-1 sequences for 18 ARVs and 2 subtype classification classifiers since the predictions were trained and tested for subtype B. The details about the system and software design, development, evaluation of the subtype classifier models were described in my previous work [291] but mentioned here to provide context.

6.5.1 Initial page and input formatting

The web server takes input as PR or RT amino acid or nucleic acid sequence(s) in FASTA format. The sequence length and type are first assessed for validity. If a nucleotide sequence is submitted, it is translated into an amino acid sequence to conform to the drug resistance prediction ANN models' format before predictions are done. In instances where the nucleotide sequence contains ambiguous bases such as R, Y, S, W, K, M, B, D, H, V, N, and ./-, a pre-expansion step is carried out following the International Union of Pure And Applied Chemistry (IUPAC) convention. For example, an ambiguous base 'W' means it is either a Thymine('T') or Adenine('A'), as such when a sequence 'ATGW' is given it is expanded to two sequences 'ATGT' and 'ATGA'. Each expanded sequence is considered a single input sequence.

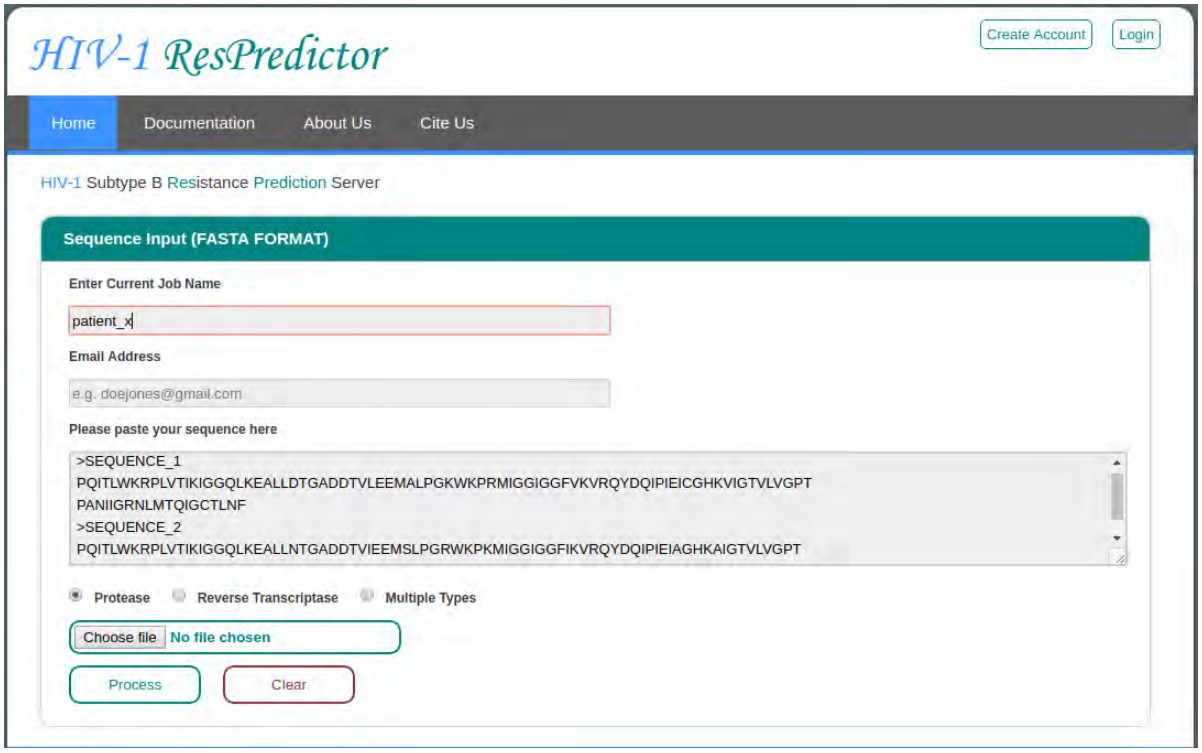


Fig 6.1. HIV-1 ResPredictor web interface. Sequence(s) in FASTA format can either be pasted in the input box or uploaded as a file to the application. The type of input sequence(s) should be selected by the user before proceeding.

The workflow of the HIV-1 ResPredictor is shown in Fig 6.2. Several sequences can be submitted at the same time only if they are amino acid sequences and nucleic acid sequences

without ambiguous bases. Each amino acid is numerically encoded as in [290], and conserved columns are removed as done during ANN training [290].

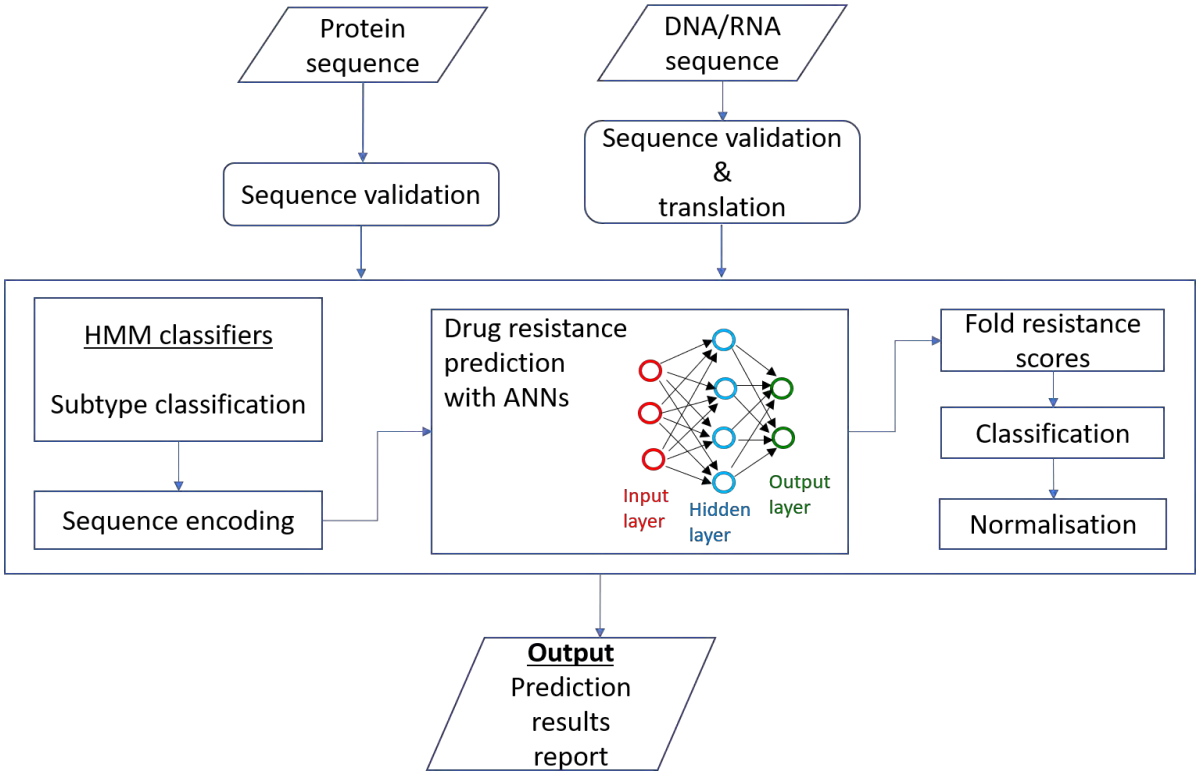


Fig 6.2. Workflow of the HIV-1 ResPredictor web application. Protein sequences or nucleic acid sequences are accepted as input, validated and/or translated to protein sequence for nucleic acid sequence(s). The main step involves subtype classification, sequence encoding and drug resistance prediction along with fold resistance score classification and normalization. A prediction summary report is finally returned.

The ANN prediction models incorporated in HIV-1 ResPredictor were re-implemented in Python and both these and the MATLAB models were assessed to ascertain that the results are the same. Bland-Altman plots were generated for each drug-based model and used to assess the level of agreement as shown in Fig 6.3 (one PR and RT inhibitor) and S6.1 Fig (PIs) and S6.2 Fig (NRTIs and NNRTIs). The mean difference for all the drug-based Bland-Altman plots was zero indicating that the MATLAB models [290] and Python models agree and do measure the same. In this case, the mean of two measurements was used because it's the best estimate [314] since the true resistance score is unknown.

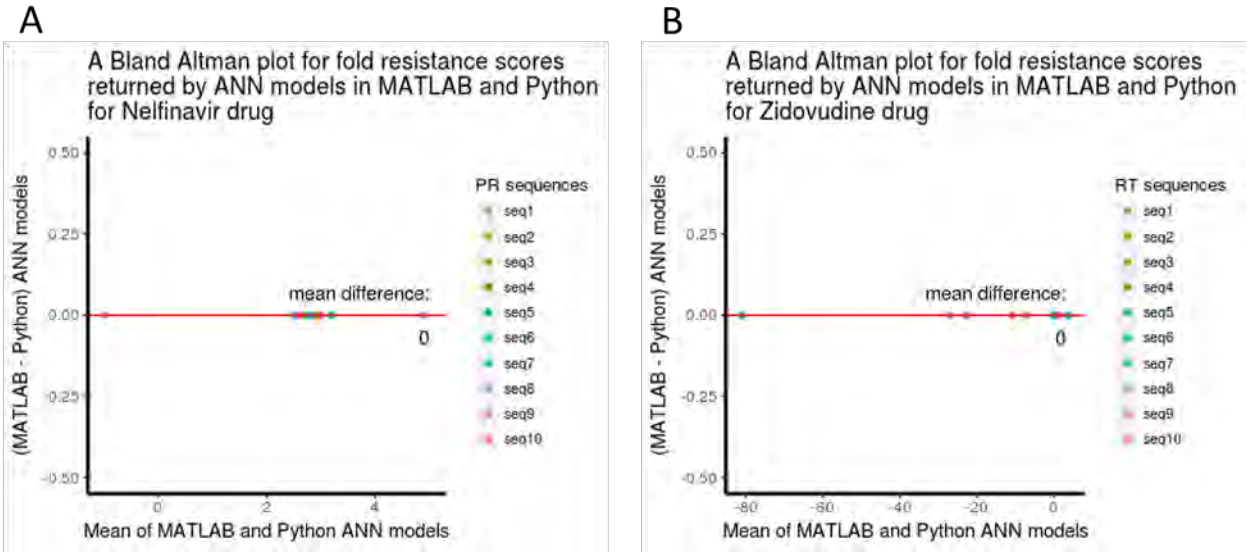


Fig 6.3. Bland Altman plots. Plots of differences between MATLAB and Python ANN models vs the mean of the two methods of protease inhibitor (6.3A) and reverse transcriptase inhibitor (6.3B). The red line represents the mean difference between these two methods.

6.5.2 Performance of HIV-1 subtype classifiers

The HIV-1 subtype classifiers were trained using HMMs to distinguish subtype B sequences from non-B subtypes since the drug resistance prediction ANN models were trained and tested on subtype B data. A threshold with the corresponding sensitivity, specificity and area under the curve (AUC) was determined for each group as shown in Table 6.1. For both PR and RT, the AUC for subtype B versus subtype A was better (67% and 93% respectively) than all the other analysis groups. Overall, the classifier for the RT enzyme performed better than that for the protease enzyme and this may have been due to the difference in the size of the training datasets.

Table 6.1. Summary of the important parameters and values obtained in classifying the subtype of protease and reverse transcriptase sequences.

	PR (Subtype)			RT (Subtype)		
Parameter analysed	**B vs A**	**B vs C**	**B vs non-B**	**B vs A**	**B vs C**	**B vs non-B**
Threshold (log probabilities)	-27.6	-30.9	-29.7	-51.0	-51.1	-50.0
Sensitivity (%)	44	56	52	77	77	75
Specificity (%)	82	74	67	44	81	80
AUC (%)	67	66	61	93	85	86

6.5.3 Results page

The sequences are then processed, and a summary report is returned as shown in Fig 5.2. First, the sequence subtype classification is made and reported as part of the final summary report. Even if the subtype is not B, the application subsequently predicts resistance of the sequence to the various ARVs, but with a comment to the user about the reliability of the predictions.

A fold resistance score is returned for each PR or RT inhibitor depending on the type of sequence(s) being processed, with the corresponding normalized fold resistance score and classification as “susceptible”, “intermediate” or “resistant”. Since the fold resistance classification cut-off values vary among drugs, the fold resistance scores are normalized. The user can download the results as a pdf file, or the results can be sent via email. To demonstrate the performance of the HIV-1 ResPredictor, an RT subtype B sequence was processed to predict its resistance to HIV-1 RT inhibitors. As shown in Fig 5.4, a report is generated on the web page with details about the queried sequence. This sample sequence was classified as subtype B with a sensitivity of 75% and specificity of 80%. A summary table for the fold

resistance scores with corresponding normalized fold resistance scores and classification is displayed for each RT inhibitor. The results in the drug resistance prediction summary table can be reordered either alphabetically for the ARV drugs and drug resistance class, or numerically ascending for fold resistance score.

Sequence Type : Reverse Transcriptase Sequence

Comment : Classification of HIV-1 subtypes was done using Hidden Markov Models. With a sensitivity of 75% and a specificity of 80%, our classifier distinguished this sequence as subtype B. These results might be unreliable.

Sequence :

PISPIETVPVKLKPGMDGPKVKQWPLTEEKIKALVEICTEMEKEGKISKIGPENPYNTPVFAIKKKNSTGWRKLVDFRELNKRTQDFWEVQLGIPHPAGLK
QKKSVTVLDVGDAYFSVPLDEDFRKYTAFTIPSVNNETPGIRYQYNVLPQGWKGSPAIFQSSMTKILEPFRKQNPEIVIYQYVDDLYVGSDLEIEQHRTKV
EELRQHLLRWGFFTPDQKHQKEPPFLWMGYELHPDKWT

Reistance :

Anti-retroviral Drug	Fold Resistance Score	Normalized Fold Resistance Score	Drug Resistance Class
Abacavir	4	53	Intermediate
Didanosine	1	36	Susceptible
Efavirenz	3	42	Intermediate
Etravirine	1	14	Susceptible
Lamivudine	199	100	Resistant
Nevirapine	2	33	Susceptible
Rilpivirine	19	100	Resistant
Stavudine	0	25	Susceptible
Tenofovir	0	24	Susceptible
Zidovudine	12	56	Intermediate

Fig 6.4. Resistance prediction results of reverse transcriptase sequence against reverse transcriptase inhibitors. The summary report shows the sequence type, subtype classification comment, submitted sequence, and drug resistance prediction summary table.

6.5.4 Comparison of subtyping tools performance results

Whereas the classification results are reported from our application, their accuracy and that of the routinely used subtyping tools (COMET, SCUEAL, HIVdb, and REGA v3) was evaluated as reported in Table 6.2 and Stanford HIVdb performed better than all the other tools. Hence a link for Stanford HIVdb is provided for concordance and in case not, better subtyping results

are obtained. Nevertheless, this does not affect the performance of the drug resistance prediction models as these don't depend on the subtype classification results although the accuracy of the prediction for other subtypes is unknown.

Table 6.2. Comparison of performance of HIV-1 ResPredictor with existing automated subtyping tools: classification of protease and reverse transcriptase sequences

Enzyme type	Metrics (%)	HIV-1 ResPredictor	HIVdb v8.6	COMET	SCUEAL	REGA v3
PR	**Sensitivity**	90.8	99.9	98.7	71.8	84.4
	Specificity	76.3	99.4	95.3	99.4	60.6
	PPV	79.3	99.4	95.5	99.2	68.2
	NPV	89.2	99.9	98.7	77.9	79.5
	Accuracy	83.6	99.7	97.0	85.6	72.5
RT	**Sensitivity**	82.3	99.9	99.9	93.7	95.3
	Specificity	93.5	93.1	91.3	93.8	77.9
	PPV	92.6	93.5	91.9	93.8	81.2
	NPV	84.1	99.9	99.9	93.7	94.3
	Accuracy	87.9	96.5	95.6	93.8	86.6

The accuracy of HIV-1 ResPredictor to classify sequences as subtype B or not B was tested in comparison with other routinely used subtyping tools (COMET, SCUEAL, HIVdb, and REGA v3). When the tools were tested with PR data, HIVdb had the highest sensitivity (99.9%), followed by COMET (98.7%), HIV-1 ResPredictor (90.8%), REGA v3 (84.4%), and SCUEAL

(71.8%). HIVdb and SCUEAL had the highest specificity (99.4%) followed by COMET (95.4), HIV-1 ResPredictor (76.3%), and REGA v3 (60.6%) as shown in Table 6.2.

When the tools were tested with RT data, it was noted that HIVdb, COMET, REGA and SCUEAL had a higher sensitivity (≥93.7%) than HIV-1 ResPredictor (82.3%). As seen in Table 6.3, the specificity values of HIVdb, COMET, HIV-1 ResPredictor and SCUEAL were higher (≥91.3%) than that of REGA v3 (77.9%). It was noted that all the tools had unassigned sequences. For a sequence to be designated as unassigned in HIV-1 ResPredictor, it had to have discordant results for at least one outcome sequence after expansion. SCUEAL labelled such sequences as 'complex', HIVdb as 'not applicable' ('NA'), COMET as 'unassigned' and REGA v3 as 'check the report'.

HIVdb had the best predictive algorithm for positive predictive values (PPV=99.4%) for PR sequences while for RT sequences, SCUEAL had the best (PPV=93.8%) although with a very insignificant difference from HIVdb (PPV=93.5%). Regarding negative predictive values (NPV), HIVdb had the best predictive score (NPV=99.9%) for PR sequences while for RT sequences, both HIVdb and COMET were the best (NPV=99.9%). For PR sequences, HIV-1 ResPredictor was better than both SCUEAL and REGA v3 to properly identify a sequence as a certain subtype when it is indeed of that subtype (NPV=89.2%). The accuracy (ability to correctly assign a subtype to a sequence) of subtyping tools was estimated using the formula in equation 5 above. HIVdb had the best accuracy for both PR (99.7%) and RT sequences (96.5%), followed by COMET (97% and 95.6% for PR and RT sequences respectively), SCUEAL (85.6% and 93.8% for PR and RT sequences respectively), HIV-1 ResPredictor (83.6% and 87.9% for PR and RT sequences respectively) and REGA v3 (72.5% and 86.6% for PR and RT sequences respectively).

A difference between the accuracy of the subtype classifiers incorporated in HIV-1 ResPredictor and their performance in comparison to other subtyping tools was noted. This might be because nucleotide sequences were used in the comparison with other subtyping tools whereas when determining their accuracy, amino acid sequences were used. These nucleotide sequences may or may not have ambiguous characters which have to be expanded before being translated to amino acid sequences. This led to discordant results from the expanded sequences hence lowering the performance of our subtype classifiers.

HIVdb presents the highest accuracy for B and non-B subtypes compared to other subtyping tools. All the tools had relatively high sensitivity and specificity values for both PR and RT sequences except REGA v3 that had the lowest specificity values for both PR and RT sequences. Our subtyping classifiers performed relatively well compared to the best subtyping tools given that they had some limitations. When training our subtyping classification models, subtype B sequences used were retrieved from HIVdb which resulted in good models. Despite the relatively good performance, our classifiers had some limitations because they were trained only on subtype B data. The classifiers were also trained only on *pol* sequences with specific lengths that are used for drug resistance prediction. No specific features were considered when training our models using HMM, yet other subtyping tools extensively utilise all the necessary information they need to perform the subtyping.

Whereas our subtype classifiers are reported and still used even without the best classification results in comparison with other subtyping tools, a link to the best tool which was Stanford HIVdb was added. Users are advised to further check the subtype of their sequences from the Stanford HIVdb server in case they were not sure of the subtype. This doesn't affect the drug resistance prediction results although the accuracy is only known for subtype B sequences which might be misleading. Our work regarding subtype classification using HMM can further

be explored and improved for better performance since it shown that it is feasible with relatively good and promising results.

In the previous work [290] done by colleagues in our lab, the accuracy of our prediction models in comparison with Shen and colleagues models [299] was demonstrated hence only testing usability in this work. A comparison was done between our server and other drug resistance prediction servers (HIVdb, geno2pheno$_{[resistance]}$, HIV-GRADE) as shown in Table 6.3. A comparison was done for the different drug classes together with checking whether mutation information is given or returned, the maximum number of sequences to be uploaded, time response and whether an informative report is returned. Apart from HIV-1 ResPredictor and geno2pheno $_{resistance}$, HIVdb and HIV-GRADE accept 500 and <1000 nucleotide sequences respectively. Only HIVdb and HIV-GRADE predict for all the drug classes considered. A difference was also noted in the run times (averaged over 5 runs) for all the prediction servers when tested on 15 PR nucleotide sequences. HIV-1 ResPredictor turns out to be the slowest tool, which is mainly because nucleotide sequences have to be translated to amino acid sequences given nature of the prediction models incorporated. The fastest tool was HIVdb with 1.4 seconds for the 15 nucleotide sequences tested

Table 6.3. Usability comparison between HIV-1 ResPredictor and other HIV resistance prediction servers

Prediction Server	PIs	NRTIs	NNRTIs	INIs	Mut	Max No. sequences	Time response[c] (Secs)	Clinical report
HIV-1 ResPredcitor	+	+	+	-	-	20[a] 1[b]	14.9	+
HIVdb	+	+	+	+	+	500	1.4	+
geno2pheno[resistance]	+	+	+	-	+	20	7.9	+
HIV-GRADE	+	+	+	+	+	>1000	2.9	+

[a]Amino acid sequences and nucleotide sequences without ambiguous characters

[b]Only nucleotide sequence with an ambiguous character

[c]Averaged over 5 runs with 15 sequences

6.6 Conclusion

HIV-1 ResPredictor is a user-friendly web application that enables users to make drug resistance predictions in patients infected with HIV-1, although the accuracy of the predictions is known only if the infection is of subtype B. The application includes a subtype classifier developed using HMM. Our classifier performed relatively well in classifying sequences as either subtype B or non-B subtype, similar to most tools apart from HIVdb that consistently performed well for all cases. From this analysis, clinicians and researchers are advised to use our server with other high performance tools like HIVdb to classify sequences and get a consensus.

This web application makes ANN models, that were reported previously [290], accessible to a wide range of users. In the era of personalised medicine and genomics, it is increasingly important to facilitate easy-to-use tools that can be used to analyse sequence data. Within

patient care environments, our application has uses as a point of care decision-support tool, aiding physicians in selecting the best ARV regimen on which new patients could be started or onto which "ARV-experienced" patients could be switched if the need arises. Beyond personalised medicine, this application has uses in research and clinical practice environments as well as public health, potentially reducing the cases of treatment failure and cross-resistance, and their associated morbidities, mortality, and resistance-associated costs. Further improvement of the ANN models will be done periodically following the availability of more data to enhance the accuracy of our server predictions.

CHAPTER SEVEN

DEVELOPMENT OF A BIOINFORMATICS EDUCATION WEB PORTAL

Chapter overview

Often students, researchers as well as academic staff search for bioinformatics information from different sources including e-journals, theses, and software. Availing education portals for easy access to bioinformatics learning materials would be of great importance to the community. Most bioinformatics applications require tools to analyse and carry out research, however students lack skills to make use of these tools. Equipping students with knowledge on how to use bioinformatics basic tools helps them as they progress in their careers. To address this, a website was designed to provide bioinformatics basic information on how to use the different online tools and databases. This website entails information from protocols written by former MSc and PhD students in RUBi under the guidance of Prof Özlem Taştan Bishop. This web portal can be accessed at https://learn-bioinfo.rubi.ru.ac.za/.

7.1 Introduction

Electronic learning (e-learning) started way back in the 1960s and has evolved over time. In academic context, e-learning refers to the mode of learning that depends on online communication and technologies [325]. Almost all universities embrace e-learning as the mode of communication between students and the academic staff. Education research studies involve the use of literature, designing experiments and analysing data to answer scientific questions. It was recommended by National Research Council that there is a need to transform undergraduate life sciences education in the 21st century, and students should work with real data and tools for life sciences research [326]. Enormous volumes of data have been accumulated in the databases due to technological advancements, but this does not match the

analysis pace. These can be used as a starting point to generate positive research outcomes hence the need to introduce bioinformatics tools to students [327].

One of the courses that integrate practical skills in the traditional teaching paradigm is bioinformatics. Application-oriented bioinformatics courses have been introduced at the undergraduate and graduate levels within existing courses including biology, computer science, physics, mathematics, and biochemistry [328–330]. Undergraduate students have been introduced to bioinformatics basic concepts, tools, and databases. Bioinformatics reinforces active learning by interconnecting theoretical lectures and laboratory sessions.

7.2 Motivation

Students normally face problems when carrying out practical lessons in the computer laboratory for the bioinformatics module. University laboratories normally do not have specific software installed to perform bioinformatics research. A strategy to overcome this problem is to use online tools and databases to avoid technical problems. However, being aware of these resources is not enough, unless the students gain a deeper understanding of the bioinformatics methods as well as guidance on how to fully utilise them.

7.3 Research aim and objectives

This work was aimed to familiarise students with the basic terminology and approaches in structural bioinformatics by availing this information via an education web portal. This is advantageous to students since the knowledge of bioinformatics skills can be carried on even after graduation.

The main objective of this work was to design and develop an education web portal for sharing the bioinformatics web resources and databases.

7.4 Implementation

The website was developed using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) with a bootstrap framework and JavaScript. The website content consists of information from different protocols on how to use bioinformatics tools and databases to research real-life sciences data. Different protocols are presented using a bootstrap treeview which shows hierarchical information. Each protocol is presented as a root node, which has parent nodes in this case the protocol subtopics with children being information under each subtopic.

Protocols contain detailed descriptions of how to apply bioinformatics tools to biological problems. These protocols were written by former MSc and PhD students in RUBi based on a template prepared by Prof Özlem Taştan Bishop. This web portal can be accessed at https://learn-bioinfo.rubi.ru.ac.za/.

7.5 Results

This website features protocols that have detailed descriptions of how to apply bioinformatics tools to problems as well as case studies. All protocols demonstrated the use of online web servers for calculations to avoid the issue of getting specialised software installed in the computer laboratories. These protocols include NCBI and BLAST, multiple sequence alignment (MSA), protein data bank (PDB), visualization of protein structures, homology detection and structure prediction (HHpred), PRIMO pipeline, protein-ligand interaction, protein motif analysis with MEME suite tools, physico-chemical properties, protein interactions calculator (PIC), ROBETTA alanine scanning, and single nucleotide polymorphism (SNP) effect prediction.

The initial page gives information about what the web portal offers to the education sector and a list of all the protocols provided (Fig 6.1).

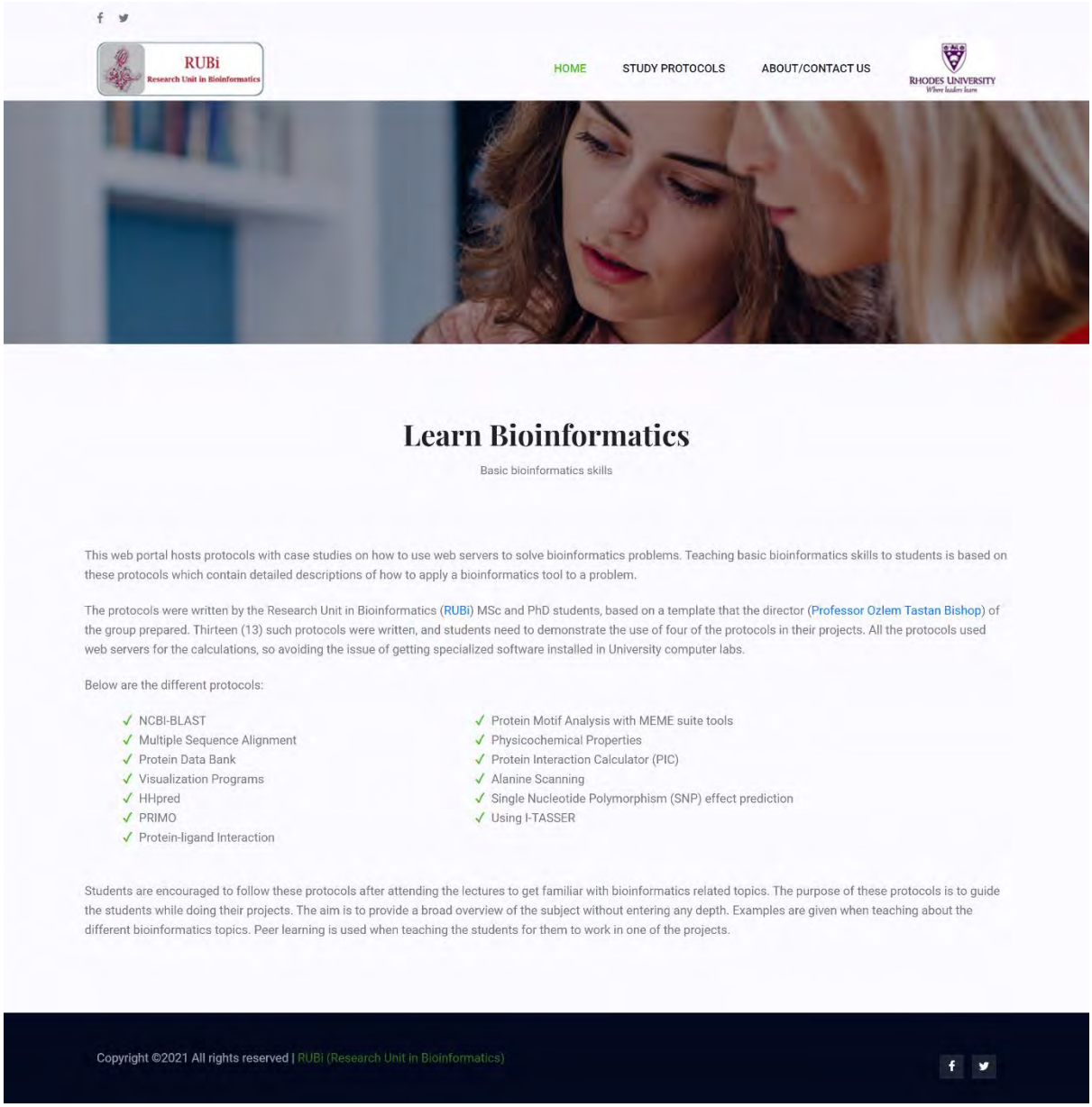


Fig 7.1. Home page for the bioinformatics web portal. This page entails information on what this website offers.

The study protocols page consists of information including an overview for each protocol (Fig 7.2), and detailed steps to achieve the goals of each protocol (Fig 7.3 and Fig 7.4). Each protocol is displayed in a treeview format where information is displayed once the arrow is clicked to open the tree branch and hidden when clicked again.

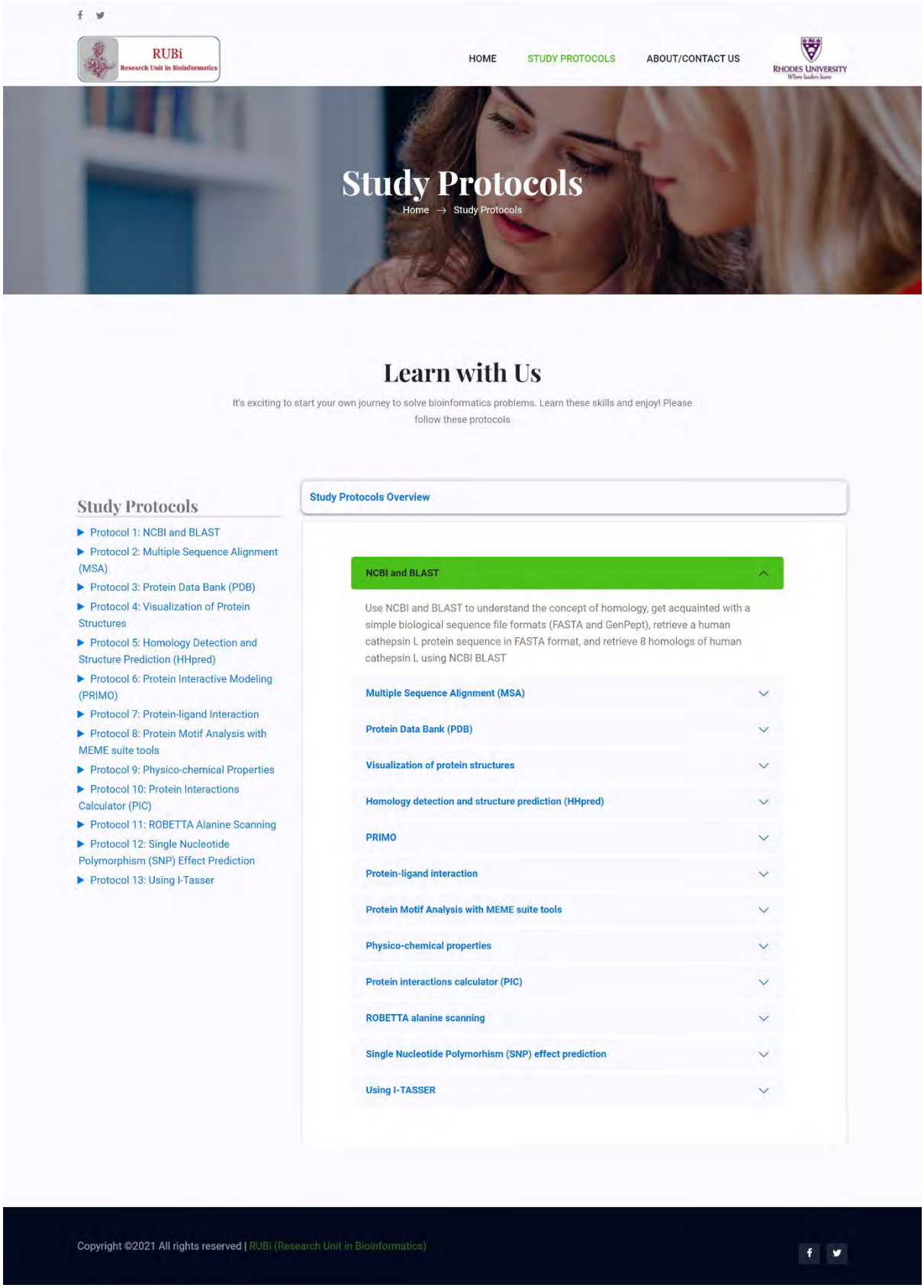


Fig 7.2. Study protocols page showing the overall overview of the bioinformatics protocols.

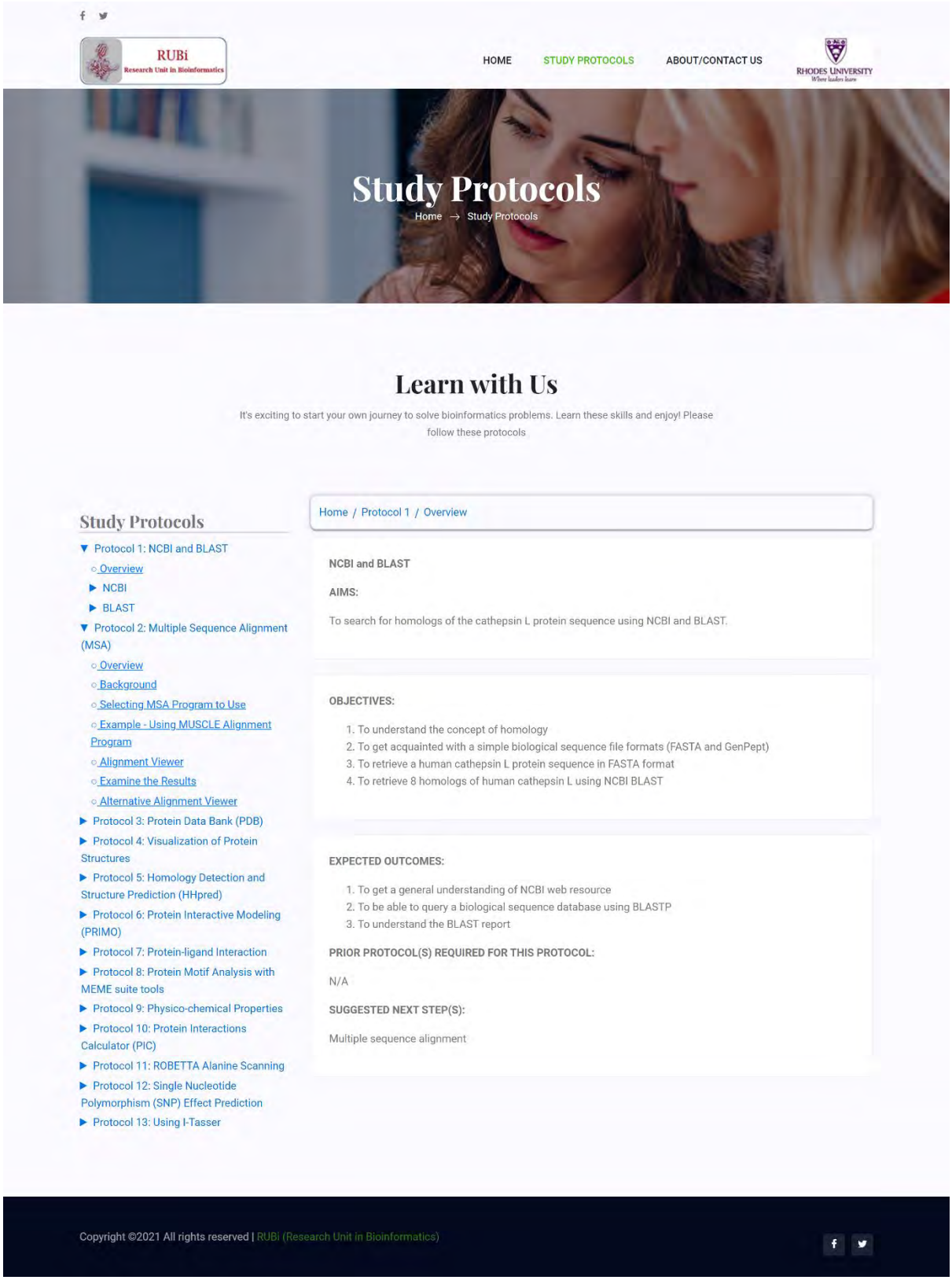


Fig 7.3. Study protocols page showing information on how to use NCBI and BLAST.

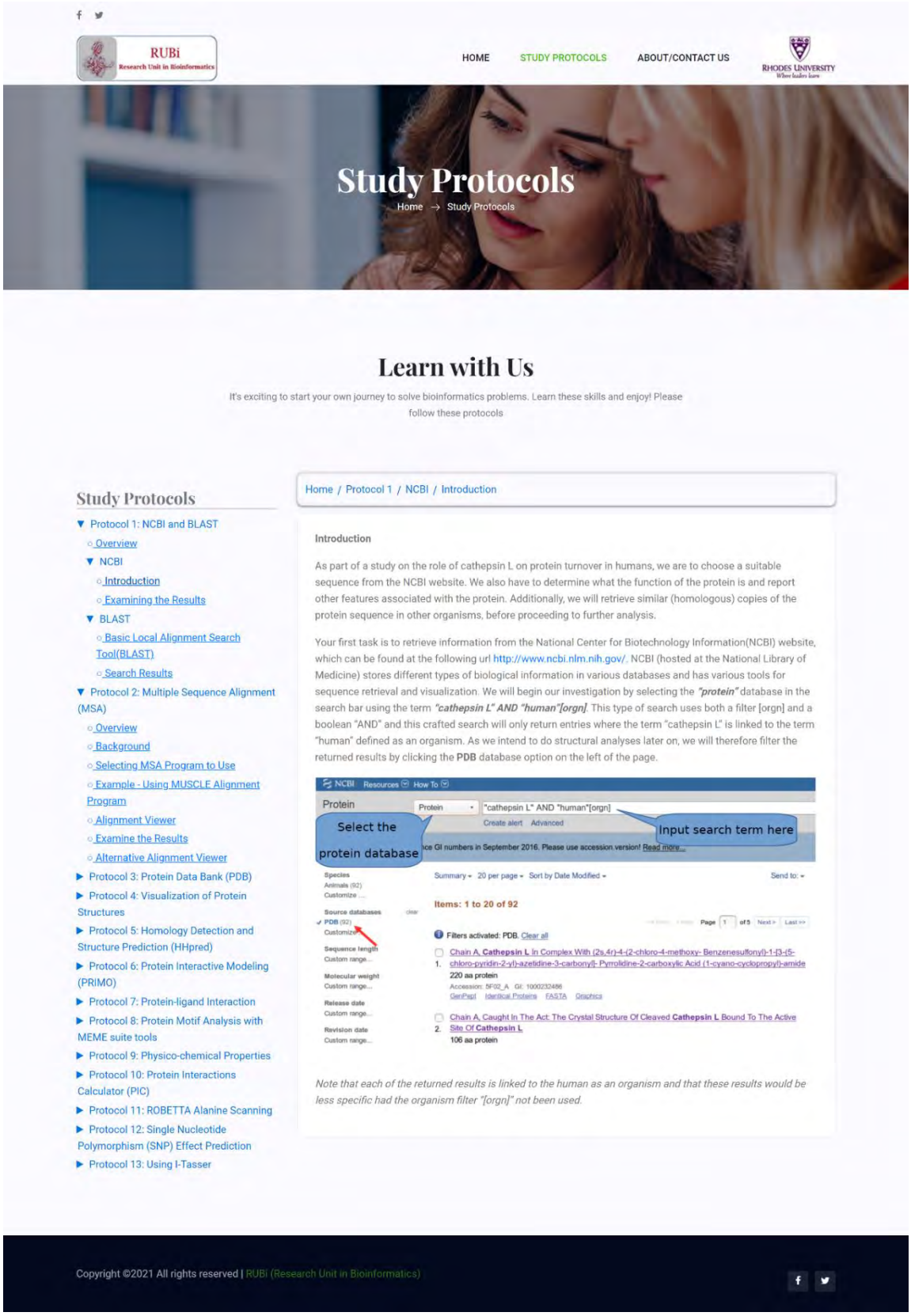


Fig 7.4. Study protocols page showing some of the information in the NCBI and BLAST protocol.

These protocols provide step by step detailed examples of using bioinformatics tools hence helping students analyse results of different studies leading to biological insights. When using these tools students do not need to install software since all the case studies involve online tools and databases.

The study protocols page is interactive, and each protocol has rich graphical learning content with well-elaborated information. This web portal enables students to gain a deeper and critical understanding of bioinformatics resources, techniques, and their applications so that they appreciate the strengths and limitations of these tools. These protocols also have open-ended laboratory exercises designed to help students grasp the key concepts and gain practical skills.

7.6 Conclusion

This website will help undergraduate students learn about and how to use several bioinformatics resources. Students will also be able to use these resources at their pace and convenience hence creating flexibility during the learning process. Learners will participate actively in the laboratory since they can access the information before the practical sessions. These web-based tools ease the teaching process of bioinformatics at universities since the need to install and pay for licenses of the proprietary software is eliminated.

CONCLUSIONS, FUTURE WORK, AND REFERENCES

8.1 Conclusions

In this work, we have successfully developed a tool – PRIMO-Complexes that models protein complexes and biological assemblies. This web server has been designed with user friendly features such as an interactive web interface, helpful guiding information at every stage during the modeling process and full control of the modeling process. While PRIMO-Complexes uses PDB files to model protein complexes, the PDB database curators are phasing them out. Further development and research in automating protein modeling should be focused on learning how to parse data from PDBx/mmCIF files and integrate these algorithms in the existing web servers. PRIMO-Complexes avails users with the option to model large protein complexes like viral capsids. This is only the start, and more additions can be made to equip and ease the cumbersome process of installing software like PyMOL to align these large structures.

Other tools and web servers that aid the process of performing bioinformatics research were developed including RUBi protein model repository, HIV-1 ResPredictor web application, and a bioinformatics education web portal. The RUBi protein repository stores verified theoretical protein models that were generated by researchers in RUBi. To the research community, this is a great advantage since it ensures reproducibility of research work. This work can be expanded by other developers whereby a large repository can be setup with curators to receive and verify protein models from the research community. This repository serves as a starting point to assist researchers in effortlessly exploring the 3D protein space to do further research and analysis.

The HIV-1 ResPredictor application that facilitates drug resistance predictions in patients infected with HIV-1 subtype B was developed. Drug resistance is still a big problem in the fight to eradicate HIV infections and mortality hence the need to devise ways to mitigate the

problem. In future, HIV-1 ResPredictor application can further be developed to increase sensitivity and provide more functionality to predict drug resistance. These easy to use systems should be adapted to benefit from intensified research in personalised medicine.

A bioinformatics education web portal was also successfully setup to help students familiarise themselves with the basic terminologies and approaches in structural bioinformatics. This portal will enable undergraduate students and researchers to learn how to use several online bioinformatics resources. Knowledge about the existence of this information, online resources and tools is not enough however, the existence of education web portals such as this one help students and researchers to fully utilise them. This web portal can further be expanded to include more information and active discussion forums for the researchers.

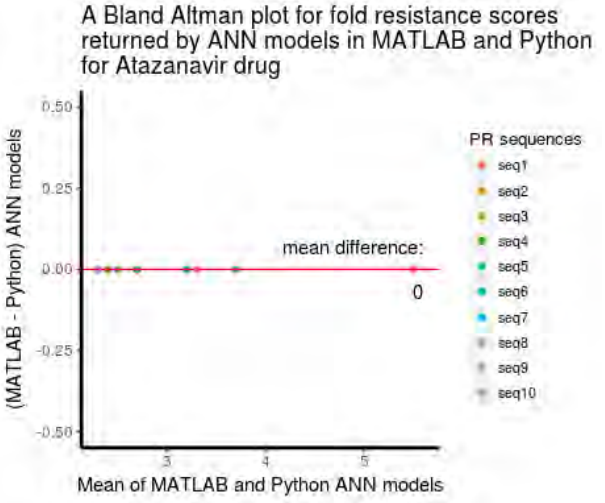
8.2 Future work

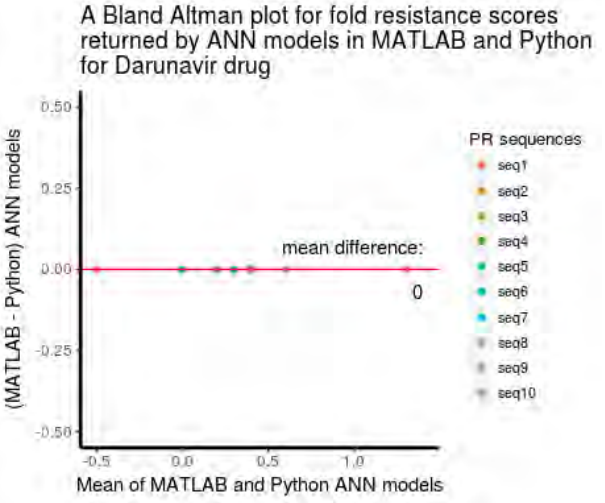
In future, development could focus on linking the PRIMO-Complexes to Human Mutation Analysis (HUMA) web server and database [331] to analyse new disease-related variations. We will also add more options for each stage such as allowing the user to specify a biological assembly template. In addition to other improvements based on user requests. The two PRIMO versions (modeling monomers and multimeric proteins) will be merged so that the user can easily access both pipelines.

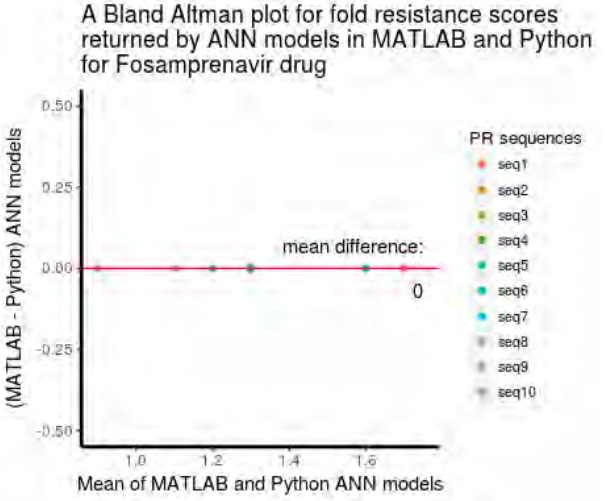
The RUBi protein model repository will grow regularly as models are added by users before or after being published. The models will be curated to ascertain the accuracy before being uploaded to the repository. The repository will further be connected to automated homology modeling pipelines – PRIMO and PRIMO-Complexes.

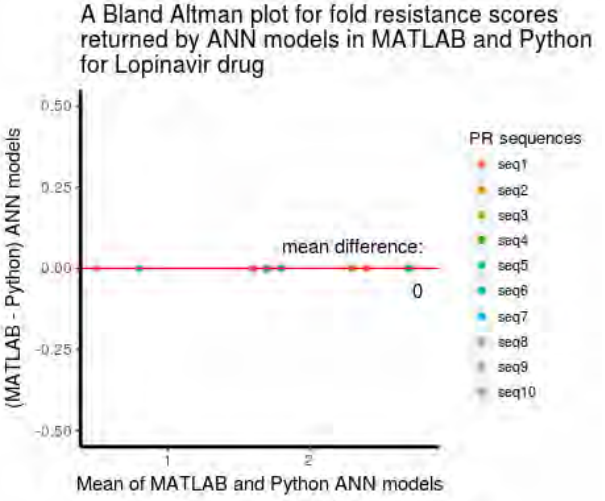
Supplementary data

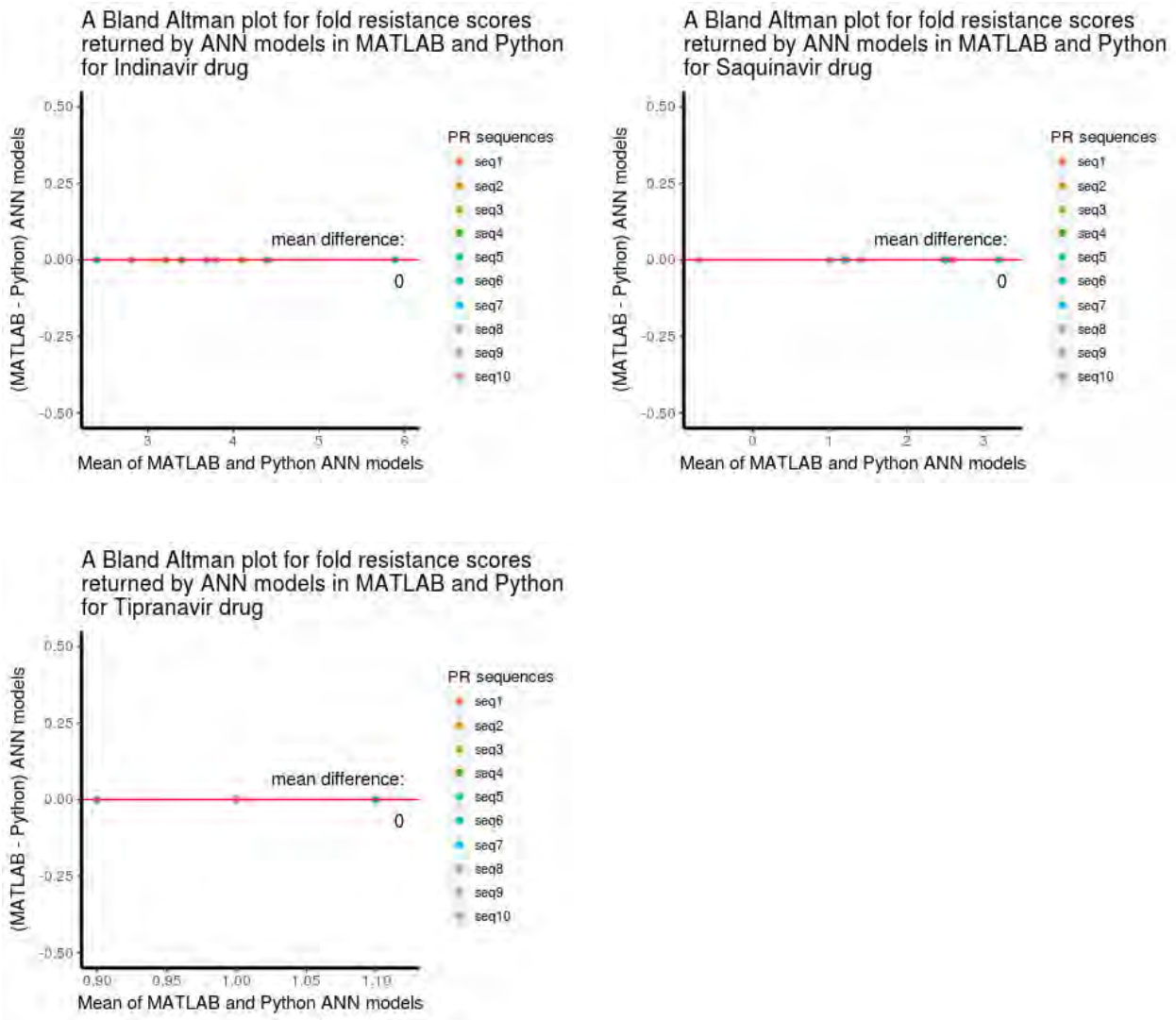
Chapter 5 supplementary data



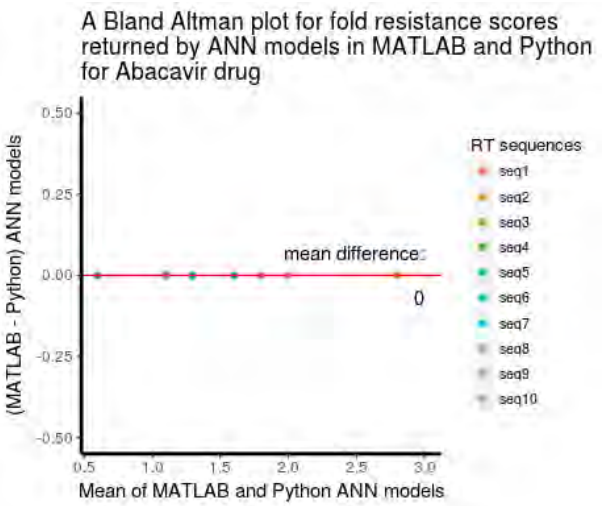


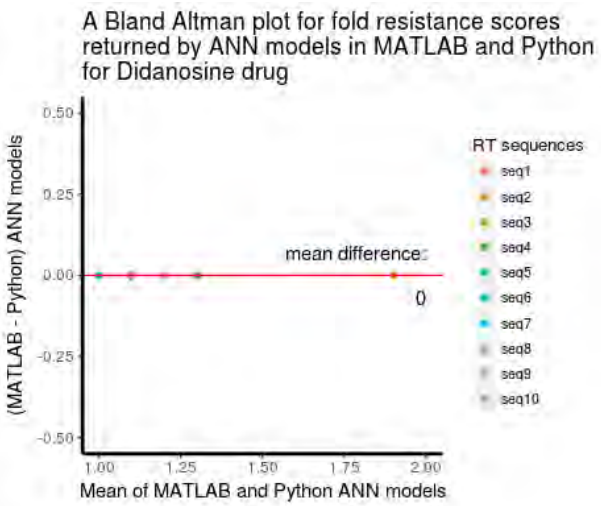


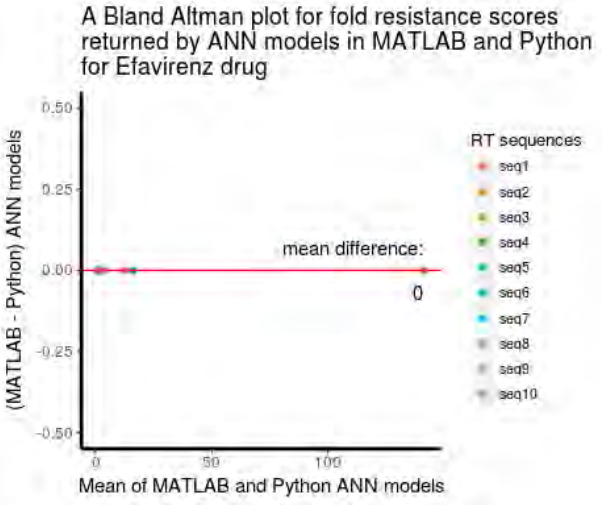


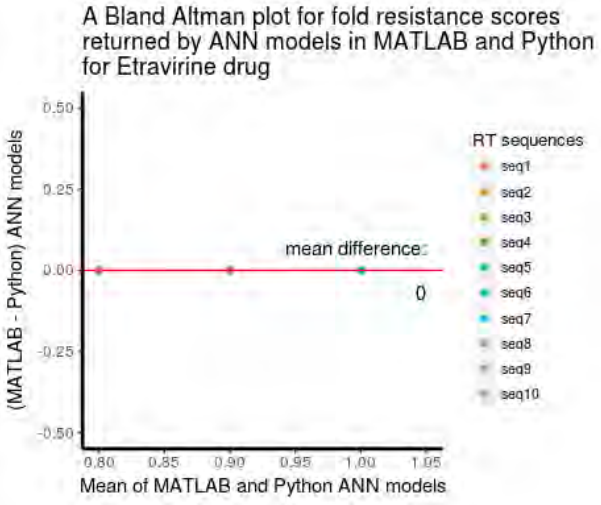


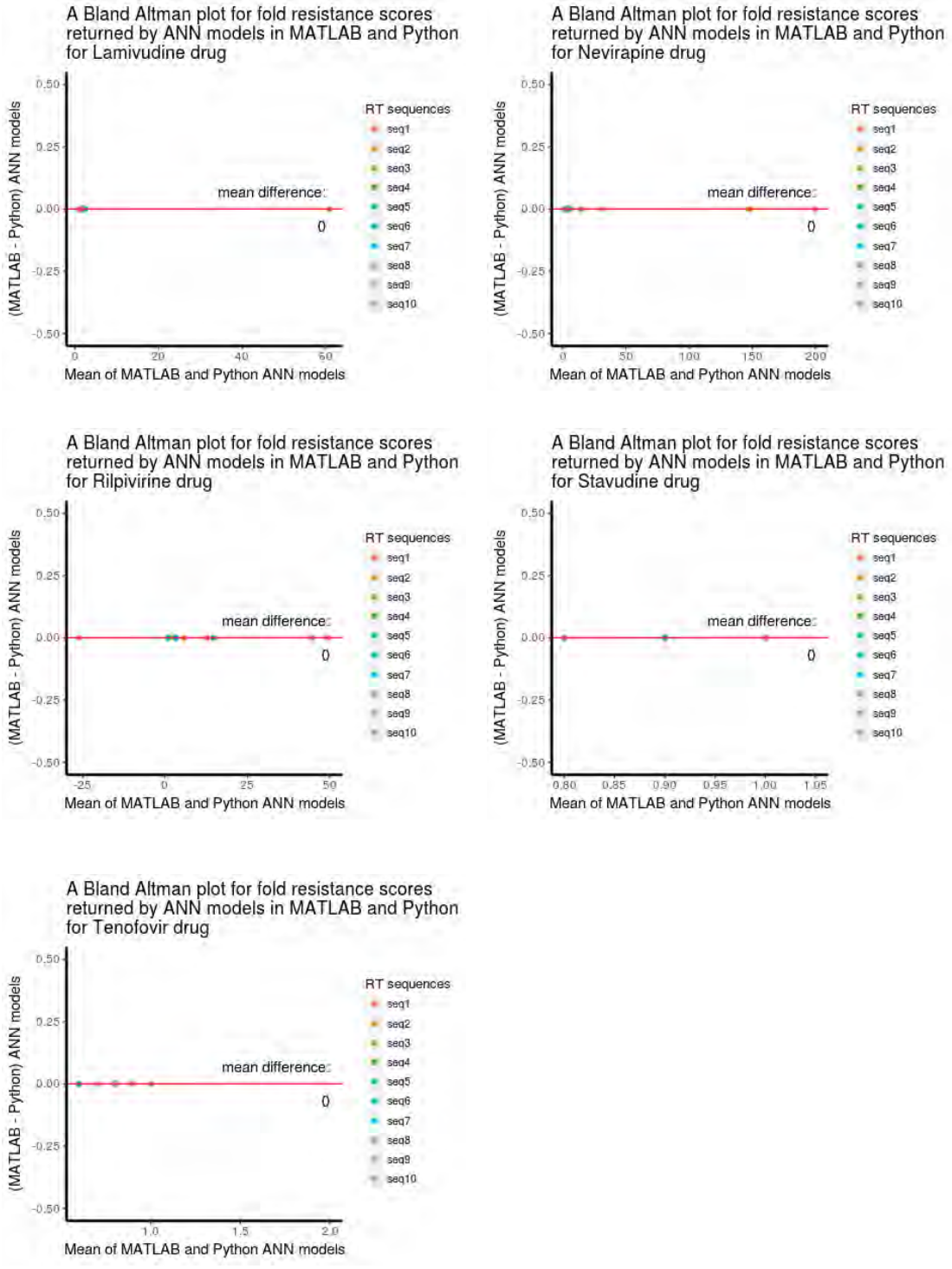
S6.1 Fig. Bland-Altman plots for protease drug resistance prediction models using R. Drug resistance prediction was done for ten protease sequences for each drug-based model in MATLAB and Python.


A Bland Altman plot for fold resistance scores returned by ANN models in MATLAB and Python for Abacavir drug
(MATLAB - Python) ANN models
0.50
0.25
0.00
-0.25
-0.50
mean difference:
0
0.5
1.0
1.5
2.0
2.5
3.0
Mean of MATLAB and Python ANN models
RT sequences
seq1
seq2
seq3
seq4
seq5
seq6
seq7
seq8
seq9
seq10


A Bland Altman plot for fold resistance scores returned by ANN models in MATLAB and Python for Didanosine drug
(MATLAB - Python) ANN models
0.50
0.25
0.00
-0.25
-0.50
mean difference:
0
1.00
1.25
1.50
1.75
2.00
Mean of MATLAB and Python ANN models
RT sequences
seq1
seq2
seq3
seq4
seq5
seq6
seq7
seq8
seq9
seq10


A Bland Altman plot for fold resistance scores returned by ANN models in MATLAB and Python for Efavirenz drug
(MATLAB - Python) ANN models
0.50
0.25
0.00
-0.25
-0.50
mean difference:
0
0
50
100
Mean of MATLAB and Python ANN models
RT sequences
seq1
seq2
seq3
seq4
seq5
seq6
seq7
seq8
seq9
seq10


A Bland Altman plot for fold resistance scores returned by ANN models in MATLAB and Python for Etravirine drug
(MATLAB - Python) ANN models
0.50
0.25
0.00
-0.25
-0.50
mean difference:
0
0.80
0.85
0.90
0.95
1.00
1.05
Mean of MATLAB and Python ANN models
RT sequences
seq1
seq2
seq3
seq4
seq5
seq6
seq7
seq8
seq9
seq10



S6.2 Fig. Bland-Altman plots for reverse transcriptase drug resistance prediction models using R. Drug resistance prediction was done for ten reverse transcriptase sequences for each drug-based model in MATLAB and Python.

References

1. Nelson DL, Cox MM. Principles of Biochemistry. 4th editio. New York: W. H. Freeman & Company; 2005.

2. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. Molecular Biology of the Cell. 4th editio. New York: Garland Science; 2002.

3. Andreeva A, Howorth D, Chothia C, Kulesha E, Murzin AG. SCOP2 prototype: a new approach to protein structure mining. Nucleic Acids Res. 2014;42: D310-4. doi:10.1093/nar/gkt1242

4. Whisstock JC, Lesk AM. Prediction of protein function from protein sequence and structure. Q Rev Biophys. 2003;36: 307–340. doi:10.1017/S0033583503003901

5. MCQ Biology.com. Multiple choice on proteins. 2021. Available: https://www.mcqbiology.com/2012/11/mcq-on-biochemistry-proteins.html

6. Sanvictores T, Farci F. Biochemistry, Primary Protein Structure. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2020. Available: https://www.ncbi.nlm.nih.gov/books/NBK564343/

7. Rehman I, Farooq M, Botelho S. Biochemistry, Secondary Protein Structure. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2020. Available: https://www.ncbi.nlm.nih.gov/books/NBK470235/

8. Whitford D. Proteins: Structure and Function. John Wiley & Sons; 2013.

9. Rehman I, Kerndt CC, Botelho S. Biochemistry, Tertiary Protein Structure. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2021. Available: https://www.ncbi.nlm.nih.gov/books/NBK470269/

10. Banach M, Konieczny L, Roterman I. Secondary and Supersecondary Structure of Proteins in Light of the Structure of Hydrophobic Cores. In: Kister A. (eds) Protein Supersecondary Structures. Methods in Molecular Biology. Humana Press, New York, NY; 2019. doi:https://doi.org/10.1007/978-1-4939-9161-7_19

11. Ouellette RJ, Rawn JD. Principles of Organic Chemistry. Elsevier Inc; 2015. doi:https://doi.org/10.1016/C2014-0-02430-6

12. Karplus M, Kuriyan J. Molecular dynamics and protein function. Proc Natl Acad Sci U S A. 2005;102: 6679–6685. doi:10.1073/pnas.0408930102

13. Hub JS, De Groot BL. Detection of functional modes in protein dynamics. PLoS Comput Biol. 2009;5. doi:10.1371/journal.pcbi.1000480

14. Zimei B, David JEC. Advances in Protein Chemistry and Structural Biology. 2011.

15. Henzler-Wildman K, Kern D. Dynamic personalities of proteins. Nature. 2007;450: 964–972. doi:10.1038/nature06522

16. Kendrew JC, Bodo G, Dintzis HM, Parrish RG, Wyckoff H, Phillips DC. A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. Nature. 1958;181: 662–666. doi:10.1038/181662a0

17. Perutz MF, F.R.S, Rossmann MG, Cullis AF, Muirhead H, Will G, et al. Structure of

Hæmoglobin: A Three-Dimensional Fourier Synthesis at 5.5-Å. Resolution, Obtained by X-Ray Analysis. Nature. 1960;185: 416–422. doi:10.1038/185416a0

18. Smyth MS, Martin JHJ. X-Ray Crystallography. J Clin Pathol Mol Pathol. 2000;53: 8–14. doi:https://doi.org/10.1136/mp.53.1.8

19. Gernert KM, Smith R, Carter DC. A simple apparatus for controlling nucleation and size in protein crystal growth. Anal Biochem. 1988;168: 141–147. doi:10.1016/0003-2697(88)90021-8

20. Luft JR, Arakali S V., Kirisits MJ, Kalenik J, Wawrzak I, Cody V, et al. Macromolecular crystallization procedure employing diffusion cells of varying depths as reservoirs to tailor the time course of equilibration in hanging- and sitting-drop vapor-diffusion and microdialysis experiments. J Appl Crystallogr. 1994;27: 443–452. doi:10.1107/S0021889893012713

21. Otwinowski, Zbyszek, Minor W. Processing of X-ray diffraction data collected in oscillation mode. Methods Enzymol. 1997;276: 307–326. doi:https://doi.org/10.1016/S0076-6879(97)76066-X

22. Ten Eyck LF. Crystallographic Fast Fourier Transforms. Acta Crystallogr Sect A. 1973;29: 183–191. doi:10.1107/S0567739473000458

23. Terwilliger TC, Berendzen J. Automated MAD and MIR structure solution. Acta Crystallogr Sect D Biol Crystallogr. 1999;55: 849–861. doi:10.1107/S0907444999000839

24. Perrakis A, Morris R, Lamzin VS. Automated protein model building combined with iterative structure refinement. Nat Struct Biol. 1999;6: 458–463. doi:10.1038/8263

25. Adams PD, Grosse-Kunstleve RW, Hung LW, Ioerger TR, McCoy AJ, Moriarty NW, et al. PHENIX: building new software for automated crystallographic structure determination. Acta Crystallogr Sect D Biol Crystallogr. 2002;58: 1948–1954. doi:10.1107/S0907444902016657

26. Minor W, Cymborowski M, Otwinowski Z, Chruszcz M. HKL-3000: the integration of data reduction and structure solution - from diffraction images to an initial model in minutes. Acta Crystallogr Sect D Biol Crystallogr. 2006;62: 859–866. doi:10.1107/S0907444906019949

27. Yang H, Guranovic V, Dutta S, Feng Z, Berman HM, Westbrook JD. Automated and accurate deposition of structures solved by X-ray diffraction to the Protein Data Bank. Acta Crystallogr Sect D Biol Crystallogr. 2004;60: 1833–1839. doi:10.1107/S0907444904019419

28. Williamson MP, Havel TF, Wiithrich K. Solution conformation of proteinase inhibitor IIA from bull seminal plasma by 1H nuclear magnetic resonance and distance geometry. J Mol Biol. 1985;182: 295–315. doi:10.1016/0022-2836(85)90347-x

29. Sprangers R, Kay LE. Quantitative dynamics and binding studies of the 20S proteasome by NMR. Nature. 2007;445: 618–622. doi:10.1038/nature05512

30. Mainz A, Jehle S, Van Rossum BJ, Oschkinat H, Reif B. Large protein complexes with extreme rotational correlation times investigated in solution by magic-angle-spinning NMR spectroscopy. J Am Chem Soc. 2009;131: 15968–15969. doi:10.1021/ja904733v

31. Kennedy MA, Montelione GT, Arrowsmith CH, Markley JL. Role for NMR in structural genomics. J Struct Funct Genomics. 2002;2: 155–169. doi:10.1023/A:1021261026670

32. Wüthrich K. The way to NMR structures of proteins. Nat Struct Biol. 2001;8: 923–925. doi:10.1038/nsb1101-923

33. Romero P, Obradovic Z, Dunker AK. Sequence data analysis for long disordered regions prediction in the calcineurin family. Genome informatics. 1997;8: 110–124. doi:10.11234/gi1990.8.110

34. Romero P, Obradovic Z, Kissinger CR, Villafranca JE, Garner E, Guilliot S, et al. Thousands of proteins likely to have long disordered regions. Pacific Symp Biocomput. 1998;3: 437–448.

35. Wright PE, Dyson HJ. Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm. J Mol Biol. 1999;293: 321–331. doi:10.1006/jmbi.1999.3110

36. Romero P, Obradovic Z, Li X, Garner EC, Brown CJ, Dunker AK. Sequence complexity of disordered protein. Proteins Struct Funct Bioinforma. 2001;42: 38–48. doi:10.1002/1097-0134(20010101)42:1<38::AID-PROT50>3.0.CO;2-3

37. Skrynnikov NR, Goto NK, Yang D, Choy WY, Tolman JR, Mueller GA, et al. Orienting domains in proteins using dipolar couplings measured by liquid-state NMR: Differences in solution and crystal forms of maltodextrin binding protein loaded with β-cyclodextrin. J Mol Biol. 2000;295: 1265–1273. doi:10.1006/jmbi.1999.3430

38. Tugarinov V, Sprangers R, Kay LE. Line narrowing in Methyl-TROSY using zero-quantum 1 H- 13 C NMR spectroscopy. J Am Chem Soc. 2004;126: 4921–4925. doi:https://doi.org/10.1021/ja039732s

39. Bertini I, Luchinat C, Parigi G, Ravera E, Reif B, Turano P. Solid-state NMR of proteins sedimented by ultracentrifugation. Proc Natl Acad Sci U S A. 2011;108: 10396–10399. doi:10.1073/pnas.1103854108

40. Henderson R, Unwin PNT. Three-dimensional model of purple membrane obtained by electron microscopy. Nature. 1975;257: 28–32. doi:https://doi.org/10.1038/257028a0

41. Zhang X, Settembre E, Xu C, Dormitzer PR, Bellamy R, Harrison SC, et al. Near-atomic resolution using electron cryomicroscopy and single-particle reconstruction. Proc Natl Acad Sci United States Am. 2008;105: 1867–1872. doi:10.1073/pnas.0711623105

42. Yu X, Jin L, Zhou ZH. 3.88 Å structure structure of cytoplasmic polyhedrosis virus by cryo-electron microscopy. Nature. 2008;453: 415–419. doi:10.1038/nature06893

43. Unwin PNT, Henderson R. Molecular structure determination by electron microscopy of unstained crystalline specimens. J Mol Biol. 1975;94: 425–440. doi:10.1016/0022-2836(75)90212-0

44. Adrian M, Dubochet J, Lepault J, McDowall AW. Cryo-electron microscopy of viruses. Nature. 1984;308: 32–36. doi:10.1038/308032a0

45. Bhella D. Cryo-electron microscopy: an introduction to the technique, and considerations when working to establish a national facility. Biophys Rev. 2019;11:

515–519. doi:10.1007/s12551-019-00571-w

46. Dubochet J, Adrian M, Chang J-J, Homo J-C, Lepault J, McDowall AW, et al. Cryo-electron microscopy of vitrified specimens. Q Rev Biophys. 1988;21: 129–228. doi:10.1017/S0033583500004297

47. Callaway E. The revolution will not be crystallized: a new method sweeps through structural biology. Nature. 2015;525: 172–174. doi:10.1038/525172a

48. Vénien-Bryan C, Li Z, Vuillard L, Boutin JA. Cryo-electron microscopy and X-ray crystallography: Complementary approaches to structural biology and drug discovery. Acta Crystallogr Sect Struct Biol Commun. 2017;73: 174–183. doi:10.1107/S2053230X17003740

49. Yip KM, Fischer N, Paknia E, Chari A, Stark H. Atomic-resolution protein structure determination by cryo-EM. Nature. 2020;587: 157–161. doi:10.1038/s41586-020-2833-4

50. C.Bernstein F, F.Koetzle T, R.Rodgers, J.B.Williams G, F.MeyerJr. E, D.Brice M, et al. The Protein Data Bank : A Computer-based Archival File for Macromolecular Structures. J Mol Biol. 1977;112: 535–542. doi:https://doi.org/10.1016/S0022-2836(77)80200-3

51. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Res. 2000;28: 235–242. doi:https://doi.org/10.1093/nar/28.1.235

52. Berman H, Henrick K, Nakamura H. Announcing the worldwide Protein Data Bank. Nat Struct Mol Biol. 2003;10: 2003. doi:https://doi.org/10.1038/nsb1203-980

53. Berman H, Henrick K, Nakamura H, Markley JL. The worldwide Protein Data Bank (wwPDB): ensuring a single , uniform archive of PDB data. Nucleic Acids Res. 2007;35: D301–D303. doi:10.1093/nar/gkl971

54. Kinjo AR, Bekker G, Suzuki H, Tsuchiya Y, Kawabata T, Ikegawa Y, et al. Protein Data Bank Japan (PDBj): updated user interfaces , resource description framework , analysis tools for large structures. Nucleic Acids Res. 2017;45: D282–D288. doi:10.1093/nar/gkw962

55. Mir S, Alhroub Y, Anyango S, Armstrong DR, Berrisford JM, Clark AR, et al. PDBe : towards reusable data delivery infrastructure at protein data bank in Europe. Nucleic Acids Res. 2018;46: D486–D492. doi:10.1093/nar/gkx1070

56. Ulrich EL, Akutsu H, Doreleijers JF, Harano Y, Ioannidis YE, Lin J, et al. BioMagResBank. Nucleic Acids Res. 2008;36: D402–D408. doi:10.1093/nar/gkm957

57. wwPDB consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. Nucleic Acids Res. 2019;47: D520–D528. doi:10.1093/nar/gky949

58. RCSB Protein Data Bank. PDB Data Distribution by Experimental Method and Molecular Type. In: H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. (2000) The Protein Data Bank Nucleic Acids Research, 28: 235-242. [Internet]. [cited 26 Oct 2021]. Available: https://www.rcsb.org/stats/summary

59. Bränd́en CI, Alwyn Jones T. Between objectivity and subjectivity. Nature. 1990;343: 687–689. doi:10.1038/343687a0

60. Sippl MJ. Recognition of errors in three-dimensional structures of proteins. Proteins Struct Funct Bioinforma. 1993;17: 355–362. doi:10.1002/prot.340170404

61. Venclovas Č, Ginalski K, Kang C. Sequence-structure mapping errors in the PDB: OB-fold domains. Protein Sci. 2004;13: 1594–1602. doi:10.1110/ps.04634604

62. Dauter Z, Wlodawer A, Minor W, Jaskolski M, Rupp B. Avoidable errors in deposited macromolecular structures: an impediment to efficient data mining. Int Union Crystallogr. 2014;1: 179–193. doi:10.1107/S2052252514005442

63. Burley SK, Berman HM, Bhikadiya C, Bi C, Chen L, Di Costanzo L, et al. RCSB Protein Data Bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. Nucleic Acids Res. 2019;47: D464–D474. doi:10.1093/nar/gky1004

64. J Callaway, M Cummings, B Deroski, P Esposito AF. Protein Data Bank contents guide: Atomic coordinate entry format description. Upt (New York) US Dep Energy Brookhaven Natl Lab. 1996.

65. Philip E. Bourne, Berman HM, McMahon B, Watenpaugh KD, D.Westbrook J, Fitzgerald PMD. Macromolecular crystallographic information file. Methods Enzymol. 1997;277: 571–590. doi:https://doi.org/10.1016/S0076-6879(97)77032-0

66. Westbrook JD, Bourne PE. STAR/mmCIF: An ontology for macromolecular structure. Bioinfomatics Ontol. 2000;16: 159–168.

67. Westbrook J, Ito N, Nakamura H, Henrick K, Berman HM. PDBML : the representation of archival macromolecular structure data in XML. Boinformatics. 2005;21: 988–992. doi:10.1093/bioinformatics/bti082

68. Young JY, Westbrook JD, Feng Z, Sala R, Peisach E, Oldfield TJ, et al. OneDep: Unified wwPDB System for Deposition, Biocuration, and Validation of Macromolecular Structures in the PDB Archive. Structure. 2017;25: 536–545. doi:10.1016/j.str.2017.01.004

69. Bateman A, Martin MJ, Orchard S, Magrane M, Agivetova R, Ahmad S, et al. UniProt: The universal protein knowledgebase in 2021. Nucleic Acids Res. 2021;49: D480–D489. doi:10.1093/nar/gkaa1100

70. Anfinsen CB. Principles that Govern the Folding of Protein Chains. Science (80-). 1973;181: 223–230.

71. Bajaj M, Blundell T. Evolution and the tertiary structure of proteins. Annu Rev Biophys Bioeng. 1984;13: 453–492. doi:https://doi.org/10.1146/annurev.bb.13.060184.002321

72. Illergård K, Ardell DH, Elofsson A. Structure is three to ten times more conserved than sequence - A study of structural response in protein cores. Proteins Struct Funct Bioinforma. 2009;77: 499–508. doi:10.1002/prot.22458

73. Abagyan RA, Batalov S. Do aligned sequences share the same fold? J Mol Biol. 1997;273: 355–368. doi:10.1006/jmbi.1997.1287

74. Grishin N V. Fold change in evolution of protein structures. J Struct Biol. 2001;134: 167–185. doi:10.1006/jsbi.2001.4335

75. Luccio E, Koehl P. A quality metric for homology modeling : the H-factor. BMC Bioinformatics. 2011;12. doi:https://doi.org/10.1186/1471-2105-12-48

76. Martí-Renom MA, Stuart AC, Fiser A, Sánchez R, Melo F, Šali A. Comparative protein structure modeling of genes and genomes. Annu Rev Biophys Biomol Struct. 2000;29: 291–325. doi:10.1146/annurev.biophys.29.1.291

77. Sheng Y, Šali A, Herzog H, Lahnstein J, Krilis S. Modelling, expression and site-directed mutagenesis of human β2-glycoprotein I. Identification of the major phospholipid binding site. J Immunol. 1996;157: 3744–51.

78. Ring CS, Sun E, Mckerrow JH, Lee GK, Rosenthal PJ, Kuntz ID, et al. Structure-based inhibitor design by using protein models for the development of antiparasitic agents. Proc Natl Acad Sci U S A. 1993;90: 3583–3587. doi:10.1073/pnas.90.8.3583

79. Xu LZ, Sánchez R, Sali A, Heintz N. Ligand specificity of brain lipid-binding protein. J Biol Chem. 1996;271: 24711–24719. doi:10.1074/jbc.271.40.24711

80. Boissel J-P, Lee W-R, Presnell SR, Cohen FE, Bunn HF. Erythropoietin structure-function relationships. Mutant proteins that test a model of tertiary structure. J Biol Chem. 1993;268: 15983–15993. doi:10.1016/s0021-9258(18)82348-1

81. Wu G, Fiser A, Ter Kuile B, Šali A, Müller M. Convergent evolution of Trichomonas vaginalis lactate dehydrogenase from malate dehydrogenase. Proc Natl Acad Sci U S A. 1999;96: 6285–6290. doi:10.1073/pnas.96.11.6285

82. Vakser IA. Evaluation of GRAMM low-resolution docking methodology on the hemagglutinin-antibody complex. Proteins Struct Funct Genet. 1997;29: 226–230. doi:10.1002/(SICI)1097-0134(1997)1+<226::AID-PROT31>3.0.CO;2-O

83. Muhammed MT, Aki-Yalcin E. Homology modeling in drug discovery: Overview, current applications, and future perspectives. Chem Biol Drug Des. 2019;93: 12–20. doi:10.1111/cbdd.13388

84. Larsson P, Wallner B, Lindahl E, Elofsson A. Using multiple templates to improve quality of homology models in automated homology modeling. Protein Sci. 2008;17: 990–1002. doi:10.1110/ps.073344908

85. Kryshtafovych A, Moult J, Tramontano A. Evaluation of the template-based modeling in CASP12. Proteins Struct Funct Bioinforma. 2017;86: 321–334. doi:10.1002/prot.25425

86. Croll TI, Sammito MD, Kryshtafovych A, Read RJ. Evaluation of template-based modeling in CASP13. Proteins Struct Funct Bioinforma. 2019;87: 1113–1127. doi:10.1002/prot.25800

87. Hatherley R, Brown DK, Glenister M, Bishop ÖT. PRIMO: An interactive homology modeling pipeline. PLoS One. 2016;11. doi:10.1371/journal.pone.0166698

88. Waterhouse A, Bertoni M, Bienert S, Studer G, Tauriello G, Gumienny R, et al. SWISS-MODEL: Homology modelling of protein structures and complexes. Nucleic Acids Res. 2018;46: W296–W303. doi:10.1093/nar/gky427

89. Söding J, Biegert A, Lupas AN. The HHpred interactive server for protein homology detection and structure prediction. Nucleic Acids Res. 2005;33: W244–W248. doi:10.1093/nar/gki408

90. Lumry R, Eyring H. Conformation changes of proteins. J Phys Chem. 1954;58: 110–120.

91. Dhingra S, Sowdhamini R, Cadet F, Offmann B. A glance into the evolution of template-free protein structure prediction methodologies. Biochimie. 2020;175: 85–92. doi:10.1016/j.biochi.2020.04.026

92. Jones DT, McGuffin LJ. Assembling Novel Protein Folds From Super-secondary Structural Fragments. Proteins Struct Funct Genet. 2003;53: 480–485. doi:10.1002/prot.10542

93. Simons KT, Kooperberg C, Huang E, Baker D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J Mol Biol. 1997;268: 209–225. doi:10.1006/jmbi.1997.0959

94. Kuhlman B, Bradley P. Advances in protein structure prediction and design. Nat Rev Mol Cell Biol. 2019;20: 681–697. doi:10.1038/s41580-019-0163-x

95. Bonneau R, Baker D. Ab Initio Protein Structure Prediction: Progress and Prospects. Annu Rev Biophys Biomol Struct. 2001;30: 173–189. doi:https://doi.org/10.1146/annurev.biophys.30.1.173

96. Bonneau R, Tsai J, Ruczinski I, Chivian D, Rohl C, Strauss CEM, et al. Rosetta in CASP4: Progress in ab initio protein structure prediction. Proteins Struct Funct Genet. 2001;45: 119–126. doi:10.1002/prot.1170

97. Raman S, Vernon R, Thompson J, Tyka M, Sadreyev R, Pei J, et al. Structure prediction for CASP8 with all-atom refinement using Rosetta. Proteins Struct Funct Bioinforma. 2009;77: 89–99. doi:10.1002/prot.22540

98. Park H, DiMaio F, Baker D. CASP11 refinement experiments with ROSETTA. Proteins Struct Funct Bioinforma. 2015;84: 314–322. doi:10.1002/prot.24862

99. Kelley LA, Sternberg MJE. Protein structure prediction on the web: A case study using the phyre server. Nat Protoc. 2009;4: 363–373. doi:10.1038/nprot.2009.2

100. Yang J, Zhang Y. I-TASSER server: New development for protein structure and function predictions. Nucleic Acids Res. 2015;43: W174–W181. doi:10.1093/nar/gkv342

101. Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J. Critical assessment of methods of protein structure prediction (CASP)—Round XIII. Proteins Struct Funct Bioinforma. 2019;87: 1011–1020. doi:10.1002/prot.25823

102. Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J. Critical assessment of methods of protein structure prediction (CASP)—Round XIV. Proteins Struct Funct Bioinforma. 2021;89: 1607–1617. doi:10.1002/prot.26237

103. Jayaram B, Dhingra P, Lakhani B, Shekhar S. Bhageerath - Targeting the near impossible: Pushing the frontiers of atomic models for protein tertiary structure prediction. J Chem Sci. 2012;124: 83–91. doi:10.1007/s12039-011-0189-x

104. Jayaram B, Dhingra P, Mishra A, Kaushik R, Mukherjee G, Singh A, et al. Bhageerath-H: A homology/ab initio hybrid server for predicting tertiary structures of monomeric soluble proteins. BMC Bioinformatics. 2014;15: 1–12. doi:10.1186/1471-2105-15-S16-S7

105. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596: 583–589. doi:10.1038/s41586-021-03819-2

106. Gao M, Nakajima An D, Parks JM, Skolnick J. AF2Complex predicts direct physical interactions in multimeric proteins with deep learning. Nat Commun. 2022;13. doi:10.1038/s41467-022-29394-2

107. Moult J, Pedersen JT, Judson R, Fidelis K. A large-scale experiment to assess protein structure prediction methods. Proteins Struct Funct Bioinforma. 1995;23: ii–iv. doi:10.1002/prot.340230303

108. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction (CASP) - Round x. Proteins Struct Funct Bioinforma. 2014;82: 1–6. doi:10.1002/prot.24452

109. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction: Progress and new directions in round XI. Proteins Struct Funct Bioinforma. 2016;84: 4–14. doi:10.1002/prot.25064

110. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction (CASP)—Round XII. Proteins-Structure Funct Bioinforma. 2018;86: 7–15. doi:https://doi.org/10.1002/prot.25415

111. Moult J, Hubbard T, Bryant SH, Fidelis K, Pedersen JT. Critical assessment of methods of protein structure prediction (CASP): Round II. Proteins Struct Funct Bioinforma. 1997;29: 2–6. doi:https://doi.org/10.1002/(SICI)1097-0134(1997)1+<2::AID-PROT2>3.0.CO;2-T

112. Moult J, Hubbard T, Fidelis K, Pedersen JT. Critical assessment of methods of protein structure prediction (CASP): Round III. Proteins-Structure Funct Bioinforma. 1999;37: 2–6. doi:https://doi.org/10.1002/(SICI)1097-0134(1999)37:3+<2::AID-PROT2>3.0.CO;2-2

113. Moult J, Fidelis K, Zemla A, Hubbard T. Critical assessment of methods of protein structure prediction (CASP): Round IV. Proteins Struct Funct Bioinforma. 2001;45: 2–7. doi:10.1002/prot.10054

114. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction (CASP) - Round V. Proteins Struct Funct Bioinforma. 2003;53: 334–339. doi:https://doi.org/10.1002/prot.10556

115. Moult J, Fidelis K, Rost B, Hubbard T, Tramontano A. Critical assessment of methods of protein structure prediction (CASP) - Round 6. Proteins Struct Funct Bioinforma. 2005;61: 3–7. doi:10.1002/prot.20716

116. Moult J, Fidelis K, Kryshtafovych A, Rost B, Hubbard T, Tramontano A. Critical assessment of methods of protein structure prediction—Round VII. Proteins Struct Funct Bioinforma. 2007;69: 3–9. doi:https://doi.org/10.1002/prot.21767

117. Moult J, Fidelis K, Kryshtafovych A, Rost B, Tramontano A. Critical assessment of

methods of protein structure prediction-Round VIII. Proteins Struct Funct Bioinforma. 2009;77: 1–4. doi:10.1002/prot.22589

118. Moult J, Fidelis K, Kryshtafovych A, Tramontano A. Critical assessment of methods of protein structure prediction (CASP)-round IX. Proteins Struct Funct Bioinforma. 2011;79: 1–5. doi:10.1002/prot.23200

119. Webb B, Sali A. Comparative protein structure modeling using MODELLER. Curr Protoc Bioinforma. 2016;54: 5.6.1-5.6.37. doi:10.1002/cpbi.3

120. Pearson WR. Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. Genomics. 1991;11: 635–650. doi:https://doi.org/10.1016/0888-7543(91)90071-L

121. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215: 403–410. doi:10.1016/S0022-2836(05)80360-2

122. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25: 3389–3402. doi:10.1093/nar/25.17.3389

123. Eddy SR. Profile hidden Markov models. Bioinformatics. 1998;14: 755–763. doi:10.1093/bioinformatics/14.9.755

124. Rost B. Twilight zone of protein sequence alignments. Protein Eng Des Sel. 1999;12: 85–94. doi:10.1093/protein/12.2.85

125. Karplus K, Barrett C, Hughey R. Hidden Markov models for detecting remote protein homologies. Bioinformatics. 1998;14: 846–856. doi:10.1093/bioinformatics/14.10.846

126. Karplus K, Hu B. Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. Bioinformatics. 2001;17: 713–720. doi:10.1093/bioinformatics/17.8.713

127. Kahsay RY, Wang G, Gao G, Liao L, Dunbrack R. Quasi-consensus-based comparison of profile hidden Markov models for protein sequences. Bioinformatics. 2005;21: 2287–2293. doi:10.1093/bioinformatics/bti374

128. Moult J. A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Curr Opin Struct Biol. 2005;15: 285–289. doi:10.1016/j.sbi.2005.05.011

129. Wallner B, Fang H, Ohlson T, Frey-Skött J, Elofsson A. Using evolutionary information for the query and target improves fold recognition. Proteins Struct Funct Bioinforma. 2004;54: 342–350. doi:10.1002/prot.10565

130. Wang G, Jr RLD. Scoring profile-to-profile sequence alignments. Protein Sci. 2004;13: 1612–1626. doi:10.1110/ps.03601504

131. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol. 1999;292: 195–202. doi:10.1006/jmbi.1999.3091

132. Kabsch W, Sander C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 1983;22: 200–203. doi:doi.org/10.1002/bip.360221211

133. Söding J. Protein homology detection by HMM-HMM comparison. Bioinformatics. 2005;21: 951–960. doi:10.1093/bioinformatics/bti125

134. Chenna R, Sugawara H, Koike T, Lopez R, Gibson TJ, Higgins DG, et al. Multiple sequence alignment with the Clustal series of programs. Nucleic Acids Res. 2003;31: 3497–3500. doi:10.1093/nar/gkg500

135. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004;32: 1792–1797. doi:10.1093/nar/gkh340

136. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol. 2013;30: 772–780. doi:10.1093/molbev/mst010

137. Notredame C, Higgins DG, Heringa J. T-coffee: a novel method for fast and accurate multiple sequence alignment. J Mol Biol. 2000;302: 205–217. doi:10.1006/jmbi.2000.4042

138. Pei J, Kim BH, Grishin N V. PROMALS3D: a tool for multiple protein sequence and structure alignments. Nucleic Acids Res. 2008;36: 2295–2300. doi:10.1093/nar/gkn072

139. Gregory TR. Understanding Evolutionary Trees. Evol Educ Outreach. 2008;1: 121–137. doi:10.1007/s12052-008-0035-x

140. Blundell TL, Sibanda BL, Sternberg MJE, Thornton JM. Knowledge-based prediction of protein structures and the design of novel molecules. Nature. 1987;326: 347–352. doi:https://doi.org/10.1038/326347a0

141. Greer J. Comparative modeling methods: Application to the family of the mammalian serine proteases. Proteins Struct Funct Bioinforma. 1990;7: 317–334. doi:10.1002/prot.340070404

142. Jones TA, Thirup S. Using known substructures in protein model building and crystallography. EMBO J. 1986;5: 819–822. doi:10.1002/j.1460-2075.1986.tb04287.x

143. Claessens M, Van Cutsem E, Lasters I, Wodak S. Modelling the polypeptide backbone with “spare parts” from known protein structures. Protein Eng Des Sel. 1989;2: 335–345. doi:10.1093/protein/2.5.335

144. Levitt M. Accurate modeling of protein conformation by automatic segment matching. J Mol Biol. 1992;226: 507–533. doi:10.1016/0022-2836(92)90964-L

145. Havel TF, Snow ME. A new method for building protein conformations from sequence alignments with homologues of known structure. J Mol Biol. 1991;217: 1–7. doi:10.1016/0022-2836(91)90603-4

146. Srinivasan S, March CJ, Sudarsanam S. An automated method for modeling proteins on known templates using distance geometry. Protein Sci. 1993;2: 277–289. doi:10.1002/pro.5560020216

147. Šali A, Blundell TL. Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol. 1993;234: 779–815. doi:10.1006/jmbi.1993.1626

148. Aszódi A, Taylor WR. Homology modelling by distance geometry. Fold Des. 1996;1: 325–334. doi:10.1016/S1359-0278(96)00048-X

149. Sutcliffe MJ, Haneef I, Carney D, Blundell TL. Knowledge based modelling of homologous proteins, part I: three-dimensional frameworks derived from the

simultaneous superposition of multiple structures. Protein Eng Des Sel. 1987;1: 377–384. doi:10.1093/protein/1.5.377

150. Petrey D, Xiang Z, Tang CL, Xie L, Gimpelev M, Mitros T, et al. Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. Proteins Struct Funct Genet. 2003;53: 430–435. doi:10.1002/prot.10550

151. Dalton JAR, Jackson RM. An evaluation of automated homology modelling methods at low target-template sequence similarity. Bioinformatics. 2007;23: 1901–1908. doi:10.1093/bioinformatics/btm262

152. Kabsch W. A solution for the best rotation to relate two sets of vectors. Acta Crystallogr Sect A. 1976;32: 922–923.

153. Adam Z, Venclovas Č, Moult J, Fidelis K. Processing and Analysis of CASP3 Protein Structure Predictions. PROTEINS Struct Funct Genet. 1999;37: 22–29. doi:10.1002/(sici)1097-0134(1999)37:3+<22::aid-prot5>3.0.co;2-w

154. Reva BA, Finkelstein A V., Skolnick J. What is the probability of a chance prediction of a protein structure with an rmsd of 6 Å? Fold Des. 1998;3: 141–147. doi:10.1016/S1359-0278(98)00019-4

155. Melo F, Sánchez R, Sali A. Statistical potentials for fold assessment. Protein Sci. 2002;11: 430–448. doi:10.1002/pro.110430

156. Dehouck Y, Gilis D, Rooman M. A new generation of statistical potentials for proteins. Biophys J. 2006;90: 4010–4017. doi:10.1529/biophysj.105.079434

157. Huang N, Jacobson MP. Physics-based methods for studying protein-ligand interactions. Curr Opin Drug Discov Devel. 2007;10: 325–331.

158. Shen M, Sali A. Statistical potential for assessment and prediction of protein structures. Protein Sci. 2006;15: 2507–2524. doi:10.1110/ps.062416606

159. Wiederstein M, Sippl MJ. ProSA-web: interactive web service for the recognition of errors in three-dimensional structures of proteins. Nucleic Acids Res. 2007;35: 407–410. doi:10.1093/nar/gkm290

160. Lazaridis T, Karplus M. Discrimination of the native from misfolded protein models with an energy function including implicit solvation. J Mol Biol. 1999;288: 477–487. doi:10.1006/jmbi.1999.2685

161. Higgins DG, Sharp PM. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. Gene. 1988;73: 237–244. doi:10.1016/0378-1119(88)90330-7

162. Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 1994;22: 4673–4680. doi:https://doi.org/10.1093/nar/22.22.4673

163. Saitou N, Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol. 1987;4: 406–425. doi:10.1093/oxfordjournals.molbev.a040454

164. Larkin MA, Blackshields G, Brown NP, Chenna R, Mcgettigan PA, McWilliam H, et al. Clustal W and Clustal X version 2.0. Bioinformatics. 2007;23: 2947–2948. doi:10.1093/bioinformatics/btm404

165. Blackshields G, Sievers F, Shi W, Wilm A, Higgins DG. Sequence embedding for fast construction of guide trees for multiple sequence alignment. Algorithms Mol Biol. 2010;5. doi:10.1186/1748-7188-5-21

166. Söding J. Protein homology detection by HMM-HMM comparison. Bioinformatics. 2005;21: 951–960. doi:10.1093/bioinformatics/bti125

167. Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li W, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol Syst Biol. 2011;7. doi:10.1038/msb.2011.75

168. Edgar RC. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics. 2004;5. doi:10.1186/1471-2105-5-113

169. Katoh K, Misawa K, Kuma KI, Miyata T. MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Res. 2002;30: 3059–3066. doi:10.1093/nar/gkf436

170. Katoh K, Toh H. PartTree: an algorithm to build an approximate tree from a large number of unaligned sequences. Bioinformatics. 2007;23: 372–374. doi:10.1093/bioinformatics/btl592

171. Katoh K, Toh H. Recent developments in the MAFFT multiple sequence alignment program. Brief Bioinform. 2008;9: 286–298. doi:10.1093/bib/bbn013

172. Rozewicki J, Li S, Amada KM, Standley DM, Katoh K. MAFFT-DASH: integrated protein sequence and structural alignment. Nucleic Acids Res. 2019;47: W5–W10. doi:10.1093/nar/gkz342

173. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol. 1970;48: 443–453. doi:10.1016/0022-2836(70)90057-4

174. Huang X, Miller W. A time-efficient, linear-space local similarity algorithm. Adv Appl Math. 1991;12: 337–357. doi:10.1016/0196-8858(91)90017-D

175. O'Sullivan O, Suhre K, Abergel C, Higgins DG, Notredame C. 3DCoffee: Combining protein sequences and structures within multiple sequence alignments. J Mol Biol. 2004;340: 385–395. doi:10.1016/j.jmb.2004.04.058

176. Armougom F, Moretti S, Poirot O, Audic S, Dumas P, Schaeli B, et al. Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee. Nucleic Acids Res. 2006;34: W604–W608. doi:10.1093/nar/gkl092

177. Pei J, Grishin N V. PROMALS: towards accurate multiple sequence alignments of distantly related proteins. Bioinformatics. 2007;23: 802–808. doi:10.1093/bioinformatics/btm017

178. MacArthur MW, Laskowski RA, Thornton JM. Knowledge-based validation of protein structure coordinates derived by X-ray crystallography and NMR spectroscopy. Curr Opin Struct Biol. 1994;4: 731–737. doi:10.1016/S0959-440X(94)90172-4

179. Laskowsk RA, MacArthurt MW, Thornton JM. Validation of protein models derived from experiment. Curr Opin Struct Biol. 1998;8: 631–639. doi:https://doi.org/10.1016/S0959-440X(98)80156-5

180. Lüthy R, Bowie JU, Eisenberg D. Assessment of protein models with three-dimensional profiles. Nature. 1992;356: 83–85. doi:10.1038/356083a0

181. DAVID E, ROLAND L, JAMES U. B. Verify3D: Assessment of protein models with three-dimensional profiles. Methods Enzymol. 1997;277: 396–404. doi:10.1016/s0076-6879(97)77022-8

182. Sippl MJ. Calculation of conformational ensembles from potentials of mena force: An approach to the knowledge-based prediction of local structures in globular proteins. J Mol Biol. 1990;213: 859–883. doi:10.1016/S0022-2836(05)80269-4

183. Benkert P, Künzli M, Schwede T. QMEAN server for protein model quality estimation. Nucleic Acids Res. 2009;37: 510–514. doi:10.1093/nar/gkp322

184. Benkert P, Tosatto SCE, Schomburg D. QMEAN: A comprehensive scoring function for model quality assessment. Proteins Struct Funct Genet. 2008;71: 261–277. doi:10.1002/prot.21715

185. Benkert P, Tosatto SCE, Schwede T. Global and local model quality estimation at CASP8 using the scoring functions QMEAN and QMEANclust. Proteins Struct Funct Bioinforma. 2009;77: 173–180. doi:10.1002/prot.22532

186. Benkert P, Biasini M, Schwede T. Toward the estimation of the absolute quality of individual protein structure models. Bioinformatics. 2011;27: 343–350. doi:10.1093/bioinformatics/btq662

187. Eramian D, Shen M, Devos D, Melo F, Sali A, Marti-Renom MA. A composite score for predicting errors in protein structure models. Protein Sci. 2006;15: 1653–1666. doi:10.1110/ps.062095806

188. Laskowski RA, MacArthur MW, Moss DS, Thornton JM. PROCHECK: a program to check the stereochemical quality of protein structures. J Appl Crystallogr. 1993;26: 283–291. doi:10.1107/s0021889892009944

189. Wu S, Skolnick J, Zhang Y. Ab initio modeling of small proteins by iterative TASSER simulations. BMC Biol. 2007;5. doi:10.1186/1741-7007-5-17

190. Zhang Y, Skolnick J. Automated structure prediction of weakly homologous proteins on a genomic scale. Proc Natl Acad Sci U S A. 2004;101: 7594–7599. doi:10.1073/pnas.0305695101

191. Zhang Y. Template-based modeling and free modeling by I-TASSER in CASP7. Proteins Struct Funct Genet. 2007;69: 108–117. doi:10.1002/prot.21702

192. Yang J, Zhang W, He B, Walker SE, Zhang H, Govindarajoo B, et al. Template-based protein structure prediction in CASP11 and retrospect of I-TASSER in the last decade. Proteins Struct Funct Bioinforma. 2016;84: 233–246. doi:10.1002/prot.24918

193. Wu S, Zhang Y. LOMETS: A local meta-threading-server for protein structure prediction. Nucleic Acids Res. 2007;35: 3375–3382. doi:10.1093/nar/gkm251

194. He B, Mortuza SM, Wang Y, Shen H Bin, Zhang Y. NeBcon: protein contact map

prediction using neural network training coupled with naïve Bayes classifiers. Bioinformatics. 2017;33: 2296–2306. doi:10.1093/bioinformatics/btx164

195. Zheng W, Li Y, Zhang C, Pearce R, Mortuza SM, Zhang Y. Deep-learning contact-map guided protein structure prediction in CASP13. Proteins Struct Funct Bioinforma. 2019;87: 1149–1164. doi:10.1002/prot.25792

196. Biasini M, Bienert S, Waterhouse A, Arnold K, Studer G, Schmidt T, et al. SWISS-MODEL: modelling protein tertiary and quaternary structure using evolutionary information. Nucleic Acids Res. 2014;42: W252–W258. doi:10.1093/nar/gku340

197. Guex N, Peitsch MC. SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. Electrophoresis. 1997;18: 2714–2723. doi:10.1002/elps.1150181505

198. Haas J, Roth S, Arnold K, Kiefer F, Schmidt T, Bordoli L, et al. The Protein Model Portal - A comprehensive resource for protein structure and model information. Database. 2013;2013. doi:10.1093/database/bat031

199. Simons KT, Bonneau R, Ruczinski I, Baker D. Ab initio protein structure prediction of CASP III targets using ROSETTA. Proteins Struct Funct Bioinforma. 1999;37: 171–176. doi:https://doi.org/10.1002/(SICI)1097-0134(1999)37:3+<171::AID-PROT21>3.0.CO;2-Z

200. Sorenson JM, Head-Gordon T. Matching simulation and experiment: a new simplified model for simulating protein folding. J Comput Biol. 2000;7: 469–481. doi:10.1089/106652700750050899

201. A.Rohl C, Strauss CEM, Misura KMS, Baker D. Protein structure prediction using Rosetta. Methods Enzymol. 2004;383: 66–93. doi:https://doi.org/10.1016/S0076-6879(04)83004-0

202. Kinch LN, Li W, Monastyrskyy B, Kryshtafovych A, Grishin N V. Evaluation of free modeling targets in CASP11 and ROLL. Proteins Struct Funct Bioinforma. 2015;84: 51–66. doi:10.1002/prot.24973

203. Ming W, Laven J, Krupers M, Thüne PC, Niemantsverdriet JW, Brongersma HH, et al. Structural Symmetry and Protein Function. Annu Rev Biophys Biomol Struct. 2000;29: 105–153.

204. Marianayagam NJ, Sunde M, Matthews JM. The power of two: Protein dimerization in biology. Trends Biochem Sci. 2004;29: 618–625. doi:10.1016/j.tibs.2004.09.006

205. Preston M. System Development Life Cycle Guide. 2020 [cited 12 Feb 2021]. Available: https://www.clouddefense.ai/blog/system-development-life-cycle

206. Tutorials Point Inc. SDLC - Overview. 2021. Available: https://www.tutorialspoint.com/sdlc/sdlc_overview.htm

207. Sommerville I. Software Engineering. 9th editio. 2010. Available: https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc

208. Brown DK, Penkler DL, Musyoka TM, Bishop OT. JMS: An open source workflow management system and web-based cluster front-end for high performance computing. PLoS One. 2015;10: 0134273. doi:10.1371/journal.pone.0134273

209. Django. The web framework for perfectionists with deadlines. 2021. Available: https://www.djangoproject.com/

210. Django REST Framework. 2021. Available: http://www.django-rest-framework.org

211. Holovaty A, Kaplan-Moss J. The Definitive Guide to Django: Web Development Done Right. second edi. Apress; 2009. Available: https://books.google.co.ug/books?id=h2tR8p-4a9QC

212. Severance C. Roy T. Fielding: Understanding the REST Style. Computer (Long Beach Calif). 2015;48: 7–9. doi:10.1109/MC.2015.170

213. Fielding RT, Gettys J, Mogul JC, Nielsen HF, Masinter L, Leach PJ, et al. RFC 2616: Hypertext Transfer Protocol - HTTP/1.1. In: The Internet Society. 1999 pp. 1–155. doi:http://www.ietf.org/rfc/rfc2616.txt

214. Oracle Corporation and/or its affiliates. MySQL. 2021. Available: https://www.mysql.com/

215. Knockout.js. 2021. Available: https://knockoutjs.com/

216. Yuan S, Chan HCS, Hu Z. Using PyMOL as a platform for computational drug design. Wiley Interdiscip Rev Comput Mol Sci. 2017;7: 1–10. doi:10.1002/wcms.1298

217. Biasini M. PV - WebGL-based protein viewer. Zenodo; doi:http://doi.org/10.5281/zenodo.20980

218. Yachdav G, Wilzbach S, Rauscher B, Sheridan R, Sillitoe I, Procter J, et al. MSAViewer: Interactive JavaScript visualization of multiple sequence alignments. Bioinformatics. 2016;32: 3501–3503. doi:10.1093/bioinformatics/btw474

219. Rose AS, Hildebrand PW. NGL Viewer: a web application for molecular visualization. Nucleic Acids Res. 2015;43: W576–W579. doi:10.1093/nar/gkv402

220. Mariani V, Biasini M, Barbato A, Schwede T. lDDT: A local superposition-free score for comparing protein structures and models using distance difference tests. Bioinformatics. 2013;29: 2722–2728. doi:10.1093/bioinformatics/btt473

221. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. Proteins Struct Funct Genet. 2004;57: 702–710. doi:10.1002/prot.20264

222. Šali A. MODELLER: a program for protein structure modeling. 2021. Available: https://salilab.org/modeller/manual/

223. Alexander PA, He Y, Chen Y, Orban J, Bryan PN. The design and characterization of two proteins with 88 % sequence identity but different structure and function. Proc Natl Acad Sci U S A. 2007;104: 11963–11968.

224. Bishop ÖT, Kroon M. Study of protein complexes via homology modeling, applied to cysteine proteases and their protein inhibitors. J Mol Model. 2011;17: 3163–3172. doi:10.1007/s00894-011-0990-y

225. Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. EMBO J. 1986;5: 823–826. doi:10.1002/j.1460-2075.1986.tb04288.x

226. Zemla A. LGA: a method for finding 3D similarities in protein structures. Nucleic

Acids Res. 2003;31: 3370–3374. doi:10.1093/nar/gkg571

227. Xu J, Zhang Y. How significant is a protein structure similarity with TM-score = 0.5? Bioinformatics. 2010;26: 889–895. doi:10.1093/bioinformatics/btq066

228. Baker D, Sali A. Protein structure prediction and structural genomics. Science (80-). 2001;294: 93–96. doi:10.1126/science.1065659

229. Zhang Y. Protein structure prediction : when is it useful ? Curr Opin Struct Biol. 2009;19: 145–155. doi:10.1016/j.sbi.2009.02.005

230. Kiefer F, Arnold K, Künzli M, Bordoli L, Schwede T. The SWISS-MODEL Repository and associated resources. Nucleic Acids Res. 2009;37: D387–D392. doi:10.1093/nar/gkn750

231. Lewis TE, Sillitoe I, Andreeva A, Blundell TL, Buchan DWA, Chothia C, et al. Genome3D: exploiting structure to help users understand their sequences. Nucleic Acids Res. 2015;43: D382–D386. doi:10.1093/nar/gku973

232. Pieper U, Webb BM, Dong GQ, Schneidman-Duhovny D, Fan H, Kim SJ, et al. ModBase, a database of annotated comparative protein structure models and associated resources. Nucleic Acids Res. 2014;42: D336–D346. doi:10.1093/nar/gkt1144

233. Isberg V, Mordalski S, Munk C, Rataj K, Harpsøe K, Hauser AS, et al. GPCRdb: an information system for G protein-coupled receptors. Nucleic Acids Res. 2016;44: D356–D364. doi:10.1093/nar/gkv1178

234. Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, et al. The Universal Protein Resource (UniProt). Nucleic Acids Res. 2005;33: D154–D159. doi:10.1093/nar/gki070

235. Pegg SC, Brown SD, Ojha S, Seffernick J, Meng EC, Morris JH, et al. Leveraging Enzyme Structure - Function Relationships for Functional Inference and Experimental Design : The Structure - Function Linkage Database. Biochemistry. 2006;45: 2545–2555. doi:10.1021/bi0521011

236. Zimmermann L, Stephens A, Nam SZ, Rau D, Kübler J, Lozajic M, et al. A Completely Reimplemented MPI Bioinformatics Toolkit with a New HHpred Server at its Core. J Mol Biol. 2018;430: 2237–2243. doi:10.1016/j.jmb.2017.12.007

237. Melo F, Devos D, Depiereux E, Feytmans E. ANOLEA: a www server to assess protein structures. Proc Int Conf Intell Syst Mol Biol. 1997;5: 187–190.

238. Murkin AS, Manning KA, Kholodar SA. Mechanism and inhibition of 1-deoxy-d-xylulose-5-phosphate reductoisomerase. Bioorg Chem. 2014;57: 171–185. doi:10.1016/j.bioorg.2014.06.001

239. Umeda T, Tanaka N, Kusakabe Y, Nakanishi M, Kitade Y, Nakamura KT. Molecular basis of fosmidomycin's action on the human malaria parasite Plasmodium falciparum. Sci Rep. 2011;1. doi:10.1038/srep00009

240. Takenoya M, Ohtaki A, Noguchi K, Endo K, Sasaki Y, Ohsawa K, et al. Crystal structure of 1-deoxy-d-xylulose 5-phosphate reductoisomerase from the hyperthermophile Thermotoga maritima for insights into the coordination of conformational changes and an inhibitor binding. J Struct Biol. 2010;170: 532–539. doi:10.1016/j.jsb.2010.03.015

241. Uziela K, Menéndez Hurtado D, Shu N, Wallner B, Elofsson A. ProQ3D: improved model quality assessments using deep learning. Bioinformatics. 2017;33: 1578–1580. doi:10.1093/bioinformatics/btw819

242. Zhou H, Zhou Y. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Sci. 2002;11: 2714–2726.

243. Linares-Pasten J, Andersson M, Karlsson E. Thermostable Glycoside Hydrolases in Biorefinery Technologies. Curr Biotechnol. 2014;3: 26–44. doi:10.2174/2211550113026660041

244. Bourne Y, Henrissat B. Glycoside hydrolases and glycosyltransferases: families and functional modules. Curr Opin Struct Biol. 2001;11: 593–600. doi:10.1016/S0959-440X(00)00253-0

245. Lombard V, Golaconda Ramulu H, Drula E, Coutinho PM, Henrissat B. The carbohydrate-active enzymes database (CAZy) in 2013. Nucleic Acids Res. 2014;42: D490–D495. doi:10.1093/nar/gkt1178

246. Koshland DE. Stereochemistry and the mechanism of enzymatic reactions. Biol Rev. 1953;28: 416–436. doi:10.1111/j.1469-185X.1953.tb01386.x

247. Michalska K, Tan K, Li H, Hatzos-Skintges C, Bearden J, Babnigg G, et al. GH1-family 6-P-β-glucosidases from human microbiome lactic acid bacteria. Acta Crystallogr Sect D Biol Crystallogr. 2013;69: 451–463. doi:10.1107/S0907444912049608

248. Fun A, Wensing AMJ, Verheyen J, Nijhuis M. Human Immunodeficiency Virus gag and protease: Partners in resistance. Retrovirology. 2012;9: 63. doi:10.1186/1742-4690-9-63

249. Tóth G, Borics A. Flap opening mechanism of HIV-1 protease. J Mol Graph Model. 2006;24: 465–474. doi:10.1016/j.jmgm.2005.08.008

250. Bertacine Dias M V., Santos JC, Libreros-Zúñiga GA, Ribeiro JA, Chavez-Pacheco SM. Folate biosynthesis pathway: mechanisms and insights into drug design for infectious diseases. Future Med Chem. 2018;10. doi:10.4155/fmc-2017-0168

251. Nzila A, Ward SA, Marsh K, Sims PFG, Hyde JE. Comparative folate metabolism in humans and malaria parasites (part I): Pointers for malaria treatment from cancer chemotherapy. Trends in Parasitology. 2005. doi:10.1016/j.pt.2005.04.002

252. Colloc'h N, Poupon A, Mornon J-P. Sequence and structural features of the T-fold, an original tunnelling building unit. Proteins Struct Funct Genet. 2000;39: 142–154. doi:10.1002/(SICI)1097-0134(20000501)39:2<142::AID-PROT4>3.0.CO;2-X

253. Rajendran V, Kalita P, Shukla H, Kumar A, Tripathi T. Aminoacyl-tRNA synthetases: Structure, function, and drug discovery. Int J Biol Macromol. 2018;111: 400–414. doi:10.1016/j.ijbiomac.2017.12.157

254. Bhatt TK, Kapil C, Khan S, Jairajpuri MA, Sharma V, Santoni D, et al. A genomic glimpse of aminoacyl-tRNA synthetases in malaria parasite Plasmodium falciparum. BMC Genomics. 2009;10: 644. doi:10.1186/1471-2164-10-644

255. Nyamai DW, Tastan Bishop Ö. Aminoacyl tRNA synthetases as malarial drug targets:

a comparative bioinformatics study. Malar J. 2019;18: 1–27. doi:10.1186/s12936-019-2665-6

256. Perona JJ, Hadd A. Structural diversity and protein engineering of the aminoacyl-tRNA Synthetases. Biochemistry. 2012;51: 8705–8729. doi:10.1021/bi301180x

257. Jackson KE, Habib S, Frugier M, Hoen R, Khan S, Pham JS, et al. Protein translation in Plasmodium parasites. Trends in Parasitology. 2011. pp. 467–476. doi:10.1016/j.pt.2011.05.005

258. Jain V, Yogavel M, Sharma A. Dimerization of Arginyl-tRNA Synthetase by Free Heme Drives Its Inactivation in Plasmodium falciparum. Structure. 2016;24: 1476–1487. doi:10.1016/j.str.2016.06.018

259. Ofir-Birin Y, Fang P, Bennett SP, Zhang HM, Wang J, Rachmin I, et al. Structural Switch of Lysyl-tRNA Synthetase between Translation and Transcription. Mol Cell. 2013;49: 30–42. doi:10.1016/j.molcel.2012.10.010

260. Yang XL, Guo M, Kapoor M, Ewalt KL, Otero FJ, Skene RJ, et al. Functional and Crystal Structure Analysis of Active Site Adaptations of a Potent Anti-Angiogenic Human tRNA Synthetase. Structure. 2007;15: 793–805. doi:10.1016/j.str.2007.05.009

261. Koh CY, Kim JE, Napoli AJ, Verlinde CLMJ, Fan E, Buckner FS, et al. Crystal structures of Plasmodium falciparum cytosolic tryptophanyl-tRNA synthetase and its potential as a target for structure-guided drug design. Mol Biochem Parasitol. 2013;189: 26–32. doi:10.1016/j.molbiopara.2013.04.007

262. Hewitt SN, Dranow DM, Horst BG, Abendroth JA, Forte B, Hallyburton I, et al. Biochemical and Structural Characterization of Selective Allosteric Inhibitors of the *Plasmodium falciparum* Drug Target, Prolyl-tRNA-synthetase. ACS Infect Dis. 2017;3: 34–44. doi:10.1021/acsinfecdis.6b00078

263. Zhou H, Sun L, Yang XL, Schimmel P. ATP-directed capture of bioactive herbal-based medicine on human tRNA synthetase. Nature. 2013;494: 121–124. doi:10.1038/nature11774

264. Ojo KK, Ranade RM, Zhang Z, Dranow DM, Myers JB, Choi R, et al. Brucella melitensis Methionyl-tRNA-Synthetase (MetRS), a potential drug target for brucellosis. PLoS One. 2016;11. doi:10.1371/journal.pone.0160350

265. Fidler DR, Murphy SE, Courtis K, Antonoudiou P, El-Tohamy R, Ient J, et al. Using HHsearch to tackle proteins of unknown function: a pilot study with PH domains. Traffic. 2016;17: 1214–1226. doi:10.1111/tra.12432

266. Nikkola M, Lindqvist Y, Schneider G. Refined Structure of Transketolase from Saccharomyces cerevisiae at 2·0 Å Resolution. J Mol Biol. 1994. doi:10.1006/jmbi.1994.1299

267. Amusengeri A, Bishop ÖT. Discorhabdin N, a South African natural compound, for Hsp72 and Hsc70 allosteric modulation: Combined study of molecular modeling and dynamic residue network analysis. Molecules. 2019;24: 188. doi:10.3390/molecules24010188

268. Corrêa TLR, dos Santos LV, Pereira GAG. AA9 and AA10: from enigmatic to essential enzymes. Appl Microbiol Biotechnol. 2016;100: 9–16. doi:10.1007/s00253-015-7040-0

269. Hemsworth GR, Henrissat B, Davies GJ, Walton PH. Discovery and characterization of a new family of lytic polysaccharide monooxygenases. Nat Chem Biol. 2014;10: 122–126. doi:10.1038/nchembio.1417

270. Moses V, Tastan Bishop Ö, Lobb KA. The evaluation and validation of copper (II) force field parameters of the Auxiliary Activity family 9 enzymes. Chem Phys Lett. 2017;678: 91–97. doi:10.1016/j.cplett.2017.04.022

271. Deu E. Proteases as antimalarial targets: strategies for genetic, chemical, and therapeutic validation. FEBS J. 2017;284: 2604–2628. doi:10.1111/febs.14130

272. Goldberg DE, Winzeler EA, Istvan ES, Gluzman I, Dharia N V., Bopp SE. Validation of isoleucine utilization targets in Plasmodium falciparum. Proc Natl Acad Sci. 2011;108: 1627–1632. doi:10.1073/pnas.1011560108

273. Gluzman IY, Liu J, Goldberg DE, Gross J, Istvan ES. Plasmodium falciparum ensures its amino acid supply with multiple acquisition pathways and redundant proteolytic enzyme systems. Proc Natl Acad Sci. 2006;103: 8840–8845. doi:10.1073/pnas.0601876103

274. Singh A, Sijwali PS, Rosenthal PJ, Gut J, Shenai BR. Expression and characterization of the Plasmodium falciparum haemoglobinase falcipain-3. Biochem J. 2015;360: 481–489. doi:10.1042/bj3600481

275. Sijwali PS, Koo J, Singh N, Rosenthal PJ. Gene disruptions demonstrate independent roles for the four falcipain cysteine proteases of Plasmodium falciparum. Mol Biochem Parasitol. 2006;150: 96–106. doi:10.1016/j.molbiopara.2006.06.013

276. Pandey KC, Sijwali PS, Craik CS, Shenai BR, Choe Y, Singh A, et al. Identification and biochemical characterization of vivapains, cysteine proteases of the malaria parasite Plasmodium vivax. Biochem J. 2004;378: 529–538. doi:10.1042/bj20031487

277. Prasad R, Atul P, Soni A, Puri SK, Sijwali PS. Expression, Characterization, and Cellular Localization of Knowpains, Papain-Like Cysteine Proteases of the Plasmodium knowlesi Malaria Parasite. Snounou G, editor. PLoS One. 2012;7: e51619. doi:10.1371/journal.pone.0051619

278. Vaughan AM, Pei Y, Kappe SHI, Lindner SE, Torii M, Miller JL. Plasmodium yoelii inhibitor of cysteine proteases is exported to exomembrane structures and interacts with yoelipain-2 during asexual blood-stage development . Cell Microbiol. 2013;15: 1508–1526. doi:10.1111/cmi.12124

279. Martins TM, Domingos A, Gonçalves LMD, Silveira H, Rosário V do, Caldeira RL, et al. Plasmodium chabaudi: Expression of active recombinant chabaupain-1 and localization studies in Anopheles sp. Exp Parasitol. 2009;122: 97–105. doi:10.1016/j.exppara.2009.03.003

280. Rosenthal PJ. Falcipains and other cysteine proteases of malaria parasites. Adv Exp Med Biol. 2011;712: 30–48. doi:10.1007/978-1-4419-8414-2_3

281. Teixeira C, Gomes JRB, Gomes P. Falcipains, Plasmodium falciparum cysteine proteases as key drug targets against malaria. Curr Med Chem. 2011;18: 1555–1572. doi:10.2174/092986711795328328

282. Kerr ID, Lee JH, Pandey KC, Harrison A, Sajid M, Rosenthal PJ, et al. Structures of falcipain-2 and falcipain-3 bound to small molecule inhibitors: implications for

substrate specificity. J Med Chem. 2009;52: 852–857. doi:10.1021/jm8013663; 10.1021/jm8013663

283. Musyoka TM, Kanzi AM, Lobb KA, Tastan Bishop Ö. Analysis of non-peptidic compounds as potential malarial inhibitors against Plasmodial cysteine proteases via integrated virtual screening workflow. J Biomol Struct Dyn. 2016;34: 2084–2101. doi:10.1080/07391102.2015.1108231

284. Pandey KC, Dixit R. Structure-Function of Falcipains: Malarial Cysteine Proteases. J Trop Med. 2012;2012: 1–11. doi:10.1155/2012/345195

285. Musyoka TM, Njuguna JN, Tastan Bishop Ö. Comparing sequence and structure of falcipains and human homologs at prodomain and catalytic active site for malarial peptide based inhibitor design. Malar J. 2019;18: 159. doi:10.1186/s12936-019-2790-2

286. UNAIDS. UNAIDS data 2018. 2018. Available: http://www.unaids.org/sites/default/files/media_asset/unaids-data-2018_en.pdf

287. Bennett DE, Bertagnolio S, Sutherland D, Gilks CF. The World Health Organization's global strategy for prevention and assessment of HIV drug resistance. Antivir Ther. 2008;13: 1–13.

288. Günthard HF, Calvez V, Paredes R, Pillay D, Shafer RW, Wensing AM, et al. Human Immunodeficiency Virus Drug Resistance: 2018 Recommendations of the International Antiviral Society–USA Panel. Clin Infect Dis. 2018; 1–11. doi:10.1093/cid/ciy463

289. Beerenwinkel N, Sing T, Lengauer T, Rahnenführer J, Roomp K, Savenkov I, et al. Computational methods for the design of effective therapies against drug resistant HIV strains. Bioinformatics. 2005;21: 3943–3950. doi:10.1093/bioinformatics/bti654

290. Sheik Amamuddy O, Bishop NT, Tastan Bishop Ö. Improving fold resistance prediction of HIV-1 against protease and reverse transcriptase inhibitors using artificial neural networks. BMC Bioinformatics. 2017;18: 369. doi:10.1186/s12859-017-1782-x

291. Nabatanzi M. Development and evaluation of a web application employing artificial neural networks to facilitate the prediction of antiretroviral drug resistance in patients infected with HIV-1 subtype B. Rhodes University, South Africa. 2018.

292. WHO. Global health sector strategy on HIV, 2016-2021. World Heal Organ. Geneva; 2016. Available: http://www.who.int/hiv/strategy2016-2021/ghss-hiv/en

293. UNAIDS. Confronting inequalities. Geneva; 2021. Available: https://www.unaids.org/sites/default/files/media_asset/2021-global-aids-update_en.pdf

294. Seyler L, Lacor P, Allard SD. Current challenges in the treatment of HIV. Polish Arch Intern Med. 2018;128: 609–616. doi:10.20452/pamw.4357

295. World Health Organization. Guidelines on the public health response to pretreatment HIV drug resistance. WHO Guidel. 2017.

296. Phillips AN, Stover J, Cambiano V, Nakagawa F, Jordan MR, Pillay D, et al. Impact of HIV drug resistance on HIV/AIDS-associated mortality, new infections, and antiretroviral therapy program costs in Sub-Saharan Africa. J Infect Dis. 2017;215: 1362–1365. doi:10.1093/infdis/jix089

297. World Health Organization, United States Centers for Disease Control and Prevention, The Global Fund to Fight AIDS T and M. Hiv drug resistance report 2017. p. 82. Available: https://apps.who.int/iris/bitstream/handle/10665/255896/9789241512831-eng.pdf?sequence=1

298. World Health Organization. Global action plan on HIV drug resistance 2017-2021: 2018 progress report. p. 14. Available: https://apps.who.int/iris/bitstream/handle/10665/255883/9789241512848-eng.pdf?sequence=1

299. Shen C, Yu X, Harrison RW, Weber IT. Automated prediction of HIV drug resistance from genotype data. BMC Bioinformatics. 2016;17: 278. doi:10.1186/s12859-016-1114-6

300. Clavel F. Mechanisms of HIV Drug Resistance: A Primer. PRN Noteb. 2004;9: 3–7.

301. Perrin L, Telenti A. HIV treatment failure: testing for HIV resistance in clinical practice. Science. 1998;280: 1871–1873. doi:10.1126/science.280.5371.1871

302. Tang MW, Liu TF, Shafer RW. The HIVdb system for HIV-1 genotypic resistance interpretation. Intervirology. 2012;55: 98–101. doi:10.1159/000331998

303. Vercauteren J, Beheydt G, Prosperi M, Libin P, Imbrechts S, Camacho R, et al. Clinical Evaluation of Rega 8: An Updated Genotypic Interpretation System That Significantly Predicts HIV-Therapy Response. PLoS One. 2013;8. doi:10.1371/journal.pone.0061436

304. Wagner S, Kurz M, Klimkait T. Algorithm evolution for drug resistance prediction: Comparison of systems for HIV-1 genotyping. Antivir Ther. 2015;20: 661–665. doi:10.3851/IMP2947

305. Beerenwinkel N, Däumer M, Oette M, Korn K, Hoffmann D, Kaiser R, et al. Geno2pheno: Estimating phenotypic drug resistance from HIV-1 genotypes. Nucleic Acids Res. 2003;31: 3850–3855. doi:10.1093/nar/gkg575

306. SIEPEL AC, HALPERN AL, MACKEN C, KORBER BTM. A Computer Program Designed to Screen Rapidly for HIV Type 1 Intersubtype Recombinant Sequences. AIDS Res Hum Retroviruses. 1995;11: 1413–1416. doi:10.1089/aid.1995.11.1413

307. Rozanov M, Plikat U, Chappey C, Kochergin A, Tatusova T. A web-based genotyping resource for viral sequences. Nucleic Acids Res. 2004;32: W654–W659. doi:10.1093/nar/gkh419

308. de Oliveira T, Deforche K, Cassol S, Salminen M, Paraskevis D, Seebregts C, et al. An automated genotyping system for analysis of HIV-1 and other microbial sequences. Bioinformatics. 2005;21: 3797–3800. doi:10.1093/bioinformatics/bti607

309. Struck D, Lawyer G, Ternes AM, Schmit JC, Bercoff DP. COMET: Adaptive context-based modeling for ultrafast HIV-1 subtype identification. Nucleic Acids Res. 2014;42. doi:10.1093/nar/gku739

310. Schultz A-K, Zhang M, Leitner T, Kuiken C, Korber B, Morgenstern B, et al. A jumping profile Hidden Markov Model and applications to recombination sites in HIV and HCV genomes. BMC Bioinformatics. 2006;7: 265. doi:10.1186/1471-2105-7-265

311. Stanford HIVdb. Genotype-Phenotype Datasets. 2014 [cited 28 Mar 2018]. Available:

https://hivdb.stanford.edu/pages/genopheno.dataset.html

312. Haley Hedlin. Genotype-Phenotype Datasets: DRMcv. In: Genotype-Phenotype Datasets: DRMcv. [Internet]. 2014 [cited 10 Jul 2018]. Available: https://hivdb.stanford.edu/download/GenoPhenoDatasets/DRMcv.R

313. Altman DG, Bland JM. Measurement in Medicine : the Analysis of Method Comparison Studiest. J R Stat Soc Ser D (The Stat. 1983;32: 307–317. doi:doi:10.2307/2987937

314. Bland JM, Altman DG. Statistical methods for assessing agreement between two methods of clinical measurement. Lancet. 1986;327: 307–310. doi:doi:10.1016/s0140-6736(86)90837-8

315. NCBI. National Center for Biotechnology Information (NCBI). In: Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information [Internet]. 1988 [cited 6 Nov 2017]. Available: https://www.ncbi.nlm.nih.gov/

316. Rabiner LR. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Readings Speech Recognit. 1990; 267–296. doi:10.1016/B978-0-08-051584-7.50027-9

317. Rabiner LR, Juang BH. An Introduction to Hidden Markov Models. IEEE ASSP Mag. 1986;3 (1): 4–16.

318. Salzberg SL, Searls DB, Kasif S. An introduction to hidden Markov models for biological sequences, by Anders Krogh in: Computational methods in molecular biology. Elsevier; 1998.

319. The MathWorks I. Hidden Markov model posterior state probabilities - MATLAB hmmdecode. [cited 6 Nov 2017]. Available: https://www.mathworks.com/help/stats/hmmdecode.html

320. Fawcett T. An introduction to ROC analysis. Pattern Recognit Lett. 2006;27: 861–874. doi:10.1016/j.patrec.2005.10.010

321. Zou KH, O'Malley AJ, Mauri L. Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. Circulation. 2007;115: 654–657. doi:10.1161/CIRCULATIONAHA.105.594929

322. Liu TF, Shafer RW. Web resources for HIV type 1 genotypic-resistance test interpretation. Clin Infect Dis. 2006;42: 1608–18. doi:10.1086/503914

323. Pond SLK, Posada D, Stawiski E, Chappey C, Poon AFY, Hughes G, et al. An evolutionary model-based algorithm for accurate phylogenetic breakpoint mapping and subtype prediction in HIV-1. PLoS Comput Biol. 2009;5. doi:10.1371/journal.pcbi.1000581

324. Pineda-Peña AC, Faria NR, Imbrechts S, Libin P, Abecasis AB, Deforche K, et al. Automated subtyping of HIV-1 genetic sequences for clinical and surveillance purposes: Performance evaluation of the new REGA version 3 and seven other tools. Infect Genet Evol. 2013;19: 337–348. doi:10.1016/j.meegid.2013.04.032

325. Nagy A. The impact of e-learning. Bruck PA, Karssen Z, Buchholz A, Zerfass A (eds) E-Content. Springer, Berlin, Heidelberg; 2005. pp. 79–96. doi:https://doi.org/10.1007/3-540-26387-X_4

326. National Research Council, Division on Earth and Life Studies, Board on Life Sciences, Committee on Undergraduate Biology Education to Prepare Research Scientists for the 21st Century. BIO 2010: Transforming undergraduate education for future research biologists. Washington, D.C: National Academies Press; 2003.

327. Maloney M, Parker J, LeBlanc M, Woodard CT, Glackin M, Hanrahan M. Bioinformatics and the undergraduate curriculum. CBE-Life Sci Educ. 2010;9: 172–174. doi:10.1187/cbe.10-03-0038

328. Furge LL, Stevens-Truss R, Moore DB, Langeland JA. Vertical and horizontal integration of bioinformatics education. Biochem Mol Biol Educ. 2009;37: 26–36. doi:10.1002/bmb.20249

329. Campbell CE, Nehm RH. A critical analysis of assessment quality in genomics and bioinformatics education research. CBE—Life Sci Educ. 2013;12: 530–541. doi:10.1187/cbe.12-06-0073

330. Honts JE. Evolving strategies for the incorporation of bioinformatics within the undergraduate cell biology curriculum. Cell Biol Educ. 2003;2: 233–247. doi:10.1187/cbe.03-06-0026

331. Brown DK, Özlem TB. HUMA: A platform for the analysis of genetic variation in humans. Hum Mutat. 2017;39: 40–51. doi:doi.org/10.1002/humu.23334