Exploring SARS-CoV-2 Antibody Data through Machine Learning Methods

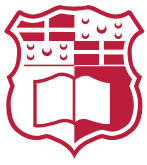
Francesca Chircop

Supervisor: Prof. Jean-Paul Ebejer

Co-Supervisor: Prof. David Saliba

June 2025

Submitted in partial fulfilment of the requirements for the degree of M.Sc Data Science.



L-Università ta' Malta
Faculty of Information & Communication Technology

Abstract

The diversity and binding specificity of antibody repertoires, particularly within the Complementarity Determining Region H3 (CDRH3) region are fundamental to immune protection and therapeutic antibody development. In this work, we integrate hierarchical clustering, motif discovery, sequence based machine learning, and user-friendly graphical interfaces into a bioinformatics framework to: (1) identify and classify SARS-CoV-2 CDRH3 families, (2) extract conserved paratope motifs, (3) predict antibody–antigen binding using both classical and deep learning models, and (4) enable accessible sequence translation and clustering through two standalone Graphical User Interfaces (GUIs). Our analysis is based on sequences from two distinct sources: the public CoV-AbDab database curated by the Oxford Protein Informatics Group (OPIG), and proprietary scFv phage display data obtained via Sanger sequencing at the University of Malta. Through clustering, we identify eight major public clonotypes in both the University of Malta dataset and the OPIG dataset, each with distinct sequence logos. A stacking ensemble comprising logistic regression, random forest, and Extreme Gradient Boosting (XGBoost) attains an Receiver Operating Characteristic Area Under the Curve (ROC-AUC) of 0.71, outperforming each individual model, including traditional machine learning approaches (logistic regression, random forest, XGBoost) and deep learning architectures (Bidirectional Long Short-Term Memory (Bi-LSTM), ProtBERT, Siamese Convolutional Neural Network (CNN)). In usability trials, biomedical students with no coding experience installed our GUIs, performed DNA to protein translation, variable annotation, clustering, and figure export in under five minutes, and contributed feedback that led to a highly intuitive interface. Together, these results provide a reproducible, end to end toolkit for rapid, in-silico antibody repertoire analysis and binding prediction, supporting both computational immunology research and experimental planning.

Acknowledgements

I am deeply grateful to my supervisors, Prof. Jean-Paul Ebejer and Prof. David Saliba, for their guidance, support, and invaluable feedback throughout this project. Their expertise in the field inspired and shaped the direction of this work.

I would also like to thank Mariana Grima, Elisa Chircop, Oriana Mazzitelli and Tymoteusz Piasecki from the University of Malta for generously sharing their Sanger sequencing dataset, testing the graphical interfaces, and patiently explaining domain-specific concepts. Their hands on feedback was instrumental in refining both the clustering analysis and the usability of the software tools.

My sincere appreciation goes to my family for their unwavering encouragement and to my course mates for stimulating discussions and camaraderie during our studies. Without their support, this thesis would not have been possible.

Contents

Abstract i

Acknowledgements ii

Contents v

List of Figures viii

List of Tables ix

List of Abbreviations x

1 Introduction 1

1.1 Motivation 1

1.2 Aims and Objectives 3

1.3 Our Approach 4

1.4 Document Structure 4

2 Background and Literature Review 6

2.1 The COVID-19 Pandemic 6

2.2 What are Antibodies and what is an Antigen? 6

2.2.1 Antibody Variable Domains: Structure and Sequence Representation 7

2.3 Phage Display and Biopanning 8

2.3.1 Biopanning: Iterative Selection of High Affinity Binders 8

2.4 Visualisation Techniques 10

2.4.1 Dendrograms and Frequency Matrices in Sequence Analysis 11

2.4.2 Sequence Logo Visualisation 12

2.5 BLAST and Motif Discovery 13

2.6 Machine Learning Approaches 15

2.6.1 Unsupervised Learning: Clustering Methods . . . 15
2.6.2 Supervised learning Methods . . . 16
2.7 Evaluation Metrics . . . 19
2.7.1 Clustering Evaluation Metrics . . . 20
2.7.2 Binding/Non-Binding Evaluation Metrics . . . 21
2.8 Related Work . . . 22
2.8.1 Clustering Methods for Antibody Repertoires . . . 22
2.8.2 Models for Binder/Non-Binder Classification . . . 24
2.9 Summary . . . 28

3 Methodology **29**
3.1 Datasets . . . 29
3.1.1 OPIG-CovAbDab Dataset . . . 30
3.1.2 University of Malta Sanger Sequencing Data . . . 30
3.2 Data Preprocessing for Clustering Analysis . . . 31
3.2.1 Preprocessing of OPIG CovAbDab Dataset for Clustering . . . 31
3.2.2 Preprocessing of provided Sanger Sequencing Dataset . . . 32
3.3 Clustering and Motif Analysis . . . 33
3.4 Deep Learning Models for Binding Prediction . . . 35
3.4.1 Data Preprocessing and Redundancy Reduction for Deep Learning Models . . . 35
3.4.2 CKSAAP Feature Representation . . . 36
3.4.3 Model Architectures . . . 37
3.4.4 Training, Validation and Testing Strategy . . . 42
3.5 Antibody Sequence Clustering and CDR Annotation Tools . . . 43
3.5.1 Sequence Analysis and Clustering Interface . . . 43
3.5.2 DNA to Protein Translation and Variable-Domain Annotation Interface . . . 44
3.6 Summary . . . 45

4 Results and Discussion **46**
4.1 Clustering and Visualisation Results . . . 46
4.1.1 OPIG CoV-AbDab CDRH3-Only Clustering . . . 46

4.1.2 Sanger Derived CDRH3 Clustering . . . 52
4.1.3 Clustering on OPIG and Sanger Derived CDRH3 Sequences . . . 56
4.1.4 Discussion . . . 58
4.2 Results of Machine Learning Models for Binding Prediction . . . 59
4.2.1 Hyperparameter Optimisation . . . 60
4.2.2 Overall Test Set Performance . . . 60
4.2.3 Receiver Operating Characteristic Analysis . . . 62
4.2.4 Confusion Matrices Analysis . . . 64
4.2.5 Sanger Binders Validation Results . . . 65
4.2.6 Discussion . . . 66
4.3 Support Tools: Usability and Utility Evaluation . . . 67
4.3.1 Functionality Overview . . . 68
4.4 Summary . . . 70

5 Conclusion **71**
5.1 Revisiting Our Aims and Objectives . . . 71
5.2 Critique and Limitations . . . 72
5.3 Future Work . . . 72
5.4 Final Remarks . . . 73

A Amino Acid Codes **86**

B OPIG Cov-AbDab CDRH3 Only Clustering **87**

C University of Malta Sanger Clustering **104**

D Combined OPIG and Sanger Clustering **112**

E Model Hyperparameters and Training Details **129**

F DNA to Protein and Variable Domain Annotation Tool **130**

G Clustering and Visualisation Tool **132**

H Code Availability **135**

List of Figures

Figure 1.1 Antibody paratope–epitope interaction 2

Figure 2.1 Antibody variable domain shown in structural and sequence views. . . 7
Figure 2.2 Short caption 9
Figure 2.3 Dendrograms illustrating Hierarchical Clustering 12
Figure 2.4 Antibody CDR sequence logo highlighting conserved residues 13
Figure 2.5 Workflow for identifying conserved motifs from antibody sequences using BLAST. 14
Figure 2.6 Siamese network architecture 19
Figure 2.7 AbAgIntPre Siamese CNN 24

Figure 3.1 High-level workfow for antbody-antigen binding analysis. 29
Figure 3.2 Workflow for DNA sequence processing 32
Figure 3.3 Stacking ensemble with logistic regression meta-learner. 39

Figure 4.1 Truncated dendrogram for antibodies 47
Figure 4.2 Truncated dendrogram for nanobodies 47
Figure 4.3 Cluster size distribution of antibody sequences 48
Figure 4.4 Cluster size distribution of nanobody sequences 48
Figure 4.5 Detailed dendrogram of Antibody Cluster 1 ($n = 44$). 49
Figure 4.6 OPIG Antibody Cluster Motifs 51
Figure 4.7 OPIG Nanobody Cluster Motifs 52
Figure 4.8 Dendrogram of 137 Sanger CDRH3 sequences 53
Figure 4.9 Size distribution of the eight Sanger CDRH3 clusters. 53
Figure 4.10 Sanger Cluster Motifs 1–4 54
Figure 4.11 Sanger Cluster Motifs 5–8 55
Figure 4.12 Truncated dendrogram (last 30 merges) for combined data 56
Figure 4.13 Cluster size distribution for the combined antibody dataset 57

Figure 4.14 Combined Antibody Dataset Motifs 58
Figure 4.15 Receiver operating characteristic curves for all models. 62
Figure 4.15 (continued) Receiver operating characteristic curve for Siamese AbAgIntPre model 63
Figure 4.16 Confusion matrices for all models 64
Figure 4.17 Translation and Annotation GUI – Main Window 68
Figure 4.18 Sequence Analysis GUI – Main Window 69

Figure B.1 Full dendrogram for antibody sequences 87
Figure B.2 Full dendrogram for nanobody sequences 88
Figure B.3 Detailed dendrogram of Antibody Cluster 2 97
Figure B.4 Detailed dendrogram of Antibody Cluster 3 98
Figure B.5 Detailed dendrogram of Antibody Cluster 4 99
Figure B.6 Detailed dendrogram of Nanobody Cluster 1 100
Figure B.7 Detailed dendrogram of Nanobody Cluster 2 101
Figure B.8 Detailed dendrogram of Nanobody Cluster 3 102
Figure B.9 Detailed dendrogram of Nanobody Cluster 4 103

Figure C.1 Dendrogram of 137 Sanger CDRH3 sequences 104
Figure C.2 Detailed dendrogram for Sanger Cluster 1 107
Figure C.3 Detailed dendrogram for Sanger Cluster 2 108
Figure C.4 Detailed dendrogram for Sanger Cluster 3 108
Figure C.5 Detailed dendrogram for Sanger Cluster 4 109
Figure C.6 Detailed dendrogram for Sanger Cluster 5 109
Figure C.7 Detailed dendrogram for Sanger Cluster 6 110
Figure C.8 Detailed dendrogram for Sanger Cluster 7 111
Figure C.9 Detailed dendrogram for Sanger Cluster 8 111

Figure D.1 Overview dendrogram of combined CoV-AbDab and Sanger CDRH3s. 112
Figure D.2 Detailed dendrogram for Combined dataset Cluster 1 122
Figure D.3 Detailed dendrogram for Combined dataset Cluster 2 122
Figure D.4 Detailed dendrogram for Combined dataset Cluster 3 123
Figure D.5 Detailed dendrogram of Subcluster 1 within Cluster 4 123
Figure D.6 Detailed dendrogram of Subcluster 2 within Cluster 4 124

Figure D.7 Detailed dendrogram of Subcluster 3 within Cluster 4 124
Figure D.8 Detailed dendrogram of Subcluster 4 within Cluster 4 125
Figure D.9 Detailed dendrogram of Subcluster 5 within Cluster 4 125
Figure D.10 Detailed dendrogram of Subcluster 6 within Cluster 4 126
Figure D.11 Detailed dendrogram of Subcluster 7 within Cluster 4 126
Figure D.12 Detailed dendrogram of Subcluster 8 within Cluster 4 126
Figure D.13 Detailed dendrogram of Subcluster 9 within Cluster 4 127
Figure D.14 Detailed dendrogram of Subcluster 10 within Cluster 4 127
Figure D.15 Detailed dendrogram of Subcluster 11 within Cluster 4 128

Figure F.1 Row Details View 130
Figure F.2 Usage Instructions Dialog 131

Figure G.1 Tool Settings Panel 132
Figure G.2 Data Preview 133
Figure G.3 Logo Generation Output 133
Figure G.4 Dendrogram Output 134
Figure G.5 Help Dialog 134

List of Tables

Table 2.1 Comparison of clustering methods in bioinformatics. Complexity and typical usage are based on commonly accepted estimates in the literature (e.g., [49, 52, 55–57]). 16

Table 4.1 Test set performance of all seven models (mean ± standard deviation) . 60

Table A.1 Standard amino acid codes 86

Table B.1 Full list of antibody sequences grouped by cluster for OPIG Dataset only. 88

Table B.2 Full list of nanobody sequences grouped by cluster for OPIG Dataset only. 92

Table C.1 Full list of sequences grouped by cluster for Sanger sequences only. . . 104

Table D.1 Full list of antibody sequences grouped by cluster for Combined Dataset 112

Table D.2 Full list of nanobody sequences grouped by cluster for Combined Dataset 119

Table E.1 Hyperparameter grids and selected values 129

List of Abbreviations

Bi-LSTM Bidirectional Long Short-Term Memory.

BLAST Basic Local Alignment Search Tool.

CDRs Complementarity-Determining Regions.

CKSAAP Composition of k-Spaced Amino Acid Pairs.

CoV-AbDab COVID Antibody Database.

DNA Deoxyribonucleic Acid.

FASTA Fast Alignment.

FR Framework.

GUI Graphical User Interface.

GUIs Graphical User Interfaces.

MSA Multiple Sequence Alignment.

NGS Next Generation Sequencing.

OPIG Oxford Protein Informatics Group.

PR-AUC Precision–Recall Curve: Area Under the Curve.

PSFM Position Specific Frequency Matrix.

ROC-AUC Receiver Operating Characteristic—Area Under the Curve.

UPGMA Unweighted Pair Group Method With Arithmetic Mean.

VH Variable Heavy.

VL Variable Light.

XGBoost Extreme Gradient Boosting.

1 Introduction

The COVID-19 pandemic has highlighted the urgent need for effective therapeutic antibodies to neutralise the SARS-CoV-2 virus [1]. Although several vaccines have been approved and used to prevent disease, some people especially those with weakened immune systems may not develop enough antibodies after getting vaccinated [2]. Antibodies play a central role in the immune response by recognising and neutralising viral pathogens, making their development vital for combating infectious diseases [3]. In this project, we investigate antibody sequence diversity and specificity through two approaches: unsupervised clustering and supervised predictive modeling. Clustering of antibody CDRH3 sequences enables the identification of groups with shared sequence features that may correspond to common binding behaviors, particularly toward the SARS-CoV-2 spike protein. We also develop and evaluate machine learning models to predict antibody antigen binding affinity, supporting the identification of high potential therapeutic candidates based on sequence information alone.

We leverage datasets from the ACCELERATE Phage Display project at the University of Malta and the Oxford Protein Informatics Group (OPIG) at the University of Oxford [4], comprising antibodies targeting SARS-CoV-2, Zika virus[1] (ZIKV), and a broad array of other antigens. This diverse collection provides a robust foundation for both clustering analysis and for training predictive models that estimate antibody–antigen binding characteristics. We also provide a Graphical User Interface (GUI) for performing clustering, translating DNA to protein sequences, and extracting CDR regions without requiring extensive computational background.

1.1 Motivation

Large scale clinical trials have demonstrated that REGEN-COV a combination of two engineered monoclonal antibodies (casirivimab and imdevimab) reduced the risk of developing COVID-19 symptoms among household contacts by approximately 80% when administered subcutaneously, and lowered the risk of hospitalisation or death by

[1] Zika virus is a mosquito transmitted flavivirus first isolated in 1947 in Uganda's Zika Forest. In most adults it causes only mild symptoms (fever, rash, joint pain), but infection during pregnancy can lead to congenital Zika syndrome (notably microcephaly), and it has been linked to Guillain–Barré syndrome in adults. [5]

roughly 70% when given early to high risk outpatients. These findings support its dual role in both the prevention and treatment of SARS-CoV-2 infection [6, 7].

Despite the clear clinical efficacy of monoclonal antibodies, the rational design of next generation antibody therapeutics depends on a deeper understanding of the sequence features that govern high-affinity binding. Antigen binding specificity is primarily determined by the Complementarity-Determining Regions (CDRs), with CDRH3 displaying the greatest sequence diversity and frequently forming the most critical contacts with viral epitopes. The antibody's paratope the binding surface created by its variable region must precisely match the viral epitope, analogous to a key fitting into a lock (Figure 1.1). This intricate complementarity in both shape and chemistry underlies the remarkable specificity and potency of monoclonal antibodies [8, 9].

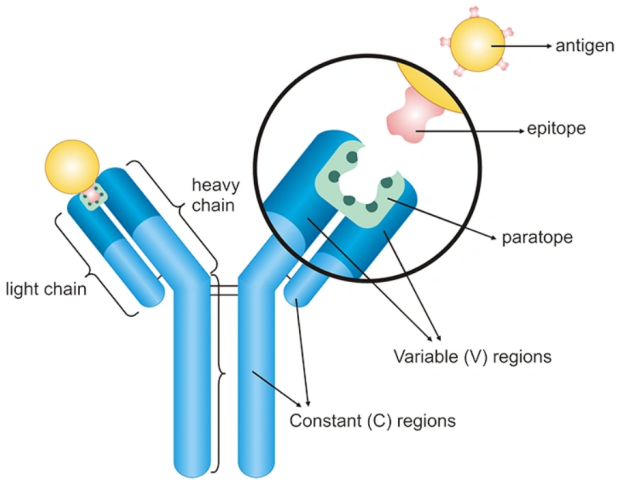


Figure 1.1 The paratope, formed by the complementarity-determining regions (CDRs) in the variable domains of the heavy and light chains, engages with the viral epitope. Adapted from CUSABIO [10].

However, the vast diversity of antibody variable region sequences particularly in terms of length, composition, and structural conformation presents significant challenges for traditional alignment based analyses [11, 12]. Multiple sequence alignment tools often introduce gaps and misalignments, which can obscure the non-linear sequence patterns critical for binding affinity [13, 14]. Overcoming these limitations is essential for advancing the computational discovery of therapeutically potent antibodies, especially against emerging pathogens [12].

1.2 Aims and Objectives

The primary aim of this project is to advance our understanding of the diversity and binding characteristics of SARS-CoV-2 antibodies by combining clustering techniques, sequence visualisation, and predictive modeling. This integrated approach helps to reveal trends in antibody structure and binding potential, particularly in the highly variable CDR-H3 region. The specific objectives are:

1. **Apply Clustering Techniques:** Use interpretable clustering methods such as hierarchical clustering and dendrogram visualisation to explore structural similarities among antibody sequences. By grouping sequences that are highly similar, we can uncover recurring motifs or "public clonotypes" that multiple antibodies use to recognize the same viral epitope. These public motifs often correspond to proven neutralising responses and can guide focused experimental follow up.

2. **Visualise Conserved Patterns:** Generate CDRH3 sequence logos that identify conserved antigen binding "hot spots", reveal variable positions amenable to engineering, and enable comparison of motif patterns across clusters to guide antibody design.

3. **Develop Predictive Models:** Train and evaluate a diverse set of sequence based classifiers including logistic regression, random forest, XGBoost, a Siamese CNN, a Bi-LSTM recurrent model, and a ProtBERT based transformer to predict antibody antigen binding affinity. Perform systematic hyperparameter tuning (e.g. grid-search or randomised search with cross-validation) for each model and compare their performance in terms of ROC AUC, precision, recall, F1-score and confusion matrices.

4. **Build Accessible Software Interfaces:** Develop GUIs that enable biomedical students and researchers to perform clustering, convert DNA to protein sequences, and extract CDR regions, without needing advanced coding experience.

1.3 Our Approach

To investigate antibody antigen binding, we develop a modular computational workflow comprising distinct tasks that address each stage of the analysis. The pipeline covers dataset curation, sequence preprocessing, clustering and motif discovery, predictive modeling, and the use of user-friendly graphical tools. Two primary datasets are used: the publicly available CoV-AbDab, which contains 12,918 entries as of October 2024 [4], and a proprietary Sanger sequencing dataset with 384 entries from the University of Malta. Preprocessing steps include sequence filtering, translation, and annotation, each tailored to the characteristics of the respective dataset. CDRH3 regions are clustered using Basic Local Alignment Search Tool (BLAST) derived distance matrices in combination with hierarchical clustering methods. Motif extraction is then performed using sequence logos to highlight conserved binding patterns. To predict antibody–antigen interactions, we evaluate multiple machine learning models, including logistic regression, random forest, XGBoost, Bi-LSTM, Siamese CNN, and ProtBERT using CKSAAP features and sequence-based embeddings. Each model is trained and validated on carefully curated datasets designed to reduce redundancy and address class imbalance. To improve accessibility and educational value, we develop two standalone graphical user interface (GUI) applications: one for clustering and motif visualization, and another for DNA-to-protein translation and CDR annotation.

1.4 Document Structure

Following the Introduction, the document is divided into four main chapters:

- **Chapter 2: Background and Literature Overview** – This chapter explains the key ideas and information needed to understand the rest of the document. It includes a review of the literature, discussing and comparing research done by other studies. This review forms the foundation for the arguments and decisions made later in the dissertation.
- **Chapter 3: Methodology** – This chapter details the research approach adopted in the study. It describes the architecture, frameworks, datasets, and algorithms that

were implemented, providing an in depth explanation of the steps taken to achieve the research objectives.

- **Chapter 4: Results and Discussion** – In this chapter we present and analyse our clustering results, evaluate predictive model performance, and showcase the GUI outputs alongside user feedback; finally, we benchmark each component against state of the art methods from the literature, discussing both performance gains and observed limitations.

- **Chapter 5: Conclusion** – The final chapter offers a summary of the key findings and contributions of the dissertation. It discusses the limitations of the current research and suggests directions for future work.

2 Background and Literature Review

This chapter provides the necessary background for understanding the computational analysis of antibody sequences, with a focus on sequence similarity, motif discovery, and predictive modeling. It begins by introducing core bioinformatics concepts and visualisation techniques, such as dendrograms, frequency matrices, and sequence logos that are commonly used to analyse and interpret sequence data. Established modern machine learning approaches are then reviewed, highlighting their role in antibody clustering and binding prediction. Finally, the chapter presents recent research relevant to this work, situating it within the broader context of computational immunology.

2.1 The COVID-19 Pandemic

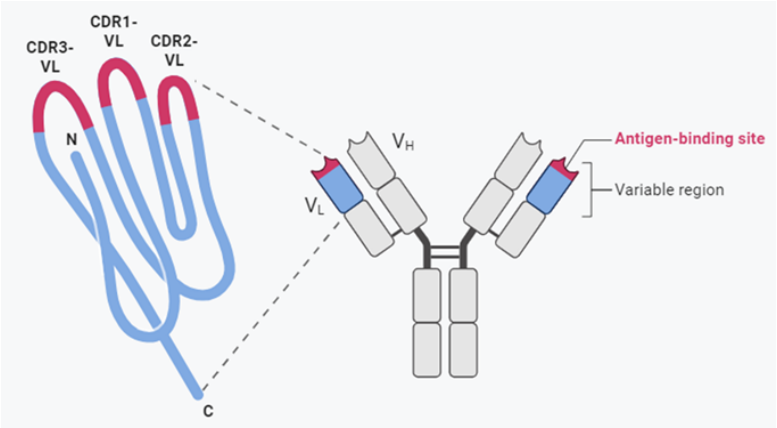
The disease COVID-19, caused by the novel coronavirus SARS-CoV-2, was first reported in Wuhan, China in December 2019 and rapidly escalated into a global pandemic, with over 650 million confirmed cases and 6.6 million deaths by early 2024 [15, 16]. Beyond its devastating human toll, COVID-19 has triggered profound social and economic disruptions worldwide [17]. The urgent need for both prophylactic vaccines and effective therapeutics spurred unprecedented efforts in antibody discovery and engineering. Neutralising monoclonal antibodies targeting the spike protein have not only served as critical early treatments for high-risk patients [18, 19], but have also provided a rich dataset of sequence function relationships that underpins much of today's computational modeling. As a result, COVID-19 has become the foremost model system for developing and benchmarking bioinformatic and machine-learning tools to predict antibody binding and antiviral efficacy [20, 21].

2.2 What are Antibodies and what is an Antigen?

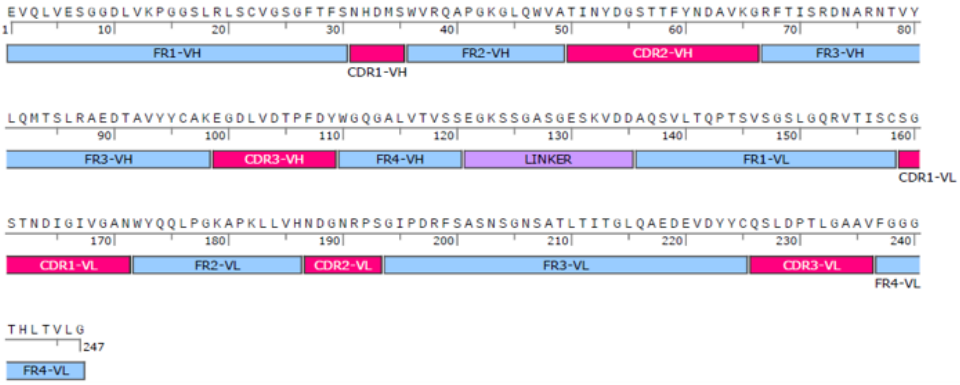
Antibodies are Y-shaped proteins produced by the immune system to identify and neutralize foreign invaders. Each antibody has two identical antigen binding sites at the tips of its arms, called the Fab (fragment antigen-binding) regions which recognize and attach to specific molecular targets called antigens on pathogens such as viruses,

bacteria, or toxins. Once an antibody binds to its target, the stem of the Y, known as the Fc (fragment crystallizable) region, interacts with other parts of the immune system to signal the destruction or removal of the threat. In the context of COVID-19, for example, antibodies that target the spike protein can prevent the virus from infecting host cells [22, 23]. Like all proteins, antibodies are composed of amino acid chains and are typically represented using one letter codes for the 20 standard amino acids (see Appendix Table A.1).

2.2.1 Antibody Variable Domains: Structure and Sequence Representation



(a) Structure-based depiction of Variable Heavy (VH)/Variable Light (VL) domains with CDRs annotated.



(b) Sequence annotation of framework (blue) and CDR (magenta) regions in VH and VL chains.

Figure 2.1 Two views of the antibody variable domains. (a) Structural view showing the CDR loops. (b) Sequence annotation of framework and CDR regions. Both figures reproduced from M. Grima [24].

The antigen binding function of an antibody is governed by its variable domains VH (heavy chain) and VL (light chain) each of which includes four conserved framework

regions (FR1–FR4) and three highly variable loops known as complementarity determining regions (CDRs). These CDRs (CDR1–3) extend from the surface of the domain and together form the antigen binding site (Figure 2.1a). Both VH and VL contribute three CDRs respectively CDR-H1 to H3 and CDR-L1 to L3 which spatially converge to interact with the target antigen. Among these, CDR-H3 is typically the most diverse and functionally dominant [9, 25].

For computational modeling, these variable domains are represented as linear amino acid sequences, which are annotated according to standard numbering schemes (e.g., IMGT, Kabat, Chothia) to identify CDR and framework boundaries. These annotations, as illustrated in Figure 2.1b, are critical for tasks such as clustering, motif analysis, and predictive modeling, and form the basis for the analyses presented in later chapters.

In practice, annotated antibody sequences are processed in the following formats:

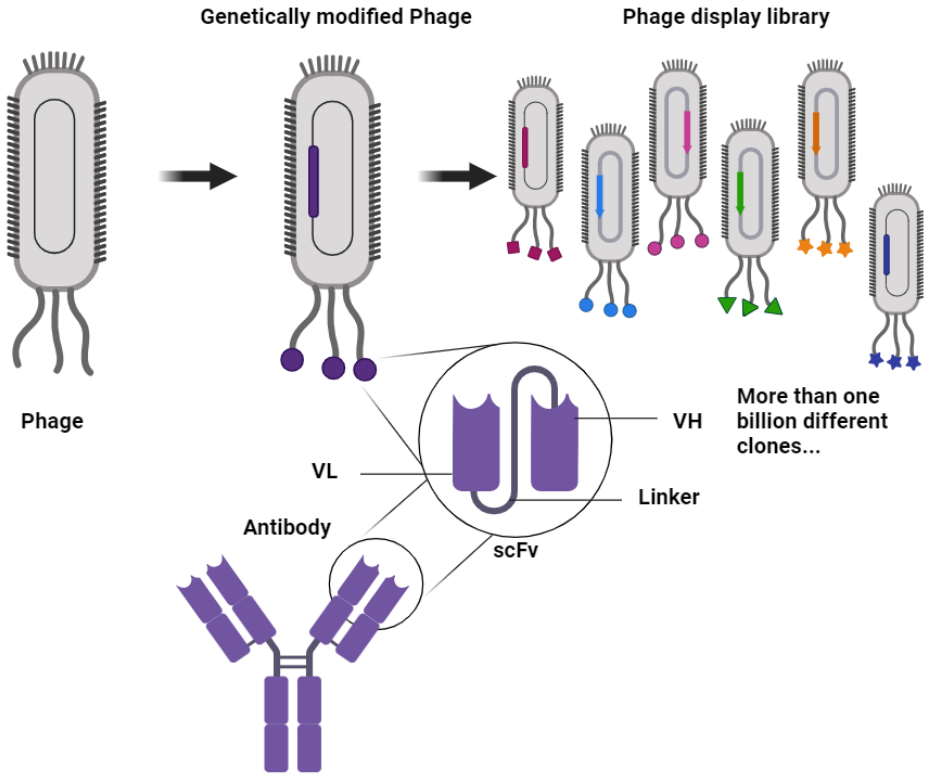
- **Raw Fast Alignment (FASTA):** Plain text files with headers and amino acid sequences.
- **Annotated CSV:** Tables listing region boundaries (e.g., FR1 start–end, CDR-H3 start–end) per sequence.
- **Numerical encodings:** One-hot vectors, residue indices, or learned embeddings (e.g., ProtBERT) for use in machine learning pipelines.

2.3 Phage Display and Biopanning

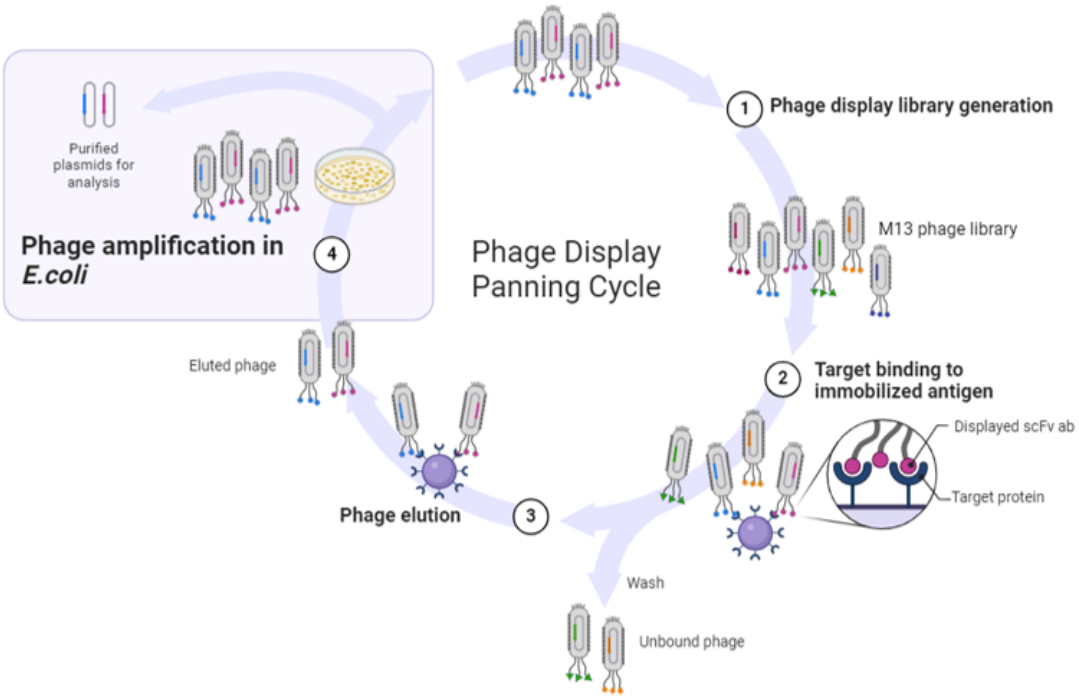
Phage display is a laboratory method that uses harmless viruses (bacteriophages) to link millions to billions of small proteins or antibody fragments (the “library”) to their encoding Deoxyribonucleic Acid (DNA) [26, 27]. Each phage particle displays one variant on its surface and carries its matching genetic “tag” inside.

2.3.1 Biopanning: Iterative Selection of High Affinity Binders

Biopanning is an iterative selection process used to isolate high affinity binders from a phage display library. Figure 2.2 (a) shows the generation of a diverse phage display library via insertion of scFv fragments into the M13 phage genome. Figure 2.2 (b)



(a) Library generation.



(b) Iterative biopanning cycles.

Figure 2.2 (a) Construction of a phage display library; (b) four step panning cycle. Both images reproduced from M.Grima [24].

outlines the four step biopanning cycle used to selectively enrich high affinity clones. Each cycle consists of the following key steps:

1. **Incubation:** The phage library is incubated with the target of interest (e.g., a surface bound protein), allowing specific binders to attach.
2. **Washing:** Unbound or weakly bound phage particles are removed through a series of washes, enriching for higher affinity binders.
3. **Elution:** Bound phage are recovered by altering buffer conditions (e.g., lowering pH or adding a competing ligand) to dissociate them from the target.
4. **Amplification:** The eluted phage are used to infect *E. coli* bacteria, which serve as host cells for phage replication. This step allows the enriched pool of binding phage to multiply and be carried forward into the next selection round.

Typically, 3–5 rounds of biopanning are performed, progressively enriching the library for high affinity binders. The DNA that encodes the peptides or antibody fragments displayed on the surface can then be extracted and sequenced to identify the most promising candidates [24, 28, 29].

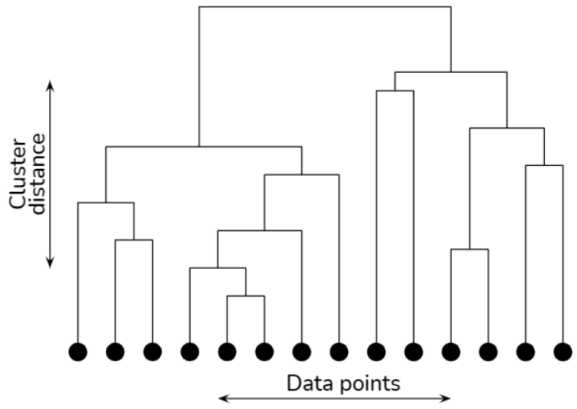
2.4 Visualisation Techniques

Visualisation plays a crucial role in interpreting and communicating the patterns hidden within large scale sequence data. By translating high-dimensional alignments and clustering results into intuitive graphics, we can more readily identify conserved motifs, assess diversity, and guide downstream experimental design. In this section, we introduce two complementary techniques. First, dendrograms and frequency matrices offer a global view of sequence relatedness and residue distributions across clusters. Second, sequence logos provide a concise, position specific depiction of conservation and variability within aligned regions. Together, these methods form a powerful toolkit for exploring antibody repertoires and antigen binding motifs.

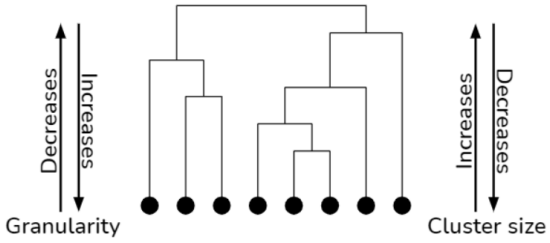
2.4.1 Dendrograms and Frequency Matrices in Sequence Analysis

Dendrograms are hierarchical, tree like diagrams commonly used to illustrate the arrangement and relationships among clusters derived from hierarchical clustering algorithms. In biological sequence analysis, dendrograms serve as essential tools for visualising evolutionary relationships and assessing sequence similarity and divergence [30]. The lengths of dendrogram branches typically represent the degree of sequence dissimilarity, providing critical insights into evolutionary divergence, functional annotations, and potential evolutionary scenarios [31–33]. The construction of dendrograms begins with the computation of a pairwise distance matrix using established distance metrics such as the Hamming distance, Jukes Cantor correction, or Kimura evolutionary models [34–36]. Subsequently, hierarchical clustering techniques, notably the unweighted pair group method with arithmetic mean (UPGMA) and the neighbour joining method, iteratively cluster sequences based on their computed distances [37, 38]. Figure 2.3 presents two aspects of the dendrogram: Figure 2.3a highlights the hierarchical clustering of individual data points based explicitly on calculated distances, clearly illustrating the relationships among clusters. On the other hand, Figure 2.3b emphasizes the trade off between cluster granularity and cluster size, illustrating how increasing granularity yields smaller, more specific clusters, while decreasing granularity leads to larger, more generalized groupings. Together, these dendrograms underscore critical frameworks fundamental to interpreting hierarchical clustering outcomes in biological sequence analysis [39].

Complementing dendrograms, frequency matrices are tabular summaries that count the occurrences of each amino acid (or nucleotide) at each position in an aligned sequence dataset. These matrices form the basis of sequence logos a graphical representation where the height of each letter is proportional to its frequency, weighted by the overall conservation at that particular sequence position [41, 42]. Frequency matrices are fundamental for uncovering conserved regions that often indicate functional importance, such as binding sites in antibody CDRs. Dendrograms offer a macro level perspective of sequence relatedness, whereas frequency matrices enable a finer analysis of residue conservation. Together, they provide a comprehensive picture of sequence variability and conservation, which is vital for subsequent analysis



(a) Hierarchical clustering dendrogram demonstrating relationships between individual data points and the distances between clusters.



(b) Dendrogram illustrating the inverse relationship between granularity (level of detail) and cluster size in hierarchical clustering.

Figure 2.3 Dendrograms illustrating hierarchical clustering in biological sequences. Branch lengths reflect sequence dissimilarity. Reproduced from Prasad Pai *et al.* [40].

in functional genomics and therapeutic design.

2.4.2 Sequence Logo Visualisation

Sequence logos are a visual tool used to summarise the information content of aligned sequences [41]. Each position in the logo is represented by a stack of letters, where the height of each letter reflects its frequency and conservation. This method is particularly useful for identifying conserved and variable residues in functionally significant regions. The creation of a sequence logo involves several steps: Multiple sequence alignment (MSA), construction of a Position Specific Frequency Matrix (PSFM), and graphical rendering of the frequency and information content, often calculated using Shannon entropy [41]. There are different types of sequence logos: frequency logos show raw occurrences, information logos emphasize conserved residues, and enrichment logos compare observed frequencies against a background distribution to highlight overrepresented residues. Sequence logos are especially valuable in immunogenetics for identifying conserved motifs in hypervariable regions, such as antibody CDRs, which may indicate critical functional sites for antigen binding.

Despite their usefulness, sequence logos are limited by the quality and size of the input alignment. Poor alignments or insufficient sampling can lead to misleading interpretations. Moreover, sequence logos do not inherently convey structural or

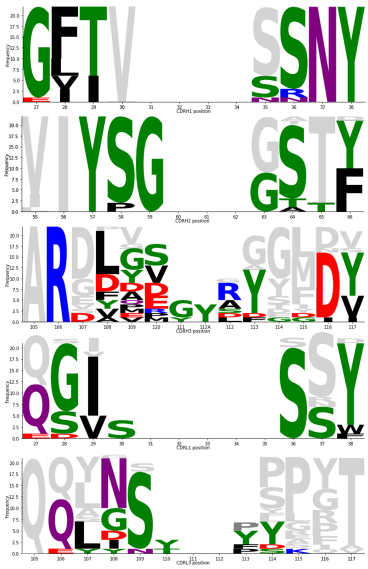


Figure 2.4 Sequence logo generated from antbody CDR sequences. Each letter's height reflects its relative frequency and conservation at a given position, highlighting residues that may be structurally or functionally important for antigen binding. Reproduced from Robinson *et al.* [43].

contextual information unless they are integrated with external data. Nonetheless, sequence logos remain powerful tools for guiding experimental design, such as site directed mutagenesis and epitope mapping, and have recently been integrated into machine learning pipelines for protein function prediction and binding affinity modeling [42, 44]. Combined with other visualisations like dendrograms, they provide a comprehensive view of sequence diversity and molecular interactions in both evolutionary and biomedical research.

2.5 BLAST and Motif Discovery

The Basic Local Alignment Search Tool (BLAST) is a foundational algorithm in computatonal biology, widely used for comparing primary biological sequence data such as DNA, RNA, or protein sequences [45]. BLAST identifies regions of local similarity between sequences using a word based heuristic strategy, allowing for rapid comparison against large databases. This local alignment approach offers a significant advantage over global alignment algorithms, particularly when searching for homologous regions within highly divergent sequences. By focusing on smaller conserved subsequences, BLAST is able to uncover biologically meaningful relationships that might be obscured in a full sequence comparison [46, 47].

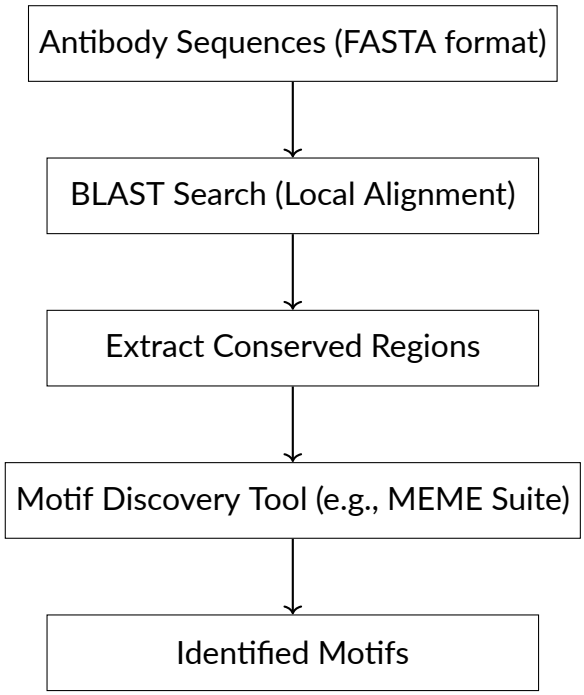


Figure 2.5 Schematic diagram of a BLAST based motif discovery pipeline. This workflow illustrates the steps from input antibody sequences in FASTA format, through local sequence alignment using BLAST, extraction of conserved regions, analysis with a motif discovery tool (e.g., MEME Suite), to the final identification of motifs.

A key strength of BLAST lies in its flexibility. The algorithm provides tunable parameters including word size, gap penaltes, and scoring matrices that enable users to adjust the balance between sensitivity and computational efficiency. For motif discovery applications, these parameters can be optimised to detect short, highly conserved regions that are often embedded within otherwise variable sequences. For instance, reducing the word size increases the sensitivity to shorter alignments, which is particularly useful when searching for functional motifs within antibody variable regions. Following BLAST alignment, conserved regions of interest can be extracted for further analysis using specialized motif discovery tools. One widely used tool in this context is the MEME Suite, which identifies statistically significant sequence patterns (motifs) using probabilistic models such as expectation maximisaton [48, 49] .

These motifs often correspond to functionally important regions such as CDRs in antibodies that are critical for antigen recognition and binding. By characterising these motifs, researchers can gain insights into the structural and functional properties of antibody repertoires. Despite its utility, BLAST based motif discovery also has limitations. The initial sensitivity of BLAST may restrict detection to motifs with

strong conservation, potentially missing subtle but functionally relevant patterns [26]. Additionally, the output of motif discovery tools is highly dependent on input quality and sequence diversity; false positives can arise if appropriate background models are not employed [50]. A schematic diagram of the BLAST-based motif discovery workfow is presented in Figure 2.5. The pipeline begins with antibody sequence data in FASTA format, which is aligned using BLAST to identify locally conserved regions. These regions are then input into a motif discovery tool such as MEME to identify shared sequence patterns across the dataset. The resulting motifs can then be mapped back to known functional domains or used as candidates for further experimental validation.

2.6 Machine Learning Approaches

In this section, we first review **unsupervised learning** methods, focusing on how clustering algorithms group antibody sequences into families and functional subtypes without requiring labels. We then explore **supervised learning techniques**, ranging from classical models like logistic regression and random forests to advanced deep learning methods such as bidirectional LSTMs and pretrained transformer models. Finally, we discuss **Siamese and other similarity learning architectures**, which learn embedding spaces where distances directly reflect functional or binding relevance.

Together, these methods illustrate the breadth of machine learning paradigms applied to antibody informatics: unsupervised discovery of sequence/structure families, supervised pattern recognition of binding sites, and metric learning frameworks that underlie modern similarity search and clustering pipelines.

2.6.1 Unsupervised Learning: Clustering Methods

Clustering is an unsupervised machine learning technique that aims to group similar data points based on a chosen metric. In bioinformatics, clustering has become an invaluable tool for analysing high-dimensional data such as protein sequences, gene expression profiles, and, in particular, antibody sequences [49]. Traditional clustering methods include k-means clustering, hierarchical clustering, and density based methods. Each approach has its own advantages and limitations in terms of scalability, sensitivity, and the ability to handle noise.

Hierarchical clustering, in particular, is widely used for the analysis of biological sequences [49, 51]. It generates dendrograms that not only illustrate groupings of data but also allow a visual assessment of the similarity between clusters. More advanced clustering algorithms incorporate additional features, such as physicochemical properties and predicted structural attributes [52, 53]. For instance, combining sequence alignment scores with structural domain information in distance metrics has been shown to enhance clustering performance in challenging biological datasets [54].

Table 2.1 presents a comparison of common clustering algorithms, highlighting the computational complexity and suitability for different types of biological data. The selection of an appropriate clustering algorithm is largely dependent on the specific goals of the analysis, as well as on the nature of the data. In the context of antibody research, where subtle sequence differences can have dramatic impacts on binding behaviour, the choice of the clustering technique may determine the success of the subsequent analysis. Increasingly, hybrid approaches that combine multiple clustering methods are being explored to leverage their complementary strengths [58].

Table 2.1 Comparison of clustering methods in bioinformatics. Complexity and typical usage are based on commonly accepted estimates in the literature (e.g., [49, 52, 55–57]).

Method	**Complexity**	**Typical Usage**
k-means	O(nkt)	Large datasets with a small, fixed number of compact, spherical clusters
Hierarchical	O(n^3)	Dendrogram construction for evolutionary studies; no need to pre-specify number of clusters
DBSCAN	O($n \log n$)	Noisy datasets with irregular cluster shapes; discovers clusters of arbitrary shape

2.6.2 Supervised learning Methods

In this section, we review the evolution of prediction methods from classical supervised learning approaches which have been widely applied to various biological sequence classification tasks [59] [60] [61] [62], particularly those designed for capturing patterns directly from sequence data.

Logistic Regression

Logistic regression is a simple yet powerful method for binary classification. It models the probability of the positive class by passing a weighted sum of the input features through a sigmoid (logistic) function, and is trained by minimizing the cross-entropy loss [63]. Because its decision boundary is linear in feature space, each learned weight directly reflects how strongly—and in which direction—a feature influences the predicted outcome, making the model highly interpretable [64]. However, it cannot capture non-linear interactions between features unless you explicitly include transformed or interaction terms.

Random Forest

Random forests are an ensemble learning method for classification (and regression) that builds many decision trees and aggregates their predictions. Each tree is trained on a different bootstrap sample of the data, and at each split a random subset of features is considered, which decorrelates the trees and reduces overfitting compared to a single decision tree [65]. Final predictions are made by majority vote (for classification) or averaging (for regression). In addition to strong predictive performance, random forests naturally provide feature-importance measures—often based on how much each feature decreases node impurity across all trees—offering insight into which inputs drive the model's decisions [66].

XGBoost

While conceptually similar to random forests, gradient boosting machines like XGBoost offer significant advantages for antibody antigen binding prediction. Rather than building trees independently, XGBoost constructs a series of weak learners sequentially, with each model correcting errors made by its predecessors [67]. Its sophisticated regularization techniques and efficient handling of sparse data make it particularly well suited for the complex, high-dimensional feature spaces typical in immunological datasets. In comparative benchmarks across multiple antibody datasets, I found XGBoost consistently outperformed other traditional methods when trained on physicochemical and structural properties derived from sequence data. The

effectiveness of these traditional approaches, however, depends heavily on feature engineering choices. This limitation has motivated the exploration of deep learning architectures that can learn meaningful representations directly from raw sequence data.

Bidirectional LSTM Networks

The sequential nature of antibody CDR loops makes recurrent neural networks (RNNs) [68] a natural modeling choice. Standard RNNs, however, often struggle with the vanishing gradient problem when modeling long range dependencies [69]. Long Short-Term Memory (LSTM) networks address this limitation through specialized gating mechanisms that regulate information flow [70]. The bidirectional variant processes sequences in both forward and reverse directions, enabling each position to incorporate context from both upstream and downstream residues [71]. In my experiments, bidirectional LSTMs have demonstrated superior performance over traditional models, particularly when predicting binding for novel antibody classes where local sequence context significantly influences binding pocket formation. This advantage stems from the architecture's ability to capture non local interactions between residues that may be spatially adjacent in the folded protein despite being distant in the linear sequence.

ProtBERT and Pretrained Protein Transformers

Transformer models represent a significant leap forward in sequence modeling capabilities, leveraging self-attention mechanisms to capture relationships between all positions simultaneously [72]. For antibody research, pretrained protein language models like ProtBERT [73] offer particularly compelling advantages. By pretraining on massive protein corpora, these models learn general protein structural and functional patterns that can be fine tuned for specific binding prediction tasks [74]. The contextualized embeddings generated by these models capture evolutionary constraints and physically meaningful relationships between residues, providing rich representations even with limited labeled binding data. My preliminary results suggest that freezing early layers of ProtBERT while fine tuning deeper layers offers an optimal balance between leveraging general protein knowledge and adapting to the specific binding task at hand.

General Siamese Architecture for Similarity Learning

Building on the Siamese concept, more generalized architectures for similarity learning have emerged as powerful tools for antibody antigen modeling. These approaches maintain the twin subnetwork structure with a shared encoder function $f(\cdot)$ that maps inputs x_i and x_j to embeddings $e_i = f(x_i)$ and $e_j = f(x_j)$ [75]. An overview of this architecture is shown in Figure 2.6. The flexibility in designing both the encoder architecture and the similarity metric enables adaptation to specific characteristics of antibody antigen interactions. The choice of loss function significantly influences

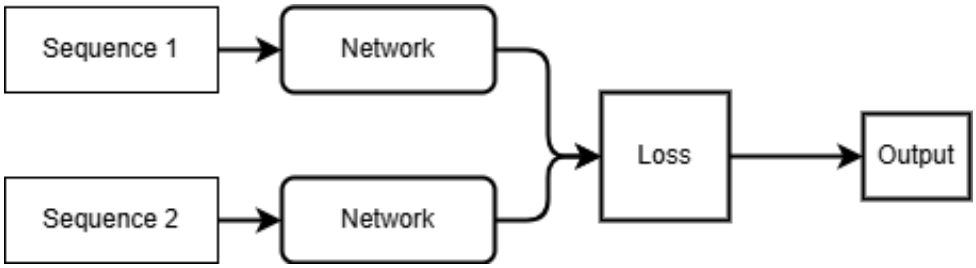


Figure 2.6 Siamese network architecture: Twin encoders with shared weights process paired inputs, and their embeddings are compared to determine binding probability.

embedding quality, with options like contrastive, triplet, and margin-based losses each offering trade-offs in convergence and discriminative power [76]. In tasks involving binding classification, particularly when combining sequence and structural features, triplet loss with effective negative mining has shown promise in producing robust embeddings, though outcomes may vary depending on model architecture and data characteristics [77]. These deep learning approaches represent the current state of the art in antibody antigen binding prediction, though their relative advantages depend on data availability, interpretability requirements, and the specific binding properties being modeled.

2.7 Evaluation Metrics

Evaluation metrics play a crucial role in assessing the performance of both clustering algorithms and binding/non-binding prediction models, particularly in bioinformatics tasks such as antibody antigen interactions. These metrics quantify a model's ability to detect meaningful patterns or make accurate predictions. The following

subsections describe several essential evaluation metrics used for clustering and binding/non-binding classification.

2.7.1 Clustering Evaluation Metrics

Clustering is an unsupervised learning method used to group similar data points based on certain criteria. In bioinformatics, clustering is often used to find similar sequences or identify patterns in large datasets. The evaluation of clustering results is challenging without a ground truth, but several metrics can still provide valuable insights.

Cophenetic Correlation Coefficient

For hierarchical clustering, a standard goodness of fit index is the *cophenetic correlation coefficient* (CCC). Once a dendrogram is constructed, the **cophenetic distance** c_{ij} between two observations i and j is the height at which their branches first merge. The CCC is simply the Pearson correlation between the original pairwise distance matrix $\{d_{ij}\}$ and the cophenetic distance matrix $\{c_{ij}\}$:

$$\text{CCC} = \frac{\sum_{i<j}\left(d_{ij}-\bar{d}\right)\left(c_{ij}-\bar{c}\right)}{\sqrt{\sum_{i<j}\left(d_{ij}-\bar{d}\right)^2}\sqrt{\sum_{i<j}\left(c_{ij}-\bar{c}\right)^2}},$$

where $\bar{d}$ and $\bar{c}$ are the means of $\{d_{ij}\}$ and $\{c_{ij}\}$, respectively [78]. Values range from -1 (poor agreement) to $+1$ (perfect representation):

- **CCC** $\approx +1$ — the dendrogram preserves the original dissimilarities almost perfectly;
- **CCC** ≈ 0 — the tree is no better than random at reflecting the distances;
- **CCC** < 0 — substantial distortions; some distance relationships are reversed.

Because CCC quantifies how faithfully a tree preserves the geometry of the data, it is useful for for deciding whether a hierarchical model is appropriate for a given dataset [78].

Elbow Method

The Elbow Method is commonly used to determine the optimal number of clusters. It involves plotting the within cluster sum of squares (inertia) against the number of clusters. The elbow point, where the decrease in inertia slows down, is often considered the optimal number of clusters [79].

2.7.2 Binding/Non-Binding Evaluation Metrics

For binding/non-binding prediction models, which are widely used in antibody antigen interaction studies, it is crucial to evaluate the model's ability to distinguish between positive and negative interactions.

Accuracy

Accuracy is the most straightforward metric, representing the proportion of correct predictions (both true positives and true negatives) out of all predictions. It is given by:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}} \tag{2.1}$$

While commonly used, accuracy can be misleading in cases of class imbalance [80].

Precision and Recall

Precision measures the proportion of true positives among all positive predictions made by the model:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.2}$$

Recall (or Sensitivity) measures the proportion of actual positives (binders) that are correctly identified:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.3}$$

Precision and Recall are essential when dealing with imbalanced datasets [81].

F1-Score

The F1-Score is the harmonic mean of Precision and Recall, providing a balance between them:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.4}$$

The F1-Score is particularly useful when there is a need to balance false positives and false negatives [81].

AUROC (Area Under the ROC Curve)

The AUROC (Area Under the Receiver Operating Characteristic Curve) is an important metric for evaluating binary classifiers. The ROC curve plots the true positive rate against the false positive rate at various classification thresholds, showing the trade-off between sensitivity and specificity. The AUROC summarises this curve into a single value: 0.5 indicates random guessing, while a value close to 1.0 indicates high classification accuracy [82].

2.8 Related Work

This section reviews recent computational work relevant to this thesis, focusing on two key areas: (i) unsupervised clustering of antibody sequences into clonotypes, and (ii) supervised prediction of antibody antigen binding from sequence data. The following subsections cover recent advances in clustering methods and predictive modeling approaches, from classical machine learning to modern deep learning architectures. Key algorithms, input representations, performance metrics, and trade-offs are highlighted to contextualise the methodological choices made in this work.

2.8.1 Clustering Methods for Antibody Repertoires

Clustering antibody sequences is a fundamental step in immunoinformatics, enabling the identification of clonotypes and the analysis of immune repertoire diversity. Accurate clonotype identification is crucial for understanding immune responses, as finding clonotypes helps us understand how B cells respond to infection. Agglomerative clustering, first formalised by Sokal and Michener [83] and popularised in molecular

biology by Eisen *et al.* [84], builds a hierarchy of clusters by iteratively merging the most similar pairs of sequences. This approach allows researchers to visualise nested relationships and apply biologically meaningful cut-offs, making it especially suitable for antibody studies where clone sizes can vary widely. Among agglomerative strategies, Ward's minimum variance linkage is widely used in repertoire analysis because it produces balanced clusters and minimizes within cluster heterogeneity; for example, it underpins the Clonify lineage assignment tool for large IGH (Immunoglobulin Heavy Chain) repertoires [85]. While agglomerative clustering is computationally intensive for very large datasets, it remains a mainstay in the field due to its interpretability and ability to accommodate uneven clone sizes. Recent years have seen the development of alternative clustering approaches, including density based methods like HDBSCAN [86], graph based techniques such as Markov Clustering (MCL) [87], and consensus frameworks that integrate multiple algorithms. Structure based and CDR based clustering methods have also emerged, aiming to identify functionally related antibodies that may have low sequence similarity but similar binding properties [88]. However, these methods require additional computational resources and specialised data, and their performance can be sensitive to sequence length and structural modeling quality. Recent benchmarking shows that structure-, paratope-, and embedding-based clustering can uncover functionally convergent antibodies missed by sequence-based methods. However, clonotype clustering based on sequence, particularly using agglomerative approaches with shared V/J gene usage and CDRH3 identity, remains a specific and robust standard [88]. However, no single method universally outperforms all others, and the most comprehensive analyses leverage multiple clustering strategies to maximize diversity and functional insight. Tools that further optimise hierarchical clustering for speed and scalability have been developed, enabling the analysis of millions of sequences on standard hardware. In practice, the choice of clustering method is guided by the research question and available data; for clonotype identification and lineage analysis, sequence based agglomerative clustering is both efficient and reliable.

In summary, while a variety of clustering methods are available, sequence based agglomerative clustering remains a robust and widely adopted standard for clonotype identification, especially when combined with other approaches to maximise biological insight [89–93]. **Guided by this rationale, we also apply agglomerative clustering to our**

own datasets (Section 3.3).

2.8.2 Models for Binder/Non-Binder Classification

Although considerable work has relied on structural data for antibody antigen binding predictions, purely sequence based approaches have been less explored historically, primarily due to the challenge of capturing structural interactions solely from sequence information. However, recent advances in deep learning have led to the emergence of several notable sequence only frameworks. These include AbAgIntPre (a Siamese convolutional network utilizing CKSAAP encoding) [59], AttABseq (an attention based model for predicting binding affinity changes) [94], and newer models such as MVSF-AB [95] and A2binder [96]. While the number of fully sequence based methods remains limited compared to structure informed approaches, their development highlights both the complexity and the growing potential of sequence driven prediction a challenge and opportunity that this thesis addresses by building upon these foundational advances. A central task in therapeutic discovery is to determine, using only sequence information, whether an antibody will bind to a given antigen. The following sections review representative models, highlighting their architectures, applications, and reported performance.

AbAgIntPre

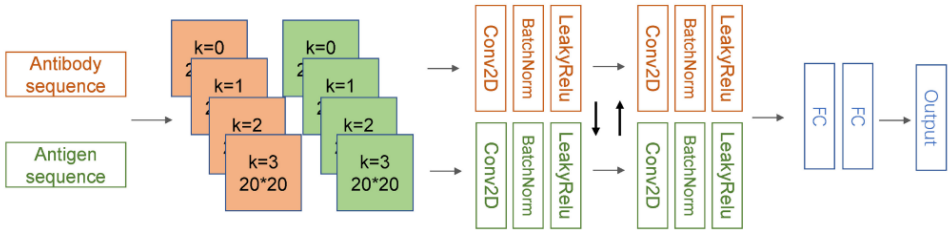


Figure 2.7 Overview of the AbAgIntPre architecture. Reproduced from Huang *et al.*, 2022 [59].

AbAgIntPre is a deep learning based model designed to predict antibody antigen interactions based solely on amino acid sequences, addressing the limitations of traditional experimental methods that often require costly and time consuming processes. The model utilizes a Siamese like CNN architecture described in Section

2.6.2, combined with a sequence encoding technique called CKSAAP (Coupled K-Spaced Amino Acid Pairs), which captures amino acid pair frequencies over varying distances. By training on large datasets of antibody antigen pairs, AbAgIntPre effectively learns the interaction patterns between these biological molecules. The architecture of AbAgIntPre, illustrated in Figure 2.7, processes both antibody and antigen sequences through parallel, shared weight convolutional pathways. In the input module, sequences are encoded using the CKSAAP scheme, capturing the frequency of amino acid pairs across multiple distances and providing a rich feature space for learning interaction patterns. The encoded features then pass through two convolution modules, each consisting of batch normalization, convolution, ReLU activation, and pooling layers, followed by two fully connected layers that map the learned features to the probability of interaction via a sigmoid activation. This design enables AbAgIntPre to achieve robust performance, with a reported AUROC of 0.82 on independent test data, and to outperform traditional machine learning approaches such as random forest, SVM, and logistic regression [59]. Furthermore, AbAgIntPre is implemented as an accessible web server, making it a valuable tool for the scientific community in high-throughput antibody antigen interaction prediction. **In our study, we re-implement this architecture as a baseline and compare it with classical ML and transformer models on the same curated SARS-CoV-2 dataset (see Section 3.3).**

AttABseq

AttABseq is another deep learning based method that utilises protein sequence data to predict antigen antibody binding affinity changes resulting from point mutations. Developed by Jin *et al.* [94], AttABseq employs an attention based architecture that processes input sequences of antigen antibody complexes, extracting relevant features for the accurate prediction of binding affinity variations caused by amino acid mutations. The model comprises three main components: an embedding block for sequence feature encoding, an attention block that integrates crucial information from protein complexes, and a predicting block that forecasts affinity changes. The attention mechanism allows AttABseq to focus on important residue interactions, which improves both prediction accuracy and generalizability across diverse antigen antibody pairs. AttABseq stands out for its ability to handle large scale datasets without requiring structural information,

making it a highly accessible tool for scenarios where experimental structures are unavailable. Benchmarking on three widely used datasets (AB1101, S1131, and the authors' internal dataset), AttABseq demonstrated a 120% improvement in accuracy over other sequence based models such as PIPR and DeepFE-PPI, as measured by the Pearson correlation coefficient between predicted and experimental binding affinity changes [94]. Specifically, on the AB1101 dataset, AttABseq achieved a Pearson correlation coefficient of 0.587, representing an 8.7% improvement over DeepFE-PPI, while on the S1131 dataset, it achieved an R^2 value of 0.368, outperforming PIPR by 77.2% in 10-fold cross-validation [94]. AttABseq also either outperformed or performed comparably to structure based methods such as FoldX, highlighting its efficacy in practical settings. By relying solely on sequence information, AttABseq provides an efficient and scalable alternative for antibody screening and design. Its strong generalization capability and state of the art results across diverse datasets underscore its potential for real world applications in therapeutic antibody optimization. The attention based interpretability analysis further enables visualization of the causal effects of point mutations on antibody antigen affinity changes at the residue level, facilitating automated antibody sequence optimization [94].

MVSF-AB

MVSF-AB (Multi-View Sequence Feature learning for accurate Antibody antigen Binding affinity prediction) is a novel deep learning framework specifically designed for predicting antibody antigen binding affinity using only sequence information. The model addresses the limitations of existing sequence based approaches, which often underperform when applied to antibody antigen affinity prediction due to imbalanced training data and inadequate model design for this specific biological context. MVSF-AB fuses multi view sequence features, including semantic features extracted using a pretrained protein language model (proteinBERT) and residue features derived from physicochemical property matrices. The semantic features are processed by a convolutional neural network (CNN) to learn global correlations between antibody and antigen sequences, while the residue features are fed into a multilayer perceptron (MLP) to capture local binding site characteristics. The outputs from both pathways are combined to produce the final affinity prediction. Experimental results demonstrate

that MVSF-AB outperforms existing sequence based and even some structure based methods in predicting the affinity of both natural and mutated antibody antigen pairs. On the SAbDab dataset, MVSF-AB achieves a root mean square error (RMSE) of 1.839 kcal/mol and a Pearson correlation coefficient (R) of 0.491 in cross-validation, with even better performance on holdout and expanded benchmark datasets. The model maintains its effectiveness when predicting the affinity of antibody mutants, highlighting its robust generalisation ability. MVSF-AB thus represents a significant advance in sequence only antibody antigen affinity prediction, offering a powerful tool for therapeutic antibody engineering and vaccine design [95].

A2binder

A2binder is a recent deep learning model designed for large-scale prediction of antibody antigen binding affinity using only sequence information. A2binder leverages a large-scale pre-trained language model for feature extraction from both antibody and antigen sequences, followed by multi-scale feature fusion and a final prediction step using a Multi-Fusion Convolutional Neural Network (MF-CNN) [96]. This architecture enables A2binder to capture complex sequence patterns and make accurate predictions even for previously unseen antigens, enhancing its utility in high-throughput screening and therapeutic antibody discovery. A2binder has been benchmarked on several datasets, including CoV-AbDab and BioMap, where it achieved state of the art performance. On the CoV-AbDab dataset, A2binder attained an ROC-AUC of 0.930 and a Precision–Recall Area Under the Curve (PR-AUC) of 0.922, outperforming other baseline methods such as AbMAP, AntiBERTa2, ESM-F, Ens-Grad, and Vanilla BERT [96]. Additionally, on the 14H and 14L datasets, A2binder demonstrated strong Spearman rank correlation coefficients of 0.553 and 0.688, respectively. When evaluated on the BioMap dataset, A2binder achieved a Spearman's correlation of 0.746 and a Pearson's correlation of 0.701, further highlighting its robustness and generalizability across different antigen types [96]. The model's ability to process both antibody and antigen sequences, and to make accurate predictions even for unknown antigens, makes it a powerful tool for antibody engineering and drug discovery. As the field evolves, models like A2binder are expected to be further benchmarked and validated on increasingly diverse datasets [96].

Structure Based and Hybrid Models

While sequence only models have made significant advances, many antibody antigen binding prediction tools rely on structural information to improve accuracy and interpretability. Structure based methods, such as AlphaFold-Multimer [97] and HelixFold-Multimer [98], predict the three-dimensional complex of antibody and antigen directly from sequences, enabling detailed analysis of binding interfaces and interactions. Other approaches combine predicted or experimentally determined structures with docking, refinement, or machine learning to further enhance prediction of binding affinity and specificity [99, 100]. These structure informed models are especially valuable when experimental data is limited or when high precision is required for therapeutic design.

Comparison between sequence based antibody antigen prediction models

Taken together, the sequence-only models split into two camps: (i) lightweight CNN/Siamese architectures that rely on hand-crafted encodings (CKSAAP in AbAgIntPre) and (ii) pipelines that leverage transformer protein-language-model embeddings (AttABseq, MVSF-AB, A2binder). Performance rises with model capacity from AUROC 0.82 in AbAgIntPre to 0.93 in A2binder but so does GPU demand and training data size. Moreover, each study benchmarks on a different subset of SAbDab derived pairs, making direct leaderboard claims tenuous. In contrast, structure-based methods achieve atomic detail but incur >10× higher inference time and hinge on accurate folding of both partners.

2.9 Summary

This chapter reviewed key methods for analysing antibody sequences. It covered visual tools (like dendrograms and sequence logos), classic tools such as BLAST, and modern machine learning approaches, including Siamese networks and transformer based models. Related work in this area was also discussed. Together, these methods provide the foundation for the models and experiments presented in the next chapters.

3 Methodology

As shown in Figure 3.1, the antibody–antigen binding analysis workflow begins with data collection and preprocessing, progresses through clustering, visualisation, and predictive modeling, and is supported by two user interfaces tailored for biomedical scientists: one for DNA to protein translation and variable domain annotation, and another for interactive exploration of clustering outcomes. This chapter presents the full design and implementation of this workflow. Section 3.1 defines the datasets. Section 3.2 describes the preprocessing steps applied to prepare the data for clustering and visualisation. Section 3.3 details the clustering and visualisation pipeline used to organise complementarity-determining region motifs and explore sequence diversity. Section 3.4 outlines the predictive modeling framework, covering feature engineering, model selection, training procedures, and evaluation metrics for forecasting binding affinity. Section 3.5 describes the development of the graphical user interfaces mentioned above. All scripts and source code used in this workflow are openly available on GitHub [1], as described in Appendix H.

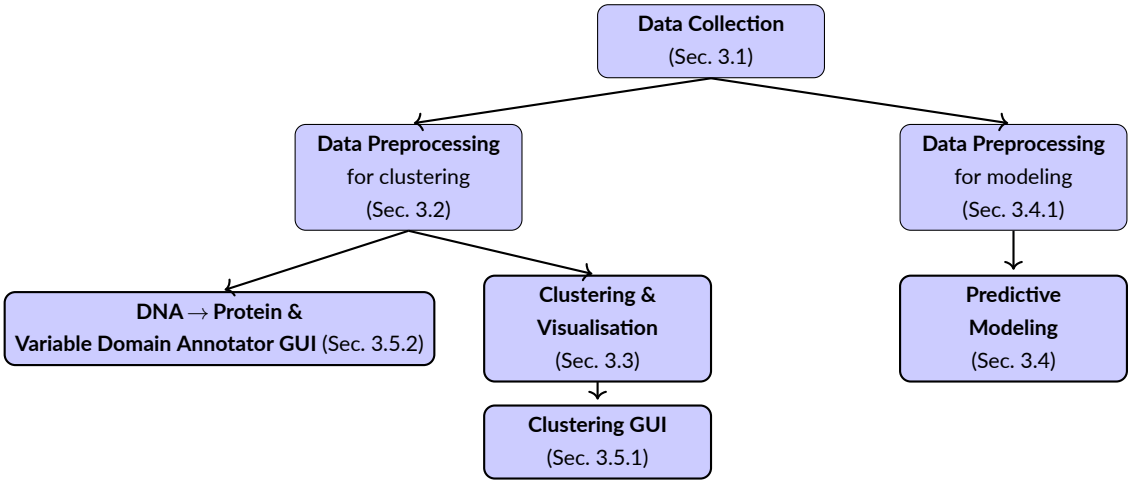


Figure 3.1 High-level workfow for antbody-antigen binding analysis.

3.1 Datasets

The dataset used in this work is aggregated from two main sources: the publicly available OPIG CovAbDab dataset [4], and proprietary data belonging to Dr. David Saliba

[1] https://github.com/francescac56

(co-supervisor), obtained through Sanger sequencing at the University of Malta.

3.1.1 OPIG-CovAbDab Dataset

The COVID Antibody Database (CoV-AbDab) at the University of Oxford [4] is an open-access repository curated by the Oxford Protein Informatics Group. As of October 2024, it comprises 12,918 experimentally characterised antibodies and nanobodies targeting SARS-CoV2 and related betacoronaviruses such as SARS-CoV-1 and MERS-CoV. Each entry includes detailed information, such as the molecule type (antibody or nanobody), binding specificity, neutralisation activity, protein and epitope targets, and origin species or library. Sequence data are provided for full variable domains, including heavy (VH or VHH) and light (VL) chains where available, along with annotated complementarity-determining regions (CDRs) based on ANARCI numbering [101]. Germline gene assignments (V and J genes for both heavy and light chains) and specific CDRH3 and CDRL3 sequences are included. Some structural information is provided through references to resolved structures in the Protein Data Bank (PDB) [102] or to antibody homology models generated using ABodyBuilder (ABB) when no experimental structure is available [103]. Additional metadata, such as source references, date of data entry, and update notes, are included to support traceability and reproducibility.

3.1.2 University of Malta Sanger Sequencing Data

A complementary dataset used in this study containing 384 (137 unique) data points was provided by Mariana Grima as part of her 2024 Master's dissertation [24]. The data originated from a naïve canine scFv [2] phage display library, made available through collaboration with Prof.Ted Hupp (University of Edinburgh) and Dr. Lisowska (ICCVS, University of Gdańsk). In this system, single-chain variable fragment (scFv) antibodies composed of VH and VL domains connected by a short flexible linker are expressed on the surface of bacteriophages. This format couples phenotype (the displayed antibody) with genotype (retrievable DNA sequence), enabling downstream sequence based analysis. The phage library consisted of a highly diverse pool of clones, each expressing

[2] Single-chain variable fragments (scFvs) are engineered antibody fragments consisting of the variable regions of the heavy (VH) and light (VL) chains, connected by a short flexible linker.

a unique scFv. These were subjected to phage display biopanning to enrich for optimal binders to specific targets, including SARS-CoV-2 proteins, immune checkpoint molecules such as TIM-3, and tumour associated antigens like CEA5. After enrichment, individual phage clones were screened for binding activity using an ELISA based assay involving target proteins, phage particles, and a fluorescently labelled anti-phage antibody. Binding strength was quantified using a TECAN Spark plate reader to measure fluorescence intensity the higher the signal, the stronger the presumed binding. Both high-fluorescence binders and low-fluorescence non-binders were subsequently sent for Sanger sequencing. The resulting DNA sequences were aligned and compared to identify patterns of similarity and potential distinguishing features between the binder and non-binder populations.

3.2 Data Preprocessing for Clustering Analysis

Before performing clustering and motif discovery, both datasets used in this study the public OPIG CoV-AbDab resource and the proprietary Sanger sequences, required preprocessing to ensure consistency, quality, and compatibility with downstream computational tools. This section outlines the steps taken to filter, clean, and annotate each dataset, focusing specifically on the extraction of CDRH3 sequences and their preparation for alignment based analysis. Separate procedures were applied to the OPIG dataset and the Sanger dataset, as detailed below.

3.2.1 Preprocessing of OPIG CovAbDab Dataset for Clustering

The OPIG Cov-AbDab dataset [4] was retrieved from the OPIG website in CSV format. To ensure a homogeneous set of clones for cluster based motif analysis, the following steps were applied:

1. **Filter by origin:** Retain only entries whose **Origin** field contains the substring *phage*, thus restricting to phage display derived clones. This filtering step ensures that only antibodies generated via phage display are retained, providing a consistent experimental background and avoiding confounding variation introduced by other discovery methods.

2. **Extract CDRH3:** Select the CDRH3 column and drop any rows with missing or non-standard characters. This ensures we only feed complete CDRH3 into the alignment and clustering pipeline. Entries with missing residues or invalid symbols would otherwise break the distance calculations and skew the results.

3. **Assign clone IDs:** Use the Name column as the unique identifier for each sequence.

4. **Deduplication:** Remove any exact duplicate CDRH3 sequences to avoid bias in downstream distance calculations.

The resulting set of phage display CDRH3 sequences containing 418 CDRH3s was then passed to the clustering pipeline (Section 3.3), where pairwise BLAST e-values were converted to distances and used for hierarchical clustering.

3.2.2 Preprocessing of provided Sanger Sequencing Dataset

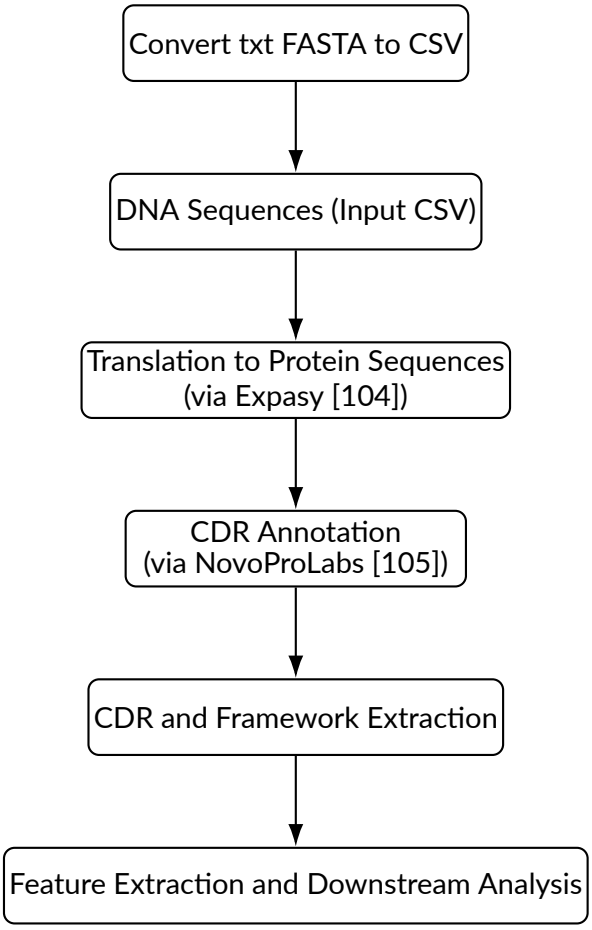


Figure 3.2 Workflow from FASTA conversion to protein translation, CDR annotation, and feature extraction.

As explained above, a subset of the dataset used in this study was provided by Dr. David Saliba's group at the University of Malta. The data consisted of monoclonal antibody DNA sequences in plain text FASTA format. Since downstream annotation tools (e.g. CDR identification, motif discovery) operate on amino acid sequences rather than nucleotide data, DNA sequences had to be converted to protein sequences. Figure 3.2 illustrates the flow taken for the preprocessing of this dataset. Initially, DNA to protein translation was performed manually using SnapGene [106], but to accelerate throughput and ensure reproducibility this step was later automated via our DNA to Protein Translation and Variable Domain Annotation Interface (Section 3.5.2). These FASTA sequences were first converted into a CSV table of DNA entries to accommodate our processing tool, which requires CSV input, duplicates were identified and removed to avoid redundancy. The CSV file then served as input for translation to protein sequences using the Expasy translation tool [104]. Protein sequences were subsequently annotated to identify complementarity determining regions (CDRs) via the NovoProLabs service [105]. The CDRs and their adjacent framework regions were combined into a feature set for subsequent clustering and motif discovery.

3.3 Clustering and Motif Analysis

This section describes our CDRH3 focused pipeline, which comprises two main stages: (1) hierarchical clustering to group sequences into putative clonotypes, and (2) motif extraction to identify conserved and variable residues within each group. Hierarchical clustering was chosen for its ability to capture the gradient like similarity structure typical of antibody repertoires, where sequence relatedness varies continuously rather than forming sharply separated clusters [83, 84]. Unlike methods such as k-means, it does not require pre-specifying the number of clusters, making it well suited for heterogeneous biological data with unknown structure [49, 51]. As discussed in Section 2.4, the resulting dendrograms visually encode pairwise relationships, supporting intuitive interpretation at both local and global levels. To quantify CDRH3 similarity, we computed all against all pairwise comparisons using `BLAST` in `blastp-short` mode [107], which is optimised for short sequences (typically < 15 residues). The resulting blast

e-values were transformed into a distance matrix using the formula:

$$d = -\log_{10}\big(\max(e\text{-value},\ 10^{-300})\big)$$

where $\max(\cdot, 10^{-300})$ caps extremely small e-values at 10^{-300} (to prevent $-\log_{10}(0)$ or numerically infinite distances). The BLAST e-value estimates the number of alignments expected by chance, with lower values indicating stronger similarity. For clustering, we applied a negative $\log_{10}$ transform to convert e-values into a continuous distance metric, capturing both sequence identity and alignment significance. While alternatives such as percent identity or bit scores exist, the log-transformed e-value remains a robust and widely used measure. We then applied Ward's linkage to the distance matrix because it minimizes within-cluster variance, producing compact and interpretable clusters to produce a hierarchical clustering of the sequences, revealing nested repertoire structures. This method is well-suited for revealing nested structures in biological sequence data, and has been effectively used in repertoire clustering and related applications [108]. To evaluate how well the dendrogram preserved the original pairwise distances, we computed the cophenetic correlation coefficient for each clustering run. Values closer to 1 indicate better agreement between the clustering and the underlying distance matrix. We first applied this pipeline to the public OPIG CoV-AbDab dataset to identify dominant public clonotypes. To maintain biological relevance while managing computational complexity, the dataset (349 sequences) was split into antibodies and nanobodies, and clustering was performed separately. For the combined analysis with our University of Malta Sanger derived dataset (137 unique antibody sequences), we used only the OPIG antibody subset to ensure compatibility. This allowed us to assess whether the Sanger clones integrated with or diverged from known public repertoires. In the second stage, we extracted motifs from each cluster by realigning sequences with Clustal Omega (v1.2.2) and computing amino acid frequency matrices. These were visualised as sequence logos using Logomaker (v0.8), highlighting conserved "anchor" residues and variable positions within CDRH3 sequences, features often linked to binding specificity [41, 42]. Frequency based logos were selected for their accessibility and interpretability, avoiding assumptions about background distributions or structural constraints. All processing steps including alignment, clustering, and visualization were implemented in Python (v3.9) using `scikit-learn`, SciPy, and custom wrappers for full

reproducibility.

3.4 Deep Learning Models for Binding Prediction

This section details the deep learning architectures applied to predict SARS-CoV-2 antibody–antigen interactions using only primary amino acid sequences, without incorporating 3D structural data such as PDB coordinates or homology models. In this study, structural data was limited or unavailable for many of the antibodies in the dataset, which precluded the use of structure based methods. While such approaches can offer mechanistic insights, they are constrained by the need for experimentally resolved or reliably modeled structures. In contrast, purely sequence based methods provide scalability and enable rapid analysis of antibody repertoires obtained through high-throughput sequencing. Recent studies have demonstrated the effectiveness of sequence only models, including AbAgIntPre [109] and ProtBERT based classifiers [110].

3.4.1 Data Preprocessing and Redundancy Reduction for Deep Learning Models

Positive and negative antibody–antigen triplets were obtained from the AbAgIntPre repository [109], provided as two separate plain text files: `positive dataset.txt` (binders) and `negative dataset.txt` (non-binders). This dataset was constructed from the CoV-AbDab database maintained by the Oxford Protein Informatics Group (OPIG) [4], as confirmed in the AbAgIntPre publication. Each line in these files consisted of three fields: the antigen name (SARS-CoV1 or SARS-CoV2), followed by the heavy chain antibody sequence and the light chain antibody sequence. This structure can be viewed in the following triplet format:

```
Antigen||Heavy chain antibody||Light chain antibody
```

All antigen entries corresponded to either SARS-CoV1 or SARS-CoV2. Full length spike glycoprotein sequences for these viruses were retrieved from NCBI GenBank (BCN86353.1 [111] for SARS-CoV2 and P59594.1 [112] for SARS-CoV1) and mapped to the triplets by matching the antigen name using BioPython's `SeqIO.parse`. The full dataset comprised 9,309 positive (binder) triplets. 1,965 involving SARS-CoV1 and

7,344 involving SARS-CoV2 and 1,710 negative (non-binder) triplets 996 SARS-CoV1 and 714 SARS-CoV2. Due to this substantial class imbalance, further class balancing steps were applied during preprocessing. To reduce variability and focus on the more clinically relevant SARS-CoV2 response, only triplets involving SARS-CoV2 were retained for model training. This restriction also helped standardise the antigen background across all samples, eliminating antigen specific confounding factors. Preliminary experiments revealed a risk of label leakage, where the model could implicitly associate antigen identity with binding status (e.g., learning to classify SARS-CoV1 pairs as binders and SARS-CoV2 pairs as non-binders), rather than learning true interaction patterns between antibody and antigen sequences. Restricting the dataset to SARS-CoV2 eliminated this bias and ensured that binding predictions were not driven by antigen class imbalance. All sequences were exported into a FASTA file for downstream processing. To remove duplicates and thereby reduce overall sequence redundancy, CD-HIT clustering was applied at 70% identity using the following command:

```
cd-hit -i pairs.fasta -o pairs_nr98.fasta -c 0.7 -n 5 -M 0 -T 4
```

Since everything was being run on Windows OS, this step was performed using the Windows Subsystem for Linux (WSL) environment to ensure compatibility with the Linux based CD-HIT tool. To ensure a balanced classification task, the number of binder and non-binder entries was equalised through undersampling of the majority class which in this case was the binders class.

3.4.2 CKSAAP Feature Representation

To represent antibody and antigen sequences in a machine readable format suitable for classical and deep learning models, we employed the Composition of k-Spaced Amino Acid Pairs (CKSAAP) encoding [113]. CKSAAP computes the frequencies of amino acid pairs separated by k intermediate residues in a sliding window, capturing both local and gapped dipeptide motifs known to be relevant for protein to protein interactions.

Given a sequence of length L, CKSAAP counts all 400 possible amino acid pairs (e.g., AA, AC, ..., YY) at each distance k (typically $k = 0, 1, 2, 3$), resulting in a feature vector of length $400 \times (k + 1)$. The final vector encodes the relative

occurrence of spaced dipeptides, normalized by the number of valid positions at each gap size. This representation captures short and medium range sequence interactions, yields a fixed length vector regardless of input length, and is compatible with classical machine learning and feedforward neural network models making it well-suited for non-sequential tasks in antibody–antigen modeling. CKSAAP has been successfully applied in pior studies such as AbAgIntPre [109] (Section 2.8.2) and remains a strong baseline for representing biochemical sequence data. In our implementation, we generated CKSAAP features for both the antibody and antigen sequences separately, along with their elementwise difference and product, resulting in a combined feature vector that captures both individual and interaction level signals. Each CKSAAP derived feature matrix (antibody and antigen vectors + their elementwise difference and product) was first standardised and then transformed via PCA to retain 95% of the variance. The resulting principal components were used as inputs for every classifier (LR, RF, XGBoost) and for the stacking meta-learner.

3.4.3 Model Architectures

This section outlines the machine learning models implemented in this study, with an emphasis on the rationale behind each choice. The selected models represent a range of complexity from interpretable linear methods to deep learning architectures chosen to evaluate performance trade-offs and feature representation capacity in predicting antibody–antigen binding.

Logistic Regression

As discussed in Section 2.6.2, logistic regression offers a simple and interpretable approach to binary classification [63, 114]. We adopted it as an initial modeling baseline due to its statistical robustness and efficiency in high-dimensional spaces. Each antibody–antigen pair was represented by a CKSAAP-derived feature vector encoding gapped dipeptide frequencies, along with their elementwise difference and product. Model training involved L_2-regularized logistic regression, with regularization strength selected via stratified cross-validated grid search to manage the bias–variance tradeoff [63]. Class weight balancing addressed residual label imbalance to ensure both positive and negative examples contributed meaningfully to optimisation. Despite its

linear decision boundary, logistic regression enables rapid prototyping and provides feature-level interpretability [115], highlighting peptide motifs or interactions associated with binding or inhibition. These properties make it a reliable and interpretable baseline for antibody antigen binding prediction.

Random Forest

As outlined in the background (Section 2.6.2), random forests combine multiple decision trees trained on bootstrapped samples and random feature subsets to improve generalization and reduce overfitting [65]. We applied this method to CKSAAP based antibody antigen features, leveraging its capacity to model non-linear interactions and combinatorial binding patterns that logistic regression cannot capture. Each tree recursively partitions the input space, learning conditional rules such as dipeptide motifs at specific gaps whose predictive value depends on other motif occurrences. This enables the model to uncover higher-order dependencies in binding behavior. Ensemble predictions are obtained by majority vote, while Gini-based feature importance provides a coarse measure of which CKSAAP dimensions contribute most. Model hyperparameters including the number of trees, maximum depth, and minimum samples per leaf were tuned via cross-validated grid search. While interpretability is reduced compared to logistic regression, random forests remain well suited to high-dimensional biological features, offering strong performance with minimal preprocessing. As such, they serve as a powerful non-linear benchmark in our modeling pipeline.

XGBoost

XGBoost was selected for its ability to model complex, non-linear interactions among CKSAAP derived features while maintaining strong regularisation to mitigate overfitting. In this study, each antibody–antigen pair was encoded using CKSAAP vectors for both sequences, along with their elementwise difference and product, as described in Section 3.4.3. The resulting features were used as input to an `XGBClassifier`, with the evaluation metric set to AUC. To optimise performance, hyperparameters were tuned via grid search using five-fold stratified cross-validation on the training set. The search spanned the number of trees (`n_estimators`), maximum tree depth (`max_depth`), and learning rate (`learning_rate`). The best performing model, selected based on

cross-validated ROC AUC, was then used to evaluate held-out test data. This model served both as a standalone classifier and as a base learner in the final stacking ensemble. Unlike a standard random forest which trains each tree independently on bootstrap samples and averages their votes, XGBoost builds its trees sequentially via gradient boosting, fitting each new tree to the residual errors of the ensemble so far. It also includes built in L_1/L_2 regularization on leaf weights, shrinkage (learning rate), and column subsampling, giving tighter control over overfitting and often faster convergence on high-dimensional CKSAAP features [67].

Stacking Ensemble

In our experiments no single classifier consistently outperformed the others across all metrics (ROC AUC, precision, recall), so we turned to stacking to leverage their complementary strengths. A stacking ensemble was constructed using logistic regression, random forest, and the tuned XGBoost model as base learners. Each base model was trained on the same PCA transformed CKSAAP feature set, capturing both antigen and antibody sequence information along with their interaction based encodings. The ensemble combines out-of-fold predictions from the base models obtained via internal cross-validation and passes them as inputs to a logistic regression meta-learner, which learns to optimally weight their outputs for final classification. Logistic regression was chosen as the meta-learner for its simplicity, interpretability, and robustness. As the base learners capture complex non-linear patterns, a linear meta-model reduces overfitting by optimally weighting their calibrated predictions, providing a well-calibrated integration without the added complexity of high-capacity learners like XGBoost.

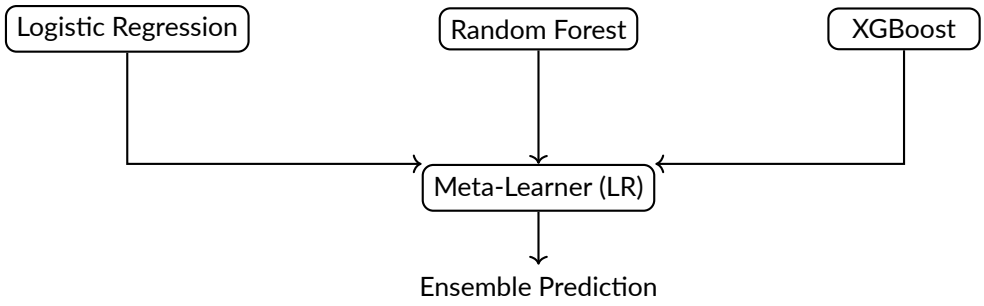


Figure 3.3 Stacking ensemble: three base learners generate out-of-fold probabilities, which a logistic regression meta-learner combines for the final prediction.

Cross-validation during stacking was performed using a five-fold stratified split to ensure consistency with the base learners' evaluation strategy. This architecture, illustrated in Figure 3.3, allows the ensemble to learn complementary decision boundaries from both linear and non-linear classifiers. By leveraging model diversity and combining their strengths, the stacking ensemble aims to produce more robust and generalizable predictions of antibody–antigen binding than any individual model alone.

Stacking was selected due to its theoretical and empirical advantages in combining diverse learning algorithms. It allows the meta-learner to exploit complementary inductive biases from heterogeneous base models, often leading to improved generalisation performance [116, 117]. Specifically, integrating linear models (e.g., logistic regression) with tree-based classifiers (e.g., random forest and XGBoost) enables the ensemble to capture both global additive patterns and complex non-linear interactions among CKSAAP features. This is particularly valuable in biological sequence modeling, where relevant signal may span both simple motifs and conditional dependencies. Stacked ensembles have been successfully applied to a range of bioinformatics problems, including protein–protein interaction prediction and functional annotation [118], reinforcing their suitability for antibody–antigen binding prediction.

Siamese CNN Architecture (AbAgIntPre)

We reimplemented the Siamese CNN architecture from AbAgIntPre [59]. As discussed in Section 2.8.2, to learn a sequence based similarity metric over CKSAAP encoded antibody–antigen pairs. The network comprises three convolutional branches: two with shared weights for antigen and antibody vectors, and a third modeling their elementwise difference and product. Outputs are concatenated and passed to a multilayer perceptron to predict binding logits (Figure 2.7). This design captures both sequence specific and relational features, enabling detection of complementary CKSAAP motifs through convolutional filters of varying kernel sizes. While the AbAgIntPre repository provided the network architecture and encoding pipeline, key training components such as the loss function, optimisation strategy, and regularisation were not documented in this publication. We filled in these gaps by assuming a standard cross-entropy loss for binary classification, using the Adam optimizer with a learning rate of 1e-3, and applying dropout and batch normalization for regularization. A ReduceLROnPlateau scheduler

and early stopping (epochs = 10) were used to stabilise training, and model selection was guided by validation AUC. Hyperparameters were tuned using 5-fold stratified cross-validation.

Bi-LSTM Sequence Model

To explicitly leverage residue order and long-range contextual dependencies, a bidirectional long short-term memory (Bi-LSTM) network was implemented. While handcrafted features such as CKSAAP effectively capture local sequence statistics, they do not account for sequential dynamics or variable-length motifs, including CDR loops. Recurrent neural networks (RNNs), and particularly Bi-LSTMs, are well-suited for modeling such dependencies in biological sequences [70, 119]. In this architecture, the antigen and antibody variable regions were concatenated into a single pseudo-sequence and one-hot encoded across the 20 canonical amino acids. One-hot encoding was selected over CKSAAP in this model to preserve the exact residue order and positional relationships between amino acids—features that are essential for recurrent models to learn context aware representations. Unlike CKSAAP, which abstracts sequences into gapped compositional features and discards sequential order, one-hot encoding retains full positional resolution, making it a natural fit for sequence to sequence models like LSTMs. The encoded sequence was passed through two stacked Bi-LSTM layers with a hidden size of $H = 256$ and a dropout rate of $p = 0.3$. The final hidden states from the forward and backward passes were concatenated and fed into a fully connected layer with dropout, followed by a sigmoid output unit to estimate the binding probability. The network was trained using binary cross-entropy loss and optimized using the AdamW optimizer with a learning rate of 1×10^{-3}. Gradient clipping (threshold = 5.0) was applied to prevent exploding gradients, and early stopping based on validation AUC (patience = 10 epochs) was used to avoid overfitting. This model captures sequence level contextual signals that may underlie antigen–antibody recognition and provides a complementary perspective to fixed-length feature encodings.

ProtBERT Based Classifier

To evaluate the benefits of unsupervised pretraining on large scale protein corpora, a transformer based embedding approach was adopted using ProtBERT [120]. In

contrast to models trained from scratch, such as Bi-LSTMs or CKSAAP-based classifiers, ProtBERT leverages the bidirectional encoder representations from transformers (BERT) framework [121], pretrained on millions of protein sequences to capture contextual residue level information informed by structure, function, and evolution. In this setup, the antibody and antigen sequences were concatenated into a single input string and tokenized using the `Rostlab/prot_bert` tokenizer. Sequences were padded or truncated to a maximum length of 512 residues. The embedding corresponding to the CLS token was extracted from the final hidden layer and passed to a two-layer multilayer perceptron (MLP) classifier head. For fine-tuning, the first six transformer layers were frozen while the remaining layers and the MLP head were trained over five epochs. Optimisation was performed using AdamW (learning rate = 2×10^{-5}, batch size = 16), with a warm-up phase covering 10% of training steps. This approach enables transfer learning from large unlabeled protein datasets and is particularly beneficial in settings with limited training data, offering robust, context aware representations for antigen–antibody interaction prediction.

3.4.4 Training, Validation and Testing Strategy

Models were trained on a stratified 90% training set (1285 samples) and evaluated on the held-out 10% test set (143 samples), with antibody antigen pairs labelled as binding or non-binding (see Section 3.4.1). This split was chosen to maximise training data availability while preserving an independent set for evaluation, given the limited overall dataset size. Stratified five-fold cross-validation was applied exclusively to the training set for hyperparameter tuning and model selection, ensuring balanced class distribution across folds. Model performance was primarily evaluated using the area under the receiver operating characteristic curve (ROC AUC), which remains robust in the presence of class imbalance. Additional metrics including accuracy, precision, recall, F1-score, and confusion matrix were reported to provide a comprehensive view of classifier behavior on the held-out test set. For models sensitive to feature scaling (e.g., logistic regression), features were standardised within the cross-validation folds to prevent data leakage. Deep learning models (Bi-LSTM, ProtBERT, Siamese CNN) were trained on the full training set with a held-out validation subset (10%) used for monitoring early stopping criteria. Early stopping was based on validation AUC, with training terminated

after 10 consecutive epochs without improvement. The classical ML methods (LR, RF) were tuned via grid-search cross-validation and did not use epoch-based early stopping. All experiments were implemented in Python using scikit-learn, PyTorch, and HuggingFace Transformers. Training and evaluation were conducted on Google Colab, utilizing NVIDIA T4 GPU to accelerate deep learning workloads.

3.5 Antibody Sequence Clustering and CDR Annotation Tools

To bridge the gap between advanced computational methods and practical applications in biomedical research, two user friendly graphical user interfaces (GUIs) were developed. These GUIs target biomedical researchers who may not have extensive programming expertise, allowing them to interact with complex data analysis pipelines via an intuitive desktop application. The two main tools are: (1) a Sequence Analysis and Clustering Interface and (2) a DNA to Protein Translation and CDR Annotation Interface.

3.5.1 Sequence Analysis and Clustering Interface

This interface streamlines the process of clustering antibody (or nanobody) sequences and visualizing the results. It integrates multiple steps into a single pipeline:

- **Data Loading:** Users can load one or more CSV files containing antibody or nanobody data (with required columns such as `Clone` and `CDRH3`). The application provides a preview of the loaded data, ensuring that the required format is met.
- **Preprocessing and BLAST Database Creation:** Once the data is loaded, sequences are consolidated and converted into FASTA format. The tool then uses BLAST to construct a searchable protein database from these sequences, facilitating further analysis.
- **Clustering and Dendrogram Generation:** The interface applies hierarchical clustering (via Agglomerative Clustering), as discussed in Section 3.3, to group similar sequences. Users can choose to specify a fixed number of clusters or allow the system to determine clusters automatically by applying a distance cut-off to

the dendrogram at a configurable distance threshold. The resulting dendrogram visually represents the sequence relationships and cluster structure.

- **Sequence Logo Generation:** For each cluster, the interface realigns sequences using Clustal Omega and computes a frequency matrix. This matrix is then used to generate sequence logos with Logomaker, which visually depict residue conservation and variability across the sequences.
- **Interactive Visualisation:** The resulting logos and dendrograms are displayed in an interactive Tkinter window. Users can zoom, pan, and save the generated figures for further analysis or publication.

3.5.2 DNA to Protein Translation and Variable-Domain Annotation Interface

SnapGene [106] (a paid, proprietary tool) requires users to import each clone as a ".dna" file and manually define its translation feature. Only after these individual steps is it possible to use the "Batch Convert File Format" function to process the files together [122]. By contrast, this interface reads a CSV of raw DNA, translates every sequence at once, and annotates all framework and CDR regions in one seamless workflow. This interface converts raw DNA sequences into protein and then partitions the antibody variable domain into all Framework (FR) and CDRs. Key features include:

- **Bulk DNA Input**: Load a single CSV of clone IDs and DNA (columns: `clone_name`, `dna_sequence`). Instant preview and schema-validation.
- **Protein Translation**: Calls the Expasy REST endpoint, parses multi-frame FASTA output, and picks the optimal ORF (Open Reading Frame) for each sequence.
- **Annotation via NovoProLabs**: Automatic submission of each protein to the NovoProLabs CDR tool; returns ANARCI-compliant numbering for FR1–FR4 and CDR1–CDR3.
- **Interactive Results and Export**: View and filter annotated sequences in a sortable table; double click for full per clone details. Export comprehensive CSV (DNA, protein, FR/CDR boundaries and sequences).

- **Open-Source, User-Friendly GUI**: Built in Python using Tkinter, with clear tooltips and context menus.

To facilitate ease of deployment on Windows systems, both interfaces were packaged as standalone desktop applications using PyInstaller. This approach allows users to run the tools without needing to install Python or any external libraries.

3.6 Summary

This chapter outlined the antibody–antigen binding pipeline data acquisition, preprocessing, clustering, motif analysis and predictive modeling and its interactive GUIs, which encapsulate complex bioinformatics steps into intuitive modules for research and education. Such democratisation of computational biology is vital for training future biomedical researchers and fostering innovation in therapeutic antibody development.

4 Results and Discussion

In this chapter, we begin by examining the global structure of antibody repertoires through hierarchical clustering of CDRH3 sequences, highlighting both public SARS-CoV-2 clonotypes and novel sequence families enriched in our Sanger sequencing dataset. We then delve into motif discovery within major CDRH3 clusters to identify conserved paratope signatures. Next, we evaluate the performance of classical and deep learning models in predicting antibody–antigen binding, before concluding with a usability assessment of the graphical interfaces developed for sequence analysis and DNA to protein converter.

4.1 Clustering and Visualisation Results

First, we present the results of clonal family characterization within the OPIG dataset (see Section 3.1.1). Next, we analyze the Sanger scFv sequencing dataset of SARS-CoV-2 Spike binders (Section 3.1.2), identifying distinct clonal families among those clones. Finally, we combine both sets to assess how the Sanger derived clones integrate with or diverge from the public repertoires in the OPIG collection. The CDRH3 is the most diverse element of an antibody's variable domain and plays a central role in antigen recognition [9]. By clustering CDRH3 sequences from both large public databases (CoV-AbDab/OPIG) and our University of Malta Sanger collection, we can identify conserved "public" lineages that recur across individuals and screen for novel "private" families that may represent unique binding modes.

4.1.1 OPIG CoV-AbDab CDRH3-Only Clustering

As described in Section 3.3, we have split the OPIG dataset into two subsets, antibodies and nanobodies. We truncated each dendrogram to the final thirty agglomeration steps (Figure 4.1) and applied a distance cutoff at 5,000 to highlight only the highest-level splits. This cutoff corresponds to a dissimilarity threshold based on $-\log_{10}(\text{e-value})$ scores from pairwise BLAST comparisons, enabling us to group sequences with broadly similar CDRH3 characteristics and thereby identify the major lineage families (clonotypes) present in each repertoire for downstream motif analysis. Under these settings, the

antibody set resolves into four clusters, while the nanobody repertoire also forms four principal groups. By omitting lower level branches, these plots make it easy to compare the dominant lineages in each dataset and choose representative structures for further analysis.

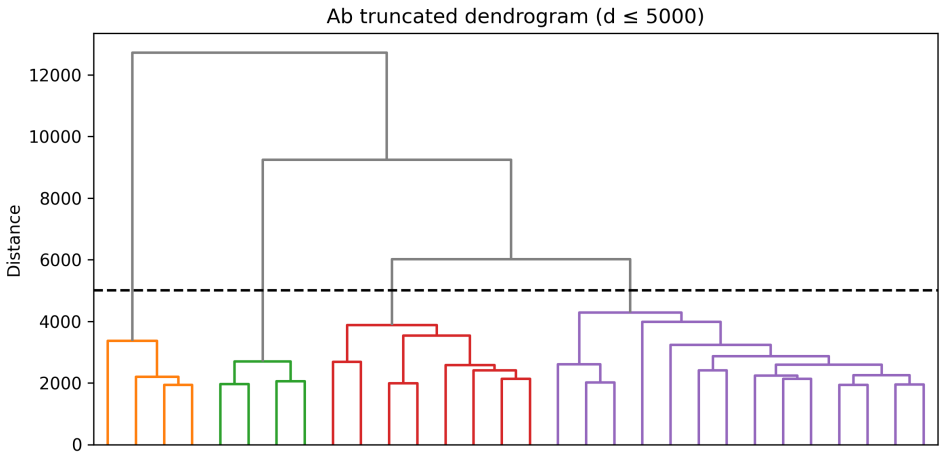


Figure 4.1 Truncated dendrogram showing only the last 30 merges to highlight the main branch structure among antibody (Ab) CDRH3 sequences. The coloured branches represent distinct clonal families identified by hierarchical clustering. The horizontal dashed line indicates the distance cutoff at $d = 5,000$, which was used to define the cluster boundaries shown in colour.

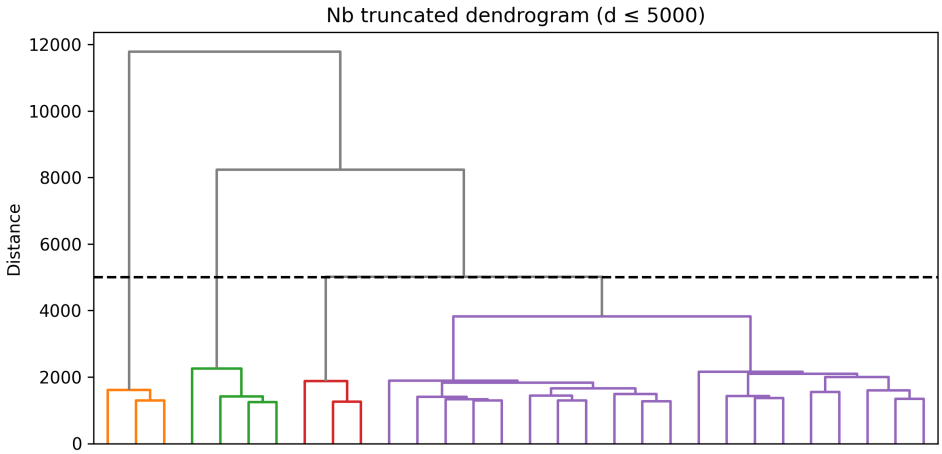


Figure 4.2 Truncated dendrogram of nanobody sequences (last 30 merges). The horizontal dashed line at distance=5000 (labelled) cuts the tree into 4 main clusters.

As shown in Figure 4.3, Cluster 4 dominates with 110 sequences, whereas Clusters 1–3 contain 44, 25, and 24 CDRs, respectively. This pronounced imbalance means that downstream analyses such as motif discovery would be dominated by

the largest group, motivating either a revision of the distance cutoff, cluster specific weighting, or targeted subsampling to ensure equitable representation across clusters.

Figure 4.3 Size of each cluster after cutting the dendrogram at distance 5,000 for Antibodies.

Applying the same 5,000 distance cutoff to nanobody CDRH3 sequences likewise yields four clusters. As Figure 4.4 shows, Cluster 4 is again the largest (94 sequences), while Clusters 1–3 contain 53, 38, and 30 sequences, respectively.

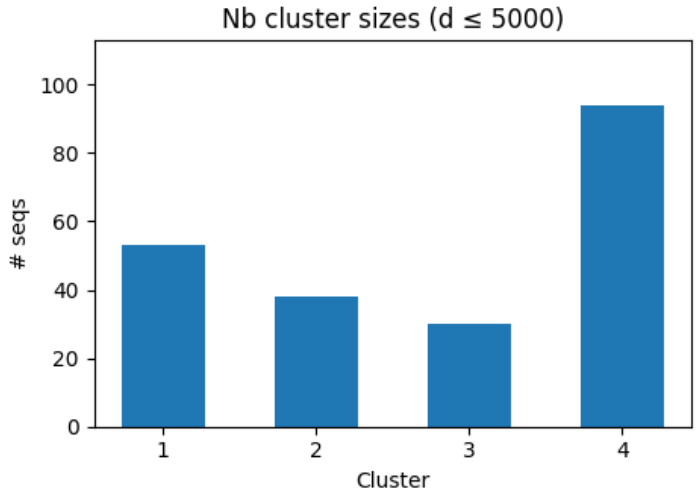


Figure 4.4 Size of each cluster after applying a distance threshold of 5,000 to the Nanobody dendrogram.

We assessed the faithfulness of our Ward-linkage clustering by computing the cophenetic correlation coefficient. For the antibody (Ab) repertoire, we obtained a cophenetic correlation of 0.8223, and for the nanobody (Nb) repertoire, 0.9177.

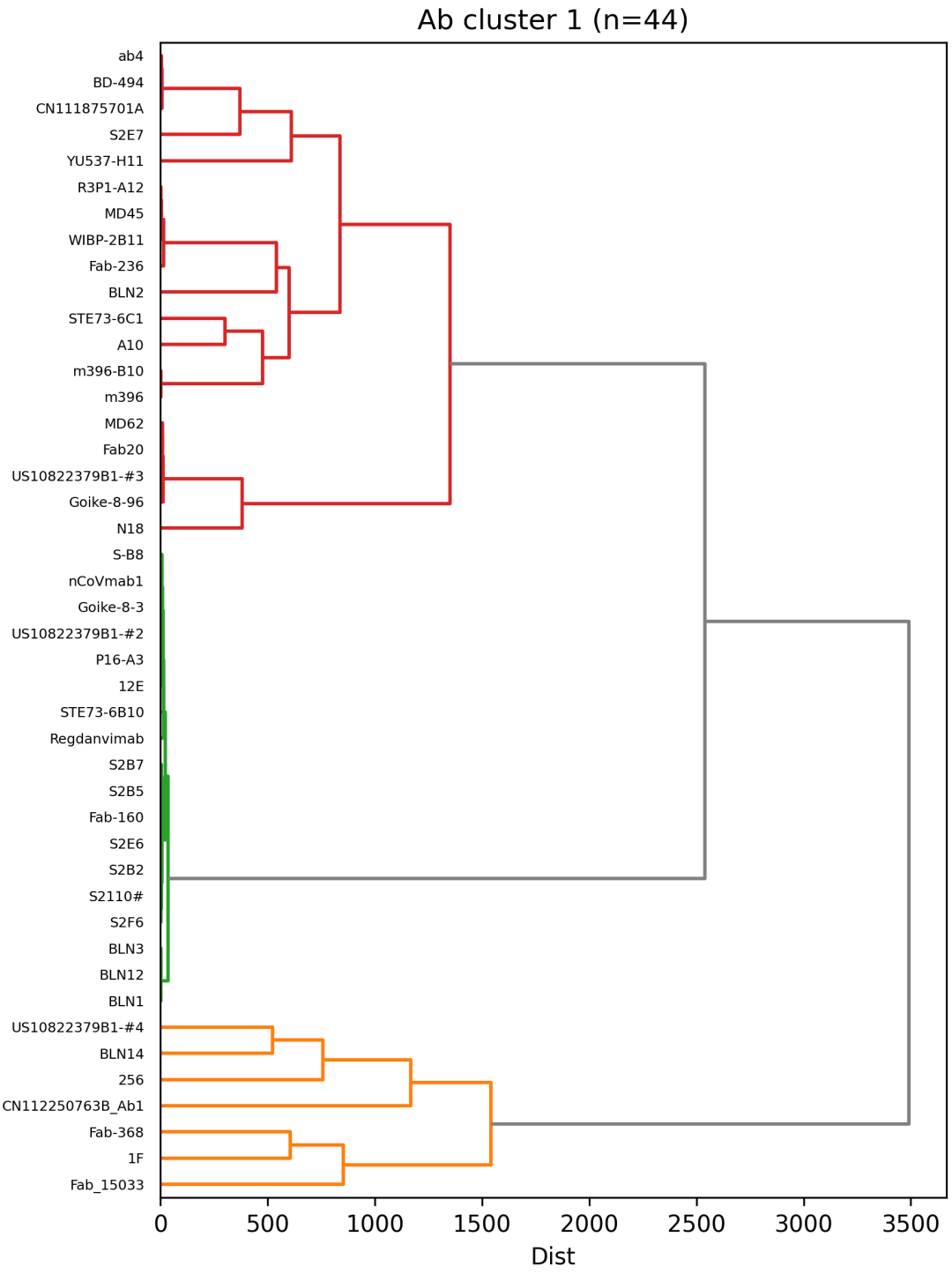


Figure 4.5 Hierarchical clustering dendrogram for antibody Cluster 1 (n=44). Coloured branches indicate distinct subfamilies of closely related antibody sequences. Tight clusters highlight conserved sequence motifs.

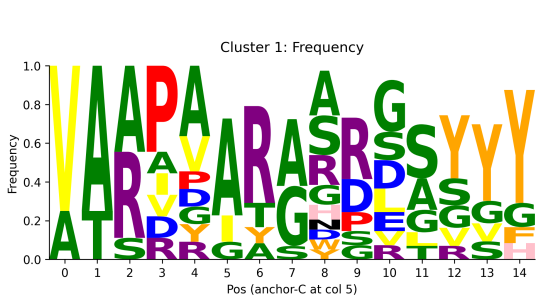
Values above 0.8 are generally taken to indicate very good preservation of the pairwise-distance structure in the dendrogram, and the especially high value for Nbs (0.92) shows that our clusters are a particularly faithful summary of the

underlying similarity relationships. Figure 4.5 shows the full dendrogram for the 44 CDRH3 sequences in Cluster 1, revealing a clear hierarchical structure of closely related clones and distinct subfamilies. Several tight clone pairs are apparent. For example, S2B5 and S2B7 coalesce essentially at zero distance, reflecting their identical "ARARGGSYYYGMDV" motif. Similarly, S2E6 and S2E7 ("ARAHRGSYYYGMDV") form another near identical pair. These two pairs then merge below a distance of 200, defining a cohesive "S2B/E" mini family (green branch). On the red branch, A10 merges with m396-B10 at a distance of ≈50, and Fab-236 with BLN2 in a similarly tight fashion.

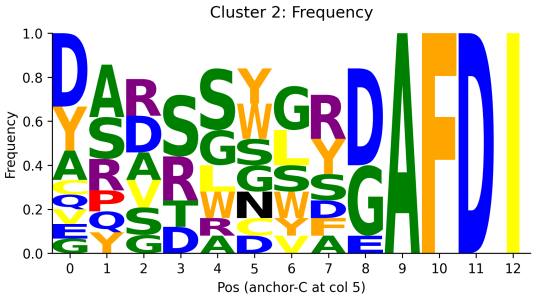
Together with additional members, these sequences merge below 500, forming a robust red sub-lineage. In the orange branch, Fab-368 and 1F cluster at ≈600, while BLN14 and 256 merge at ≈800. These four then coalesce by ≈1100, indicating another well defined lineage within Cluster 1. This entire orange sub-cluster remains distinct from the red and green branches until a much higher distance (≈3500), at which point all major sub-lineages are unified. In summary, Cluster 1 is characterised by tight sequence families with well defined subfamily structure, and the main sub-clusters remain separated until high linkage distances, indicating substantial sequence diversity within this set.

For deeper inspection, Appendix B provides the full dendrograms for each cluster (Figures B.2–B.9) alongside a comprehensive listing of all CDRH3 sequences by cluster (Tables B.1 and B.2). We also generated sequence logo motifs to quickly show which amino acids are always conserved and which vary highlighting key binding positions and giving us a representative sequence to study further. Below is a summary of the eight major CDRH3 clusters, highlighting core motifs, representative antibodies and nanobodies.

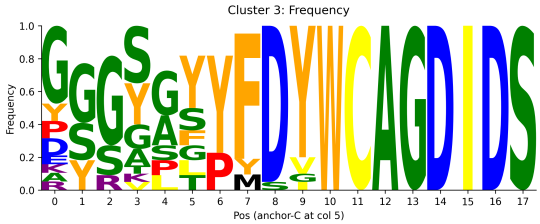
Antibodies Motifs



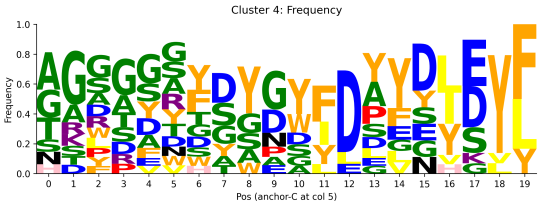
(a) Antibody Cluster 1 Motif (n=44)



(b) Antibody Cluster 2 Motif (n=25)



(c) Antibody Cluster 3 Motif (n=24)



(d) Antibody Cluster 4 Motif (n=119)

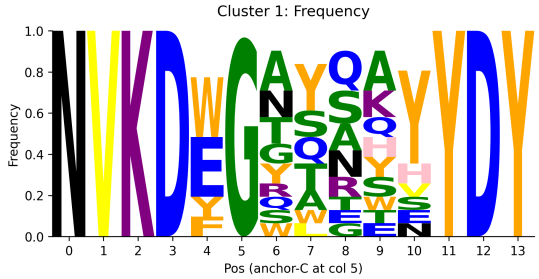
Figure 4.6 Frequency motifs for OPIG antibody Clusters 1–4. (a) Cluster 1; (b) Cluster 2; (c) Cluster 3; (d) Cluster 4.

Figure 4.6 shows the generated position wise amino acid frequencies for each antibody cluster, visualized as sequence logos. Cluster 1 exhibits low variability at the start of the sequence, with most sequences beginning with the same two residue prefix. The middle segment shows moderate diversity, and the final position is dominated by a single amino acid. Cluster 2 consists of uniformly short sequences that share a highly consistent start and end, with only limited variation in the middle positions. Cluster 3 contains longer sequences (17–18 residues) with a consistent prefix, a recurring core pattern in the middle, and a fully conserved multi residue suffix at the end. Cluster 4 is the most diverse overall, but several positions still show strong preferences, particularly early in the sequence, at a central anchor point, and near the end.

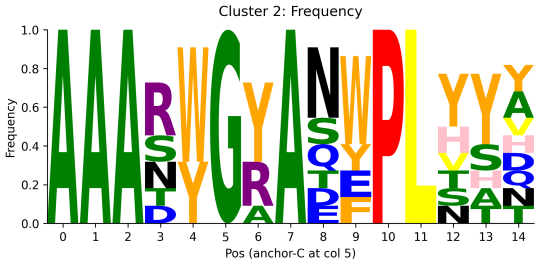
Nanobodies Motifs

Figure 4.7 displays position wise amino acid frequencies for each nanobody cluster as sequence logos. Cluster 1 shows high conservation in the first few positions, with four positions (1–4) nearly fixed, followed by a more variable middle segment and a consistent ending pattern dominated by three residues. Cluster 2 is also highly

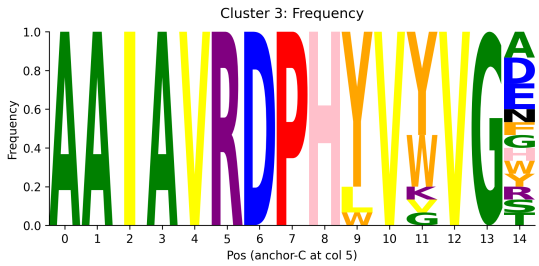
conserved, particularly at the start and around positions 5–11, where multiple positions show low variability.



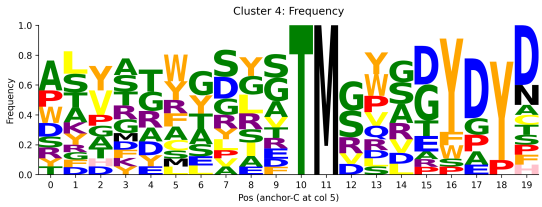
(a) Nanobody Cluster 1 Motif (n=53)



(b) Nanobody Cluster 2 Motif (n=38)



(c) Nanobody Cluster 3 Motif (n=30)



(d) Nanobody Cluster 4 Motif (n=94)

Figure 4.7 Motifs for OPIG nanobody clusters 1–4. (a) Cluster 1; (b) Cluster 2; (c) Cluster 3; (d) Cluster 4.

Cluster 3 displays low sequence variability, with nearly all positions from 0 through 10 fully conserved across the cluster. This includes a fixed 11 residue prefix common to nearly all sequences. Some minor variation begins to appear at positions 11–14, but even there, a few amino acids are clearly dominant. The overall motif is highly consistent, making this one of the most conserved clusters in the set. In contrast, Cluster 4 is more variable overall, but includes two highly conserved positions at 10 and 11, along with several other positions showing moderate preferences across the sequence.

4.1.2 Sanger Derived CDRH3 Clustering

Following the OPIG dataset results, we applied Ward's method to the pairwise $-\log_{10}(\text{e-value})$ distances for the 137 University of Malta Sanger CDRH3 sequences. Applying a threshold of $d = 1,500$ (Fig. C.1) to the resulting dendrogram yielded eight clusters, one large cluster and seven much smaller groups. The clustering quality is supported by a cophenetic correlation of 0.8557. For complete dendrograms of each

cluster please refer to Appendix C.

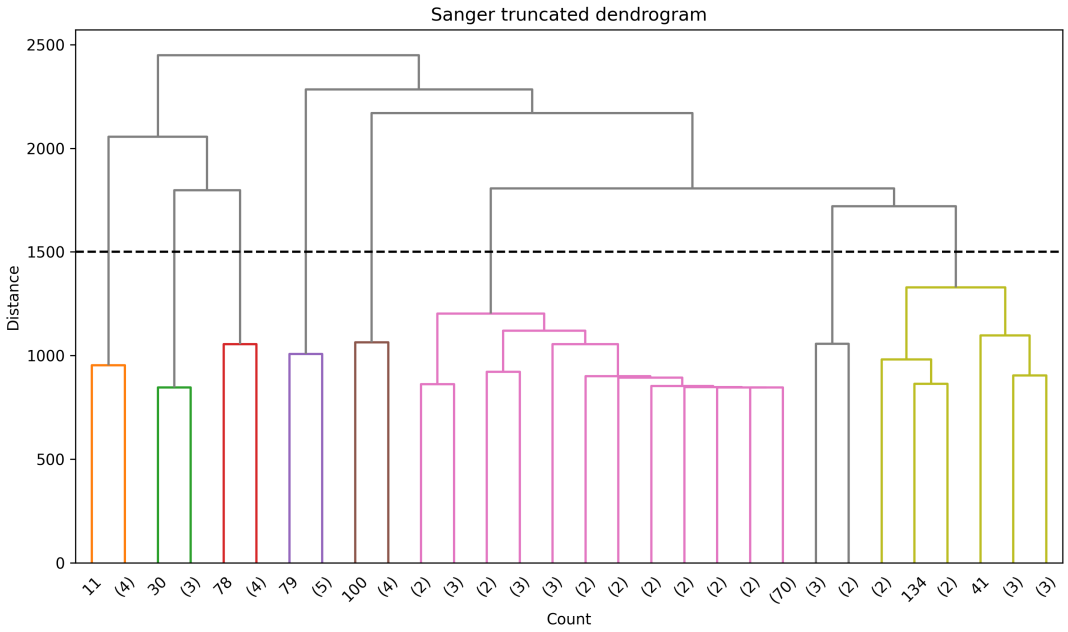


Figure 4.8 Dendrogram of 137 Sanger CDRH3 sequences

Figure 4.9 shows the relative sizes of the eight clusters. One cluster contains 95 CDRH3s (≈70 % of the dataset), while the remaining four contain only 4–10 CDRH3s each.

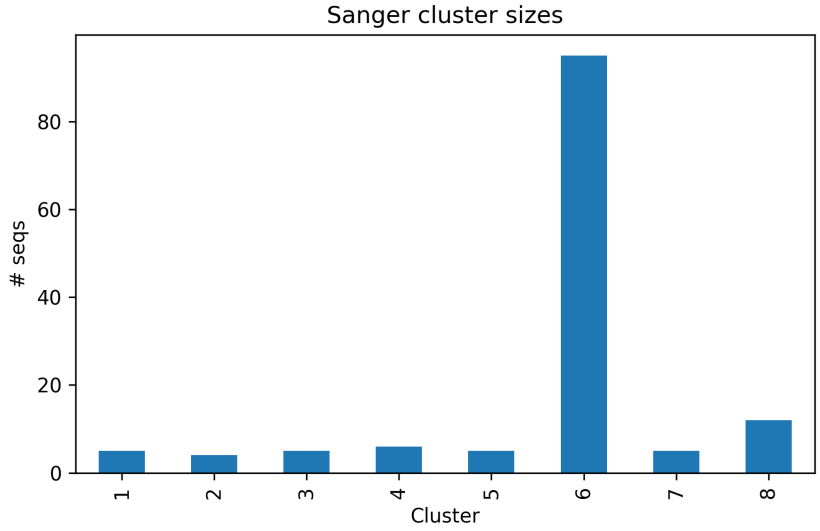
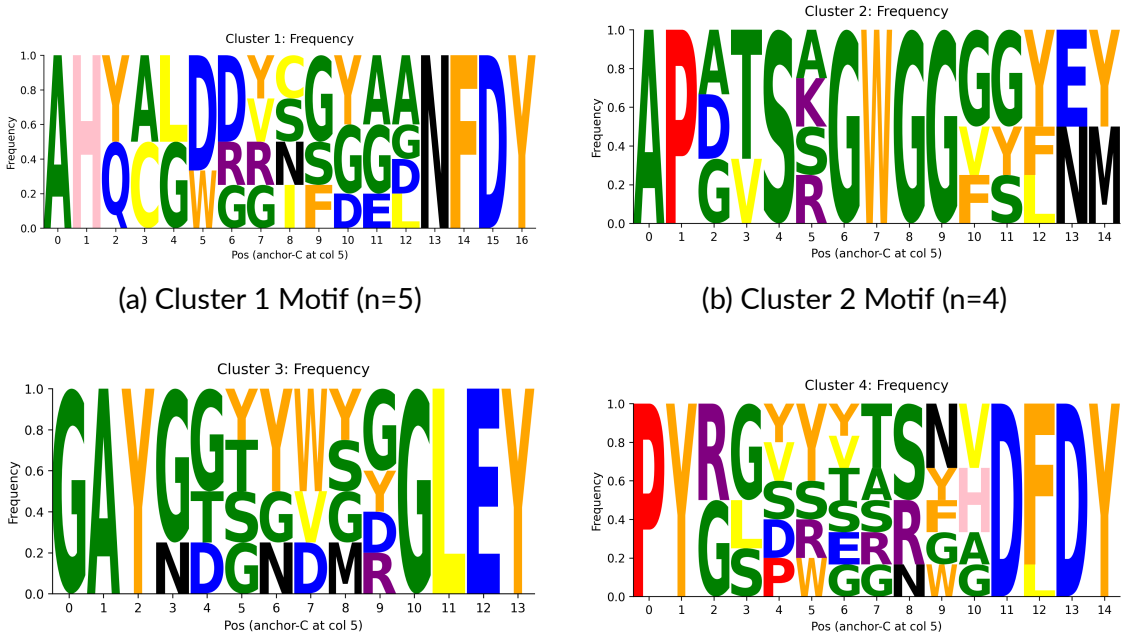


Figure 4.9 Size distribution of the eight Sanger CDRH3 clusters.

Motif analysis per Cluster



(a) Cluster 1 Motif (n=5)

(b) Cluster 2 Motif (n=4)

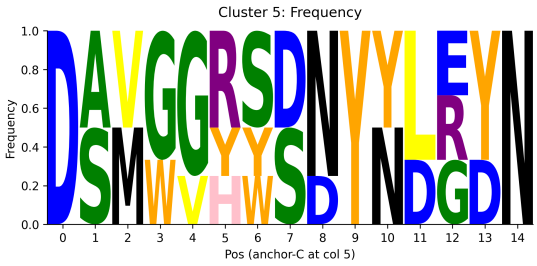
(c) Cluster 3 Motif (n=5)

(d) Cluster 4 Motif (n=6)

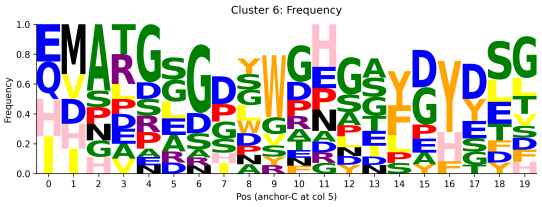
Figure 4.10 Frequency motifs for Sanger clusters 1–4. (a) Cluster 1; (b) Cluster 2; (c) Cluster 3; (d) Cluster 4.

Figure 4.10 displays position wise amino acid frequency distributions for the first four sequence clusters of the sanger sequencing dataset. Cluster 1 begins with a conserved prefix across the first three positions, where a small set of residues dominate mainly A and H. Variability increases in the middle region (positions 4–12), with several positions showing a broader distribution of characters. Toward the end, the sequence converges again into a consistent suffix. Cluster 2 begins with a partially conserved prefix across positions 0–5, where a few residues dominate but some variation remains. The middle region (positions 6–9) is highly conserved, with specific residues occurring nearly uniformly across all sequences. This is followed by a moderately stable segment (positions 10–12), and the final two positions (13–14) show slightly increased variability while still favoring a small set of characters. Cluster 3 exhibits a well defined structure with a highly consistent prefix (positions 0–5) and a stable four residue suffix (positions 10–13). The central region (positions 4–9) shows the most variation, with a broader distribution of characters across sequences. Overall, the motif forms a conserved start and end, connected by a moderately flexible central segment. Cluster 4 starts with a relatively conserved prefix across the first few positions. This is followed by a

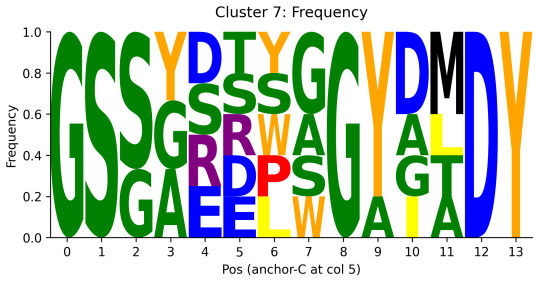
diverse central region with multiple residues appearing at each site. Toward the end of the sequence, the motif becomes increasingly consistent, with the final four positions forming a well defined pattern shared across the cluster.



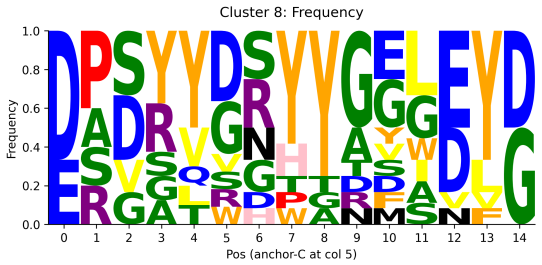
(a) Cluster 5 Motif (n=5)



(b) Cluster 6 Motif (n=95)



(c) Cluster 7 Motif (n=5)



(d) Cluster 8 Motif (n=12)

Figure 4.11 Frequency motifs for Sanger clusters 5–8. (a) Cluster 5; (b) Cluster 6; (c) Cluster 7; (d) Cluster 8.

Cluster 5 begins with a low variability prefix, followed by a moderately consistent central region where a few residues dominate. Positions 9–11 form a particularly conserved core shared across nearly all sequences. Toward the end of the sequence, variability increases, with several positions showing broader distributions of amino acids. Cluster 6 displays the highest overall sequence variability among all clusters analyzed in this section. While the first few positions show some repeated patterns, the central region (positions 6–13) is highly diverse, with no consistent sequence pattern. Toward the end of the sequence, variability remains high but a few positions begin to show modest convergence across the cluster. This broad distribution suggests a loosely defined or highly heterogeneous motif structure. Cluster 7 begins with a highly conserved prefix, followed by a semi-stable segment that introduces moderate variability across a few positions. The middle of the sequence is more diverse but still shows some repeated patterns. The final portion of the sequence exhibits strong convergence, particularly at the last two positions, resulting in a distinct and well structured ending

motif. Cluster 8 starts with a conserved prefix dominated by D and E residues, followed by a strongly conserved short segment in the middle. This leads into a more diverse region (positions 7–10), where sequence variation increases. Toward the end, the motif becomes increasingly stable, with a clearly recurring multi residue suffix shared across most sequences. The result is a structured motif with alternating regions of low and high variability.

4.1.3 Clustering on OPIG and Sanger Derived CDRH3 Sequences

To see how the 137 University of Malta Sanger derived CDRH3 sequences fit into the public SARS-CoV-2 repertoire, we merged them with all CDRH3 sequences from CoV-AbDab and reran hierarchical clustering (Ward linkage on the $-\log_{10}$ e-value distance) using a cutoff of $d = 5000$. For consistency, we limited the OPIG dataset to antibody sequences only, since the University of Malta data consists only of scFVs.

Figure D.1 shows the entire dendrogram; Figure 4.12 illustrates the clustering produced by truncating the tree at height 5,000, yielding four clusters. As visualized in Figure 4.13, the distribution is highly imbalanced, with Cluster 4 encompassing the majority of sequences ($n \approx 240$), while the remaining clusters are much smaller ($n = 25$–50). This pattern is consistent with the OPIG only analysis, where a small number of dominant public families coexisted with more diverse, lower frequency groups.

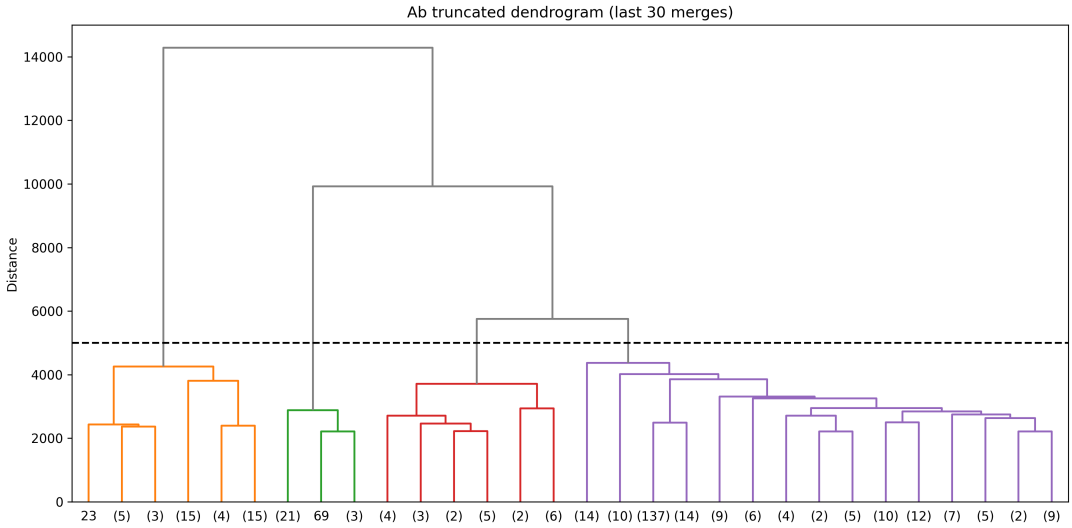


Figure 4.12 Truncated dendrogram (last 30 merges) for combined data

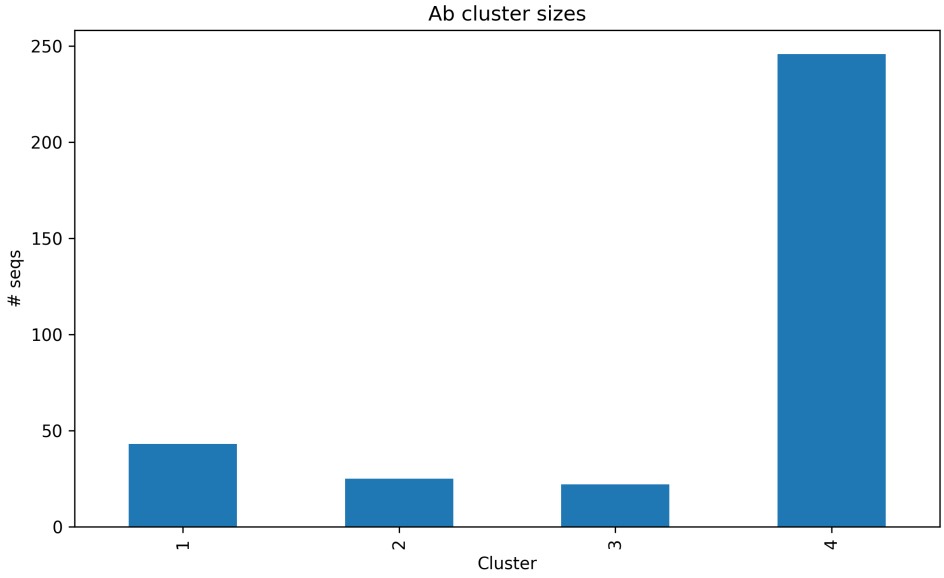
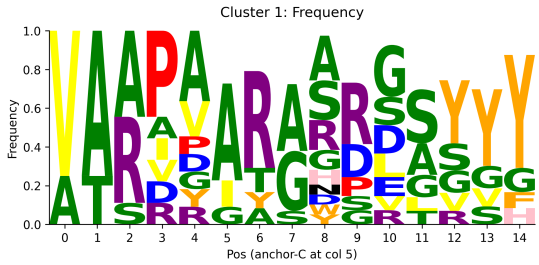
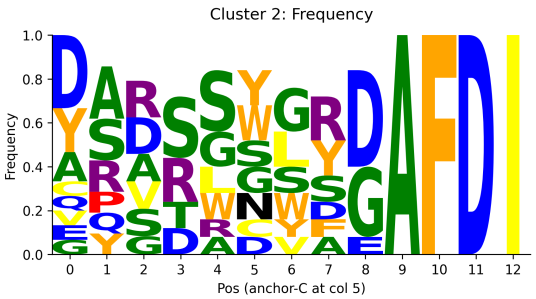


Figure 4.13 Cluster size distribution for the combined antibody dataset

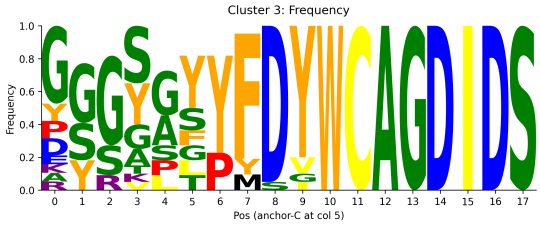
For complete dendrograms of Clusters 1–4, see Appendix D. To assess the impact of incorporating Sanger derived sequences, we regenerated the antibody cluster motifs using the combined OPIG and Sanger dataset. Sequence logos and consensus motifs are important because they reveal conserved residues within each cluster, which may correspond to shared antigen binding properties or structural roles. By comparing these motifs across datasets (OPIG-only vs. combined with Sanger), we can assess whether new sequences reinforce known public antibody patterns or introduce novel diversity. The preservation of key motifs in the combined dataset supports the validity of the clustering and suggests that Sanger derived antibodies belong to the same dominant immune lineages.As shown in Figure 4.14, the overall motif structures remain consistent across all four clusters when compared to the OPIG only version (Figure 4.6). The conserved regions, residue preferences, and overall sequence variability patterns are preserved, indicating that the additional sequences reinforce existing trends without introducing substantial divergence. This suggests that the Sanger sequences are well aligned with the patterns already captured in the OPIG repertoire and do not substantially alter cluster level motif profiles.



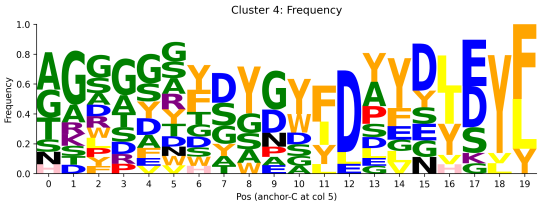
(a) Cluster 1 Motif



(b) Cluster 2 Motif



(c) Cluster 3 Motif



(d) Cluster 4 Motif

Figure 4.14 Frequency-information motifs for the combined OPIG antibody dataset, clusters 1–4. (a) Motif 1; (b) Motif 2; (c) Motif 3; (d) Motif 4.

Given the large size of Cluster 4, we decided to perform a secondary clustering to identify internal substructure. This cluster was further divided into 11 subclusters, which are visualized in Appendix D.

4.1.4 Discussion

In this analysis, we aimed to assess how our University of Malta Sanger derived antibody sequences align with established public CDRH3 lineages by integrating them into the OPIG CoV-AbDab dataset and replicating the clustering and motif extraction pipeline. The inclusion of 137 Sanger derived sequences did not significantly alter the structure of the dominant clusters or their characteristic sequence motifs. As depicted in Figure 4.14, the motif patterns across the combined dataset closely mirror those from the OPIG only analysis (Figure 4.6). Key features of each cluster such as conserved prefixes, stable cores, and repeated suffixes remain consistent.

This consistency suggests that the Sanger derived sequences are broadly representative of the public SARS-CoV-2 response repertoire captured in the OPIG dataset. Notably, the Sanger sequences reinforce dominant sequence trends rather than introducing divergent motifs or novel cluster structures. This observation aligns with the

understanding that Sanger sequencing, despite its lower throughput compared to Next Generation Sequencing (NGS), provides high fidelity, full length antibody sequences that are valuable for repertoire analyses. Moreover, the size distributions of the clusters remain similar before and after integration, with a single dominant cluster representing the majority of sequences and several smaller, distinct groups comprising the remainder. This reinforces the robustness of the high level CDRH3 cluster structure and indicates that the public antibody landscape remains stable even as new sequences are introduced.

Although this section does not directly assess antigen binding, the presence of conserved sequence motifs particularly in clusters with tightly defined start and end segments, may reflect structural or functional constraints relevant to binding. Prior research has shown that recurring CDRH3 patterns often correspond to public or convergent antibody responses, especially in the context of viral infection [123]. In this light, several Sanger only clusters that exhibit strong internal conservation could represent antibodies targeting similar or shared epitopes.

While direct comparisons between Sanger derived repertoires and OPIG based clustering are limited in the literature, the broader field supports integrating heterogeneous sequencing sources for comprehensive analysis. For example, the Observed Antibody Space (OAS) database compiles antibody sequences from both Sanger and NGS (Next generation sequencing) workflows, demonstrating the feasibility and utility of unified repertoire analysis across sequencing platforms [124]. Our findings contribute to this broader effort by showing that small scale, high quality datasets like ours can align closely with large public repertoires, both structurally and compositionally.

4.2 Results of Machine Learning Models for Binding Prediction

In this section, we present the results of our sequence based binary classifiers for predicting SARS-CoV-2 antibody–antigen binding. We compare seven architectures of increasing complexity logistic regression, random forest, XGBoost, a Bi-LSTM recurrent model, a ProtBERT transformer, a Siamese CNN (AbAgIntPre), and a stacking ensemble on a held out test set of positive (binding) and negative (non-binding) pairs. By

comparing models from basic linear methods up to advanced deep and metric learning architectures, we aim to pinpoint the most accurate yet practical approach for predicting antibody–antigen binding. For each model we report ROC AUC, accuracy, precision, recall, and F1-score, and we analyse both ROC curves and confusion matrices to characterise each method's sensitivity and specificity trade-offs.

4.2.1 Hyperparameter Optimisation

Prior to training, we balanced the binder and non-binder classes by random undersampling of the majority class to ensure a 1:1 ratio. We then performed an extensive hyperparameter search for each model using grid search (or 5-fold cross-validation for tree based and ensemble methods) on the training set. We applied early stopping for iterative learners (XGBoost, Bi-LSTM, Siamese CNN) to prevent overfitting. The final selected values, shown in Table E.1, reflect the best performing settings under these criteria.

4.2.2 Overall Test Set Performance

Table 4.1 summarises each model's test set performance (after five fold stratified cross validation and grid-search tuning; see Table E.1 and Section 3.4). By ROC AUC, all methods comfortably exceed random-chance (0.5), but only the stacking ensemble pushes past 0.70, reaching 0.711 with accuracy of 0.643. Generally, an AUC of 0.70–0.80 is considered "acceptable" discrimination in biomedical settings, so our ensemble sits at the lower end of that range [125].

Table 4.1 Test set performance of all seven models (mean $\pm$ standard deviation)

Model	ROC AUC	Acc.	Precision	Recall	F1
Logistic Regression	0.653 ± 0.02	$\mathbf{0.643 \pm 0.02}$	$\mathbf{0.644 \pm 0.02}$	$\mathbf{0.643 \pm 0.05}$	0.628 ± 0.03
Random Forest	0.617 ± 0.03	0.594 ± 0.03	0.597 ± 0.03	0.594 ± 0.05	0.554 ± 0.04
XGBoost	0.690 ± 0.01	0.616 ± 0.02	0.616 ± 0.02	0.616 ± 0.03	0.621 ± 0.03
Stacking Ensemble (LR, XGBoost, RF)	$\mathbf{0.711 \pm 0.02}$	$\mathbf{0.643 \pm 0.01}$	0.643 ± 0.02	$\mathbf{0.643 \pm 0.01}$	$\mathbf{0.648 \pm 0.01}$
Bi-LSTM	0.619 ± 0.02	0.607 ± 0.01	0.605 ± 0.01	0.607 ± 0.02	0.597 ± 0.03
ProtBERT	0.664 ± 0.03	0.629 ± 0.02	0.630 ± 0.01	0.629 ± 0.01	0.629 ± 0.02
AbAgIntPre (Siamese)	0.694 ± 0.01	0.585 ± 0.01	0.623 ± 0.03	0.585 ± 0.01	0.587 ± 0.02

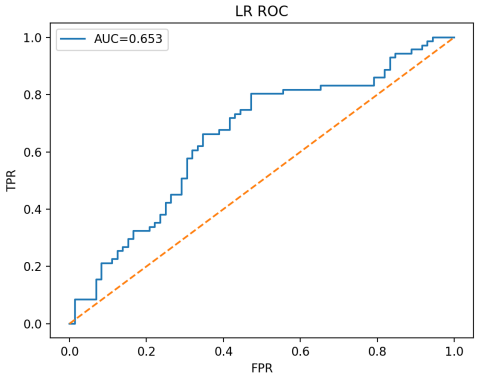
Among classical baselines, logistic regression (AUC 0.653) surprisingly outperforms the more flexible random forest (0.617), suggesting that linear k-mer [1] features capture

[1] A k-*mer* is a contiguous subsequence of length k extracted from a longer biological sequence; see [126] for further details.

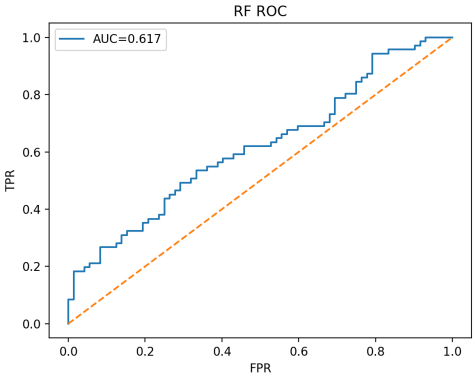
much of the binding signal without overfitting. This outcome can also be attributed to the relatively modest dataset size, which constrains the effective capacity of more complex, high-variance models such as random forests. In such cases, simpler linear models like logistic regression can generalise better, especially when the dominant relationships in the k-mer feature space are approximately additive.

XGBoost (0.690) narrows the gap to the ensemble, demonstrating that gradient boosting of k-mer counts effectively models non-linear interactions. Deep learning models show a mixed picture. The Bi-LSTM (0.619) underperforms most methods, likely due to limited training data for sequential architectures. ProtBERT (0.664) improves on logistic regression and Bi-LSTM, but still falls short of the stacking ensemble, indicating that pretrained amino-acid embeddings add useful context but cannot fully replace task specific feature engineering. Our Siamese CNN (AbAgIntPre) achieves AUC 0.694 comparable to XGBoost, highlighting that a pairwise contrastive architecture can nearly match tree based methods while offering greater specificity (fewer false positives). Precision, recall, and F1-scores echo these trends: the stacking ensemble leads on F1 (0.648), logistic regression maintains balanced precision/recall around 0.64, and random forest trails markedly (F1 0.554). Overall, while no model achieves “excellent” discrimination (ROC AUC > 0.80), our results demonstrate that simple linear and tree based approaches, when combined, outperform standalone deep learners on this moderate sized (1285 samples for training and 143 for testing) SARS-CoV-2 antibody antigen dataset.

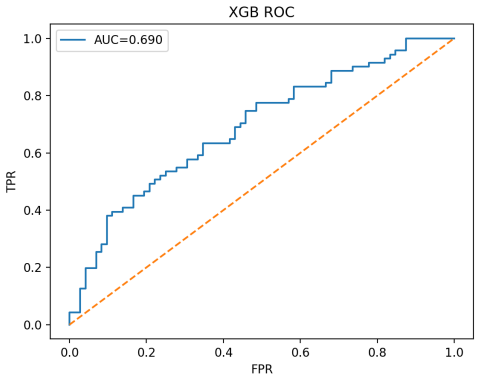
4.2.3 Receiver Operating Characteristic Analysis



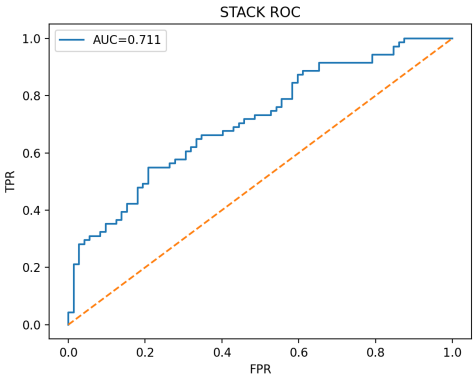
(a) Logistic Regression



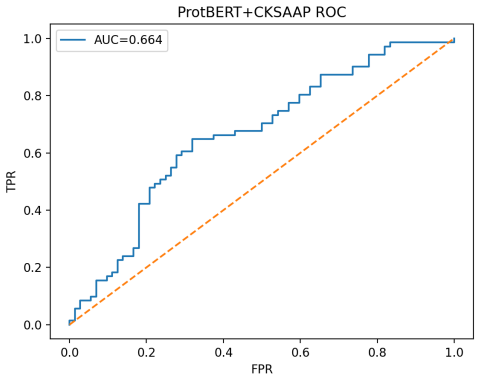
(b) Random Forest



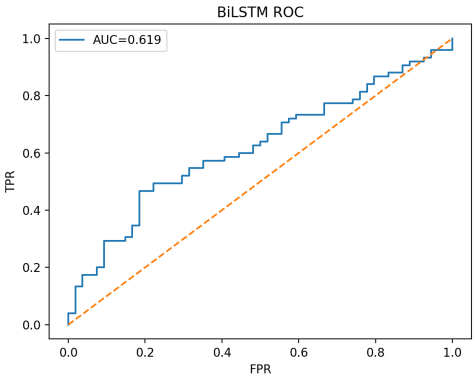
(c) XGBoost



(d) Stacked Ensemble



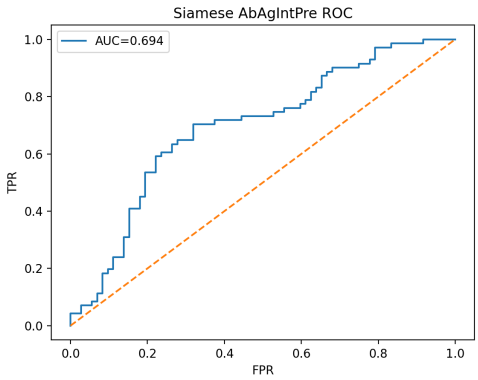
(e) ProtBERT+CKSAAP



(f) BiLSTM

Figure 4.15 Receiver operating characteristic curves for all models.

Figure 4.15 presents the Receiver Operating Characteristic (ROC) curves for all seven models on the SARS-CoV-2 binding prediction task. All models exceed the random baseline (AUC = 0.5), confirming that they perform better than random. However, the



(g) Siamese Network

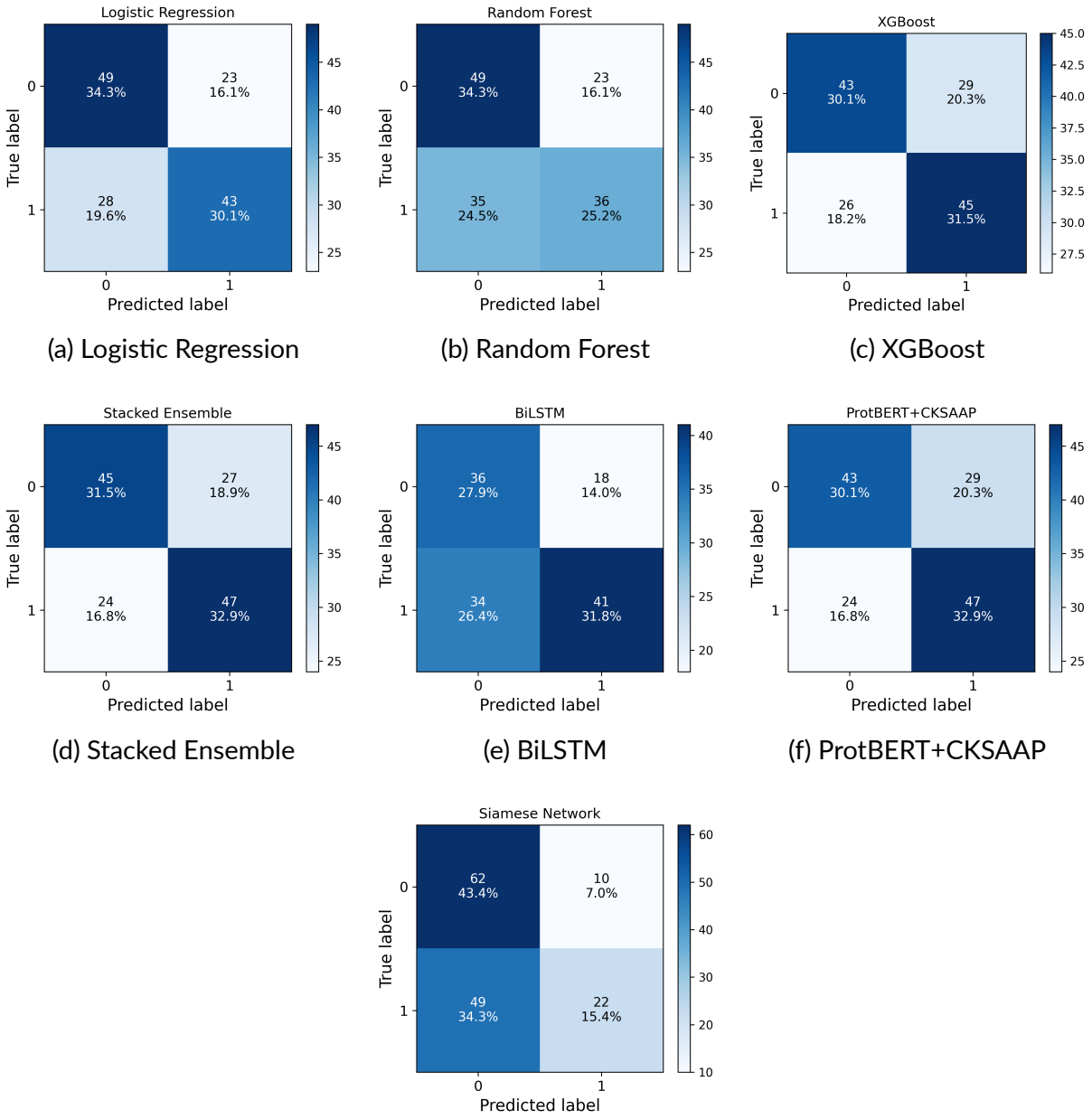
Figure 4.15 (continued) Receiver operating characteristic curve for Siamese AbAgIntPre model

overall separation between positive and negative classes remains modest across the board. Among classical approaches, Logistic Regression (Fig.4.15a) shows the most stable early rise among the baselines, reaching an AUC of 0.653, though the curve remains relatively shallow and irregular in places. Random Forest (Fig.4.15b) performs the worst overall (AUC = 0.617), with a gradual slope that stays close to the diagonal. XGBoost (Fig. 4.15c) performs better, achieving an AUC of 0.690 with a smoother and more confident curve shape, particularly in the low FPR region. The Stacked Ensemble (Fig. 4.15d) attains the highest AUC of 0.711. While this does not reflect strong discrimination in absolute terms, it suggests that model ensembling is the most effective strategy within this dataset's constraints, especially given the limited size and complexity of available features. Deep learning models demonstrate limited gains. ProtBERT with CKSAAP (Fig.4.15e) reaches AUC 0.664, higher than baseline models but still moderate while Bi-LSTM (Fig.4.15f) shows the weakest curve among neural models (AUC = 0.619), likely due to overfitting and insufficient training data for sequential modeling. The Siamese network (AbAgIntPre) (Fig. 4.15) achieves AUC 0.694. While still below the ensemble and XGBoost, its ROC curve rises sharply at low FPRs (False Positive Rates), suggesting some utility in scenarios where false positives are especially costly. That said, the overall performance remains limited.

Overall, our best sequence only model (stacking ensemble) achieves ROC AUC ≈0.71. Other sequence based methods report higher AUCs (e.g., AbAgIntPre ≈0.82[109]), and structure informed paratope predictors like Parapred reach ≈0.88 ±

0.004 [127]. This indicates that incorporating 3D or paratope context features or substantially more training data will be needed to close the performance gap.

4.2.4 Confusion Matrices Analysis



(a) Logistic Regression (b) Random Forest (c) XGBoost

(d) Stacked Ensemble (e) BiLSTM (f) ProtBERT+CKSAAP

(g) Siamese Network

Figure 4.16 Confusion matrices for all seven models evaluated on the SARS-CoV-2 binding prediction task using a 0.5 decision threshold. Each figure shows the distribution of predicted vs. true labels. Darker colours show higher counts in each cell.

To better understand how each model balances sensitivity and specificity in practice, we present confusion matrices that visualise the distribution of true and false predictions.

This allows for a more interpretable, example level comparison of classification behavior beyond overall metrics like ROC AUC. Figure 4.16 shows the confusion matrices for all seven models evaluated on the SARS-CoV-2 binding prediction task, using a fixed decision threshold of 0.5. Each matrix displays true positives (TP) and true negatives (TN) along the diagonal, while false positives (FP) and false negatives (FN) appear off diagonal. A notable trade-off is observed in the Siamese CNN (panel g), which demonstrates the fewest false positives (10) but a relatively large number of false negatives (49). This yields a high specificity of 0.86 but a low sensitivity of 0.31, indicating a conservative decision strategy that favors avoiding false alarms at the cost of more missed binders—consistent with its ROC curve behavior.

In contrast, the stacking ensemble (panel d) achieves a more balanced error distribution, with 24 false negatives and 27 false positives. This mirrors its accuracy of 0.643 and supports its strong F1-score of 0.648, reinforcing the ensemble's advantage in maintaining even performance across classes. Among classical learners, logistic regression (panel a) exhibits a mild bias toward false negatives (28 vs. 23), while XGBoost (panel c) shifts this pattern slightly, showing more false positives (29) than false negatives (26). These complementary tendencies help explain the ensemble's improved results when combining both models.

Deep learning models reveal varied patterns. Bi-LSTM (panel e) misclassifies a larger proportion of positive samples, with 34 false negatives and 18 false positives, resulting in moderate sensitivity (0.55) and specificity (0.67). ProtBERT combined with CKSAAP features (panel f) improves sensitivity to 0.66 by reducing false negatives to 24 but increases false positives to 29, slightly lowering specificity to 0.60. Overall, the confusion matrices confirm the trends seen in AUC and F1 scores, highlighting that no model achieves both high sensitivity and specificity, though the stack ensemble achieves the best trade-off.

4.2.5 Sanger Binders Validation Results

To assess the generalisability of our predictive models beyond the CoV-AbDab test set, we applied the best performing classifier the stacking ensemble to the in-house Sanger dataset. Although sequences were preprocessed to match the expected input format, the model performed poorly. Specifically, the receiver operating characteristic (ROC)

area under the curve (AUC) was approximately 0.50, suggesting no better than random discrimination between presumed binders and non-binders. While this indicates limited generalisation, it provides a valuable lesson about the effects of domain shift and the importance of diverse training data. Several factors may contribute to this outcome:

- Differences in sample origin, source diversity, and sequencing context may introduce a domain shift between the OPIG dataset (used for training) and the in-house repertoire.
- The limited size of the training dataset may have caused overfitting to CoV-AbDab specific motifs, reducing generalisability to unrelated sequences.

This result underscores the importance of external validation and highlights the challenges of domain adaptation in antibody–antigen prediction. More diverse and experimentally annotated datasets are needed to improve model robustness and ensure applicability across different repertoires.

4.2.6 Discussion

In this study, we evaluated a range of sequence based models for predicting SARS-CoV-2 antibody–antigen binding, including classical machine learning methods, transformer based architectures, recurrent networks, and a custom Siamese CNN (AbAgIntPre). While several models demonstrated performance above random chance (AUC > 0.6), none reached what is typically considered high discriminatory power (AUC > 0.80). The best performing model in our experiments the stacking ensemble achieved an AUC of 0.711 and F1-score of 0.648, outperforming all individual classifiers but still within a moderate range.

To contextualise our findings, we compare our results with recent sequence based models highlighted in our literature review: AbAgIntPre, AttABseq, MVSF-AB, and A2binder. AttABseq, an attention based model, demonstrated strong predictive performance on curated benchmarks, achieving high Pearson correlation and R^2 values, though these metrics are not directly comparable to our AUC results. MVSF-AB, which combines multi-view sequence features and a protein language model, reported robust performance on the SAbDab dataset, with a root mean square error (RMSE) of 1.84 kcal/mol and a Pearson correlation coefficient of 0.49. A2binder, leveraging large scale

pre-training and a multi-fusion CNN, achieved state of the art results on the CoV-AbDab and BioMap datasets, with ROC-AUC values up to 0.93 and Spearman correlations as high as 0.75 [96].

Our implementation of the AbAgIntPre Siamese network yielded an AUC of 0.694, close to XGBoost and logistic regression, but noticeably below the 0.82 AUC reported in the original AbAgIntPre publication (note that the 0.82 refers to their generic model evaluated on a broad, non-COVID dataset; the SARS-CoV specific variant also showed competitive performance, but its exact AUC was not separately reported). Although we used the same dataset and replicated the model architecture based on the available repository, the lack of access to the full implementation (e.g. training hyperparameters) may explain this discrepancy. This reflects a broader challenge in computational immunology: the difficulty of reproducibility when methodological details are omitted or ambiguous.

While these models all report strong performance on large, diverse datasets, our dataset is relatively modest in size, with around 1,200 sequences for training, and targets a single virus family. These constraints likely limit the performance ceiling of even sophisticated models. The relatively, diagonal ROC curves of deep models like Bi-LSTM and the underwhelming performance of ProtBERT with CKSAAP (AUC 0.664) reinforce the conclusion that pretrained embeddings or sequential modeling architectures alone do not guarantee success in low-data regimes.

Overall, our findings highlight the practical tradeoffs between model complexity and data availability. Simple, interpretable models like logistic regression and XGBoost remain competitive in smaller datasets and offer transparency in feature importance. More complex neural architectures may become viable with larger, more diverse antibody–antigen sequence data. For future work, integrating structure informed features or expanding the dataset with synthetic augmentation could help bridge the gap toward state of the art models.

4.3 Support Tools: Usability and Utility Evaluation

To support biomedical students and researchers in performing complex sequence analysis tasks without requiring programming expertise, two standalone desktop

applications were developed using Python (Tkinter) and packaged with PyInstaller. These graphical user interfaces (GUIs) encapsulate the core pipelines for DNA to protein translation and antibody sequence clustering, making advanced bioinformatics methods accessible through intuitive point and click workflows. The following sections highlight their functionality and present findings from a pilot usability session. Additional interface screenshots are available in Appendices F and G.

4.3.1 Functionality Overview

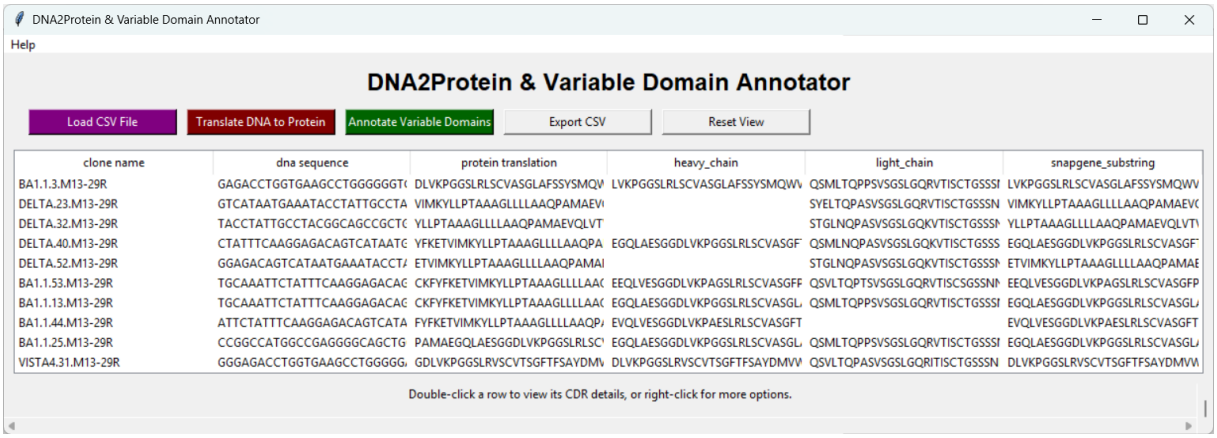


Figure 4.17 DNA-to-Protein Translation and CDR Annotation Interface: (1) bulk DNA import, (2) translation progress, (3) ANARCI FR/CDR boundaries, (4) interactive results table and export button.

Figure 4.17 presents the DNA to protein Translation and Annotation Interface. This tool allows users to load raw DNA sequences from a CSV file and automatically generate annotated protein sequences. Internally, it detects the optimal open reading frame (ORF), translates sequences using the Expasy REST API, and annotates the framework and complementarity determining regions (FRs/CDRs) via NovoProLabs' antibody annotation tool. Results are displayed in an interactive table where users can explore per clone details and export a comprehensive dataset including DNA, protein sequences, and region boundaries.

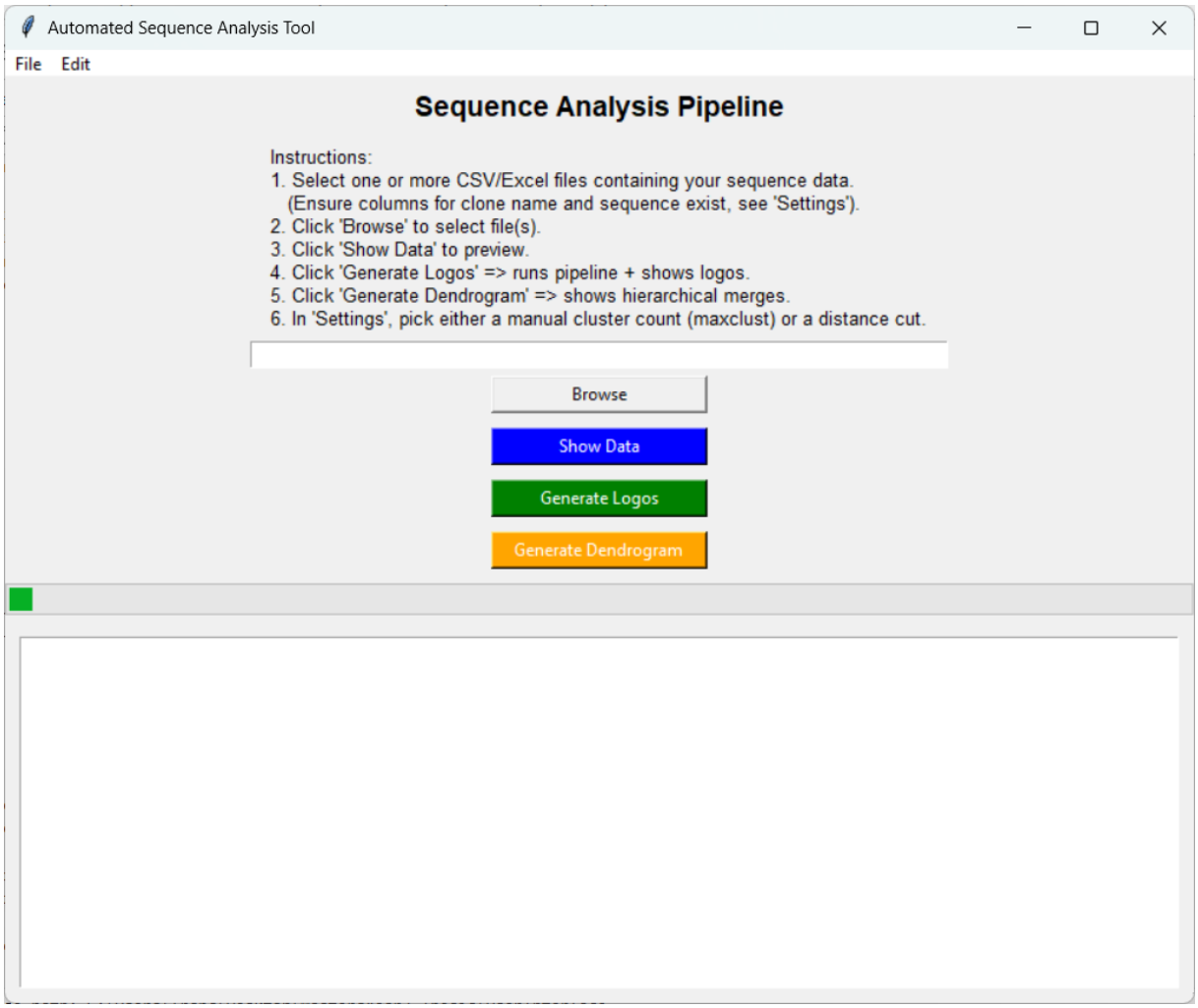


Figure 4.18 Sequence Analysis and Clustering Interface: (1) data-load panel, (2) BLAST-db build status, (3) clustering parameters, (4) preview and export controls.

Figure 4.18 shows the Sequence Analysis and Clustering Interface. After importing one or more CSV files containing antibody or nanobody sequences (e.g., CDRH3, CDRL3, VH), the tool constructs a BLAST protein database and applies hierarchical clustering using user defined distance thresholds or automatic dendrogram cuts. For each cluster, sequences are aligned with Clustal Omega and visualized as sequence logos using Logomaker. Users can inspect cluster dendrograms, explore sequence motifs, and export all visualizations for downstream use.

Pilot Usability Session

To assess the accessibility and practicality of the interfaces, a guided 30 minute usability session was conducted with three biomedical students at the University of Malta. None of the participants had prior coding experience. The session used a guided, task-based

format. Participants were observed completing key analysis steps data import, sequence annotation, clustering, and result interpretation while verbalising any difficulties. Observations focused on navigation ease, interface clarity, and output responsiveness, providing direct insight into usability and accessibility for non-programming users. During the session, all users were able to successfully install the applications and load sample datasets without assistance, translate raw DNA sequences, annotate antibody variable regions, perform clustering, and generate sequence logos, and export annotated data and figures suitable for academic presentations or publications.

Users noted two limitations during initial testing: (1) short antibody sequences were occasionally excluded from BLAST results, and (2) some output tables contained empty rows. In response, the software was updated to include the `blastp-short` parameter set, improving sensitivity for shorter peptides, and the table parsing logic was revised to remove spurious entries. Subsequent feedback confirmed that these revisions resolved the issues. Participants reported that the interfaces were easy to use, even for users unfamiliar with programming or command line tools. They also found the visual outputs such as cluster dendrograms and sequence logos helpful in understanding antibody sequence diversity and motif conservation. These findings support the GUIs' role as both practical research tools and effective teaching aids. By abstracting away technical complexity, the applications enable biomedical students and early career researchers to engage with antibody sequence data, perform meaningful analyses, and develop foundational skills in computational immunology.

4.4 Summary

We clustered antibody and nanobody repertoires using hierarchical methods, revealing four major CDRH3 sequence families in both the OPIG and Sanger datasets. Machine learning models for binding prediction achieved moderate performance, with the best model (a stacking ensemble) reaching an AUC of 0.711. However, generalisation to the Sanger set was poor, highlighting domain shift challenges. Finally, two user friendly GUI tools were developed and successfully tested with non-programmers, demonstrating their value for education and research in antibody analysis.

5 Conclusion

First, hierarchical clustering of over 300 CDRH3 sequences from the OPIG and Sanger datasets revealed eight major sequence families across antibodies and nanobodies, including six smaller clusters and two dominant lineages, one in each repertoire. These clusters were characterized by distinct, highly conserved motifs such as "AP…GWGG…YEY" and "NVKD…YDV," offering insight into potential structural or functional constraints. Second, our motif analysis distilled each family into interpretable sequence logos, highlighting conserved residues and substructures relevant to epitope targeting. Third, we benchmarked seven machine learning models on SARS-CoV-2 binding prediction and found that a stacking ensemble of linear and tree based classifiers outperformed standalone deep learning approaches, with the highest AUC (0.711) and F1-score (0.648), emphasizing the value of feature diversity in moderate scale datasets. Finally, we developed two standalone GUI applications: one for DNA to protein translation and CDR annotation, and another for BLAST based clustering and motif visualization. These tools were validated in a pilot session with biomedical students, confirming their usability and educational value in computational immunology.

5.1 Revisiting Our Aims and Objectives

We set out to advance our understanding of SARS-CoV-2 antibody repertoires by:

1. **Applying clustering techniques** to discover public CDRH3 clonotypes via hierarchical clustering and dendrogram visualization.
2. **Visualizing conserved motifs** in major clusters using sequence logos to pinpoint antigen binding "hot spots."
3. **Developing predictive models**, including logistic regression, random forest, XGBoost, Bi-LSTM, ProtBERT, Siamese CNN (AbAgIntPre), and a stacking ensemble, to classify antibody–antigen binding, with systematic hyperparameter tuning and threshold analyses.
4. **Building user-friendly GUIs** for DNA translation, CDR annotation, clustering, and motif visualization, enabling non-programmers to reproduce our workflows.

Each aim has been met: we applied clustering techniques and identified eight public CDRH3 families and their conserved motifs; demonstrated that a stacking ensemble attains acceptable discrimination (AUC = 0.711); and implemented two GUIs that help researchers cluster and convert DNA to protein sequences and annotate CDRs in under five minutes without prior coding experience.

5.2 Critique and Limitations

This study faces several constraints that temper the generality of its conclusions. The data are unavoidably biased: the public CoV-AbDab archive is rich in recurrent, well characterised CDRH3 loops, whereas the in-house Sanger cohort comprises only 137 sequences from a single canine library. As a result, rare or lineage specific motifs are under represented, and performance estimates should be interpreted as lower bounds for a truly global repertoire. All predictive models in this study rely only on sequence information using either CKSAAP counts or learned embeddings and do not take into account important structural factors that affect binding, such as the shape of the binding site (paratope) or how flexible the antibody is. This omission likely caps the maximum achievable AUC around 0.71. Similarly, the clustering pipeline relies on linear sequence similarity and does not incorporate three-dimensional structural distances, limiting the ability to detect “shape-public” clonotypes. Finally, the two graphical user interfaces were packaged only for Windows using PyInstaller. Without cross platform builds, unit testing, or continuous integration, these tools risk failing silently if dependent websites (e.g., Expasy or NovoProLabs) change format or enforce rate limits.

5.3 Future Work

Building on the foundations laid in this work, several promising avenues for future development emerge. One key direction is the incorporation of structural information such as homology models or AlphaFold derived CDRH3 conformations into the clustering process to capture shape based clonotypes that may be missed by sequence only comparisons. Another opportunity lies in expanding the framework to support cross-pathogen generalisation by applying it to antibody responses against diverse viral

and bacterial antigens, such as influenza or HIV, to identify conserved or broadly reactive paratope patterns. From a practical standpoint, transforming the current desktop GUIs into web based or cloud hosted tools would facilitate collaborative usage and ensure accessibility across platforms. Additionally, incorporating an active learning loop where experimental binding data from new assays informs iterative model retraining could significantly improve predictive performance and motif resolution. Finally, exploring alternative modeling approaches, such as epitope aware predictors that account for antigen context, may enhance the biological relevance and specificity of sequence based binding classifiers. Together, these extensions offer a path toward more robust, generalisable, and interpretable tools for computational antibody analysis.

5.4 Final Remarks

This study presents a unified framework for antibody repertoire analysis that combines data science approaches such as sequence clustering and predictive modeling with motif discovery and accessible software tools. By integrating public SARS-CoV-2 data with our Sanger derived sequences, we identified dominant CDRH3 lineages and conserved motifs relevant to antibody design. Model benchmarking showed that classical ensemble methods outperform standalone deep networks on moderate sized datasets, and that reasonable binding predictions are possible using sequence data alone. Importantly, we translated these methods into two intuitive GUIs, enabling non-programmers to conduct translation, annotation, and clustering tasks with ease. While constrained by dataset scope and lack of structural input, the modular pipeline is readily extendable. Future directions include adding structure aware features, expanding datasets, and deploying the tools in web based environments. Overall, this work contributes practical tools and methodological insights to bioinformatics and computational immunology, supporting both research and education in antibody informatics.

References

[1] C. Gaebler *et al.*, "Evolution of antibody immunity to sars-cov-2," *Nature*, vol. 591, pp. 639–644, 2021.

[2] B. J. Boyarsky *et al.*, "Antibody response to 2-dose SARS-CoV-2 mRNA vaccine series in solid organ transplant recipients," *JAMA*, vol. 325, no. 21, pp. 2204–2206, 2021. doi: 10.1001/jama.2021.7489.

[3] I. Setliff *et al.*, "Multi-donor longitudinal antibody repertoire sequencing reveals convergent antibody responses to sars-cov-2 infection," *Cell Reports Medicine*, vol. 2, no. 11, p. 100 414, 2021.

[4] Oxford Protein Informatics Group, *Covabdab: The covid antibody database*, https://opig.stats.ox.ac.uk/webapps/covabdab/, Accessed: 2023-03-17, 2020.

[5] G. W. A. Dick, S. F. Kitchen, and A. J. Haddow, "Zika virus. ii. pathogenicity and physical properties," *Transactions of the Royal Society of Tropical Medicine and Hygiene*, vol. 46, pp. 521–534, 1952. doi: 10.1016/0035-9203(52)90042-4.

[6] M. P. O'Brien, E. Forleo-Neto, B. J. Musser, F. Isa, K.-C. Chan, N. Sarkar, *et al.*, "Subcutaneous regen-cov antibody combination to prevent covid-19," *The New England Journal of Medicine*, vol. 385, no. 13, pp. 1184–1195, Sep. 2021. doi: 10.1056/NEJMoa2109682.

[7] D. M. Weinreich, S. Sivapalasingam, T. Norton, S. Ali, H. Gao, R. Bhore, *et al.*, "Regen-cov antibody combination and outcomes in outpatients with covid-19," *The New England Journal of Medicine*, vol. 385, no. 23, e81, Dec. 2021. doi: 10.1056/NEJMoa2108163.

[8] E. A. Kabat and T. T. Wu, "Identical v region amino acid sequences and segments of sequences in antibodies of different specificities," *Biochemistry*, vol. 16, no. 9, pp. 3188–3193, 1977. doi: 10.1021/bi00635a021.

[9] J. L. Xu and M. M. Davis, "Diversity in the cdr3 region of v(h) is sufficient for most antibody specificities," *Immunological Reviews*, vol. 176, pp. 125–136, 2000. doi: 10.1034/j.1600-065x.2000.1760113.x.

[10] CUSABIO, *How to validate an antibody*, Accessed: 2025-05-16, n.d. [Online]. Available: https://www.cusabio.com/c-21077.html.

[11] H. W. Schroeder Jr and L. Cavacini, "Structure and function of immunoglobulins," *Journal of Allergy and Clinical Immunology*, vol. 125, no. 2, S41–S52, 2010. doi: 10.1016/j.jaci.2009.09.046.

[12] G. Georgiou, G. C. Ippolito, J. Beausang, C. E. Busse, H. Wardemann, and S. R. Quake, "The promise and challenge of high-throughput sequencing of the antibody repertoire," *Nature Biotechnology*, vol. 32, no. 2, pp. 158–168, 2014. doi: 10.1038/nbt.2782.

[13] R. C. Edgar, "Muscle: Multiple sequence alignment with high accuracy and high throughput," *Nucleic acids research*, vol. 32, no. 5, pp. 1792–1797, 2004.

[14] K. Katoh and D. M. Standley, "Mafft multiple sequence alignment software version 7: Improvements in performance and usability," *Molecular Biology and Evolution*, vol. 30, no. 4, pp. 772–780, 2013. doi: 10.1093/molbev/mst010.

[15] N. Zhu, D. Zhang, W. Wang, X. Li, B. Yang, and J. e. a. Song, "A novel coronavirus from patients with pneumonia in china, 2019," *New England Journal of Medicine*, vol. 382, pp. 727–733, 2020.

[16] World Health Organization, *Who coronavirus (covid-19) dashboard*, https://covid19.who.int/, accessed: 2025-05-14, 2024.

[17] Johns Hopkins University, *Covid-19 map — johns hopkins coronavirus resource center*, https://coronavirus.jhu.edu/map.html, accessed: 2025-05-14, 2024.

[18] D. M. Weinreich, S. Sivapalasingam, T. Norton, S. Ali, H. Gao, and R. e. a. Bhore, "Regn-cov2, a neutralizing antibody cocktail, in outpatients with covid-19," *New England Journal of Medicine*, vol. 384, pp. 238–251, 2021.

[19] P. Chen, A. Nirula, B. Heller, R. L. Gottlieb, J. Boscia, and J. e. a. Morris, "Sars-cov-2 neutralizing antibody ly-cov555 in outpatients with covid-19," *New England Journal of Medicine*, vol. 384, pp. 229–237, 2021.

[20] K. Gao *et al.*, "Methodology-centered review of molecular modeling, simulation, and prediction of SARS-CoV-2," *Chemical Reviews*, vol. 122, no. 13, pp. 11 287–11 368, 2022, Epub 2022 May 20. doi: 10.1021/acs.chemrev.1c00965.

[21] H. Lv, L. Shi, J. W. Berkenpas, F.-Y. Dao, and H. Zulfiqar, "Application of artificial intelligence and machine learning for COVID-19 drug discovery and vaccine design," *Briefings in Bioinformatics*, vol. 22, no. 6, bbab320, 2021. doi: 10.1093/bib/bbab320.

[22] A. K. Abbas, A. H. Lichtman, and S. Pillai, *Basic Immunology: Functions and Disorders of the Immune System*, 7th ed. Elsevier, 2022, isbn: 978-0323757485.

[23] B. Alberts *et al.*, *Molecular Biology of the Cell*, 6th ed. Garland Science, 2015, isbn: 978-0815344322.

[24] M. Grima, “Characterization of canine scfv phage display libraries,” MSc thesis, University of Malta, 2023. [Online]. Available: https://www.um.edu.mt/library/oar/handle/123456789/129869.

[25] R. M. MacCallum, A. C. Martin, and J. M. Thornton, “Antibody–antigen interactions: Contact analysis and binding site topography,” *Journal of Molecular Biology*, vol. 262, no. 5, pp. 732–745, 1996.

[26] G. P. Smith, “Filamentous fusion phage: Novel expression vectors that display cloned antigens on the virion surface,” *Science*, vol. 228, no. 4705, pp. 1315–1317, 1985. doi: 10.1126/science.4001944.

[27] J. McCafferty, A. D. Griffiths, G. Winter, and D. J. Chiswell, “Phage antibodies: Filamentous phage displaying antibody variable domains,” *Nature*, vol. 348, no. 6301, pp. 552–554, 1990. doi: 10.1038/348552a0.

[28] S. F. Parmley and G. P. Smith, “Antibody-selectable filamentous fd phage vectors: Affinity purification of target genes,” *Gene*, vol. 73, no. 2, pp. 305–318, 1988. doi: 10.1016/0378-1119(88)90495-7.

[29] H. R. Hoogenboom, “Overview of antibody phage-display technology and its applications,” in *Methods in Molecular Biology*, S. Dübel, Ed., vol. 178, Humana Press, 2002, pp. 1–37. doi: 10.1385/1-59259-240-6:001.

[30] D. Gamermann, A. Montagud, J. A. Conejero, P. Fernández de Córdoba, and J. F. Urchueguía, “Large scale evaluation of differences between network-based and pairwise sequence-alignment-based methods of dendrogram reconstruction,” *arXiv preprint arXiv:1709.09236*, 2017.

[31] F. Murtagh and P. Contreras, “Algorithms for hierarchical clustering: An overview, ii,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 6, e1219, 2017.

[32] J. Felsenstein, *Inferring Phylogenies*. Sinauer Associates, 2004.

[33] R. R. Sokal and F. J. Rohlf, *Biometry: The Principles and Practice of Statistics in Biological Research*. W.H. Freeman, 1995.

[34] R. W. Hamming, “Error detecting and error correcting codes,” *The Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[35] T. H. Jukes and C. R. Cantor, *Evolution of protein molecules*, H. N. Munro, Ed. Academic Press, 1969, pp. 21–132.

[36] M. Kimura, “A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences,” *Journal of Molecular Evolution*, vol. 16, no. 2, pp. 111–120, 1980.

[37] N. Saitou and M. Nei, “The neighbor-joining method: A new method for reconstructing phylogenetic trees,” *Molecular Biology and Evolution*, vol. 4, no. 4, pp. 406–425, 1987.

[38] K. Tamura, D. Peterson, D. Peterson, G. Stecher, M. Nei, and S. Kumar, “Mega5: Molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods,” *Molecular Biology and Evolution*, vol. 28, no. 10, pp. 2731–2739, 2011. doi: 10.1093/molbev/msr121.

[39] W. Hennig, *Phylogenetic Systematics*. University of Illinois Press, 1966.

[40] P. Pai, *Hierarchical clustering explained*, *Towards Data Science*, Accessed: 2025-03-30, Apr. 2019. [Online]. Available: https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8.

[41] T. D. Schneider and R. M. Stephens, “Sequence logos: A new way to display consensus sequences,” *Nucleic Acids Research*, vol. 18, no. 20, pp. 6097–6100, 1990.

[42] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner, “Weblogo: A sequence logo generator,” *Genome Research*, vol. 14, no. 6, pp. 1188–1190, 2004. doi: 10.1101/gr.849004.

[43] S. A. Robinson, M. I. J. Raybould, C. Schneider, W. K. Wong, C. Marks, and C. M. Deane, “Epitope profiling using computational structural modelling demonstrated on coronavirus-binding antibodies,” *PLOS Computational Biology*, vol. 17, no. 12, pp. 1–20, Dec. 2021. doi: 10.1371/journal.pcbi.1009675. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1009675.

[44] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, “Predicting the sequence specificities of dna- and rna-binding proteins by deep learning,” *Nature Biotechnology*, vol. 33, pp. 831–838, 2015.

[45] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, 1990.

[46] C. Camacho, G. Coulouris, V. Avagyan, *et al.*, “BLAST+: architecture and applications,” *BMC Bioinformatics*, vol. 10, no. 421, pp. 1–9, 2009.

[47] G. M. Boratyn, J. Thierry-Mieg, D. Thierry-Mieg, B. Busby, and T. L. Madden, "Magic-BLAST, an accurate RNA-seq aligner for long and short reads," *BMC Bioinformatics*, vol. 20, no. 1, p. 405, 2019.

[48] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pp. 28–36, 1994.

[49] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[50] T. L. Bailey *et al.*, "Meme suite: Tools for motif discovery and searching," *Nucleic Acids Research*, vol. 37, no. Web Server issue, W202–W208, 2009. doi: 10.1093/nar/gkp335.

[51] D. Wei, Q. Jiang, Y. Wei, and S. Wang, "A novel hierarchical clustering algorithm for gene sequences," *BMC Bioinformatics*, vol. 13, no. 174, 2012. doi: 10.1186/1471-2105-13-174.

[52] F. Murtagh and P. Contreras, "Methods of hierarchical clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012. doi: 10.1002/widm.1077.

[53] A. Zemla *et al.*, "Stralcp: Structure alignment-based clustering of proteins," *Nucleic Acids Research*, vol. 35, no. 22, e150, 2007. doi: 10.1093/nar/gkm1049.

[54] Q. Ma, G.-W. Chirn, R. Cai, J. D. Szustakowski, and N. R. Nirmala, "Clustering protein sequences with a novel metric transformed from sequence similarity scores and sequence alignments with neural networks," *BMC Bioinformatics*, vol. 6, no. 242, 2005. doi: 10.1186/1471-2105-6-242.

[55] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982. doi: 10.1109/TIT.1982.1056489.

[56] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, AAAI Press, 1996, pp. 226–231.

[57] C. C. Aggarwal, Ed., *Data Classification: Algorithms and Applications*. CRC Press, 2013, isbn: 9781466558212.

[58] G. Yu, L. Ren, J. Wang, C. Domeniconi, and X. Zhang, "Multiple clusterings: Recent advances and perspectives," *Computer Science Review*, vol. 50, p. 100 621, 2024. doi: 10.1016/j.cosrev.2024.100621.

[59] X. Huang, Q.-H. Zhang, and Y. Li, "AbAgIntPre: A siamese convolutional neural network for antibody–antigen binding prediction," *Bioinformatics*, vol. 38, no. Suppl. 1, pp. i322–i330, 2022. doi: 10.1093/bioinformatics/btac233.

[60] J. T. Shapiro, D. VanInsberghe, I. A. Sidorov, and K. G. Andersen, "Classlog: Logistic regression for the classification of genetic sequences," *Frontiers in Virology*, vol. 3, p. 1 215 012, 2023. doi: 10.3389/fviro.2023.1215012.

[61] V. C. Nashine, D. Hamelberg, and A. Saran, "Automatic structure classification of small proteins using random forest approach," *BMC Bioinformatics*, vol. 11, no. 1, p. 364, 2010. doi: 10.1186/1471-2105-11-364.

[62] A. W. Services, *Fine-tune and deploy the protbert model for protein classification using amazon sagemaker*, https://aws.amazon.com/blogs/machine-learning/fine-tune-and-deploy-the-protbert-model-for-protein-classification-using-amazon-sagemaker/, 2021.

[63] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd. Wiley, 2013, isbn: 9780470582473.

[64] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006, isbn: 978-0-387-31073-2.

[65] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.

[66] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[67] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[68] R. M. Schmidt, *Recurrent neural networks (rnns): A gentle introduction and overview*, 2019. arXiv: 1912.05911 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1912.05911.

[69] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[70] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[71] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, vol. 4, 1997, pp. 2527–2530.

[72] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6000–6010. [Online]. Available: https://arxiv.org/abs/1706.03762.

[73] A. Elnaggar *et al.*, "Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7112–7127, 2022. doi: 10.1109/TPAMI.2021.3095381.

[74] A. Rives *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 118, no. 15, e2016239118, 2021. doi: 10.1073/pnas.2016239118.

[75] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, 1993, pp. 737–744.

[76] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE, 2006, pp. 1735–1742. doi: 10.1109/CVPR.2006.100.

[77] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[78] R. R. Sokal and F. J. Rohlf, "The comparison of dendrograms by objective methods," *Taxon*, vol. 11, no. 2, pp. 33–40, 1962.

[79] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985. doi: 10.1007/BF02294245.

[80] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011. [Online]. Available: https://www.researchgate.net/publication/228676767_Evaluation_From_Precision_Recall_and_F-Measure_to_ROC_Informedness_Markedness_and_Correlation.

[81] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, 1st. Cambridge, MA: MIT Press, 2008.

[82] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

[83] R. R. Sokal and C. D. Michener, "A statistical method for evaluating systematic relationships," *University of Kansas Science Bulletin*, vol. 38, pp. 1409–1438, 1958.

[84] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14 863–14 868, 1998.

[85] B. Briney, A. Inderbitzin, C. Joyce, and D. R. Burton, "Clonify: Unseeded antibody lineage assignment from next-generation sequencing data," *Scientific Reports*, vol. 6, no. 1, p. 23 901, 2016.

[86] L. McInnes and J. Healy, "Hdbscan: Hierarchical density based clustering," *Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017. doi: 10.21105/joss.00205.

[87] S. van Dongen, "A fast and scalable algorithm for clustering protein sequences," *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, vol. 8, pp. 281–291, 2000.

[88] D. Chomicz *et al.*, "Benchmarking antibody clustering methods using sequence, structural, and machine learning similarity measures for antibody discovery applications," *Frontiers in Molecular Biosciences*, vol. 11, p. 1 352 508, 2024.

[89] Z. Chen, A. M. Collins, and Y. Wang, "Clustering-based identification of clonally-related immunoglobulin gene sequence sets," *Immunome Research*, vol. 6, p. 3, 2010. doi: 10.1186/1745-7580-6-3. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2946782/.

[90] N. T. Gupta, K. D. Adams, A. W. Briggs, S. C. Timberlake, F. Vigneault, and S. H. Kleinstein, "Hierarchical clustering can identify b cell clones with high confidence in ig repertoire sequencing data," *The Journal of Immunology*, vol. 198, no. 6, pp. 2489–2499, 2017. doi: 10.4049/jimmunol.1601850.

[91] M. Foglierini, A. Lanzavecchia, and F. Sallusto, "Exploring the impact of clonal definition on b-cell diversity," *Frontiers in Immunology*, vol. 14, p. 1 123 968, 2023. doi: 10.3389/fimmu.2023.1123968.

[92] N. Nouri and S. H. Kleinstein, "Performance-optimized partitioning of clonotypes from high-throughput immunoglobulin repertoire sequencing data," *bioRxiv*, 2017. doi: 10.1101/175315.

[93] G. Shahaf, M. Barak, N. S. Zuckerman, N. Swerdlin, M. Gorfine, and R. Mehr, "Reconstructing and mining the b cell repertoire with immunediversity," *Nucleic Acids Research*, vol. 43, no. 7, e43, 2015. doi: 10.1093/nar/gkv129.

[94] Z. Jin *et al.*, "Attabseq: Incorporating self-attention into antibody sequence analysis," *Nature Methods*, vol. 21, no. 3, pp. 234–245, 2024. doi: 10.1038/s41592-024-01456-9.

[95] M. Li *et al.*, "Mvsf-ab: Accurate antibody-antigen binding affinity prediction via multi-view sequence feature learning," *Bioinformatics*, vol. 41, no. 5, btae579, May 2025. doi: 10.1093/bioinformatics/btae579.

[96] H. He *et al.*, "De novo generation of sars-cov-2 antibody cdrh3 with a pre-trained generative large language model," *Nature Communications*, vol. 15, Aug. 2024. doi: 10.1038/s41467-024-50903-y. [Online]. Available: https://doi.org/10.1038/s41467-024-50903-y.

[97] R. Evans *et al.*, "Protein complex prediction with alphafold-multimer," *bioRxiv*, 2021, Cold Spring Harbor Laboratory preprint. doi: 10.1101/2021.10.04.463034.

[98] X. Fang *et al.*, "Helixfold-multimer: Elevating protein complex structure prediction to new heights," *arXiv preprint arXiv:2404.10260*, 2024, preprint.

[99] D. Kozakov *et al.*, "The cluspro web server for protein–protein docking," *Nature Protocols*, vol. 12, no. 2, pp. 255–278, Feb. 2017. doi: 10.1038/nprot.2016.169.

[100] C. Dominguez, R. Boelens, and A. M. J. J. Bonvin, "Haddock: A protein–protein docking approach based on biochemical or biophysical information," *Journal of the American Chemical Society*, vol. 125, no. 7, pp. 1731–1737, Feb. 2003. doi: 10.1021/ja026939x.

[101] J. Dunbar and C. M. Deane, "ANARCI: Antigen receptor numbering and receptor classification," *Bioinformatics*, vol. 32, no. 2, pp. 298–300, 2016. doi: 10.1093/bioinformatics/btv552.

[102] H. M. Berman *et al.*, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000. doi: 10.1093/nar/28.1.235.

[103] J. Leem, J. Dunbar, G. Georges, J. Shi, and C. M. Deane, "Abodybuilder: Automated antibody structure prediction with data-driven accuracy estimation," *mAbs*, vol. 8, no. 7, pp. 1259–1268, 2016. doi: 10.1080/19420862.2016.1205773.

[104] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R. D. Appel, and A. Bairoch, "The expasy server: Proteomics tools for the life sciences," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3784–3788, 2003. doi: 10.1093/nar/gkg563.

[105] N. Bioscience, *Antibody cdr annotation tool*, https://www.novoprolabs.com/tools/cdr, Accessed: 2025-04-27.

[106] SnapGene, *SnapGene® software*, https://www.snapgene.com, Dotmatics, 2025.

[107] National Center for Biotechnology Information (US), *Blastp application options*, https://www.ncbi.nlm.nih.gov/books/NBK279684/table/appendices.T.blastp_application_options/, Accessed 2025-05-10, 2023.

[108] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: Which algorithms implement it correctly?" *Journal of Classification*, vol. 31, no. 3, pp. 274–295, 2014. doi: 10.1007/s00357-014-9161-z.

[109] emersON106, *AbAgIntPre: Antibody–Antigen Interaction Prediction Dataset*, https://github.com/emersON106/AbAgIntPre, Accessed: 2025-05-03, 2021.

[110] A. Elnaggar *et al.*, "Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. doi: 10.1109/TPAMI.2021.3095381. [Online]. Available: https://doi.org/10.1109/TPAMI.2021.3095381.

[111] National Center for Biotechnology Information, *Surface glycoprotein [severe acute respiratory syndrome coronavirus 2] (bcn86353.1)*, https://www.ncbi.nlm.nih.gov/protein/BCN86353.1, Accessed: 2025-05-03, 2020.

[112] National Center for Biotechnology Information, *Spike glycoprotein [severe acute respiratory syndrome coronavirus] (p59594.1)*, https://www.ncbi.nlm.nih.gov/protein/P59594.1, Accessed: 2025-05-03, 2003.

[113] Z. Chen, Y.-Z. Chen, X.-F. Wang, C. Wang, R.-X. Yan, and Z. Zhang, "Prediction of ubiquitination sites by using the composition of k-spaced amino acid pairs," *PLoS ONE*, vol. 6, no. 7, e22930, 2011. doi: 10.1371/journal.pone.0022930. [Online]. Available: https://doi.org/10.1371/journal.pone.0022930.

[114] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd. Springer, 2009, isbn: 978-0-387-84858-7.

[115] C. Molnar, *Interpretable Machine Learning*. lulu.com, 2020.

[116] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[117] L. Breiman, “Stacked regressions,” *Machine Learning*, vol. 24, no. 1, pp. 49–64, 1996.

[118] M. W. Libbrecht and W. S. Noble, “Machine learning applications in genetics and genomics,” *Nature Reviews Genetics*, vol. 16, no. 6, pp. 321–332, 2015.

[119] J. Zhou and O. G. Troyanskaya, “Predicting effects of noncoding variants with deep learning–based sequence model,” *Nature Methods*, vol. 12, no. 10, pp. 931–934, 2015. doi: 10.1038/nmeth.3547.

[120] A. Elnaggar *et al.*, “Prottrans: Towards cracking the language of life’s code through self-supervised deep learning and high performance computing,” *arXiv preprint arXiv:2007.06225*, 2021, Version 3, revised May 4, 2021. [Online]. Available: https://arxiv.org/abs/2007.06225.

[121] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[122] GSL Biotech LLC, *Batch Convert Sequence Files to a New Format*, SnapGene Support, https://support.snapgene.com/hc/en-us/articles/10384258920980-Batch-Convert-Sequence-Files-to-a-New-Format, Accessed: 29 April 2025, 2022.

[123] J. Glanville *et al.*, “Identifying specificity groups in the t cell receptor repertoire,” *Nature*, vol. 547, no. 7661, pp. 94–98, 2017. doi: 10.1038/nature22976. [Online]. Available: https://doi.org/10.1038/nature22976.

[124] A. Kovaltsuk, J. Leem, S. Kelm, J. Snowden, C. M. Deane, and K. Krawczyk, “Observed antibody space: A resource for data mining next-generation sequencing of antibody repertoires,” *Nucleic Acids Research*, vol. 46, no. D1, pp. D1043–D1048, 2018. doi: 10.1093/nar/gkx1054. [Online]. Available: https://doi.org/10.1093/nar/gkx1054.

[125] N. White, R. Parsons, G. S. Collins, A. G. Barnett, and the Prognosis Research Group, “Evidence of questionable research practices in clinical prediction models,” *BMC Medicine*, vol. 21, no. 1, p. 339, 2023. doi: 10.1186/s12916-023-03048-6. [Online]. Available: https://doi.org/10.1186/s12916-023-03048-6.

[126] P. E. Compeau and P. A. Pevzner, “Bioinformatics algorithms: An active learning approach,” *Chapter 3: Strings, Genomes and k-mers*, 2015, Cold Spring Harbor Laboratory Press.

[127] E. Liberis, P. Veličković, P. Sormanni, M. Vendruscolo, and P. Liò, "Parapred: Antibody paratope prediction using convolutional and recurrent neural networks," *Bioinformatics*, vol. 34, no. 17, pp. 2944–2950, 2018. doi: 10.1093/bioinformatics/bty305.

Appendix A Amino Acid Codes

Table A.1 Standard amino acid codes. Each code corresponds to a specific amino acid residue found in protein sequences. These codes are used throughout protein analysis and sequence logos.

Code	Three-letter	Amino Acid
A	Ala	Alanine
R	Arg	Arginine
N	Asn	Asparagine
D	Asp	Aspartate
C	Cys	Cysteine
E	Glu	Glutamate
Q	Gln	Glutamine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
L	Leu	Leucine
K	Lys	Lysine
M	Met	Methionine
F	Phe	Phenylalanine
P	Pro	Proline
S	Ser	Serine
T	Thr	Threonine
W	Trp	Tryptophan
Y	Tyr	Tyrosine
V	Val	Valine

Appendix B OPIG Cov-AbDab CDRH3 Only Clustering

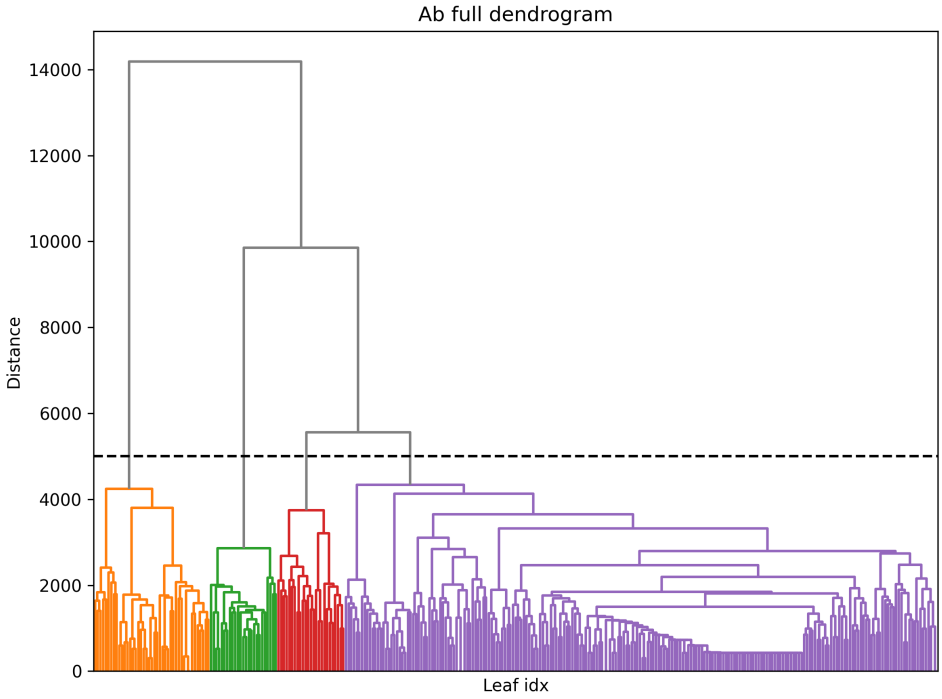


Figure B.1 Full hierarchical clustering dendrogram of antibody CDRH3 sequences.

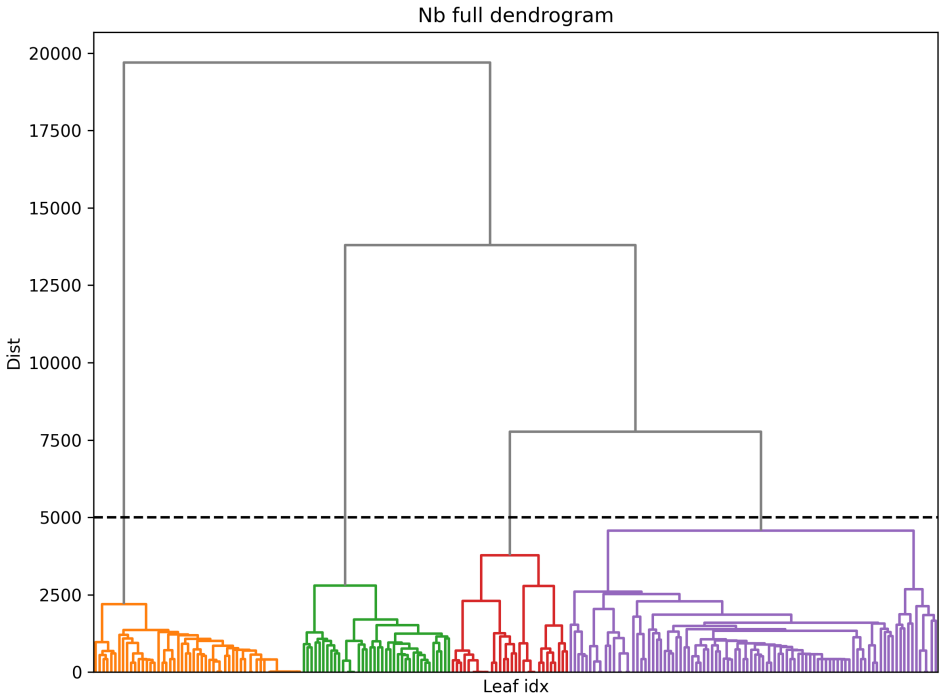


Figure B.2 Full hierarchical clustering dendrogram of nanobody CDRH3 sequences.

Table B.1 Full list of antibody sequences grouped by cluster for OPIG Dataset only.

ID	Sequence	Cluster
R15-F7	ASGAFYYGSGSYPFDY	3
scFv2-7	ARISGSGYFYPFDI	3
1F	ARHVRSAYYYGSGSYRDEGNWFDP	1
Jaiswal_scFv	AKTTTAFDY	4
Li_D2	AKDVYSESGSGSYYDY	3
R1-26	ARGQLGPWVGVDY	4
R1-30	ARQGWLRGNFDY	4
R2-3	ASQLWLRGAFDI	2
R2-6	ARKGWLRGAFDI	2
R1-32	ARENGYSGYGAAANFDL	4
S2B2	ARAYTGSYYYGMDV	1
S2B5	ARARGGSYYYGMDV	1
S2B7	ARSRGGGYYYGMDV	1
S2E6	ARAHRGSYYYGMDV	1
S2E7	ASDVAGHHGMDV	1
S2F6	ARANWGSYYYGMDV	1
S2110#	ARATWGNYYYGMDV	1
XK01	ARERGYSGYGAAYYFDY	3
XK02	ASWLYGDPISFDY	4
CR3022-B6	AGGSGISTPMDV	4
FC08	VRERGYSGYGAAYYFDY	3
CR3014-C8	ARGISPFYFDY	3
P17	ARHATLMNNKDI	4
BD-494	ARDLVVYGMDV	1
WIBP-2B11	ARDLMEVGGMDV	1
256	ARDGYGSGSDYYYYYYMDV	1

Continued on next page

Table B.1 – continued from previous page

ID	Sequence	Cluster
Tang_1E09	ARSGDSDAFDI	2
10C	ARQLSYCSTGVSCPGAI	4
11A	ARQGDYSGPSINY	4
12E	ARVGYYGDYAWGYYYYGMDV	1
15D	KGIYDVTGSSFDS	4
1F8	ARDRWMTTRAFDI	2
23E	ARARPSDPYDGSGFDAFDI	2
26H	ARERWLQIGEDAFDI	2
27D	ARNFGEDFDY	4
28G	ARLRPNSSGWTDY	4
2G7	AKATTVTTYFDY	3
3A1	ARDQGISANFKDAFDI	2
3B11	ARVGYCSSTSCHIGAFDI	2
3B12	STTSCASGAFDI	2
3C12	ARDYYGSGARGFDY	3
4A3	AKDADSFDY	4
4S1	AKAADTNYGWRTAGSIDA	4
4S12	AKAAGSCNYGYSCSYYIDT	4
4S15	AKYSGSWTAYAYITAGSIDA	4
4S3	AKSVGGGGWHAATIDA	4
4S4	AKDSTNTGGCGGYDIDA	4
6A	AKDYGEGFDY	4
80R	ARDRSYYLDY	4
8C	ARDWGYSFGD	4
8D3	ARRPLYDGDYGYPMDY	3
91M	ARDQGWGWDGTEYYSDY	4
92N	ARDDLSDYGEWLGPDY	4
A10	ARGGNGGMDV	1
ab1	ARGYGDYYFDY	3
ab4	ARETVSYGMDV	1
ab5	ARDRGYSSGWTDGFDI	4
ABP18	GAHIAAEYFQH	4
Asarnow_3D11	ARRWWLRGAFDI	2
Asarnow_5A6	ARLITMVRGEDY	4
B01	ARWDFASPYYPGSSGLDY	3
B1	ARGVAVAGTWDWFDP	4
B10	TSVCSGGSCYQ	4
B2	AKVGEVGSREWSAFDV	4
B8	STDSGSIGEF	4
BLN1	VTAPAITGSPEAYSYYYGMDV	1
BLN10	ATGPAIAAAATGWFDP	4
BLN12	VTAPVITGSPEAYSYYYGMDV	1
BLN14	AKDHDDGYYFYYYMDV	1
BLN2	AASPAVRGSPSNFYYYHGMDV	1
BLN3	VAAPVITGSPEAYSYYYGMDV	1
BLN4	ARMAYQVYYYDSSGYYDAFDI	2
BLN7	ATSRVAGTPNWFHP	4
BLN8	ARDLGSGWYP	4
C7	TRISGYYGAGSGGAMDV	3
CN111778218A-A9	AKGTDAFDY	4
CN111778218A-E11	AKNSDSFDY	4
CN111778218A-F10	AKDSDTFDY	4
CN111778218A-F5	AKNDSSFDY	4

Continued on next page

Table B.1 – continued from previous page

ID	Sequence	Cluster
CN111778218A-H9	AKSTNTFDY	4
CN111875700A-1A6	ARDALGWYFDV	3
CN111875701A	ARQGSHYGMDV	1
CN112250763B_Ab1	AREHTVAPVYGFDV	1
CN113045647A-1	AAFPGMDDDSVFNY	4
CN113045647A-2	AASTQDPGYMDFTTEY	4
CN113045647A-3	AAGFLDSSIMQRIVVGYATDY	4
CN113045647A-4	AEGRDSWWPSHYTMVPQRKYNAY	4
CN113045647A-5	AANYYHLFVMHYQWY	4
CR3001	ARHRFRHVFDY	3
CR3002	ARYYSRSLKAFDY	4
CR3006	AKDGSPRTPSFDY	4
CR3009	AKGLFMVTTYAFDY	4
CR3013	AKGLTPLYFDY	3
CR3015	ARGLSLRP	4
CR3018	AKFNPFTSFDY	4
CR3022-G11	AGGDGVSTPMDV	4
D2	FTATFAMDY	4
E3	ARHNAQFGELLVPQDAFDM	4
Fab_15033	ARSYYYGGFGMDY	1
Fab-108	AKDGSQLAYLVEYFQH	4
Fab-120	AKDFGGGTRYDYWYFDL	4
Fab-128	ARDGRYGSGSYPFDY	3
Fab-158	ARDPGGSYSNDAFDI	2
Fab-160	ARANSLRYYYGMDV	1
Fab-178	ARDISSWYEITKFDP	4
Fab-180	AREAVAGTHPQAGDFDL	4
Fab-192	STYYYDSSGYSTDY	3
Fab-236	ASRGIQLLPRGMDV	1
Fab-254	AREGDGYNFYFDY	3
Fab-298	ASDPRDDIAGGY	4
Fab-324	ARVGDYGDYIVSPFDL	4
Fab-349	AGNHAGTTVTSEYFQH	4
Fab-368	ARGSSGYYYG	1
Fab-46	ARGDSRDAFDI	2
Fab-52	ARDRGDTIDY	4
Fab-56	ARDIGPIDY	4
Fab-64	ARDTYGGKVTYFDY	3
Fab-80	ARSTRELPEVVDWYFDL	4
Fab-82	ARSRALYGSGSYFDY	3
Fab20	ARGGWSSSAGGYYGMDV	1
FC05	ATTTPFSSSYWFDP	4
FC11	AKAMFLGDSSGLTGLDMDV	4
G10	ARSGWDDAFDI	2
Goike-12C8	ATGPAVRRGSWFDP	4
Goike-1D1	ARIPIATHLGSDY	4
Goike-3-18	AKQAGAYCSGGSCYSSSEADY	4
Goike-3-26	ARPYSGSYWGYFDY	3
Goike-3B9	ARDGGGYVSY	4
Goike-4A5	AKASQLFWLGQFTRDGFDI	4
Goike-4C7	ATAAAVRGRGTIDY	4
Goike-6-3A	ARGTIYFDRSGYRRVDPFHI	4
Goike-7-6	ATRFAVYGDYLIDY	4

Continued on next page

Table B.1 – continued from previous page

ID	Sequence	Cluster
Goike-7A8	ATGSPFDRTQNWFDP	4
Goike-8-3	ARGGVVDTYYYYGMDV	1
Goike-8-42	ARDYGRGGV	4
Goike-8-96	AKAPGQWLRFHYYGMDV	1
Goike-8B5	ARELPPGRMVVPATYWHFDL	4
Goike-P3C6	APGRSLY	4
Goike-P4A3	ARDDTGRVGSGWYCPLY	4
H12	ARGGWCTGDCDARTFVWFEP	4
H6	ARGGGYDPWFAY	4
II62	AKAAGSFDY	4
JMB2002	ASLASYSSGWEDVFDI	4
Lsc12	AKAAGPDCCYTASNIDA	4
Lsc13	AKGSSGSCGSCAGNIDA	4
Lsc14	AKSSYCGGGYYSAANIDT	4
Lsc16	ARSSCGGDYETGCIDA	4
Lsc18	AKSGFRNGGGWSSAGLIDA	4
Lsc22	AKTIYGGGWSGYGDSIDG	4
Lsc8	AKESGAGGWNAGNVIDA	4
M14D3	ASSNYYGSGSYYPRSAFDI	2
M1A	ARDPVVVVINGDEAFDI	2
m396	ARDTVMGGMDV	1
m396-B10	ARDTATGGMDV	1
MD17	VKDQDSSSWYDAFDI	2
MD45	ARDLSVRGGMDV	1
MD47	AKDLVTAPSYEAFDI	2
MD62	ARDLQYYGMDV	1
MD63	VKDQDSNSWYDAFDI	2
MD65	ARDLAVAGAFDI	2
N18	ARGYWGSGYHYYGMDV	1
NBP10	YYAGGGFDV	4
NBP11	FSVGPGGPFDS	4
nCoV-163	ARSYGDFYVDF	4
nCoVmab1	ARGDGSDDYYYGMDV	1
P16-A3	ARAYSSSWLLQSFYYYGMDV	1
R3P1-A12	ARDLSEKGGMDV	1
Regdanvimab	ARIPGFLRYRNRYYYYGMDV	1
S-B8	AREYYYGMDV	1
S-E6	ATPGAIMGALHI	4
S2A3	AKDQYVSTDFDI	4
S2A6	AKHLYGSWAFDI	2
S2D5	AKVSSQTLRFDY	4
SA59B	AKRSLSFDY	4
SK1	VKDIYYRDRNLGFAFDI	2
Ssc20	AKSVNNSWSTGEDIDV	4
Ssc22	AKNNYNNGVDAAGDIDA	4
Ssc26	AKGGNGCSSGDHAGQIDA	4
Ssc29	ARSPGGAYSGSIDT	4
Ssc33	ARGAPGCDTWCWYGAAFIDA	4
Ssc35	AKGSGSACIWSGWCAGDIDS	4
Ssc37	AKSAYGGWTYADNIDA	4
STE70-1E12	VRDGYNFNNWFDP	4
STE72-1B6	AKAPYGDFRGLWYFDY	3
STE72-1G5	AVGGVQLWLTY	4

Continued on next page

Table B.1 – continued from previous page

ID	Sequence	Cluster
STE72-2G4	ARFFYDSSGYSTDY	3
STE72-4C10	AREYSSSWYGLGAFDI	2
STE72-4E12	ARDLSGGLDY	4
STE72-8E1	ARGGPKRSGSPFDV	3
STE73-2B2	ARVSGWYFGAFDI	2
STE73-2C2	ARGHDNLDY	4
STE73-2E9	ARGKFDY	4
STE73-2G8	ARWSGTYYDY	4
STE73-6B10	ARDRLRYGDSGSYYYYGMDV	1
STE73-6C1	ARSYVGGMDV	1
STE73-6C8	ARSIAALNWFDP	4
STE73-9G3	ARDLVLGSGSSND	4
STE90-C11	ARDVADAFDI	2
US10822379B1-#2	AREQQQLVPHYYYYGMDV	1
US10822379B1-#3	AKTYYDFWRTYYGMDV	1
US10822379B1-#4	ARDRYYTMDV	1
YU536-D04	ARDLVVMGLDV	4
YU537-H11	ARGESGSPYGMDV	1

Table B.2 Full list of nanobody sequences grouped by cluster for OPIG Dataset only.

Name	Sequence	Cluster
Sun_Nb1C6	AAELFCPWPDIGTMSPAEYKY	4
Sun_Nb1B5	AADSGWVGYSLDPYQYNY	4
F6	ARVESGSGWLDF	4
VHH_132a	AKNRRGGWTVSDLGD	4
VHH_134	AAGQLGWIADCLELADYNGYNY	4
4A12	VKDFGHLGQMAS	4
4AD5	VKDLGFADH	4
4C5	AREWHSGYDY	4
Bn03_n3113v	VSNWASGSTGDY	4
Bn03_n3130v	ATRSPFGDYAFSY	4
CN111647076A_4A10	VKDFVVGETAEFSY	4
CN111825762A-A1	AALASSGYSRDYGAYDY	4
CN111825762A-B4	APRNGSPSVFEILLVSVY	4
CN111825762A-B6	ASGAVPAHQIGFRSTTLY	4
CN111825762A-B9	ANVIGTVNAYGAASKPAY	4
CN111825762A-E3	AAVAMLPFLTAVTPRPGY	4
CN111825762A-E6	AFGPAPKPQNVLTTALPY	4
CN111825762A-E7	APLLASAFVLMYGSRHLY	4
CN111825762A-F1	AAYLSSSRSGDYY	4
CN111825762A-F4	APGPGFTTMDRSQARIAY	4
CN111825762A-F5	AHRFQTRVRTTNPIESEY	4
CN111825762A-F8	AFGHTHMVRPGSTVMIMY	4
CN111825762A-G2	AYVVPYAIAAGAPDQIGY	4
CN111825762A-G3	AASDRLSGLRSYGY	4
CN111825762A-G5	APRVRLKVRFQDRVMVTY	4
CN111825762A-H3	ARTSEARYRGYPRFRVMFY	4
CN111825762A-H4	ADRNRVRGYDPCLHGY	4
CN111825762A-H6	AAIAVRDPHYVGVGGY	3

Continued on next page

Table B.2 – continued from previous page

Name	Sequence	Cluster
CN112062838A-sdAb	VKDFVGADGPFVFDY	4
CN112062840A	AAAYRYNGRDYYDRYDY	4
CN113563463A-1FC	AAIVDGWI	4
CN113563463A-2FC	AAWYIKMNSDMHVQREWE	4
CN113563463A-3FC	AAMSIGWPELF	4
CN113563463A-4FC	AAIETVHGHMI	4
CN113563463A-5FC	AAIDTEYGQNI	4
CN113563463A-6FC	AAWPTQDGYAA	4
Feng_17F6	RAYSLSAGMCAWMGYI	4
Feng_20G6	RAYSTTGDERDCRWQGYI	4
Fu2	AVGPSFSYTGSTYYRSELPWDYDY	4
H11-D4	ARTENVRSLLSDYATWPYDY	4
H11-H4	AQTHYVSYLLSDYATWPYDY	4
Hanke_C11	NAWVPVEEVAGIARQFQEV	4
Hanke_C7	NRAAVDGGGGYVPGRGDY	4
Hanke_D4	NAKGSSWYDLGGGAGDDY	4
Hanke_D9	NAEFGTPPVGYDY	4
Hanke_E11	ASGGEPLPRYYTDYASWVDY	4
Hanke_E2	HLRTFRRAGADTIPY	4
Hanke_F1	AYYTGRMATGWGNGGWKEYDY	4
Hanke_F12	AAGGEPLPRYWTDYASWVDY	4
Hanke_G1	AAGGEPLPRYWSDYASWVDY	4
Hanke_G2	AAGGEGYDSDYGPPLAPDY	4
Hanke_G6	AAERWGYSDCVAGYGMDY	4
Hong-1B5	VAADSHNSRCYLGRSYVNY	4
Hong-1H6	AATLYRVNCAKREFDK	4
Hong-2F7	NAMGRGSGSRCDNWDPNY	4
Hong-7A3	AAGSWYNQWGYSMDY	4
Hong-8A2	AAHGTYDKYAPCGGFAGTYTY	4
Hong-8A4	IIEALSGY	4
LR1	AAAEWGYEWPLYYASSWY	2
LR11	AAAYWGWDWPLNSQDYWY	2
LR15	AAADWGYNWPLIREEYEY	2
LR16	AAADWGYNIPLNITDYWY	2
LR2	AAAMNGYNEPLYSYDYEY	2
LR3	AAASWGYEWPLVYDDYWY	2
LR5	AAATWGYHWPLGAWDYWY	2
LR6	AAATWGYSWPLEHDEYWY	2
LR7	AAAFHGEQYPLYTNKYHY	2
LR8	AAANYGANFPLQANTYFY	2
MR10	NVKDEGATTKVYDY	1
MR14	NVKDWGAANKYYDY	1
MR17	NVKDDGQLAYHYDY	1
MR2	NVKDYGWYNSQYDY	1
MR3	NVKDYGAASWEYDY	1
MR4	NVKDFGGHQAYYDY	1
MR6	NVKDEGDTSASYDY	1
MR7	NVKDEGYFSDEYDY	1
MR8	NVKDWGSSNQYYDY	1
Nanosota-1C	MAGSKSGHELDH	4
NRL-N-A9	NIIPKSDQGAVNT	4
NRL-N-B6	ASGRYLGGITSYSQGDFAP	4
NRL-N-C2	AKYQAAVHQEKEDY	4

Continued on next page

Table B.2 – continued from previous page

Name	Sequence	Cluster
NRL-N-E10	AARAGPLGFELSATSSAEYDY	4
NRL-N-E2	ATNTRWTYFSPTVPDRYDY	4
P14-F8	AVAASGDTFEGRSDPDY	4
P14-F8-35	AVAASPATFEGRSDPDY	4
P14-F8-38	AVAASGDTFFGRSDPDY	4
RBD-Nb3	AAGPIYRAEVRQSDFPY	4
RBD-Nb35	ATRTNWFYGACLPKVSDFGS	4
S-Nb82	AASPFKSVVLGPGLYHH	4
S-Nb91	AAGGRCRYAGPLRSDFTY	4
Sb#1	RVFVGWHY	4
Sb#10	TVYVGYEY	3
Sb#11	EVEVGKWY	4
Sb#12	YVWVGQEY	3
Sb#13	WVIVGEYY	4
Sb#14	YVYVGSSY	3
Sb#15	FVYVGRSY	3
Sb#16	IVWVGAQY	3
Sb#17	HVWVGSLY	3
Sb#18	YVYVGASY	3
Sb#2	RAVYVGMHY	3
Sb#20	YVYVGKSY	3
Sb#21	FVGVGTHY	4
Sb#22	FVYVGKSY	3
Sb#25	NVKDFGTHHYAYDY	1
Sb#26	NVKDKGMAVQWYDY	1
Sb#27	NVKDEGDMFTAYDY	1
Sb#28	NVKDSGQWRQEYDY	1
Sb#29	NVKDFGYTWHEYDY	1
Sb#3	VVWVGHNY	3
Sb#30	NVKDYGQAHAYYDY	1
Sb#31	NVKDTGTTEDYDY	1
Sb#32	NVKDAGRVYNSYDY	1
Sb#33	NVKDTGTYRFYYDY	1
Sb#34	NVKDAGVYNRYYDY	1
Sb#35	NVKDWGFASHAYDY	1
Sb#36	NVKDFGWQHQEYDY	1
Sb#37	NVKDSGSFNQAYDY	1
Sb#38	NVKDYGVHFKRYDY	1
Sb#39	NVKDAGNTTSAYDY	1
Sb#4	EVQVGAWY	4
Sb#40	NVKDIDAEAYAYDY	1
Sb#41	NVKDSGQWRVQYDY	1
Sb#42	NVKDHGAQNQMYDY	1
Sb#45	NVKDVGHHYEYYDY	1
Sb#46	NVKDKGQMRAAYDY	1
Sb#47	NVKDYGSSYYKYDY	1
Sb#48	NVKDAGSSYWDYD	1
Sb#49	AAARWGRTKPLNTYYYSY	2
Sb#5	RVHVGAHY	4
Sb#50	AAATEGHAHALYRLHYY	4
Sb#51	RVWVGTHY	3
Sb#52	AAAYVGAENPLPYSMYGY	2
Sb#53	AAADYGASDPLWFIHYLY	2

Continued on next page

Table B.2 – continued from previous page

Name	Sequence	Cluster
Sb#55	AAANYGSNFPLAEEDYWY	2
Sb#56	AAAYYFGDDIPLWWEAYSY	2
Sb#58	AAARWGRHMPLSATEYSY	2
Sb#59	AAAAWGNSAPLTTYRYYY	2
Sb#6	YVYVGAQY	3
Sb#61	AAADWGYDWPLWDEWYWY	2
Sb#62	AAANYGANYPLYSQQYSY	2
Sb#63	AAANYGANEPLQYTHYNY	2
Sb#64	AAASYGAAHPLSIMRYYY	2
Sb#65	AAASYGANFPLKASDYSY	2
Sb#67	AAATWGHSWPLYNDEYWY	2
Sb#68	AAAAWGYAWPLHQDDYWY	2
Sb#69	AAATWGYSWPLIAEYNWY	2
Sb#7	FVKVGNWY	4
Sb#71	AAANWGYSWPLYEADDWY	2
Sb#8	YVYVGGSY	3
Sb#9	RVFVGMHY	4
Sb100	AAANWGYSWPLYQTEYWY	2
Sb12	YVKVGEWY	3
Sb13	YVVVGWGY	3
Sb15	FVVVGNGY	4
Sb17	LVYVGATY	3
Sb23	AVQVGYWY	4
Sb25	YVYVGAGY	3
Sb27	YVWVGRSY	3
Sb30	YVYVGESY	3
Sb32	VVWVGEVY	3
Sb37	NVKDEGNTTAYYDY	1
Sb38	NVKDFGTQEHYYDY	1
Sb39	NVKDFGGYRYYYDY	1
Sb40	NVKDEGAIAKNYDY	1
Sb42	NVKDEGYTGYYYDY	1
Sb43	NVKDWGSQDRYYDY	1
Sb45	NVKDEGKSSQVYDY	1
Sb46	NVKDVGNDQKSYDY	1
Sb47	NVKDWGTYSTYYDY	1
Sb50	NVKDWGWLAQYYDY	1
Sb52	NVKDEGMWQHYYDY	1
Sb54	NVKDEGNSQSHYDY	1
Sb56	NVKDAGNSKALYDY	1
Sb57	NVKDWGRAGARYDY	1
Sb58	NVKDMDRWRTTYDY	1
Sb6	YVWVGNQY	3
Sb60	NVKDWGYEYEGYDY	1
Sb61	NVKDTGTYQAWYDY	1
Sb62	NVKDWGGYQWYYDY	1
Sb63	NVKDYGAQAHYYDY	1
Sb67	NVKDWGTYNSYYDY	1
Sb7	LVYVGSTY	3
Sb71	AAAHYGDNFPLAYQAYLY	2
Sb75	AAARWGRDEPLYHYYYSY	2
Sb76	NVKDIGAQEVHYDY	1
Sb78	AAANYGNNWPLTGVNYWY	2

Continued on next page

Table B.2 – continued from previous page

Name	Sequence	Cluster
Sb8	YVWVGDSY	3
Sb83	AAAKYGQNFPLSYHAYRY	2
Sb84	AAARYGRSDPLHYHEYSY	2
Sb85	AAASWGYTWPLYTYDYWY	2
Sb88	NVKDSGQYRENYDY	1
Sb9	WVYVGDYY	3
Sb90	AAARWGRQYPLTFVYYSY	2
Sb93	AAARWGRTYPLSYMAYTY	2
Sb94	AAARWGRYEPLHYAYYSY	2
Sb95	AAASYGANWPLVSAAYTY	2
Sb97	AAARYGHAQAPLHYFWYGY	2
sdAb-1E2	AAQDSAYIKSKGSRAYEY	4
sdAb-2F2	AAHHIPTKHPAFPDFRDY	4
sdAb-3F11	AAEAFVQSPYSGSHTTKY	4
sdAb-4D8	AADQYEWWVPGEVGPYLY	4
sdAb-5F8	AAHYEFNDFVWQGYSSDY	4
SR1	YVYVGFSY	3
SR31	AVMVGFWY	4
SR38	AVHVGQTY	4
SR4	YVWVGHTY	3
SR5	YVYVGTSY	3
SR7	YVYVGYSY	3
US10822379B1-#5	VRLPMIKKSFDI	4
US10822379B1-#6	ARVWLYGSGYMDV	4
US10822379B1-#7	AKDVSYHADV	4
US10822379B1-#8	ARDNLGYRPSENLYGMDV	4
US10822379B1-#9	ARGGITGTTPIDY	4
WuN1	YTERWKPRGIERD	4

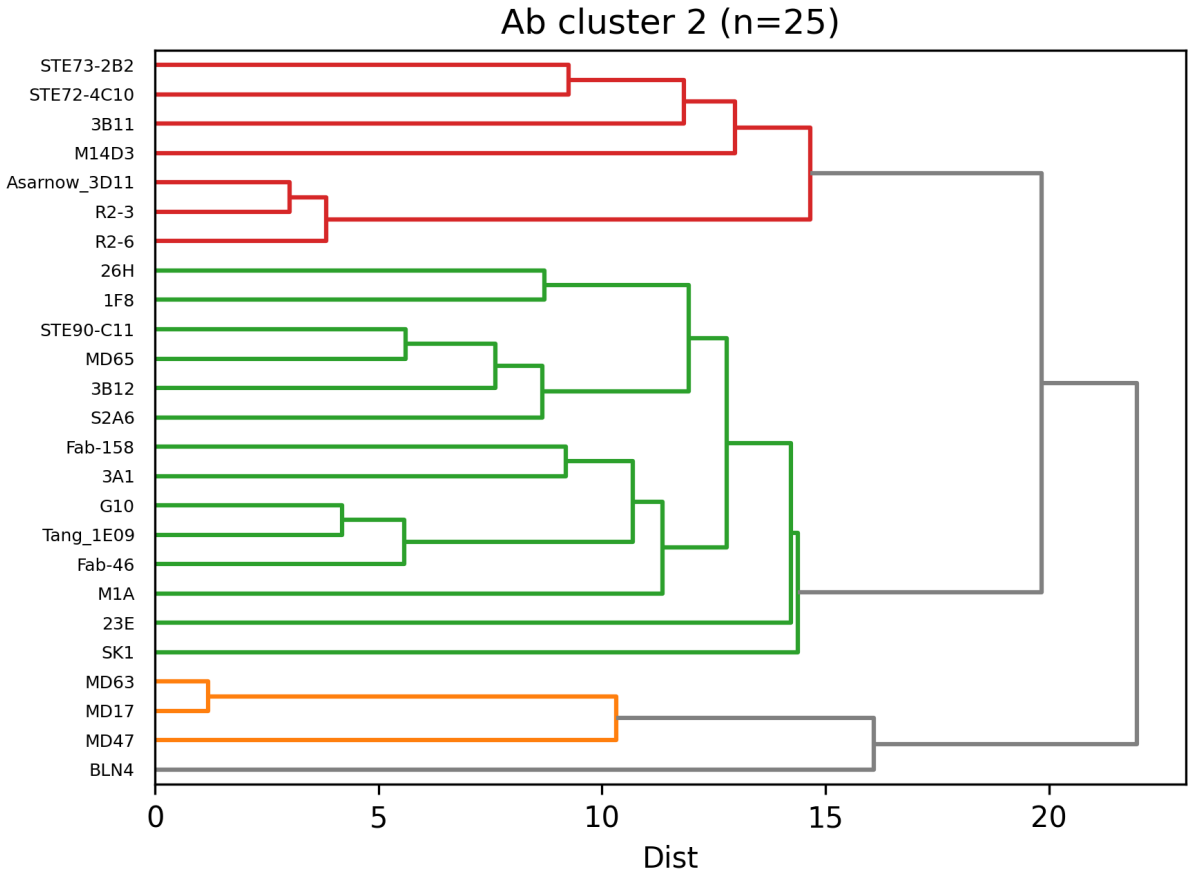


Figure B.3 Detailed dendrogram of Antibody Cluster 2

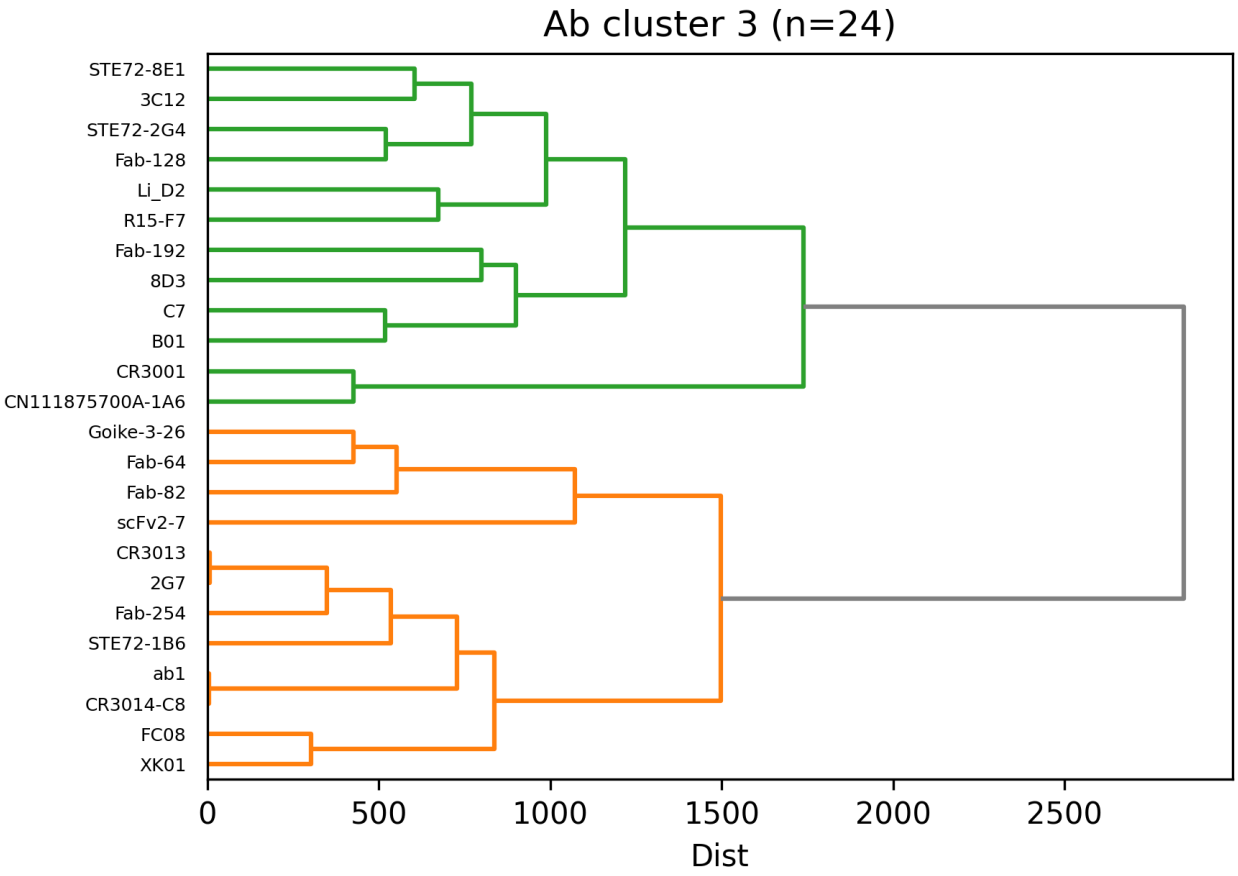


Figure B.4 Detailed dendrogram of Antibody Cluster 3

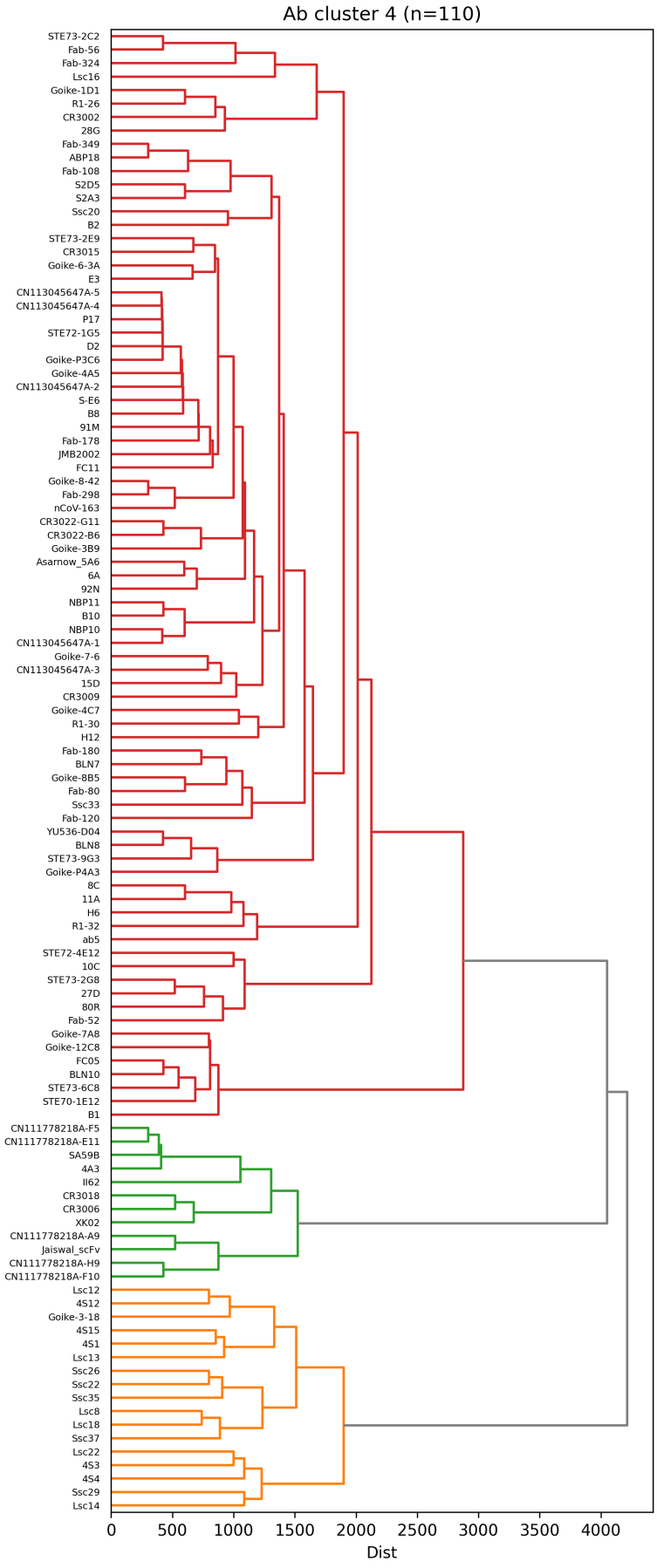


Figure B.5 Detailed dendrogram of Antibody Cluster 4

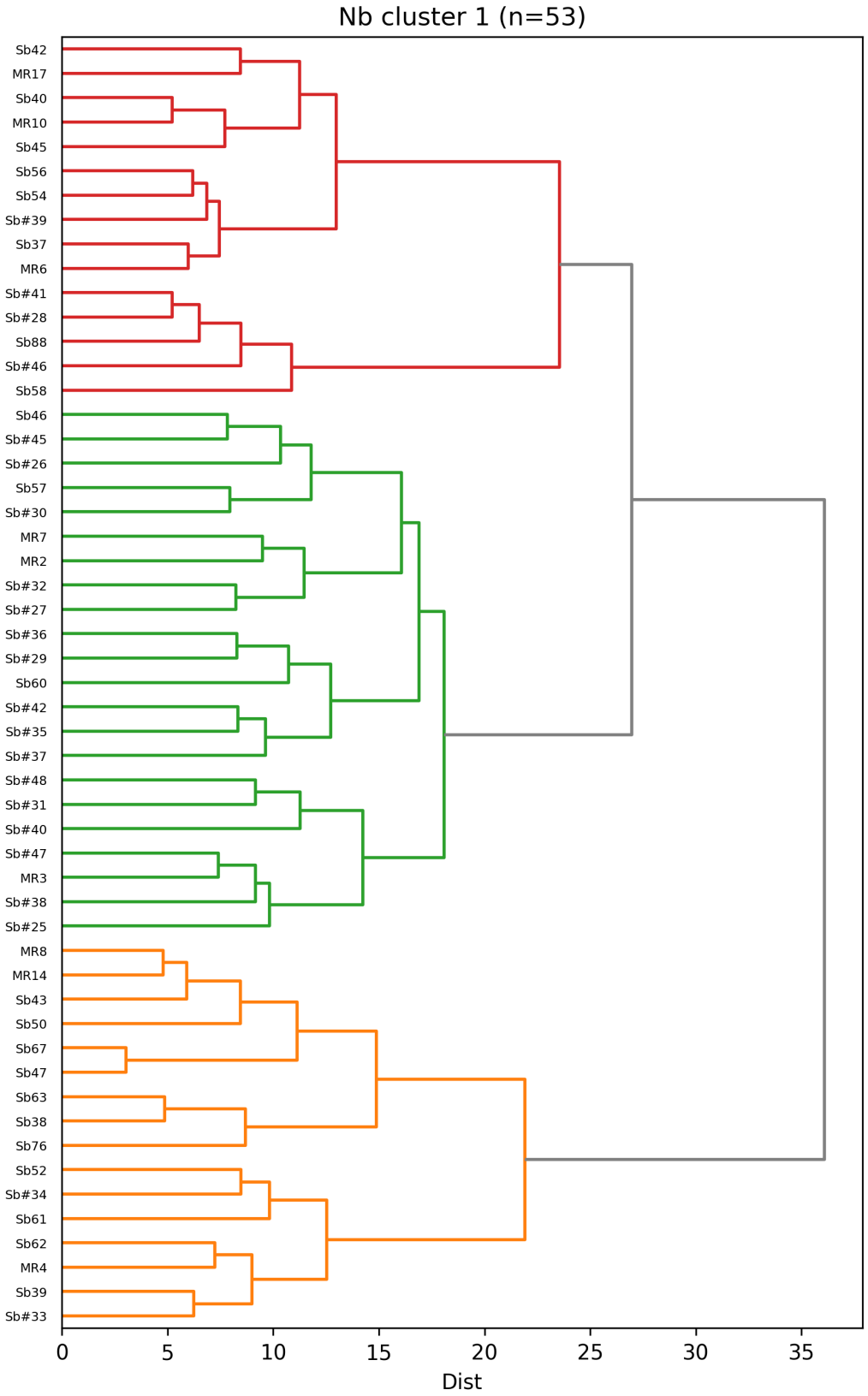


Figure B.6 Detailed dendrogram of Nanobody Cluster 1

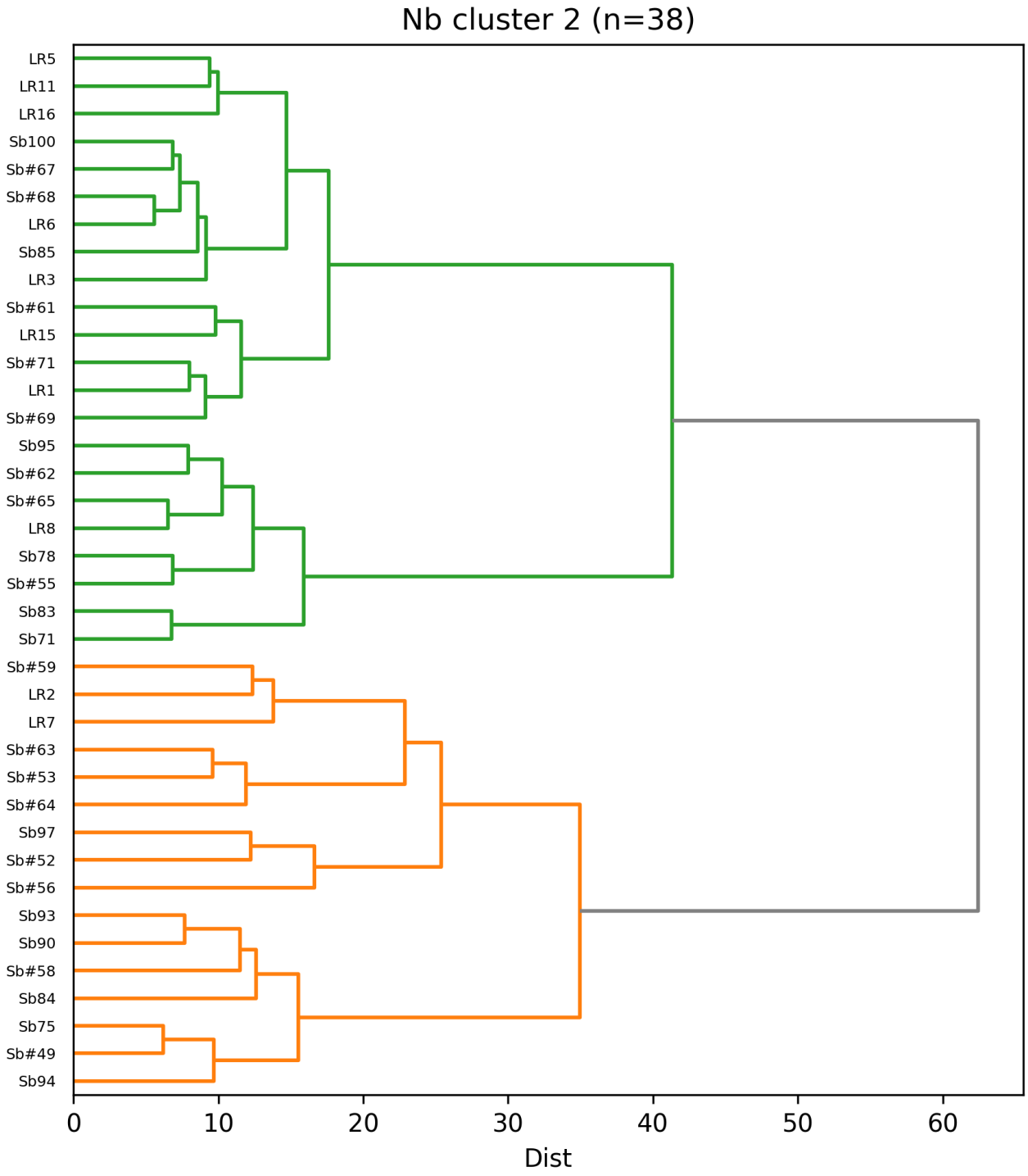


Figure B.7 Detailed dendrogram of Nanobody Cluster 2

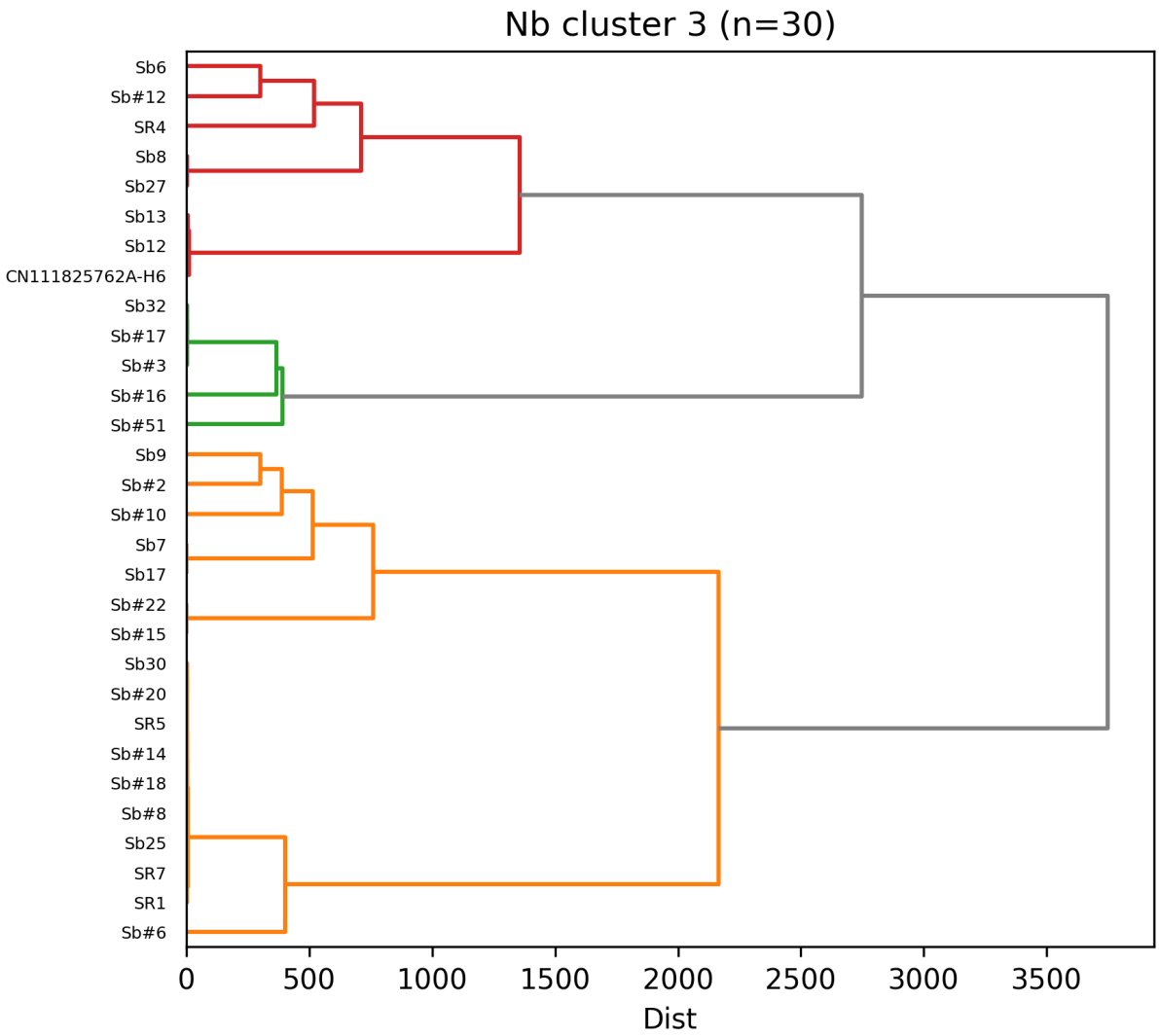


Figure B.8 Detailed dendrogram of Nanobody Cluster 3

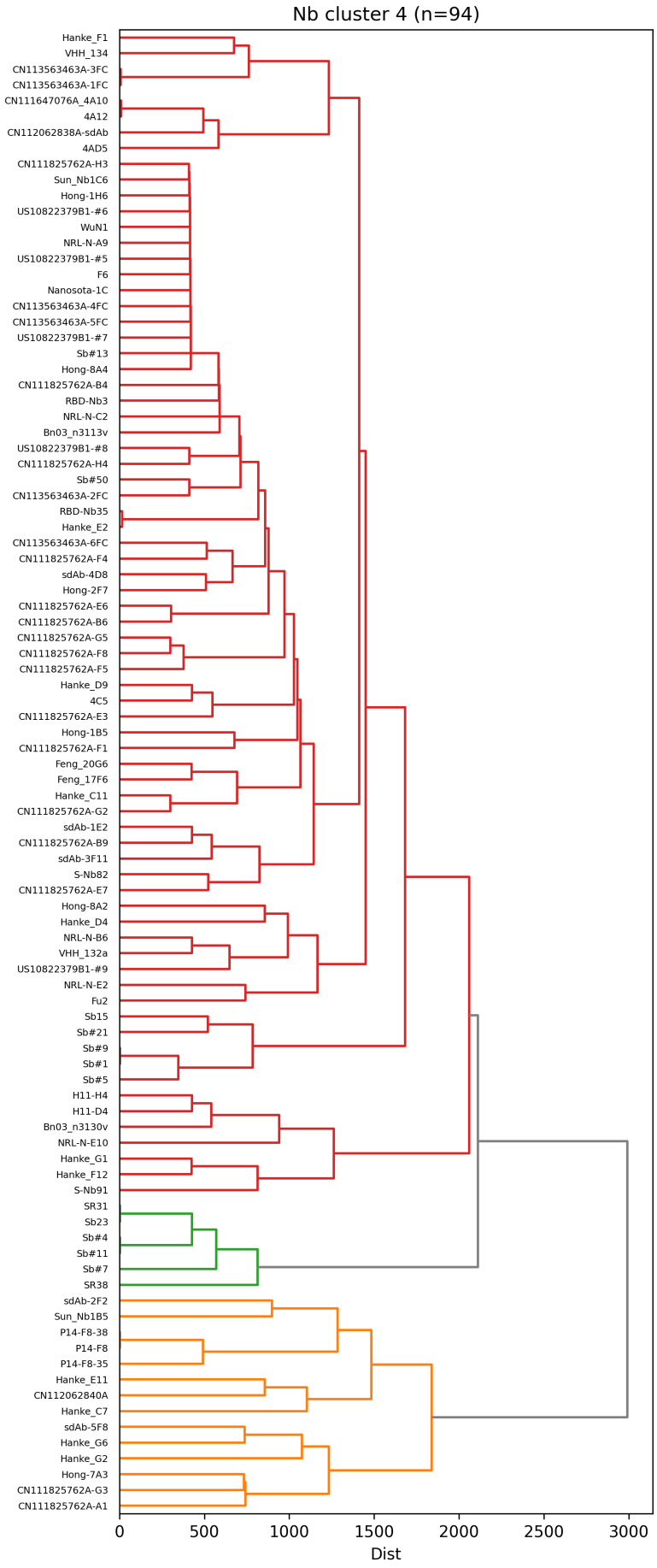


Figure B.9 Detailed dendrogram of Nanobody Cluster 4

Appendix C University of Malta Sanger Clustering

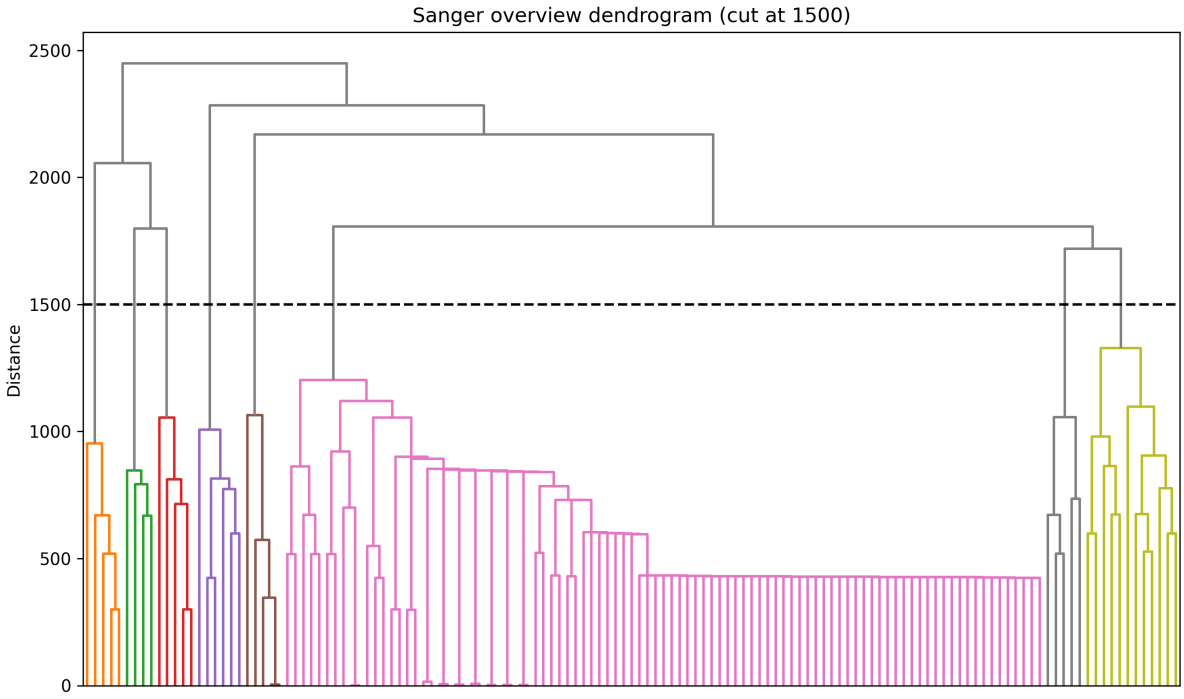


Figure C.1 Dendrogram of 137 Sanger CDRH3 sequences

Table C.1 Full list of sequences grouped by cluster for Sanger sequences only.

Name	Sequence	Cluster
BA5_31	EPSYVRHHTTGIDL	8
BA5_33	DRVWGHRH	6
BA5_34	DGSWHLGN	6
BA5_35	GGTFGYMLD	6
BA5_36	DRGTNIYGYPQFFDQ	6
BA5_37	WGRGYTQGSV	6
BA5_39	RSSWYYPVH	6
BA5_40	AYDRHYGGSVFGH	8
BA5_43	GRWSNLEY	5
BA5_45	TQMHAACGH	6
BA5_49	AHQAGWDYNSYEANFDY	1
BA5_51	RRSGGGGNFDY	1
BA5_53	TSRYARGWNVDH	8
BA5_56	HPLPARKQLAEAFAH	6
BA5_60	EPTSRGSLEY	6
BA5_62	SHMAVGLGH	6
BA5_64	EGMWIPIDY	6
BA5_65	GAHHNLNY	6
BA5_68	ESVHGYIYGYRDY	6
BA5_71	GGHSDNYYLGYNLEY	5
BA5_77	GRGSPSH	6

Continued on next page

Table C.1 – continued from previous page

Name	Sequence	Cluster
BA5_88	EEINERKRGLFTHYYGHFDH	6
BA5_89	EGKYGYVSIDH	6
BA5_90	GGYHH	6
T3.1.M13-27R	DPIPWVTSALGY	6
T3.7.M13-27R	GGDDY	6
DELTA.40.M13-29R	DGVIGYADNFDY	1
BA1.1.3.M13-29R	DDGWGYSKY	6
BA1.1.44.M13-29R	SSNIYWGPSESEY	6
VISTA4.31.M13-29R	DYWDNAFGY	6
BA1.1.53.M13-29R	DVSSGWGGFYYNMDY	2
DELTA.33.M13-29R	RSSWYYPIH	6
DELTA.54.M13-29R	DGDELAPDY	6
BA1.1.46.M13-29R	DPWDTSLA	6
BA1.1.31.M13-29R	APPGHTRLFDN	6
DELTA.26.M13-29R	APATSAGWGGGGY	2
BA1.1.9.M13-29R	LVRHAFGH	6
BA1.1.50.M13-29R	AFDY	6
BA1.1.41.M13-29R	DDGCGYSKY	6
VISTA4.37.M13-29R	EGDDK	6
DELTA.39.M13-29R	WDAEGIQY	6
MSNFR5.1.M13-29R	DAVRYDSYYGVDY	8
PDNFR4.16.M13-29R	GKWFDN	6
MSNFR4.18.M13-29R	DFFGY	6
PDNFR4.1.M13-29R	GSDLAY	6
MSNFR4.5.M13-29R	EITAHY	6
CEA5FR4.6.M13-29R	LGIRPPDLGY	6
CEA5FR4.14.M13-29R	APGTSRGWGGGGF	2
CEA5FR4.22.M13-29R	ARSSGWAWTSSMDY	6
CEA5FR4.38.M13-29R	YCLDDGCFDALNFDY	1
CEA5FR4.46.M13-29R	GDYYDSAEY	8
DELTA.3.M13-29R	DSSASGVDFDY	4
CEA5FR4.54.M13-29R	GGLEY	3
CEA5FR4.15.M13-29R	AVEDYDSFYQVGY	6
CEA5FR4.7.M13-29R	DGHYDRATYDFED	6
CEA5FR4.31.M13-29R	GFWGNWGHPDLES	6
CEA5FR4.23.M13-29R	NGNSE	6
CEA5FR4.39.M13-29R	PREETYDEY	6
CEA5FR4.55.M13-29R	RGSGSGYLEY	6
CEA5FR4.47.M13-29R	TSRNHDFDY	4
CEA5FR4.8.M13-29R	PPLDL	6
CEA5FR4.16.M13-29R	STWGPPGFDF	6
CEA5FR4.32.M13-29R	GGSTSWGYDLDY	7
CEA5FR4.24.M13-29R	LPYVTSWHDLDY	4
CEA5FR4.40.M13-29R	DRDSLGNYYGMDY	8
CEA5FR4.56.M13-29R	DHSV	6
CEA5FR4.48.M13-29R	DWNAFDY	6
CEA5FR4.9.M13-29R	GPYSSNSYDPEY	6
CEA5FR4.17.M13-29R	GSSIGWDHPEY	6
CEA5FR4.1.M13-29R	DLGGHSYL	6
CEA5FR4.25.M13-29R	ESYGFLN	6
CEA5FR4.49.M13-29R	DIGWDRLQS	6
CEA5FR4.41.M13-29R	HTFDY	6
CEA5FR4.57.M13-29R	ELGGAIADY	7

Continued on next page

Table C.1 – continued from previous page

Name	Sequence	Cluster
CEA5FR4.2.M13-29R	SWYGYGEY	8
CEA5FR4.18.M13-29R	DLPGSGFDY	6
CEA5FR4.26.M13-29R	GAYGDTYVMRGLEY	3
CEA5FR4.10.M13-29R	DRYDSPYNFEY	8
CEA5FR4.42.M13-29R	NGYYDSYGLEY	3
CEA5FR4.34.M13-29R	PYRSSWYGRNGDFDY	4
CEA5FR4.58.M13-29R	QWGYGAELEY	8
CEA5FR4.50.M13-29R	GGGY	6
CEA5FR4.3.M13-29R	WASGTVEY	6
CEA5FR4.19.M13-29R	AGEGIPGY	6
CEA5FR4.27.M13-29R	GFGGSDY	6
CEA5FR4.51.M13-29R	SARDYAGYATDY	7
CEA5FR4.59.M13-29R	GVRERNFADFDY	4
CEA5FR4.43.M13-29R	DHSL	6
CEA5FR4.4.M13-29R	GGYYGTSYVDFDY	4
CEA5FR4.28.M13-29R	AHPSVFLEY	6
CEA5FR4.36.M13-29R	DHYTFDY	6
CEA5FR4.12.M13-29R	SSYYGNYYAEGDL	8
CEA5FR4.44.M13-29R	GTSGWGGGLEY	3
CEA5FR4.52.M13-29R	EWWGALEY	6
CEA5FR4.20.M13-29R	RGGSNWFGEY	6
CEA5FR4.60.M13-29R	KGWGGVSLEY	2
CEA5FR4.13.M13-29R	EGLIHMDTDNFES	6
CEA5FR4.5.M13-29R	PGSSYPWG	6
CEA5FR4.21.M13-29R	DSMWVRSSDYNLEY	5
CEA5FR4.29.M13-29R	QGHDSHYEDFEY	6
CEA5FR4.37.M13-29R	AVGGYYDNLEY	5
CEA5FR4.53.M13-29R	LRAVDIGY	6
GITR24.11.M13-29R	GPYNGH	6
CEA5FR4.45.M13-29R	STNANDVYPFNLDY	6
GITR24.03.M13-29R	VPGLDNFDS	6
GITR24.35.M13-29R	GGGTILPD	6
GITR24.43.M13-29R	GSPYYESYES	6
CEA5FR4.67.M13-29R	EIVATGRGALSDLEF	6
CEA5FR4.75.M13-29R	PGSYQY	6
CEA5FR4.76.M13-29R	ARDGGDY	6
CEA5FR4.68.M13-29R	PLSGEY	6
GITR24.53.M13-29R	EAWNTYGPVY	6
GITR24.45.M13-29R	GSSYDRWSGYDMDY	7
CEA5FR4.77.M13-29R	SVYHDSSTPEY	6
CEA5FR4.85.M13-29R	GGGNWYDGLEY	3
CEA5FR4.61.M13-29R	GPLTMATTFPEY	6
GITR24.22.M13-29R	IAAWKDEY	6
GITR24.54.M13-29R	DRDNLEY	5
GITR24.07.M13-29R	DAYSGRGNPEY	6
CEA5FR4.78.M13-29R	GGESREPALAY	6
GITR24.23.M13-29R	GVVGTYGDLEY	8
CEA5FR4.79.M13-29R	STGTGRDDLLY	6
CEA5FR4.87.M13-29R	QDWKY	6
CEA5FR4.63.M13-29R	ESPGGYGMDY	7
CEA5FR4.71.M13-29R	GDWGDS	6
CEA5FR4.64.M13-29R	DPPRGFDY	6
CEA5FR4.72.M13-29R	HNDAKY	6

Continued on next page

Table C.1 – continued from previous page

Name	Sequence	Cluster
CEA5FR4.80.M13-29R	GEGRNWFNIES	6
GITR24.56.M13-29R	RWGAAH	6
CEA5FR4.88.M13-29R	DPGYGTYTYNFDQ	6
CEA5FR4.65.M13-29R	EAVSKY	6
CEA5FR4.81.M13-29R	SPWLATDGY	6
CEA5FR4.89.M13-29R	GGANFDY	1
CEA5FR4.73.M13-29R	TPSSWAEGLFDY	6
GITR24.50.M13-29R	PGYYDRYYGELEY	8
CEA5FR4.82.M13-29R	VGRVTAAPLD	6
CEA5FR4.74.M13-29R	GTTTASEY	6

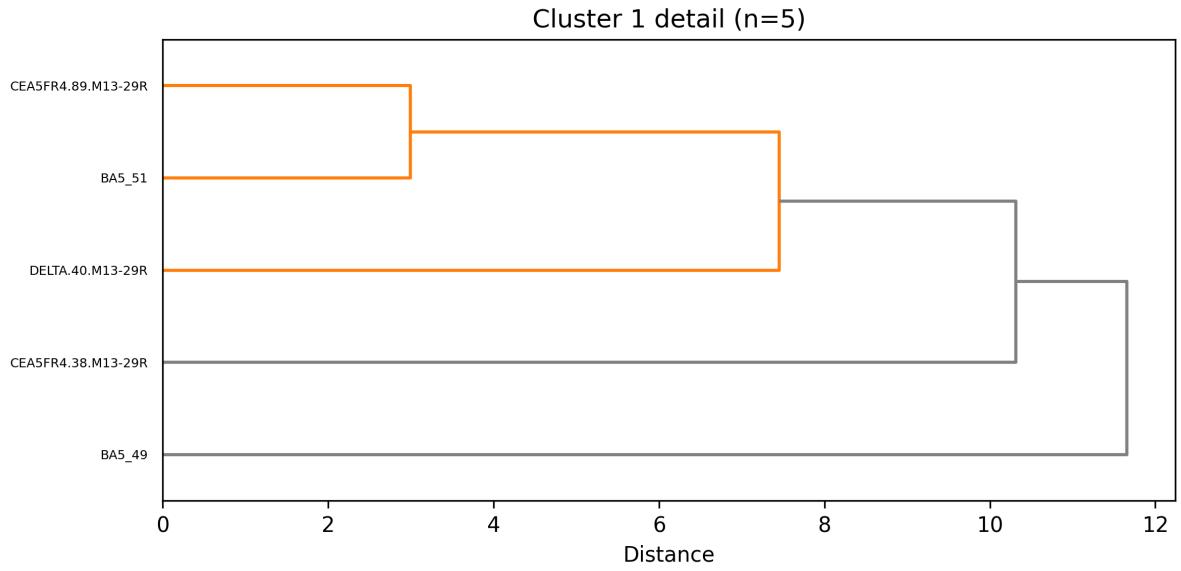


Figure C.2 Detailed dendrogram for Sanger Cluster 1

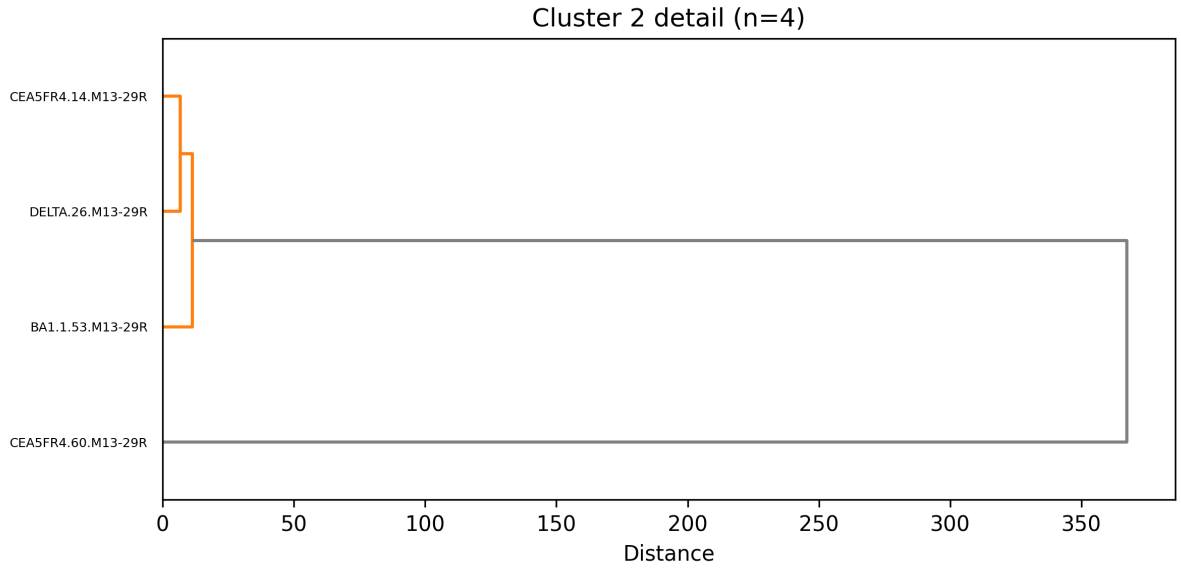


Figure C.3 Detailed dendrogram for Sanger Cluster 2

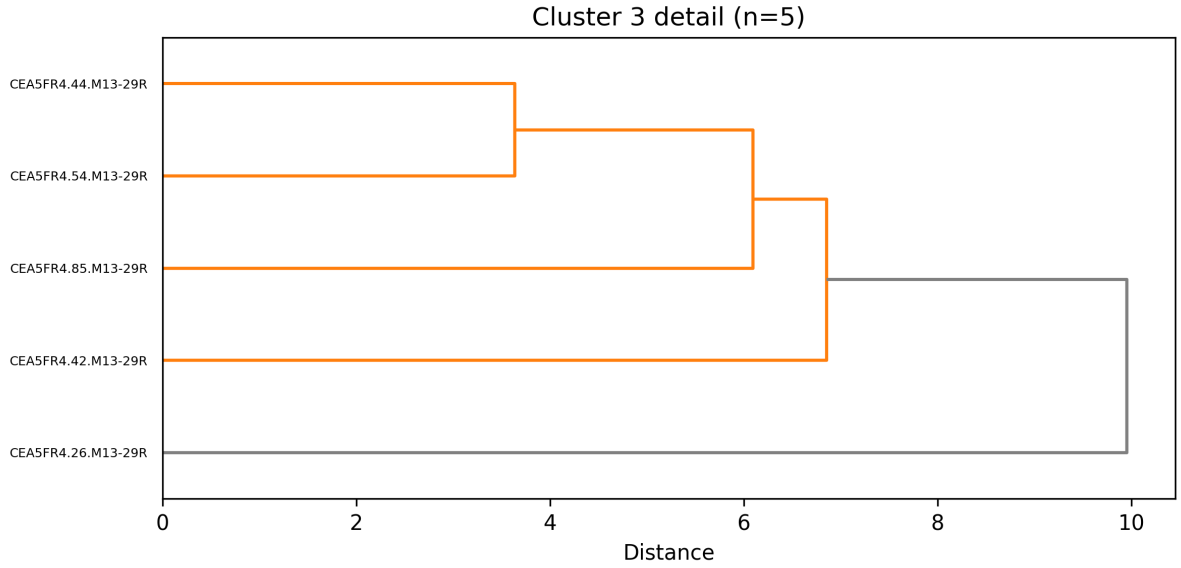


Figure C.4 Detailed dendrogram for Sanger Cluster 3

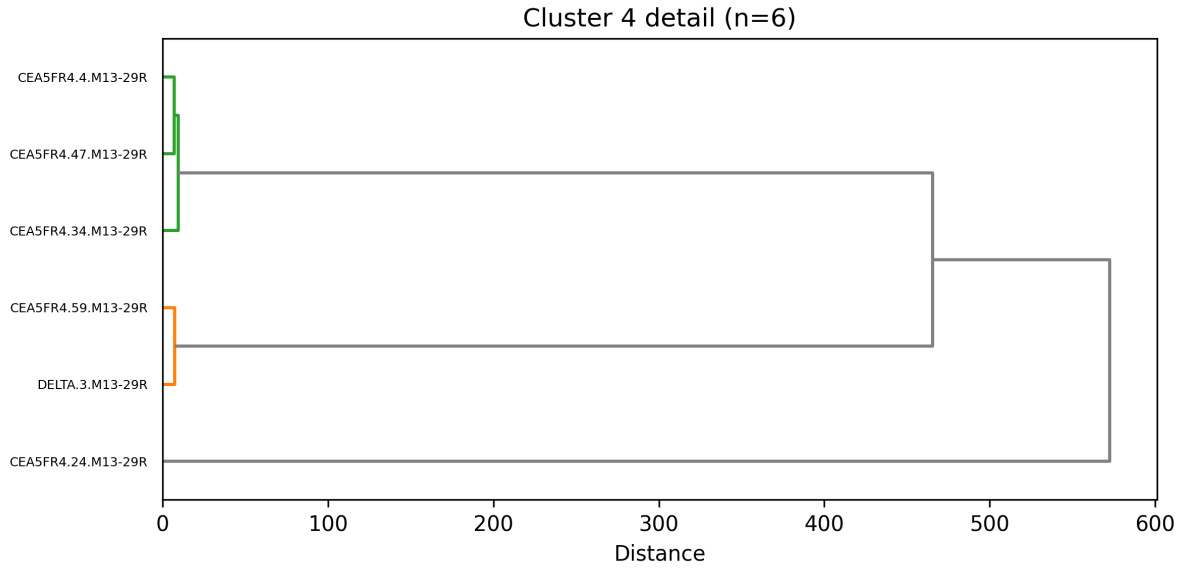


Figure C.5 Detailed dendrogram for Sanger Cluster 4

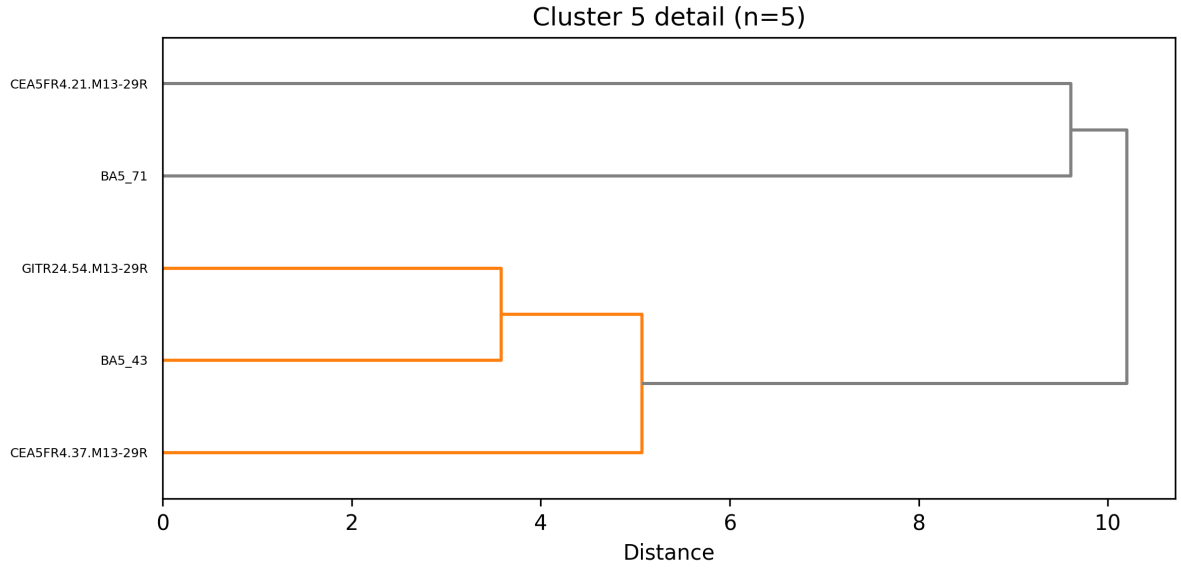


Figure C.6 Detailed dendrogram for Sanger Cluster 5

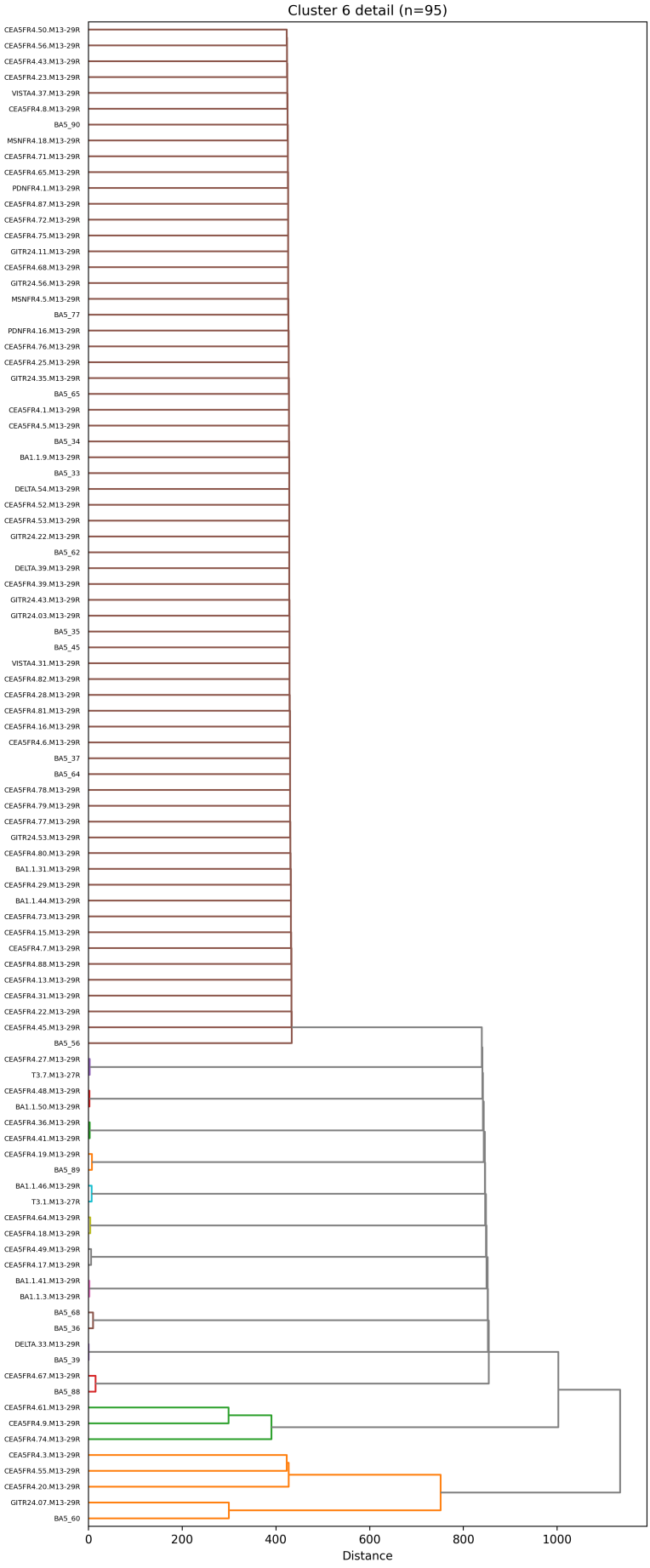


Figure C.7 Detailed dendrogram for Sanger Cluster 6

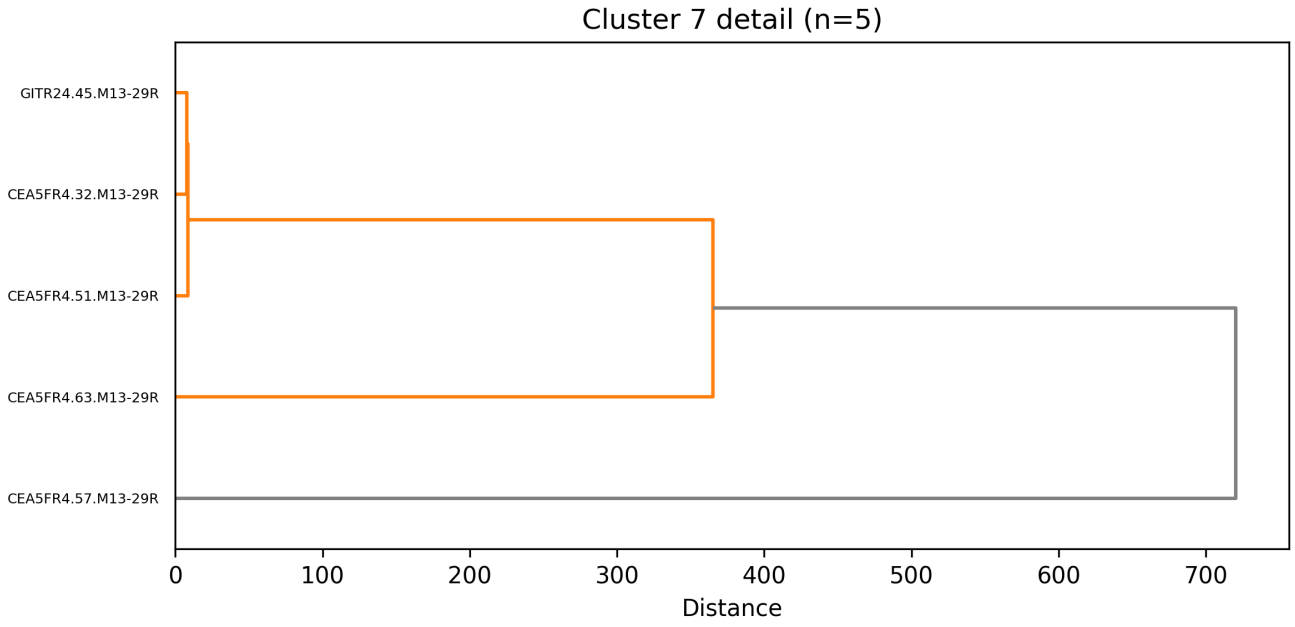


Figure C.8 Detailed dendrogram for Sanger Cluster 7

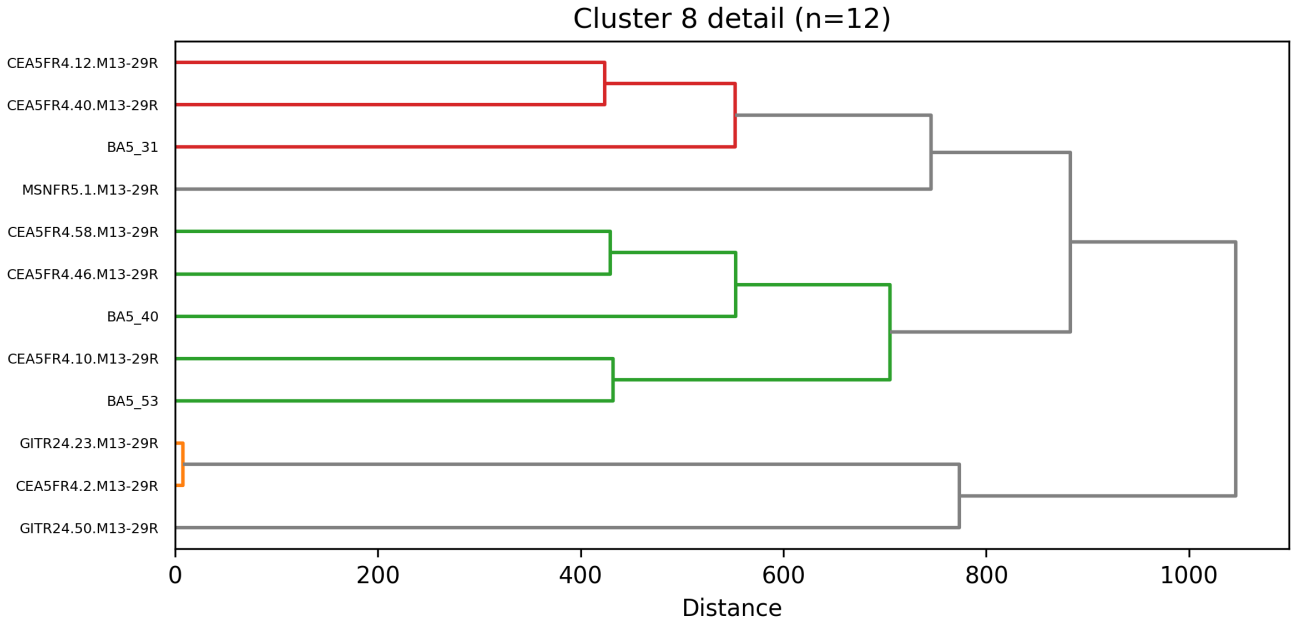


Figure C.9 Detailed dendrogram for Sanger Cluster 8

Appendix D Combined OPIG and Sanger Clustering

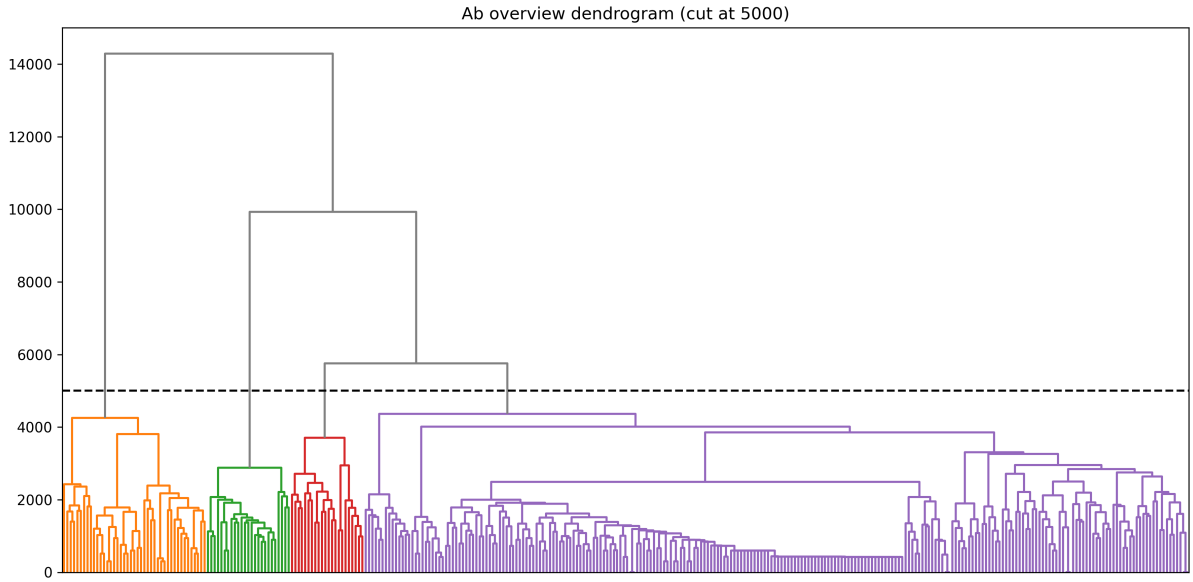


Figure D.1 Overview dendrogram of combined CoV-AbDab and Sanger CDRH3s.

Table D.1 Full list of antibody sequences grouped by cluster for Combined Dataset

Name	Sequence	Cluster
scFv2-7	ARISGSGYFYPFDI	3
1F	ARHVRSAYYYGSGSYRDEGNWFDP	1
Jaiswal_scFv	AKTTTAFDY	4
Li_D2	AKDVYSESGSGSYYDY	3
R1-26	ARGQLGPWVGVDY	4
R1-30	ARQGWLRGNFDY	4
R2-3	ASQLWLRGAFDI	2
R2-6	ARKGWLRGAFDI	2
R1-32	ARENGYSGYGAAANFDL	4
S2B2	ARAYTGSYYYGMDV	1
S2B5	ARARGGSYYYGMDV	1
S2B7	ARSRGGGYYYGMDV	1
S2E6	ARAHRGSYYYGMDV	1
S2E7	ASDVAGHHGMDV	1
S2F6	ARANWGSYYYGMDV	1
XK01	ARERGYSGYGAAYYFDY	3
XK02	ASWLYGDPISFDY	4
CR3022-B6	AGGSGISTPMDV	4
FC08	VRERGYSGYGAAYYFDY	3
CR3014-C8	ARGISPFYFDY	3
P17	ARHATLMNNKDI	4
BD-494	ARDLVVYGMDV	1
WIBP-2B11	ARDLMEVGGMDV	1
256	ARDGYGSGSDYYYYYYMDV	1
Tang_1E09	ARSGDSDAFDI	2

Continued on next page

Table D.1 – continued from previous page

Name	Sequence	Cluster
10C	ARQLSYCSTGVSCPGAI	4
11A	ARQGDYSGPSINY	4
12E	ARVGYYGDYAWGYYYYGMDV	1
15D	KGIYDVTGSSFDS	4
1F8	ARDRWMTTRAFDI	2
23E	ARARPSDPYDGSGFDAFDI	2
26H	ARERWLQIGEDAFDI	2
27D	ARNFGEDFDY	4
28G	ARLRPNSSGWTDY	4
2G7	AKATTVTTYFDY	3
3A1	ARDQGISANFKDAFDI	2
3B11	ARVGYCSSTSCHIGAFDI	2
3B12	STTSCASGAFDI	2
3C12	ARDYYGSGARGFDY	3
4A3	AKDADSFDY	4
4S1	AKAADTNYGWRTAGSIDA	4
4S12	AKAAGSCNYGYSCSYYIDT	4
4S15	AKYSGSWTAYAYITAGSIDA	4
4S3	AKSVGGGGWHAATIDA	4
4S4	AKDSTNTGGCGGYDIDA	4
6A	AKDYGEGFDY	4
80R	ARDRSYYLDY	4
8C	ARDWGYSFGD	4
8D3	ARRPLYDGDYGYPMDY	3
91M	ARDQGWGWDGTEYYSDY	4
92N	ARDDLSDYGEWLGPDY	4
A10	ARGGNGGMDV	1
ab1	ARGYGDYYFDY	3
ab4	ARETVSYGMDV	1
ab5	ARDRGYSSGWTDGFDI	4
ABP18	GAHIAAEYFQH	4
Asarnow_3D11	ARRWWLRGAFDI	2
Asarnow_5A6	ARLITMVRGEDY	4
B01	ARWDFASPYYPGSSGLDY	3
B1	ARGVAVAGTWDWFDP	4
B10	TSVCSGGSCYQ	4
B2	AKVGEVGSREWSAFDV	4
B8	STDSGSIGEF	4
BLN1	VTAPAITGSPEAYSYYYGMDV	1
BLN10	ATGPAIAAAATGWFDP	4
BLN12	VTAPVITGSPEAYSYYYGMDV	1
BLN14	AKDHDDGYYFYYYMDV	1
BLN2	AASPAVRGSPSNFYYYHGMDV	1
BLN3	VAAPVITGSPEAYSYYYGMDV	1
BLN4	ARMAYQVYYYDSSGYYDAFDI	2
BLN7	ATSRVAGTPNWFHP	4
BLN8	ARDLGSGWYP	4
C7	TRISGYYGAGSGGAMDV	3
CN111778218A-A9	AKGTDAFDY	4
CN111778218A-E11	AKNSDSFDY	4
CN111778218A-F10	AKDSDTFDY	4
CN111778218A-F5	AKNDSSFDY	4
CN111778218A-H9	AKSTNTFDY	4

Continued on next page

Table D.1 – continued from previous page

Name	Sequence	Cluster
CN111875700A-1A6	ARDALGWYFDV	4
CN111875701A	ARQGSHYGMDV	1
CN112250763B_Ab1	AREHTVAPVYGFDV	1
CN113045647A-1	AAFPGMDDDSVFNY	4
CN113045647A-2	AASTQDPGYMDFTTEY	4
CN113045647A-3	AAGFLDSSIMQRIVVGYATDY	4
CN113045647A-4	AEGRDSWWPSHYTMVPQRKYNAY	4
CN113045647A-5	AANYYHLFVMHYQWY	4
CR3001	ARHRFRHVFDY	4
CR3002	ARYYSRSLKAFDY	4
CR3006	AKDGSPRTPSFDY	4
CR3009	AKGLFMVTTYAFDY	4
CR3013	AKGLTPLYFDY	3
CR3015	ARGLSLRP	4
CR3018	AKFNPFTSFDY	4
CR3022-G11	AGGDGVSTPMDV	4
D2	FTATFAMDY	4
E3	ARHNAQFGELLVPQDAFDM	4
Fab_15033	ARSYYYGGFGMDY	1
Fab-108	AKDGSQLAYLVEYFQH	4
Fab-120	AKDFGGGTRYDYWYFDL	4
Fab-128	ARDGRYGSGSYPFDY	3
Fab-158	ARDPGGSYSNDAFDI	2
Fab-160	ARANSLRYYYGMDV	1
Fab-178	ARDISSWYEITKFDP	4
Fab-180	AREAVAGTHPQAGDFDL	4
Fab-192	STYYYDSSGYSTDY	4
Fab-236	ASRGIQLLPRGMDV	1
Fab-254	AREGDGYNFYFDY	3
Fab-298	ASDPRDDIAGGY	4
Fab-324	ARVGDYGDYIVSPFDL	4
Fab-349	AGNHAGTTVTSEYFQH	4
Fab-368	ARGSSGYYYG	1
Fab-46	ARGDSRDAFDI	2
Fab-52	ARDRGDTIDY	4
Fab-56	ARDIGPIDY	4
Fab-64	ARDTYGGKVTYFDY	3
Fab-80	ARSTRELPEVVDWYFDL	4
Fab-82	ARSRALYGSGSYFDY	3
Fab20	ARGGWSSSAGGYYGMDV	1
FC05	ATTTPFSSSYWFDP	4
FC11	AKAMFLGDSSGLTGLDMDV	4
G10	ARSGWDDAFDI	2
Goike-12C8	ATGPAVRRGSWFDP	4
Goike-1D1	ARIPIATHLGSDY	4
Goike-3-18	AKQAGAYCSGGSCYSSSEADY	4
Goike-3-26	ARPYSGSYWGYFDY	3
Goike-3B9	ARDGGGYVSY	4
Goike-4A5	AKASQLFWLGQFTRDGFDI	4
Goike-4C7	ATAAAVRGRGTIDY	4
Goike-6-3A	ARGTIYFDRSGYRRVDPFHI	4
Goike-7-6	ATRFAVYGDYLIDY	4
Goike-7A8	ATGSPFDRTQNWFDP	4

Continued on next page

Table D.1 – continued from previous page

Name	Sequence	Cluster
Goike-8-3	ARGGVVDTYYYYGMDV	1
Goike-8-42	ARDYGRGGV	4
Goike-8-96	AKAPGQWLRFHYYGMDV	1
Goike-8B5	ARELPPGRMVVPATYWHFDL	4
Goike-P3C6	APGRSLY	4
Goike-P4A3	ARDDTGRVGSGWYCPLY	4
H12	ARGGWCTGDCDARTFVWFEP	4
H6	ARGGGYDPWFAY	4
II62	AKAAGSFDY	4
JMB2002	ASLASYSSGWEDVFDI	4
Lsc12	AKAAGPDCCYTASNIDA	4
Lsc13	AKGSSGSCGSCAGNIDA	4
Lsc14	AKSSYCGGGYYSAANIDT	4
Lsc16	ARSSCGGDYETGCIDA	4
Lsc18	AKSGFRNGGGWSSAGLIDA	4
Lsc22	AKTIYGGGWSGYGDSIDG	4
Lsc8	AKESGAGGWNAGNVIDA	4
M14D3	ASSNYYGSGSYYPRSAFDI	2
M1A	ARDPVVVVINGDEAFDI	2
m396	ARDTVMGGMDV	1
m396-B10	ARDTATGGMDV	1
MD17	VKDQDSSSWYDAFDI	2
MD45	ARDLSVRGGMDV	1
MD47	AKDLVTAPSYEAFDI	2
MD62	ARDLQYYGMDV	1
MD63	VKDQDSNSWYDAFDI	2
MD65	ARDLAVAGAFDI	2
N18	ARGYWGSGYHYYGMDV	1
NBP10	YYAGGGFDV	4
NBP11	FSVGPGGPFDS	4
nCoV-163	ARSYGDFYVDF	4
nCoVmab1	ARGDGSDDYYYGMDV	1
P16-A3	ARAYSSSWLLQSFYYYGMDV	1
R15-F7	ASGAFYYGSGSYPFDY	3
R3P1-A12	ARDLSEKGGMDV	1
Regdanvimab	ARIPGFLRYRNRYYYYGMDV	1
S-B8	AREYYYGMDV	1
S-E6	ATPGAIMGALHI	4
S2A3	AKDQYVSTDFDI	4
S2A6	AKHLYGSWAFDI	2
S2D5	AKVSSQTLRFDY	4
SA59B	AKRSLSFDY	4
SK1	VKDIYYRDRNLGFAFDI	2
Ssc20	AKSVNNSWSTGEDIDV	4
Ssc22	AKNNYNNGVDAAGDIDA	4
Ssc26	AKGGNGCSSGDHAGQIDA	4
Ssc29	ARSPGGAYSGSIDT	4
Ssc33	ARGAPGCDTWCWYGAAFIDA	4
Ssc35	AKGSGSACIWSGWCAGDIDS	3
Ssc37	AKSAYGGWTYADNIDA	4
STE70-1E12	VRDGYNFNNWFDP	4
STE72-1B6	AKAPYGDFRGLWYFDY	3
STE72-1G5	AVGGVQLWLTY	4

Continued on next page

Table D.1 – continued from previous page

Name	Sequence	Cluster
STE72-2G4	ARFFYDSSGYSTDY	4
STE72-4C10	AREYSSSWYGLGAFDI	2
STE72-4E12	ARDLSGGLDY	4
STE72-8E1	ARGGPKRSGSPFDV	3
STE73-2B2	ARVSGWYFGAFDI	2
STE73-2C2	ARGHDNLDY	4
STE73-2E9	ARGKFDY	4
STE73-2G8	ARWSGTYYDY	4
STE73-6B10	ARDRLRYGDSGSYYYYGMDV	1
STE73-6C1	ARSYVGGMDV	1
STE73-6C8	ARSIAALNWFDP	4
STE73-9G3	ARDLVLGSGSSND	4
STE90-C11	ARDVADAFDI	2
YU536-D04	ARDLVVMGLDV	4
YU537-H11	ARGESGSPYGMDV	1
T3.1.M13-27R	DPIPWVTSALGY	4
T3.7.M13-27R	GGDDY	4
DELTA.40.M13-29R	DGVIGYADNFDY	4
BA1.1.3.M13-29R	DDGWGYSKY	4
BA1.1.44.M13-29R	SSNIYWGPSESEY	4
VISTA4.31.M13-29R	DYWDNAFGY	4
BA1.1.53.M13-29R	DVSSGWGGFYYNMDY	1
DELTA.41.M13-29R	GGYHH	4
DELTA.33.M13-29R	RSSWYYPIH	4
DELTA.54.M13-29R	DGDELAPDY	4
BA1.1.46.M13-29R	DPWDTSLA	4
BA1.1.5.M13-29R	RSSWYYPVH	4
BA1.1.31.M13-29R	APPGHTRLFDN	4
DELTA.26.M13-29R	APATSAGWGGGGY	4
BA1.1.8.M13-29R	ESVHGYIYGYRDY	4
BA1.1.9.M13-29R	LVRHAFGH	4
BA1.1.50.M13-29R	AFDY	4
BA1.1.41.M13-29R	DDGCGYSKY	4
VISTA4.37.M13-29R	EGDDK	4
DELTA.39.M13-29R	WDAEGIQY	4
MSNFR5.1.M13-29R	DAVRYDSYYGVDY	4
PDNFR4.16.M13-29R	GKWFDN	4
MSNFR4.18.M13-29R	DFFGY	4
PDNFR4.1.M13-29R	GSDLAY	4
MSNFR4.5.M13-29R	EITAHY	4
CEA5FR4.6.M13-29R	LGIRPPDLGY	4
CEA5FR4.14.M13-29R	APGTSRGWGGGGF	4
CEA5FR4.22.M13-29R	ARSSGWAWTSSMDY	4
CEA5FR4.38.M13-29R	YCLDDGCFDALNFDY	4
CEA5FR4.46.M13-29R	GDYYDSAEY	4
DELTA.3.M13-29R	DSSASGVDFDY	4
CEA5FR4.54.M13-29R	GGLEY	4
CEA5FR4.15.M13-29R	AVEDYDSFYQVGY	4
CEA5FR4.7.M13-29R	DGHYDRATYDFED	4
CEA5FR4.31.M13-29R	GFWGNWGHPDLES	4
CEA5FR4.23.M13-29R	NGNSE	4
CEA5FR4.39.M13-29R	PREETYDEY	4
CEA5FR4.55.M13-29R	RGSGSGYLEY	3

Continued on next page

Table D.1 – continued from previous page

Name	Sequence	Cluster
CEA5FR4.47.M13-29R	TSRNHDFDY	4
CEA5FR4.8.M13-29R	PPLDL	4
CEA5FR4.16.M13-29R	STWGPPGFDF	4
CEA5FR4.32.M13-29R	GGSTSWGYDLDY	4
CEA5FR4.24.M13-29R	LPYVTSWHDLDY	4
CEA5FR4.40.M13-29R	DRDSLGNYYGMDY	1
CEA5FR4.56.M13-29R	DHSV	4
CEA5FR4.48.M13-29R	DWNAFDY	4
CEA5FR4.9.M13-29R	GPYSSNSYDPEY	4
CEA5FR4.17.M13-29R	GSSIGWDHPEY	4
CEA5FR4.1.M13-29R	DLGGHSYL	4
CEA5FR4.25.M13-29R	ESYGFLN	4
CEA5FR4.49.M13-29R	DIGWDRLQS	4
CEA5FR4.41.M13-29R	HTFDY	4
CEA5FR4.57.M13-29R	ELGGAIADY	4
CEA5FR4.2.M13-29R	SWYGYGEY	4
CEA5FR4.18.M13-29R	DLPGSGFDY	4
CEA5FR4.26.M13-29R	GAYGDTYVMRGLEY	4
CEA5FR4.10.M13-29R	DRYDSPYNFEY	4
CEA5FR4.42.M13-29R	NGYYDSYGLEY	4
CEA5FR4.34.M13-29R	PYRSSWYGRNGDFDY	4
CEA5FR4.58.M13-29R	QWGYGAELEY	4
CEA5FR4.50.M13-29R	GGGY	4
CEA5FR4.3.M13-29R	WASGTVEY	4
CEA5FR4.19.M13-29R	AGEGIPGY	4
CEA5FR4.27.M13-29R	GFGGSDY	4
CEA5FR4.51.M13-29R	SARDYAGYATDY	4
CEA5FR4.59.M13-29R	GVRERNFADFDY	4
CEA5FR4.43.M13-29R	DHSL	4
CEA5FR4.4.M13-29R	GGYYGTSYVDFDY	4
CEA5FR4.28.M13-29R	AHPSVFLEY	4
CEA5FR4.36.M13-29R	DHYTFDY	4
CEA5FR4.12.M13-29R	SSYYGNYYAEGDL	4
CEA5FR4.44.M13-29R	GTSGWGGGLEY	4
CEA5FR4.52.M13-29R	EWWGALEY	4
CEA5FR4.20.M13-29R	RGGSNWFGEY	4
CEA5FR4.60.M13-29R	KGWGGVSLEY	4
CEA5FR4.13.M13-29R	EGLIHMDTDNFES	4
CEA5FR4.5.M13-29R	PGSSYPWG	4
CEA5FR4.21.M13-29R	DSMWVRSSDYNLEY	4
CEA5FR4.29.M13-29R	QGHDSHYEDFEY	4
CEA5FR4.37.M13-29R	AVGGYYDNLEY	4
CEA5FR4.53.M13-29R	LRAVDIGY	4
GITR24.11.M13-29R	GPYNGH	4
CEA5FR4.45.M13-29R	STNANDVYPFNLDY	4
GITR24.03.M13-29R	VPGLDNFDS	4
GITR24.35.M13-29R	GGGTILPD	4
GITR24.43.M13-29R	GSPYYESYES	4
CEA5FR4.67.M13-29R	EIVATGRGALSDLEF	4
CEA5FR4.75.M13-29R	PGSYQY	4
CEA5FR4.76.M13-29R	ARDGGDY	4
CEA5FR4.68.M13-29R	PLSGEY	4
GITR24.53.M13-29R	EAWNTYGPVY	4

Continued on next page

Table D.1 – continued from previous page

Name	Sequence	Cluster
GITR24.45.M13-29R	GSSYDRWSGYDMDY	4
CEA5FR4.77.M13-29R	SVYHDSSTPEY	4
CEA5FR4.85.M13-29R	GGGNWYDGLEY	4
CEA5FR4.61.M13-29R	GPLTMATTFPEY	4
GITR24.22.M13-29R	IAAWKDEY	4
GITR24.54.M13-29R	DRDNLEY	4
GITR24.07.M13-29R	DAYSGRGNPEY	4
CEA5FR4.78.M13-29R	GGESREPALAY	4
GITR24.23.M13-29R	GVVGTYGDLEY	4
CEA5FR4.79.M13-29R	STGTGRDDLLY	4
CEA5FR4.87.M13-29R	QDWKY	4
CEA5FR4.63.M13-29R	ESPGGYGMDY	1
CEA5FR4.71.M13-29R	GDWGDS	4
CEA5FR4.64.M13-29R	DPPRGFDY	4
CEA5FR4.72.M13-29R	HNDAKY	4
CEA5FR4.80.M13-29R	GEGRNWFNIES	4
GITR24.56.M13-29R	RWGAAH	4
CEA5FR4.88.M13-29R	DPGYGTYTYNFDQ	4
CEA5FR4.65.M13-29R	EAVSKY	4
CEA5FR4.81.M13-29R	SPWLATDGY	4
CEA5FR4.89.M13-29R	GGANFDY	4
CEA5FR4.73.M13-29R	TPSSWAEGLFDY	4
GITR24.50.M13-29R	PGYYDRYYGELEY	4
CEA5FR4.82.M13-29R	VGRVTAAPLD	4
CEA5FR4.74.M13-29R	GTTTASEY	4
BA5_31	EPSYVRHHTTGIDL	4
BA5_33	DRVWGHRH	4
BA5_34	DGSWHLGN	4
BA5_35	GGTFGYMLD	4
BA5_36	DRGTNIYGYPQFFDQ	4
BA5_37	WGRGYTQGSV	4
BA5_40	AYDRHYGGSVFGH	4
BA5_43	GRWSNLEY	4
BA5_45	TQMHAACGH	4
BA5_49	AHQAGWDYNSYEANFDY	4
BA5_51	RRSGGGGNFDY	4
BA5_53	TSRYARGWNVDH	4
BA5_56	HPLPARKQLAEAFAH	4
BA5_60	EPTSRGSLEY	4
BA5_62	SHMAVGLGH	4
BA5_64	EGMWIPIDY	4
BA5_65	GAHHNLNY	4
BA5_71	GGHSDNYYLGYNLEY	4
BA5_77	GRGSPSH	4
BA5_88	EEINERKRGLFTHYYGHFDH	4
BA5_89	EGKYGYVSIDH	4

Table D.2 Full list of nanobody sequences grouped by cluster for Combined Dataset

Name	Sequence	Cluster
Sun_Nb1C6	AAELFCPWPDIGTMSPAEYKY	4
Sun_Nb1B5	AADSGWVGYSLDPYQYNY	4
F6	ARVESGSGWLDF	4
VHH_132a	AKNRRGGWTVSDLGD	4
VHH_134	AAGQLGWIADCLELADYNGYNY	4
4A12	VKDFGHLGQMAS	4
4AD5	VKDLGFADH	4
4C5	AREWHSGYDY	4
Bn03_n3113v	VSNWASGSTGDY	4
Bn03_n3130v	ATRSPFGDYAFSY	4
CN111647076A_4A10	VKDFVVGETAEFSY	4
CN111825762A-A1	AALASSGYSRDYGAYDY	4
CN111825762A-B4	APRNGSPSVFEILLVSVY	4
CN111825762A-B6	ASGAVPAHQIGFRSTTLY	4
CN111825762A-B9	ANVIGTVNAYGAASKPAY	4
CN111825762A-E3	AAVAMLPFLTAVTPRPGY	4
CN111825762A-E6	AFGPAPKPQNVLTTALPY	4
CN111825762A-E7	APLLASAFVLMYGSRHLY	4
CN111825762A-F1	AAYLSSSRSGDYY	4
CN111825762A-F4	APGPGFTTMDRSQARIAY	4
CN111825762A-F5	AHRFQTRVRTTNPIESEY	4
CN111825762A-F8	AFGHTHMVRPGSTVMIMY	4
CN111825762A-G2	AYVVPYAIAAGAPDQIGY	4
CN111825762A-G3	AASDRLSGLRSYGY	4
CN111825762A-G5	APRVRLKVRFQDRVMVTY	4
CN111825762A-H3	ARTSEARYRGYPRFRVMFY	4
CN111825762A-H4	ADRNRVRGYDPCLHGY	4
CN111825762A-H6	AAIAVRDPHYVGVGGY	3
CN112062838A-sdAb	VKDFVGADGPFVFDY	4
CN112062840A	AAAYRYNGRDYYDRYDY	4
CN113563463A-1FC	AAIVDGWI	4
CN113563463A-2FC	AAWYIKMNSDMHVQREWE	4
CN113563463A-3FC	AAMSIGWPELF	4
CN113563463A-4FC	AAIETVHGHMI	4
CN113563463A-5FC	AAIDTEYGQNI	4
CN113563463A-6FC	AAWPTQDGYAA	4
Feng_17F6	RAYSLSAGMCAWMGYI	4
Feng_20G6	RAYSTTGDERDCRWQGYI	4
Fu2	AVGPSFSYTGSTYYRSELPWDYDY	4
H11-D4	ARTENVRSLLSDYATWPYDY	4
H11-H4	AQTHYVSYLLSDYATWPYDY	4
Hanke_C11	NAWVPVEEVAGIARQFQEV	4
Hanke_C7	NRAAVDGGGGYVPGRGDY	4
Hanke_D4	NAKGSSWYDLGGGAGDDY	4
Hanke_D9	NAEFGTPPVGYDY	4
Hanke_E11	ASGGEPLPRYYTDYASWVDY	4
Hanke_E2	HLRTFRRAGADTIPY	4
Hanke_F1	AYYTGRMATGWGNGGWKEYDY	4
Hanke_F12	AAGGEPLPRYWTDYASWVDY	4
Hanke_G1	AAGGEPLPRYWSDYASWVDY	4
Hanke_G2	AAGGEGYDSDYGPPLAPDY	4
Hanke_G6	AAERWGYSDCVAGYGMDY	4

Continued on next page

Table D.2 – continued from previous page

Name	Sequence	Cluster
Hong-1B5	VAADSHNSRCYLGRSYVNY	4
Hong-1H6	AATLYRVNCAKREFDK	4
Hong-2F7	NAMGRGSGSRCDNWDPNY	4
Hong-7A3	AAGSWYNQWGYSMDY	4
Hong-8A2	AAHGTYDKYAPCGGFAGTYTY	4
Hong-8A4	IIEALSGY	4
LR1	AAAEWGYEWPLYYASSWY	2
LR11	AAAYWGWDWPLNSQDYWY	2
LR15	AAADWGYNWPLIREEYEY	2
LR16	AAADWGYNIPLNITDYWY	2
LR2	AAAMNGYNEPLYSYDYEY	2
LR3	AAASWGYEWPLVYDDYWY	2
LR5	AAATWGYHWPLGAWDYWY	2
LR6	AAATWGYSWPLEHDEYWY	2
LR7	AAAFHGEQYPLYTNKYHY	2
LR8	AAANYGANFPLQANTYFY	2
MR10	NVKDEGATTKVYDY	1
MR14	NVKDWGAANKYYDY	1
MR17	NVKDDGQLAYHYDY	1
MR2	NVKDYGWYNSQYDY	1
MR3	NVKDYGAASWEYDY	1
MR4	NVKDFGGHQAYYDY	1
MR6	NVKDEGDTSASYDY	1
MR7	NVKDEGYFSDEYDY	1
MR8	NVKDWGSSNQYYDY	1
Nanosota-1C	MAGSKSGHELDH	4
NRL-N-A9	NIIPKSDQGAVNT	4
NRL-N-B6	ASGRYLGGITSYSQGDFAP	4
NRL-N-C2	AKYQAAVHQEKEDY	4
NRL-N-E10	AARAGPLGFELSATSSAEYDY	4
NRL-N-E2	ATNTRWTYFSPTVPDRYDY	4
P14-F8	AVAASGDTFEGRSDPDY	4
P14-F8-35	AVAASPATFEGRSDPDY	4
P14-F8-38	AVAASGDTFFGRSDPDY	4
RBD-Nb3	AAGPIYRAEVRQSDFPY	4
RBD-Nb35	ATRTNWFYGACLPKVSDFGS	4
S-Nb82	AASPFKSVVLGPGLYHH	4
S-Nb91	AAGGRCRYAGPLRSDFTY	4
Sb100	AAANWGYSWPLYQTEYWY	2
Sb12	YVKVGEWY	3
Sb13	YVVVGWGY	3
Sb15	FVVVGNGY	4
Sb17	LVYVGATY	3
Sb23	AVQVGYWY	4
Sb25	YVYVGAGY	3
Sb27	YVWVGRSY	3
Sb30	YVYVGESY	3
Sb32	VVWVGEVY	4
Sb37	NVKDEGNTTAYYDY	1
Sb38	NVKDFGTQEHYYDY	1
Sb39	NVKDFGGYRYYYDY	1
Sb40	NVKDEGAIAKNYDY	1
Sb42	NVKDEGYTGYYYDY	1

Continued on next page

Table D.2 – continued from previous page

Name	Sequence	Cluster
Sb43	NVKDWGSQDRYYDY	1
Sb45	NVKDEGKSSQVYDY	1
Sb46	NVKDVGNDQKSYDY	1
Sb47	NVKDWGTYSTYYDY	1
Sb50	NVKDWGWLAQYYDY	1
Sb52	NVKDEGMWQHYYDY	1
Sb54	NVKDEGNSQSHYDY	1
Sb56	NVKDAGNSKALYDY	1
Sb57	NVKDWGRAGARYDY	1
Sb58	NVKDMDRWRTTYDY	1
Sb6	YVWVGNQY	3
Sb60	NVKDWGYEYEGYDY	1
Sb61	NVKDTGTYQAWYDY	1
Sb62	NVKDWGGYQWYYDY	1
Sb63	NVKDYGAQAHYYDY	1
Sb67	NVKDWGTYNSYYDY	1
Sb7	LVYVGSTY	3
Sb71	AAAHYGDNFPLAYQAYLY	2
Sb75	AAARWGRDEPLYHYYYSY	2
Sb76	NVKDIGAQEVHYDY	1
Sb78	AAANYGNNWPLTGVNYWY	2
Sb8	YVWVGDSY	3
Sb83	AAAKYGQNFPLSYHAYRY	2
Sb84	AAARYGRSDPLHYHEYSY	2
Sb85	AAASWGYTWPLYTYDYWY	2
Sb88	NVKDSGQYRENYDY	1
Sb9	WVYVGDYY	3
Sb90	AAARWGRQYPLTFVYYSY	2
Sb93	AAARWGRTYPLSYMAYTY	2
Sb94	AAARWGRYEPLHYAYYSY	2
Sb95	AAASYGANWPLVSAAYTY	2
Sb97	AAARYGHAQAPLHYFWYGY	2
sdAb-1E2	AAQDSAYIKSKGSRAYEY	4
sdAb-2F2	AAHHIPTKHPAFPDFRDY	4
sdAb-3F11	AAEAFVQSPYSGSHTTKY	4
sdAb-4D8	AADQYEWWVPGEVGPYLY	4
sdAb-5F8	AAHYEFNDFVWQGYSSDY	4
SR1	YVYVGFSY	3
SR31	AVMVGFWY	4
SR38	AVHVGQTY	4
SR4	YVWVGHTY	3
SR5	YVYVGTSY	3
SR7	YVYVGYSY	3
WuN1	YTERWKPRGIERD	4

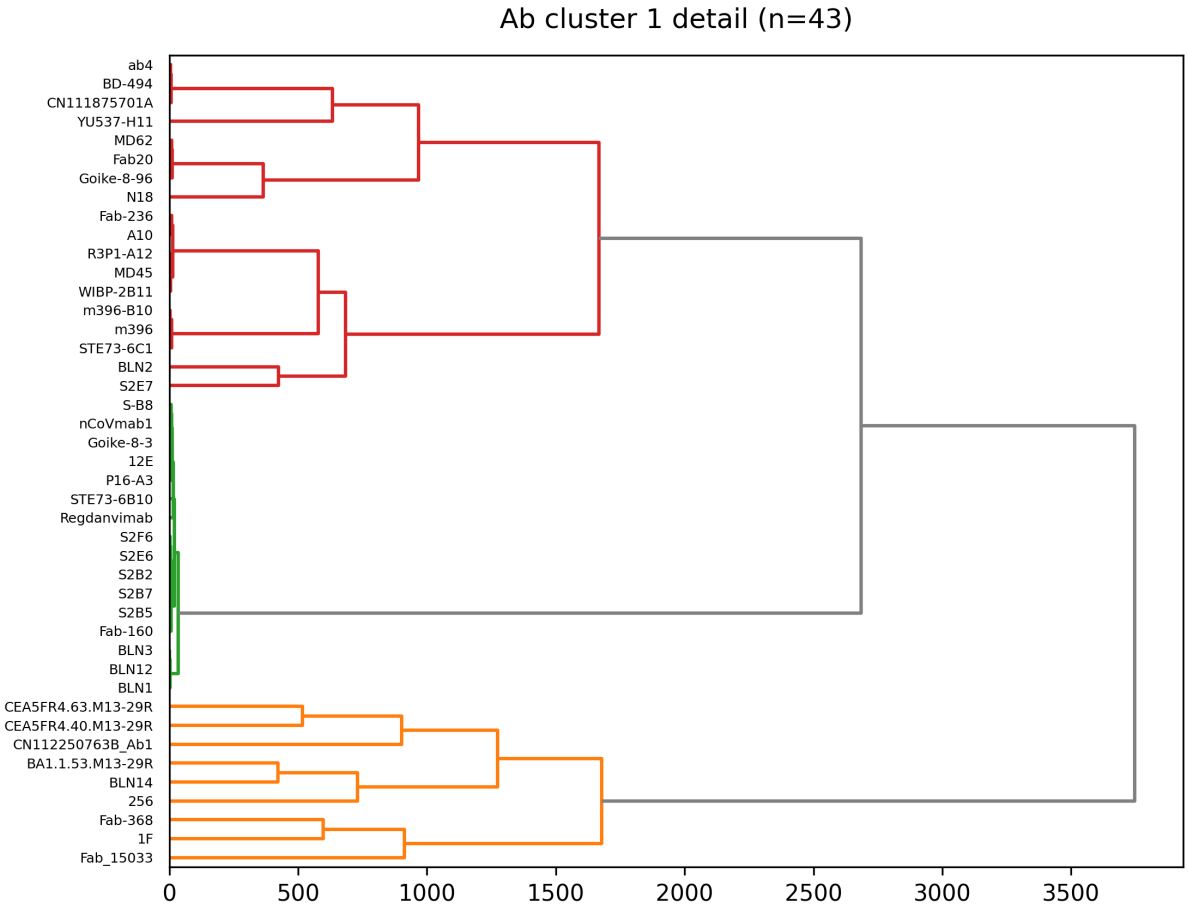


Figure D.2 Detailed dendrogram for Combined dataset Cluster 1

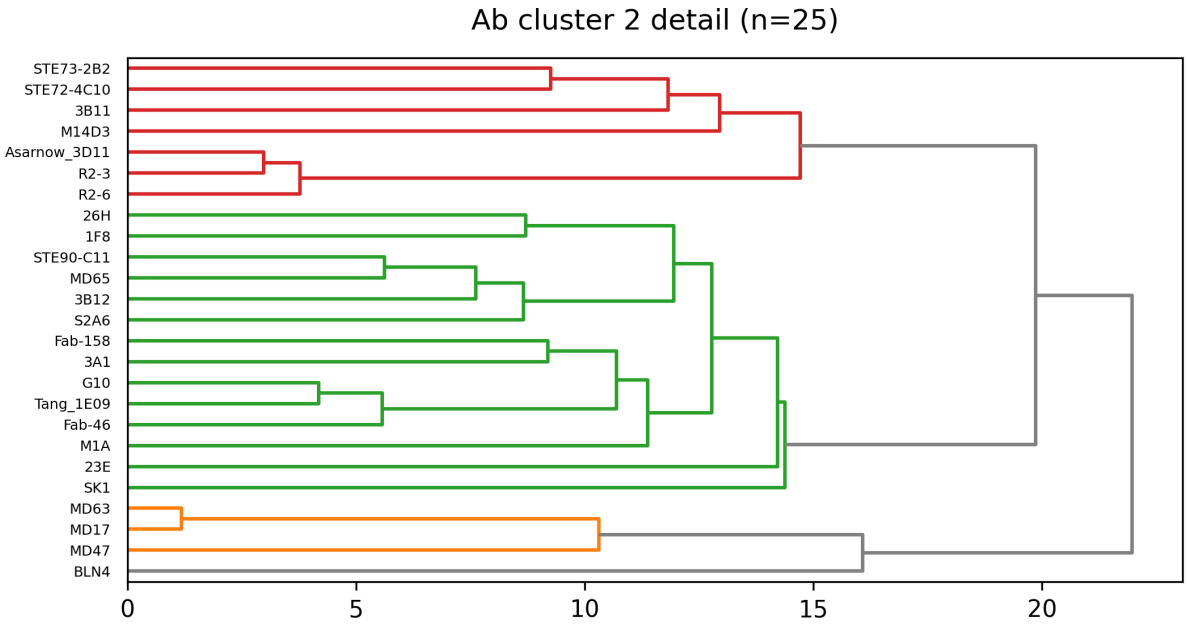


Figure D.3 Detailed dendrogram for Combined dataset Cluster 2

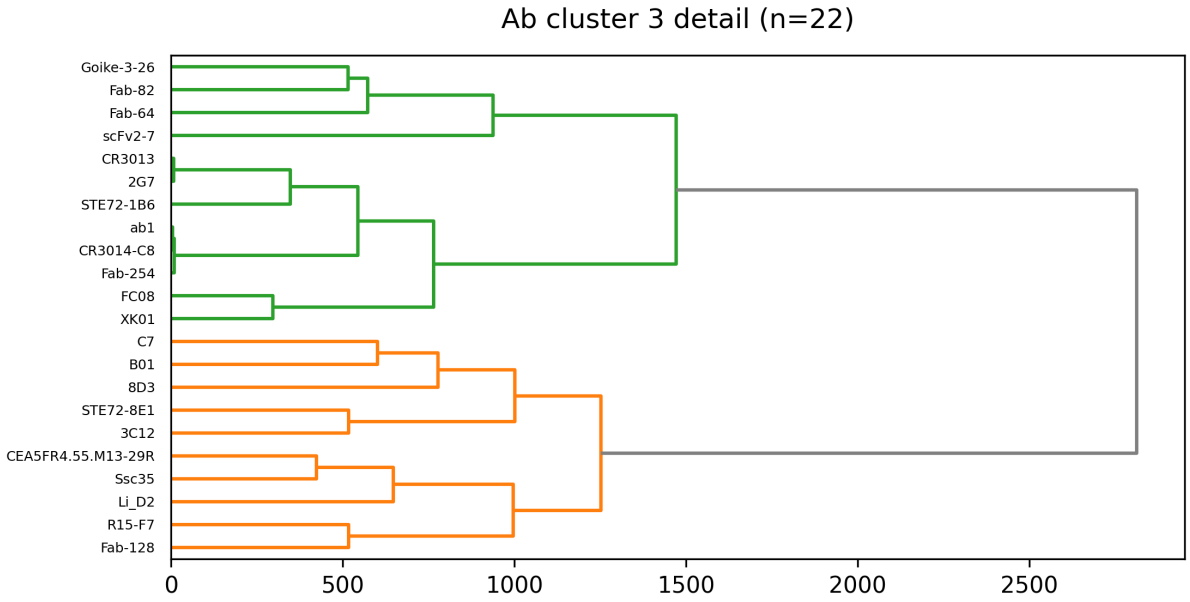


Figure D.4 Detailed dendrogram for Combined dataset Cluster 3

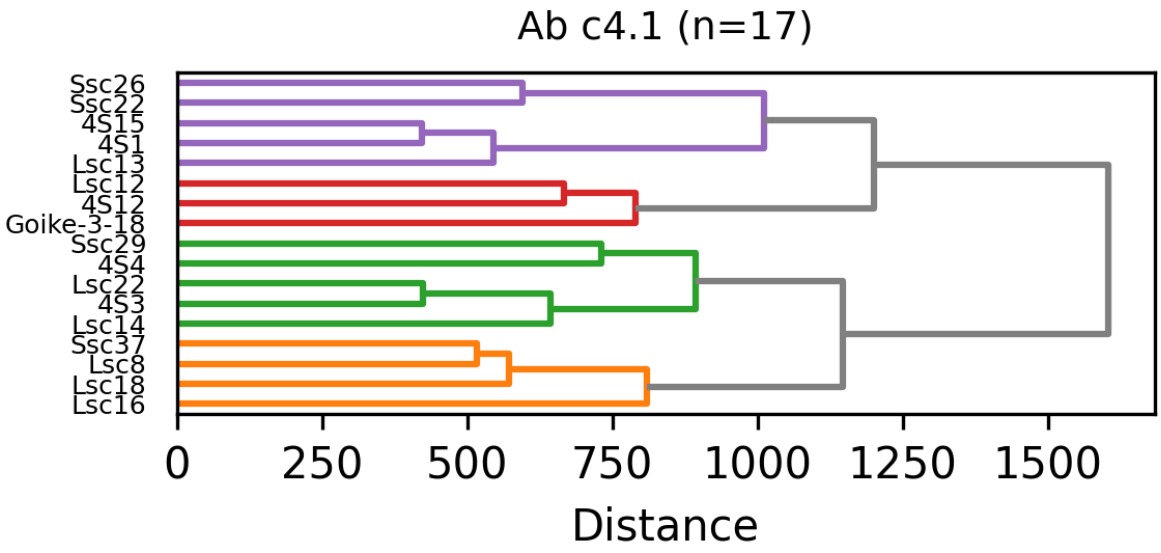


Figure D.5 Detailed dendrogram of Subcluster 1 within Cluster 4

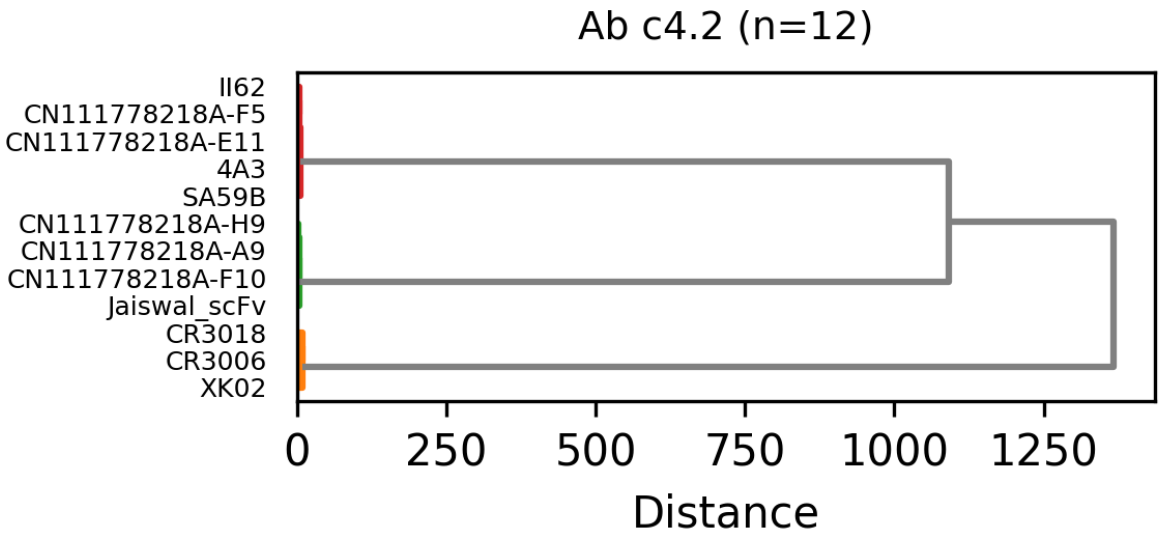


Figure D.6 Detailed dendrogram of Subcluster 2 within Cluster 4

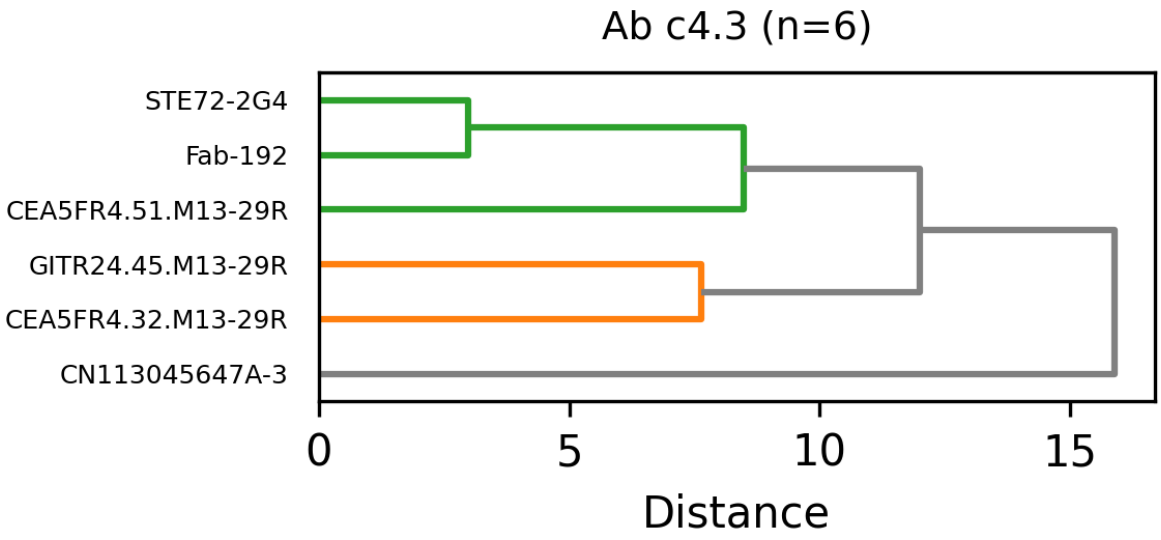


Figure D.7 Detailed dendrogram of Subcluster 3 within Cluster 4

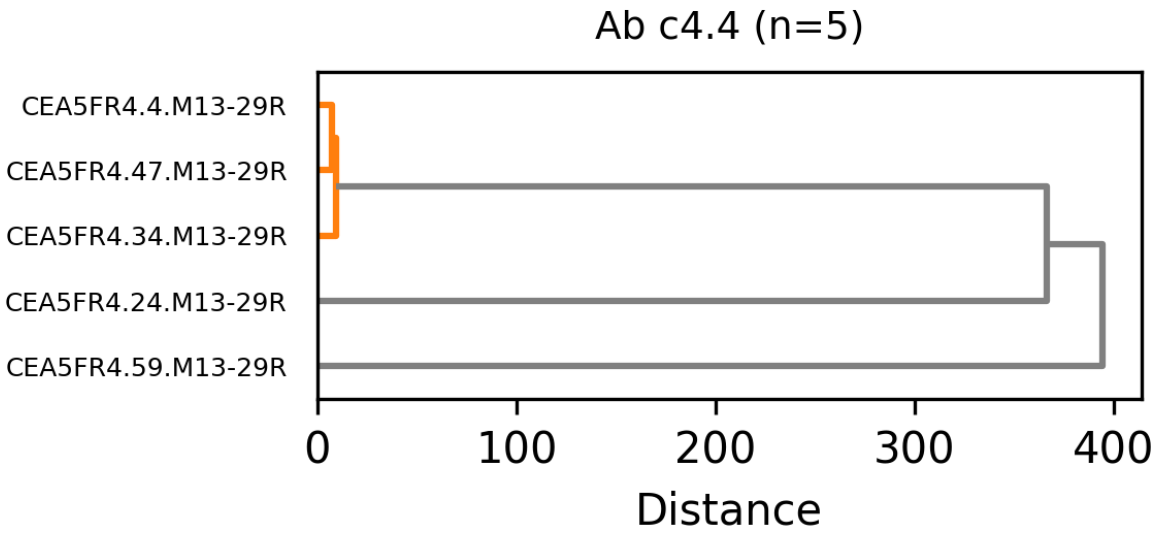


Figure D.8 Detailed dendrogram of Subcluster 4 within Cluster 4

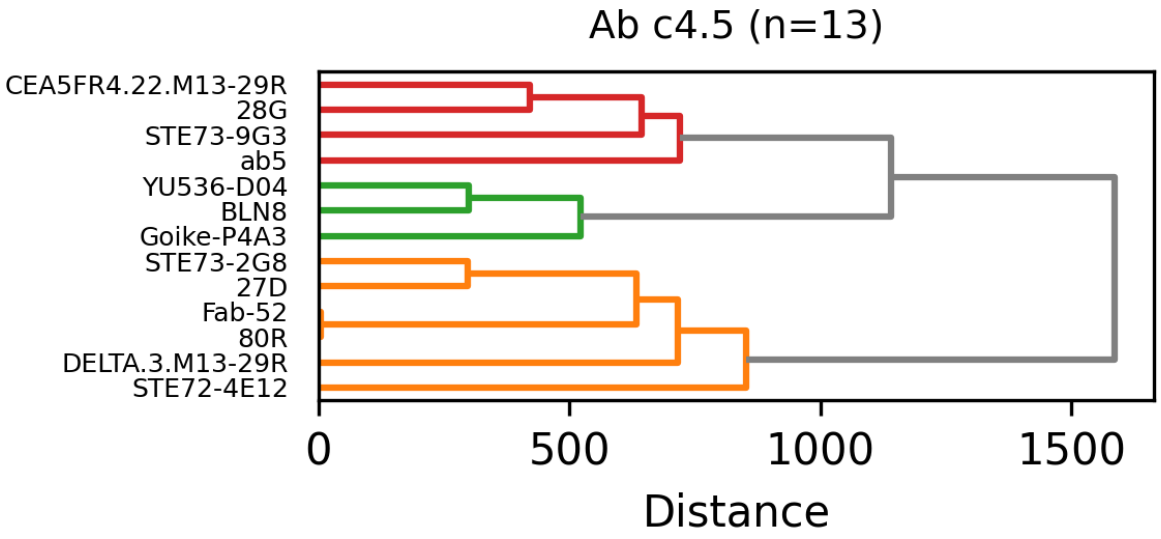


Figure D.9 Detailed dendrogram of Subcluster 5 within Cluster 4

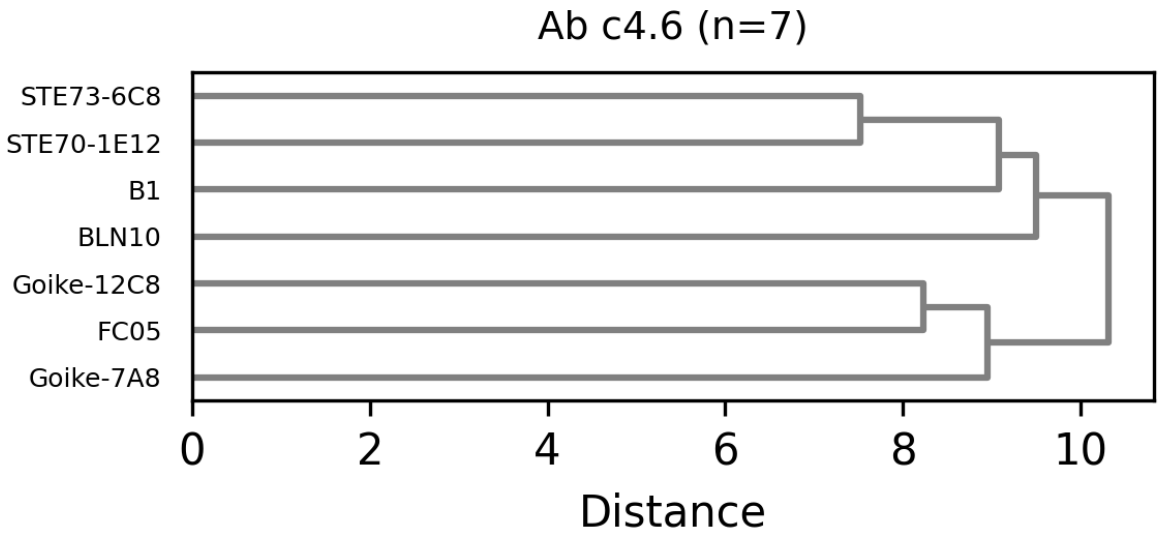


Figure D.10 Detailed dendrogram of Subcluster 6 within Cluster 4

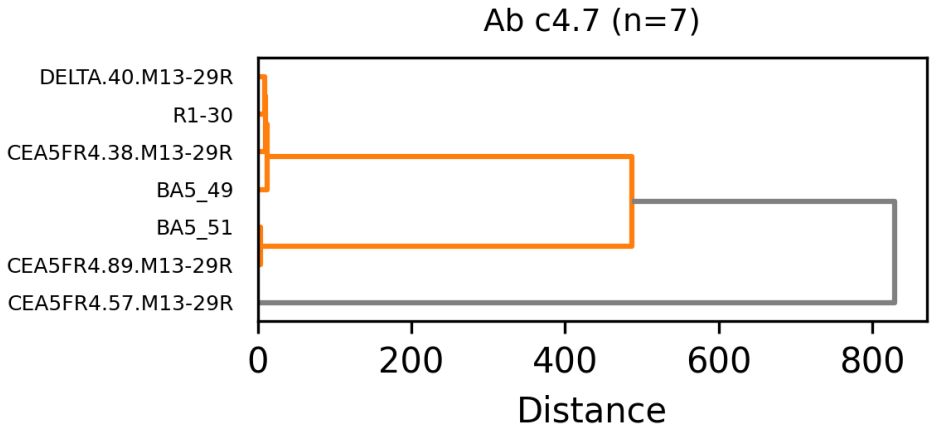


Figure D.11 Detailed dendrogram of Subcluster 7 within Cluster 4

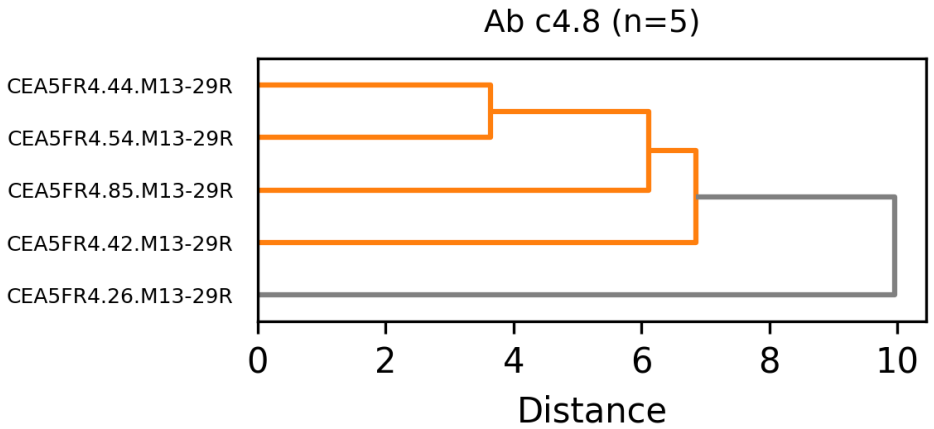


Figure D.12 Detailed dendrogram of Subcluster 8 within Cluster 4

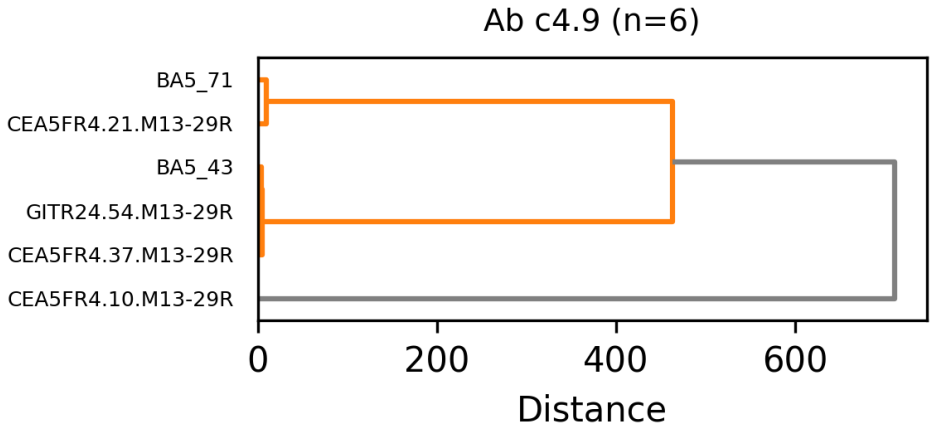


Figure D.13 Detailed dendrogram of Subcluster 9 within Cluster 4

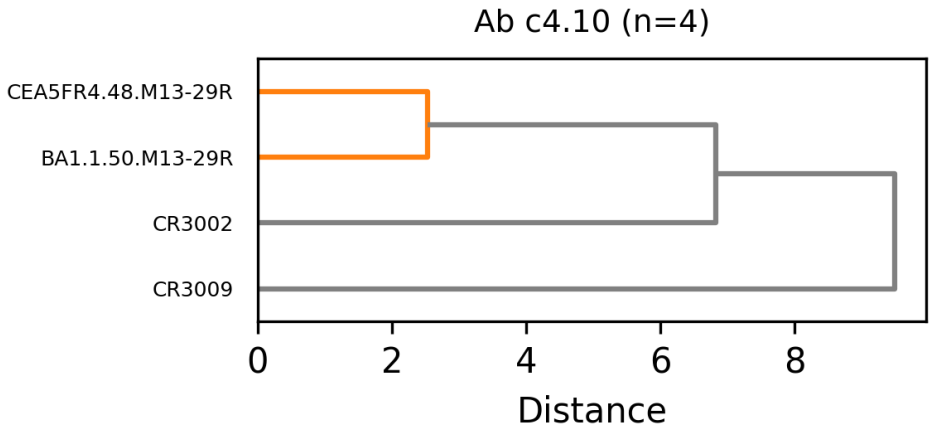


Figure D.14 Detailed dendrogram of Subcluster 10 within Cluster 4

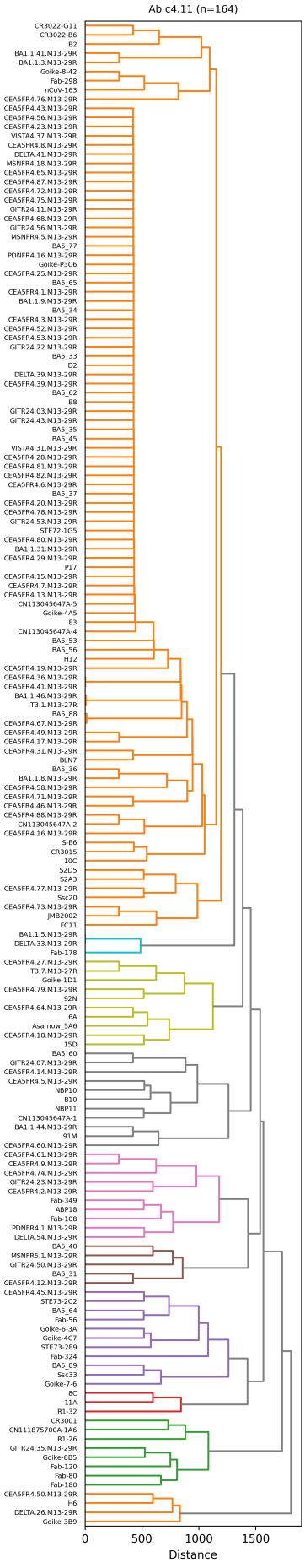


Figure D.15 Detailed dendrogram of Subcluster 11 within Cluster 4

Appendix E Model Hyperparameters and Training Details

Table E.1 Hyperparameter grids and selected values

Model	Grid & Selected
Logistic Regression	$C \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\} \rightarrow C = 1$; class weights = `balanced`
Random Forest	$n\_estimators \in \{100, 200, 300, 500\} \rightarrow 200$, $max\_depth \in \{4, 6, 8, 10\} \rightarrow 10$, $min\_samples\_leaf \in \{1, 2, 4\} \rightarrow 2$
XGBoost	$n\_estimators \in \{100, 300\} \rightarrow 300$, $max\_depth \in \{4, 6\} \rightarrow 4$, $learning\_rate \in \{0.001, 0.05, 0.1\} \rightarrow 0.1$, early stopping = 10 epochs
Bi-LSTM	hidden size $\in \{128, 256\} \rightarrow 256$, dropout $\in \{0.2, 0.3, 0.5\} \rightarrow 0.3$, batch size = 32, LR = 1×10^{-3}, early stopping = 10 epochs
ProtBERT	frozen layers $\in \{4, 6\} \rightarrow 6$, batch size = 16, LR = 2×10^{-5}, warm-up = 10% of steps
Siamese CNN	filters $\in \{16, 32\} \rightarrow 32$, kernels $\in \{3, 5, 7\} \rightarrow 5$, dropout = 0.3, batch size = 32, LR = 1×10^{-3}, early stopping = 10 epochs
Stacking Ensemble	base learners: LR, RF, XGB; meta-learner: logistic regression on 5-fold out-of-fold preds

Appendix F DNA to Protein and Variable Domain Annotation Tool

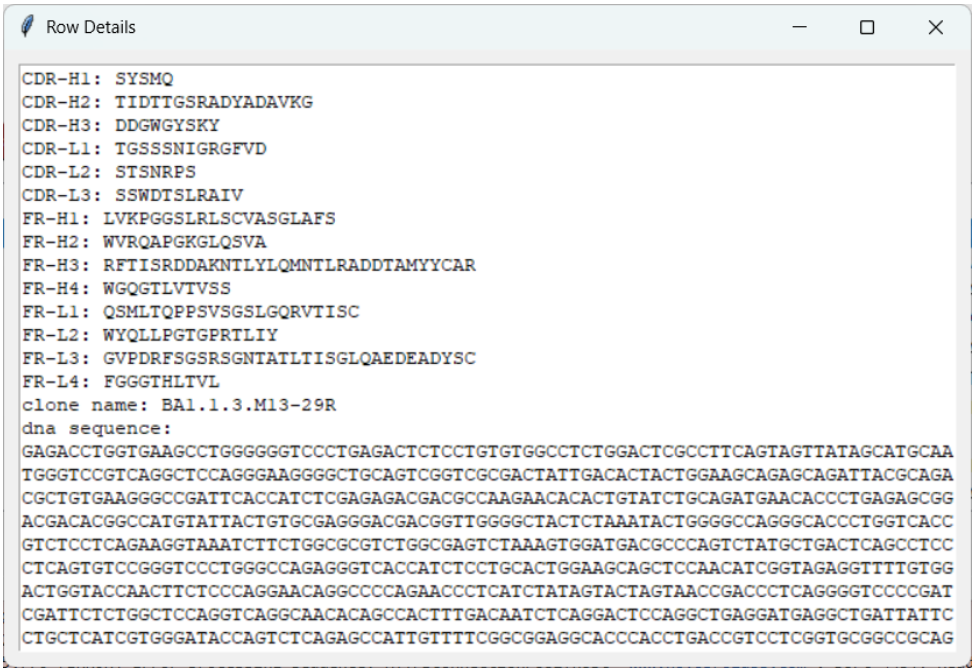


Figure F.1 DNA2Protein and Variable Domain Annotator – Row Details view showing the identified CDR-H1 through CDR-H3 and FR-H1 through FR-L4 regions, along with the underlying DNA and translated protein sequences for a selected clone.

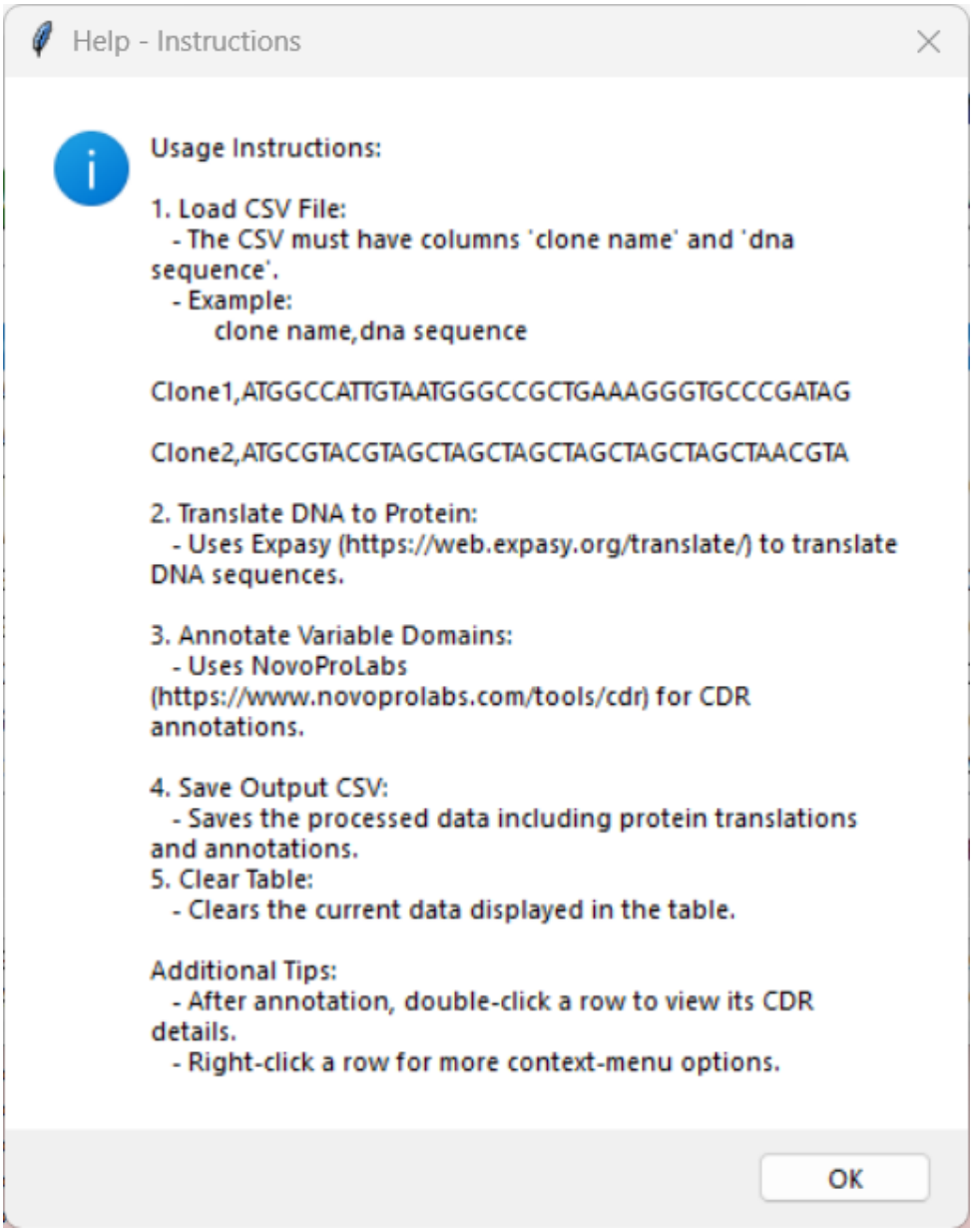


Figure F.2 DNA2Protein and Variable Domain Annotator – Help dialog summarising the workflow: loading a CSV of clone names and DNA, translating to protein via Expasy, annotating variable domains (CDRs/FRs) via NovoProLabs, and saving or clearing results.

Appendix G Clustering and Visualisation Tool

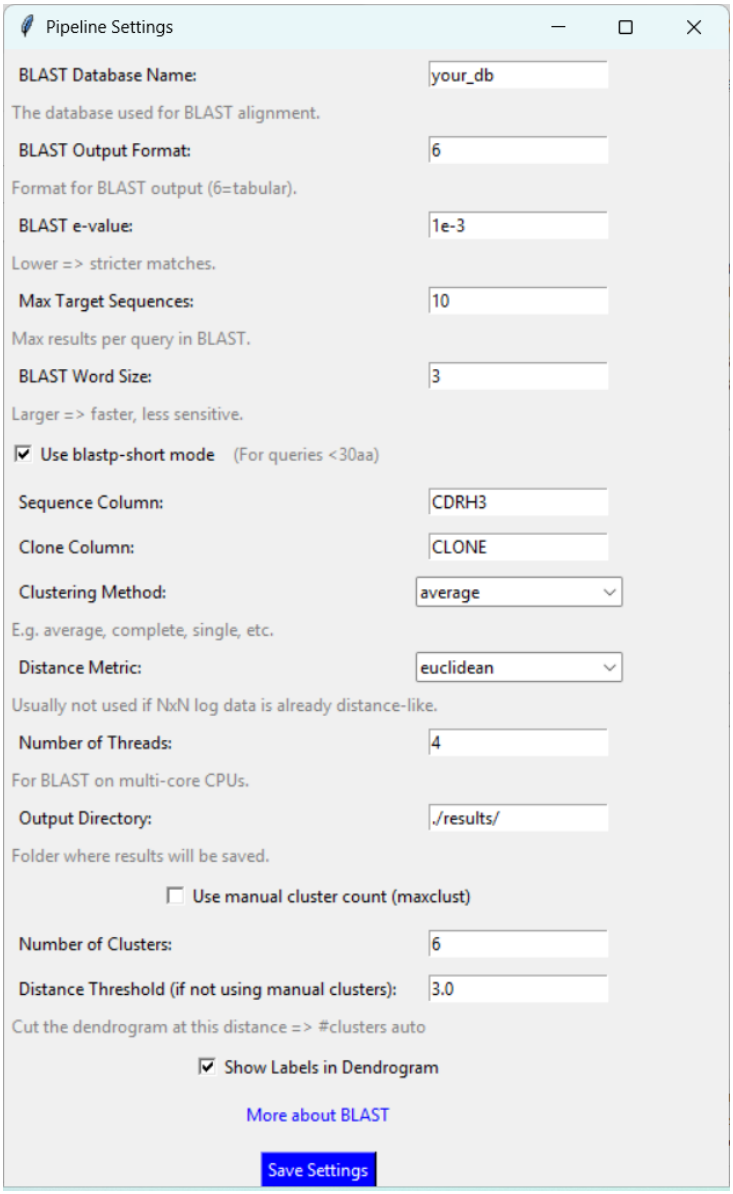


Figure G.1 Sequence Analysis and Clustering Interface – Settings panel. Configure the BLAST database name, distance metric, clustering method, number of threads, output directory, and cluster-cut parameters.

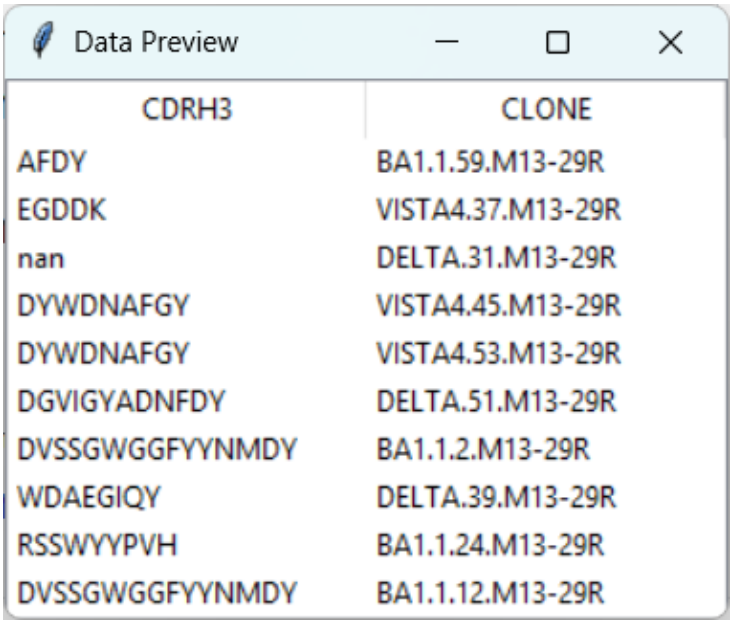


Figure G.2 Sequence Analysis and Clustering Interface – Data preview. After selecting one or more CSV files, this table displays the loaded clone IDs and sequences (e.g. CDRH3).

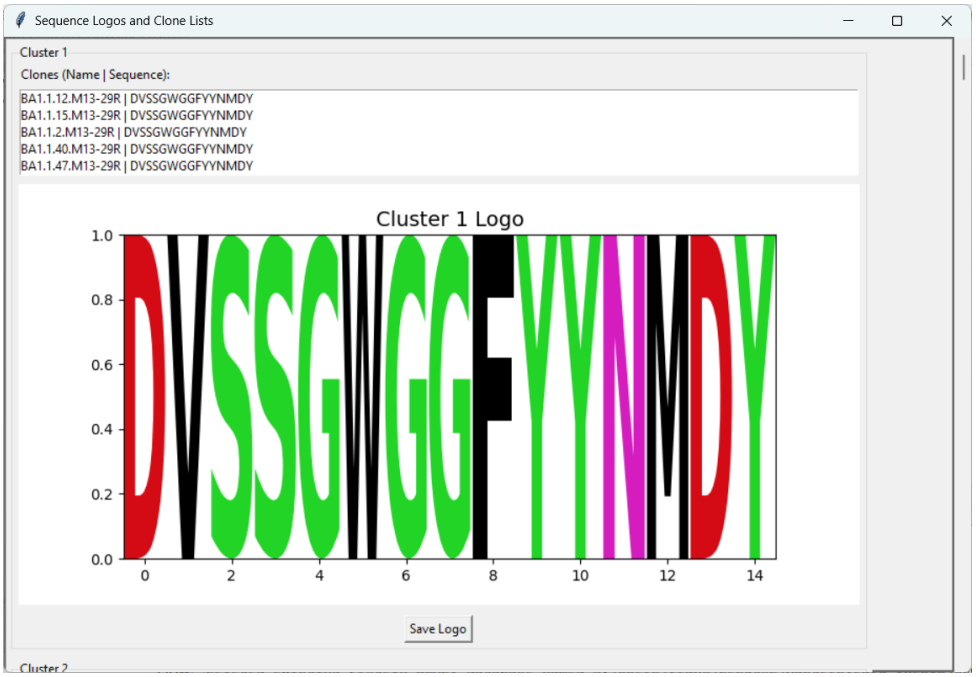


Figure G.3 Sequence Analysis and Clustering Interface – Sequence logo generation. A logo is shown here for the selected cluster, illustrating residue conservation and variability.

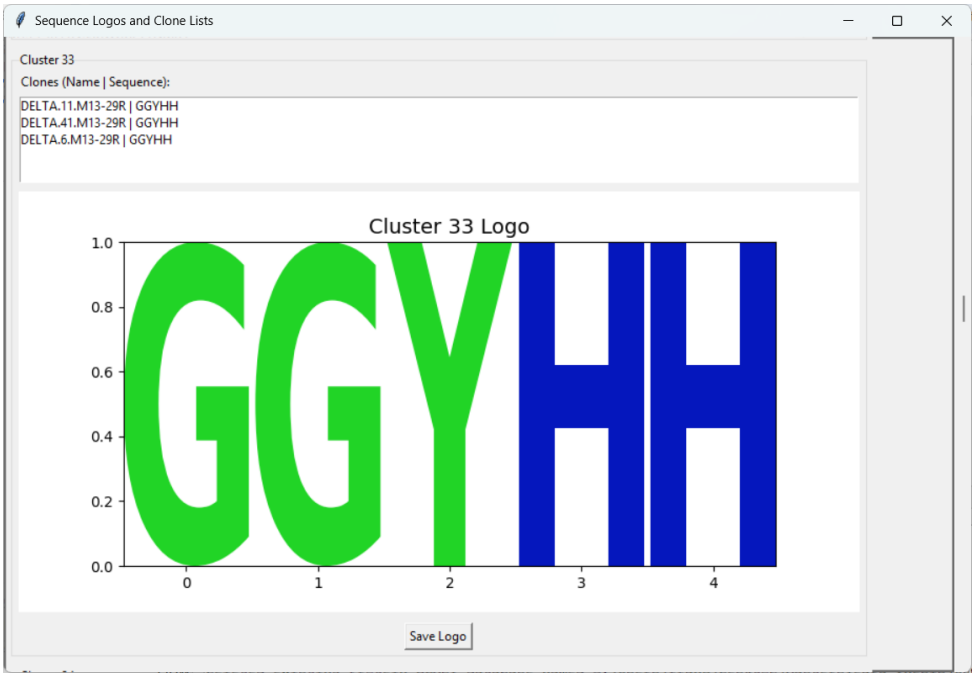


Figure G.4 Sequence Analysis and Clustering Interface – Sequence logo generation. A nother logo is shown here for another selected cluster, illustrating residue conservation and variability.

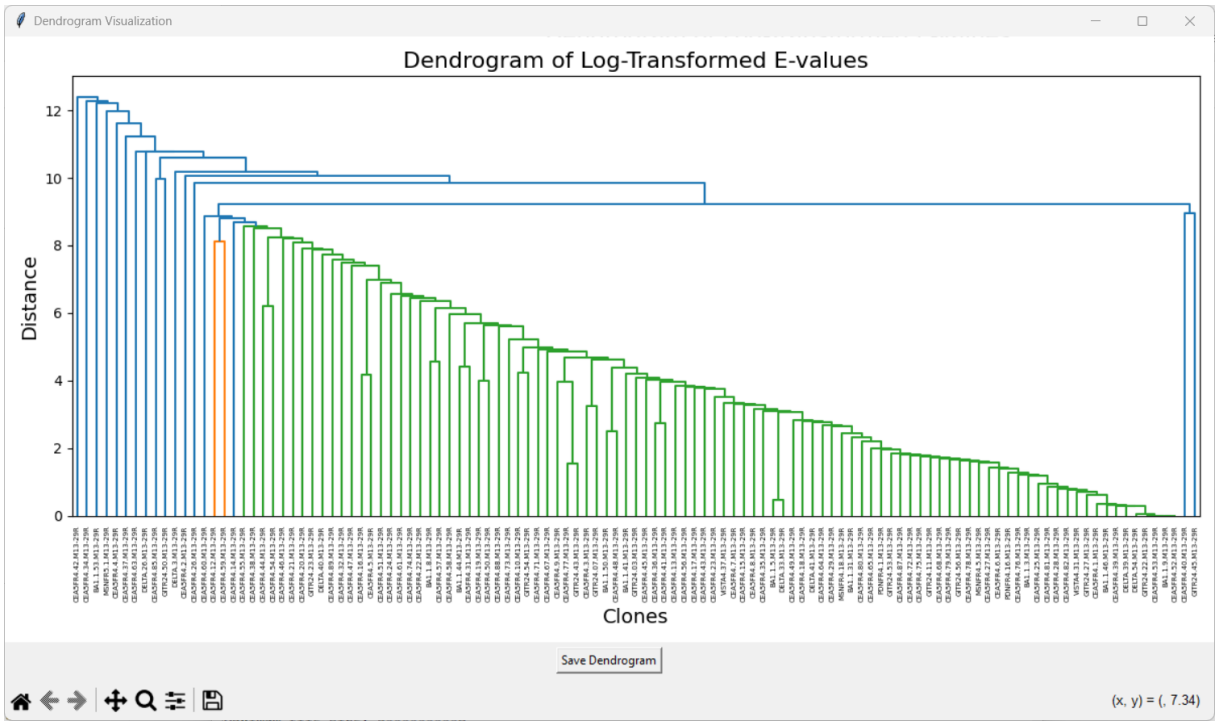


Figure G.5 Sequence Analysis and Clustering Interface – Dendrogram output. Hierarchical clustering of the loaded sequences is visualized here, with branch colors corresponding to cluster membership.

Appendix H Code Availability

All of the analysis and figure-generation scripts used in this work are openly available on GitHub at:

`https://github.com/francescac56/Dissertation2025`

You can clone the repository via

```
git clone https://github.com/francescac56/Dissertation2025.git
```

Detailed instructions for setup and usage are provided in the README file included in the repository.