# DEEP LEARNING METHODS FOR ANTIBODY STRUCTURE PREDICTION AND DESIGN

**by**

**Jeffrey A. Ruffolo**

**A dissertation submitted to Johns Hopkins University**

**in conformity with the requirements for the degree of**

**Doctor of Philosophy**

**Baltimore, Maryland**

**March 2023**

# Abstract

Antibodies are important immunological proteins, with the capacity to bind and neutralize a broad range of pathogens. The diversity of antibodies is conferred through genetic recombination and mutation, largely focused in a complementarity determining region composed of six loops. This natural diversity and binding capability has made antibodies an increasingly important therapeutic and diagnostic tool. However, despite their biological and medical significance, modeling and design of antibodies remains a challenge.

In the first half of this dissertation, I detail the development of a series of tools (DeepH3, DeepAb, and IgFold) to model increasingly complex portions of the antibody variable domain. These methods have progressively advanced the state-of-the-art in antibody modeling, first over traditional homology modeling approaches, then over highly accurate generalist methods for structure prediction. IgFold, the current-generation antibody structure prediction model, is capable of high-throughput antibody structure prediction with accuracy comparable to the best generalist methods, but in a fraction of the time. The speed and accuracy of IgFold should allow structure-based investigation on the scale of immune repertoires and accelerate the rational design of antibody therapeutics.

In the second half, I present work on generative language models for protein sequences. The first project describes ProGen2, a suite of language models trained at massive scale. I demonstrate that these models can be used to generate protein sequences resembling those produced by nature and to rank the relative fitness of protein sequences. The second project describes IgLM, a language model designed specifically for antibody design. IgLM can be used to create antibody libraries with favorable therapeutic properties or to generate full-length sequences with a specific species and chain type.

Taken together, my work has advanced our understanding of antibody structure through improved modeling, and shown how we might more effectively leverage natural antibody sequence data to achieve design of novel therapeutic molecules.

# Thesis Committee

Jeffrey Gray (Primary Advisor)
Professor
Department of Chemical and Biomolecular Engineering
Johns Hopkins Whiting School of Engineering

Jeremias Sulam (Advisor)
Assistant Professor
Department of Biomedical Engineering
Johns Hopkins Whiting School of Engineering

Doug Barrick (Reader)
Professor, Chair, and Vice Dean
T.C. Jenkins Department of Biophysics
Johns Hopkins Krieger School of Arts and Sciences

Juliette Lecomte
Professor
T.C. Jenkins Department of Biophysics
Johns Hopkins Krieger School of Arts and Sciences

Marc Ostermeier
Professor
Department of Chemical and Biomolecular Engineering
Johns Hopkins Whiting School of Engineering

Jamie Spangler
Assistant Professor
Department of Biomedical Engineering
Johns Hopkins Whiting School of Engineering

# Acknowledgments

First and foremost, I would like to my advisor, Jeffrey Gray. I believe the lab environment Jeff has cultivated is an excellent reflection of his own values. It is a warm and welcoming space where people are encouraged to be creative, to take risks, and above all to pursue their interests. I have been fortunate to have had the opportunity to work with Jeff for the past four years. I have learned a great deal from him, and I am grateful for his guidance and support.

Next, I would like to thank my co-advisor, Jeremias Sulam. Jere possesses an incredible versatility and depth of knowledge. No matter what project I brought to him, he always responded with enthusiasm and insight. I particularly appreciate his ability to ask the questions that cut to the heart of the matter, be it a technical problem or a philosophical one.

I would also like to thank my committee members, Doug Barrick, Juliette Lecomte, Marc Ostermeier, and Jamie Spangler. They have challenged me to think deeply about my work and consider broader perspectives and implications.

As I reflect back on my scientific journey, I would be remiss if I did not mention the early mentors who helped me along the way. First, thank you to my first research advisor, Andrew McClellan (Dr. M). The first time I visited

Dr M's lab at the University of Missouri, I was amazed by the volume and intricacy of instruments he had constructed. Perhaps foreshadowing, however, I would be working in another room on my laptop. Dr. M and I spent countless hours pouring over neuronal simulations. In the process, I learned a great deal from him about the scientific process and how to form intuitions about complex systems. My second research advisor, Yi Shang, introduced me to protein modeling and showed me that I could combine my two curiosities: computer science and biochemistry. During my time working with Dr. Shang, I developed deep interests that would continue to grow through my graduate studies.

During my time in the Gray lab, I had the pleasure of working alongside many excellent people. I recall one of the first conversations I had with Jeff Gray, walking across campus excitedly trying to understand the implications of all of the exciting new research that was changing the field. It felt impossible to keep up with, but during my rotation in the lab I got the chance to dive right in. I was fortunate to work alongside Sai Pooja Mahajan for my rotation and in the years following. Pooja was a great guide through the lab and helped me get up to speed quickly. I would also like to thank the other members of the Gray lab, past and present, that I had the opportunity to work with. Morgan Nance, Ameya Harmalkar, Rahel Frick, Sudhanshu Shanker, Lee-Shin Chu, Rituparna Samanta, Michael Chungyoun, and Fatima Talib were all excellent labmates to spend my time with. I also had the privilege of working with many excellent undergraduates. I would like to give a special acknowledgement to Deniz Akpinaroglu and Richard Shuai. Deniz and I worked together for nearly two

years on antibody modeling and, despite the ongoing pandemic, we managed to have many great conversations and learn a lot together. Richard joined the next summer and was a delight to work with on antibody language modeling for the next year. I wish them both the best in their graduate studies.

My doctoral studies were funded the by the National Institutes of Health and AstraZeneca, through the Johns Hopkins – AstraZeneca Scholars program. As part of this program, I was paired with an excellent mentor and collabrator, Gilad Kaplan. Through our collaboration, I got invaluable exposure to the practical aspects of antibody engineering and the drug development process. I would like to thank Gilad for his mentorship and the entire team at AstraZeneca for their support. I would also like to thank Takashi Tsukomoto, the program coordinator, for his guidance throughout my participation in the program.

Through a chance encounter at a virtual poster session, I had the chance to discuss my work on antibody language models with Ali Madani. This conversation would lead to several more, and eventually develop into a fruitful collaboration with his team at Salesforce Research. I would like to thank Ali for this opportunity and the entire team at Salesforce Research for their support.

When I arrived at Johns Hopkins, I was greeted by a cohort who would go on to become great friends. Franklin Aviles-Vazquez, Sanim Rahman, Anson Dang, Esther Park, Tom Zhang, Fran Harris, Briana Whitehead, and Amanda Qu were the most supportive and encouraging group of people I could've asked to share my graduate school experience with. None of us knew we

were signing up for a PhD that would be so shaped by a pandemic, but I think we made the best of it. I have so many fond memories with this group, from Korean BBQ with Anson, Esther, and Tom, to late nights with Franklin and Sanim, to holiday parties at Fran's. I'd like to give a special thank you to my close friend and roommate Franklin Aviles-Vazquez. I have tremendously benefitted from his genuine nature and his compassion for the people around him. I am grateful for the many conversations we've had about science, the world, and life in general.

I would like to finish by thanking my family. My parents, Cynthia and Andrew Ruffolo, have always been my biggest supporters. Throughout my life, they have always encouraged me to pursue my passions and to reach for the stars. A special thanks goes to my mother, who nurtured a love of science in me from a young age (though perhaps she was hoping I would be inclined towards medicine). Finally I'd like to thank my siblings, Alexis and Zachary Ruffolo, who have been through it all with me.

*"Just as mathematics turned out to be the right description language for physics, we think AI will prove to be the right method for understanding biology."*

–Demis Hassabis, DeepMind

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Proteins as the building blocks of life

Scientists approach understanding of biology at several distinct scales, ranging from the molecular to the organismal. The molecular scale is perhaps the most fundamental, and it is the scale at which proteins operate. Proteins are the building blocks of life, and they are responsible for nearly all of the functions of living organisms, ranging from catalysis of biochemical reactions to neutralization of invading pathogens. We approach the study of protein function through an understanding their structures. Proteins are composed of amino acids, which combine to form an unbranched polymer. There are twenty canonical amino acids, which vary in size, hydrophobicity, charge, and polarity. The order of amino acids determines the distinct three-dimensional structure of a protein, which enables its function. Protein structure is stabilized by a set of primarily non-covalent forces between their constituent amino acids, including hydrogen bonding, hydrophobic packing, and van der Waals interactions.

Despite our understanding of the forces that stabilize protein structure, we have only been able to determine the structures of a small fraction of the proteins that exist in nature. This is due to the difficulty of determining the structures of proteins in their native environments, which are often complex and dynamic. The structures of proteins are often determined by X-ray crystallography, which requires the protein to be crystallized in a laboratory setting. This is a time-consuming process, and it is not always possible to crystallize a protein of interest. As a result, we have only been able to determine the structures of a small fraction of the proteins that exist in nature. This is a major bottleneck in our understanding of protein function, and it is a major challenge in the field of structural biology.

### 1.1.1 Physically inspired methods for protein structure prediction

Since Pauling, Corey, and Branson deduced the primary elements of protein secondary structure in 1951 [1, 2], researchers have developed increasingly sophisticated methods for predicting protein structure. Many such methods are inspired by Afinsen's thermodynamic hypothesis, which postulates that the native structure of protein corresponds to the lowest-energy state on a landscape encoded by its amino acid sequence [3, 4]. As such, the process of predicting the folded state of a protein can be reduced to searching over candidate conformations in search of the energetic minimum. In practice, this is a computationally intractable problem, as the number of possible conformations grows exponentially with the length of the protein [5]. Methods for predicting protein structure have therefore relied on human intuition and our

understanding of the folding process to reduce the search space.

This era of protein structure prediction method development is well-exemplified by Rosetta, a suite of tools for macromolecular modeling and design [6]. Central to Rosetta-based methods is the energy function, a set of physical and statistic terms that approximate the favorability of a structural conformation given its amino acid sequence. The standard Rosetta energy function, *ref2015* [7], is made up of terms including:

- atomic attractive and repulsive energies
- atomic solvation energy
- electrostatic interaction energy
- hydrogen-bonding energy
- backbone torsion angle probability
- side-chain rotamer probability

Rosetta then formulates protein structure prediction as a Monte Carlo optimization problem, alternating between sampling conformations and calculating their likelihood under the energy function [8]. Beyond prediction of individual protein structures, Rosetta has enabled advances in modeling interactions between pairs of molecules, including protein-protein [9, 10], protein-ligand [11], and protein-glycan [12], as well as modeling of non-protein polymers such as RNA [13]. These extensive capabilities powered a number of feats in protein design [14], including design of novel topologies [15] and

assemblies [16], improvement of protein expression and stability [17], and therapeutic engineering [18, 19, 20]. However, despite the steady improvement of phsyically inspired methods like Rosetta with growing data and increasingly sophisticated functionalities, the accuracy of these approaches remained unsatisfactory for structural modeling, and the success of Rosetta-based methods in protein design required enormous compute resources. These obstacles began to be fade in 2018 with the successful application of deep learning to protein structure prediction [21].

### 1.1.2 Learning to predict protein structures from data

The Critical Assessment of protein Structure Prediction (CASP) is a biennial experiment for blind evaluation of methods for protein structure prediction. For twenty-four years, CASP witnessed steady improvements in predictors' ability to model the three-dimensional structures of proteins. Then, in 2018 at CASP13 [21], the accuracy of predicted structures began to rapidly accelerate with the development of deep learning methods from a small number of participants [22, 23, 24]. Early deep learning methods for protein structure prediction took inspiration from their predecessors, seeking to learn a set of inter-residue potentials that could be used to determine a protein structure through established methods like the CNS package [25] or Rosetta energy minimization. To achieve accurate prediction of these potentials, they employed neural networks designed for computer vision, namely convolutional neural networks (CNNs) [26] and residual neural networks (ResNets) [27]. It is important to note that deep learning alone did not enable the rapid improvement

in accuracy of protein structure prediction. Rather, it was the combination of deep learning with the availability of large amounts of sequence data, which had already shown promise for protein structure prediction through extraction of co-evolutionary information [28, 29].

Through the course of evolution, proteins acquire mutations in a constrained fashion that introduces diversity while maintaining (or gradually shifting) functionality [28]. Given sufficient examples of related proteins sampled across evolution, we can use machine learning to infer spatial relationships between groups of residues. Effective processing of these relationships, paired with an model architecture designed for the nuances of protein structure prediction, is the basis for the success of AlphaFold2 [30]. AlphaFold2 operates in two main stages: (1) processing of multiple sequence alignments and (2) prediction of protein structures. The first stage is a multi-task learning problem based on multiple-sequence alignment (MSA) input, with some aspects reminiscent of earlier approaches. The MSA processing unit, termed EvoFormer, learns to predict inter-residue distance potentials from a corrupted MSA. The MSA is corrupted by randomly masking identities from the input, which are then predicted by the EvoFormer. This learning task is inspired by techniques from natural language processing [31] (discussed more later), and had previously shown promise for protein representation learning [32]. The second stage, termed the structure module, uses the sequential and pairwise representations learned by the EvoFormer to place a set of per-residue coordinate frames in their correct tertiary positions. The entire AlphaFold2 model is trained end-to-end, meaning its outputs are directly aligned with the

intended purpose of the model (in contrast to previous models that produced potentials for energy minimization [33]).

## 1.2 Paradigms in machine learning

At a high level, machine learning can be divided into supervised and or unsupervised learning. Supervised learning is the process of learning a function that maps an input to an output, given a set of labeled examples. Protein structure prediction is an example of supervised learning, in which the amino acid sequence is the input and the three-dimensional structure is the output. In unsupervised learning, the objective is to extract patterns from data without labels. A prominent example of unsupervised learning is clustering, in which a set of data points is partitioned into groups based on their similarity. More relevant to this thesis is a class of unsupervised learning called self-supervised learning, in which the unlabeled data is used as both an input and an output for learning. Self-supervised learning is a powerful tool for learning from unlabeled data, and has been used prominently in natural language processing for textual representation learning [31] and generation [34]. The prediction of masked residues from an MSA in AlphaFold2 is an example of self-supervised learning [30]. Below, I discuss several prominent applications of supervised and self-supervised learning to protein modeling and design to motivate and contextualize the work of this thesis.

### 1.2.1 Applications of supervised learning for proteins

As discussed above, protein structure prediction is the marquee application of supervised learning for proteins today. However, supervised learning has a long history of applications to protein modeling and design. Among the earliest applications was the use of neural networks for secondary structure prediction [35, 36]. Foreshadowing the advances to come, these methods utilized sequence profiles and position-specific scoring matrices (PSSMs) to predict secondary structures from amino acid sequences. Supervised learning has also been used for rational protein design [37], peptide binder design [38], prediction of T-cell epitopes [39], and identification of protein-protein interactions [40]. More recently, supervised learning has shown promise for structure-conditioned protein sequence design. The first such approach utilized a graph neural network (GNN) to itereatively decode protein sequences (one amino acid at a time) given a specified protein structure [41]. This method outperformed its contemporaries, notably Rosetta [6], in terms of amino acid recovery (i.e., how often the model produces the known residue identity at each position) on a benchmark of natural proteins. In recent years, this idea has been extended and improved upon for design of protein-protein interactions, multi-state design, and protein fitness estimation [42, 43]. In addition to the examples highlighted here, supervised learning has natural applications to other classification tasks in protein modeling and design, such as prediction of protein function [44], binding affinity [45], and stability [46].

### 1.2.2 Applications of self-supervised learning for proteins

The commoditization of next-generation DNA sequencing techniques has lead to a significant growth of protein sequence data [47], massively outnumbering those with experimentally determined structures [48]. Although prominiNently utilized in the context of structure prediction, the promise of this data is not limited to supervised tasks. Self-supervised learning makes use of unlabeled data by corrupting or hiding portions of the input in some way, and then learning to reconstruct the original data. In natural language processing, there are two dominant self-supervised learning schemes: masked language modeling and autoregressive language modeling. Masked language modeling is a task in which a fraction of the input tokens are randomly masked, and the model is trained to predict the masked tokens [31]. Autoregressive language modeling is a task in which the model is trained to predict the next token in a sequence, given the previous tokens [34, 49]. The process of learning to perform these tasks is typically referred to as pretraining, as they are challenging, yet conceptually simple tasks that require the model to build a representation of the data distribution that can be useful for downstream applications. Both of these tasks have proven useful for modeling proteins. ESM-1b [50] is a masked language model trained on 250 million non-redundant sequences from UniRef [51]. Through pretraining on evolutionarily diverse protein sequences, ESM-1b learns representations that encode biophysical properties of amino acids, similarity of remote homologs, and contact patterns between distant residues. Another model, ProGen, was trained for autoregressive modeling of protein sequences [52]. Sequences generated by ProGen are predicted to adopt stable

folds (measured by Rosetta energy) and mirror the evolutionary signatures of natural proteins. In later work, ProGen was finetuned (i.e., further trained) on lysoyzme proteins and used to generate functional enzymes that diverge significantly from sequences produced by nature [53].

## 1.3 Machine learning for antibodies

Antibodies are a class of proteins that are critical for human health. They are typically composed of four protein chains (two identical heavy and two identical light) that pair up (heavy with light) then assemble into a large Y-shaped complex (Figure 1.1, left). In their biological role, antibodies function as an agent of the adaptive immune system, responsible for recognizing and neutralizing pathogens (referred to as antigens). To function effectively in this role, antibodies must be able to recognize a wide variety of antigens, while maintaining a high degree of specificity. This is achieved through a combination of genetic diversity and antigen-driven selection. The genetic diversity of antibodies begins with recombination of V(D)J genes, which can be mixed and matched to produce a variety of naive antibody sequences. After a specific antibody emerges from recombination events, somatic hypermutation further diversifies the antibody through accumulation of mutations in its antigen-binding region ($F_V$, Figure 1.1). Mutations that increase the affinity of the antibody for its antigen are selected for in a process called affinity maturation. Throughout this process, only the variable (antigen-binding) region is substantially changed, while the constant regions ($F_C$, Figure 1.1) remain largely the same. The exception to this conservation is class switching

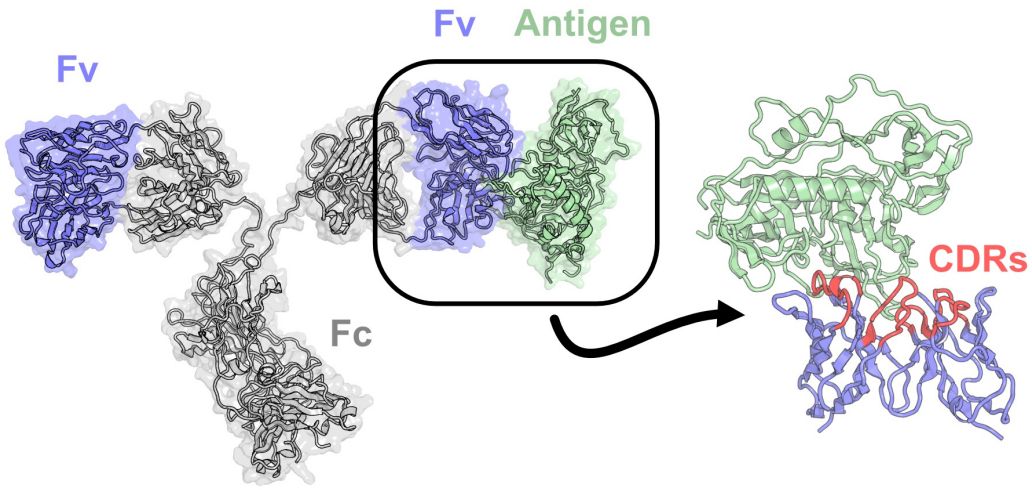


**Figure 1.1:** Components of antibody structure.

recombination, in which the $F_C$ region of an antibody is replaced with another. This process allows an antibody to interface with different effector cells to elicit alternative immune responses.

The binding of an antibody to its antigen is mediated by six hyper-variable loops in its variable region, referred to as the complementarity determining regions (CDRs). During affinity maturation, these loops undergo significant genetic alteration, including mutations, insertions, and deletions. As a result, these loops are highly variable in sequence and length. Structural modeling of these loops can provide insights into the binding mechanisms of antibodies, and can be used to guide the design of novel antibody therapeutics.

### 1.3.1 Homology modeling for antibody structures

Prior to the work presented in this thesis, homology modeling was the standard approach to predict antibody structures [54]. In homology modeling,

the antibody sequence of interest is aligned to experimentally determined structures sharing a similar sequence [55, 56, 57]. If a sufficient number of templates can be found, they can be grafted together to form an accurate structure. For the more conserved framework regions of the antibody $F_V$, there are usually sufficient templates to produce highly accurate homology models (with 1 Å RMSD). Indeed, for five of the six hypervariable CDR loops, there are typically sufficient templates to classify the loop sequence into a canonical structural class [58]. However, the immense conformational diversity of the third CDR loop of the heavy chain (CDR H3) makes it difficult to identify suitable templates in many cases. This is a major limitation of homology modeling for antibody structure prediction, as the CDR H3 loop occupies a central role in the antigen-binding surface. As such, inaccurate prediction of the CDR H3 loop can reduce utility for downstream applications, such as antibody-antigen docking and therapeutic design.

## 1.4 Dissertation outline

My thesis presents a series of methods developed to address key problems in antibody structure prediction and design with machine learning. The next three chapters focus on methods for antibody structure prediction, while the following two chapters focus on generative language modeling for protein sequence design.

In Chaper 2, I detail the development of DeepH3 [59], a method for scoring and modeling CDR H3 loops of antibody varaible regions. DeepH3 demonstrated that antibody-specific models could make better use of experimentally

determined structures than homology modeling for this critical component of antibody structure. Further, with DeepH3 we showed that antibody-specific deep learning models can outperform generalist structure prediction methods, giving rise to a distinct subfield of method development [60, 61, 62]. In Chapter 3, I present DeepAb [63], a method for prediction of the entire antibody $F_V$ region. DeepAb further demonstrated that deep learning methods offer improvements over homology modeling approaches. Additionally, the construction of DeepAb provided interpretability and use as a method for ranking antibody design candidates. In Chapter 4, I present IgFold [64], a method for fast, accurate antibody structure prediction. With IgFold, we made use of considerably more data than is available in the PDB through self-supervised learning and synthetic dataset generation. These advances allow IgFold to predict state-of-the-art structures in a fraction of the time required by prior deep learning methods. As a demonstration of these capabilities, we released a set of 1.4 million predicted antibody structures to the research community.

In Chapter 5, I describe the ProGen2 suite of autoregressive protein language models [65], developed in collaboration with Salesforce Research. We trained a series of models – ranging in scale from 151 million up to 6.4 billion parameters (the largest ever such model) – on protein sequences from genomic, metagenomic, and immune repertoire sequences. ProGen2 is useful in a variety of sequence generation contexts, and it is a state-of-the-art predictor of protein fitness. In Chapter 6, I present IgLM [66], a language model designed specifically for antibody library generation. IgLM generates sequences conditioned on a given species and chain type, and it can also be

used for diversification of targeted segments of an antibody sequence. The latter capability enables design of therapeutic antibody libraries, which we show resemble sequences produced by humans and have favorable physical properties.

Finally, in Chapter 7, I reflect on my contributions presented in this dissertation and discuss future directions for machine learning in antibody structure prediction and design.

# References


[1] Linus Pauling, Robert B Corey, and Herman R Branson. "The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain". In: *Proceedings of the National Academy of Sciences* 37.4 (1951), pp. 205–211.

[2] Linus Pauling and Robert B Corey. "Configurations of polypeptide chains with favored orientations around single bonds: two new pleated sheets". In: *Proceedings of the National Academy of Sciences* 37.11 (1951), pp. 729–740.

[3] Christian B Anfinsen. "Principles that govern the folding of protein chains". In: *Science* 181.4096 (1973), pp. 223–230.

[4] CB Anfinsen and HA Scheraga. "Experimental and theoretical aspects of protein folding". In: *Advances in protein chemistry* 29 (1975), pp. 205–300.

[5] Cyrus Levinthal. "How to fold graciously". In: *Mossbauer Spectroscopy in Biological Systems* 67 (1969), pp. 22–24.

[6] Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian W Kaufman, P Douglas Renfrew, Colin A Smith, Will Sheffler, et al. "ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules". In: *Methods in Enzymology*. Vol. 487. Elsevier, 2011, pp. 545–574.

[7] Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. "The Rosetta all-atom energy function for macromolecular modeling and design". In: *Journal of Chemical Theory and Computation* 13.6 (2017), pp. 3031–3048.

[8] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. "Protein structure prediction using Rosetta". In: *Methods in Enzymology*. Vol. 383. Elsevier, 2004, pp. 66–93.

[9] Jeffrey J Gray, Stewart Moughon, Chu Wang, Ora Schueler-Furman, Brian Kuhlman, Carol A Rohl, and David Baker. "Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations". In: *Journal of Molecular Biology* 331.1 (2003), pp. 281–299.

[10] Chu Wang, Philip Bradley, and David Baker. "Protein–protein docking with backbone flexibility". In: *Journal of Molecular Biology* 373.2 (2007), pp. 503–519.

[11] Steven A Combs, Samuel L DeLuca, Stephanie H DeLuca, Gordon H Lemmon, David P Nannemann, Elizabeth D Nguyen, Jordan R Willis, Jonathan H Sheehan, and Jens Meiler. "Small-molecule ligand docking into comparative models with Rosetta". In: *Nature Protocols* 8.7 (2013), pp. 1277–1298.

[12] Morgan L Nance, Jason W Labonte, Jared Adolf-Bryfogle, and Jeffrey J Gray. "Development and evaluation of GlycanDock: a protein–glycoligand docking refinement algorithm in Rosetta". In: *The Journal of Physical Chemistry B* 125.25 (2021), pp. 6807–6820.

[13] Clarence Yu Cheng, Fang-Chieh Chou, and Rhiju Das. "Modeling complex RNA tertiary folds with Rosetta". In: *Methods in Enzymology*. Vol. 553. Elsevier, 2015, pp. 35–64.

[14] Po-Ssu Huang, Scott E Boyken, and David Baker. "The coming of age of de novo protein design". In: *Nature* 537.7620 (2016), pp. 320–327.

[15] Brian Kuhlman, Gautam Dantas, Gregory C Ireton, Gabriele Varani, Barry L Stoddard, and David Baker. "Design of a novel globular protein fold with atomic-level accuracy". In: *Science* 302.5649 (2003), pp. 1364–1368.

[16] Neil P King, William Sheffler, Michael R Sawaya, Breanna S Vollmar, John P Sumida, Ingemar André, Tamir Gonen, Todd O Yeates, and David Baker. "Computational design of self-assembling protein nanomaterials with atomic level accuracy". In: *Science* 336.6085 (2012), pp. 1171–1174.

[17] Adi Goldenzweig, Moshe Goldsmith, Shannon E Hill, Or Gertman, Paola Laurino, Yacov Ashani, Orly Dym, Tamar Unger, Shira Albeck, Jaime Prilusky, et al. "Automated structure-and sequence-based design of proteins for high bacterial expression and stability". In: *Molecular Cell* 63.2 (2016), pp. 337–346.

[18] Aaron Chevalier, Daniel-Adriano Silva, Gabriel J Rocklin, Derrick R Hicks, Renan Vergara, Patience Murapa, Steffen M Bernard, Lu Zhang, Kwok-Ho Lam, Guorui Yao, et al. "Massively parallel de novo protein design for targeted therapeutics". In: *Nature* 550.7674 (2017), pp. 74–79.

[19] Daniel-Adriano Silva, Shawn Yu, Umut Y Ulge, Jamie B Spangler, Kevin M Jude, Carlos Labão-Almeida, Lestat R Ali, Alfredo Quijano-Rubio, Mikel Ruterbusch, Isabel Leung, et al. "De novo design of potent and selective mimics of IL-2 and IL-15". In: *Nature* 565.7738 (2019), pp. 186–191.

[20] Thomas W Linsky, Renan Vergara, Nuria Codina, Jorgen W Nelson, Matthew J Walker, Wen Su, Christopher O Barnes, Tien-Ying Hsiang, Katharina Esser-Nobis, Kevin Yu, et al. "De novo design of potent and resilient hACE2 decoys to neutralize SARS-CoV-2". In: *Science* 370.6521 (2020), pp. 1208–1214.

[21] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. "Critical assessment of methods of protein structure prediction (CASP)—Round XIII". In: *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1011–1020.

[22] Jinbo Xu. "Distance-based protein folding powered by deep learning". In: *Proceedings of the National Academy of Sciences* 116.34 (2019), pp. 16856–16865.

[23] Wei Zheng, Yang Li, Chengxin Zhang, Robin Pearce, SM Mortuza, and Yang Zhang. "Deep-learning contact-map guided protein structure prediction in CASP13". In: *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1149–1164.

[24] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792 (2020), pp. 706–710.

[25] Axel T Brünger, Paul D Adams, G Marius Clore, Warren L DeLano, Piet Gros, Ralf W Grosse-Kunstleve, J-S Jiang, John Kuszewski, Michael Nilges, Navraj S Pannu, et al. "Crystallography & NMR system: A new software suite for macromolecular structure determination". In: *Acta Crystallographica Section D: Biological Crystallography* 54.5 (1998), pp. 905–921.

[26] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[28] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. "Protein 3D structure computed from evolutionary sequence variation". In: *PLOS One* 6.12 (2011), e28766.

[29] Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. "Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era". In: *Proceedings of the National Academy of Sciences* 110.39 (2013), pp. 15674–15679.

[30] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[32] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. "MSA transformer". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8844–8856.

[33] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. "Improved protein structure prediction using predicted interresidue orientations". In: *Proceedings of the National Academy of Sciences* 117.3 (2020), pp. 1496–1503.

[34] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. "Improving language understanding by generative pre-training". In: (2018).

[35] Burkhard Rost and Chris Sander. "Improved prediction of protein secondary structure by use of sequence profiles and neural networks." In: *Proceedings of the National Academy of Sciences* 90.16 (1993), pp. 7558–7562.

[36] David T Jones. “Protein secondary structure prediction based on position-specific scoring matrices”. In: *Journal of Molecular Biology* 292.2 (1999), pp. 195–202.

[37] Gisbert Schneider and Paul Wrede. “The rational design of amino acid sequences by artificial neural networks and simulated molecular evolution: de novo design of an idealized leader peptidase cleavage site”. In: *Biophysical Journal* 66.2 (1994), pp. 335–344.

[38] Gisbert Schneider, Wieland Schrödl, Gerd Wallukat, Johannes Müller, Eberhard Nissen, Wolfgang Rönspeck, Paul Wrede, and Rudolf Kunze. “Peptide design by artificial neural networks and computer-based evolutionary search”. In: *Proceedings of the National Academy of Sciences* 95.21 (1998), pp. 12179–12184.

[39] Morten Nielsen, Claus Lundegaard, Peder Worning, Sanne Lise Lauemøller, Kasper Lamberth, Søren Buus, Søren Brunak, and Ole Lund. “Reliable prediction of T-cell epitopes using neural networks with novel sequence representations”. In: *Protein Science* 12.5 (2003), pp. 1007–1017.

[40] Yanay Ofran and Burkhard Rost. “Predicted protein–protein interaction sites from local sequence information”. In: *FEBS Letters* 544.1-3 (2003), pp. 236–239.

[41] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. “Generative models for graph-based protein design”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[42] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. “Robust deep learning–based protein sequence design using ProteinMPNN”. In: *Science* 378.6615 (2022), pp. 49–56.

[43] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. “Learning inverse folding from millions of predicted structures”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 8946–8970.

[44] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. “Structure-based protein function prediction using graph convolutional networks”. In: *Nature Communications* 12.1 (2021), p. 3168.

[45] K Yugandhar and M Michael Gromiha. "Protein–protein binding affinity prediction from amino acid sequence". In: *Bioinformatics* 30.24 (2014), pp. 3583–3589.

[46] Jianlin Cheng, Arlo Randall, and Pierre Baldi. "Prediction of protein stability changes for single-site mutations using support vector machines". In: *Proteins: Structure, Function, and Bioinformatics* 62.4 (2006), pp. 1125–1132.

[47] UniProt Consortium. "UniProt: a worldwide hub of protein knowledge". In: *Nucleic Acids Research* 47.D1 (2019), pp. D506–D515.

[48] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. "The Protein Data Bank". In: *Nucleic Acids Research* 28.1 (2000), pp. 235–242.

[49] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901.

[50] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118.

[51] Baris E Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H Wu. "UniRef: comprehensive and non-redundant UniProt reference clusters". In: *Bioinformatics* 23.10 (2007), pp. 1282–1288.

[52] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. "ProGen: Language modeling for protein generation". In: *arXiv preprint arXiv:2004.03497* (2020).

[53] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. "Large language models generate functional protein sequences across diverse families". In: *Nature Biotechnology* (2023), pp. 1–8.

[54] Juan C Almagro, Alexey Teplyakov, Jinquan Luo, Raymond W Sweet, Sreekumar Kodangattil, Francisco Hernandez-Guzman, and Gary L Gilliland. *Second Antibody Modeling Assessment (AMA-II)*. 2014.

[55] Jinwoo Leem, James Dunbar, Guy Georges, Jiye Shi, and Charlotte M Deane. "ABodyBuilder: Automated antibody structure prediction with data–driven accuracy estimation". In: *MAbs*. Vol. 8. 7. Taylor & Francis. 2016, pp. 1259–1268.

[56] Brian D Weitzner, Jeliazko R Jeliazkov, Sergey Lyskov, Nicholas Marze, Daisuke Kuroda, Rahel Frick, Jared Adolf-Bryfogle, Naireeta Biswas, Roland L Dunbrack Jr, and Jeffrey J Gray. "Modeling and docking of antibody structures with Rosetta". In: *Nature Protocols* 12.2 (2017), pp. 401–416.

[57] Dimitri Schritt, Songling Li, John Rozewicki, Kazutaka Katoh, Kazuo Yamashita, Wayne Volkmuth, Guy Cavet, and Daron M Standley. "Repertoire Builder: high-throughput structural modeling of B and T cell receptors". In: *Molecular Systems Design & Engineering* 4.4 (2019), pp. 761–768.

[58] Jared Adolf-Bryfogle, Qifang Xu, Benjamin North, Andreas Lehmann, and Roland L Dunbrack Jr. "PyIgClassify: a database of antibody CDR structural classifications". In: *Nucleic Acids Research* 43.D1 (2015), pp. D432–D438.

[59] Jeffrey A Ruffolo, Carlos Guerra, Sai Pooja Mahajan, Jeremias Sulam, and Jeffrey J Gray. "Geometric potentials from deep learning improve prediction of CDR H3 loop structures". In: *Bioinformatics* 36.Supplement_1 (2020), pp. i268–i275.

[60] Deniz Akpinaroglu, Jeffrey A Ruffolo, Sai Pooja Mahajan, and Jeffrey J Gray. "Simultaneous prediction of antibody backbone and side-chain conformations with deep learning". In: *PLOS One* 17.6 (2022), e0258173.

[61] Natalia Zenkova, Ekaterina Sedykh, Tatiana Shugaeva, Vladislav Strashko, Timofei Ermak, and Aleksei Shpilman. "Simple End-to-end Deep Learning Model for CDR-H3 Loop Structure Prediction". In: *arXiv preprint arXiv:2111.10656* (2021).

[62] Brennan Abanades, Guy Georges, Alexander Bujotzek, and Charlotte M Deane. "ABlooper: fast accurate antibody CDR loop structure prediction with accuracy estimation". In: *Bioinformatics* 38.7 (2022), pp. 1877–1880.

[63] Jeffrey A Ruffolo, Jeremias Sulam, and Jeffrey J Gray. "Antibody structure prediction using interpretable deep learning". In: *Patterns* 3.2 (2022), p. 100406.

[64] Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. "Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies". In: *bioRxiv* (2022), pp. 2022–04.

[65] Erik Nijkamp, Jeffrey Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. "Progen2: exploring the boundaries of protein language models". In: *arXiv preprint arXiv:2206.13517* (2022).

[66] Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. "Generative Language Modeling for Antibody Design". In: *bioRxiv* (2021).

# Chapter 2

# Geometric potentials from deep learning improve prediction of CDR H3 loop structures

Adapted from Jeffrey A Ruffolo, Carlos Guerra, Sai Pooja Mahajan, Jeremias Sulam, and Jeffrey J Gray. "Geometric potentials from deep learning improve prediction of CDR H3 loop structures". *Bioinformatics* 36.Supplement1 (2020), pp. i268-i275. Reproduced with permission.

## 2.1 Abstract

Antibody structure is largely conserved, except for a complementarity-determining region featuring six variable loops. Five of these loops adopt canonical folds which can typically be predicted with existing methods, while the remaining loop (CDR H3) remains a challenge due to its highly diverse set of observed conformations. In recent years, deep neural networks have proven to be effective at capturing the complex patterns of protein structure. This work proposes

DeepH3, a deep residual neural network that learns to predict inter-residue distances and orientations from antibody heavy and light chain sequence. The output of DeepH3 is a set of probability distributions over distances and orientation angles between pairs of residues. These distributions are converted to geometric potentials and used to discriminate between decoy structures produced by RosettaAntibody and predict new CDR H3 loop structures de novo. When evaluated on the Rosetta antibody benchmark dataset of 49 targets, DeepH3-predicted potentials identified better, same and worse structures [measured by root-mean-squared distance (RMSD) from the experimental CDR H3 loop structure] than the standard Rosetta energy function for 33, 6 and 10 targets, respectively, and improved the average RMSD of predictions by 32.1% (1.4 Å). Analysis of individual geometric potentials revealed that inter-residue orientations were more effective than inter-residue distances for discriminating near-native CDR H3 loops. When applied to de novo prediction of CDR H3 loop structures, DeepH3 achieves an average RMSD of 2.2 $\pm$ 1.1 Å on the Rosetta antibody benchmark.

## 2.2 Introduction

The adaptive immune system of vertebrates is responsible for coordinating highly specific responses to pathogens. In such a response, B cells of the adaptive immune system secrete antibodies to bind and neutralize their respective antigen. The central role of antibodies in adaptive immunity makes them attractive for the development of new therapeutics. However, rational

design of antibodies is hindered by the difficulty of experimental determination of macromolecular structures in a high-throughput manner. Advances in computational modeling of antibody structures provides an alternative to experiments, but computations are not yet sufficiently accurate and reliable.

Antibody structure consists of two sets of heavy and light chains that form a highly conserved framework region ($F_C$) and two variable regions responsible for antigen binding ($F_V$). The structural conservation of the $F_C$ is functionally significant, enabling the recognition of different antibody isotypes by their receptors and the $F_C$ lends well to homology modeling. The $F_V$ contains several segments of sequence hypervariability that provide the structural diversity necessary to bind a variety of antigens. This diversity is largely focused in six $\beta$-strand loops known as the complementarity determining regions (CDRs). Five of these loops (L1-L3, H1 and H2) typically fold into one of several canonical conformations [1] that are predicted well by existing methods [2]. However, the third CDR loop of the heavy chain (H3) is observed in a diverse set of conformations and remains a challenge to model [3, 4, 5, 6, 7, 8, 9]. Although the CDR loops are sometimes flexible and context-dependent, the change is typically small (<1 Å) between bound and unbound forms [10]. Because each antibody CDR H3 sequence evolves in an individual organism, evolutionary sequence history is not generally available (although there are exceptions [11, 12]).

Application of deep learning techniques has yielded significant advances in the prediction of protein structure in recent years. At CASP13, AlphaFold

[13] and RaptorX [14] demonstrated that inter-residue distances could be accurately learned from sequence and co-evolutionary features. Both approaches used deep residual network architectures with dilated convolutions to predict inter-residue distances, which provide a more complete structural description than contacts alone. trRosetta built on this progress by expanding beyond distances to predict a set of interresidue orientations [15]. This rich set of inter-residue geometries allows trRosetta to outperform leading approaches on the CASP13 dataset, even with a shallower network [15].

The effectiveness of inter-residue orientations for discriminating protein structures has also recently been demonstrated by methods such as SBROD and KORP [16, 17]. SBROD is a single-model quality assessment function that considers inter-residue interactions, backbone atom interactions, hydrogen bonding and solvent-solute interactions. Those features are extracted from a set of decoys from various CASP experiments and the SBROD scoring function is trained via ridge regression [16]. KORP is a knowledge-based potential constructed from a set of six inter-residue geometric descriptors similar to those of trRosetta [17]. Structures are scored according to a 6D joint probability distribution extracted from a database of non-redundant protein structures.

Our work expands on the progress in general protein structure prediction by applying similar techniques to a challenging problem in antibody structure prediction. Specifically, we propose DeepH3, a deep residual network that learns to predict inter-residue distances and orientations from antibody heavy and light chain sequence alone. When compared to state-of-the-art scoring methods, DeepH3 can identify near-native CDR H3 loops more accurately.

When used for de novo prediction of CDR H3 loop structures, DeepH3 produces lower-root-mean-squared distance (RMSD) structures than existing methods.

## 2.3 Methods

### 2.3.1 Overview

DeepH3 is a deep residual network [18] that learns to predict inter-residue distances and orientations from antibody heavy and light chain sequences. The architecture of DeepH3 draws inspiration from RaptorX [19, 14], which performed well on general protein structure prediction at CASP13. The relative scarcity of structural data for antibodies compared to general proteins presents challenges (as in any subproblems of structure prediction). We alleviate this limitation by reducing the depth of our network compared to previous methods, and we verify the generalization by examining performance on a highly diverse benchmark dataset. The outputs of DeepH3 are converted into geometric potentials to better discriminate between CDR H3 loop structures (decoys) generated using a standard homology modeling approach [20] and to predict new CDR H3 loop structures de novo.

### 2.3.2 Antibody structure datasets

#### Benchmark dataset

The Rosetta antibody benchmark dataset consists of 49 $F_V$ structures with CDR H3 loop lengths ranging from 9 to 20 residues [20, 21]. These structures were selected from the PyIgClassify database [22] based on their quality, with

each having resolution of 2.5 Å or better, a maximum R value of 0.2 and a maximum B factor of 80.0 $Å^2$ for every atom [20, 21]. The diversity of the set is enhanced by ensuring that no two structures share a common CDR H3 loop sequence, but the set is limited by the restriction to structures from humans and mice.

#### Training dataset

The training dataset for this work was extracted from SAbDab, a curated database of all antibody structures in the Protein Data Bank [23]. We enforced thresholds of 99% sequence identity and 3.0 Å resolution to produce a balanced, high-quality dataset. This high sequence identity cutoff was chosen due to the high conservation of sequence characteristic of antibodies. In cases where multiple chains existed for the same structure, only the first chain in the PDB file was used. Finally, any structures present in the Rosetta antibody benchmark dataset were removed. These steps resulted in 1433 structures, of which a random 95% were used for model training and 5% were used for validation. This small validation set was found to be sufficient to control for overfitting. Note that testing is carried out on an independent benchmark sharing no structures with the training/validation sets.

### 2.3.3 Learning inter-residue geometries from antibody sequence

#### Input features

Unlike most comparable networks, DeepH3 relies only on amino acid sequence as input. For general protein structure prediction, current methods

typically utilize some combination of multiple sequence alignments (MSAs), sequence profiles, co-evolutionary data, secondary structures, etc. [13, 19, 14, 15]. While these additional input features provide rich information for general protein structure predictions, each antibody evolves independently in one single organism, and we rarely have relevant evolutionary histories for CDR H3 loop sequences. Thus, we omit sequence alignment data like MSAs. DeepH3 takes as input a one-hot encoded sequence formed by concatenating the target heavy and light chains ($F_V$) sequences. A chain delimiter is added to the last position in the heavy chain, resulting in an input of dimension $L \times 21$, where $L$ is the cumulative length of the heavy and light chain sequences.

**Inter-residue geometries**

In addition to inter-residue distances, DeepH3 is also trained to predict the set of dihedral and planar angles previously proposed for trRosetta [15]. For two residues i and j, the relative orientation is defined by six parameters [$d$, $\omega$, $\theta_{ij}$, $\theta_{ji}$, $\phi_{ij}$ and $\phi_{ji}$, Figure 2.1A-B, adapted from [15]]. The distance ($d$) is defined using $C_\beta$ atoms or for glycine residues, $C_\alpha$. Distances were discretized into 26 bins, with 24 in the range of [4, 16 Å] and two additional bins for all distances below 4 Å or above 16 Å. The dihedral angle $\omega$ is formed by atoms $C_{\alpha i}$, $C_{\beta i}$, $C_{\beta j}$ and $C_{\alpha j}$, and the dihedral angle $\theta_{ij}$ is formed by atoms $N_i$, $C_{\alpha i}$, $C_{\beta i}$, and $C_{\beta j}$. Both dihedral angles were discretized into 26 equal-sized bins in the range of [-180, 180deg]. The planar angle $\phi_{ij}$ is formed by atoms $C_{\alpha i}$, $C_{\beta i}$, and $C_{\beta j}$. Planar angles were discretized into 26 equal-sized bins in the range of [0, 180deg]. Orientation angles were not calculated for glycine residues, due to the absence of the $C_\beta$ atom.

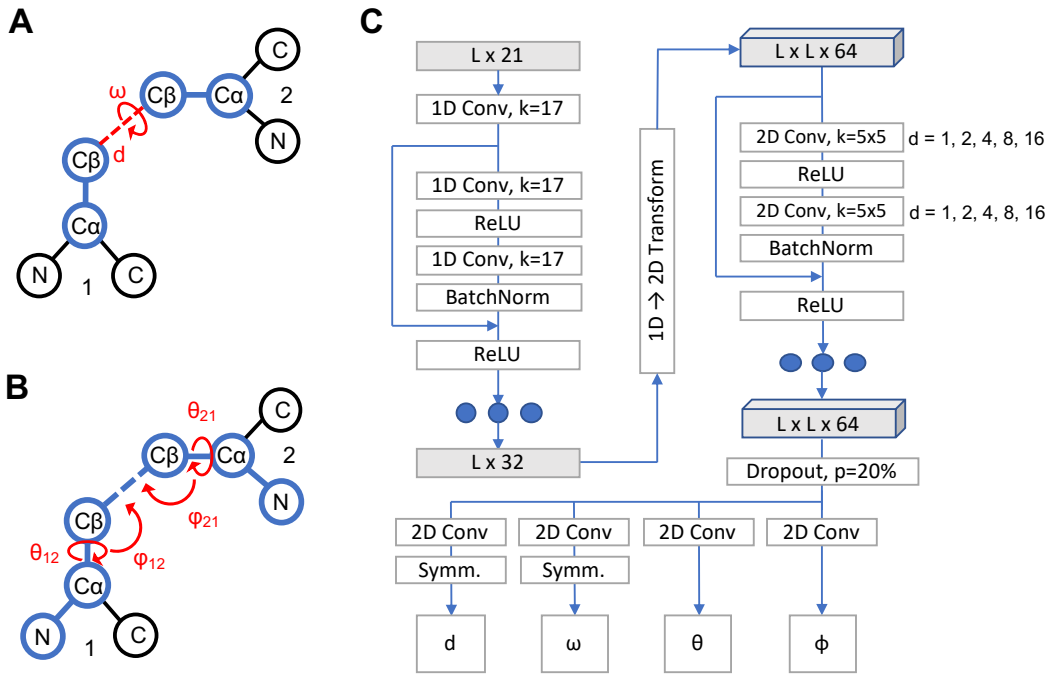


**Figure 2.1:** Architecture of DeepH3 deep residual neural network

(A) Illustration of the distance $d$ and dihedral $\omega$ for two residues. (B) Illustration of the dihedrals $\theta_{12}$ and $\theta_{21}$ and planar angles $\phi_{12}$ and $\phi_{21}$ for two residues. (C) Architecture diagram of residual neural network to learn inter-residue geometries from concatenated antibody $F_V$ chain sequences.

### Network architecture

DeepH3 applies a series of 1D and 2D convolutions to the aforementioned sequence input feature to predict four inter-residue geometries, as diagrammed in Figure 2.1C. The first 1D convolution (kernel size of 17) projects the $L \times 21$ input features up to an $L \times 32$ tensor. Next, the $L \times 32$ tensor passes through a set of three 1D residual blocks (two 1D convolutions with kernel size of 17), which maintain dimensionality. Following the 1D residual blocks, the sequential channels are transformed to pairwise by redundantly expanding the

$L \times 32$ tensor to dimension $L \times L \times 32$ and concatenating with the transpose, resulting in a $L \times L \times 64$ tensor. This tensor passes through 25 2D residual blocks (two 2D convolutions with kernel size of $5 \times 5$) that maintain dimensionality. Dilation of the 2D convolutions cycles through values of 1, 2, 4, 8 and 16 every five blocks (five cycles in total). Each of the preceding convolutions is followed by a batch normalization. Next, the network branches into four paths, which each apply a 2D convolution (kernel size of $5 \times 5$) to project down to dimension $L \times L \times 26$ (for 26 output bins). Symmetry is enforced for the d and $\omega$ branches after the final convolution by summing the resulting tensor with its transpose. The four resulting $L \times L \times 26$ tensors are converted to pairwise probability distributions for each output using the softmax function.

**Training**

Categorical cross-entropy loss was calculated for each output tensor and the resulting losses were summed with equal weight before back propagation. The Adam optimizer was used with an initial learning rate of 0.01 and reduction of learning rate upon plateauing of total loss. Dropout was used after the last 2D residual block, with entire channels being zeroed out at 20% probability. The network was trained using 95% of antibody dataset described above (1388 structures) for 30 epochs. Each epoch utilized the entire training dataset, with a batch size of 4. Training lasted about 35 hours using one NVIDIA Tesla K80 GPU on the Maryland Advanced Research Computing Center (MARCC).

### 2.3.4 Network predictions as geometric potentials

#### Implementation

We applied DeepH3 to each sequence in the Rosetta antibody benchmark dataset to produce pairwise probability distributions for the four output geometries. Distributions for pairs of residues that did not include a member of the CDR H3 (according to Chothia number) loop were discarded. Additionally, pairs of residues for which the maximum probability bin of the distance output was greater than 12 Å were discarded to focus on local interactions that are likely to carry biophysical meaning. We also disregarded those predicted distributions that were not informative enough, chosen as those with a maximum probability below 10%. The remaining distributions were converted to potentials by taking the negative natural log of each output bin probability. Continuous, differentiable Rosetta constraints (AtomPair for d, Dihedral for $\omega$ and $\theta$ and Angle for $\phi$) were created for each potential using the built-in spline function. Within Rosetta, a histogram corresponding to each pairwise potential is fit to a cubic spline. These constraint functions are used calculate the DeepH3 energy term for each structure.

#### CDR H3 loop discrimination

To test the effectiveness of predicted geometric potentials for discriminating between near-native CDR H3 loops, we collected a set of 2800 decoy structures generated by RosettaAntibody for each of the 49 Rosetta antibody benchmark targets [20]. These structures were generated by homology modeling, with decoys for each target assuming various heavy/light-chain orientations and

non-H3 CDR loop conformations [20, 24]. After scoring each structure with DeepH3, we compared the discrimination performance to three other stateof-the-art scoring methods: SBROD [16], KORP [17] and the ref2015 full-atom energy function (referred to as Rosetta energy) [25].

#### Discrimination score

The discrimination score is a common metric for measuring the success of structure prediction calculations by assessing whether the minimum energy structures are near-native, with a lower value being indicative of a more successful prediction [21]. To compare between different energy schemes, we first scale the scores for all decoy structures such that the 95th percentile energy has a value of 0.0 and the 5th percentile energy has a value of 1.0. The discrimination score is then calculated as [26]:

$$D = \sum_{r \in \{1,1.5,2,2.5,3,4,6\}} \min_{i,\mathrm{RMSD}(i) \in [0,r]} \mathrm{E}_i - \min_{i,\mathrm{RMSD}(i) \in [0,\infty]} \mathrm{E}_i \tag{2.1}$$

where r is the RMSD cutoff in Å, $\mathrm{E}_i$ is the scaled energy for the $i$-th decoy structure, and the discrimination score, D, is the sum of the energy differences for the best scoring decoys above and below each RMSD cutoff.

### 2.3.5 De novo prediction of CDR H3 loop structures

#### DeepH3 prediction of crystal Fv framework

We applied the Rosetta LoopModeler protocol [27, 28] to each target in the Rosetta antibody benchmark to build the CDR H3 loop onto the $\mathrm{F}_V$ crystal structure. Prior to modeling, the crystallographic loop was extended by

setting $\phi$ and $\psi$ angles to 180deg to emulate a blind prediction. Throughout the modeling process, the KIC algorithm was guided only by DeepH3 energy, with all Rosetta energy function terms disabled. For each target, 500 decoys were generated. We elected to use a relatively low number of decoys after observing faster convergence with DeepH3 energy than is typical for Rosetta energy.

**trRosetta heavy chain prediction**

The most similar approach to DeepH3 is trRosetta for general protein structure prediction. To better understand the impacts of designing a network specifically for antibody structures, we tested the performance of trRosetta on the Rosetta antibody benchmark using the public trRosetta server [15]. Because trRosetta was designed to predict the structure of single-chain proteins, we submitted only heavy chain sequences (i.e. omitting the light chain). The five resulting structures were aligned to the heavy chain in the crystal structure to measure the RMSDs of the CDR H3 loop heavy atoms.

## 2.4 Results

### 2.4.1 DeepH3 accurately predicts inter-residue geometries

To evaluate the accuracy of DeepH3's predictions, we applied our model to the entire Rosetta antibody benchmark dataset (not seen during training or validation). For residue pairs involving a CDR H3 loop residue, the predicted values for each geometry are plotted against experimental structure values in Figure 2.2. We limit our analysis to pairs including an H3 loop residue to

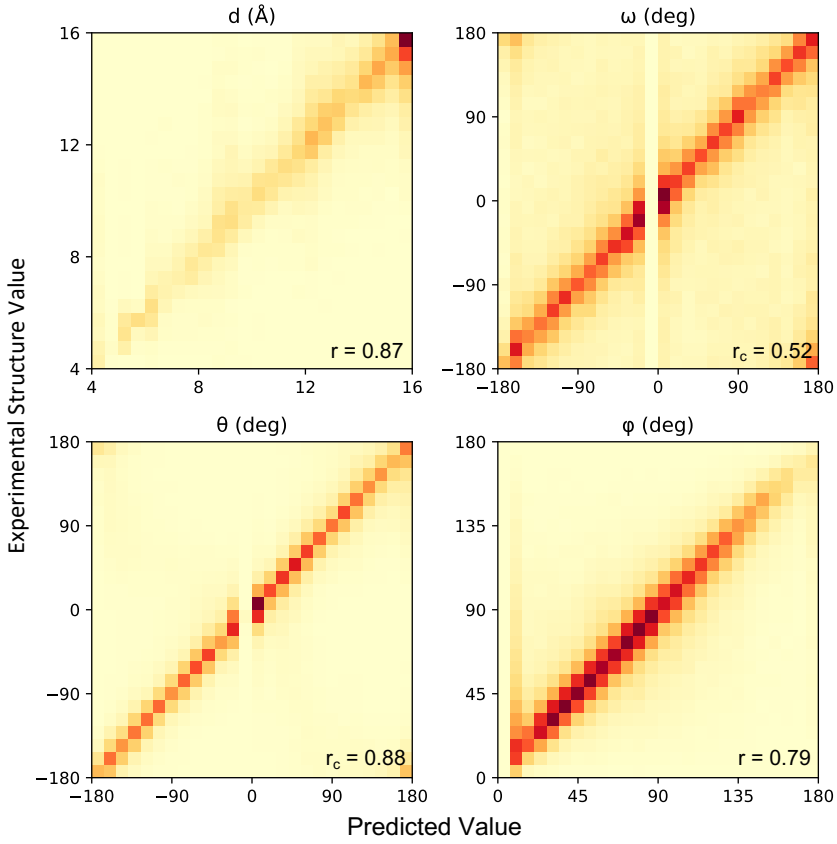


**Figure 2.2:** Accuracy of predicted inter-residue geometries

Pearson correlation coefficients (for d and $\phi$) and circular correlation coefficients (for $\omega$ and $\theta$) are calculated between DeepH3 predictions and experimental values.

ensure that DeepH3 is effectively learning the most variable regions of the antibody structure, rather than just the conserved framework. DeepH3 displays effective learning across all outputs; the Pearson correlation coefficients (r) for d and $\phi$ were 0.87 and 0.79, respectively, and the circular correlation coefficients ($r_c$) for dihedrals $\omega$ and $\theta$ were 0.52 and 0.88, respectively.

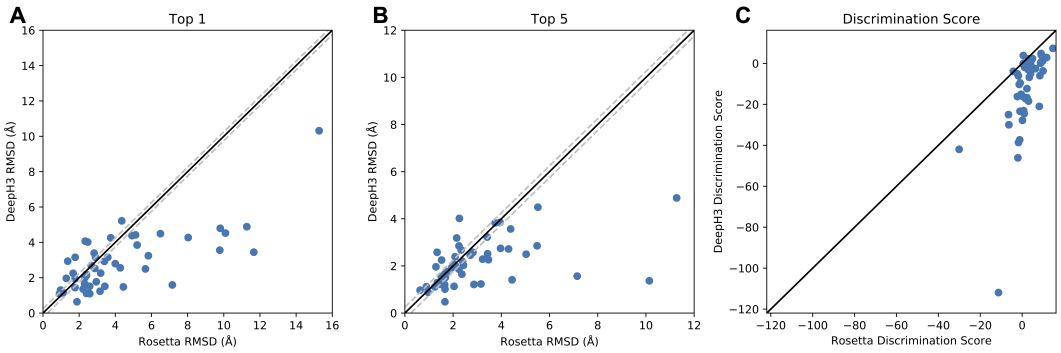


**Figure 2.3:** Effectiveness of predicted inter-residue geometries for decoy discrimination

(A, B) Comparison of the quality of structures selected by Rosetta energy and DeepH3 energy (using all geometric potentials). The quality of structures is considered the same if the difference in RMSD is within 60.25 Å , indicated with dashed lines. (A) DeepH3 energy selected better-, sameand worse-RMSD structures for 33, 6 and 10 out of 49 targets, respectively, when the best-scoring structures were compared (top 1). (B) When the set of five best-scoring structures were considered (top 5), DeepH3 energy identified better-, sameand worse-RMSD structures for 24, 16 and 9 out of 49 targets, respectively. (C) Comparison of the discrimination scores for Rosetta energy and DeepH3 energy.

### 2.4.2 Geometric potentials discriminate near-native CDR H3 loop structures

To evaluate the effectiveness of DeepH3 energy for identifying near-native structures, predicted DeepH3 geometric histograms were converted to potentials (Section 2) that were then evaluated on RosettaAntibody generated structure decoys. Reported RMSD values are measured between the heavy atoms of CDR H3 loops after aligning the $F_V$ backbone heavy atoms. When the best-scoring structures (top 1) by Rosetta energy and DeepH3 energy were compared, DeepH3 selected better-, same-, and worse-RMSD structures for 33, 6 and 10 out of 49 targets, respectively, with an average RMSD improvement of 1.4 Å (Figure 2.3A). When the set of five best-scoring structures (top

**Table 2.1:** Performance of DeepH3 energy versus alternative methods for selecting low-RMSD antibody decoys

| | Top 1 | | | | Top 5 | | | |
|---|---|---|---|---|---|---|---|---|
| Energy function | Better | Same | Worse | ΔRMSD | Better | Same | Worse | ΔRMSD |
| SBROD | 38 | 6 | 5 | -1.8 | 35 | 11 | 3 | -1.1 |
| KORP | 32 | 10 | 7 | -0.9 | 25 | 18 | 6 | -0.6 |
| Rosetta | 33 | 6 | 10 | -1.4 | 24 | 16 | 9 | -0.8 |

Top-1 metrics compare the RMSD of the best-scoring structure by DeepH3 energy against that of a given energy function. Top-5 metrics compare the lowest-RMSD structure among the five best-scoring structures selected by DeepH3 energy and that of a given energy function. The average difference in RMSD between the structures selected by DeepH3 energy and a given energy function is reported as ΔRMSD (Å). "Better," "Same," and "Worse" indicate the number of targets that achieve a lower, same, or higher RMSD, respectively, when scored by DeepH3.

5) by Rosetta energy and DeepH3 energy were considered, DeepH3 energy identified better-, same-, and worse RMSD structures for 24, 16 and 9 out of 49 targets, respectively, with an average RMSD improvement of 0.8 Å (Figure 2.3B). We also compared the ability of Rosetta energy and DeepH3 energy to discriminate between decoys for each benchmark target (Figure 2.3C, Table 2.2). The mean discrimination scores for Rosetta energy and DeepH3 energy across the benchmark were 1.7 and -12.2, respectively, indicating that DeepH3 was much more successful in general. When individual targets are considered, DeepH3 energy was successful in discriminating between decoys for 36 out of 49 targets, while Rosetta energy was successful for only 15 out of 49 targets.

To compare against alternative state-of-the-art methods, we also scored the RosettaAntibody decoy using SBROD [16] and KORP [17] (Tables 2.1 and 2.2). In a comparison of the top-rated structures from the decoy set, DeepH3 demonstrated improvements over SBROD (38 targets were better, 6 same and

**Table 2.2:** Discrimination score metrics for DeepH3 energy and several state-of-the-art energy functions

| Energy terms | Successful | Unsuccessful | Mean $D$ |
|---|---|---|---|
| SBROD | 8 | 41 | 3.7 |
| KORP | 21 | 28 | 0.2 |
| Rosetta | 15 | 34 | 1.7 |
| DeepH3 | 36 | 13 | -12.2 |
| d | 32 | 17 | -7.4 |
| $\omega$ | 32 | 17 | -7.8 |
| $\theta$ | 38 | 11 | -15.6 |
| $\phi$ | 36 | 13 | -9.6 |

DeepH3 energy is further divided into individual inter-residue geometries. Negative discrimination scores, $D$, are considered successful and positive are considered unsuccessful.

5 worse; average ΔRMSD of -1.8 Å). The comparison of the five top-scoring structures was similar (35 better, 11 same and 3 worse; ΔRMSD = -1.1 Å). In general, SBROD was unsuccessful in discriminating near-native decoys, with only 8 out of 49 benchmark targets having a negative discrimination score and an average $D$ of 3.7. DeepH3 also outperformed KORP among best-scoring structures (32 better, 10 same and 7 worse; ΔRMSD = -0.9 Å) and when comparing the lowest-RMSD structure among the five best-scoring decoys for each target (25 better, 18 same and 6 worse; ΔRMSD = -0.6 Å). KORP was generally unsuccessful in discriminating near-native CDR H3 loop decoys, with only 21 out of 49 targets having negative discrimination scores and an average D = 0.2.

To provide a better understanding of how predicted geometric potentials improve discrimination between CDR H3 structures, we detail two case studies: anti-ALOX12 scF$_V$ (scF$_V$ of mouse antibody with a 12-residue CDR H3 loop, PDB ID: 4H0H) and anti-dansyl mAb (humanized mouse antibody with

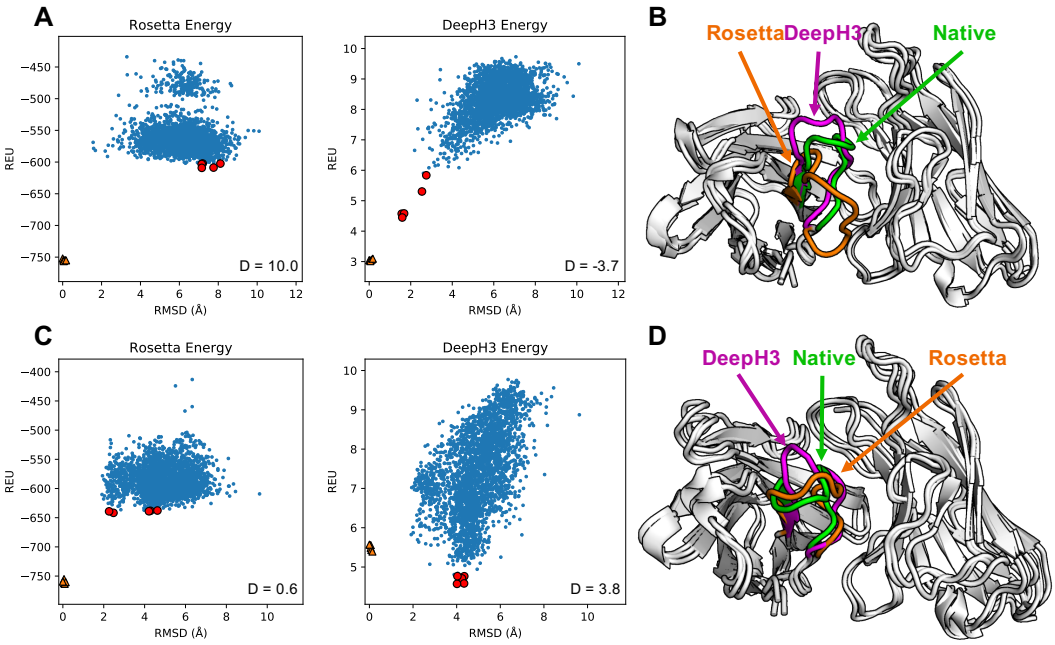


**Figure 2.4:** Results for two Rosetta antibody benchmark targets

(A) Plots of Rosetta energy and DeepH3 energy versus RMSD from the experimental structure for 2800 decoy structures for anti-ALOX12 $scF_V$. The five best-scoring structures in each funnel plot are indicated in red. Five relaxed native structures are plotted as orange triangles. (B) Experimental structure of anti-ALOX12 $scF_V$ (green) with best-scoring structures by Rosetta energy (orange, 7.2 Å RMSD) and DeepH3 energy (violet, 1.6 Å RMSD). (C) Plots of energy versus RMSD from the experimental structure for anti-dansyl mAb. (D) Experimental structure of anti-dansyl mAb (green) with best-scoring structures by Rosetta energy (orange, 2.5 Å RMSD) and DeepH3 energy (violet, 4.0 Å RMSD).

a 12-residue CDR H3 loop, PDB ID: 1DLF) [21]. Figure 2.4A and Figure 2.4C shows energy funnels for anti-ALOX12 and anti-dansyl, respectively, with the discrimination score calculated for each. For anti-ALOX12, Rosetta energy displays little ability to discriminate with structures ranging from 2 to 8 Å RMSD (D = 10.0). DeepH3 energy, however, earns a negative discrimination score (D = -3.7), indicating an ability to easily distinguish the near-native structures. The best scoring anti-ALOX12 decoy structures as selected by Rosetta energy

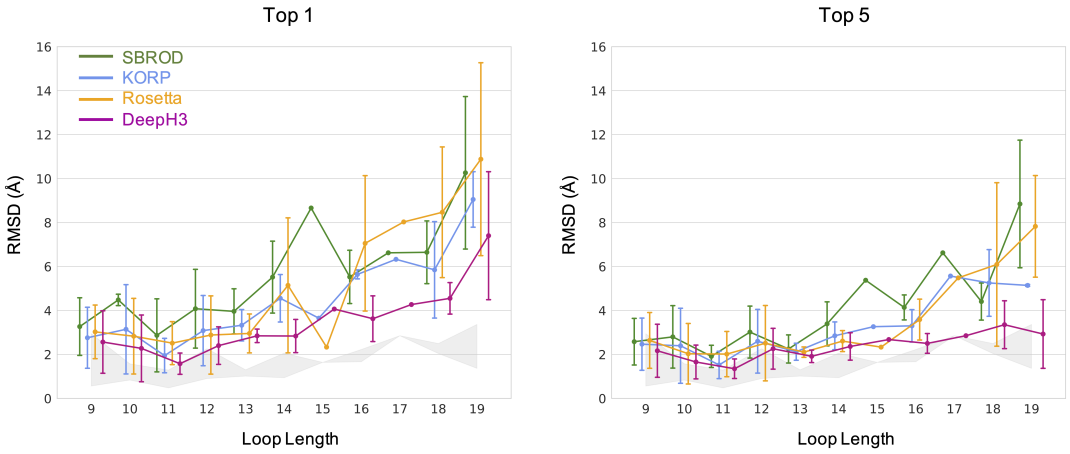


**Figure 2.5:** Performance of DeepH3 and alternative methods across various loop lengths

Comparison across loop lengths of the error in structures selected by SBROD (green), KORP (blue), Rosetta energy (orange) and DeepH3 score (violet). The shaded areas show the range of lowest RMSD values sampled for targets across loop lengths.

(orange, 7.2 Å RMSD) and DeepH3 energy (violet, 1.6 Å RMSD) are shown in Figure 2.4B and Figure 2.4D.

For anti-dansyl, Rosetta energy is generally unsuccessful in discriminating between decoys (D = 0.6), again with minor energetic differences across a wide range of RMSD values. DeepH3 energy appears to converge to an alternative loop conformation around 4 Å RMSD, resulting in a poor discrimination score (D = 3.8). Figure 2.4D shows the best-scoring anti-dansyl decoy structures as selected by Rosetta energy (orange, 2.5 Å RMSD) and DeepH3 energy (violet, 4.0 Å RMSD).

### 2.4.3 Longer loops remain a challenge

The Rosetta antibody benchmark dataset encompasses a diverse set of CDR H3 loop lengths. Longer loops introduce greater degrees of freedom (two DOFs

**Table 2.3:** Performance of DeepH3 energy versus alternative methods for selecting low-RMSD antibody decoys

| | Top 1 | | | | Top 5 | | | |
|---|---|---|---|---|---|---|---|---|
| Energy function | Better | Same | Worse | ΔRMSD | Better | Same | Worse | ΔRMSD |
| d | 27 | 9 | 13 | -1.1 | 22 | 14 | 13 | -0.5 |
| $\omega$ | 30 | 8 | 11 | -1.3 | 26 | 14 | 9 | -0.4 |
| $\theta$ | 31 | 7 | 11 | -1.5 | 23 | 13 | 13 | -0.7 |
| $\phi$ | 29 | 7 | 13 | -1.4 | 26 | 14 | 9 | -0.8 |

Top-1 metrics compare the RMSD of the best-scoring structure by Rosetta energy against that of a given DeepH3 potential. Top-5 metrics compare the lowest-RMSD structure among the five best-scoring structures selected by Rosetta energy and that of a given DeepH3 potential. The average difference in RMSD between the structures selected by a given DeepH3 potential and Rosetta energy is reported as ΔRMSD (Å).

per residue), and thus present additional challenges to effective sampling and discrimination. To investigate the performance of DeepH3 across loop lengths, we sub-divided the benchmark targets by length and compared to three alternative scoring methods: SBROD, KORP and the Rosetta energy function (Figure 2.5). For nearly every loop length considered, DeepH3 identified the lowest RMSD structures according to the top-1 and top-5 criteria (see above). For several loop lengths, DeepH3 identified decoys near the lowest-RMSD for particular targets in the dataset, as indicated by the shaded region. In general, the average RMSD increased with loop length across all four methods, though DeepH3 displayed notable consistency across loop lengths according to the top-5 criteria.

### 2.4.4 Orientation potentials are more effective than distance potentials

We also evaluated the utility of individual geometric potentials for selecting low-RMSD decoys (Table 2.3). Notably, when DeepH3 distance potentials

alone were used, performance was only moderately better than Rosetta energy. When the best-scoring structures by Rosetta energy and distance potentials were compared, distance potentials selected better-, same-, and worse-RMSD structures for 27, 9 and 13 out of 49 targets, respectively, with an average RMSD improvement of 1.1 Å. When the set of five best-scoring structures by Rosetta energy and distance potentials were considered, DeepH3 energy identified better-, same-, and worse-RMSD structures for 22, 14 and 13 out of 49 targets, respectively, with an average RMSD improvement of 0.5 Å. Individual orientation potentials were more effective at selecting low-RMSD decoys than distance, even matching or outperforming the total DeepH3 energy by some metrics. We also calculated discrimination scores for each geometric potential (Table 2.2). Distance and $\omega$ orientation potentials displayed the weakest performance among geometric potentials but still showed significant improvement over Rosetta energy, with 32 out of 49 simulations being successful for both. The other orientation potentials produced more successful simulations and lower mean discrimination scores.

### 2.4.5 DeepH3 effectively predicts new CDR H3 structures de novo

The ultimate goal of DeepH3 was to improve the de novo prediction of CDR H3 loops. Towards this end, we used DeepH3 to create potentials that we then used in Rosetta for de novo structure prediction of the CDR H3 loops (Section 2). The average ($\pm$ SD) RMSD of the best-scoring structures generated with DeepH3 potentials for each target (top 1) was 2.2 $\pm$ 1.1 Å. When the set of five best-scoring structures for each target were considered (top 5), the average

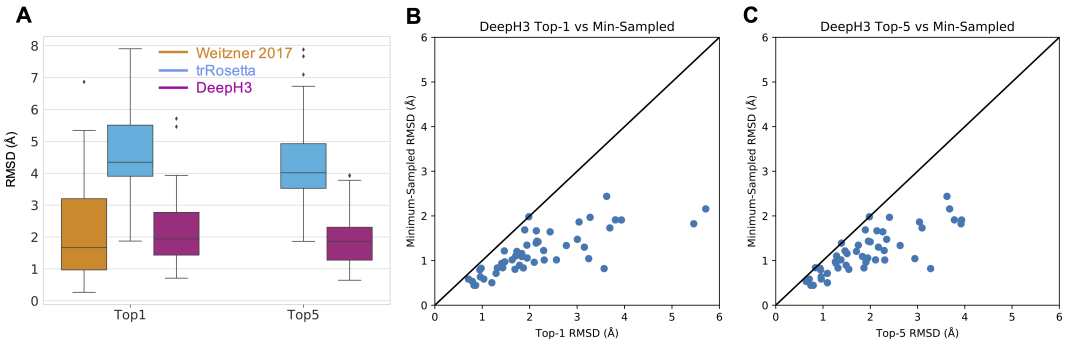


**Figure 2.6:** Performance of DeepH3 for de novo CDR H3 loop structure prediction

(A) DeepH3 achieves lower average RMSD (2.2 $\pm$ 1.1 Å) than trRosetta (4.7 $\pm$ 1.4 Å) and ties Weitzner et al. (2.2 $\pm$ 1.5 Å) [24] when the best scoring structures for each target were compared (top 1). When the lowest-RMSD structure among the five best-scoring structures were considered (top 5), DeepH3 (1.9 $\pm$ 0.9 Å) outperformed trRosetta (4.3 $\pm$ 1.3 Å). Top-5 metrics were not available for Weitzner et al. (B) Comparison of the minimum RMSD sampled by DeepH3 to the RMSD of the best-scoring structure (top 1) for each target. (C) Comparison of the minimum RMSD sampled by DeepH3 to the lowest RMSD within the set of five best-scoring structures (top 5) for each target.

RMSD fell to 1.9 $\pm$ 0.9 Å. We compare the best-scoring structures generated with DeepH3 potentials to those published by Weitzner et al. [21] (Figure 2.6A) and find effectively equivalent performance ($\Delta$RMSD < 0.1 Å) (Top-5 metrics were not reported by Weitzner et al.). The recently published trRosetta provides another deep learning prediction method to compare. trRosetta is trained broadly on diverse protein structures, and DeepH3 has fewer input features (just sequence). trRosetta is designed for single-chain proteins, so we omitted the light chain and predicted structures for the heavy chain alone. On the same benchmark, trRosetta achieves average accuracies of 4.7 $\pm$ 1.4 Å(top 1) and 4.3 $\pm$ 1.3 Å(top 5, Figure 2.6A). Compared to trRosetta, DeepH3's top-1 and top-5 metrics are 2.5 Å and 2.4 Å RMSD better, respectively.

To better understand the sampling performance of DeepH3, we compared the lowest-RMSD decoy sampled to the best-scoring (top 1, Figure 2.6B) and the lowest-RMSD among the five best-scoring (top 5, Figure 2.6C). DeepH3 samples structures with sub-angstrom RMSD for 38.8% of the targets and 95.9% for <2 Å. On the other hand, DeepH3 is able to identify a sub-angstrom decoy as the best-scoring structure (top 1) for 14.3% of targets and 55.1% for <2 Å. When considering the set of five best-scoring decoys (top 5), DeepH3 identifies a sub-angstrom decoy for 18.4% of targets and 63.2% for <2 Å. These results are promising and point to possibility of further refining the DeepH3 geometric potentials for de novo prediction.

## 2.5 Discussion

The results here suggest that the significant advances by deep learning approaches in general protein structure can be realized in subproblems in structural modeling. Specifically, we demonstrate that a deep residual network can effectively capture the local inter-residue interactions that define antibody CDR H3 loop structure. DeepH3 achieves these results without MSAs and co-evolutionary data, while using significantly fewer residual blocks (3 1D and 25 2D blocks) than similar networks, such as AlphaFold (220 2D blocks) [13], RaptorX (6 1D and 60 2D blocks) [19, 14] and trRosetta (61 2D blocks) [15]. Fewer blocks may suffice because we limited our focus to antibodies, which are highly conserved, rather than the entire universe of protein structures. By omitting MSAs and co-evolutionary data, we demonstrate that these features, which have seemed essential to the advances in general protein

structure prediction, may not be necessary for some subproblems. In the future, similar specialized networks could achieve enhanced performance in other challenging areas of protein structure prediction, but further research is required.

Breakdown of DeepH3 energy into individual geometric potentials revealed that inter-residue orientations were significantly more effective for scoring CDR H3 loop structures than distances. This finding was surprising, given the improvements that distances alone have enabled in general protein structure prediction. This observation could also underlie the improved performance of trRosetta compared to methods that do not use orientations. Alternatively, distance restraints may be more effective at placing residues globally while local interactions in loops are better captured by inter-residue orientations.

Application of DeepH3 to de novo prediction of CDR H3 loop structures highlights the promise of deep learning in this challenging area. Comparison with the results from Weitzner et al., which leveraged an explicit H3-kink geometric constraint [21], demonstrates that DeepH3 effectively learned challenging features of H3 loop structure. While this work focused only on the CDR H3 loop, we anticipate that applying DeepH3 to other aspects of antibody structure prediction may yield further advances. Because DeepH3 learns from full $F_V$ heavy and light chain sequences, the current network may already capture other critical aspects of antibody structure prediction (VL-VH orientations [20], non-H3 CDR loop conformations [2] etc.), though future work will be necessary to explore these areas.

# References


[1] Cyrus Chothia, Arthur M Lesk, Anna Tramontano, Michael Levitt, Sandra J Smith-Gill, Gillian Air, Steven Sheriff, Eduardo A Padlan, David Davies, William R Tulip, et al. "Conformations of immunoglobulin hypervariable regions". In: *Nature* 342.6252 (1989), pp. 877–883.

[2] Benjamin North, Andreas Lehmann, and Roland L Dunbrack Jr. "A new clustering of antibody CDR loop conformations". In: *Journal of Molecular Biology* 406.2 (2011), pp. 228–256.

[3] Juan C Almagro, Alexey Teplyakov, Jinquan Luo, Raymond W Sweet, Sreekumar Kodangattil, Francisco Hernandez-Guzman, and Gary L Gilliland. *Second Antibody Modeling Assessment (AMA-II)*. 2014.

[4] Monica Berrondo, Susana Kaufmann, and Manuel Berrondo. "Automated Aufbau of antibody structures from given sequences using Macromoltek's SmrtMolAntibody". In: *Proteins: Structure, Function, and Bioinformatics* 82.8 (2014), pp. 1636–1645.

[5] Marc Fasnacht, Ken Butenhof, Anne Goupil-Lamy, Francisco Hernandez-Guzman, Hongwei Huang, and Lisa Yan. "Automated antibody structure prediction using Accelrys tools: results and best practices". In: *Proteins: Structure, Function, and Bioinformatics* 82.8 (2014), pp. 1583–1598.

[6] Johannes KX Maier and Paul Labute. "Assessment of fully automated antibody homology modeling protocols in molecular operating environment". In: *Proteins: Structure, Function, and Bioinformatics* 82.8 (2014), pp. 1599–1610.

[7] Hiroki Shirai, Kazuyoshi Ikeda, Kazuo Yamashita, Yuko Tsuchiya, Jamica Sarmiento, Shide Liang, Tatsuaki Morokata, Kenji Mizuguchi, Junichi Higo, Daron M Standley, et al. "High-resolution modeling of antibody structures by a combination of bioinformatics, expert knowledge, and

molecular simulations". In: *Proteins: Structure, Function, and Bioinformatics* 82.8 (2014), pp. 1624–1635.

[8] Brian D Weitzner, Daisuke Kuroda, Nicholas Marze, Jianqing Xu, and Jeffrey J Gray. "Blind prediction performance of RosettaAntibody 3.0: grafting, relaxation, kinematic loop modeling, and full CDR optimization". In: *Proteins: Structure, Function, and Bioinformatics* 82.8 (2014), pp. 1611–1623.

[9] Kai Zhu, Tyler Day, Dora Warshaviak, Colleen Murrett, Richard Friesner, and David Pearlman. "Antibody structure determination using a combination of homology modeling, energy-based refinement, and loop prediction". In: *Proteins: Structure, Function, and Bioinformatics* 82.8 (2014), pp. 1646–1655.

[10] Inbal Sela-Culang, Shahar Alon, and Yanay Ofran. "A systematic comparison of free and bound antibodies reveals binding-related conformational changes". In: *The Journal of Immunology* 189.10 (2012), pp. 4890–4899.

[11] Susan H Eshleman, Oliver Laeyendecker, Kai Kammers, Athena Chen, Mariya V Sivay, Sanjay Kottapalli, Brandon M Sie, Tiezheng Yuan, Daniel R Monaco, Divya Mohan, et al. "Comprehensive profiling of HIV antibody evolution". In: *Cell Reports* 27.5 (2019), pp. 1422–1433.

[12] Xueling Wu, Tongqing Zhou, Jiang Zhu, Baoshan Zhang, Ivelin Georgiev, Charlene Wang, Xuejun Chen, Nancy S Longo, Mark Louder, Krisha McKee, et al. "Focused evolution of HIV-1 neutralizing antibodies revealed by structures and deep sequencing". In: *Science* 333.6049 (2011), pp. 1593–1602.

[13] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792 (2020), pp. 706–710.

[14] Jinbo Xu. "Distance-based protein folding powered by deep learning". In: *Proceedings of the National Academy of Sciences* 116.34 (2019), pp. 16856–16865.

[15] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. "Improved protein structure prediction using predicted interresidue orientations". In: *Proceedings of the National Academy of Sciences* 117.3 (2020), pp. 1496–1503.

[16] Mikhail Karasikov, Guillaume Pagès, and Sergei Grudinin. "Smooth orientation-dependent scoring function for coarse-grained protein quality assessment". In: *Bioinformatics* 35.16 (2019), pp. 2801–2808.

[17] José Ramón López-Blanco and Pablo Chacón. "KORP: knowledge-based 6D potential for fast protein and loop modeling". In: *Bioinformatics* 35.17 (2019), pp. 3013–3019.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[19] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. "Accurate de novo prediction of protein contact map by ultra-deep learning model". In: *PLOS Computational Biology* 13.1 (2017), e1005324.

[20] Nicholas A Marze, Sergey Lyskov, and Jeffrey J Gray. "Improved prediction of antibody VL–VH orientation". In: *Protein Engineering, Design and Selection* 29.10 (2016), pp. 409–418.

[21] Brian D Weitzner and Jeffrey J Gray. "Accurate structure prediction of CDR H3 loops enabled by a novel structure-based C-terminal constraint". In: *The Journal of Immunology* 198.1 (2017), pp. 505–515.

[22] Jared Adolf-Bryfogle, Qifang Xu, Benjamin North, Andreas Lehmann, and Roland L Dunbrack Jr. "PyIgClassify: a database of antibody CDR structural classifications". In: *Nucleic Acids Research* 43.D1 (2015), pp. D432–D438.

[23] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. "SAbDab: the structural antibody database". In: *Nucleic Acids Research* 42.D1 (2014), pp. D1140–D1146.

[24] Brian D Weitzner, Jeliazko R Jeliazkov, Sergey Lyskov, Nicholas Marze, Daisuke Kuroda, Rahel Frick, Jared Adolf-Bryfogle, Naireeta Biswas, Roland L Dunbrack Jr, and Jeffrey J Gray. "Modeling and docking of antibody structures with Rosetta". In: *Nature Protocols* 12.2 (2017), pp. 401–416.

[25] Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. "The Rosetta all-atom energy function for macromolecular modeling and design". In: *Journal of Chemical Theory and Computation* 13.6 (2017), pp. 3031–3048.

[26] Patrick Conway, Michael D Tyka, Frank DiMaio, David E Konerding, and David Baker. "Relaxation of backbone bond geometry improves protein energy landscape modeling". In: *Protein Science* 23.1 (2014), pp. 47–55.

[27] Daniel J Mandell, Evangelos A Coutsias, and Tanja Kortemme. "Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling". In: *Nature Methods* 6.8 (2009), pp. 551–552.

[28] Amelie Stein and Tanja Kortemme. "Improvements to robotics-inspired conformational sampling in rosetta". In: *PLOS One* 8.5 (2013), e63090.

# Chapter 3

# Antibody structure prediction using interpretable deep learning

Adapted from Jeffrey A Ruffolo, Jeremias Sulam, and Jeffrey J Gray. "Antibody structure prediction using interpretable deep learning". *Patterns* 3.2 (2022), p. 100406. Reproduced with permission.

## 3.1 Abstract

Therapeutic antibodies make up a rapidly growing segment of the biologics market. However, rational design of antibodies is hindered by reliance on experimental methods for determining antibody structures. Here, we present DeepAb, a deep learning method for predicting accurate antibody $F_V$ structures from sequence. We evaluate DeepAb on a set of structurally diverse, therapeutically relevant antibodies and find that our method consistently outperforms the leading alternatives. Previous deep learning methods have operated as "black boxes" and offered few insights into their predictions. By

introducing a directly interpretable attention mechanism, we show our network attends to physically important residue pairs (e.g., proximal aromatics and key hydrogen bonding interactions). Finally, we present a novel mutant scoring metric derived from network confidence and show that for a particular antibody, all eight of the top-ranked mutations improve binding affinity. This model will be useful for a broad range of antibody prediction and design tasks.

## 3.2 Introduction

The adaptive immune system of vertebrates is capable of mounting robust responses to a broad range of potential pathogens. Critical to this flexibility are antibodies, which are specialized to recognize a diverse set of molecular patterns with high affinity and specificity. This natural role in the defense against foreign particles makes antibodies an increasingly popular choice for therapeutic development [1, 2]. Presently, the design of therapeutic antibodies comes with significant barriers [1]. For example, the rational design of antibody-antigen interactions often depends upon an accurate model of antibody structure. However, experimental methods for protein structure determination such as crystallography, NMR, and cryo-EM are low throughput and time consuming.

Antibody structure consists of two heavy and two light chains that assemble into a large Y-shaped complex. The crystallizable fragment ($F_C$) region is involved in immune effector function and is highly conserved within isotypes. The variable fragment ($F_V$) region is responsible for antigen binding through

a set of six hypervariable loops that form a complementarity determining region (CDR). Structural modeling of the $F_V$ is critical for understanding the mechanism of antigen binding and for rational engineering of specific antibodies. Most methods for antibody $F_V$ structure prediction employ some form of grafting, by which pieces of previously solved $F_V$ structures with similar sequences are combined to form a predicted model [3, 4, 5, 6]. Because much of the $F_V$ is structurally conserved, these techniques are typically able to produce models with an overall root-mean-square deviation (RMSD) less than 1 Å from the native structure. However, the length and conformational diversity of the third CDR loop of the heavy chain (CDR H3) make it difficult to identify high-quality templates. Further, the H3 loop's position between the heavy and light chains makes it dependent on the chain orientation and multiple adjacent loops [7, 8]. Thus the CDR H3 loop presents a longstanding challenge for $F_V$ structure prediction methods [9].

Machine learning methods have become increasing popular for protein structure prediction and design problems [10]. Specific to antibodies [11], machine learning has been applied to predict developability [12], improve humanization [13], generate sequence libraries [14], and predict antigen interactions [15, 16]. In this work, we build on advances in general protein structure prediction [17, 18, 19] to predict antibody $F_V$ structures. Our method consists of a deep neural network for predicting inter-residue distances and orientations and a Rosetta-based protocol for generating structures from network predictions. We show that deep learning approaches can predict more

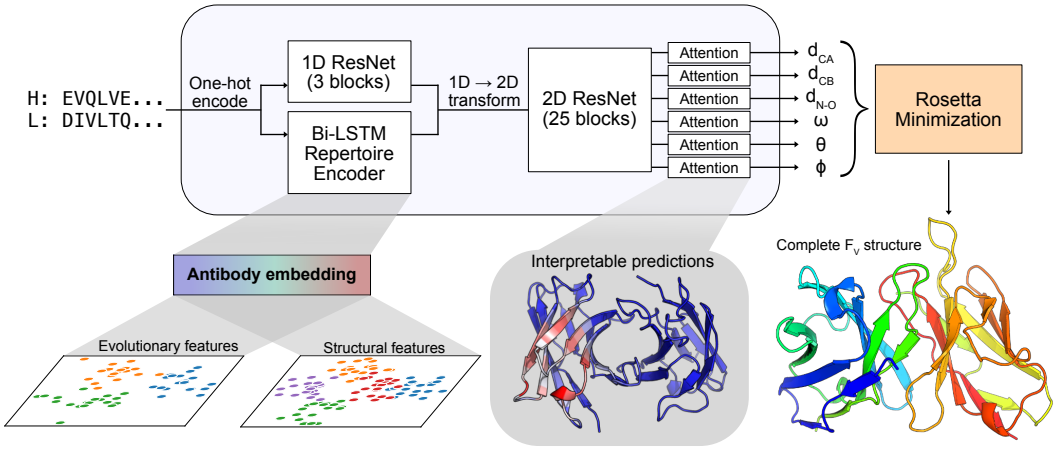


**Figure 3.1:** Diagram of DeepAb method for antibody structure prediction

Starting from heavy and light chain sequences, the network predicts a set of inter-residue geometries describing the $F_V$ structure. Predictions are used for guided structure realization with Rosetta. Two interpretable components of the network are highlighted: a pretrained antibody sequence model and output attention mechanisms.

accurate structures than grafting-based alternatives, particularly for the challenging CDR H3 loop. The network used in this work is designed to be directly interpretable, providing insights that could assist in structural understanding and antibody engineering efforts. We conclude by demonstrating that our network can distinguish mutational variants with improved binding using a prediction confidence metric. To facilitate further studies, all the code for this work, as well as pretrained models, is provided.

## 3.3 Results

### 3.3.1 Overview of the method

Our method for antibody structure prediction, DeepAb, consists of two main stages (Figure 3.1). The first stage is a deep residual convolutional network

that predicts $F_V$ structure, represented as relative distances and orientations between pairs of residues. The network requires only heavy and light chain sequences as input and is designed with interpretable components to provide insight into model predictions. The second stage is a fast Rosetta-based protocol for structure realization using the predictions from the network.

**Predicting inter-residue geometries from sequence**

Due to the limited number of $F_V$ crystal structures available for supervised learning, we sought to make use of the abundant immunoglobin sequences from repertoire sequencing studies [20]. We leveraged the power of unsupervised representation learning to embed general patterns from immunoglobin sequences that are not evident in the small subset with known structures into a latent representation. Although transformer models have become increasingly popular for unsupervised learning on protein sequences [21, 22, 23], we chose a recurrent neural network (RNN) model for ease of training on the limited data available. The fixed-size hidden state of RNNs forms an explicit information bottleneck ideal for representation learning. In the recent UniRep method, RNNs were demonstrated to learn rich feature representations from protein sequences when trained on next-amino-acid prediction [24]. For our purposes, we developed an RNN encoder-decoder model [25]; the encoder is a bidirectional long short-term memory (biLSTM) and the decoder is a long short-term memory (LSTM) [26]. Briefly, the encoder learns to summarize an input sequence residue-by-residue into a fixed-size hidden state. This hidden state is transformed into a summary vector and passed to the decoder, which learns to reconstruct the original sequence one residue at a time. The model

is trained using cross-entropy loss on a set of 118,386 paired heavy and light chain sequences from the Observed Antibody Space (OAS) database [27]. After training the network, we generated embeddings for antibody sequences by concatenating the encoder hidden states for each residue. These embeddings are used as features for the structure prediction model described below.

The choice of protein structure representation is critical for structure prediction methods [10]. We represent the $F_V$ structure as a set of inter-residue distances and orientations, similar to previous methods for general protein structure prediction [18, 19]. Specifically, we predict inter-residue distances between three pairs of atoms ($C_\alpha$—$C_\alpha$, $C_\beta$-$C_\beta$, $N$—$O$) and the set of inter-residue dihedrals ($\omega$: $C_\alpha$—$C_\beta$—$C_\beta$—$C_\alpha$, $\theta$: $N$—$C_\alpha$—$C_\beta$-$C_\beta$) and planar angles ($\phi$: $C_\alpha$-$C_\beta$-$C_\beta$) first described by Yang et al. and shown in their Figure 1 [18]. Each output geometry is discretized into 36 bins, with an additional bin indicating distant residue pairs $d_{C_\alpha} > 18$ Å. All distances are predicted in the range of 0-18 Å, with a bin width of 0.5 Å. Dihedral and planar angles are discretized uniformly into bins of 10 and 5 degrees, respectively.

The general architecture of the structure prediction network is similar to our previous method for CDR H3 loop structure prediction [28], with two notable additions: embeddings from the pretrained language model and interpretable attention layers (Figure 3.1). The network takes as input the concatenated heavy and light chain sequences. The concatenated sequence is one-hot encoded and passed through two parallel branches: a 1D ResNet and the pretrained language model. The outputs of the branches are combined and transformed into pairwise data. The pairwise data pass through a deep

2D ResNet that constitutes the main component of the predictive network. Following the 2D ResNet, the network separates into six output branches, corresponding to each type of geometric measurement. Each output branch includes a recurrent criss-cross attention module, allowing each residue pair in the output to aggregate information from all other residue pairs. The attention layers provide interpretability that is often missing from protein structure prediction models.

We opted to train with focal loss [29] rather than cross-entropy loss to improve the calibration of model predictions, as models trained with cross-entropy loss have been demonstrated to overestimate the likelihood of their predicted labels [30]. We pay special attention to model calibration as later in this work we attempt to distinguish between potential antibody variants on the basis of prediction confidence, which requires greater calibration. The model is trained on a nonredundant (at 99% sequence identity) set of 1,692 $F_V$ structures from the Structural Antibody Database (SAbDab) [3]. The pretrained language model, used as a feature extractor, is not updated while training the predictor network.

**Structure realization**

Similar to previous methods for general protein structure prediction [17, 18, 19], we used constrained minimization to generate full 3D structures from network predictions. Unlike previous methods, which typically begin with some form of $\phi$-$\psi$ torsion sampling, we created initial models via multi-dimensional scaling (MDS). We opted to build initial structures through MDS, rather than

torsion sampling, due to the high conservation of the framework structural regions. Through MDS, we can obtain accurate 3D coordinates for the conserved framework residues, thus bypassing costly sampling for much of the antibody structure [31]. As a reminder, the relative positions of all backbone atoms are fully specified by the predicted $L \times L$ inter-residue $_{C_\alpha}$ , $\omega$, $\theta$, and $\phi$ geometries. Using the modal-predicted output bins for these four geometries, we construct a distance matrix between backbone atoms. From this distance matrix, MDS produces an initial set of 3D coordinates that are subsequently refined through constrained minimization.

Network predictions for each output geometry were converted to energetic potentials by negating the raw model logits (i.e., without softmax activation). These discrete potentials were converted to continuous constraints using a cubic spline function. Starting from the MDS model, the constraints are used to guide quasi-Newton minimization (L-BFGS) within Rosetta [32, 33]. First, the constraints are jointly optimized with a simplified Rosetta centroid energy function to produce a coarse-grained $F_V$ structure with the sidechains represented as a single atom. Next, constrained full-atom relaxation was used to introduce sidechains and remove clashes. After relaxation, the structure was minimized again with constraints and the Rosetta full-atom energy function (ref2015) [34]. This optimization procedure was repeated to produce 50 structures, and the lowest energy structure was selected as the final model. Although we opted to produce 50 candidate structures, five should be sufficient in practice due to the high convergence of the protocol (Figure 3.17). Five candidate structures can typically be predicted in 10 min on a standard

CPU, making our method slower than grafting-only approaches (seconds to minutes per sequence), but significantly faster than extensive loop sampling (hours per sequence).

### 3.3.2 Benchmarking methods for Fv structure prediction

To evaluate the performance of our method, we chose two independent test sets. The first is the RosettaAntibody benchmark set (47 targets), which has previously been used to evaluate antibody structure prediction methods [8, 28, 35]. The second is a set of clinical-stage therapeutic antibodies (45 targets), which was previously assembled to study antibody developability [36]. Taken together, these sets represent a structurally diverse, therapeutically relevant benchmark for comparing antibody $F_V$ structure prediction methods.

#### Deep learning outperforms grafting methods

Although our method bears resemblance to deep learning methods for general protein structure prediction, we opted to compare to antibody-specific methods as we have previously found general methods to not yet be capable of producing high-quality structures of the challenging CDR loops [28]. Instead, we compared the performance of our method on the RosettaAntibody benchmark and therapeutic benchmark to three antibody-specific alternative methods: RosettaAntibody-G [4, 6], RepertoireBuilder [5], and ABodyBuilder [3]. Each of these methods is based on a grafting approach, by which complete $F_V$ structures are assembled from sequence-similar fragments of previously solved structures. To produce the fairest comparison, we excluded structures

**Table 3.1:** Performance of Fv structure prediction methods on benchmarks

| Method | OCD | H Fr (Å) | H1 (Å) | H2 (Å) | H3 (Å) | L Fr (Å) | L1 (Å) | L2 (Å) | L3 (Å) |
|---|---|---|---|---|---|---|---|---|---|
| RosettaAntibody Benchmark | | | | | | | | | |
| RosettaAntibody-G | 5.19 | 0.57 | 1.22 | 1.14 | 3.48 | 0.67 | 0.80 | 0.87 | 1.06 |
| RepertoireBuilder | 5.26 | 0.58 | 0.86 | 1.00 | 2.94 | 0.51 | 0.63 | 0.52 | 1.03 |
| ABodyBuilder | 4.69 | 0.50 | 0.99 | 0.88 | 2.94 | 0.49 | 0.72 | 0.52 | 1.09 |
| DeepAb | 3.67 | 0.43 | 0.72 | 0.85 | 2.33 | 0.42 | 0.55 | 0.45 | 0.86 |
| RosettaAntibody Benchmark | | | | | | | | | |
| RosettaAntibody-G | 5.43 | 0.63 | 1.42 | 1.05 | 3.77 | 0.55 | 0.89 | 0.83 | 1.48 |
| RepertoireBuilder | 4.37 | 0.62 | 0.91 | 0.96 | 3.13 | 0.47 | 0.71 | 0.52 | 1.08 |
| ABodyBuilder | 4.37 | 0.49 | 1.05 | 1.02 | 3.00 | 0.45 | 1.04 | 0.50 | 1.35 |
| DeepAb | 3.52 | 0.40 | 0.77 | 0.68 | 2.52 | 0.37 | 0.60 | 0.42 | 1.02 |

Orientational coordinate distance (OCD) is a unitless quantity calculated by measuring the deviation from native of four heavy-light chain coordinates.8 Heavy chain framework (H Fr) and light chain framework (L Fr) RMSDs are measured after superimposing the heavy and light chains, respectively. CDR loop RMSDs are measured using the Chothia loop definitions after superimposing the framework region of the corresponding chain. All RMSDs are measured over backbone heavy atoms.

with greater than 99% sequence identity for the whole $F_V$ from use for grafting (similar to our training data set). We evaluated each method according to the backbone heavy-atom RMSD of the CDR loops and the framework regions of both chains. We also measured the orientational coordinate distance (OCD) [8], a metric for heavy-light chain orientation accuracy. OCD is calculated as the sum of the deviations from native of four orientation coordinates (packing angle, interdomain distance, heavy-opening angle, light-opening angle) divided by the standard deviation of each coordinate [8]. The results of the benchmark are summarized in Table 3.1.

Our deep learning method showed improvement over all grafting-based methods on every metric considered. On both benchmarks, the structures predicted by our method achieved an average OCD less than 4, indicating that predicted structures were typically within one standard deviation of the

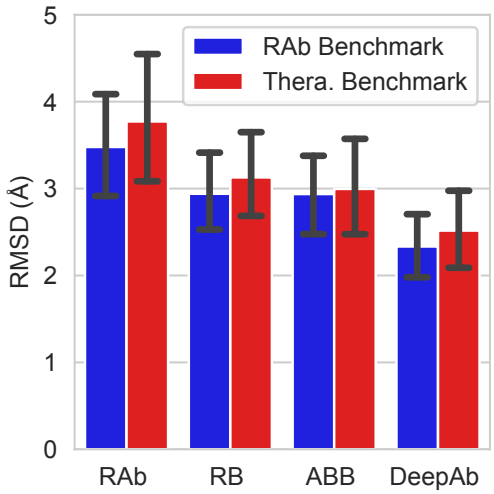


**Figure 3.2:** Comparison of CDR H3 loop structure prediction accuracy

Average RMSD of H3 loops predicted by RosettaAntibody-G (RAb), RepertoireBuilder (RB), ABodyBuilder (ABB), and DeepAb on the two benchmarks. Error bars show standard deviations for each method on each benchmark.

native structure for each of the orientational coordinates. All of the methods predicted with sub-angstrom accuracy on the heavy and light chain framework regions, which are highly conserved. Still, our method achieved average RMSD improvements of 14%-18% for the heavy chain framework and 16%-17% for light chain framework over the next best methods on the benchmarks. We also observed consistent improvement over grafting methods for CDR loop structure prediction.

### Comparison of CDR H3 loop modeling accuracy

The most significant improvements by our method were observed for the CDR H3 loop (Figure 3.2). On the RosettaAntibody benchmark, our method predicted H3 loop structures with an average RMSD of 2.33 Å($\pm$ 1.32 Å), a

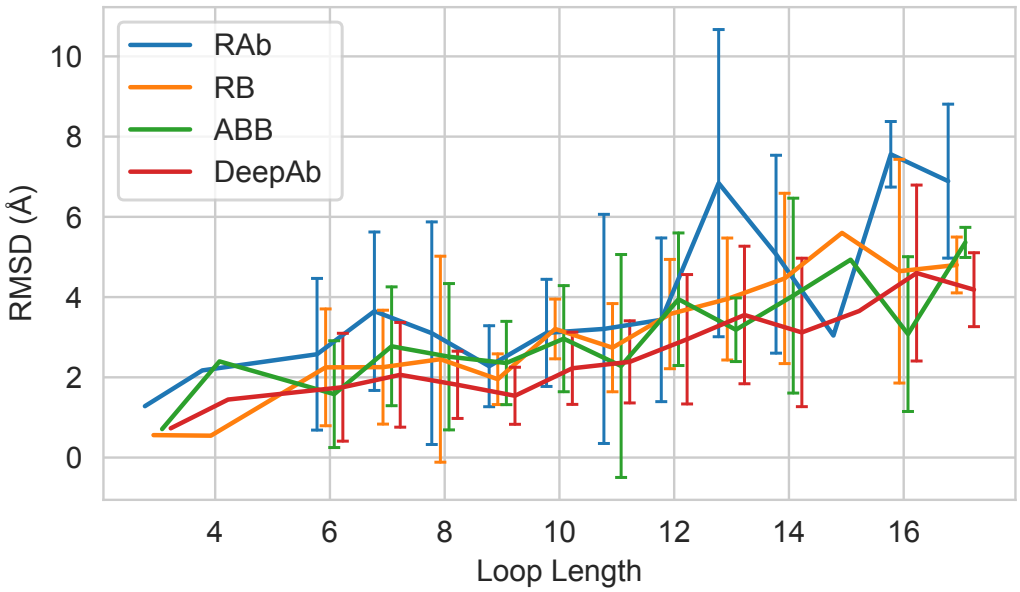


**Figure 3.3:** Length dependency of CDR H3 loop structure prediction accuracy

Average RMSD of H3 loops by length for all benchmark targets. Error bars show standard deviations for loop lengths corresponding to more than one target.

22% improvement over the next best method. On the therapeutic benchmark, our method had an average H3 loop RMSD of 2.52 Å($\pm$ 1.50 Å), a 16% improvement over the next best method. The difficulty of predicting CDR H3 loop structures is due in part to the wide range of observed loop lengths. To understand the impact of H3 loop length on our method's performance, we compared the average RMSD for each loop length across both benchmarks (Figure 3.3). In general, all of the methods displayed degraded performance with increasing H3 loop length. However, DeepAb typically produced the most accurate models for each loop length.

We also examined the performance of each method on individual benchmark targets. In Figure 3.4, we plot the CDR H3 loop RMSD of our method

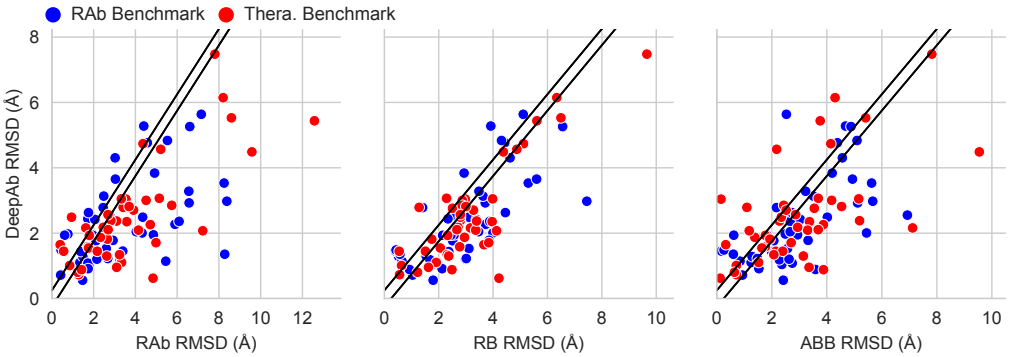


**Figure 3.4:** Head-to-head CDR H3 loop structure prediction comparison

Direct comparison of DeepAb and alternative methods H3 loop RMSDs, with diagonal band indicating predictions that were within $\pm$ 0.25 Å.

versus that of the alternative methods. Predictions with an RMSD difference less than 0.25 Å(indicated by diagonal bands) were considered equivalent in quality. When compared to RosettaAntibody-G, RepertoireBuilder, and ABodyBuilder, our method predicted more/less accurate H3 loop structures for 64/17, 59/16, and 53/22 out of 92 targets, respectively. Remarkably, our method was able to predict nearly half of the H3 loop structures (42 of 92) to within 2 Å RMSD. RosettaAntibody-G, RepertoireBuilder, and ABodyBuilder achieved RMSDs of 2 Å or better on 26, 23, and 26 targets, respectively.

**Accurate prediction of challenging, therapeutically relevant targets**

To underscore and illustrate the improvements achieved by our method, we highlight two examples from the benchmark sets. The first is rituximab, an anti-CD20 antibody from the therapeutic benchmark (PDB: 3PP3) [37]. In Figure 3.5, the native structure of the 12-residue rituximab H3 loop (white) is compared to our method's prediction (green, 2.1 Å RMSD) and the predictions from the grafting methods (blue, 3.3-4.1 Å RMSD). The prediction

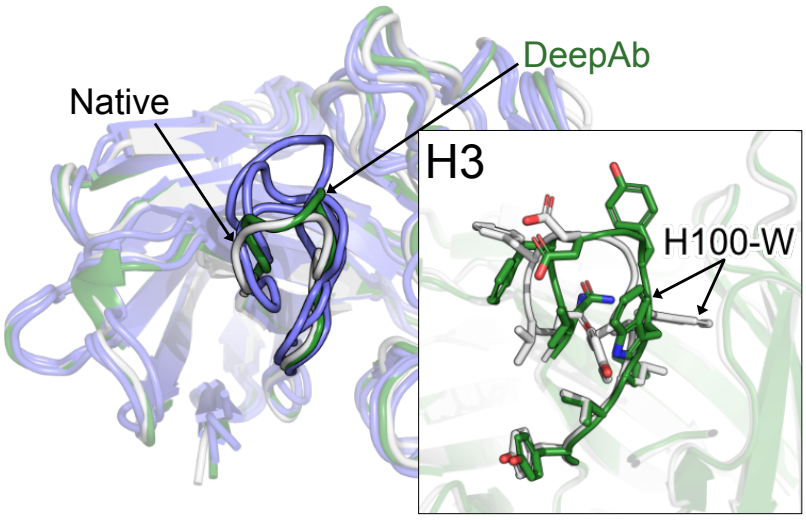


**Figure 3.5:** Rituximab CDR H3 loop structure prediction comparison

Comparison of native rituximab H3 loop structure (white, PDB: 3PP3) to predictions from DeepAb (green, 2.1 Å RMSD) and alternative methods (blue, 3.3-4.1 Å RMSD).

from our method captures the general topology of the loop well, even placing many of the side chains near the native structure. The second example is sonepcizumab, an anti-sphingosine-1-phosphate antibody from the RosettaAntibody benchmark (PDB: 3I9G) [38]. In Figure 3.6, the native structure of the 12-residue H3 loop (white) is compared to our method's prediction (green, 1.8 Å) and the predictions from the grafting methods (blue, 2.9-3.9 Å RMSD). Again, our method captures the overall shape of the loop well, enabling accurate placement of several side chains. Interestingly, the primary source of error by our method in both cases is a tryptophan residue (around position H100) facing in the incorrect direction.

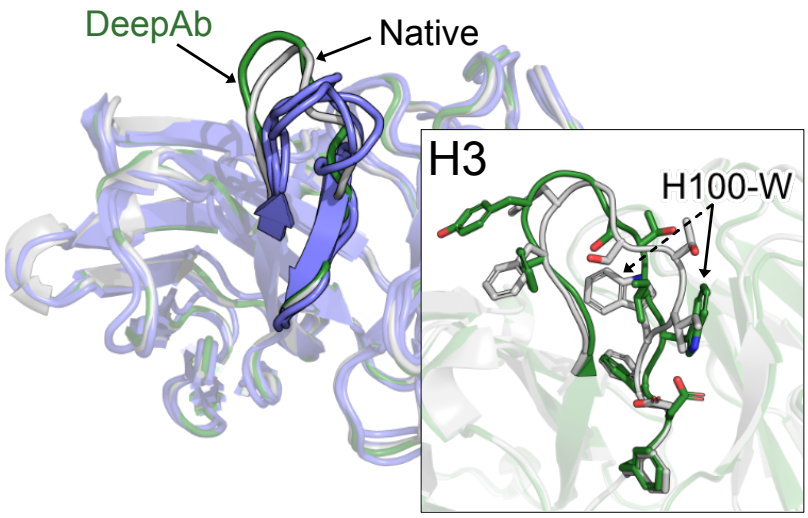


**Figure 3.6:** Sonepcizumab CDR H3 loop structure prediction comparison

Comparison of native sonepcizumab H3 loop structure (white, PDB: 3I9G) to predictions from DeepAb (green, 1.8 Å RMSD) and alternative methods (blue, 2.9-3.9 Å RMSD).

### Impact of network architecture on H3 loop modeling accuracy

The model presented in this work includes two primary additions over previous work for predicting H3 loop structures [28]: pretrained LSTM sequence embeddings and criss-cross attention over output branches. To better understand the impact of each of these enhancements, we trained two additional model ensembles following the same procedure as described for the full model. The first model acts as a baseline, without LSTM features or criss-cross attention, and the second introduces the LSTM features. We made predictions for each of the 92 benchmark targets and compared the H3 loop modeling performance of these models to the full model (Figure 3.18A). The baseline model achieved an average H3 loop RMSD of 2.71 Å, outperforming grafting-based methods. Addition of the LSTM features yielded a moderate improvement in

H3 accuracy (0.1 Å RMSD), while addition of criss-cross attention provided a slightly larger improvement (0.2 Å RMSD). We also analyzed the H3 loop lengths of each target while comparing the ablation models (Figure 3.18B) and found that improvements were relatively consistent across lengths.

### 3.3.3 Interpretability of model predictions

Despite the popularity of deep learning approaches for protein structure prediction, little attention has been paid to model interpretability. Interpretable models offer utility beyond their primary predictive task [39, 40]. The network used in this work was designed to be directly interpretable and should be useful for structural understanding and antibody engineering.

#### Output attention tracks model focus

Each output branch in the network includes a criss-cross attention module [41], similar to the axial attention used in other protein applications [23, 42, 43]. We have selected the criss-cross attention in order to efficiently aggregate information over a 2D grid (e.g., pairwise distance and orientation matrices). The criss-cross attention operation allows the network to attend across output rows and columns when predicting for each residue pair (as illustrated in Figure 3.7). Through the attention layer, we create a matrix $\mathbf{A} \in R^{L \times L}$ (where $L$ is the total number of residues in the heavy and light chain $F_V$ domains) containing the total attention between each pair of residues (see experimental procedures). To illustrate the interpretative power of network attention, we considered an anti-peptide antibody (PDB: 4H0H) from the RosettaAntibody

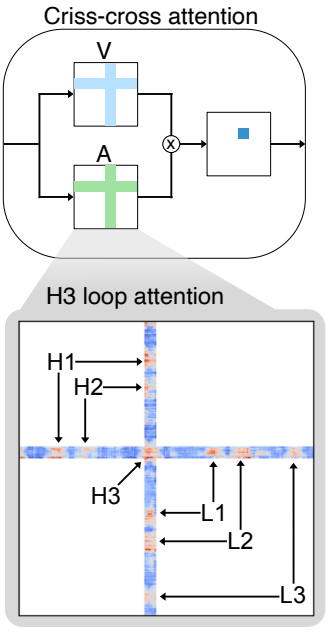


**Figure 3.7:** Criss-cross attention mechanism

Diagram of attention mechanism (with attention matrix A and value matrix V) and example H3 loop attention matrix, with attention on other loops indicated. Attention values increase from blue to red.

benchmark set. Our method performed well on this example (H3 RMSD = 1.2 Å), so we expected it would provide insights into the types of interactions that the network captures well. We collected the attention matrix for $d_{C_\alpha}$ predictions and averaged over the residues belonging to each CDR loop to determine which residues the network focuses on while predicting each loop's structure (Figure 3.8A). As expected, the network primarily attends to residues surrounding each loop of interest. For the CDR1-2 loops, the network attends to the residues in the neighborhood of the loop, with little attention paid to the opposite chain. For the CDR3 loops, the network attends more broadly across the heavy-light chain interface, reflecting the interdependence between the loop conformations and the overall orientation of the chains.

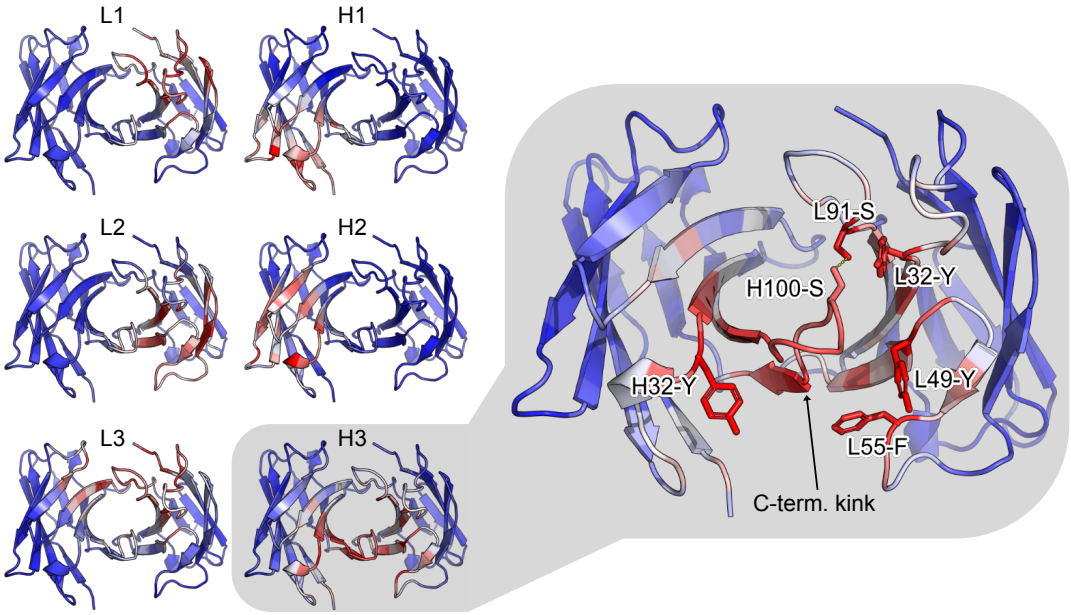


**Figure 3.8:** Attention interpretation for CDR loops

Model attention over Fv structure while predicting each CDR loop for an anti-peptide antibody (PDB: 4H0H). Key interactions identified by attention are shown for predicted CDR H3 loop structure. The top five non-H3 attended residues (H32-Y, L32-Y, L49-Y, L55-F, and L91-S) are labeled, as well as an H3 residue participating in a hydrogen bond (H100-S).

To better understand what types of interactions the network considers, we examined the residues assigned high attention while predicting the H3 loop structure (Figure 3.8B). Within the H3 loop, we found that the highest attention was on the residues forming the C-terminal kink. This structural feature has previously been hypothesized to contribute to H3 loop conformational diversity [44], and it is likely critical for correctly predicting the overall loop structure. Of the five non-H3 residues with the highest attention, we found that one was a phenylalanine and three were tyrosines. The coordination of these bulky side chains appears to play a significant role in the predicted H3 loop conformation. The fifth residue was a serine from the L3 loop (residue

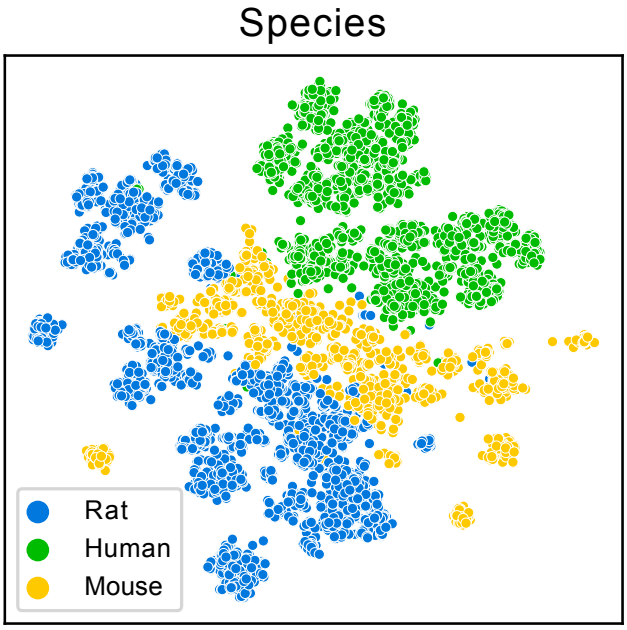


**Figure 3.9:** Sequence embeddings organize by species

Two-dimensional t-SNE projection of sequence-averaged LSTM embeddings labeled by source species.

L91) that forms a hydrogen bond with a serine of the H3 loop (residue H100), suggesting some consideration by the model of biophysical interactions between neighboring residues. To understand how the model attention varies across different H3 loops and neighboring residues, we performed a similar analysis for the 47 targets of the RosettaAntibody benchmark (Figure 3.19). Although some neighboring residues were consistently attended to, we observed noticeable changes in attention patterns across the targets (Figure 3.20), demonstrating the sensitivity of the attention mechanism for identifying key interactions for a broad range of structures.

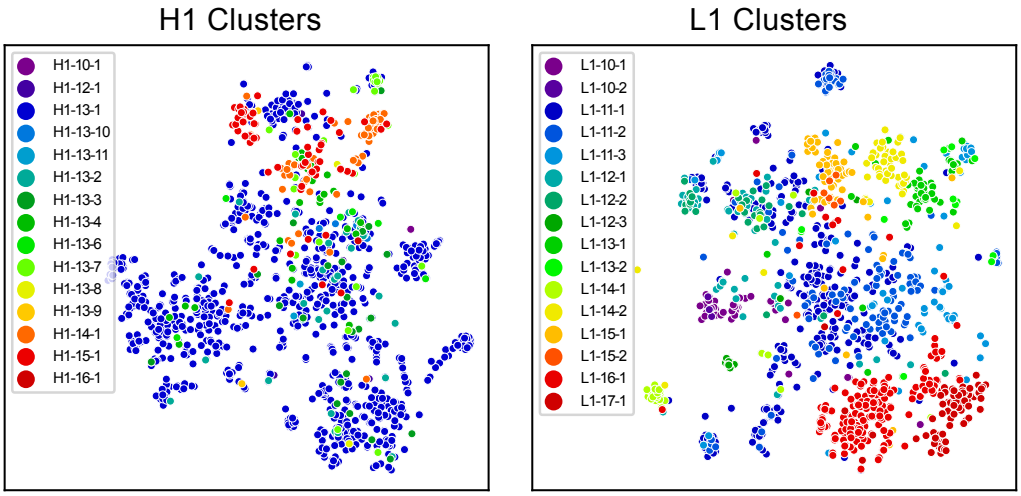


**Figure 3.10:** CDR loop embeddings organize by canonical clusters

Two-dimensional t-SNE projects of LSTM embeddings averaged over CDR1 loop residues labeled by loop structural clusters.

### Repertoire sequence model learns evolutionary and structural representations

To better understand what properties of antibodies are accessible through unsupervised learning, we interrogated the representation learned by the LSTM encoder, which was trained only on sequences. First, we passed the entire set of paired heavy and light chain sequences from the OAS database through the network to generate embeddings like those used for the structure prediction model. The variable-length embedding for each sequence was averaged over its length to generate a fixed-size vector describing the entire sequence. We projected the vector embedding for each sequence into two dimensions via t-distributed stochastic neighbor embedding (t-SNE)[45] and found that the

sequences were naturally clustered by species (Figure 3.9). Because the structural data set is predominately composed of human and murine antibodies, the unsupervised features are likely providing evolutionary context that is otherwise unavailable.

The five non-H3 CDR loops typically adopt one of several canonical conformations [25, 46]. Previous studies have identified distinct structural clusters for these loops and described each cluster by a characteristic sequence signature [47]. We hypothesized that our unsupervised learning model should detect these sequence signatures and thus encode information about the corresponding structural clusters. Similar to before, we created fixed-size embedding vectors for the five non-H3 loops by averaging the whole-sequence embedding over the residues of each loop (according to Chothia definitions [48]). In Figure 3.10, we show t-SNE embeddings for the CDR1 loops labeled by their structural clusters from PyIgClassify [47]. These loops are highlighted because they have the most uniform class balance among structural clusters; similar plots for the remaining loops are provided in Figure 3.21. We observed clustering of labels for both CDR1 loops, indicating that the unsupervised model has captured some structural features of antibodies through sequence alone.

### 3.3.4 Applicability to antibody design

Moving toward the goal of antibody design, we sought to test our method's ability to distinguish between beneficial and disruptive mutations. First, we gathered a previously published deep mutational scanning (DMS) data set for

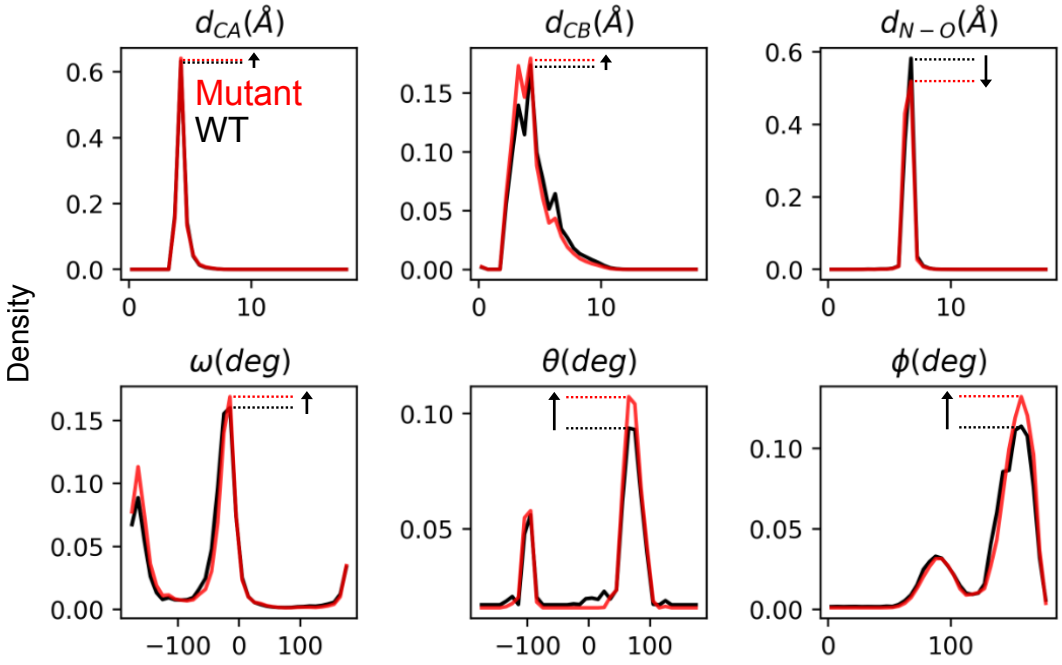


**Figure 3.11:** Visualization of changes in inter-residue potentials upon mutation

Diagram of ΔCCE calculation for model output predictions for an arbitrary residue pair. Plots show the change in probability density of the predicted geometries for the residue pair after making a mutation.

an anti-lysozyme antibody [49]. Anti-lysozyme was an ideal subject for evaluating our network's design capabilities, as it was part of the benchmark set and thus already excluded from training. In the DMS data set, anti-lysozyme was subjected to mutational scanning at 135 positions across the $F_V$, including the CDR loops and the heavy-light chain interface. Each variant was transformed into yeast and measured for binding enrichment over the wild type.

### Prediction confidence is indicative of mutational tolerability

We explored two strategies for evaluating mutations with our network. First, we measured the change in the network's structure prediction confidence for

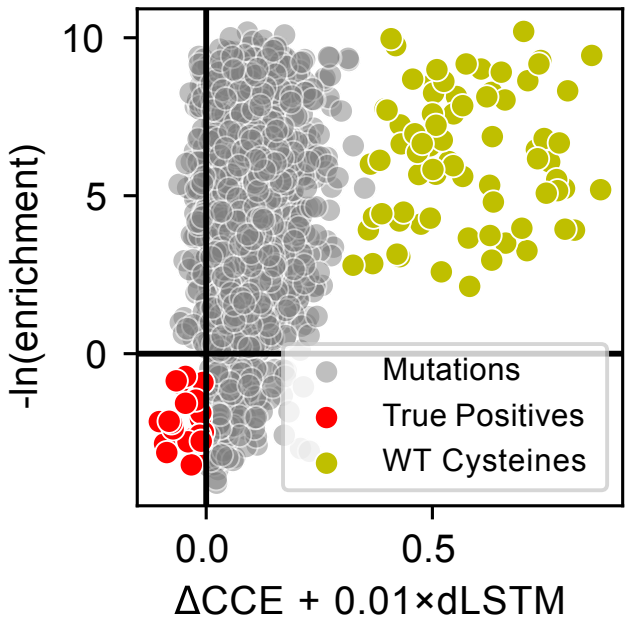


**Figure 3.12:** Comparison of network variant scoring with experimental data

Plot of the combined network metric against experimental binding enrichment over wild type, with negative values corresponding to beneficial mutations for both axes. True positive predictions (red) and mutations to wild type cysteines (yellow) are highlighted.

a variant sequence relative to the wild type (visualized in Figure 3.11) as a change in categorical cross-entropy:

$$\Delta CCE(\text{seq}_{\text{wt}}, \text{seq}_{\text{var}}) = \sum_{ij \in \text{neighbors}} \sum_{g \in \text{outputs}} \log \frac{\max_{g_{ij}} P(g_{ij}|\text{seq}_{\text{wt}})}{\max_{g_{ij}} P(g_{ij}|\text{seq}_{\text{var}})}$$

where $\text{seq}_{\text{wt}}$ and $\text{seq}_{\text{var}}$ are the wild type and variant sequences, respectively, and the conditional probability term describes the probability of a particular geometric output $g_{ij} \in \{d_{C_\alpha,ij}, d_{C_\beta,ij}, d_{N-O,ij}, \omega_{ij}, \theta_{ij}, \phi_{ij}\}$ given $\text{seq}_{\text{wt}}$ or $\text{seq}_{\text{var}}$ . Only residue pairs $ij$ with predicted $d_{C_\alpha} < 10$ Å were used in the calculation. Second, we used the LSTM decoder described previously to calculate the

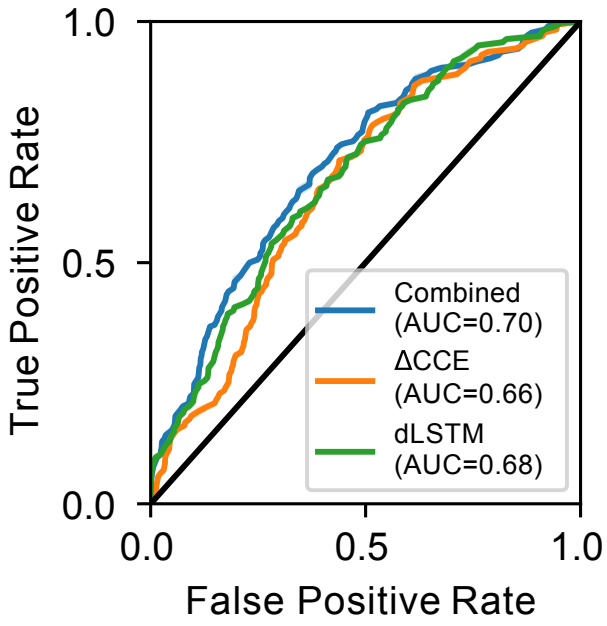


**Figure 3.13:** Classification performance of network variant scoring

Receiver operating characteristic for predicting experimental binding enrichment over wild type with the combined network metric and each component metric. Area under the curve (AUC) values are provided for each metric.

negative log likelihood of a particular point mutation given the wild type sequence, termed dLSTM:

$$\text{dLSTM}(\text{seq}_{\text{var}}|z_{\text{wt}}) = -\log P(\text{seq}_{\text{var, i}} = \text{aa}|z_{\text{wt}}, \text{seq}_{\text{var, i-1}})$$

where $\text{seq}_{\text{var}}$ is a variant sequence with a point mutation to aa at position $i$, and $z_{wt}$ is the biLSTM encoder summary vector for the wild type sequence. To evaluate the discriminative power of the two metrics, we calculated ΔCCE and dLSTM for each variant in the anti-lysozyme data set. We additionally calculated a combined metric as $\Delta CCE + 0.01 \times dLSTM$, roughly equating the magnitudes of both terms, and compared to the experimental binding data

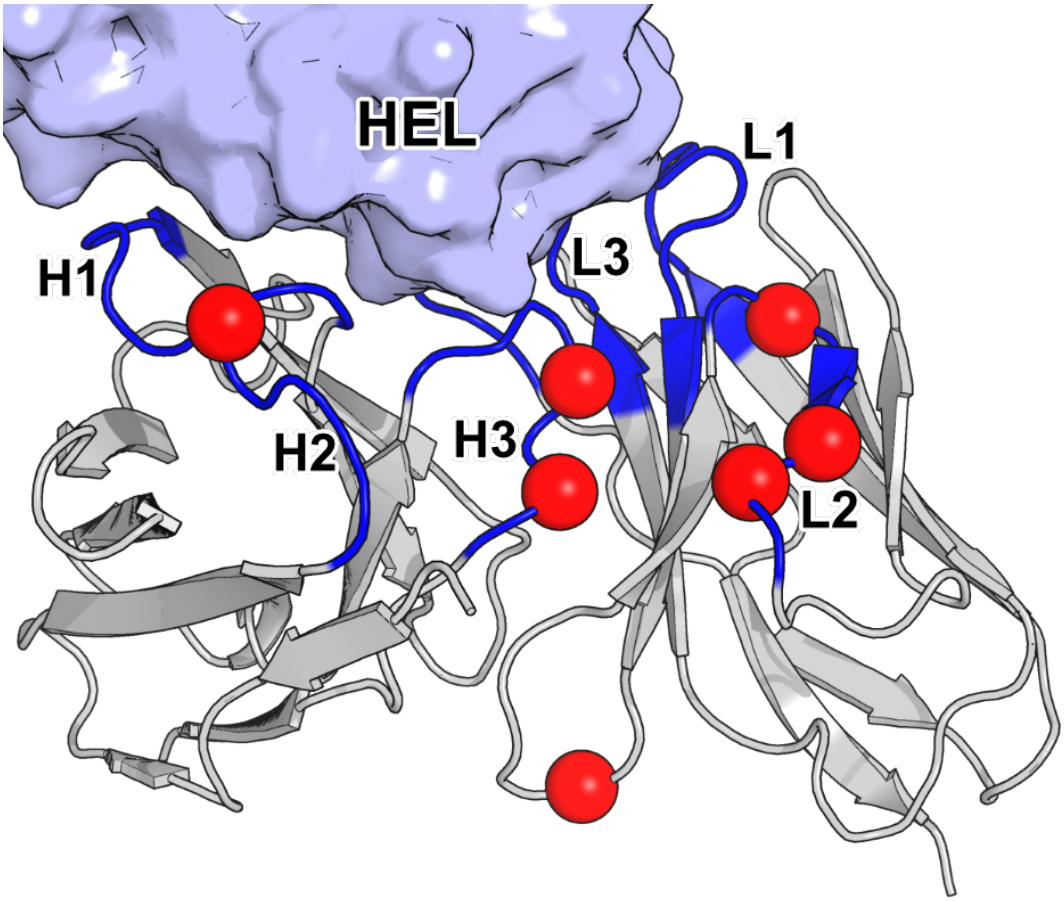


**Figure 3.14:** Position of true positive predictions on anti-lysozyme Fv structure.

(Figure 3.12). Despite having no explicit knowledge of the antigen, the network was moderately predictive of experimental binding enrichment (Figure 3.13). The most successful predictions (true positives in Figure 3.12) were primarily for mutations in CDR loop residues (Figure 3.14). This is not surprising, given that our network has observed the most diversity in these hypervariable regions and is likely less calibrated to variance among framework residues. Nevertheless, if the $\Delta CCE + 0.01 \times dLSTM$ were for ranking, all the top-8 and 22 of the top-100 single-point mutants identified would have experimental binding enrichments above the wild type (Figure 3.15).

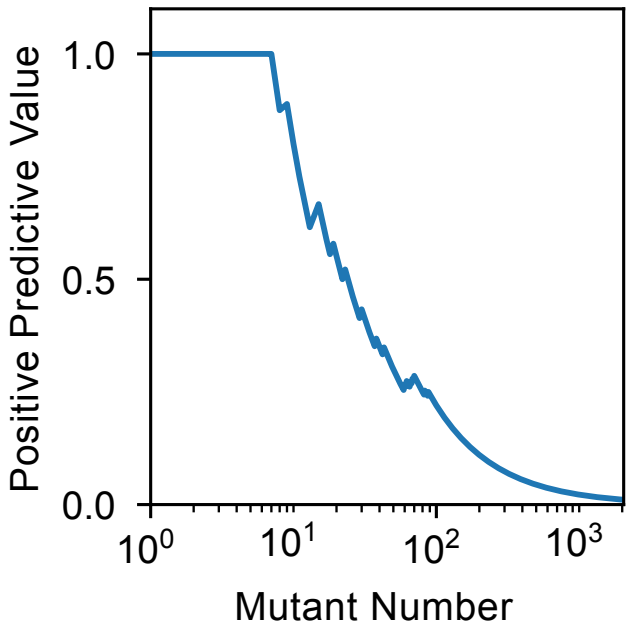


**Figure 3.15:** Positivie predictive value of network variant scoring

Positive predictive value for mutants ranked by the combined metric.

### Network distinguishes stability-enhanced designs

The anti-lysozyme DMS data set was originally assembled to identify residues for design of multi-point variants [49]. The authors designed an anti-lysozyme variant with eight mutations, called D44.1$^{des}$, that displayed improved thermal stability and nearly 10-fold increase in affinity. To determine whether our network could recognize the cumulative benefits of multiple mutations, we created a set of variants with random mutations at the same positions. We calculated ΔCCE for D44.1$^{des}$ and the random variants and found that the model successfully distinguished the design (Figure 3.16). We found similar success at distinguishing enhanced multi-point variants for other targets from

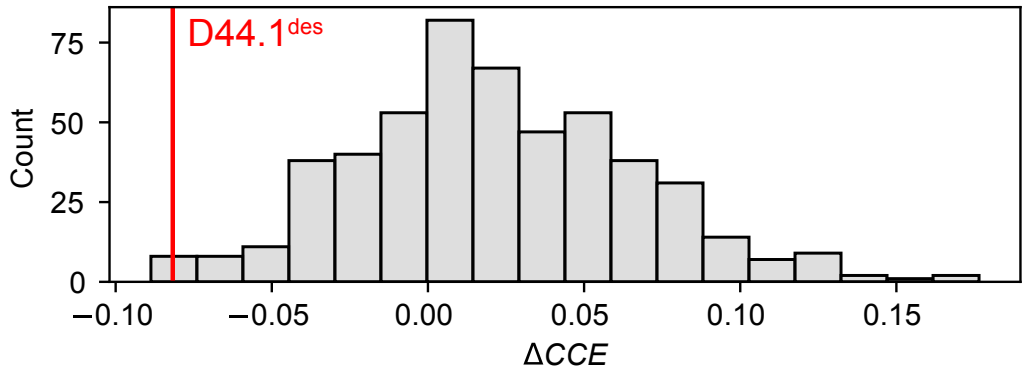


**Figure 3.16:** Identification of previously designed anti-HEL variant

Comparison of ΔCCE for a designed eight-point variant (D44.1$^{des}$, red) to sequences with random mutations at the same positions.

the same publication (Figure 3.22), suggesting that our approach will be a useful screening step for a broad range of antibody design tasks. Despite being trained only for structure prediction, these results suggest that our model may be a useful tool for screening or ranking candidates in antibody design pipelines.

## 3.4 Discussion

The results presented in this work build on advances in general protein structure prediction to effectively predict antibody $F_V$ structures. We found that our deep learning method consistently produced more accurate structures than grafting-based alternatives on benchmarks of challenging, therapeutically relevant targets. Although we focused on prediction of $F_V$ structures, our method is also capable of modeling single-chain nanobodies (Figure 3.23). In these limited cases, the framework RMSD and several of the CDR1 and CDR2

loops are predicted with subAngstrom accuracy. However, we observe that the CDR3 predictions tend to resemble antibody $F_V$ CDR H3 loops, indicating that there may be value in training models specifically for nanobody structure prediction.

As deep learning methods continue to improve, model interpretability will become increasingly important to ensure practitioners can gain insights beyond the primary predictive results. In addition to producing accurate structures, our method also provides interpretable insights into its predictions. Through the attention mechanism, we can track the network's focus while predicting $F_V$ structures. We demonstrated interpretation of predictions for a CDR H3 loop and identified several interactions with neighboring residues that the model deemed important for structure. In the future, similar insights could be used within antibody engineering workflows to prevent disruption of key interactions, reducing the need for time-consuming human analysis and focusing antibody library design.

As part of this work, we developed an unsupervised representation model for antibody sequences. We found that critical features of antibody structure, including non-H3 loop clusters, were accessible through a simple LSTM encoder-decoder model. While we limited training to known pairs of heavy and light chains, several orders of magnitude more unpaired immunoglobins have been identified through next-generation repertoire sequencing experiments [27]. We anticipate that a more advanced language model trained on this larger sequence space will enable further advances across all areas of antibody bioinformatics research.

While this work was under review, improved deep learning methods for general protein structure prediction were published [42, 43]. These methods make extensive use of attention for the end-to-end prediction of protein structures. Both methods additionally separate pairwise residue information from evolutionary information in the form of multiple sequence alignments, with RoseTTAFold going further and learning a nascent structural representation in a third track. While these methods were designed for single-chain predictions, we anticipate that similar methods may yield advances in protein complex prediction (including antibody $F_V$ structures). Further improvements still may come from directly incorporating the antigen into predictions, as antigen binding can lead to significant conformational changes [50]. DMPfold [51], a similar method for general proteins, has been shown to contain flexibility information within inter-residue distance distributions [52]. In principle, DeepAb might provide similar insights into CDR loop flexibility, but further investigation is necessary.

Deep learning models for antibody structure prediction present several promising avenues toward antibody design. In this work, we demonstrated how our network could be used to suggest or screen point mutations. Even with no explicit knowledge of the antigen, this approach was already moderately predictive of mutational tolerability. Further, because our approach relies only on the model outputs for a given sequence, it is capable of screening designs for any antibody. Inclusion of antigen structural context through extended deep learning models or traditional approaches like Rosetta should only improve these results. Other quantities of interest such as stability or

developability metrics could be predicted by using the DeepAb network for transfer learning or feature engineering [12]. Furthermore, comparable networks for general protein structure prediction have recently been re-purposed for design through direct sequence optimization [53, 54, 55]. With minimal modification, our network should enable similar methods for antibody design.

## 3.5 Methods

### 3.5.1 Independent test sets

To evaluate the performance of our method, we considered two independent test sets. The first is the RosettaAntibody benchmark set of 49 structures, which was previously assembled to evaluate methods over a broad range of CDR H3 loop lengths (ranging 7-17 residues) [8, 35]. Each structure in this set has greater than 2.5 Å resolution, a maximum R value of 0.2, and a maximum B factor of 80 $Å^2$. The second comes from a set of 56 clinical-stage antibody therapeutics with solved crystal structures, which was previously assembled to study antibody developability [36]. We removed five of the therapeutic antibodies that were missing one or more CDR loops (PDB: 3B2U, 3C08, 3HMW, 3S34, and 4EDW) to create a therapeutic benchmark set. The two sets shared two common antibodies (PDB: 3EO9 and 3GIZ) that we removed from the therapeutic benchmark set.

While benchmarking alternative methods, we found that some methods were unable to produce structures for every target. Specifically, RosettaAntibody failed to produce predictions for four targets (PDB: 1X9Q, 3IFL, 4D9Q, and 4K3J) and both RepertoireBuilder and ABodyBuilder failed to produce

predictions for two targets (PDB: 4O02 and 5VVK). To compare consistently across all methods, we report values for only the targets that all methods succeeded in modeling. However, we note that DeepAb was capable of producing structures for all of the targets attempted. From the RosettaAntibody benchmark set, we omit PDB: 1X9Q and 3IFL. From the therapeutic benchmark set, we omit PDB: 4D9Q, 4K3J, 4O02, and 5VKK. We additionally omit the long L3 loop of target 3MLR, which not all alternative methods were able to model. In total, metrics are reported for 92 targets: 47 from the RosettaAntibody benchmark and 45 from the therapeutic benchmark. We use the Chothia CDR loop definitions to measure RMSD throughout this work [48].

### 3.5.2 Representation learning on repertoire sequences

#### Training data set

To train the sequence model, paired $F_V$ heavy and light chain sequences were collected from the OAS database [27], a set of immunoglobin sequences from next-generation sequencing experiments of immune repertoires. Each sequence in the database had previously been parsed with ANARCI [56] to annotate sequences and detect potentially erroneous entries. For this work, we extract only the $F_V$ region of the sequences, as identified by ANARCI. Sequences indicated to have failed ANARCI parsing were discarded from the training data set. We additionally remove any redundant sequences. These steps resulted in a set of 118,386 sequences from five studies [57, 58, 59, 60, 61] for model training.

### Model and training details

To learn representations of immunoglobin sequences, we adopted an RNN encoder-decoder model [25] consisting of two LSTMs [26]. In an encoder-decoder model, the encoder learns to summarize the input sequence into a fixed-dimension summary vector, from which the decoder learns to reconstruct the original sequence. For the encoder model, we used a bidirectional twolayer stacked LSTM with a hidden state size of 64. The model input was created by concatenation of paired heavy and light chain sequences to form a single sequence. Three additional tokens were added to the sequence to mark the beginning of the heavy chain, the end of the heavy chain, and the end of the light chain. The concatenated sequence was one-hot encoded, resulting in an input of dimension $(L + 3) \times 23$, where $L$ is the combined heavy and light chain length. The summary vector is generated by stacking the final hidden states from the forward and backward encoder LSTMs, followed by a linear transformation from 128 to 64 dimensions and *tanh* activation. For the decoder model, we used a two-layer stacked LSTM with a hidden state size of 64. The decoder takes as input the summary vector and the previously decoded amino acid to sequentially predict the original amino acid sequence.

The model was trained using cross-entropy loss and the Adam optimizer [62] with a learning rate of 0.01, with learning rate reduced upon plateauing of validation loss. A teacher forcing rate of 0.5 was used to stabilize training. The model was trained on one NVIDIA K80 GPU, requiring 4 hours for 5 epochs over the entire data set. We used a batch size of 128, maximized to fit into GPU memory.

### 3.5.3 Predicting inter-residue geometries from antibody sequence

#### Training data set

To train the structure prediction model, we collected a set of $F_V$ structures from the SAbDab [63], a curated set of antibody structures from the PDB [64]. We removed structures with less than 4 Å resolution and applied a 99% sequence identity threshold to remove redundant sequences. We chose this high sequence similarity due to the high conservation characteristic of antibody sequences, as well as the over-representation of many identical therapeutic antibodies in structural databases. Additionally, we hoped to expose the model to examples of small sequence variations that lead to differences in structures. This is particularly important for the challenging CDR H3 loop, which has been observed to occupy an immense diversity of conformations even at the level of four-level fragments [65]. Finally, any targets from the benchmark sets, or structures with 99% sequence similarity to a target, were removed from the training data set. These steps resulted in a set of 1,692 $F_V$ structures, a mixture of antigen bound and unbound, for model training.

#### Model and training details

The structure prediction model takes as input concatenated heavy and light chain sequences. The sequences are one-hot encoded and passed through two parallel branches: a 1D ResNet and the biLSTM encoder described above. For the 1D ResNet, we add an additional delimiter channel to mark the end of the heavy chain, resulting in a dimension of $L \times 21$, where $L$ is the combined

heavy and light chain length. The 1D ResNet begins with a 1D convolution that projects the input features up to dimension $L \times 64$, followed by three 1D ResNet blocks (two 1D convolutions with kernel size 17) that maintain dimensionality. The second branch consists of the pretrained biLSTM encoder. Before passing the one-hot encoded sequence to the biLSTM, we add the three delimiters described previously, resulting in dimension $(L+3) \times 23$. From the biLSTM, we concatenate the hidden states from the forward and backward LSTMs after encoding each residue, resulting in dimension $L \times 128$. The outputs of the 1D ResNet and the biLSTM are stacked to form a final sequential tensor of dimension $L \times 160$. We transform the sequential tensor to pairwise data by concatenating row- and column-wise expansions. The pairwise data, dimension $L \times L \times 320$, is passed to the 2D ResNet. The 2D ResNet begins with a 2D convolution that reduces dimensionality to $L \times L \times 64$, followed by 25 2D ResNet blocks (two 2D convolutions with kernel size $5 \times 5$) that maintain dimensionality. The 2D ResNet blocks cycle through convolution dilation values of 1, 2, 4, 8, and 16 (five cycles in total). After the 2D ResNet, the network branches into six separate paths. Each output branch consists of a 2D convolution that projects down to dimension $L \times L \times 37$, followed by a recurrent criss-cross attention (RCCA) module [41]. The RCCA modules use two criss-cross attention operations that share weights, allowing each residue pair to gather information across the entire spatial dimension. Attention queries and keys are projected to dimension $L \times L \times 1$ (one attention head). Symmetry is enforced for $d_{C_\alpha}$, $d_{C_\beta}$, and $\omega$ predictions by averaging the final outputs with their transposes. All convolutions in the network are followed by ReLU activation. In total, the model contains about 6.4 million trainable

parameters.

We trained five models on random 90/10% training/validation splits and averaged over model logits to make predictions, following previous methods [19]. Models were trained using focal loss [29] and the Adam optimizer [62] with a learning rate of 0.01, with learning rate reduced upon plateauing of validation loss. Learning rate was reduced upon plateauing of the validation loss. Each model was trained on one NVIDIA K80 GPU, requiring 60 hours for 60 epochs over the entire data set.

### 3.5.4 Structure realization

#### Multi-dimensional scaling

From the network predictions, we create real-value matrices for the $d_{C_\beta}$, $\omega$, $\theta$, and $\phi$ outputs by taking the midpoint value of the modal probability bin for each residue pair. From these real-valued distances and orientations, we create an initial backbone atom ($N$, $C_\alpha$, and $C$) distance matrix. For residue pairs predicted to have $d_{C_\beta} > 18$ Å, we approximate the distances between atoms using the Floyd-Warshall shortest path algorithm [66]. From this distance matrix, we use MDS [67] to produce an initial set of 3D coordinates. The initial structures from MDS typically contained atom clashes and non-ideal geometries that required further refinement.

#### Energy minimization refinement

Initial structures from MDS were refined by constrained energy minimization in Rosetta. For each pair of residues, the predicted distributions for each

output were converted to energy potentials by negating the raw model logits (i.e., without softmax activation) and dividing by the squared $d_{C_\alpha}$ prediction.

The discrete potentials were converted to continuous functions using the built-in Rosetta spline function. We disregarded potentials for residue pairs with predicted $d_{C_\alpha} > 18$ Å, as well as those with a modal bin probability below 10%. For $d_{N-O}$ potentials, we also discarded with predicted $d_{N-O} > 5$ Å or modal bin probability below 30% to create a local backbone hydrogen-bonding potential. The remaining potentials are applied to the MDS structure as inter-residue constraints in Rosetta.

Modeling in Rosetta begins with a coarse-grained representation, in which the side-chain atoms are represented as a single artificial atom (centroid). The centroid model is optimized by gradient-based energy minimization (*MinMover*) using the L-BFGS algorithm [32, 33]. The centroid energy function includes the following score terms in addition to learned constraints: vdw (clashes), cen_hb (hydrogen bonds), and rama and omega (backbone torsion angles). After centroid optimization, we add side-chain atoms and relax the structure to reduce steric clashes (*FastRelax*). Finally, we repeat the gradient-based energy minimization step in the full-atom representation to produce a final model. We repeat this procedure to produce 50 decoy models and select the structure with the lowest energy as the final prediction. Only the relaxation step in the protocol is non-deterministic, leading to high convergence among decoys. In practice, we expect 5-10 decoys will be sufficient for most applications.

### 3.5.5 Predicting structures with other recent methods

To contextualize the performance of our method, we benchmarked three recent methods for antibody $F_V$ structure prediction: RosettaAntibody-G [6], RepertoireBuilder [5], and ABodyBuilder [3]. RosettaAntibody-G predictions were generated using the command-line arguments recommended by Jeliazkov et al. (Appendix S1). We note that we only used the RosettaAntibody grafting protocol (*antibody*), omitting the extensive but time-consuming H3 loop sampling ($antibody_{H3}$) [4, 6]. RepertoireBuilder and ABodyBuilder predictions were generated using their respective web servers. For each target in the benchmarks, we excluded structures with sequence similarity greater than 99% from use for predictions, to mirror the conditions of our training set. We note that this sequence cutoff does not prevent methods from grafting identical loops from slightly different sequences.

### 3.5.6 Attention matrix calculation

During the criss-cross attention operation [41], we create an attention matrix $\mathbf{A} \in R^{L\times L\times(2L-1)}$, where for each residue pair in the $L \times L$ spatial dimension, we have $2L-1$ entries corresponding to the attention values over other residue pairs in the same row and column (including the residue pair itself). To interpret the total attention between pairs of residues, we simplify the attention matrix to $\mathbf{A}' \in R^{L\times(2L-1)}$, where for each residue $i$ in the sequence, we only consider the attention values in the $i$-th row and column. In $\mathbf{A}'$, for each residue $i$ there are two attention values for each other residue $j$, corresponding to the rowand column-wise attention between $i$ and $j$. We further

simplify by summing these rowand column-wise attention values, resulting in an attention matrix $\mathbf{A}'' \in R^{L \times L}$, containing the total attention between pairs of residues. In the main text, we refer to $\mathbf{A}''$ as $\mathbf{A}$ for simplicity.

## 3.6 Appendix

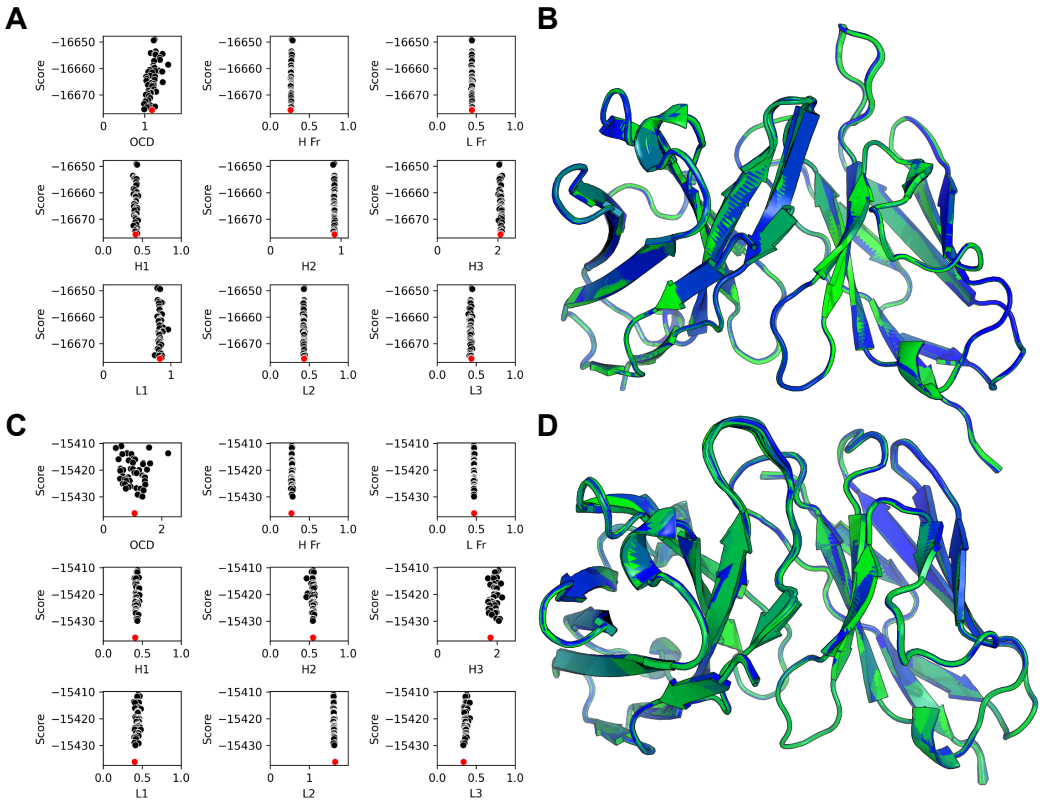


**Figure 3.17:** Convergence of predicted structures for two benchmark examples

(A) Funnel plots showing accuracy (OCD, RMSD) versus score for 50 DeepAb decoys for target 3PP3 (therapeutic benchmark), with low-scoring structure in red. (B) Superimposed decoy structures for target 3PP3. (C) Funnel plots showing accuracy (OCD, RMSD) versus score for 50 DeepAb decoys for target 3I9G (RosettaAntibody benchmark), with low-scoring structure in red. (D) Superimposed decoy structures for target 3I9G.

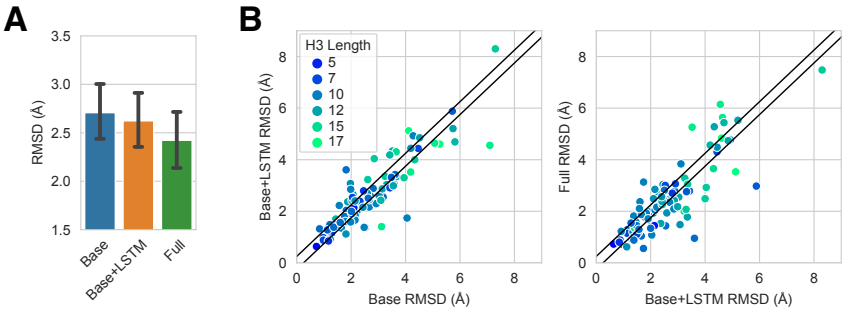


**Figure 3.18:** Impact of architecture additions on H3 loop accuracy

(A) Average RMSD of H3 loops predicted by baseline model (without LSTM features or CCA), baseline model with LSTM features, and full model. Error bars show standard deviations for each model on each benchmark. (B) Direct comparison of H3 RMSD for each target as architecture is expanded, with diagonal bands indicating predictions that were within $\pm$ 0.25 Å. Point color indicates H3 loop length for each target.

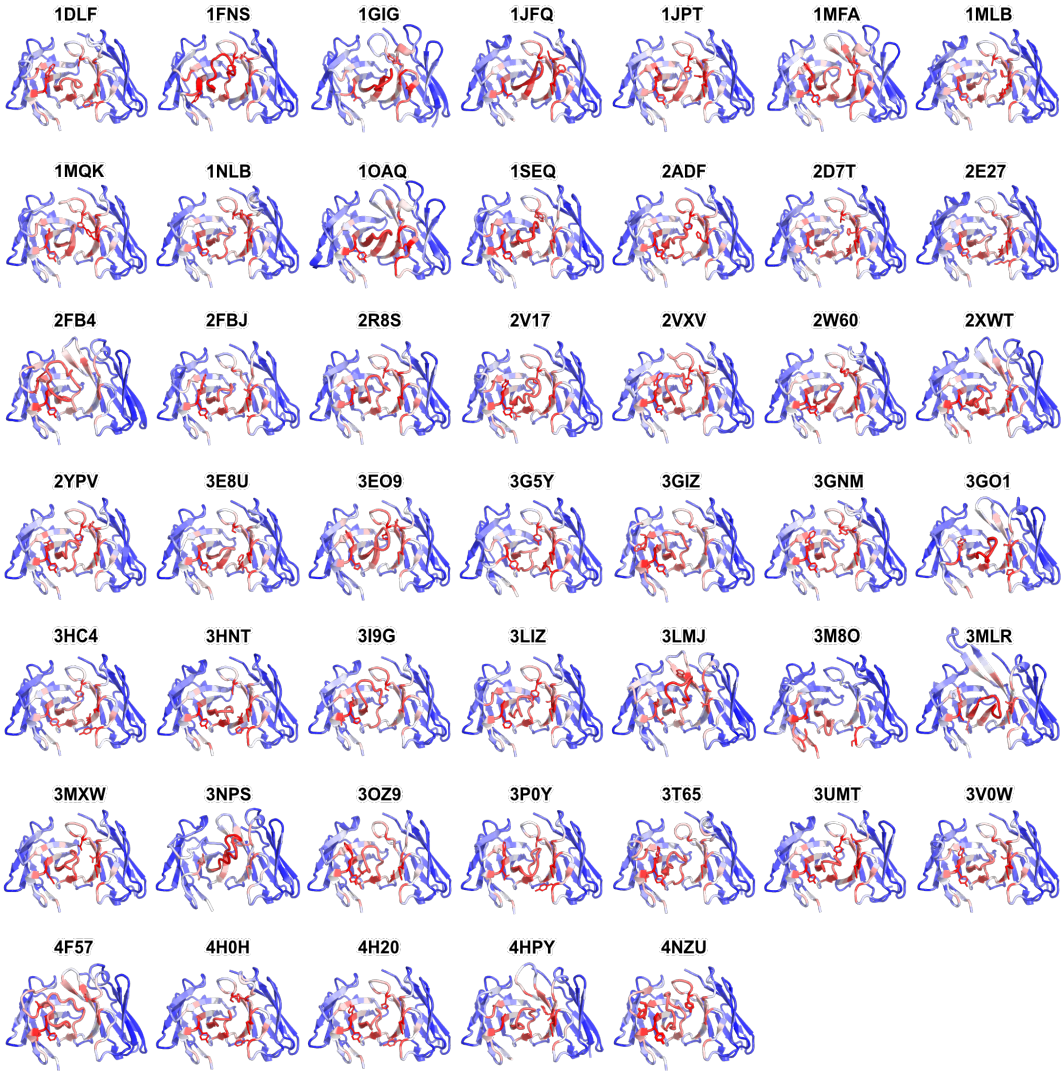


**Figure 3.19:** H3 loop attention for RosettaAntibody benchmark targets

Model $C_\alpha$ attention while predicting H3 loop structures for each of the 47 targets in the RosettaAntibody benchmark. Attention values increase from blue to red. For each target, the side chains of the five most attended non-H3 residues are represented as sticks.

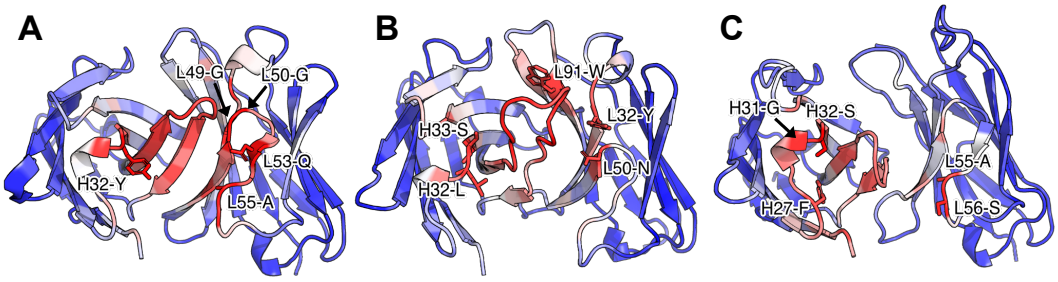


**Figure 3.20:** Variability of key residues identified by attention mechanism

Model $C_\alpha$ attention while predicting H3 loop structures for three targets in the RosettaAntibody benchmark. Attention values increase from blue to red. For each target, the side chains of the five most attended non-H3 residues are represented as sticks. (A) H3 attention for 1OAQ prediction. (B) H3 attention for 3MLJ prediction. (C) H3 attention for 3M8O prediction.

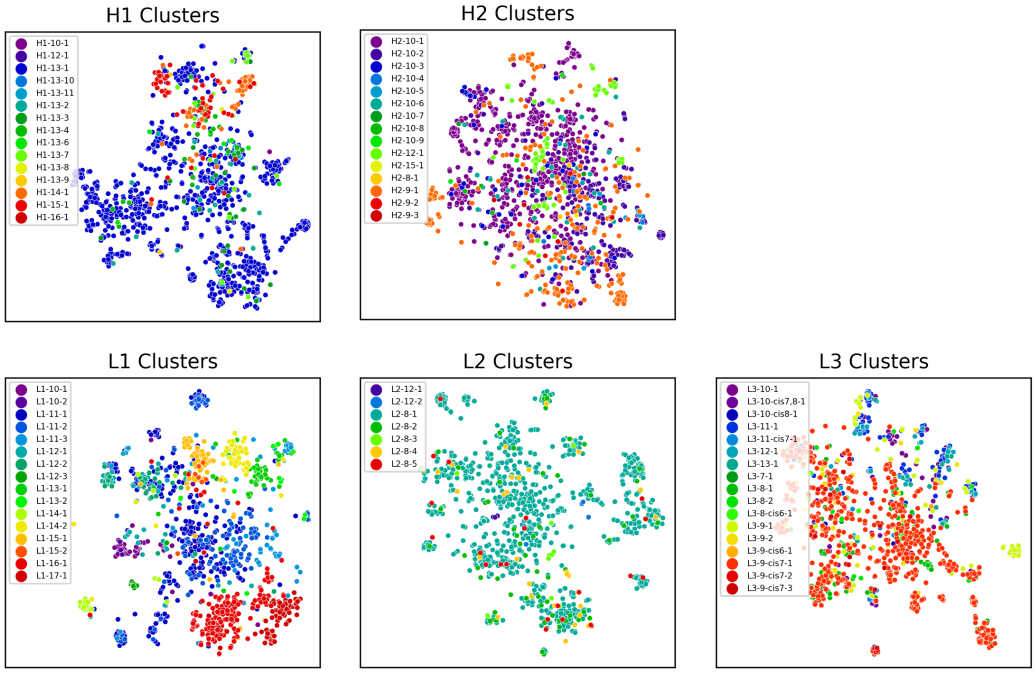


**Figure 3.21:** Non-H3 CDR loop t-SNE embeddings labeled by structural clusters

CDR-specific embeddings are created by averaging the bi-LSTM encoder hidden states of residues for each CDR loop.

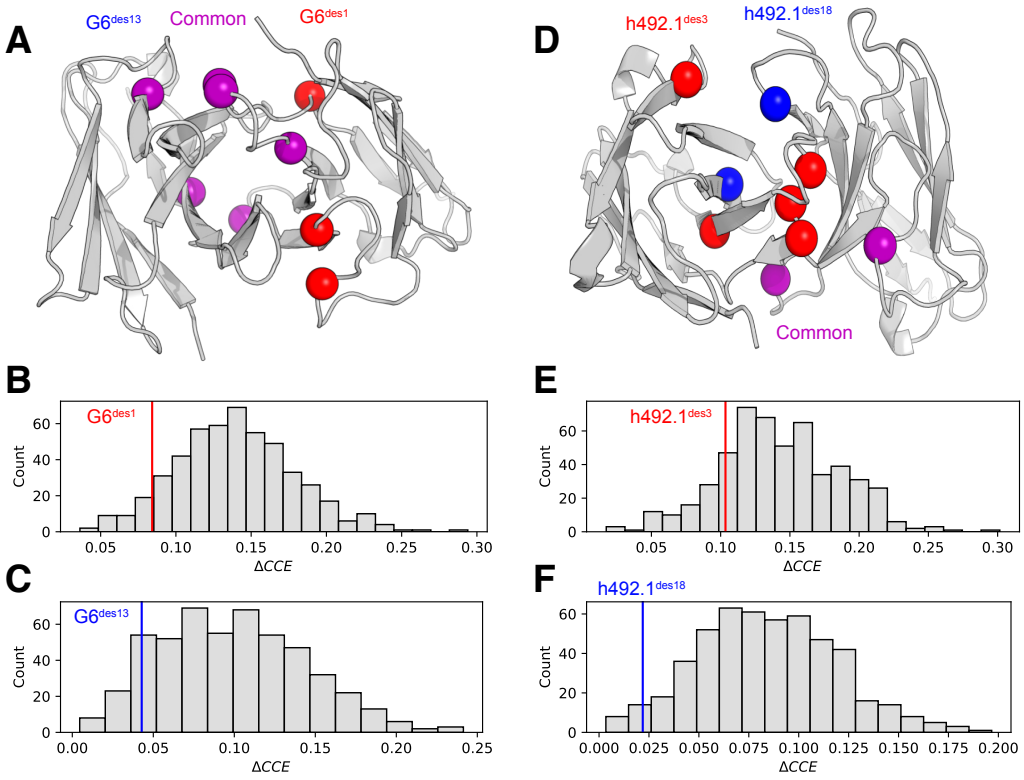


**Figure 3.22:** Identification of stable multi-point variants for two AbLIFT designs

Both wild type structures were present in the training dataset, resulting in a slight bias for the native sequence. (A) Mutation positions for two anti-VEGF multi-point variants presented by Warszawski et al. (B) Comparison of ΔCCE values for G6$^{des1}$ (nine-point variant) and random nine-point variants at the same positions. (C) Comparison of ΔCCE values for G6$^{des13}$ (six-point variant) and random six-point variants at the same positions. (D) Mutation positions for two anti-QSOX1 multi-point variants. (E) Comparison of ΔCCE values for h492.1$^{des3}$ (seven-point variant) and random seven-point variants at the same positions. (F) Comparison of ΔCCE values for h492.1$^{des18}$ (four-point variant) and random four-point variants at the same positions.

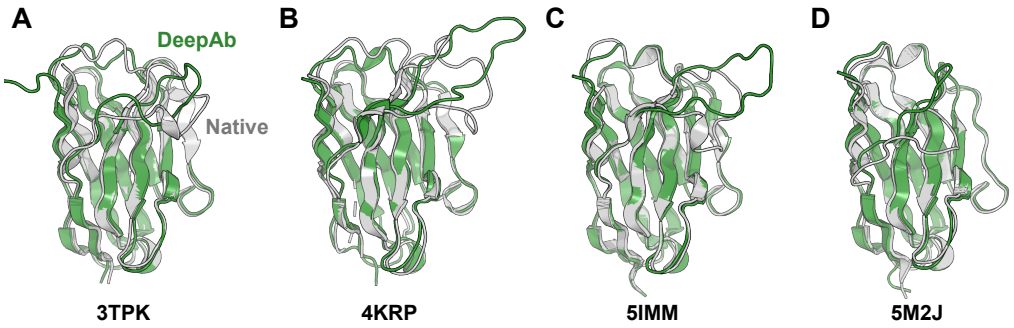


**Figure 3.23:** Nanobody structures predicted by DeepAb

Four nanobody structures predicted by DeepAb (green) aligned to native structures (gray). Prediction accuracy is reported as RMSDs over the framework region and the three CDR loops. (A) Predicted structure for nanobody 3TPK (framework: 0.58 Å, CDR1: 3.29 Å, CDR2: 1.07 Å, CDR3: 4.73 Å). (B) Predicted structure for nanobody 4KRP (framework: 0.82 Å, CDR1: 2.36 Å, CDR2: 2.07 Å, CDR3: 5.56 Å). (C) Predicted structure for nanobody 5IMM (framework: 0.46 Å, CDR1: 1.89 Å, CDR2: 0.58 Å, CDR3: 7.60 Å). (D) Predicted structure for nanobody 5M2J (framework: 1.01 Å, CDR1: 1.12 Å, CDR2: 0.86 Å, CDR3: 8.34 Å)

# References


[1] Ruei-Min Lu, Yu-Chyi Hwang, I-Ju Liu, Chi-Chiu Lee, Han-Zen Tsai, Hsin-Jung Li, and Han-Chung Wu. "Development of therapeutic antibodies for the treatment of diseases". In: *Journal of Biomedical Science* 27.1 (2020), pp. 1–30.

[2] Hélène Kaplon and Janice M Reichert. "Antibodies to watch in 2021". In: *MAbs*. Vol. 13. 1. Taylor & Francis. 2021, p. 1860476.

[3] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Claire Marks, Jaroslaw Nowak, Cristian Regep, Guy Georges, Sebastian Kelm, Bojana Popovic, and Charlotte M Deane. "SAbPred: a structure-based antibody prediction server". In: *Nucleic Acids Research* 44.W1 (2016), W474–W478.

[4] Brian D Weitzner, Jeliazko R Jeliazkov, Sergey Lyskov, Nicholas Marze, Daisuke Kuroda, Rahel Frick, Jared Adolf-Bryfogle, Naireeta Biswas, Roland L Dunbrack Jr, and Jeffrey J Gray. "Modeling and docking of antibody structures with Rosetta". In: *Nature Protocols* 12.2 (2017), pp. 401–416.

[5] Dimitri Schritt, Songling Li, John Rozewicki, Kazutaka Katoh, Kazuo Yamashita, Wayne Volkmuth, Guy Cavet, and Daron M Standley. "Repertoire Builder: high-throughput structural modeling of B and T cell receptors". In: *Molecular Systems Design & Engineering* 4.4 (2019), pp. 761–768.

[6] Jeliazko R Jeliazkov, Rahel Frick, Jing Zhou, and Jeffrey J Gray. "Robustification of rosettaantibody and rosetta snugdock". In: *PLOS One* 16.3 (2021), e0234282.

[7] James Dunbar, Angelika Fuchs, Jiye Shi, and Charlotte M Deane. "ABangle: characterising the VH–VL orientation in antibodies". In: *Protein Engineering, Design and Selection* 26.10 (2013), pp. 611–620.

[8] Nicholas A Marze, Sergey Lyskov, and Jeffrey J Gray. "Improved prediction of antibody VL–VH orientation". In: *Protein Engineering, Design and Selection* 29.10 (2016), pp. 409–418.

[9] Juan C Almagro, Alexey Teplyakov, Jinquan Luo, Raymond W Sweet, Sreekumar Kodangattil, Francisco Hernandez-Guzman, and Gary L Gilliland. *Second Antibody Modeling Assessment (AMA-II)*. 2014.

[10] Wenhao Gao, Sai Pooja Mahajan, Jeremias Sulam, and Jeffrey J Gray. "Deep learning in protein structural modeling and design". In: *Patterns* 1.9 (2020), p. 100142.

[11] Jordan Graves, Jacob Byerly, Eduardo Priego, Naren Makkapati, S Vince Parish, Brenda Medellin, and Monica Berrondo. "A review of deep learning methods for antibodies". In: *Antibodies* 9.2 (2020), p. 12.

[12] Xingyao Chen, Thomas Dougherty, Chan Hong, Rachel Schibler, Yi Cong Zhao, Reza Sadeghi, Naim Matasci, Yi-Chieh Wu, and Ian Kerman. "Predicting antibody developability from sequence using machine learning". In: *bioRxiv* (2020), pp. 2020–06.

[13] Claire Marks, Alissa M Hummer, Mark Chin, and Charlotte M Deane. "Humanization of antibodies using a machine learning approach on large-scale repertoire data". In: *Bioinformatics* 37.22 (2021), pp. 4041–4047.

[14] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. "Protein design and variant prediction using autoregressive generative models". In: *Nature Communications* 12.1 (2021), p. 2403.

[15] Srivamshi Pittala and Chris Bailey-Kellogg. "Learning context-aware structural representations to predict antigen and antibody binding interfaces". In: *Bioinformatics* 36.13 (2020), pp. 3996–4003.

[16] Rahmad Akbar, Philippe A Robert, Milena Pavlović, Jeliazko R Jeliazkov, Igor Snapkov, Andrei Slabodkin, Cédric R Weber, Lonneke Scheffer, Enkelejda Miho, Ingrid Hobæk Haff, et al. "A compact vocabulary of paratope-epitope interactions enables predictability of antibody-antigen binding". In: *Cell Reports* 34.11 (2021), p. 108856.

[17] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792 (2020), pp. 706–710.

[18] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. "Improved protein structure prediction using predicted interresidue orientations". In: *Proceedings of the National Academy of Sciences* 117.3 (2020), pp. 1496–1503.

[19] Jinbo Xu, Matthew Mcpartlon, and Jin Li. "Improved protein structure prediction by deep learning irrespective of co-evolution information". In: *Nature Machine Intelligence* 3.7 (2021), pp. 601–609.

[20] George Georgiou, Gregory C Ippolito, John Beausang, Christian E Busse, Hedda Wardemann, and Stephen R Quake. "The promise and challenge of high-throughput sequencing of the antibody repertoire". In: *Nature Biotechnology* 32.2 (2014), pp. 158–168.

[21] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118.

[22] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. "Transformer protein language models are unsupervised structure learners". In: *bioRxiv* (2020), pp. 2020–12.

[23] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. "MSA transformer". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8844–8856.

[24] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. "Unified rational protein engineering with sequence-based deep representation learning". In: *Nature Methods* 16.12 (2019), pp. 1315–1322.

[25] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[26] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM". In: *Neural Computation* 12.10 (2000), pp. 2451–2471.

[27] Aleksandr Kovaltsuk, Jinwoo Leem, Sebastian Kelm, James Snowden, Charlotte M Deane, and Konrad Krawczyk. "Observed Antibody Space: a resource for data mining next-generation sequencing of antibody repertoires". In: *The Journal of Immunology* 201.8 (2018), pp. 2502–2509.

[28] Jeffrey A Ruffolo, Carlos Guerra, Sai Pooja Mahajan, Jeremias Sulam, and Jeffrey J Gray. "Geometric potentials from deep learning improve prediction of CDR H3 loop structures". In: *Bioinformatics* 36.Supplement_1 (2020), pp. i268–i275.

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[30] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. "Calibrating deep neural networks using focal loss". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15288–15299.

[31] Jingfen Zhang, Qingguo Wang, Bogdan Barz, Zhiquan He, Ioan Kosztin, Yi Shang, and Dong Xu. "MUFOLD: A new solution for protein 3D structure prediction". In: *Proteins: Structure, Function, and Bioinformatics* 78.5 (2010), pp. 1137–1152.

[32] Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian W Kaufman, P Douglas Renfrew, Colin A Smith, Will Sheffler, et al. "ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules". In: *Methods in Enzymology*. Vol. 487. Elsevier, 2011, pp. 545–574.

[33] Julia Koehler Leman, Brian D Weitzner, Steven M Lewis, Jared Adolf-Bryfogle, Nawsad Alam, Rebecca F Alford, Melanie Aprahamian, David Baker, Kyle A Barlow, Patrick Barth, et al. "Macromolecular modeling and design in Rosetta: recent methods and frameworks". In: *Nature Methods* 17.7 (2020), pp. 665–680.

[34] Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. "The Rosetta

all-atom energy function for macromolecular modeling and design". In: *Journal of Chemical Theory and Computation* 13.6 (2017), pp. 3031–3048.

[35] Brian D Weitzner and Jeffrey J Gray. "Accurate structure prediction of CDR H3 loops enabled by a novel structure-based C-terminal constraint". In: *The Journal of Immunology* 198.1 (2017), pp. 505–515.

[36] Matthew IJ Raybould, Claire Marks, Konrad Krawczyk, Bruck Taddese, Jaroslaw Nowak, Alan P Lewis, Alexander Bujotzek, Jiye Shi, and Charlotte M Deane. "Five computational developability guidelines for therapeutic antibody profiling". In: *Proceedings of the National Academy of Sciences* 116.10 (2019), pp. 4025–4030.

[37] Gerhard Niederfellner, Alfred Lammens, Olaf Mundigl, Guy J Georges, Wolfgang Schaefer, Manfred Schwaiger, Andreas Franke, Kornelius Wiechmann, Stefan Jenewein, Jerry W Slootstra, et al. "Epitope characterization and crystal structure of GA101 provide insights into the molecular basis for type I/II distinction of CD20 antibodies". In: *Blood, The Journal of the American Society of Hematology* 118.2 (2011), pp. 358–367.

[38] Jonathan M Wojciak, Norman Zhu, Karen T Schuerenberg, Kelli Moreno, William S Shestowsky, Masao Hiraiwa, Roger Sabbadini, and Tom Huxford. "The crystal structure of sphingosine-1-phosphate in complex with a Fab fragment reveals metal bridging of an antibody and its antigen". In: *Proceedings of the National Academy of Sciences* 106.42 (2009), pp. 17717–17722.

[39] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017).

[40] Zachary C Lipton. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57.

[41] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. "Ccnet: Criss-cross attention for semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 603–612.

[42] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. "Accurate prediction of protein structures and interactions using a three-track neural network". In: *Science* 373.6557 (2021), pp. 871–876.

[43] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[44] Brian D Weitzner, Roland L Dunbrack, and Jeffrey J Gray. "The origin of CDR H3 structural diversity". In: *Structure* 23.2 (2015), pp. 302–311.

[45] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of Machine Learning Research* 9.11 (2008).

[46] Benjamin North, Andreas Lehmann, and Roland L Dunbrack Jr. "A new clustering of antibody CDR loop conformations". In: *Journal of Molecular Biology* 406.2 (2011), pp. 228–256.

[47] Jared Adolf-Bryfogle, Qifang Xu, Benjamin North, Andreas Lehmann, and Roland L Dunbrack Jr. "PyIgClassify: a database of antibody CDR structural classifications". In: *Nucleic Acids Research* 43.D1 (2015), pp. D432–D438.

[48] Cyrus Chothia and Arthur M Lesk. "Canonical structures for the hypervariable regions of immunoglobulins". In: *Journal of Molecular Biology* 196.4 (1987), pp. 901–917.

[49] Shira Warszawski, Aliza Borenstein Katz, Rosalie Lipsh, Lev Khmelnitsky, Gili Ben Nissan, Gabriel Javitt, Orly Dym, Tamar Unger, Orli Knop, Shira Albeck, et al. "Optimizing antibody affinity and stability by the automated design of the variable light-heavy chain interfaces". In: *PLOS Computational Biology* 15.8 (2019), e1007207.

[50] Monica L Fernández-Quintero, Johannes Kraml, Guy Georges, and Klaus R Liedl. "CDR-H3 loop ensemble in solution–conformational selection upon antibody binding". In: *MAbs*. Vol. 11. 6. Taylor & Francis. 2019, pp. 1077–1088.

[51] Joe G Greener, Shaun M Kandathil, and David T Jones. "Deep learning extends de novo protein modelling coverage of genomes using iteratively predicted structural constraints". In: *Nature Communications* 10.1 (2019), p. 3977.

[52] Dominik Schwarz, Guy Georges, Sebastian Kelm, Jiye Shi, Anna Vangone, and Charlotte M Deane. "Co-evolutionary distance predictions contain flexibility information". In: *Bioinformatics* 38.1 (2022), pp. 65–72.

[53] Johannes Linder and Georg Seelig. "Fast differentiable DNA and protein sequence optimization for molecular design". In: *arXiv preprint arXiv:2005.11275* (2020).

[54] Ivan Anishchenko, Samuel J Pellock, Tamuka M Chidyausiku, Theresa A Ramelot, Sergey Ovchinnikov, Jingzhou Hao, Khushboo Bafna, Christoffer Norn, Alex Kang, Asim K Bera, et al. "De novo protein design by deep network hallucination". In: *Nature* 600.7889 (2021), pp. 547–552.

[55] Christoffer Norn, Basile IM Wicky, David Juergens, Sirui Liu, David Kim, Doug Tischer, Brian Koepnick, Ivan Anishchenko, Foldit Players, David Baker, et al. "Protein sequence design by conformational landscape optimization". In: *Proceedings of the National Academy of Sciences* 118.11 (2021), e2017228118.

[56] James Dunbar and Charlotte M Deane. "ANARCI: antigen receptor numbering and receptor classification". In: *Bioinformatics* 32.2 (2016), pp. 298–300.

[57] Leonard D Goldstein, Ying-Jiun J Chen, Jia Wu, Subhra Chaudhuri, Yi-Chun Hsiao, Kellen Schneider, Kam Hon Hoi, Zhonghua Lin, Steve Guerrero, Bijay S Jaiswal, et al. "Massively parallel single-cell B-cell receptor sequencing enables rapid discovery of diverse antigen-reactive antibodies". In: *Communications Biology* 2.1 (2019), p. 304.

[58] Ian Setliff, Andrea R Shiakolas, Kelsey A Pilewski, Amyn A Murji, Rutendo E Mapengo, Katarzyna Janowska, Simone Richardson, Charissa Oosthuysen, Nagarajan Raju, Larance Ronsard, et al. "High-throughput mapping of B cell receptor sequences to antigen specificity". In: *Cell* 179.7 (2019), pp. 1636–1646.

[59] Jacob D Eccles, Ronald B Turner, Nicole A Kirk, Lyndsey M Muehling, Larry Borish, John W Steinke, Spencer C Payne, Paul W Wright, Deborah Thacker, Sampo J Lahtinen, et al. "T-bet+ memory B cells link to local cross-reactive IgG upon human rhinovirus infection". In: *Cell Reports* 30.2 (2020), pp. 351–366.

[60] Wafaa B Alsoussi, Jackson S Turner, James B Case, Haiyan Zhao, Aaron J Schmitz, Julian Q Zhou, Rita E Chen, Tingting Lei, Amena A Rizk, Katherine M McIntire, et al. "A potently neutralizing antibody protects mice against SARS-CoV-2 infection". In: *The Journal of Immunology* 205.4 (2020), pp. 915–922.

[61] Hamish W King, Nara Orban, John C Riches, Andrew J Clear, Gary Warnes, Sarah A Teichmann, and Louisa K James. "Single-cell analysis of human B cell maturation predicts how antibody class switching shapes selection dynamics". In: *Science Immunology* 6.56 (2021), eabe6291.

[62] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[63] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. "SAbDab: the structural antibody database". In: *Nucleic Acids Research* 42.D1 (2014), pp. D1140–D1146.

[64] Helen M Berman, Tammy Battistuz, Talapady N Bhat, Wolfgang F Bluhm, Philip E Bourne, Kyle Burkhardt, Zukang Feng, Gary L Gilliland, Lisa Iype, Shri Jain, et al. "The Protein Data Bank". In: *Acta Crystallographica Section D: Biological Crystallography* 58.6 (2002), pp. 899–907.

[65] Cristian Regep, Guy Georges, Jiye Shi, Bojana Popovic, and Charlotte M Deane. "The H3 loop of antibodies shows unique structural characteristics". In: *Proteins: Structure, Function, and Bioinformatics* 85.7 (2017), pp. 1311–1318.

[66] Robert W Floyd. "Algorithm 97: shortest path". In: *Communications of the ACM* 5.6 (1962), p. 345.

[67] Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

# Chapter 4

# Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies

Adapted from Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. "Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies". *bioRxiv* (2022). Reproduced with permission.

## 4.1 Abstract

Antibodies have the capacity to bind a diverse set of antigens, and they have become critical therapeutics and diagnostic molecules. The binding of antibodies is facilitated by a set of six hypervariable loops that are diversified through genetic recombination and mutation. Even with recent advances, accurate structural prediction of these loops remains a challenge. Here, we present IgFold, a fast deep learning method for antibody structure prediction.

IgFold consists of a pretrained language model trained on 558M natural antibody sequences followed by graph networks that directly predict backbone atom coordinates. IgFold predicts structures of similar or better quality than alternative methods (including AlphaFold) in significantly less time (under 25 seconds). Accurate structure prediction on this timescale makes possible avenues of investigation that were previously unfeasible. As a demonstration of IgFold's capabilities, we predicted structures for 1.4 million paired antibody sequences, providing structural insights to 500-fold more antibodies than have experimentally determined structures.

## 4.2 Introduction

Antibodies play a critical role in the immune response against foreign pathogens. Through genetic recombination and hyper-mutation, the adaptive immune system is capable of generating a vast number of potential antibodies. Immune repertoire sequencing provides a glimpse into an individual's antibody population [1]. Analysis of these repertoires can further our understanding of the adaptive immune response [2] and even suggest potential therapeutics [3]. However, sequence data alone provides only a partial view into the immune repertoire. The interactions that facilitate antigen binding are determined by the structure of a set of six loops that make up a complementarity determining region (CDR). Accurate modeling of these CDR loops provides insights into these binding mechanisms and promises to enable rational design of specific antibodies [4]. Five of the CDR loops tend to adopt canonical folds that can be predicted effectively by sequence similarity [5]. However, the third CDR

loop of the heavy chain (CDR H3) has proven a challenge to model due to its increased diversity, both in sequence and length [6, 7]. Further, the position of the H3 loop at the interface between the heavy and light chains makes its conformation dependent on the inter-chain orientation [8, 9]. Given its central role in binding, advances in prediction of H3 loop structures are critical for understanding antibody-antigen interactions and enabling rational design of antibodies.

Deep learning methods have brought about a revolution in protein structure prediction [10, 11]. With the development of AlphaFold, accurate protein structure prediction has largely become accessible to all [12]. Beyond monomeric proteins, AlphaFold-Multimer has demonstrated an impressive ability to model protein complexes [13]. However, performance on antibody structures remains to be extensively validated. Meanwhile, antibody-specific deep learning methods such as DeepAb [14] and ABlooper [15] have significantly improved CDR loop modeling accuracy, including for the challenging CDR H3 loop [7, 16]. DeepAb predicts a set of inter-residue geometric constraints that are fed to Rosetta to produce a complete $F_V$ structure [14]. ABlooper predicts CDR loop structures in an end-to-end fashion, with some post-prediction refinement required, while also providing an estimate of loop quality [15]. Another tool, NanoNet [17], has been trained specifically for prediction of single-chain antibodies (nanobodies) and provides fast predictions. While effective, certain design decisions limit the utility of both models. DeepAb predictions are relatively slow (ten minutes per sequence), cannot effectively incorporate template data, and offer little insight into expected

quality. ABlooper, while faster and more informative, relies on external tools for framework modeling, cannot incorporate CDR loop templates, and does not support nanobody modeling.

Concurrent with advances in structure prediction, self-supervised learning on massive sets of unlabeled protein sequences has shown remarkable utility across protein modeling tasks [18, 19]. Embeddings from transformer encoder models trained for masked language modeling have been used for variant prediction [20], evolutionary analysis [21, 22], and as features for protein structure prediction [23, 24]. Auto-regressive transformer models have been used to generate functional proteins entirely from sequence learning [25]. The wealth of immune repertoire data provided by sequencing experiments has enabled development of antibody-specific language models. Models trained for masked language modeling have been shown to learn meaningful representations of immune repertoire sequences [22, 26, 27], and even repurposed to humanize antibodies [28]. Generative models trained on sequence infilling have been shown to generate high-quality antibody libraries [29, 30].

In this work, we present IgFold: a fast, accurate model for end-to-end prediction of antibody structures from sequence. IgFold leverages embeddings from AntiBERTy [22], a language model pretrained on 558M natural antibody sequences, to directly predict the atomic coordinates that define the antibody structure. Our model was the first to combine a single-sequence pretrained language model with an equivariant structure module for protein structure prediction, an approach which has since seen success for general protein structure prediction [31, 32]. Predictions from IgFold match the accuracy of the

recent AlphaFold models [10, 13] while being much faster (under 25 seconds). IgFold also provides flexibility beyond the capabilities of alternative antibody-specific models, including robust incorporation of template structures and support for nanobody modeling.

## 4.3 Results

### 4.3.1 End-to-end prediction of antibody structure

Our method for antibody structure prediction, IgFold, utilizes learned representations from the pretrained AntiBERTy language model to directly predict 3D atomic coordinates (Figure 4.1). Structures from IgFold are accompanied by a per-residue accuracy estimate, which provides insights into the quality of the prediction.

#### Embeddings from pretrained model encode structural features

The limited number of experimentally determined antibody structures (thousands [33]) presents a difficultly in training an effective antibody structure predictor. In the absence of structural data, self-supervised language models provide a powerful framework for extracting patterns from the significantly greater number (billions [34]) of natural antibody sequences identified by immune repertoire sequencing studies. For this work, we used AntiBERTy [22], a transformer language model pretrained on 558M natural antibody sequences, to generate embeddings for structure prediction. Similar to the role played by alignments of evolutionarily related sequences for general protein structure prediction [35], embeddings from AntiBERTy act as a contextual

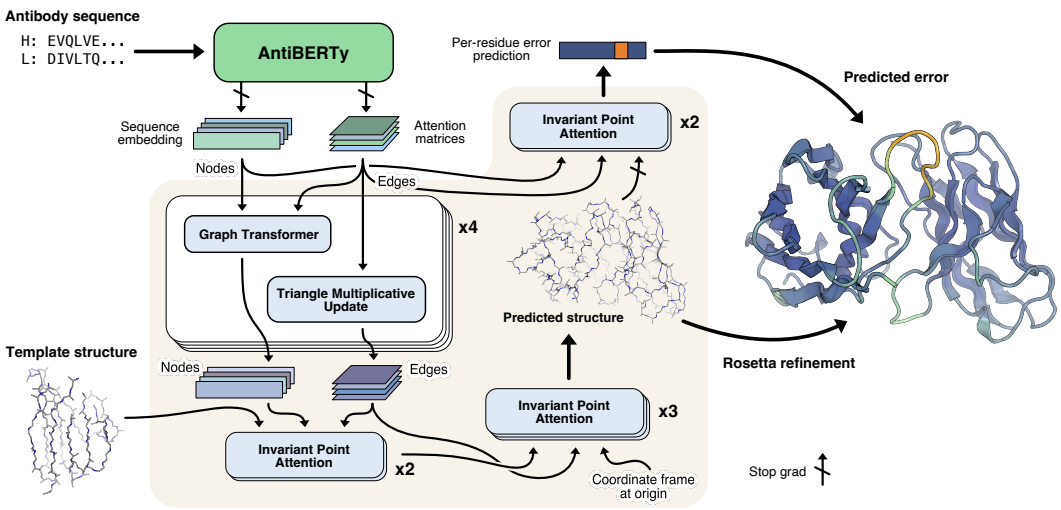


**Figure 4.1:** Diagram of method for end-to-end prediction of antibody structures

Antibody sequences are converted into contextual embeddings using AntiBERTy, a pretrained language model. From these representations, IgFold uses a series of transformer layers to directly predict atomic coordinates for the protein backbone atoms. For each residue, IgFold also provides an estimation of prediction quality. Refinement of predictions and addition of side chains is performed by Rosetta.

representation that places individual sequences within the broader antibody space.

Prior work has demonstrated that protein language models can learn structural features from sequence pretraining alone [18, 36]. To investigate whether sequence embeddings from AntiBERTy contained nascent structural features, we generated embeddings for the set of 3,467 paired antibody sequences with experimentally determined structures in the PDB. For each sequence, we extracted the portions of the embedding corresponding to the six CDR loops and averaged to obtain fixed-sized CDR loop representations (one per loop). We then collected the embeddings for each CDR loop across all sequences and visualized using two-dimensional t-SNE (Figure 4.15). To determine whether

the CDR loop representations encoded structural features, we labeled each point according to its canonical structural cluster. For CDR H3, which lacks canonical clusters, we instead labeled by loop length. For the five CDR loops that adopt canonical folds, we observed some organization within the embedded space, particularly for CDR1 loops. For the CDR H3 loop, we found that the embedding space did not separate into natural clusters, but was rather organized roughly in accordance with loop length. These results suggest that AntiBERTy has learned some distinguishing structural features of CDR loops through sequence pretraining alone.

**Coordinate prediction from sequence embeddings**

To predict 3D atomic coordinates from sequence embeddings, we adopt a graphical representation of antibody structure, with each residue as a node and information passing between all pairs of residues (Figure 4.1). The nodes are initialized using the final hidden layer embeddings from AntiBERTy. To initialize the edges, we collect the full set of inter-residue attention matrices from each layer of AntiBERTy. These attention matrices are a useful source of edge information as they encode the residue-residue information pathways learned by the pretrained model. For paired antibodies, we concatenate the sequence embeddings from each chain and initialize inter-chain edges to zero. We do not explicitly provide a chain break delimiter, as the pretrained language model already includes a positional embedding for each sequence. The structure prediction model begins with a series of four graph transformer [37] layers interleaved with edge updates via the triangle multiplicative layer proposed for AlphaFold [10].

Following the initial graph transformer layers, we incorporate structural template information into the nascent representation using invariant point attention (IPA) [10]. In contrast to the application of IPA for the AlphaFold structure module, we fix the template coordinates and use IPA as a form of structure-aware self-attention. This enables the model to incorporate the local structural environment into the sequence representation directly from the 3D coordinates, rather than switching to an inter-residue representation (e.g., distance or contact matrices). We use two IPA layers to incorporate template information. Rather than search for structural templates for training, we generate template-like structures by corruption of the true label structures. Specifically, for 50% of training examples, we randomly select one to six consecutive segments of twenty residues and move the atomic coordinates to the origin. The remaining residues are provided to the model as a template. The deleted segments of residues are hidden from the IPA attention, so that the model only incorporates structural information from residues with meaningful coordinates.

Finally, we use another set of IPA layers to predict the final 3D antibody structure. Here, we employ a strategy similar to the AlphaFold structure module [10] and train a series of three IPA layers to translate and rotate each residue from an initialized position at the origin to the final predicted position. We depart slightly from the AlphaFold implementation and learn separate weights for each IPA layer, as well as allow gradient propagation through the rotations. To train the model for structure prediction, we minimize the mean-squared error between the predicted coordinates and the experimental

structure after Kabsch alignment. In practice, we observe that the first IPA layer is sufficient to learn the global arrangement of residues (albeit in a compact form), while the second and third layers function to produce the properly scaled structure with correct bond lengths and angles (Figure 4.17).

**Per-residue error prediction**

Simultaneously with structure prediction training, we additionally train the model to estimate the error in its own predictions. For error estimation, we use two IPA layers that operate similarly to the template incorporation layers (i.e., without coordinate updates). The error estimation layers take as input the final predicted structure, as well as a separate set of node and edge features derived from the initial AntiBERTy features. We stop gradient propagation through the error estimation layers into the predicted structure to prevent the model from optimizing for accurately estimated, but highly erroneous structures. For each residue, the error estimation layers are trained to predict the deviation of the $N$, $C_\alpha$, $C$, and $C_\beta$ atoms from the experimental structure after a Kabsch alignment of the beta barrel residues. We use a different alignment for error estimation than structure prediction to more closely mirror the conventional antibody modeling evaluation metrics. The model is trained to minimize the L1 norm of the predicted $C_\alpha$ deviation minus the true deviation.

**Structure dataset augmentation with AlphaFold**

We sought to train the model on as many immunoglobulin structures as possible. From the Structural Antibody Databae (SAbDab) [33], we obtained 4,275 structures consisting of paired antibodies and single-chain nanobodies. Given

the remarkable success of AlphaFold for modeling both protein monomers and complexes, we additionally explored the use of data augmentation to produce structures for training. To produce a diverse set of structures for data augmentation, we clustered [38] the paired and unpaired partitions of the Observed Antibody Space [34] at 40% and 70% sequence identity, respectively. This clustering resulted in 16,141 paired sequences and 26,971 unpaired sequences. Because AlphaFold-Multimer [13] was not yet released, all predictions were performed with the original AlphaFold model [10]. For the paired sequences, we modified the model inputs to enable complex modeling by inserting a gap in the positional embeddings (i.e., AlphaFold-Gap [12, 13]). For the unpaired sequences, we discarded the predicted structures with average pLDDT (AlphaFold error estimate) less than 85, leaving 22,132 structures. These low-confidence structures typically correponded to sequences with missing residues at the N-terminus. During training, we sample randomly from the three datasets with examples weighted inversely to the size of their respective datasets, such that roughly one third of total training examples come from each dataset.

### 4.3.2 Antibody structure prediction benchmark

To evaluate the performance of IgFold against recent methods for antibody structure prediction, we assembled a non-redundant set of antibody structures deposited after compiling our training dataset. We chose to compare performance on a temporally separated benchmark to ensure that none of the methods evaluated had access to any of the structures during training. In

total, our benchmark contains 197 paired antibodies and 71 nanobodies.

**Predicted structures are high quality before refinement**

As an end-to-end model, IgFold directly predicts structural coordinates as its output. However, these immediate structure predictions are not guaranteed to satisfy realistic molecular geometries. In addition to incorporating missing atomic details (e.g., side chains), refinement with Rosetta [39] corrects any such abnormalities. To better understand the impact of this refinement step, we compared the directly predicted structures for each target in the benhmark to their refined counterparts. In general, we observed very little change in the structures (Figure 4.18), with an average RMSD less than 0.5 Å before and after refinement. The exception to this trend is abnormally long CDR loops, particularly CDR H3. We compared the pre- and post-refinement structures for benchmark targets with three of the longest CDR H3 loops to those with shorter loops and found that the longer loops frequently contained unrealistic bond lengths and backbone torsion angles (Figure 4.19). Similar issues have been observed in recent previous work [15], indicating that directly predicting atomically correct long CDR loops remains a challenge.

**Accurate antibody structures in a fraction of the time**

We compared the performance of IgFold against a mixture of grafting and deep learning methods for antibody structure prediction. Although previous work has demonstrated significant improvements by deep learning over grafting-based methods, we continue to benchmark against grafting to track its performance as increasingly many antibody structures become available.

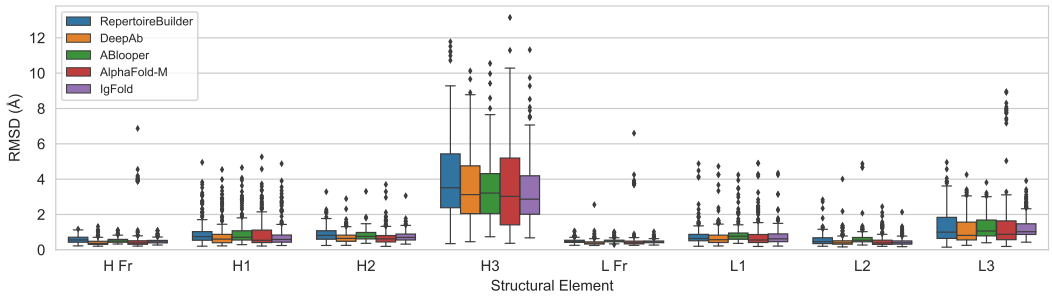


**Figure 4.2:** Comparison of methods for antibody structure prediction

Benchmark performance of RepertoireBuilder, DeepAb, ABlooper, AlphaFold-Multimer, and IgFold for paired antibody structure prediction. All root-mean-squared-deviation (RMSD) values calculated over backbone heavy atoms after alignment of the respective framework residues.

For each benchmark target, we predicted structures using RepertoireBuilder [40], DeepAb [14], ABlooper [15], and AlphaFold-Multimer [13]. We opted to benchmark the ColabFold [12] implementation of AlphaFold, rather than the original pipeline from DeepMind, due to its significant runtime acceleration and similar accuracy. Of these methods, RepertoireBuilder utilizes a grafting-based algorithm for structure prediction and the remaining use some form of deep learning. DeepAb and ABlooper are both trained specifically for paired antibody structure prediction, and have previously reported comparable performance. AlphaFold-Multimer has demonstrated state-of-the-art performance for protein complex prediction – however, performance on antibody structures specifically remains to be evaluated.

The performance of each method was assessed by measuring the backbone heavy-atom (N, $C_\alpha$, C, O) RMSD between the predicted and experimentally determined structures for the framework residues and each CDR loop. All RMSD values are measured after alignment of the framework residues. In

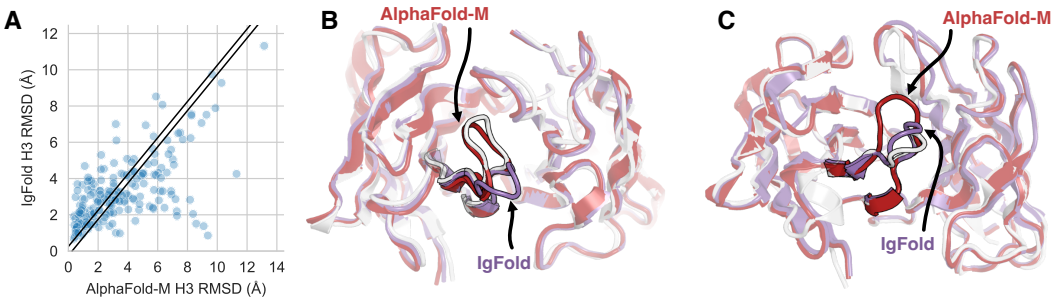


**Figure 4.3:** Comparison between IgFold and AlphaFold-Multimer for CDR H3 loop structure prediction

(A) Per-target comparison of CDR H3 loop structure prediction for IgFold and AlphaFold-Multimer, with each point representing the $RMSD_{H3}$ for both methods on a single benchmark target. (B) Comparison of predicted CDR H3 loop structures for target 7N3G ($L_{H3}$ = 10 residues) for IgFold ($RMSD_{H3}$ = 4.69 Å) and AlphaFold-Multimer ($RMSD_{H3}$ = 0.98 Å). (C) Comparison of predicted CDR H3 loop structures for target 7RNJ ($L_{H3}$ = 9 residues) for IgFold ($RMSD_{H3}$ = 1.18 Å) and AlphaFold-Multimer ($RMSD_{H3}$ = 3.46 Å).

general, we observed state-of-the-art performance for all of the deep learning methods while grafting performance continued to lag behind (Figure 4.2, Table 4.1). On average, all of the antibody-specific methods predicted both the heavy and light chain framework structures with high accuracy (0.43-0.53 Å and 0.41 - 0.51 Å, respectively). AlphaFold-Multimer typically performed well on framework residues, except for a set of fourteen predictions where the model predicted C-terminal strand swaps between the heavy and light chains 4.20. For the CDR1 and CDR2 loops, all methods produced sub-angstrom predictions on average. The largest improvement in prediction accuracy by deep learning methods is observed for the CDR3 loops.

We also considered the predicted orientation between the heavy and light chains, which is an important determinant of the overall binding surface [8, 9]. Accuracy of the inter-chain orientation was evaluated by measuring the

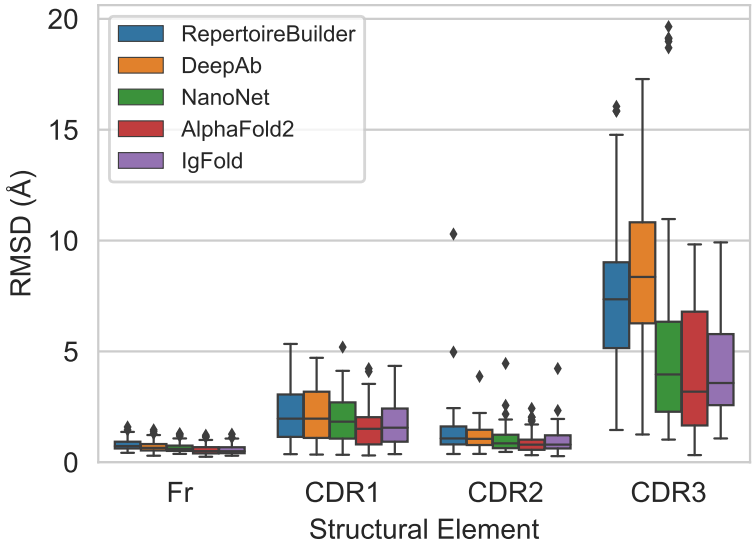


**Figure 4.4:** Comparison of methods for nanobody structure prediction

Benchmark performance of RepertoireBuilder, DeepAb, NanoNet AlphaFold2, and IgFold for nanobody structure prediction. All root-mean-squared-deviation (RMSD) values calculated over backbone heavy atoms after alignment of the framework residues.

deviation from native of the inter-chain packing angle, inter-domain distance, heavy-opening angle, and light-opening angle. Each of these orienational coordinates are rescaled by dividing by their respective standard deviations (calculated over the set of experimentally determined antibody structures) and summed to obtain an orientational coordinate distance (OCD) [9]. We found that in general deep learning methods produced $F_V$ structures with OCD values near four, indicating that the predicted structures are typically within about one standard deviation of the native structures for each of the components of OCD.

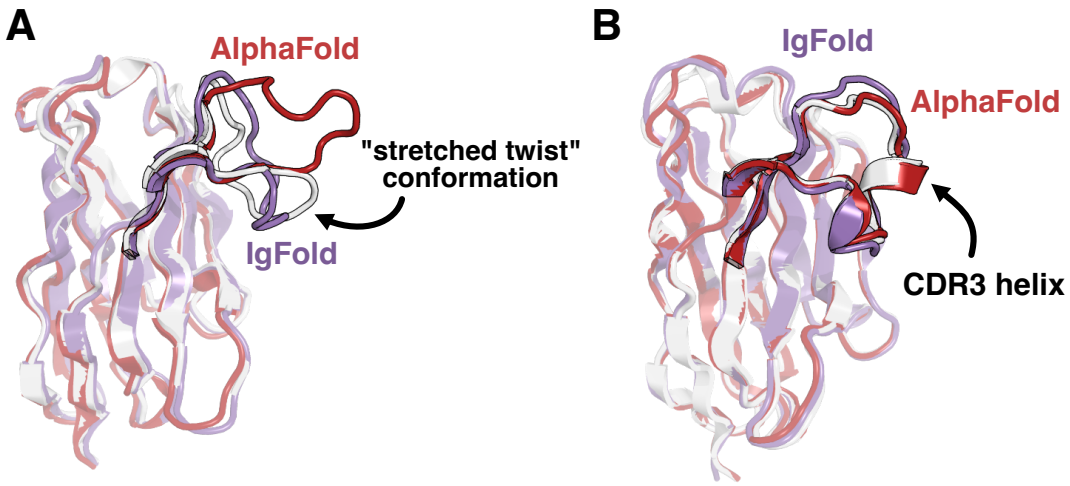


**Figure 4.5:** Comparison between IgFold and AlphaFold2 for nanobody CDR3 loop structure prediction

(A) Comparison of predicted CDR H3 loop structures for target 7AQZ ($L_{CDR3}$ = 15 residues) for IgFold ($RMSD_{CDR3}$ = 2.87 Å) and AlphaFold ($RMSD_{CDR3}$ = 7.08 Å). (B) Comparison of predicted CDR H3 loop structures for target 7AR0 ($L_{CDR3}$ = 17 residues) for IgFold ($RMSD_{CDR3}$ = 2.34 Å) and AlphaFold ($RMSD_{CDR3}$ = 0.84 Å).

Given the comparable aggregate performance of the deep learning methods, we further investigated the similarity between the structures predicted by each method. For each pair of methods, we measured the RMSD of framework and CDR loop residues, as well as the OCD, between the predicted structures for each benchmark target (Figure 4.24). We additionally plotted the distribution of structural similarities between IgFold and the alternative methods (Figure 4.25). We found that the framework structures (and their relative orientations) predicted by IgFold resembled those of DeepAb and ABlooper, but were less similar to those of RepertoireBuilder and AlphaFold-Multimer. The similarity between IgFold and ABlooper is expected, given that ABlooper predictions were based on IgFold-predicted framework structures. We also observed that the heavy chain CDR loops from IgFold, DeepAb, and ABlooper

**Table 4.1:** Accuracy of predicted antibody Fv structures

| Method | OCD | H Fr (Å) | H1 (Å) | H2(Å) | H3 (Å) | L Fr (Å) | L1 (Å) | L2(Å) | L3 (Å) |
|---|---|---|---|---|---|---|---|---|---|
| RepertoireBuilder | 5.09 | 0.59 | 1.00 | 0.90 | 4.15 | 0.49 | 0.81 | 0.57 | 1.32 |
| DeepAb | 3.60 | 0.43 | 0.86 | 0.72 | 3.57 | 0.41 | 0.75 | 0.48 | 1.16 |
| ABlooper | 4.42 | 0.53 | 0.98 | 0.83 | 3.54 | 0.51 | 0.92 | 0.67 | 1.32 |
| AlphaFold-Multimer | 4.18 | 0.69 | 0.95 | 0.74 | 3.56 | 0.66 | 0.84 | 0.51 | 1.59 |
| IgFold | 3.82 | 0.48 | 0.85 | 0.76 | 3.27 | 0.46 | 0.76 | 0.46 | 1.30 |

were quite similar on average. We observe further similarity on light chain CDRs between IgFold and DeepAb. These agreements likely extend from training on similar, antibody-focused datasets.

**Deep learning methods converge on CDR H3 accuracy**

The average prediction accuracy for the highly variable, conformationally diverse CDR H3 loop was relatively consistent among the four deep learning methods evaluated (Table 4.1), though IgFold performed the best on average. Given this convergence in performance, we again considered the similarity between the CDR H3 loop structures predicted by each method. IgFold, DeepAb, and ABlooper produced the most similar CDR H3 loops, with an average RMSD of 2.01 - 2.34 Å between predicted structures for the three methods . This may reflect the similar training datasets used for the methods, which were limited to antibody structures. AlphaFold-Multimer, by contrast, predicted the most distinct CDR H3 loops, with an average RMSD 3.10 - 3.57 Å from the other deep learning methods.

The dissimilarity of predictions between IgFold and AlphaFold-Multimer is surprising, given the extensive use of AlphaFold-predicted structures for training IgFold. When we compared the per-target accuracy of IgFold and

AlphaFold-Multimer, we found many cases where one method predicted the CDR H3 loop accurately while the other failed (Figure 4.3A). Indeed, approximately 20% of CDR H3 loops predicted by the two methods were greater than 4 Å RMSD apart, meaning the methods often predict distinct conformations. To illustrate the structural implications of these differences in predictions, we highlight two targets from the benchmark where IgFold and AlphaFold-Multimer diverge. In one such target (target 7N3G [41], Figure 4.3B), AlphaFold-Multimer effectively predicts the CDR H3 loop structure ($RMSD_{H3}$ = 0.98 Å) while IgFold predicts a distinct, and incorrect, conformation ($RMSD_{H3}$ = 4.69 Å). However, for another example (target 7RNJ [42], Figure 4.3C), IgFold more accurately predicts the CDR H3 loop structure ($RMSD_{H3}$ = 1.18 Å) while AlphaFold-Multimer predicts an alternative conformation ($RMSD_{H3}$ = 3.46 Å).

**Fast nanobody structure prediction remains a challenge**

Single domain antibodies, or nanobodies, are an increasingly popular format for therapeutic development [43]. Structurally, nanobodies share many similarities with paired antibodies, but with the notable lack of a second immunoglobulin chain. This, along with increased nanobody CDR3 loop length, makes accessible a wide range of CDR3 loop conformations not observed for paired antibodies [44]. We compared the performance of IgFold for nanobody structure prediction to RepertoireBuilder [40], DeepAb [14], NanoNet [17], and AlphaFold [10] (Figure 4.4, Table 4.2). We omitted ABlooper from the comparison as it predicts only paired antibody structures.

As with paired antibodies, all methods evaluated produced highly accurate predictions for the framework residues, with the average RMSD ranging from 0.57 Å to 0.80 Å. No method achieves sub-angstrom accuracy on average for CDR1 loops, though AlphaFold and IgFold achieve the best performance. For CDR2 loops, we observe a substantial improvement by IgFold and the other deep learning methods over RepertoireBuilder, with AlphaFold achieving the highest accuracy on average. For the CDR3 loop, RepertoireBuilder prediction quality is highly variable (average $RMSD_{CDR3}$ of 7.54 Å), reflective of the increased difficultly of identifying suitable template structures for the long, conformationally diverse loops. DeepAb achieves the worst performance for CDR3 loops, with an average $RMSD_{CDR3}$ of 8.52 Å, probably because its training dataset was limited to paired antibodies [14], and thus the model has never observed the full range of conformations accessible to nanobody CDR3 loops. NanoNet, trained specifically for nanobody structure prediction, outperforms DeepAb (average $RMSD_{CDR3}$ of 5.43 Å). AlphaFold displays the best performance for CDR3 loops, with an average $RMSD_{CDR3}$ of 4.00 Å, consistent with its high accuracy on general protein sequences. IgFold CDR3 predictions tend to be slightly less accurate than those of AlphaFold (average $RMSD_{CDR3}$ of 4.25 Å), but are significantly faster to produce (fifteen seconds for IgFold, versus six minutes for the ColabFold implementation of AlphaFold).

To better understand the distinctions between IgFold- and AlphaFold-predicted nanobody structures, we highlight two examples from the benchmark. First, we compared the structures predicted by both methods for the

benchmark target 7AQZ (unpublished, Figure 4.5A). This nanobody features a 15-residue CDR3 loop that adopts the "stretched-twist" conformation [44], in which the CDR3 loop bends to contact the framework residues that would otherwise be obstructed by a light chain in a paired antibody. IgFold correctly predicts this nanobody-specific loop conformation ($RMSD_{CDR3}$ = 2.87 Å), while AlphaFold predicts an extended CDR3 conformation ($RMSD_{CDR3}$ = 7.08 Å). Indeed, there are other cases where either IgFold or AlphaFold correctly predicts the CDR3 loop conformation while the other fails (see off-diagonal points in Figure 4.23G). In the majority of such cases, AlphaFold predicts the correct conformation, yielding the lower average CDR3 RMSD. In a second example, we compared the structures predicted by both methods for the benchmark target 7AR0 (unpublished, Figure 4.5B). This nanobody has a long 17-residue CDR3 loop with a short helical region. Although both methods correctly predict the loop conformation, IgFold fails to predict the helical secondary structure, resulting in a less accurate prediction ($RMSD_{CDR3}$ = 2.34 Å) than that of AlphaFold ($RMSD_{CDR3}$ = 0.84 Å). Such structured loops highlight a key strength of AlphaFold, which was trained on a large dataset of general proteins and has thus encountered a broad variety of structral arrangements, over IgFold, which has observed relatively few such structures within its training dataset.

### 4.3.3 Error predictions identify inaccurate CDR loops

Although antibody structure prediction methods continue to improve, accurate prediction of abnormal CDR loops (particularly long CDR H3 loops)

**Table 4.2:** Accuracy of predicted nanobody structures

| Method | Fr (Å) | CDR1 (Å) | CDR2(Å) | CDR3 (Å) |
|---|---|---|---|---|
| RepertoireBuilder | 0.80 | 2.12 | 1.37 | 7.54 |
| DeepAb | 0.72 | 2.14 | 1.14 | 8.52 |
| NanoNet | 0.66 | 1.94 | 1.05 | 5.43 |
| AlphaFold | 0.57 | 1.61 | 0.88 | 4.00 |
| IgFold | 0.58 | 1.73 | 0.98 | 4.25 |

remains inconsistent [6, 14, 15]. Determining whether a given structural prediction is reliable is critical for effective incorporation of antibody structure prediction into workflows. During training, we task IgFold with predicting the deviation of each residue's $C_\alpha$ atom from the native (under alignment of the beta barrel residues). We then use this predicted deviation as a per-residue error estimate to assess expected accuracy of different structural regions.

To assess the utility of IgFold's error predictions for identifying inaccurate CDR loops, we compared the average predicted error for each CDR loop to the RMSD between the predicted loop and the native structure for the paired $F_V$ and nanobody benchmarks. We observed significant correlations between the predicted error and the loop RMSDs from native for all the paired $F_V$ CDR loops (Figure 4.26). For CDR H2 and CDR L2 loops, the correlations between predicted and measured RMSD were notably weaker. However, given the relatively high accuracy of predictions for these loops, there was little error to detect. For nanobodies, we observed significant correlations between the predicted error and RMSD for all the CDR loops (Figure 4.27). Interestingly, for all loops the model tended to predict lower RMSD than was measured. This may be a result of the imbalance between the smaller number of residues with higher RMSD (CDR loops) and the greater number with lower

RMSD (framework residues). In the future, this miscalibration may be solved by using a weighted loss function that penalizes larger errors more heavily. However, the model's ability to effectively rank the accuracy of different CDR loops is still useful for identifying potentially inaccurate predictions.

For the challenging-to-predict, conformationally diverse CDR3 loops, we observed significant correlations for both paired antibody H3 loops (Figure 4.6A, $\rho$ = 0.76) and nanobody CDR3 loops (Figure 4.6B, $\rho$ = 0.47). To illustrate the utility of error estimation for judging CDR H3 loop predictions, we highlight three examples from the benchmark. The first is the benchmark target 7O4Y [45], a human anti-CD22 antibody with a 12-residue CDR H3 loop. For 7O4Y, IgFold accurately predicts the extended beta sheet structure of the CDR H3 loop ($RMSD_{H3}$ = 1.64 Å), and estimates a correspondingly lower RMSD (Figure 4.8A). The second target is 7RKS [46], a human anti-SARS-CoV-2-receptor-binding-domain antibody with a 18-residue CDR H3 loop. IgFold struggles to predict the structured beta sheet within this long H3 loop, instead predicting a broad ununstructured conformation ($RMSD_{H3}$ = 6.33 Å). Appropriately, the error estimation for the CDR H3 loop of 7RKS is much higher (Figure 4.8B). The third example is 7O33 [47], a mouse anti-PAS (proine/alanine-rich sequence) antibody with a 3-residue CDR H3 loop. Again, IgFold accurately predicts the structure of this short loop ($RMSD_{H3}$ = 1.49 Å) and provides a correspondingly low error estimate (Figure 4.8C).

Antibody engineering campaigns often deviate significantly from the space of natural antibody sequences [48]. Predicting structures for such heavily engineered sequences is challenging, particularly for models trained primarily

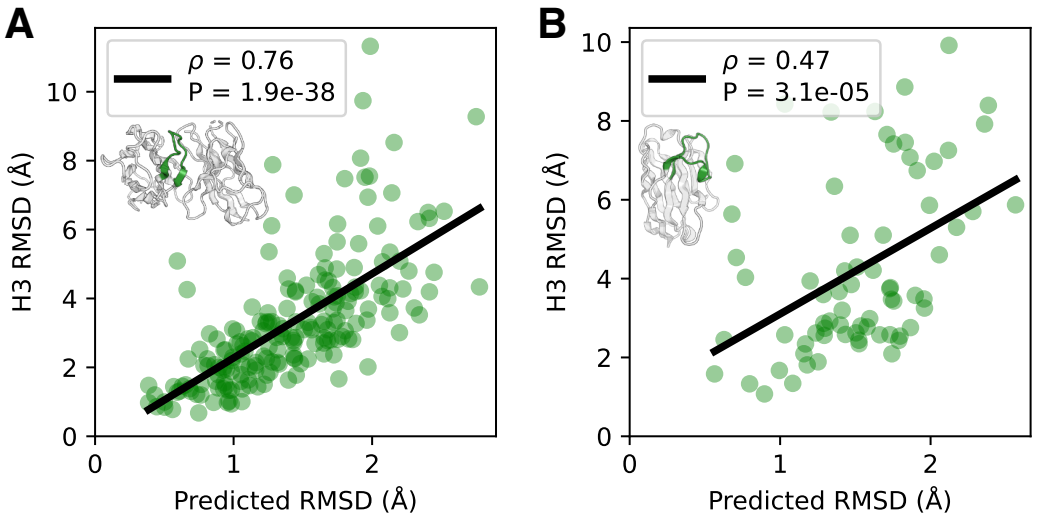


**Figure 4.6:** Error estimation for predicted antibody structures

(A) Comparison of CDR H3 loop RMSD to predicted error for paired antibody structure benchmark. Gray space represents cumulative average RMSD of predicted CDR H3 loops from native structure. (B) Comparison of CDR3 loop RMSD to predicted error for nanobody structure benchmark. Gray space represents cumulative average RMSD of predicted CDR3 loops from native structure.

on natural antibody structural data (such as IgFold). To investigate whether IgFold's error estimations can identify likely mistakes in such sequences, we predicted the structure of an anti-HLA (human leukocyte antigen) antibody with a sequence randomized CDR H1 loop [49] (Figure 4.7). As expected, there is significant error in the predicted CDR H1 loop structure. However, the erroneous structure is accompanied by a high error estimate, revealing that the predicted conformation is likely to be incorrect. This suggests that the RMSD predictions from IgFold are sensitive to unnatural antibody sequences and should be informative for a broad range of antibody structure predictions.

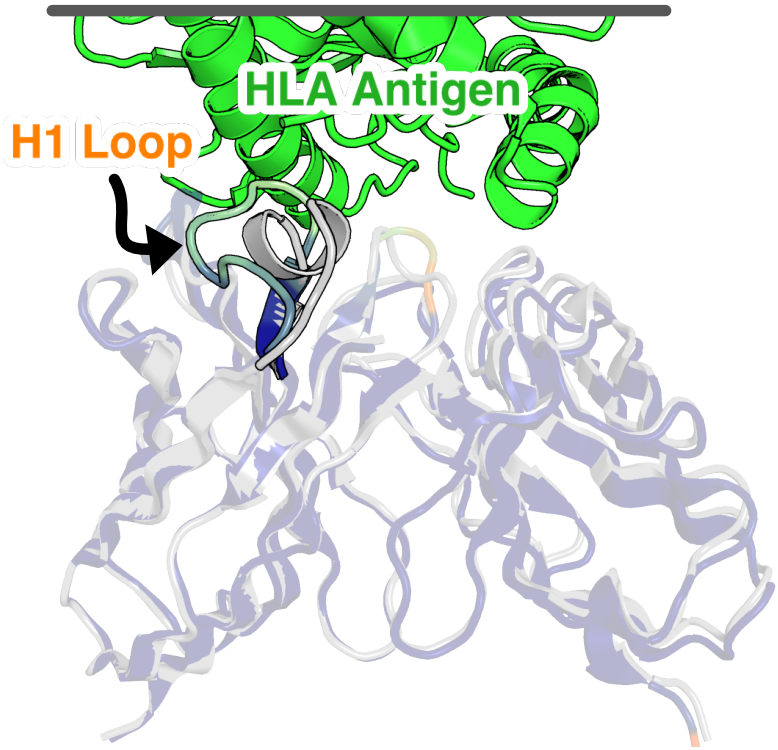


**Figure 4.7:** Predicted structure and error estimation for anti-HLA antibody with a randomized CDR H1 loop.

### 4.3.4 Template data is successfully incorporated into predictions

For many antibody engineering workflows, partial structural information is available for the antibody of interest. For example, crystal structures may be available for the parent antibody upon which new CDR loops were designed. Incorporating such information into structure predictions is useful for improving the quality of structure models. We simulated IgFold's behavior in this scenario by predicting structures for the paired antibody and nanobody benchmark targets while providing the coordinates of all non-H3 residues as templates. In general, we found that IgFold was able to incorporate the

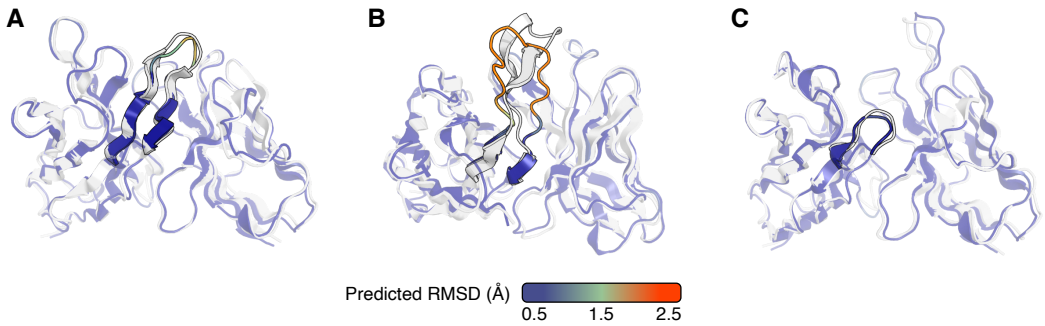


**Figure 4.8:** Examples of error estimation for CDR H3 loops

(A) Predicted structure and error estimation for benchmark target 7O4Y ($L_{H3}$ = 12 residues). (B) Predicted structure and error estimation for benchmark target 7RKS ($L_{H3}$ = 18 residues). (C) Predicted structure and error estimation for benchmark target 7O33 ($L_{H3}$ = 3 residues).

template data into its predictions, with the average RMSD for all templated CDR loops being significantly reduced (IgFold[Fv-H3]: Figure 4.9, IgFold[Fv-CDR3]: Figure 4.11). Although these results are not surprising, they showcase a key functionality lacking in prior antibody-specific methods [14, 15, 17].

Having demonstrated successful incorporation of structural data into predictions using templates, we next investigated the impact on accuracy of the untemplated CDR H3 loop predictions. For the majority of targets, we found little change in the accuracy of CDR H3 loop structures with the addition of non-H3 template information (Figure 4.10). For nanobodies, we observe more cases with substantial improvement to CDR3 loop predictions given template data (Figure 4.12).

We additionally experimented with providing the entire crystal structure to IgFold as template information. In this scenario, IgFold sucessfully incorporates the structural information of all CDR loops (including H3) into its predictions (IgFold[Fv]: Figure 4.9, Figure 4.11). Interestingly, the model's

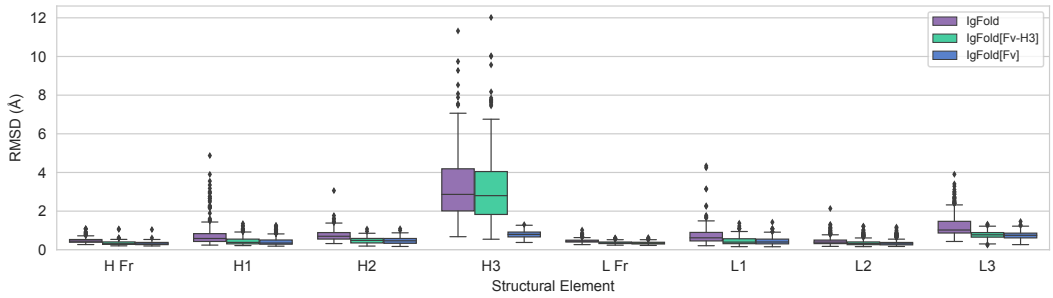


**Figure 4.9:** Incorporation of templates into antibody structure prediction

Paired antibody structure prediction benchmark results for IgFold without templates, IgFold given the $F_V$ structure without the CDR H3 loop (IgFold[Fv-H3]), and IgFold given the complete Fv structure (IgFold[Fv]).

incorporation of non-CDR3 templated regions also improves when the full structural context is provided, indicating that the model is not simply recapitulating template structures, but combining their content with its predictions. Although this approach is of little practical value for structure prediction (as the correct structure is already known) it may be a useful approach for instilling structural information into pretrained embeddings, which are valuable for other antibody learning tasks.

### 4.3.5 Minimal refinement yields faster predictions

Although the performance of the deep learning methods for antibody structure prediction is largely comparable, the speed of prediction is not. Grafting-based methods, such as RepertoireBuilder, tend to be much faster than deep learning methods (if a suitable template can be found). However, as reported above, this speed is obtained at the expense of accuracy. Recent deep learning methods for antibody structure prediction, including DeepAb, ABlooper, and

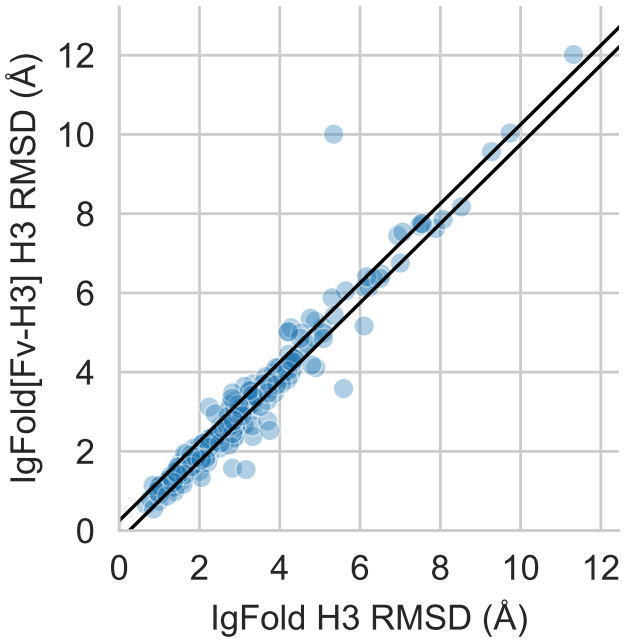


**Figure 4.10:** Effects of templates on CDR H3 loop structure prediction

Per-target comparison of CDR H3 loop structure prediction for IgFold and IgFold[Fv-H3], with each point representing the $RMSD_{H3}$ for both methods on a single benchmark target.

NanoNet, have claimed faster prediction of antibody structures as compared to general methods like AlphaFold. For our benchmark, all deep learning methods were run on identical hardware (12-core CPU with one A100 GPU), allowing us to directly compare their runtimes. All computed runtimes are measured from sequence to full-atom structure, using the recommended full-atom refinement protocols for each method. We could not evaluate the runtimes of RepertoireBuilder as no code has been published. The results of this comparison are summarized in Figure 4.13.

For paired antibodies, we find that IgFold is significantly faster any other method tested. On average, IgFold takes 23 seconds to predict a full-atom

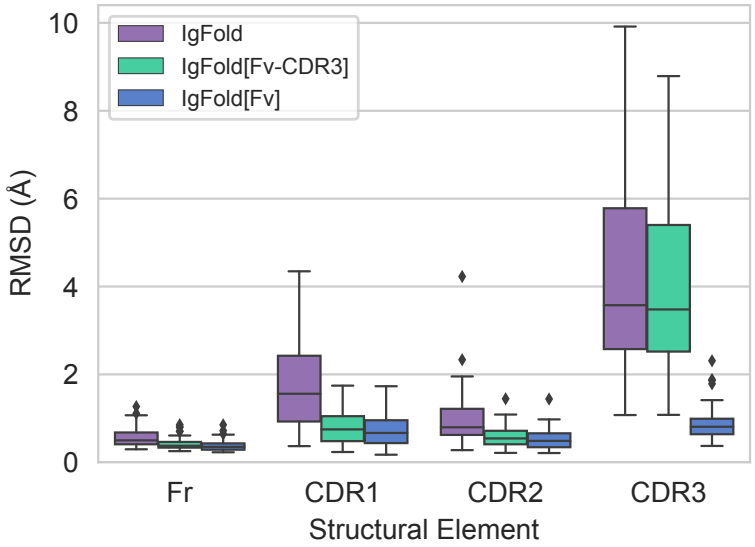


**Figure 4.11:** Incorporation of templates into nanobody structure prediction

Nanobody structure prediction benchmark results for IgFold without templates, IgFold given the $F_V$ structure without the CDR3 loop (IgFold[Fv-CDR3]), and IgFold given the complete Fv structure (IgFold[Fv]).

structure from sequence. The next fastest method, ABlooper, averages nearly three minutes (174 seconds) for full-atom structure prediction. Although ABlooper rapidly predicts coordinates in an end-to-end fashion, the outputs require expensive refinement in OpenMM to correct for geometric abnormalities and add side chains. The ColabFold [12] implementation of AlphaFold-Multimer evaluated here averages just over seven minutes (435 seconds) on average for full-atom structure prediction. This is considerably faster than the original implementation of AlphaFold-Multimer, which required an expensive MSA search and repeated model compilation for every prediction. Finally, the slowest method for paired antibody structure prediction was DeepAb, which averaged over twelve minutes (750 seconds). DeepAb is considerably slower

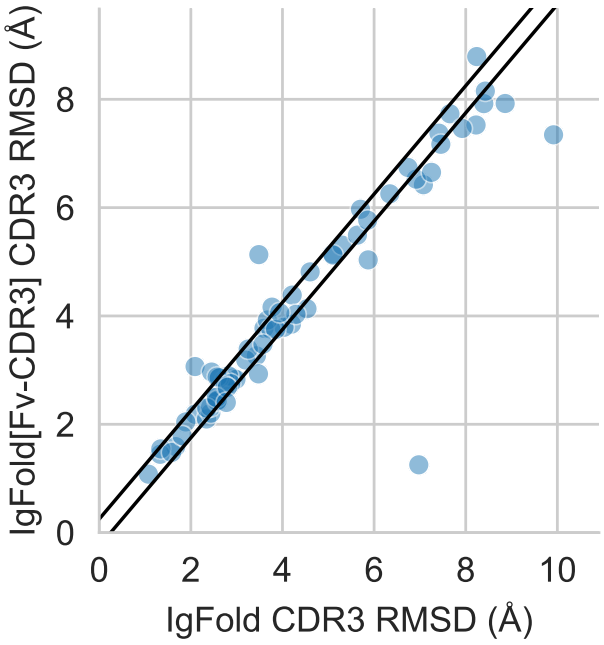


**Figure 4.12:** Effects of templates on CDR3 loop structure prediction

Per-target comparison of CDR3 loop structure prediction for IgFold and IgFold[Fv-CDR], with each point representing the $RMSD_{CDR3}$ for both methods on a single benchmark target.

by design, as it requires minimization of predicted inter-residue potentials in Rosetta. We also investigated the impact of sequence length on prediction times. In general, the runtimes of all methods increased with sequence length (Figure 4.28A). DeepAb and ABlooper were the most sensitive to sequence length, with AlphaFold-Multimer and IgFold scaling more favorably.

For nanobodies, we again find that IgFold outpaces alternative methods for full-atom structure prediction, requiring an average of 15 seconds. NanoNet was similarly fast, averaging 15 seconds for full-atom structure prediction. Similar to ABlooper for paired antibodies, NanoNet outputs require expensive refinement to correct for unrealistic backbone geometries and add side chains.

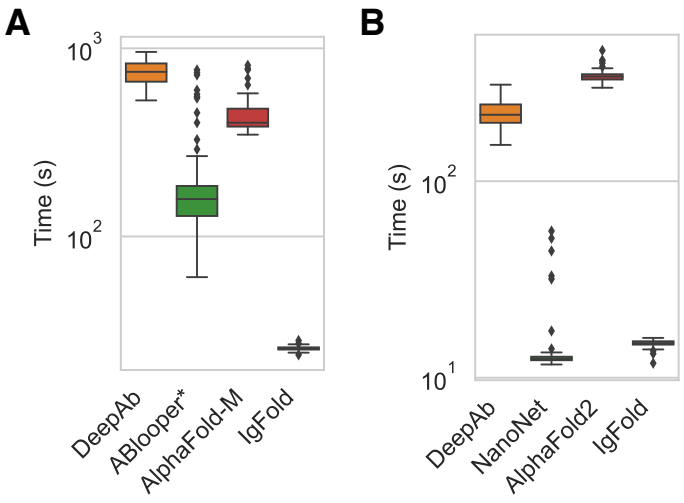


**Figure 4.13:** Runtime benchmark for antibody structure prediction methods

(A) Runtime comparison of evaluated methods on the paired antibody structure prediction benchmark. ABlooper runtimes are calculated given an IgFold-predicted framework, and thus represent an underestimation of actual runtime. (B) Runtime comparison of evaluated methods on the nanobody structure prediction benchmark.

DeepAb was able to predict nanobody structures in just under four minutes (224 seconds) on average. Finally, the slowest method for nanobody structure prediction was AlphaFold, which averaged nearly six minutes (345 seconds). As with paired antibodies, we also investigated the impact of sequence length on prediction times. In general, the runtimes of all methods increased with sequence length (Figure 4.28B). Although NanoNet had several outlier cases that required significant refinement, the prediction times for a majority of targets increased with sequence length. We also note that for methods capable of predicting both nanobody and paired antibody structures, runtimes tend to roughly double in the paired setting (scaling linearly with total length), as expected.

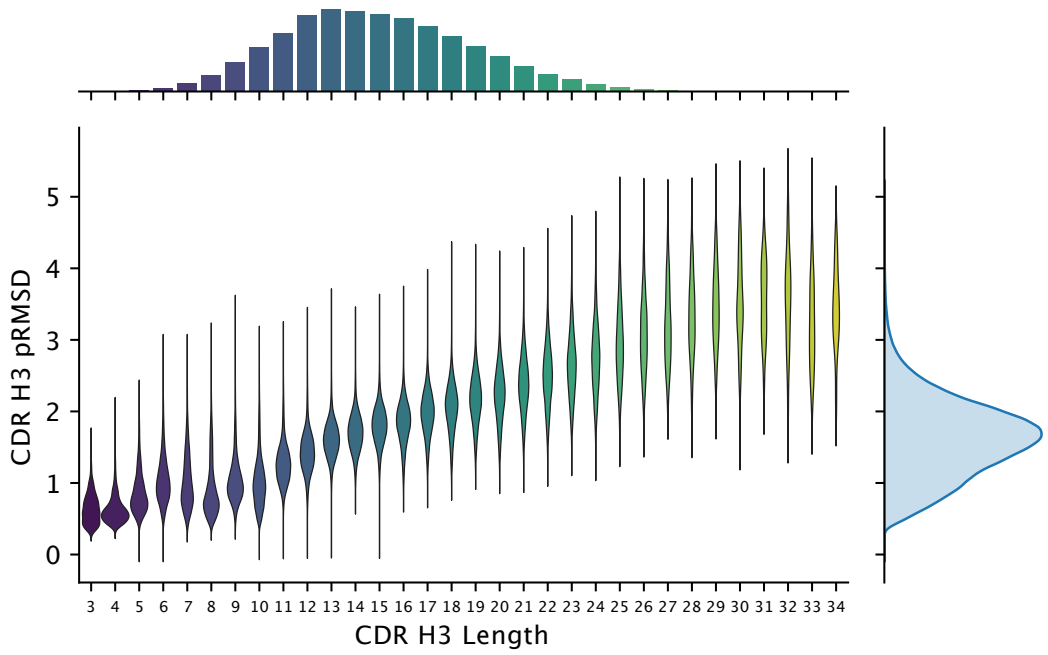


**Figure 4.14:** Estimated error for large-scale human antibody structure predictions

Distribution of predicted RMSD and CDR H3 loop lengths for 1.3M predicted human paired antibody structures.

### 4.3.6 Large-scale prediction of paired antibody structures

The primary advantage of IgFold over other highly accurate methods like AlphaFold is its speed at predicting antibody structures. This speed enables large-scale prediction of antibody structures on modest compute resources. Prior work exploring large-scale predictions of antibody structures have provided insight into the structural commonalities across individuals, and provide evidence of a public structural repertoire [50]. Further, comparison on the basis of structure (rather than sequence alone) has enabled discovery of convergent binders that diverge significantly in sequence [51]. To demonstrate the utility of IgFold's speed for such analyses, we predicted structures for

two non-redundant sets of paired antibodies. The first set consists of 104,994 paired antibody sequences (clustered at 95% sequence identity) from the OAS database [34]. These sequences are made up of 35,731 human, 16,356 mouse, and 52,907 rat antibodies. The second set contains 1,340,180 unique paired human antibody sequences from the immune repertoires of four unrelated individuals [52]. These sequences span the affinity maturation spectrum, consisting of both naive and memory B-cell sequences. The structures are predicted with low estimated RMSD by IgFold, indicating that they are accurate (Figure 4.29 and 4.30). We highlight the predicted accuracy of the CDR H3 loops for the 1.3M human antibody sequences in Figure 4.14. The median length and predicted RMSD for this set are 13 residues and 1.95 Å, respectively. We note that the predicted RMSD values tend to be underestimations, and in practice the actual H3 loop RMSDs, were structures to be experimentally determined, would likely be higher. As of October 2022, only 2,448 unique paired antibody structures have been determined experimentally [33], and thus our predicted dataset represents an over 500-fold expansion of antibody structural space. These structures are made available for use in future studies.

## 4.4 Discussion

Protein structure prediction methods have advanced significantly in recent years, and they are now approaching the accuracy of the experimental structures upon which they are trained [10]. These advances have been enabled in large part by effective exploitation of the structural information present

in alignments of evolutionarily related sequences (MSAs). However, constructing a meaningful MSA is time-consuming, contributing significantly to the runtime of general protein structure prediction models, and making high-throughput prediction of many protein structures computationally prohibitive for many users. In this work, we presented IgFold: a fast, accurate model that specializes in prediction of antibody structures. We demonstrated that IgFold matches the accuracy of the highly accurate AlphaFold-Multimer model [13] for paired antibody structure prediction, and approaches the accuracy of AlphaFold for nanobodies. Though prediction accuracy is comparable, IgFold is significantly faster than AlphaFold, and is able to predict structures in seconds. Further, for many targets IgFold and AlphaFold predict distinct conformations, which should be useful in assembling structural ensembles for applications where flexibility is important. Predicted structures are accompanied by informative error estimates, which provide critical information on the reliability of structures.

Analyses of immune repertoires have traditionally been limited to sequence data alone [1], as high-throughput antibody structure determination was experimentally prohibitive and prediction methods were too slow or inaccurate. However, incorporation of structural context has proven valuable, particularly for identification of sequence-dissimilar binders to common epitopes [53]. For example, grafting-based methods have been used to identify sequence-diverse but structurally similar antibodies against SARS-CoV-2 [51]. The increased accuracy of IgFold, coupled with its speed, will make such methods more effective. Additionally, consideration of structural uncertainty

via IgFold's error estimation should reduce the rate of false positives when operating on large volumes of sequences. As a demonstration of IgFold's capabilities, we predicted structures for over 1.4 million paired antibody sequences spanning three species. These structures expand on the number of experimentally determined antibody structures by a factor of 500. The majority of these structures are predicted with high confidence, suggesting that they are reliable. Although our analysis of these structures was limited, we are optimistic that this large dataset will be useful for future studies and model development.

Despite considerable improvements by deep learning methods for general protein complex prediction, prediction of antibody-antigen binding remains a challenge. Even the recent AlphaFold-Multimer model, which can accurately predict the interactions of many proteins, is still unable to predict how or whether an antibody will bind to a given antigen [13]. One of the key barriers to training specialized deep learning models for antibody-antigen complex prediction is the limited availability of experimentally determined structures. The large database of predicted antibody structures presented in this work may help reduce this barrier if it can be employed effectively. In the meantime, IgFold will provide immediate benefits to existing antibody-antigen docking methods. For traditional docking methods, the improvements to speed and accuracy by IgFold should be sufficient to make them more effective [54, 55]. For newer docking methods that incorporate structural flexibility, the error estimates from IgFold may be useful for directing enhanced sampling [56].

Deep learning methods trained on antibody sequences and structures hold

great promise for design of novel therapeutic and diagnostic molecules. Generative models trained on large numbers of natural antibody sequences can produce effective libraries for antibody discovery [29, 30]. Self-supervised models have also proven effective for humanization of antibodies [28]. Meanwhile, methods like AlphaFold and RoseTTAFold have been adapted for gradient-based design of novel protein structures and even scaffolding binding loops [57, 58]. IgFold will enable similar applications, and will additionally be useful as an oracle to test or score novel antibody designs. Finally, embeddings from IgFold (particularly when injected with structural information from templates) will be useful features for future antibody design tasks.

## 4.5 Methods

### 4.5.1 Predicting antibody structure from sequence

The architecture and training procedure for IgFold are described below. Full details of the model architecture hyperparameters are detailed in Table 4.3. In total, IgFold contains 1.6M trainable parameters.

#### Generating AntiBERTy embeddings

To generate input features for structure prediction, we use the pretrained AntiBERTy language model [22]. AntiBERTy is a bidirectional transformer trained by masked language modeling on a set of 558M antibody sequences from the Observed Antibody Space. For a given sequence, we collect from AntiBERTy the final hidden layer state and the attention matrices for all layers. The hidden state of dimension $L \times 512$ is reduced to dimension $L \times d_{\text{node}}$ by a

fully connected layer. The attention matrices from all 8 layers of AntiBERTy (with 8 attention heads per layer) are stacked to form an $L \times L \times 64$ tensor. The stacked attention tensor is transformed to dimension $L \times L \times d_{\text{edge}}$ by a fully connected layer.

**IgFold model implementation**

The IgFold model takes as input per-residue embeddings (nodes) and inter-residue attention features (edges). These initial features are processed by a series node updates via graph transformer layers [37] and edge updates via triangular multiplicative operations [10]. Next, template data is incorporated via fixed-coordinate invariant point attention. Finally, the processed nodes and edges are used to predict the antibody backbone structure via invariant point attention. We detail each of these steps in the following subsections. Where possible, we use the same notation as in the original papers.

*Node updates via graph transformer layers*. Residue node embeddings are updated by graph transformer (GT) layers, which extend the powerful transformer architecture to include edge information [37]. Each GT layer takes as input a series of node embeddings $H^{(l)} = \{h_1, h_2, ..., h_L\}$, with $h_i \in \mathbb{R}^{d_{\text{node}}}$, and edges $e_{ij} \in \mathbb{R}^{d_{\text{edge}}}$. We calculate the multi-head attention for each node $i$ to all other nodes $j$ as follows:

$$q_{c,i} = \mathbf{W}_{c,q} h_i$$

$$k_{c,j} = \mathbf{W}_{c,k} h_j$$

$$e_{c,ij} = \mathbf{W}_{c,e} e_{ij}$$

$$\alpha_{c,ij} = \frac{\langle q_{c,i}, k_{c,j} + e_{c,ij} \rangle}{\sum_{u \in L} \langle q_{c,i}, k_{c,u} + e_{c,iu} \rangle}$$

where $\mathbf{W}_{c,q}, \mathbf{W}_{c,k}, \mathbf{W}_{c,e} \in \mathbb{R}^{d_{\text{node}} \times d_{\text{gt-head}}}$ are learnable parameters for the key, query, and edge tranformations for the $c$-th attention head with hidden size $d_{\text{gt-head}}$. In the above, $\langle q, k \rangle = \exp \frac{q^T k}{\sqrt{d}}$ is the exponential of the standard scaled dot product attention operation. Using the calculated attention, we aggregate updates from all nodes $j$ to node $i$ as follows:

$$v_{c,j} = \mathbf{W}_{c,v} h_j$$

$$\hat{h}_i = \|_c^C \left[ \sum_{j \in L} \alpha_{c,ij} (v_{c,j} + e_{c,ij}) \right]$$

where $\mathbf{W}_{c,v} \in \mathbb{R}^{d_{\text{node}} \times d_{\text{gt-head}}}$ is a learnable parameter for the value transformation for the $c$-th attention head. In the above, $\|$ is the concatenation operation over the outputs of the $C$ attention heads. Following the original GT, we use a gated residual connection to combine the updated node embedding with the previous node embedding:

$$\beta_i = \text{sigm}(\mathbf{W}_g [\hat{h}_i; h_i; \hat{h}_i - h_i])$$

$$h_i^{\text{new}} = (1 - \beta_i) h_i + \beta_i \hat{h}_i$$

where $\mathbf{W}_g \in \mathbb{R}^{3 * d_{\text{node}} \times 1}$ is a learnable parameter that controls the strength of the gating function.

*Edge updates via triangular multiplicative operations*. Inter-residue edge embeddings are updated using the efficient triangular multiplicative operation proposed for AlphaFold [10]. Following AlphaFold, we first calculate updates using the "outgoing" triangle edges, then the "incoming" triangle edges. We

calculate the outgoing edge transformations as follows:

$$a_{ij} = \text{sigm}(\mathbf{W}_{a,g} e_{ij}) \mathbf{W}_{a,v} e_{ij}$$

$$b_{ij} = \text{sigm}(\mathbf{W}_{b,g} e_{ij}) \mathbf{W}_{b,v} e_{ij}$$

where $\mathbf{W}_{\text{a},v}, \mathbf{W}_{\text{b},v} \in \mathbb{R}^{d_{\text{edge}} \times 2 * d_{\text{edge}}}$ are learnable parameters for the transformations of the "left" and "right" edges of each triangle, and $\mathbf{W}_{\text{a},g}, \mathbf{W}_{\text{b},g} \in \mathbb{R}^{d_{\text{edge}} \times 2 * d_{\text{edge}}}$ are learnable parameters for their respective gating functions. We calculate the outgoing triangle update for edge $ij$ as follows:

$$g_{ij}^{\text{out}} = \text{sigm}(\mathbf{W}_{c,g}^{out} e_{ij})$$

$$\hat{e}_{ij}^{\text{out}} = g_{ij}^{\text{out}} \odot \mathbf{W}_{c,v}^{\text{out}} \sum_{k \in L} (a_{ik} \odot b_{jk})$$

$$e_{ij}^{\text{new}} = e_{ij} + \hat{e}_{ij}^{\text{out}}$$

where $\mathbf{W}_{c,v}^{out} \in \mathbb{R}^{2 * d_{\text{edge}} \times d_{\text{edge}}}$ and $\mathbf{W}_{c,g}^{\text{out}} \in \mathbb{R}^{d_{\text{edge}} \times d_{\text{edge}}}$ are learnable parameters for the value and gating transformations, respectively, for the outgoing triangle update to edge $e_{ij}$. After applying the outgoing triangle update, we calculate the incoming triangle update similarly as follows:

$$g_{ij}^{\text{in}} = \text{sigm}(\mathbf{W}_{c,g}^{\text{in}} e_{ij})$$

$$\hat{e}_{ij}^{\text{in}} = g_{ij}^{\text{in}} \odot \mathbf{W}_{c,v}^{\text{in}} \sum_{k \in L} (a_{ki} \odot b_{kj})$$

$$e_{ij}^{\text{new}} = e_{ij} + \hat{e}_{ij}^{\text{in}}$$

where $\mathbf{W}_{c,v}^{\text{in}} \in \mathbb{R}^{2 * d_{\text{edge}} \times d_{\text{edge}}}$ and $\mathbf{W}_{c,g}^{\text{in}} \in \mathbb{R}^{d_{\text{edge}} \times d_{\text{edge}}}$ are learnable parameters for the value and gating transformations, respectively, for the incoming triangle update to edge $e_{ij}$. Note that $a_{ij}$ and $b_{ij}$ are calulated using separate sets of

learnable parameters for the outgoing and incoming triangle updates.

*Template incorporation via invariant point attention*. To incorporate structural template information into the node embeddings, we adopt the invariant point attention (IPA) algorithm proposed for AlphaFold [10]. The updated node and edge embeddings correspond to the single and paired representations, respectively, as described in the original implementation. The IPA layer is followed by a three-layer feedforward transition block as in the original implementation. Because our objective is to incorporate known structural data into the embedding, we omit the translational and rotational updates used in the AlphaFold structure module. We incorporate partial structure information by masking the attention between residue pairs that do not both have known coordinates. As a result, when no template information is provided, the node embeddings are updated only using the transition layers.

*Structure realization via invariant point attention*. The processed node and edge embeddings are passed to a block of three IPA layers to predict the residue atomic coordinates. Following the structure module of AlphaFold, we adopt a "residue gas" representation, in which each residue is represented by an independent coordinate frame. The coordinate frame for each residue is defined by four atoms (N, $C_\alpha$, C, and $C_\beta$) placed with ideal bond lengths and angles. We initialize the structure with all residue frames having $C_\alpha$ at the origin and task the model with predicting a series of translations and rotations that assemble the complete structure. Contrary to the AlphaFold implementation, we do not share parameters across the IPA layers, but instead learn separate parameters for each layer.

**Table 4.3:** IgFold hyperparameters

| Parameter | Value | Description |
|---|---|---|
| $d_{\text{node}}$ | 64 | Node dimension |
| $d_{\text{edge}}$ | 64 | Edge dimension |
| $d_{\text{gt-head}}$ | 32 | Graph transformer attention head dimension |
| $n_{\text{gt-head}}$ | 8 | Graph transformer attention head number |
| $d_{\text{gt-ff-dim}}$ | 256 | Graph transformer feedforward transition dimension |
| $n_{\text{gt-layers}}$ | 4 | Graph transformer layers |
| $d_{\text{ipa-temp-head-scalar}}$ | 16 | Template IPA scalar attention head dimension |
| $d_{\text{ipa-temp-head-point}}$ | 4 | Template IPA point attention head dimension |
| $n_{\text{ipa-temp-head}}$ | 8 | Template IPA attention head number |
| $d_{\text{ipa-temp-ff-dim}}$ | 64 | Template IPA feedforward transition dimension |
| $d_{\text{ipa-temp-ff-layers}}$ | 3 | Template IPA feedforward transition layers |
| $n_{\text{ipa-temp-layers}}$ | 2 | Template IPA layers |
| $d_{\text{ipa-str-head-scalar}}$ | 16 | Structure IPA scalar attention head dimension |
| $d_{\text{ipa-str-head-point}}$ | 4 | Structure IPA point attention head dimension |
| $n_{\text{ipa-str-head}}$ | 8 | Structure IPA attention head number |
| $d_{\text{ipa-str-ff-dim}}$ | 64 | Structure IPA feedforward transition dimension |
| $d_{\text{ipa-str-ff-layers}}$ | 3 | Structure IPA feedforward transition layers |
| $n_{\text{ipa-str-layers}}$ | 3 | Structure IPA layers |
| $d_{\text{ipa-err-head-scalar}}$ | 16 | Error prediction IPA scalar attention head dimension |
| $d_{\text{ipa-err-head-point}}$ | 4 | Error prediction IPA point attention head dimension |
| $n_{\text{ipa-err-head}}$ | 4 | Error prediction IPA attention head number |
| $d_{\text{ipa-err-ff-dim}}$ | 64 | Error prediction IPA feedforward transition dimension |
| $d_{\text{ipa-err-ff-layers}}$ | 3 | Error prediction IPA feedforward transition layers |
| $n_{\text{ipa-err-layers}}$ | 2 | Error prediction IPA layers |

### Training procedure

The model is trained using a combination of structure prediction and error estimation loss terms (Figure 4.16). The primary structure prediction loss is the mean-squared-error between the predicted residue frame atom coordinates (N, $C_\alpha$, C, and $C_\beta$) and the label coordinates after Kabsch alignment of all atoms. We additionally apply an L1 loss to the inter-atomic distances of the $(i, i+1)$ and $(i, i+2)$ backbone atoms to encourage proper bond lengths and

secondary structures. Finally, we use an L1 loss for error prediction, where the label error is calculated as the $C_\alpha$ deviation of each residue after Kabsch alignment of all atoms belonging to beta sheet residues. The total loss is the sum of the structure prediction loss, the inter-atomic distance loss, and the error prediction loss:

$$\begin{aligned} \text{Loss}(x_{\text{pred}}, x_{\text{label}}) = & L_{\text{coords}}(x_{\text{pred}}, x_{\text{label}}) \\ & + \text{clamp}(10 \times L_{\text{bonds}}(x_{\text{pred}}), 1) \\ & + L_{\text{error}}(x_{\text{pred}}, x_{\text{label}}) \end{aligned} \tag{4.1}$$

where $x_{\text{pred}}$ and $x_{\text{label}}$ are the predicted and experimentally determined structures, respectively. We scale the bond length loss by a factor of 10 (effectively applying the loss on the nanometer scale) and clamp losses greater than 1. Clamping the bond length loss allows the model to learn global arrangement of residues early in training then improve smaller details (e.g., bond lengths) later in training.

During training we sampled structures evenly between the SAbDab dataset [33] and the paired and unpaired synthetic structure datasets. We held out 10% of the SAbDab structures for validation during training. We used the RAdam optimizer [59] with an initial learning rate of $5 \times 10^{-4}$, with learning rate decayed on a cosine annealing schedule. We trained an ensemble of four models with different random seeds. Each model trained for $2 \times 10^6$ steps, with a batch size of one structure. Training took approximately 110 hours per model on a single A100 GPU.

**Ensemble structure prediction**

To generate a structure prediction for a given sequence, we first make predictions with each of the four ensemble models. We then use the predicted error to select a single structure from the set of four. Rather than use the average predicted error over all residues, we instead rank the structures by the $90^{th}$ percentile residue error. Typically, the $90^{th}$ percentile residue error corresponds to the challenging CDR3 loop. Thus, we effectively select the structure with the lowest risk of significant error in the CDR3 loop.

**Refinement procedure**

Predicted structures from the IgFold model undergo two stages of refinement to resolve non-realistic features and add side-chain atoms. First, the backbone structure is optimized in PyTorch using a loss function consisting of idealization terms and an RMSD constraint:

$$\begin{aligned} \text{Loss}(x_{\text{ref}}, x_{\text{pred}}) = {} & L_{\text{bond-length}}(x_{\text{ref}}) \\ & + L_{\text{bond-angle}}(x_{\text{ref}}) \\ & + L_{\text{peptide-dihedral}}(x_{\text{ref}}) \\ & + L_{\text{coords}}(x_{\text{ref}}, x_{\text{pred}}) \end{aligned} \tag{4.2}$$

where $x_{\text{ref}}$ and $x_{\text{pred}}$ are the updated and originally predicted structures, respectively. We optimize bond lengths and planar angles according to the standard values reported by Engh and Huber [60]. The peptide bond dihedral angle is optimized to be in the trans conformation. The coordinate loss term

is the same as used in model training, but instead of measuring deviation from an experimentally determined structure, it is constraining the updated structure to stay close to the original model prediction. The refinement is performed using the Adam optimizer [61] with a learning rate of 0.02 for 80 steps. Next, the structure is refined in Rosetta using the standard *ref2015* energy function [39]. Rosetta refinement progresses through three stages: (1) full-atom energy minimization, (2) side chain repacking, (3) full-atom energy minimization. Each minimization stage is performed for 100 steps with constraints to the starting coordinates.

### 4.5.2 Benchmarking antibody structure prediction methods

#### Benchmark datasets

To evaluate the performance of IgFold and other antibody structure prediction methods, we collected a set of high-quality paired and single-chain antibody structures from SAbDab. To ensure none of the deep learning models were trained using structures in the benchmark, we only used structures deposited between July 1, 2021, and September 1, 2022, (after DeepAb, ABlooper, AlphaFold, and IgFold were trained). Structures were filtered at 99% sequence identity. From these structures, we selected those with resolution greater than 3.0 Å. Finally, we removed structures with CDR H3 loops longer than 20 residues (according to Chothia numbering). These steps resulted in 197 paired and 71 single-chain antibody structures for benchmarking methods.

### Alternative methods

We compared the performance of IgFold to five alternative methods for antibody structure prediction: RepertoireBuilder, DeepAb, ABlooper, NanoNet, and AlphaFold. RepertoireBuilder structures were predicted using the web server, omitting structures released after July 1, 2021 (benchmark collection date). All of the following methods were run on identical computational hardware, with a 12-core CPU and one A100 GPU. DeepAb structures are generated using the public code repository, with five decoys per sequence as recommended in the publication [14]. ABlooper structures are predicted using the public code repository, with CDR loops built onto frameworks predicted by IgFold. We diverge from the original publication's usage of ABodyBuilder [62] for predicting framework structures because the ABodyBuilder web server does not permit omission of enough template structures to perform proper benchmarking (and no code is available). Instead, we used IgFold framework structures because the model did not produce any outliers or failures on these residues. ABlooper predictions were refined using the provided OpenMM [63] pipeline. NanoNet structures were predicted using the public code repository [17], with full-atom refinement processing performed using the provided MODELLER [64] pipeline. AlphaFold (and AlphaFold-Multimer) structures were predicted using the optimized ColabFold repository [12]. The ColabFold pipeline utilizes the model weights trained by DeepMind, but replaces the time-consuming MSA generation step with a faster search via MMseqs2 [65]. For both AlphaFold and AlphaFold-Multimer, we made predictions with all five pretrained models and selected the highest-ranking structure for

benchmarking.

## 4.6 Appendix

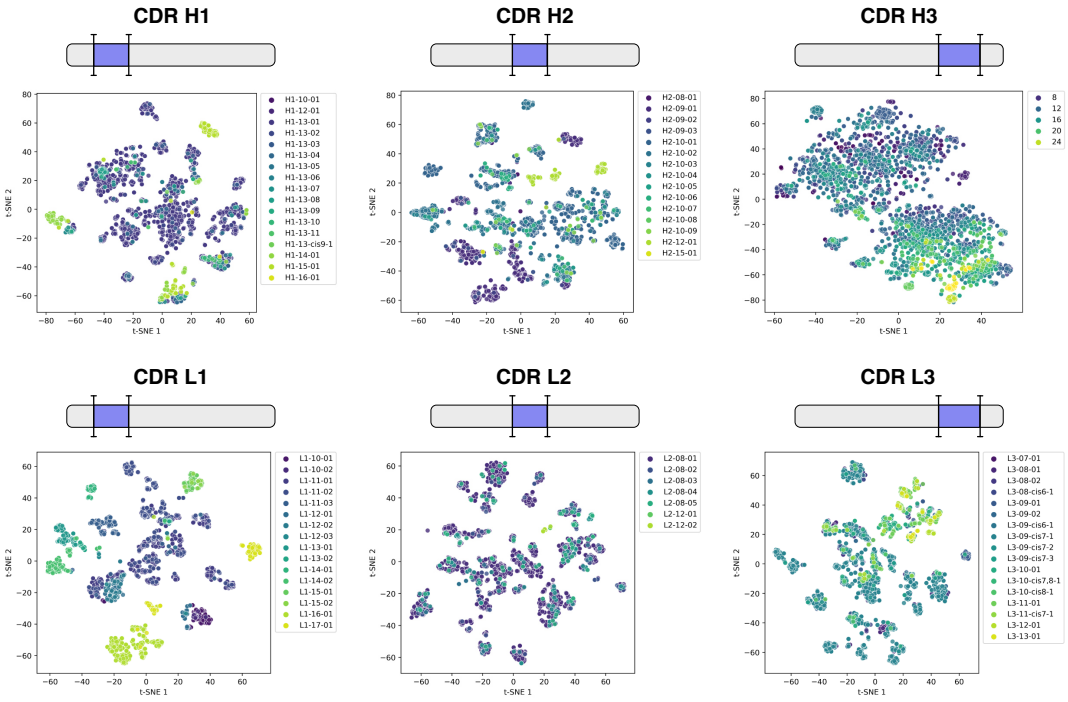


**Figure 4.15:** Visualization of AntiBERTy sequence embeddings for CDR loops

Each point corresponds to one sequence with an experimentally determined paired antibody structure. For each sequence, segments corresponding to CDR loops are extracted from the embedding and averaged to form a fixed-size representation. For each CDR loop, all representations are collected and visualized via two-dimensional t-SNE. For CDR H1-H2 and CDR L1-L3, points are colored by the canonical cluster from their respective structures. For CDR H3, points are colored according to loop length, as canonical structures are not defined.

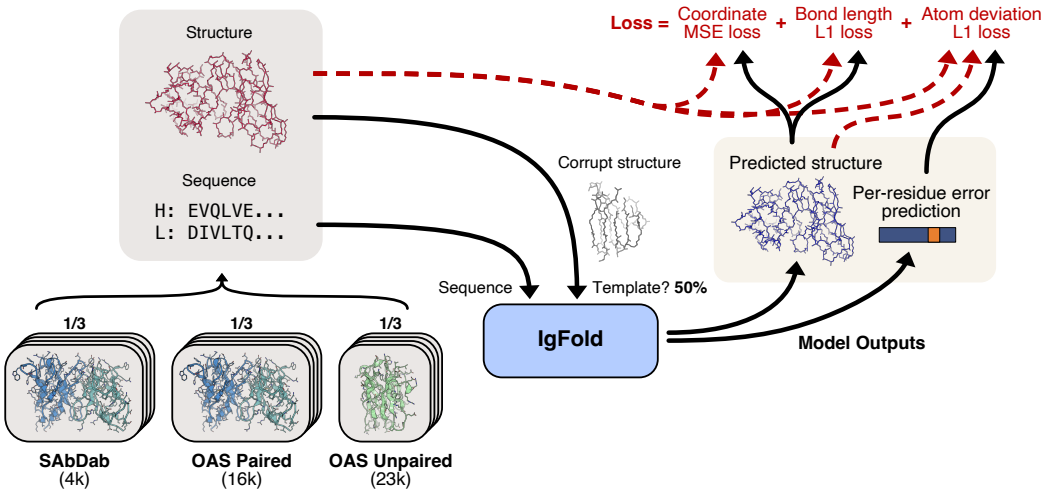


**Figure 4.16:** Diagram of IgFold training procedure

IgFold is trained using a combination of experimentaly determined structures and synthetic data from AlphaFold2. From amino-acid sequence inputs, IgFold predicts the antibody backbone structure. To enable incorporation of template structures, IgFold is provided with a partial structural solution for 50% of training examples. The model is trained using a combination of objectives for coordinate accuracy (RMSD), backbone geometry (bond/psuedo-bond lengths), and error estimation (aligned RMSD).

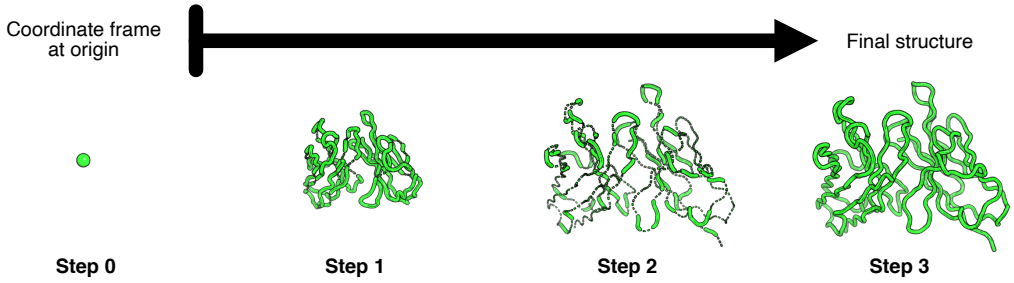


**Figure 4.17:** Stepwise prediction of paired antibody structure by invariant point attention

Predicted 3D coordinates for paired antibody structure after each invariant point attention layer, beginning from initialization of all residues at the origin. After the first layer, an initial compact structure resembling the final prediction is visible. After the second layer, the compact structure is expanded to proper scale, but with numerous chain breaks, as well as abnormal bond lengths and angles. After the third and final layer, most abnormal backbone geometries are resolved.

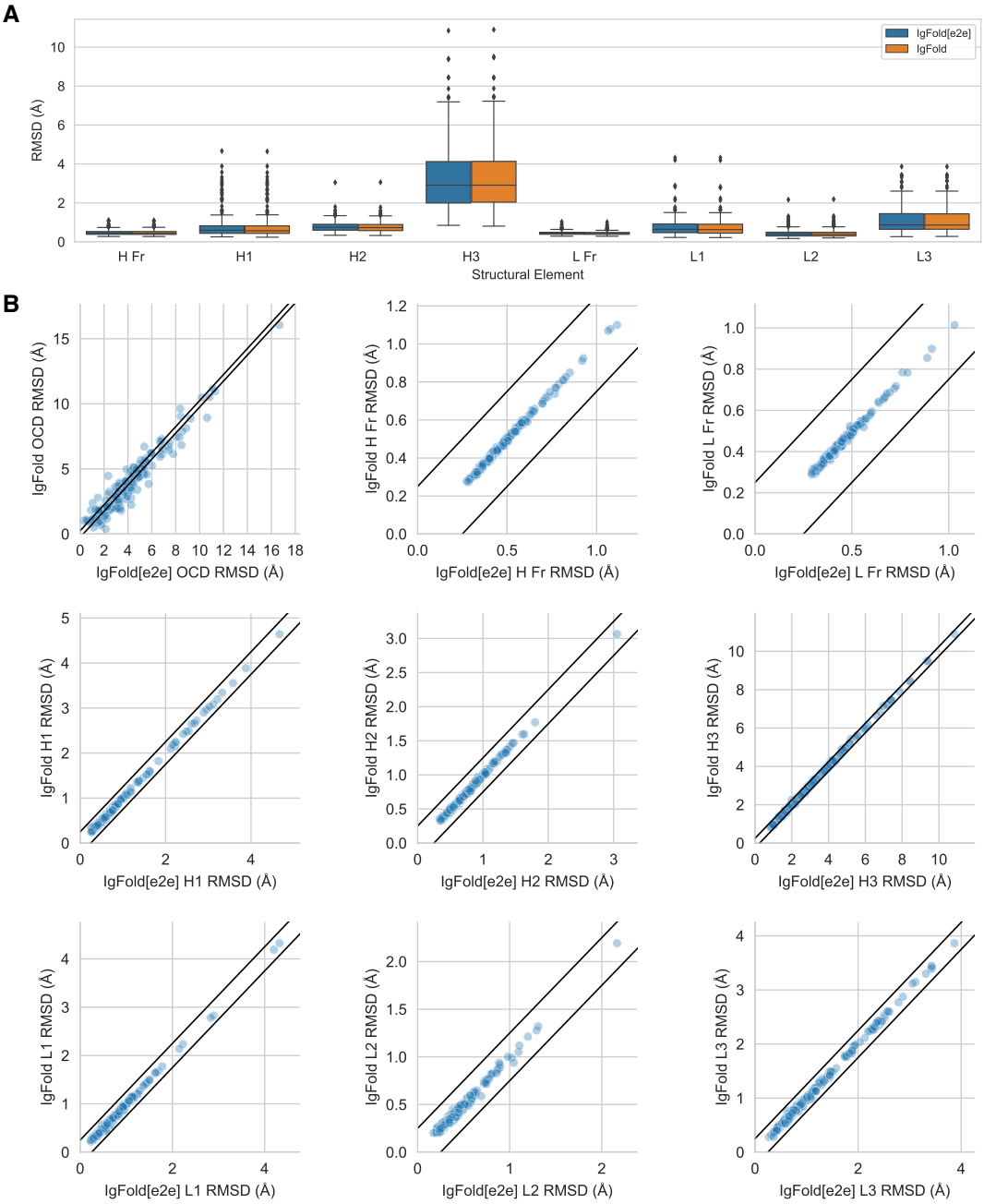


**Figure 4.18:** Impact of refinement on antibody structure prediction accuracy

Comparison of paired antibody structure prediction accuracy before and after refinement in Rosetta. (A) Summary of framework and CDR loop structure prediction RMSD for direct model predictions (IgFold[e2e]) and their refined counterparts (IgFold). (B) Direct comparison of unrefined and refined IgFold predictions for interchain orientation (OCD), framework RMSD, and CDR loop RMSD. Points within diagonal bands have differences within 0.25 units for OCD and 0.25 Å for RMSDs.

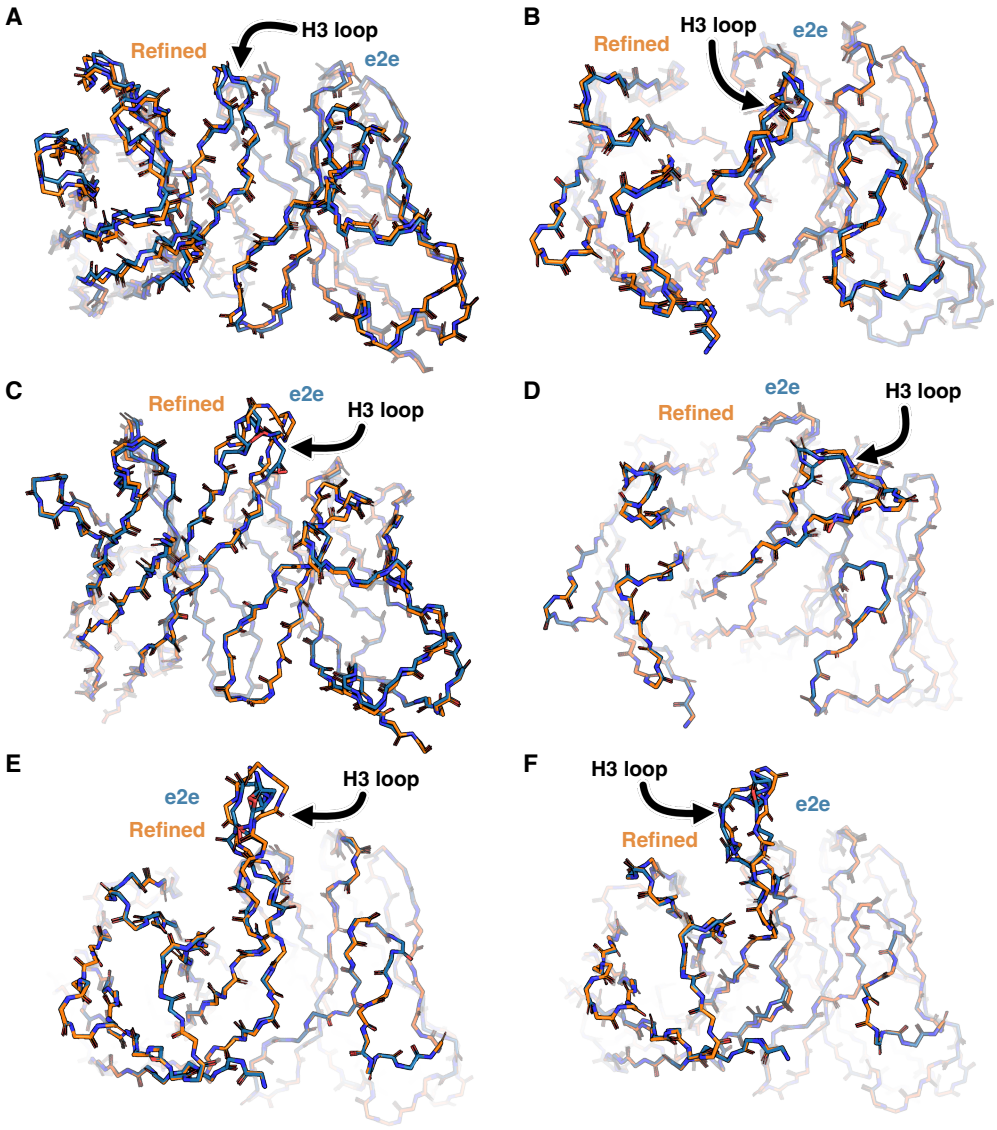


**Figure 4.19:** Effect of refinement on predicted paired antibody structures

(A-F) Comparison of predicted paired $F_V$ structures before (e2e, orange) and after (Refined, blue) refinement in Rosetta. (A) Comparison for benchmark target 7ARN, with $L_{H3} = 10$. (B) Comparison for benchmark target 7RAH, with $L_{H3} = 12$. (C) Comparison for benchmark target 7KEO, with $L_{H3} = 15$. (D) Comparison for benchmark target 7MF7, with $L_{H3} = 20$. (E) Comparison for benchmark target 7RDK, with $L_{H3} = 20$. (F) Comparison for benchmark target 7RDM, with $L_{H3} = 20$.

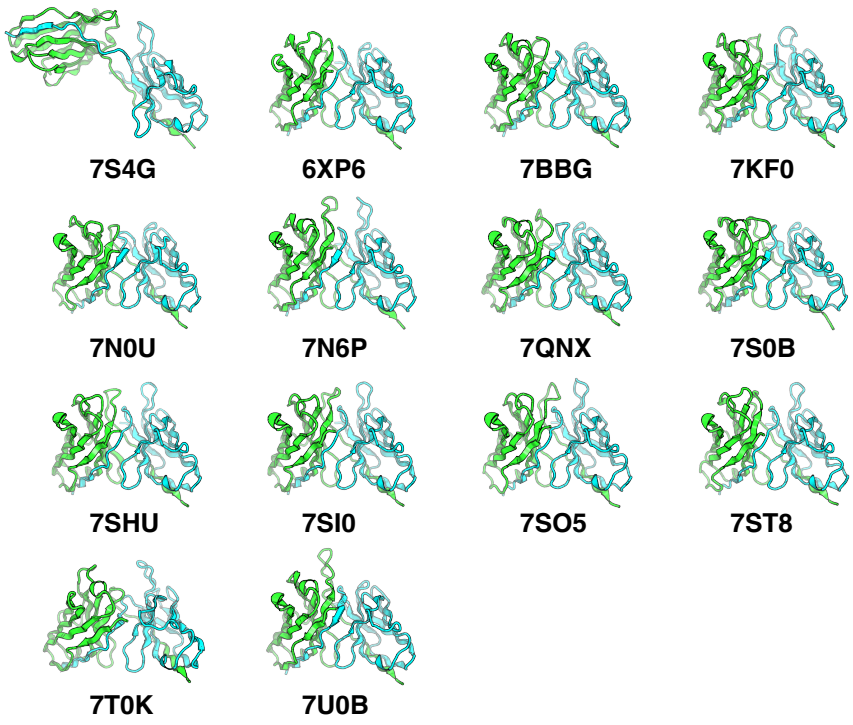


**Figure 4.20:** Strand swapping in AlphaFold predictions

AlphaFold-Multimer predicts strand swaps for fourteen of the paired antibody benchmark targets. In all cases, the C-terminal strands of the heavy and light chains are swapped.

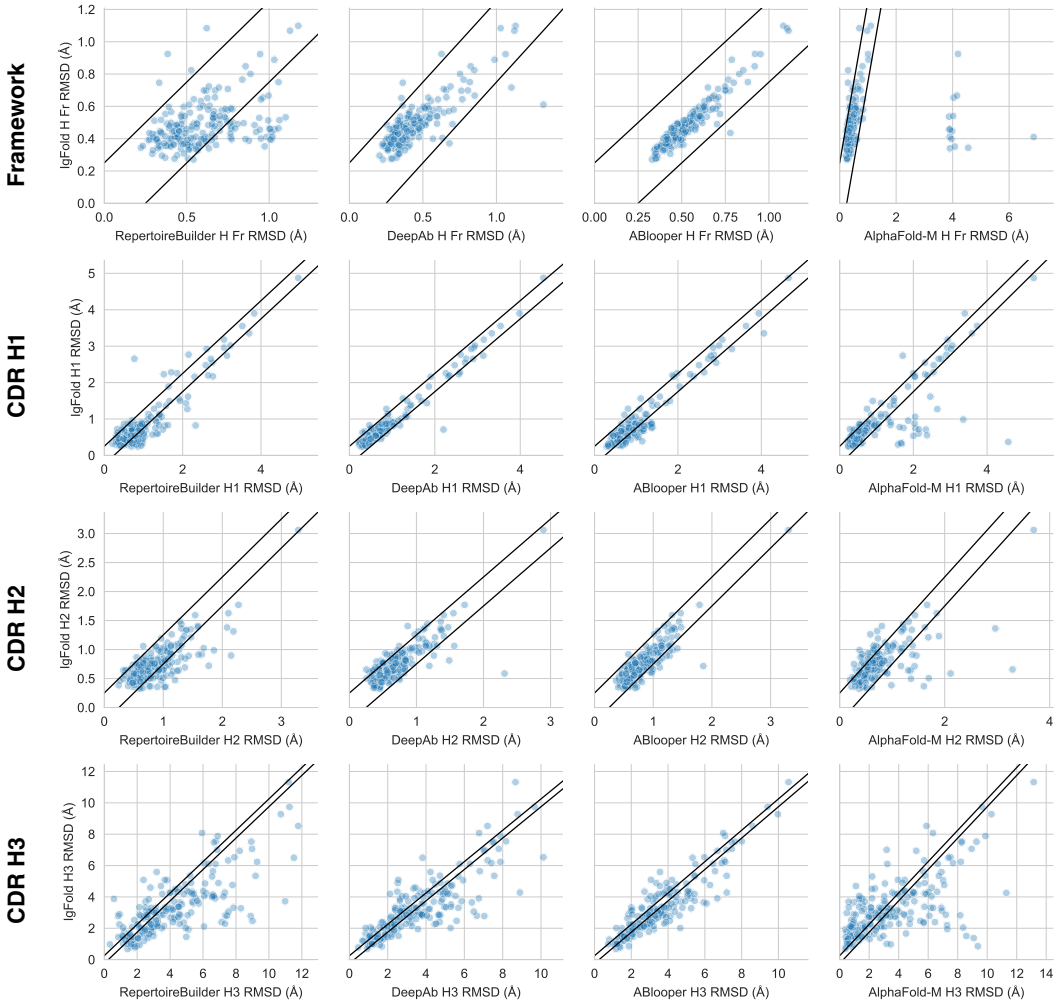


**Figure 4.21:** Comparison of methods for paired antibody heavy chain structure prediction

Scatter plots show heavy-chain RMSD metrics for benchmark structures predicted by IgFold compared to four alternative methods: RepertoireBuilder, DeepAb, ABlooper, and AlphaFold-Multimer. Each point corresponds to one benchmark target, with points between the diagonal bands having differences within 0.25 Å RMSD.

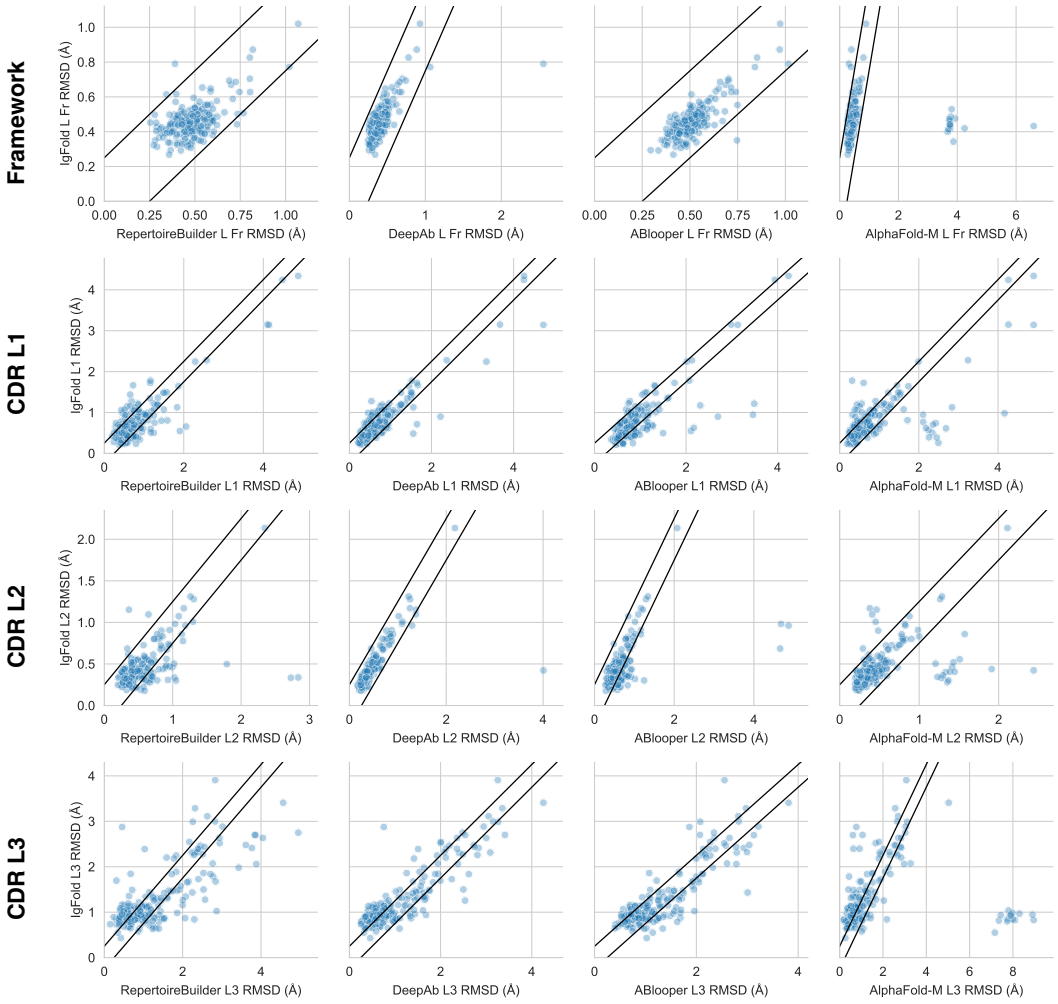


**Figure 4.22:** Comparison of methods for paired antibody light chain structure prediction

Scatter plots show light-chain RMSD metrics for benchmark structures predicted by IgFold compared to four alternative methods: RepertoireBuilder, DeepAb, ABlooper, and AlphaFold-Multimer. Each point corresponds to one benchmark target, with points between the diagonal bands having differences within 0.25 Å RMSD.

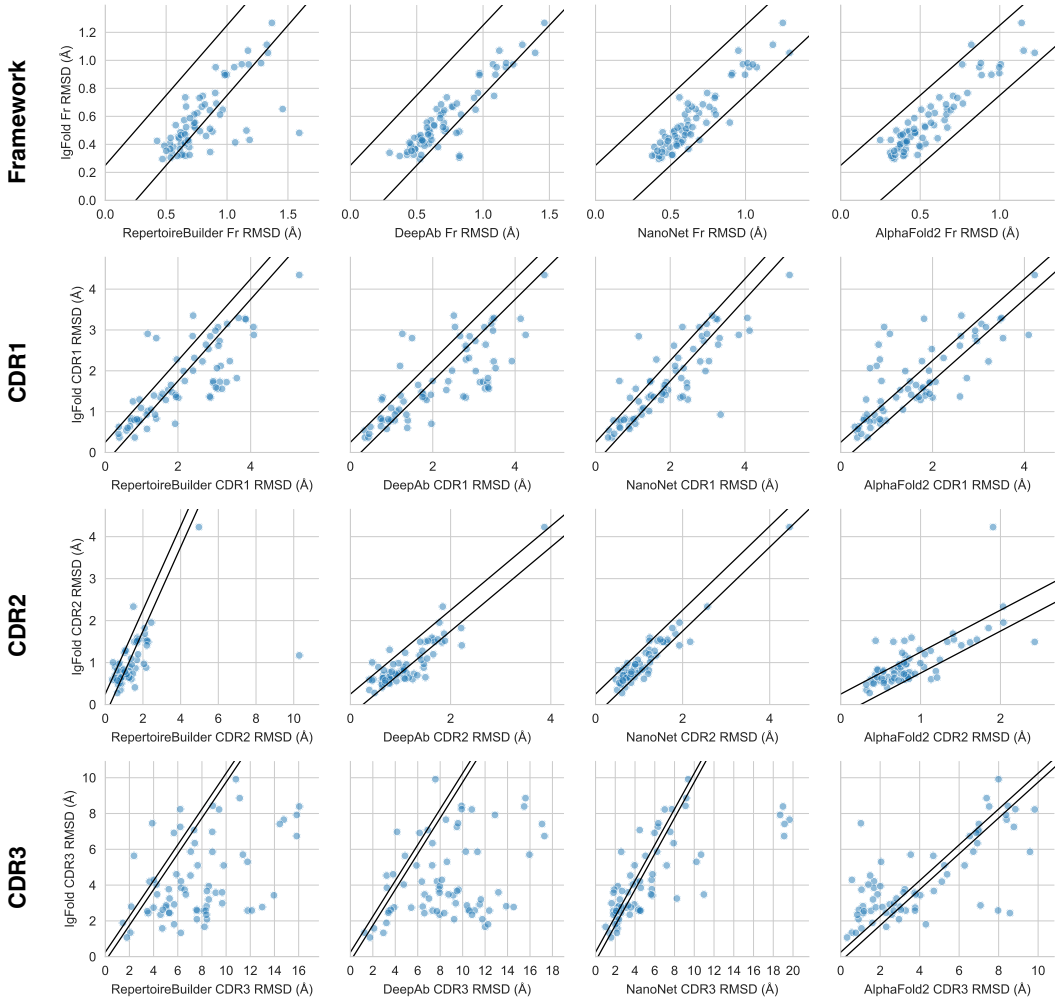


**Figure 4.23:** Comparison of methods for nanobody structure prediction

Scatter plots show nanobody RMSD metrics for benchmark structures predicted by IgFold compared to three alternative methods: RepertoireBuilder, DeepAb, and AlphaFold. Each point corresponds to one benchmark target, with points between the diagonal bands having differences within 0.25 Å RMSD.

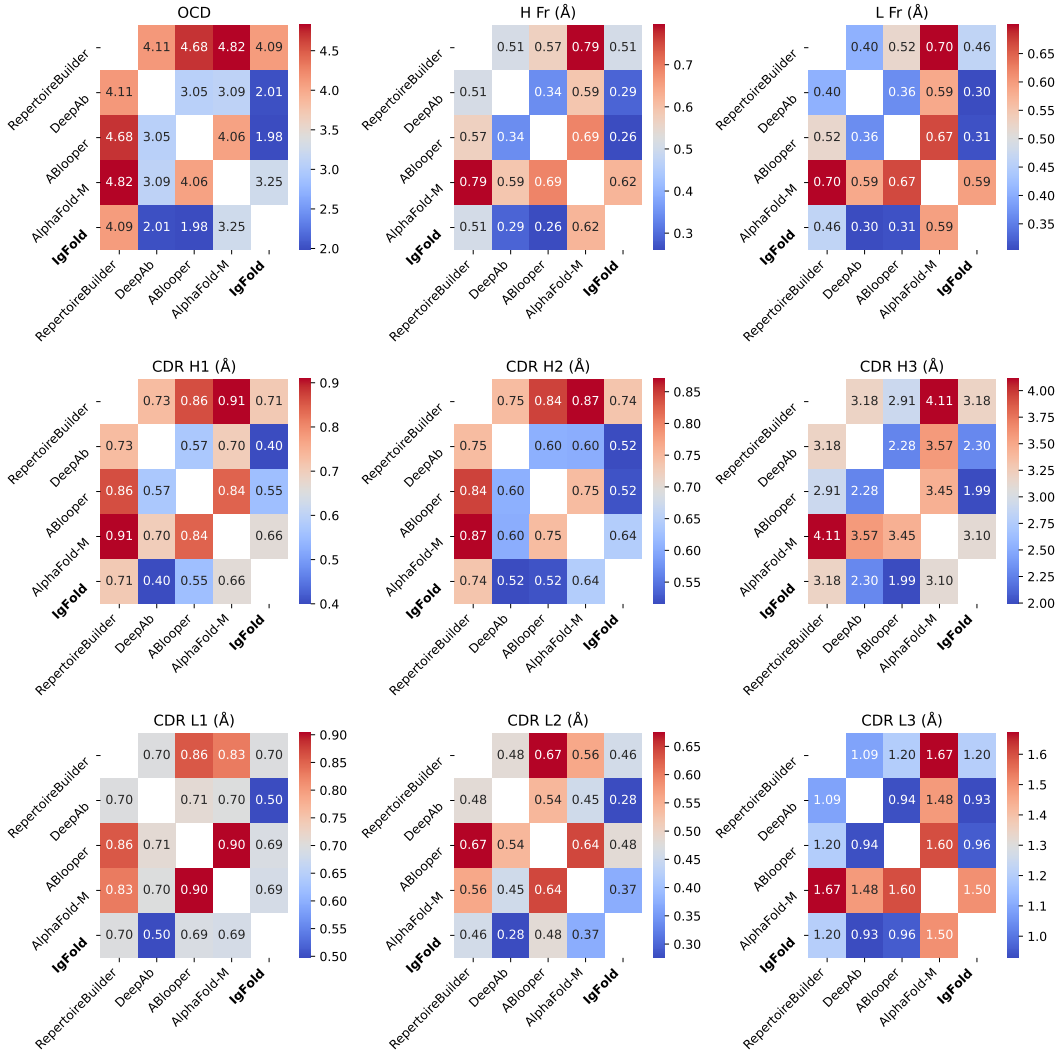


**Figure 4.24:** Similarity of predicted paired antibody structures

Pairwise analysis of similarities between predicted paired antibody structures for RepertoireBuilder, DeepAb, ABlooper, AlphaFold-Multimer, and IgFold. Each grid point corresponds to the average similarity metric (OCD or RMSD (Å)) over the full paired antibody benchmark for a given pair of methods.

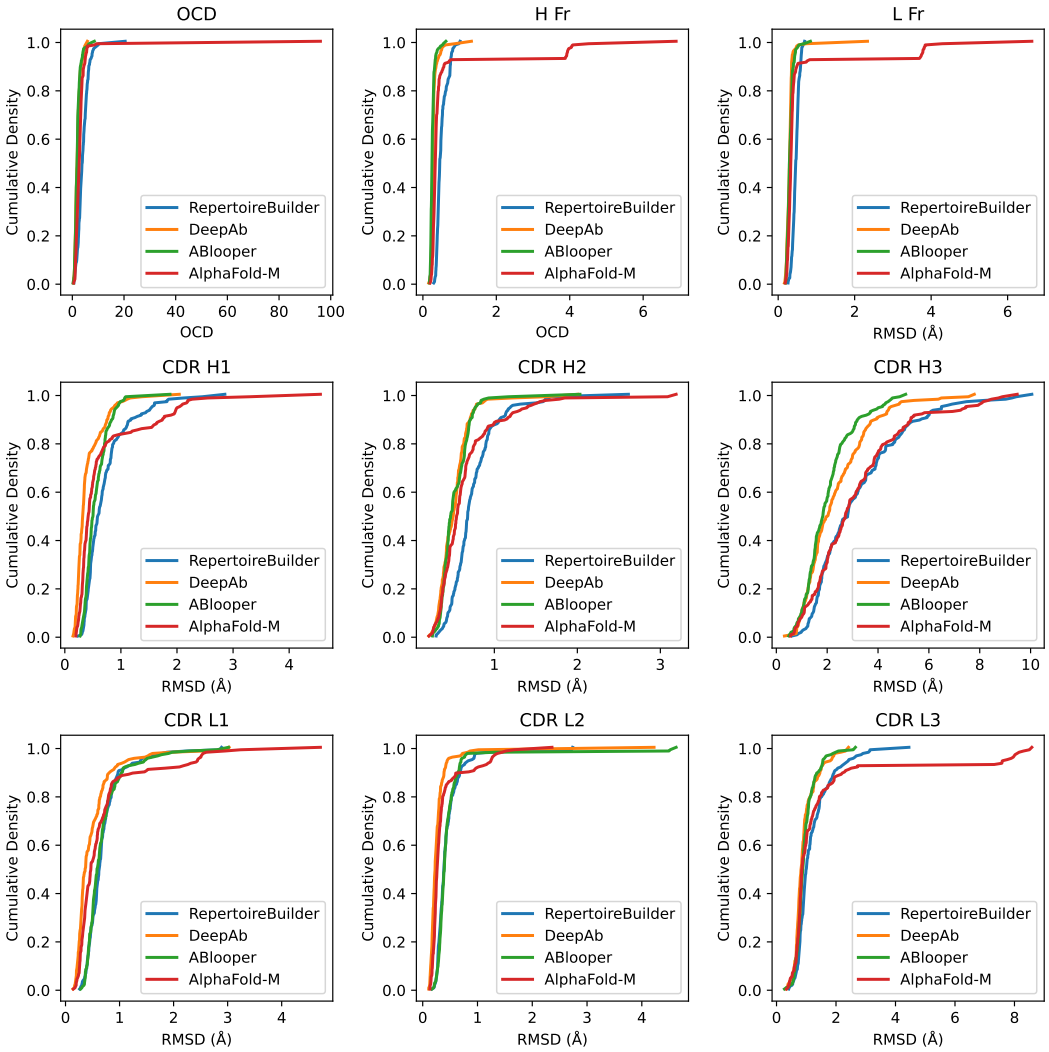


**Figure 4.25:** Similarity of IgFold-predicted paired antibody structures to alternative methods

Distribution of similarity metrics (OCD or RMSD (Å)) between IgFold and alternative methods (RepertoireBuilder, DeepAb, ABlooper, AlphaFold-Multimer). Each curve shows the cumulative density of the similarity metric for the paired benchmark targets.

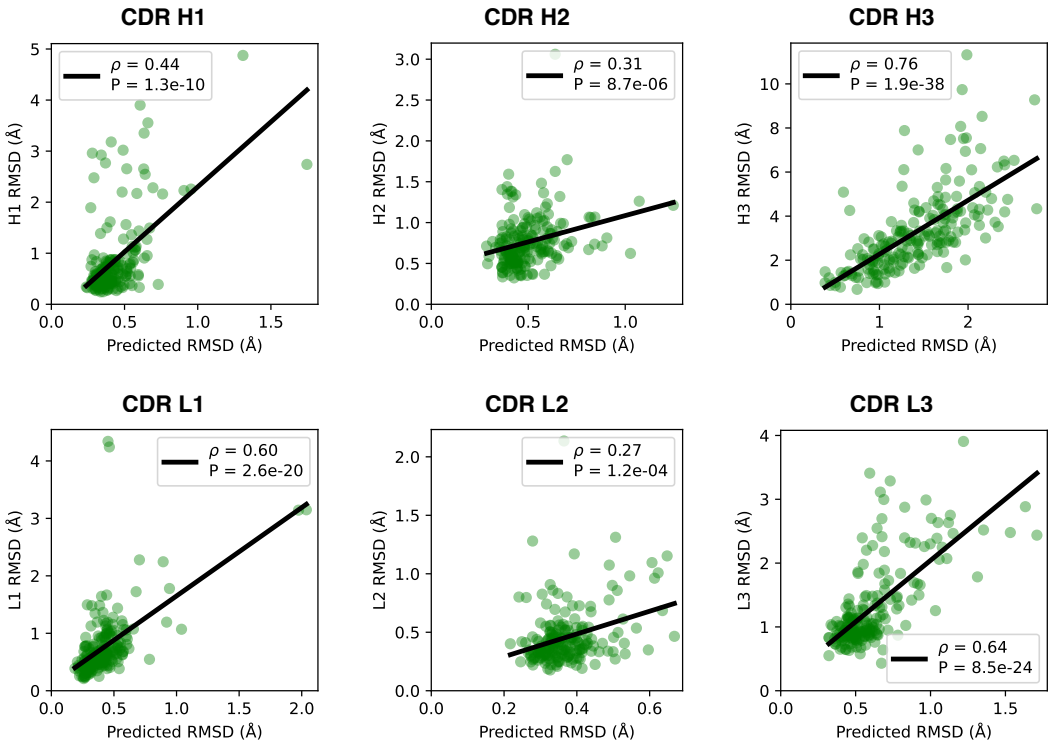


**Figure 4.26:** Estimation of paired antibody CDR loop accuracy

Average predicted error from IgFold for paired antibody CDR loops compared with the true CDR loop RMSD. Spearman values ($\rho$) linear fits are given for plots with significant correlations.

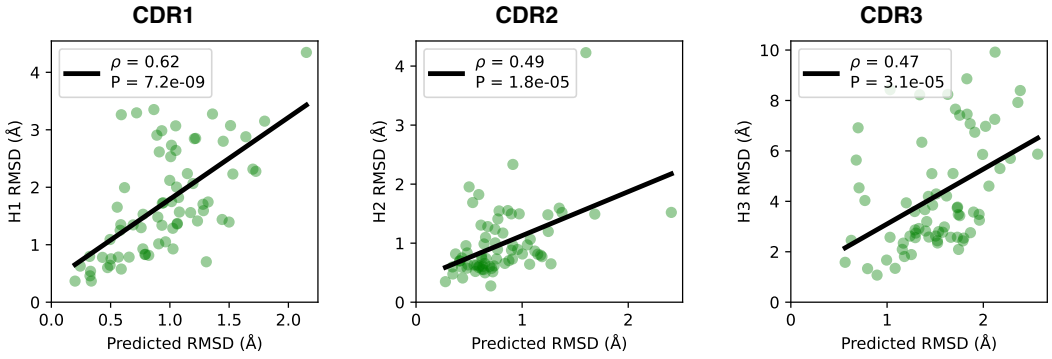


**Figure 4.27:** Estimation of nanobody CDR loop accuracy

Average predicted error from IgFold for nanobody CDR loops compared with the true CDR loop RMSD. Spearman values ($\rho$) linear fits are given for plots with significant correlations.

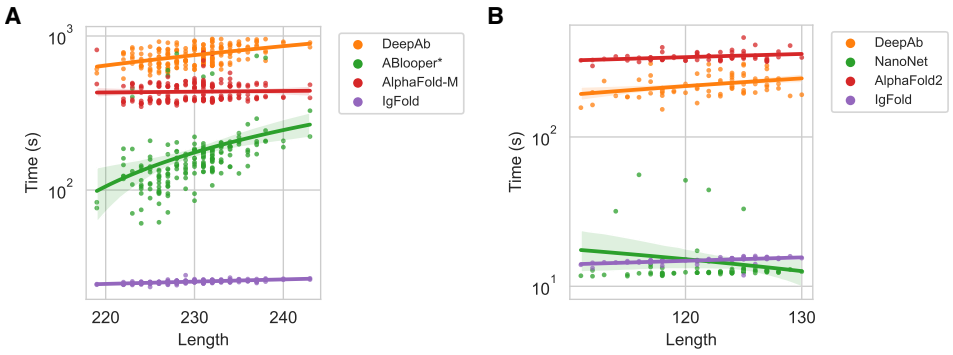


**Figure 4.28:** Relationship between sequence length and prediction runtime

(A) Per-target runtime on paired antibody structure prediction benchmark for evaluated methods. ABlooper runtimes are calculated given an IgFold-predicted framework structure, and thus represent a slight underestimation. (B) Per-target runtime on nanobody structure prediction benchmark for evaluated methods.

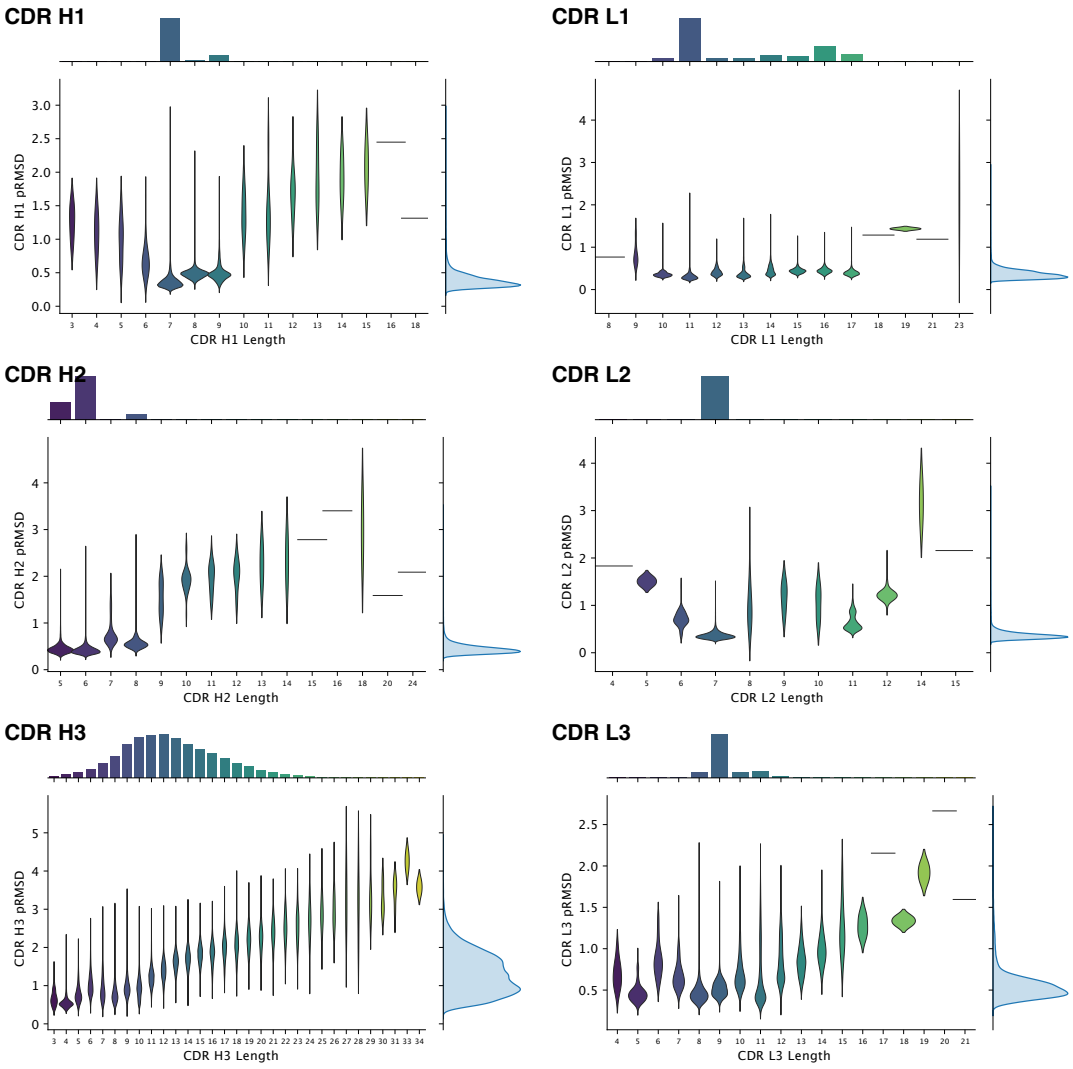


**Figure 4.29:** Analysis of large-scale OAS antibody structure predictions

Distribution of average predicted RMSD for 104,994 predicted paired antibody structures from the Observed Antibody Space.

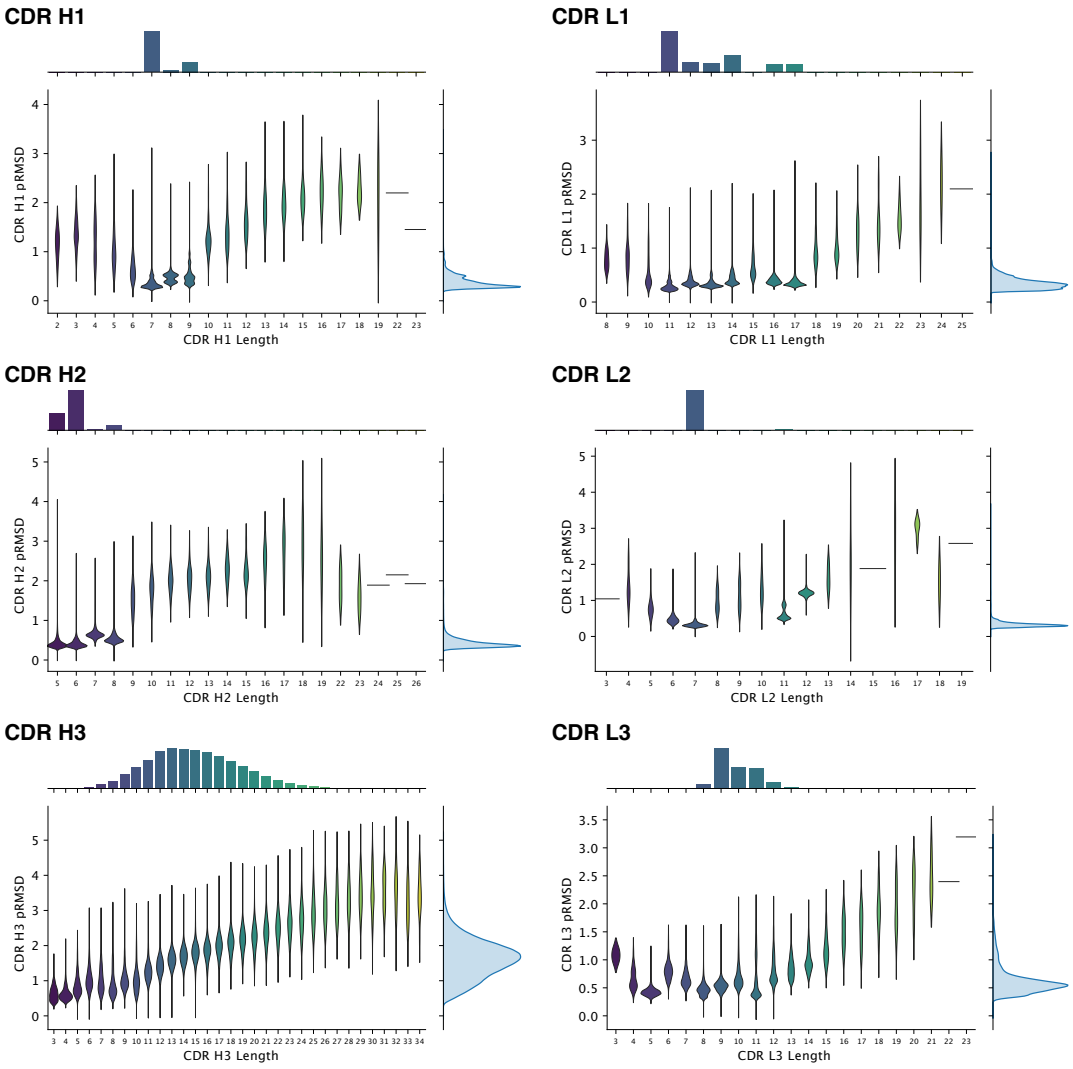


**Figure 4.30:** Analysis of large-scale human antibody structure predictions

Distribution of average predicted RMSD for 1,340,180 predicted paired antibody structures from the four unrelated human donors.

# References


[1] George Georgiou, Gregory C Ippolito, John Beausang, Christian E Busse, Hedda Wardemann, and Stephen R Quake. "The promise and challenge of high-throughput sequencing of the antibody repertoire". In: *Nature Biotechnology* 32.2 (2014), pp. 158–168.

[2] Daniel Neumeier, Alexander Yermanos, Andreas Agrafiotis, Lucia Csepregi, Tasnia Chowdhury, Roy A Ehling, Raphael Kuhn, Raphaël Brisset-Di Roberto, Mariangela Di Tacchio, Renan Antonialli, et al. "Phenotypic determinism and stochasticity in antibody repertoires of clonally expanded plasma cells". In: *bioRxiv* (2021).

[3] Sai T Reddy, Xin Ge, Aleksandr E Miklos, Randall A Hughes, Seung Hyun Kang, Kam Hon Hoi, Constantine Chrysostomou, Scott P Hunicke-Smith, Brent L Iverson, Philip W Tucker, et al. "Monoclonal antibodies isolated without screening by analyzing the variable-gene repertoire of plasma cells". In: *Nature Biotechnology* 28.9 (2010), pp. 965–969.

[4] Jared Adolf-Bryfogle, Oleks Kalyuzhniy, Michael Kubitz, Brian D Weitzner, Xiaozhen Hu, Yumiko Adachi, William R Schief, and Roland L Dunbrack Jr. "RosettaAntibodyDesign (RAbD): A general framework for computational antibody design". In: *PLOS Computational Biology* 14.4 (2018), e1006112.

[5] Jared Adolf-Bryfogle, Qifang Xu, Benjamin North, Andreas Lehmann, and Roland L Dunbrack Jr. "PyIgClassify: a database of antibody CDR structural classifications". In: *Nucleic Acids Research* 43.D1 (2015), pp. D432–D438.

[6] Juan C Almagro, Alexey Teplyakov, Jinquan Luo, Raymond W Sweet, Sreekumar Kodangattil, Francisco Hernandez-Guzman, and Gary L Gilliland. *Second Antibody Modeling Assessment (AMA-II)*. 2014.

[7] Jeffrey A Ruffolo, Carlos Guerra, Sai Pooja Mahajan, Jeremias Sulam, and Jeffrey J Gray. "Geometric potentials from deep learning improve prediction of CDR H3 loop structures". In: *Bioinformatics* 36.Supplement_1 (2020), pp. i268–i275.

[8] James Dunbar, Angelika Fuchs, Jiye Shi, and Charlotte M Deane. "ABangle: characterising the VH–VL orientation in antibodies". In: *Protein Engineering, Design and Selection* 26.10 (2013), pp. 611–620.

[9] Nicholas A Marze, Sergey Lyskov, and Jeffrey J Gray. "Improved prediction of antibody VL–VH orientation". In: *Protein Engineering, Design and Selection* 29.10 (2016), pp. 409–418.

[10] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[11] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. "Accurate prediction of protein structures and interactions using a three-track neural network". In: *Science* 373.6557 (2021), pp. 871–876.

[12] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. "ColabFold: making protein folding accessible to all". In: *Nature Methods* (2022), pp. 1–4.

[13] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew W Senior, Timothy Green, Augustin Žídek, Russell Bates, Sam Blackwell, Jason Yim, et al. "Protein complex prediction with AlphaFold-Multimer". In: *bioRxiv* (2021).

[14] Jeffrey A Ruffolo, Jeremias Sulam, and Jeffrey J Gray. "Antibody structure prediction using interpretable deep learning". In: *Patterns* 3.2 (2022), p. 100406.

[15] Brennan Abanades, Guy Georges, Alexander Bujotzek, and Charlotte M Deane. "ABlooper: Fast accurate antibody CDR loop structure prediction with accuracy estimation". In: *Bioinformatics* 38.7 (2022), pp. 1877–1880.

[16] Deniz Akpinaroglu, Jeffrey A Ruffolo, Sai Pooja Mahajan, and Jeffrey J Gray. "Improved antibody structure prediction by deep learning of side chain conformations". In: *bioRxiv* (2021).

[17] Tomer Cohen, Matan Halfon, and Dina Schneidman-Duhovny. "NanoNet: Rapid and accurate end-to-end nanobody modeling by deep learning". In: *Frontiers in Immunology* 13 (2022), p. 958584.

[18] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021).

[19] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. "ProtTrans: towards cracking the language of Life's code through self-supervised deep learning and high performance computing". In: *arXiv preprint arXiv:2007.06225* (2020).

[20] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. "Language models enable zero-shot prediction of the effects of mutations on protein function". In: *bioRxiv* (2021).

[21] Brian L Hie, Kevin K Yang, and Peter S Kim. "Evolutionary velocity with protein language models predicts evolutionary dynamics of diverse proteins". In: *Cell Systems* (2022).

[22] Jeffrey A Ruffolo, Jeffrey J Gray, and Jeremias Sulam. "Deciphering antibody affinity maturation with language models and weakly supervised learning". In: *arXiv preprint arXiv:2112.07782* (2021).

[23] Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Charlotte Rochereau, George M Church, Peter Karl Sorger, and Mohammed N AlQuraishi. "Single-sequence protein structure prediction using language models from deep learning". In: *bioRxiv* (2021).

[24] Yiyu Hong, Juyong Lee, and Junsu Ko. "A-Prot: Protein structure modeling using MSA transformer". In: *BMC Bioinformatics* 23.1 (2022), pp. 1–11.

[25] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. "Large language models generate functional protein sequences across diverse families". In: *Nature Biotechnology* (2023), pp. 1–8.

[26] Jinwoo Leem, Laura S Mitchell, James HR Farmery, Justin Barton, and Jacob D Galson. "Deciphering the language of antibodies using self-supervised learning". In: *bioRxiv* (2021).

[27] Tobias H Olsen, Iain H Moal, and Charlotte M Deane. "AbLang: an antibody language model for completing antibody sequences". In: *Bioinformatics Advances* 2.1 (2022), vbac046.

[28] David Prihoda, Jad Maamary, Andrew Waight, Veronica Juan, Laurence Fayadat-Dilman, Daniel Svozil, and Danny A Bitton. "BioPhi: a platform for antibody design, humanization, and humanness evaluation based on natural antibody repertoires and deep learning". In: *MAbs*. Vol. 14. 1. Taylor & Francis. 2022, p. 2020203.

[29] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. "Protein design and variant prediction using autoregressive generative models". In: *Nature Communications* 12.1 (2021), pp. 1–11.

[30] Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. "Generative Language Modeling for Antibody Design". In: *bioRxiv* (2021).

[31] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. "Language models of protein sequences at the scale of evolution enable accurate structure prediction". In: *bioRxiv* (2022).

[32] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. "High-resolution de novo structure prediction from primary sequence". In: *bioRxiv* (2022).

[33] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. "SAbDab: the structural antibody database". In: *Nucleic Acids Research* 42.D1 (2014), pp. D1140–D1146.

[34] Aleksandr Kovaltsuk, Jinwoo Leem, Sebastian Kelm, James Snowden, Charlotte M Deane, and Konrad Krawczyk. "Observed Antibody Space: a resource for data mining next-generation sequencing of antibody repertoires". In: *The Journal of Immunology* 201.8 (2018), pp. 2502–2509.

[35] Mohammed AlQuraishi. "Machine learning in protein structure prediction". In: *Current Opinion in Chemical Biology* 65 (2021), pp. 1–8.

[36] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. "Transformer protein language models are unsupervised structure learners". In: *International Conference on Learning Representations*. 2020.

[37] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. "Masked label prediction: Unified message passing model for semi-supervised classification". In: *arXiv preprint arXiv:2009.03509* (2020).

[38] Martin Steinegger and Johannes Söding. "Clustering huge protein sequence sets in linear time". In: *Nature Communications* 9.1 (2018), pp. 1–8.

[39] Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. "The Rosetta all-atom energy function for macromolecular modeling and design". In: *Journal of Chemical Theory and Computation* 13.6 (2017), pp. 3031–3048.

[40] Dimitri Schritt, Songling Li, John Rozewicki, Kazutaka Katoh, Kazuo Yamashita, Wayne Volkmuth, Guy Cavet, and Daron M Standley. "Repertoire Builder: high-throughput structural modeling of B and T cell receptors". In: *Molecular Systems Design & Engineering* 4.4 (2019), pp. 761–768.

[41] Frauke Muecksch, Yiska Weisblum, Christopher O Barnes, Fabian Schmidt, Dennis Schaefer-Babajew, Zijun Wang, Julio CC Lorenzi, Andrew I Flyak, Andrew T DeLaitsch, Kathryn E Huey-Tubman, et al. "Affinity maturation of SARS-CoV-2 neutralizing antibodies confers potency, breadth, and resilience to viral escape mutations". In: *Immunity* 54.8 (2021), pp. 1853–1868.

[42] Dora Pinto, Maximilian M Sauer, Nadine Czudnochowski, Jun Siong Low, M Alejandra Tortorici, Michael P Housley, Julia Noack, Alexandra C Walls, John E Bowen, Barbara Guarino, et al. "Broad betacoronavirus neutralization by a stem helix–specific human antibody". In: *Science* 373.6559 (2021), pp. 1109–1116.

[43] Femke Van Bockstaele, Josefin-Beate Holz, and Hilde Revets. "The development of nanobodies for therapeutic applications." In: *Current Opinion in Investigational Drugs* 10.11 (2009), pp. 1212–1224.

[44] Aroop Sircar, Kayode A Sanni, Jiye Shi, and Jeffrey J Gray. "Analysis and modeling of the variable region of camelid single-domain antibodies". In: *The Journal of Immunology* 186.11 (2011), pp. 6357–6367.

[45] June Ereño-Orbea, Xianglei Liu, Taylor Sicard, Iga Kucharska, Wei Li, Dorota Borovsky, Hong Cui, Yang Feng, Dimiter S Dimitrov, and Jean-Philippe Julien. "Structural details of monoclonal antibody m971 recognition of the membrane-proximal domain of CD22". In: *Journal of Biological Chemistry* 297.2 (2021).

[46] Claudia A Jette, Alexander A Cohen, Priyanthi NP Gnanapragasam, Frauke Muecksch, Yu E Lee, Kathryn E Huey-Tubman, Fabian Schmidt, Theodora Hatziioannou, Paul D Bieniasz, Michel C Nussenzweig, et al. "Broad cross-reactivity across sarbecoviruses exhibited by a subset of COVID-19 donor-derived neutralizing antibodies". In: *Cell Reports* 36.13 (2021), p. 109760.

[47] J Schilz, U Binder, L Friedrich, M Gebauer, C Lutz, M Schlapschy, A Schiefner, and A Skerra. "Molecular recognition of structurally disordered Pro/Ala-rich sequences (PAS) by antibodies involves an Ala residue at the hot spot of the epitope". In: *Journal of Molecular Biology* 433.18 (2021), p. 167113.

[48] Juan C Almagro, Martha Pedraza-Escalona, Hugo Iván Arrieta, and Sonia Mayra Pérez-Tapia. "Phage display libraries for antibody therapeutic discovery and development". In: *Antibodies* 8.3 (2019), p. 44.

[49] Rahel Frick, Lene S Høydahl, Jan Petersen, M Fleur Du Pré, Shraddha Kumari, Grete Berntsen, Alisa E Dewan, Jeliazko R Jeliazkov, Kristin S Gunnarsen, Terje Frigstad, et al. "A high-affinity human TCR-like antibody detects celiac disease gluten peptide–MHC complexes and inhibits T cell activation". In: *Science Immunology* 6.62 (2021), eabg4925.

[50] Matthew IJ Raybould, Claire Marks, Aleksandr Kovaltsuk, Alan P Lewis, Jiye Shi, and Charlotte M Deane. "Public Baseline and shared response structures support the theory of antibody repertoire functional commonality". In: *PLOS Computational Biology* 17.3 (2021), e1008781.

[51] Sarah A. Robinson, Matthew I. J. Raybould, Constantin Schneider, Wing Ki Wong, Claire Marks, and Charlotte M. Deane. "Epitope profiling using computational structural modelling demonstrated on coronavirus-binding antibodies". In: *PLOS Computational Biology* 17.12 (2021), pp. 1–20. DOI: 10.1371/journal.pcbi.1009675. URL: https://doi.org/10.1371/journal.pcbi.1009675.

[52] David B Jaffe, Payam Shahi, Bruce A Adams, Ashley M Chrisman, Peter M Finnegan, Nandhini Raman, Ariel E Royall, FuNien Tsai, Thomas Vollbrecht, Daniel S Reyes, et al. "Functional antibodies exhibit light chain coherence". In: *bioRxiv* (2022).

[53] Wing Ki Wong, Sarah A Robinson, Alexander Bujotzek, Guy Georges, Alan P Lewis, Jiye Shi, James Snowden, Bruck Taddese, and Charlotte M Deane. "Ab-Ligity: identifying sequence-dissimilar antibodies that bind to the same epitope". In: *MAbs*. Vol. 13. 1. Taylor & Francis. 2021, p. 1873478.

[54] Aroop Sircar and Jeffrey J Gray. "SnugDock: paratope structural optimization during antibody-antigen docking compensates for errors in antibody homology models". In: *PLOS Computational Biology* 6.1 (2010), e1000644.

[55] Jeliazko R Jeliazkov, Rahel Frick, Jing Zhou, and Jeffrey J Gray. "Robustification of RosettaAntibody and Rosetta SnugDock". In: *PLOS One* 16.3 (2021), e0234282.

[56] Ameya Harmalkar, Sai Pooja Mahajan, and Jeffrey J Gray. "Induced fit with replica exchange improves protein complex structure prediction". In: *PLoS computational biology* 18.6 (2022), e1010124.

[57] Christoffer Norn, Basile IM Wicky, David Juergens, Sirui Liu, David Kim, Doug Tischer, Brian Koepnick, Ivan Anishchenko, Foldit Players, David Baker, et al. "Protein sequence design by conformational landscape optimization". In: *Proceedings of the National Academy of Sciences* 118.11 (2021), e2017228118.

[58] Jue Wang, Sidney Lisanza, David Juergens, Doug Tischer, Ivan Anishchenko, Minkyung Baek, Joseph L Watson, Jung Ho Chun, Lukas F Milles, Justas Dauparas, et al. "Deep learning methods for designing proteins scaffolding functional sites". In: *bioRxiv* (2021).

[59] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. "On the variance of the adaptive learning rate and beyond". In: *arXiv preprint arXiv:1908.03265* (2019).

[60] Richard A Engh and Robert Huber. "Accurate bond and angle parameters for X-ray protein structure refinement". In: *Acta Crystallographica Section A: Foundations of Crystallography* 47.4 (1991), pp. 392–400.

[61] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[62] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Claire Marks, Jaroslaw Nowak, Cristian Regep, Guy Georges, Sebastian Kelm, Bojana Popovic, and Charlotte M Deane. "SAbPred: a structure-based antibody prediction server". In: *Nucleic Acids Research* 44.W1 (2016), W474–W478.

[63] Peter Eastman, Jason Swails, John D Chodera, Robert T McGibbon, Yutong Zhao, Kyle A Beauchamp, Lee-Ping Wang, Andrew C Simmonett, Matthew P Harrigan, Chaya D Stern, et al. "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics". In: *PLOS Computational Biology* 13.7 (2017), e1005659.

[64] Narayanan Eswar, David Eramian, Ben Webb, Min-Yi Shen, and Andrej Sali. "Protein structure modeling with MODELLER". In: *Structural Proteomics*. Springer, 2008, pp. 145–159.

[65] Martin Steinegger and Johannes Söding. "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets". In: *Nature Biotechnology* 35.11 (2017), pp. 1026–1028.

# Chapter 5

# Exploring the boundaries of protein language models

Adapted from Erik Nijkamp*, Jeffrey Ruffolo*, Eli N Weinstein, Nikhil Naik, and Ali Madani. "Progen2: exploring the boundaries of protein language models". *arXiv* (2022). Reproduced with permission. *Joint first authors.

## 5.1 Abstract

Attention-based models trained on protein sequences have demonstrated incredible success at classification and generation tasks relevant for artificial intelligence-driven protein design. However, we lack a sufficient understanding of how very large-scale models and data play a role in effective protein model development. We introduce a suite of protein language models, named ProGen2, that are scaled up to 6.4B parameters and trained on different sequence datasets drawn from over a billion proteins from genomic,

metagenomic, and immune repertoire databases. ProGen2 models show state-of-the-art performance in capturing the distribution of observed evolutionary sequences, generating novel viable sequences, and predicting protein fitness without additional finetuning. As large model sizes and raw numbers of protein sequences continue to become more widely accessible, our results suggest that a growing emphasis needs to be placed on the data distribution provided to a protein sequence model. Our models and code are open-sourced for widespread adoption in protein engineering.

## 5.2 Introduction

Proteins are the workhorse of life – performing essential and versatile functions critical to sustain human health and the environment. Engineering proteins for our desired purposes enables use-cases in industries across pharmaceuticals, agriculture, specialty chemicals, and fuel. Current tools for protein engineering are limited and, as a consequence, mainly rely on directed evolution [1], a process of stochastically mutating a starting/wild-type sequence, measuring each variant, and iterating until sufficiently optimized for improved function, also referred to as fitness. Nature as an underlying generative process has yielded a rich, complex distribution of proteins. Due to exponentially-broken barriers in DNA sequencing, we now collect natural sequences at a previously-unimaginable pace. In parallel, we have seen machine learning models perform exceedingly well at capturing data distributions of images and natural language [2, 3]. In particular, the transformer [4] has proven to be a powerful language model and can serve as a universal

computation engine [5] across data modalities.

Language modeling tries to capture the notion that some sequences are more likely than others by density estimation. For large language models (LLMs), transformer models equipped with self-attention mechanisms [6] have shown to be particularly well suited to capture dependency among sequence elements while being capable to scale vast amounts of model parameters [7, 8]. In this work, we adopt causal LLMs in the form of auto-regressive decoders for the modeling of proteins. The raw amino acid sequences which constitute a protein are considered as observed sequences for the maximum likelihood-based learning. The problem of conditional protein generation is naturally cast as a next-token prediction task. Specifically, few-shot learning [3] models tasks as auto-regressive sampling conditional on a small set of examples (or shots). Notably, LLMs possess the capacity to solve the intended task by increasing the number of parameters without task-specific finetuning of the model. These few-shot abilities appear to emerge under certain parameter thresholds [9], which motivates the exploration of such capabilities for protein engineering.

Methods for generating protein sequences that are functional and have desired properties have recently seen tremendous progress. Simple, traditional methods that leverage multiple sequence alignments of similar proteins, such as ancestral sequence reconstruction [10], have demonstrated the ability to generate useful proteins but are limited in scope. A host of statistical and machine learning techniques exist to access a larger sequence space. Most still train on a fixed protein family to capture co-evolutionary signals present within a set

of homologous sequences – ranging from direct coupling analysis techniques [11] to generative adversarial networks [12]. More versatile models trained on unaligned and unrelated sequences have emerged [13] for functional sequence design. Language models, in particular, provide a powerful architecture to learn from large sets of amino acid sequences across families for the purpose of generating diverse, realistic proteins [14, 15]. Sequences generated by protein language models (PLMs) are typically predicted to adopt well-folded structures, despite diverging significantly in sequence space. PLMs can be further focused on specific families of interest by finetuning on a subset of relevant proteins. In prior work, finetuning the ProGen model on a set of lysozyme families yielded proteins retaining functional behavior, and even rivaling that of a natural hen egg white lysozyme [16]. Similar strategies have been employed for domain-specific PLMs, such as the antibody-specific IgLM model [17]. By conditioning on chain type and species-of-origin, IgLM is capable of generating diverse sets of antibodies resembling those of natural immune repertoires.

Understanding the functional effects of sequence mutations is critical for the rational design of proteins. Methods for predicting such effects typically fit into one of two categories: family-specific models trained on aligned sequences or universal models trained on unaligned sequences. Models based on alignments of sequences [18, 19, 20] face several key challenges limiting their application to protein engineering tasks. First, for proteins with few evolutionary neighbors, the MSA is likely to be shallow and contain little information about functional constraints. Second, for some families of proteins (such as

antibodies), there are many sequences available, but they are non-trivial to align. Finally, evaluation of novel variants requires that new sequences be aligned to the MSA used for training; this can be challenging in cases with significant insertions or deletions (indels). These limitations prompted the development of fitness predictors based unaligned sets of sequences, particularly transformer models trained on large databases of protein sequences. ESM-1v [21] tasks a transformer encoder model trained via masked-language modeling with estimating heuristic likelihood of mutations relative to the wild type sequences. Autoregressive PLMs have also been applied to fitness prediction [13]. These models are intrinsically capable of modeling indels, as well as epistatic mutations. The RITA family of models [22] demonstrated that not only do autoregressive PLMs effectively estimate protein fitness, but performance can be further improved by scaling model capacity. Tranception [23] demonstrated that combining autoregressive language models with retrieval [24] capabilities provides a means of enhancing a generalist model with family-specific information from MSAs at inference.

In this work, we perform a study on the effect of very large-scale models and data. We train a suite of models ranging from 151M to 6.4B parameters (one of the largest published for a single protein transformer) on different datasets collectively totaling 1B protein sequences from genomic, metagenomic, and immune repertoire databases. We analyze the generations from universal and family-specific models through predicted structural and biophysical properties. Finally, we examine fitness prediction on existing experimental datasets which motivate hypotheses on the role of data distribution

**Table 5.1:** Model performance on held-out test sets

| Model Name | Parameters | Test-max90 (ppl) | Test-max50 (ppl) |
|---|---|---|---|
| ProGen2-small | 151M | 12.9 | 15.0 |
| ProGen2-medium | 764M | 11.2 | 14.3 |
| ProGen2-large | 2.7B | 11.1 | 14.4 |
| ProGen2-xlarge | 6.4B | 9.9 | 13.9 |

Increasing number of parameters allows the model to better capture the distribution of observed evolutionary sequences. Performance is measured as the perplexity of held-out test sequences at various maximum sequence identity thresholds, i.e. test-max50 is more difficult and out-of-distribution.

and alignment in protein language modeling.

## 5.3 Results

### 5.3.1 Capturing the distribution of observed proteins

We first evaluate the capacity of ProGen2 to capture the distribution of natural sequences. In particular, we focused on its ability to predict unobserved natural sequences, quantifying performance in terms of perplexity on a heldout test set. Perplexity can be intuitively interpreted as the average number of residues considered by the model at each position. As such, a model that has better captured the protein sequence data distribution should produce lower perplexity values. Indeed, we find that larger models yield substantially lower perplexities, consistent with the idea that, despite massive model size, we are far from the overfitting regime (Table 5.1).

For a sequence $x = (x_1, x_2, \ldots, x_n)$ of $n$ tokens and a language model $p$, the perplexity is calculated as

$$ppl(x) = \exp -\frac{1}{n} \sum_{i=1}^{n} \ln p(x_i)$$

We report the average perplexity over the held-out partitions of the datasets. We caution, however, that these results only reflect the capacity of the model to capture the training distribution from which the data were drawn, not necessarily relevant measures of molecular fitness.

### 5.3.2 Protein sequence generation

Given the capacity of the ProGen2 family of models for capturing the distribution of observed evolutionary sequences, we next assessed the ability of the models to generate novel sequences. We evaluated sequence generation in three settings: 1) universal protein generation from pretraining, 2) fold-specific generation after finetuning, and 3) antibody generation after domain-specific pretraining.

#### Pretrained models generate diverse protein sequences

Prior work has demonstrated that sequences generated by PLMs can adopt a wide variety of folds, often with significant deviation in sequence from observed proteins [14, 15]. To assess the generative capacity of ProGen2 models, we generated 5,000 sequences with the ProGen2-xlarge model. The three-dimensional structure of each sequence was predicted using AlphaFold2 [25]. For each structure, we identified the most structurally similar natural protein the in the PDB [26] using Foldseek [27]. In Figure 5.1, we show the relationship between structural similarity to natural proteins (TMscore) and AlphaFold2 prediction confidence (pLDDT). The majority of structures were confidently predicted (median pLDDT of 90.0) and had structural homologs

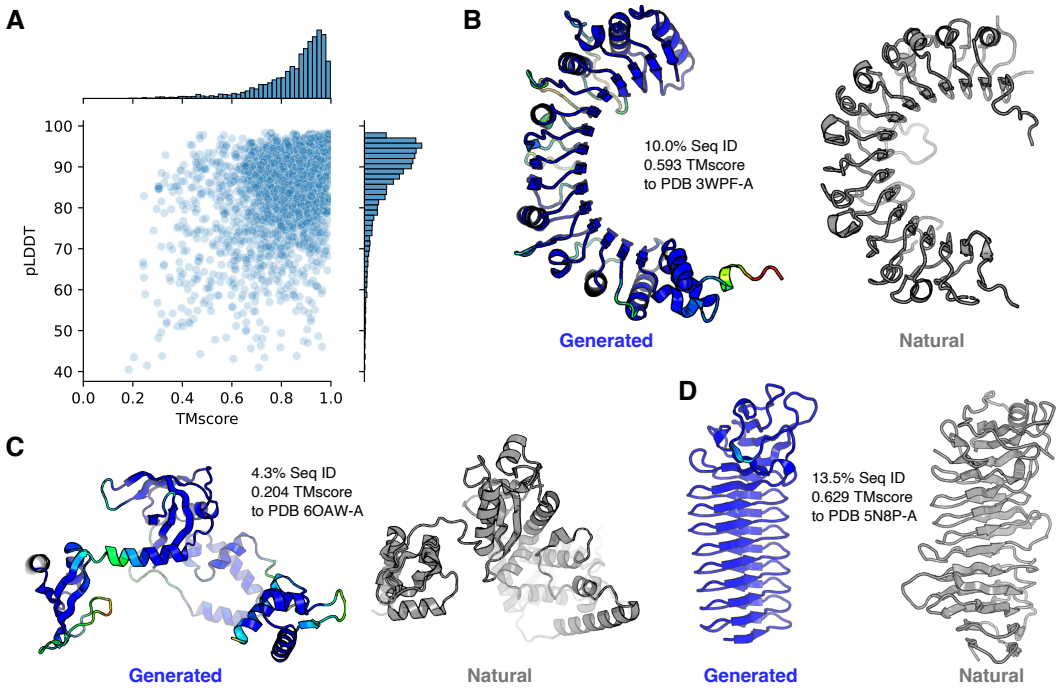


**Figure 5.1:** Generating from a pretrained language model trained on a universal protein dataset

(A) Relationship between AlphaFold2 prediction confidence (pLDDT) and similarity to natural protein structures in the PDB (TMscore). (B-D) Comparison of predicted structures for generated sequences (left, colored by pLDDT) and their closest structural counterparts in the PDB (right, gray). Sequence identities and TMscores are calculated against the closest structural matches in the PDB. (B) Solenoid-fold protein generated by the model, with very low sequence identity and high structural similarity to a toll-like receptor protein. The generated protein replaces several alpha helices on the outer edge of the fold with beta sheets, resulting in a smaller curvature compared to that of its most similar natural counterpart. (C) Multi-domain $\alpha$+$\beta$-fold generated protein with very low sequential or structural similarity to natural proteins. (D) Generated protein resembling prokaryotic surface protein. The generated protein contains more ordered secondary structure (uniform-length beta sheets, shorter loops) than other beta-roll folds found in the PDB.

in the PDB (median TMscore of 0.89). However, closer inspection of predicted structures revealed several unique characteristics of the generated sequences. In Figure 5.1B, we show a generated sequence adopting a solenoid fold. The

closest structural homolog in the PDB is the mouse toll-like receptor 9 (PDB ID 3WPF-A), a similarly folding solenoid protein. Interestingly, although the inner face of the generated solenoid fold is composed entirely of beta sheets (as in the natural protein), the outer face combines both alpha helices and beta strands, resulting in a larger central angle (smaller curvature). Further, despite adopting similar folds, the sequence identity between the generated and natural proteins is only 10.0%. For another generated sequence, adopting a multi-domain $\alpha$+$\beta$-fold (Figure 5.1C), the most similar natural protein was an uncharacterized protein (PDB ID 6OAW-A) with a low TMscore of 0.204 and little sequence overlap (4.3% identity). In a final case study, we highlight a generated sequence with a predicted structure resembling a prokaryotic RsaA surface protein (PDB ID 5N8P-A). Both structures adopt a similar $\beta$-roll fold (TMscore 0.629) yet have a low level of sequence identity (13.5%). Interestingly, we observe that the generated protein resembles an idealized version of the natural protein, with uniform beta sheets and connecting loops. Taken together, these examples illustrate some of the unique properties of sequences generated by ProGen2. While the generated sequences often fold into structures resembling those produced by nature, they frequently do so with significant sequence deviations, and may adopt novel folds in some cases.

**Finetuning enables family-specific sequence generation**

Next, we considered generation from a model finetuned on protein sequences adopting a common structural architecture. The ProGen2-large model was finetuned for two epochs on 1M sequences, from Gene3D [28] and CATH

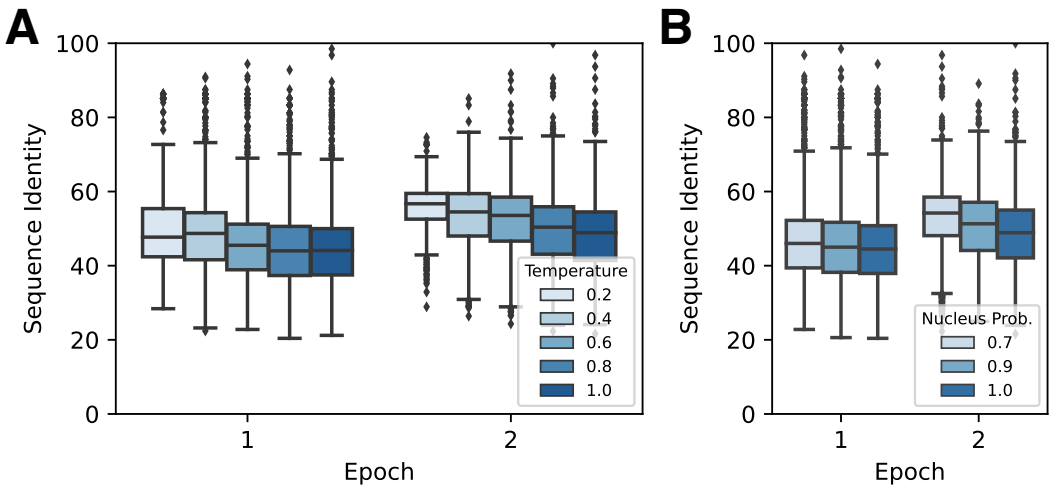


**Figure 5.2:** Effect of finetuning on the sequence similarity of generated proteins to natural proteins

(A) Higher sampling temperature generates more diverse protein sequences. (B) Higher nucleus-sampling probability produces greater sequence diversity.

[29], adopting a two-layer sandwich architecture (CATH 3.30). To understand the effects of extended finetuning, we generated 10,000 sequences using the model parameters after the first and second epoch of finetuning. For all generated sequences, we calculated the sequence identity against the training dataset using MMseqs2 [30]. As expected, we observed higher similarity to observed evolutionary sequences with extended finetuning (Figure 5.2). Among sequences generated with the same model checkpoints, sampling parameters are strongly correlated with sequence novelty (i.e., higher sampling temperature or nucleus probability yields lower sequence identity). To assess the effect of sampling parameters on structure diversity within the common architecture, we predicted structures for all 20,000 sequences with AlphaFold2 and calculated TMscores against the PDB using Foldseek. A similar trend emerged, with more restrictive sampling parameters typically yielding

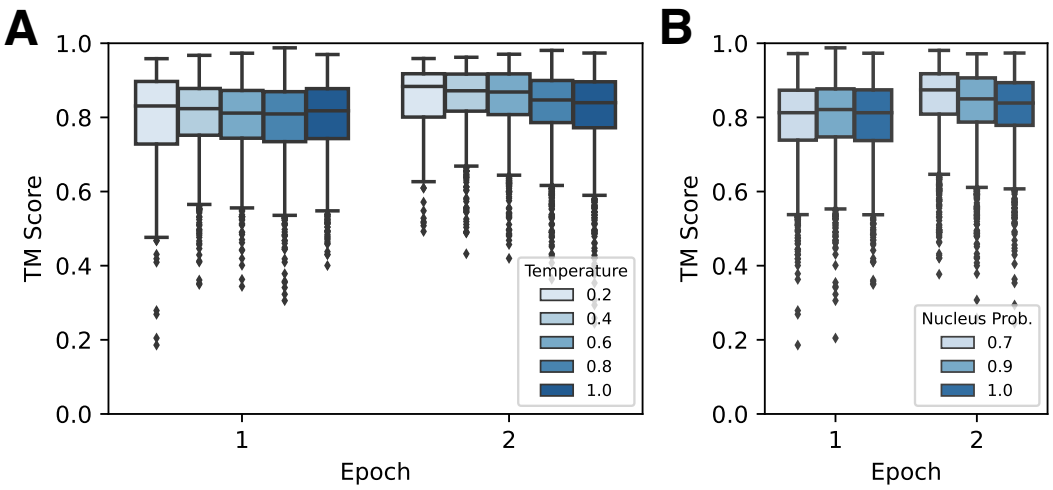


**Figure 5.3:** Effect of finetuning on the structural similarity of generated proteins to natural proteins

(A) In general, lower sampling temperature results in sequences adopting structures more similar (higher TMscore) to those found in the PDB. (B) Lower nucleus sampling probability yields generations with reduced structural diversity.

structures more closely resembling natural proteins (Figure 5.3). Among the more novel structures, the primary source of diversity is in the ligand-binding regions, while the non-binding regions resemble natural proteins (Figure 5.4A-B). In two such cases, the ligand-binding region is less confidently predicted by AlphaFold2 and features rearrangements as compared to the closest natural homologs (Figure 5.4A-B). Interestingly, in both cases the predicted structures present a clear cavity suitable for a ligand, and even mimic the proximal secondary structures of natural proteins. The lower prediction confidence for these regions could be due to the truncated AlphaFold2 prediction process (one recycle) or the ligand-agnostic nature of the model itself. In another case, the predicted structure of the generated sequence confidently recapitulates the ligand-binding region (Figure 5.4C). These results demonstrate that the

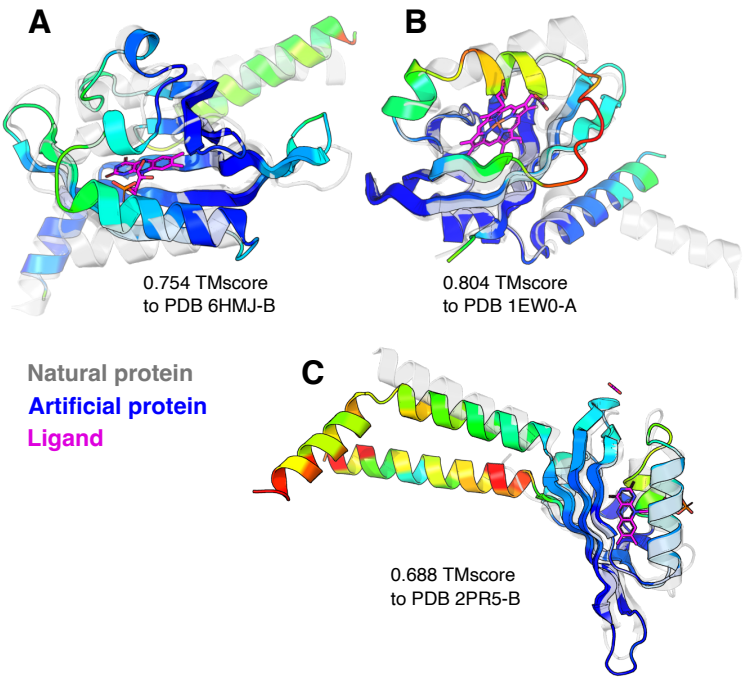


**Figure 5.4:** Examples of proteins generated by a finetuned model

Comparison of predicted structures for sequences generated by the finetuned language model (colored by pLDDT) and the most structurally similar proteins in the PDB (transparent). Ligands bound by the natural proteins are shown in pink. (A) Generated protein adopting a similar fold to a natural protein binding a flavin mononucleotide ligand. The helical secondary structure of the generated protein matches that of the natural protein near the ligand-binding site, but a shorter loop restricts the space available for binding. (B) Generated protein closely resembling a natural protoporhyrin-binding protein. The structure of the generated protein appears to properly accommodate the ligand, but is predicted with low confidence in the unstructured loop regions near the binding site. (C) Generated protein similar to a natural flavin-mononucleotide-binding protein. The binding site of the generated protein is confidently predicted and reserves appropriate space for the ligand.

sequences generated by a finetuned model sample diversity at functional regions, while maintaining the common architecture of the training dataset.

**Immune repertoire pretraining for antibody sequence generation**

Generation of antibody sequences is of particular interest for construction of libraries for therapeutic discovery [13, 17]. However, only relatively small generative models have been trained for this task to date. We investigated the properties of antibody sequences generated by a 764M parameter model pretrained on 554 million natural antibodies, named ProGen2-OAS. First, we generated 52K non-redundant antibody sequences with the pretrained model. However, experimental limitations of sequencing studies result in over half of antibody sequences in the OAS being truncated at the N-termini by 15 or more residues [31]. As such, direct generation from the model yields sequences mirroring the training distribution, rather than fully formed antibody sequences. To overcome this bias in the data and produce full-length antibody sequences, we initiated generation with a three-residue motif commonly found at the beginning of human heavy chain sequences (EVQ) [17]. Using this prompting strategy, we generated an additional 470K full-length antibody sequences (Figure 5.5). In Figure 5.6, we compare the sequence similarity of unprompted and prompted generations to the training distribution. Notably, the prompted sequences share significantly greater sequence identity with the training distribution, likely due to the inclusion of the highly conserved FW1 region that is frequently absent in the N-terminally-truncated unprompted sequences. Intriguingly, we also observe an inverse relationship between more restrictive sampling parameters (lower temperature, higher nucleus probability) and sequence identity to the training dataset. We observe a similar trend for the predicted structures of generated antibody sequences, as measured by TM

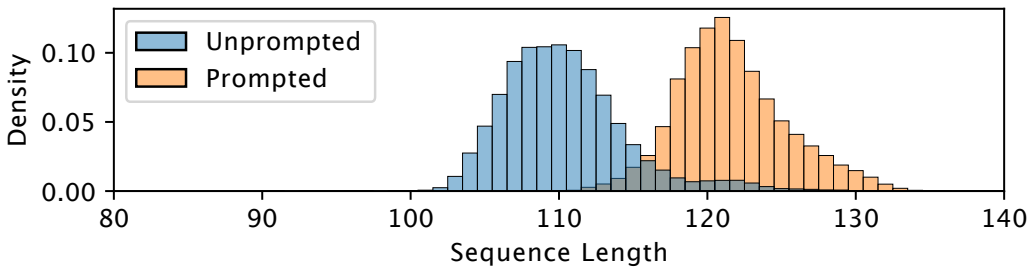


**Figure 5.5:** Comparison of sequence lengths for unprompted and prompted generation strategies

score against the PDB (Figure 5.7).

Potential antibody therapeutics often require extensive optimization to improve their physical properties. Collectively referred to as developability, these properties include thermal stability, expression, aggregation propensity, and solubility [32]. Here, we focused on quantifying the aggregation propensity and solubility of generated sequences according to their SAP scores [33] and CamSol-intrinsic profiles [34]. We found that for both aggregation propensity and solubility, sequences generated with less restrictive parameters display improved developability (Figure 5.8 and Figure 5.9). Given the effective zero-shot predictive capabilities of PLMs [22, 23], we also investigated whether a univerally pretrained model could be used to filter generated antibody libraries and improve their developability profiles. In Figure 5.10, we compare the aggregation propensity and solubility of the full set of generated sequences with the top-50% as scored by the ProGen2-base model. Among the top-ranked sequences, aggregation propensity improves only marginally, while the solubility of the sequences shows a favorable shift. These results provide meaningful guidance for generation of antibody sequence libraries

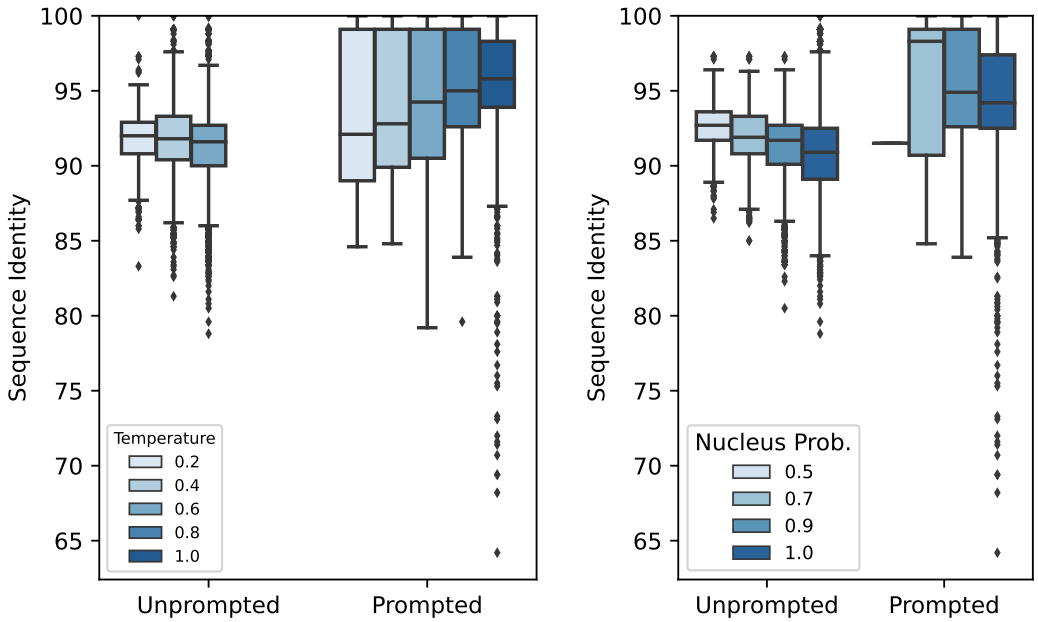


**Figure 5.6:** Comparison of sequence identity to the training dataset for unprompted and prompted generations

Full-length antibody sequences from prompting exhibit generally higher sequence identity. Interestingly, higher sampling temperature tends to produce sequences more similar to the training dataset, while lower nucleus sampling probability yields sequences more closely matching the training dataset.

with PLMs. In practice, generating with less restrictive sampling parameters and filtering with a universal PLM should provide the most developable set of sequences.

### 5.3.3 Zero-shot fitness prediction

Generative models for protein sequence design should ideally learn a representation that aligns with our desired functional attributes. Experimental techniques in the wet laboratory have allowed for the collection of protein libraries that associate a given sequence to one or many functional scalar values,

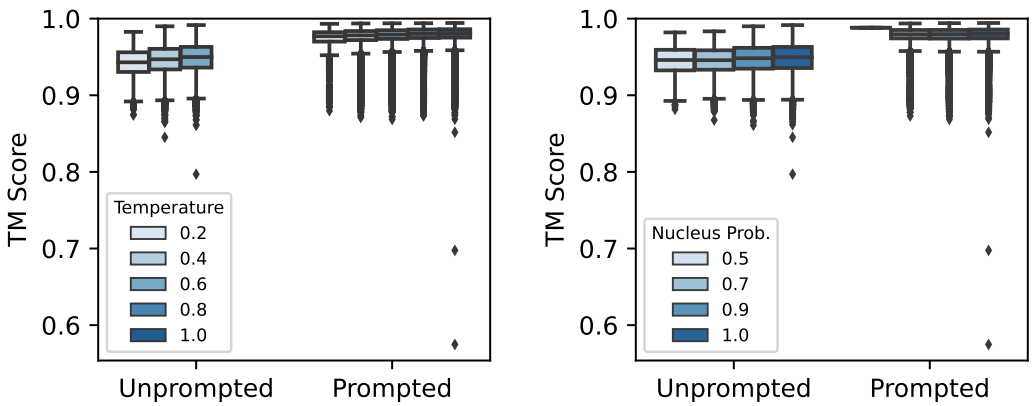


**Figure 5.7:** Structural similarity of generated antibody sequence to natural proteins

which describes a *fitness landscape*. We examine how experimentally-measured fitness landscapes correlate with a generative model's likelihood in a zero-shot manner, meaning there is no additional finetuning in a supervised setting with assay-labeled examples or an unsupervised setting with a focused set of homologous sequences.

**Scale does not improve fitness prediction on narrow landscapes**

For a proper comparison to the models of Hesslow et al. [22] – with a similar architecture to ProGen2 but trained on a different data distribution – we first characterize zero-shot performance on narrow fitness landscapes from Riesselman, Ingraham, and Marks [19] which is comprised mainly of single substitution deep mutational scan experiments. We observe in Figure 5.11 (Table 5.2) that our smallest model (ProGen2-small), with an order of magnitude less parameters to RITA-XL, exhibits higher average performance across zero-shot tasks, indicating the importance of pretraining data distributions. In contrast to RITA, the ProGen2 training data is a mixture comprised of an

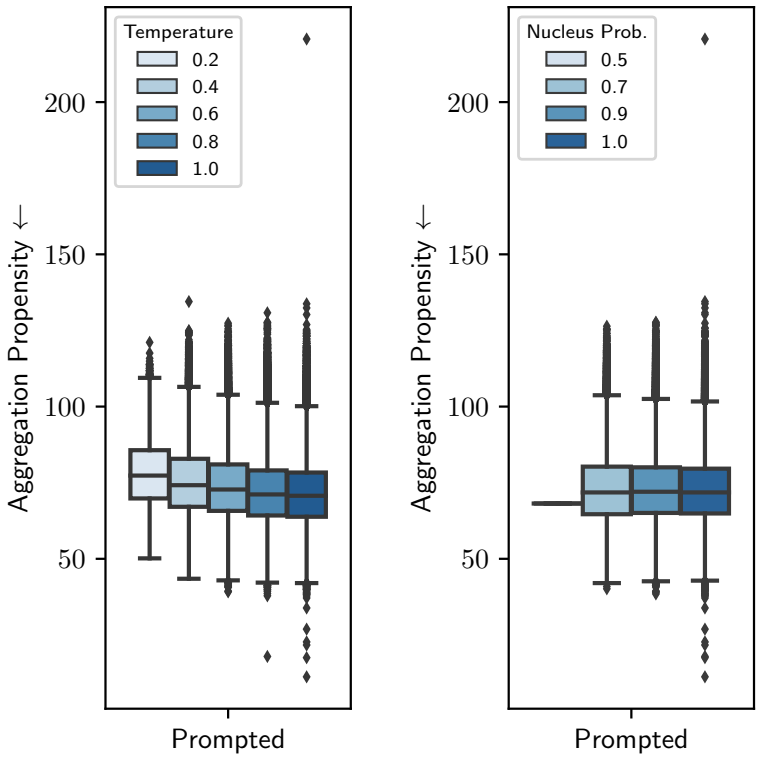


**Figure 5.8:** Impact of sampling parameters on aggregation propensity of generated antibody sequences

Higher sampling temperature results in lower aggregation propensity for generated sequences, while changing nucleus probability has limited effect.

identity-reduced set of sequences from Uniref along with sequences from metagenomic sources. Our best ProGen2 model outperforms or matches all other baselines spanning a variety of differing modeling strategies– amplifying the importance of understanding what set of sequences are provided to the model for training.

Intriguingly, we find that as model capacity increases, performance at zero-shot fitness prediction (averaged across all datasets in the narrow landscape) peaks at 764M parameters (ProGen2-base) before decreasing with larger and

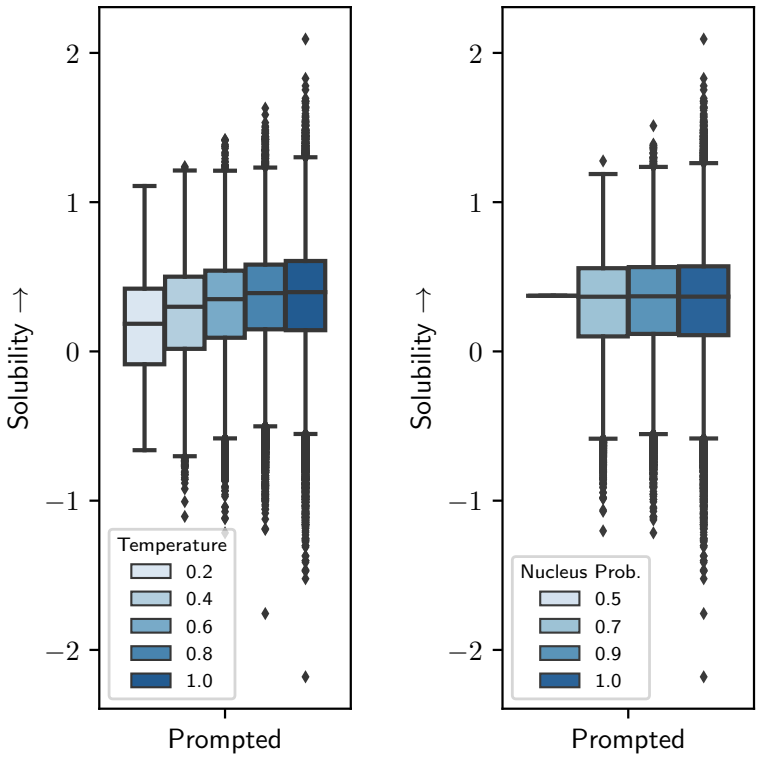


**Figure 5.9:** Impact of sampling parameters on solubility of antibody sequences

Higher sampling temperature results in higher solubility for generated sequences, while changing nucleus probability has limited effect.

larger models (Figure 5.11). This stands in contrast to model perplexity, which improves systematically with model scale (Table 5.1). Our results are in line with Weinstein et al. [35], which suggests that fitness estimates from misspecified models can systematically outperform fitness estimates from well-specified models (even in the limit of infinite data). Intuitively, this result says that phylogenetic biases and other distortions in the dataset can be partially corrected for by using a relatively small but well-chosen model, which is capable of describing the key features present in real fitness landscapes but is not capable of exactly matching the data distribution. Our results provide

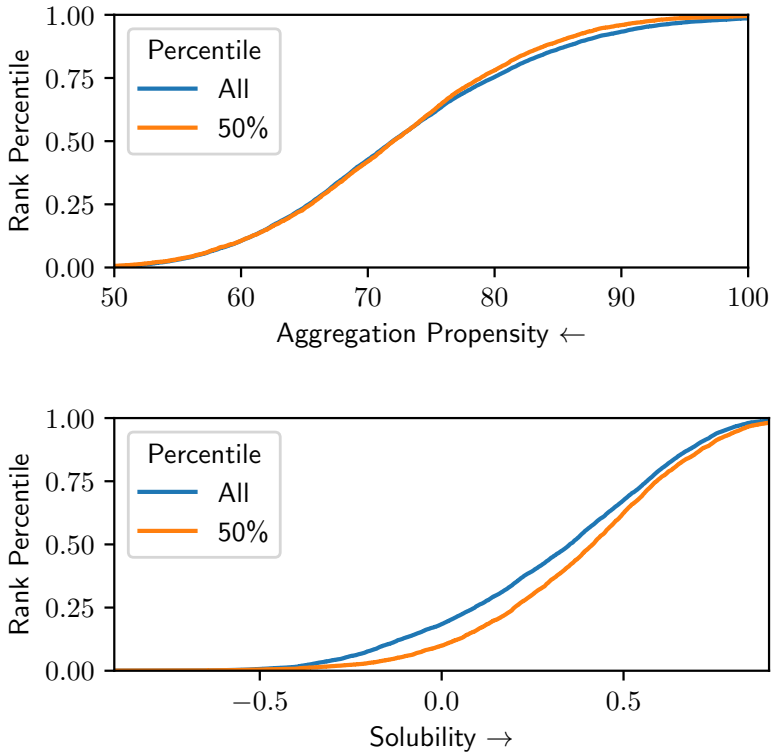


**Figure 5.10:** Ranking generated antibody sequences with universal model

Likelihood ranking of generated antibody sequences with the ProGen2-base language model. Aggregation propensity is not significantly reduced among the top-50% ranked antibody generations. Solubility is improved by selecting the top 50% of ranked antibody generations.

the first evidence that this effect can hold not only in the context of single protein family datasets but also in the context of large-scale datasets containing evolutionarily diverse proteins, and using large-scale transformer models.

### Scale improves fitness prediction on wide mutational landscapes

Although bigger models may not translate into better zero-shot fitness performance in general, they may still have advantages in certain cases. Most

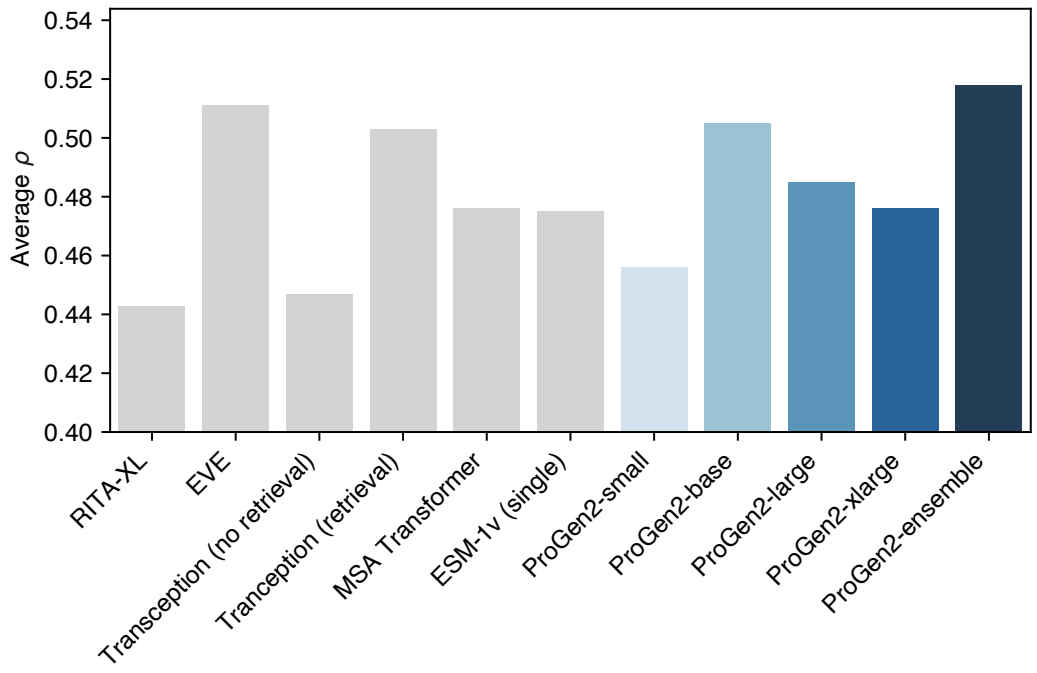


**Figure 5.11:** Zero-shot fitness prediction performance of ProGen2 models and alternative methods on narrow fitness landscapes

Model scale provides limited performance benefits, and even degrades zero-shot capabilities for the largest models.

of the available fitness assays to which we compare focus on well-studied proteins with large numbers of evolutionarily similar sequences, and measure the fitness/functionality of mutants only one or two mutations away from a wild-type sequence. Intuitively, regions of sequence space with very low probability under $p_0$ are likely to be especially poorly described with smaller models, and so in these regions both fitness estimation and generation may suffer. Empirically, we find some suggestive evidence that larger models outperform smaller models at fitness estimation in wider landscapes where sequences are farther from any natural sequence (Figure 5.12, Table 5.3). In

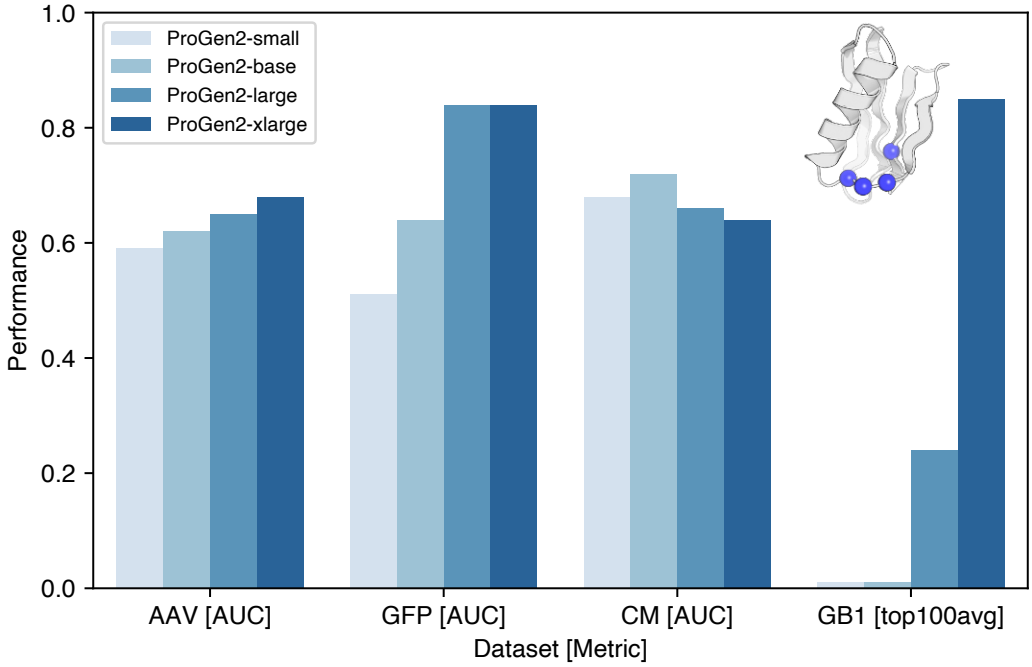


**Figure 5.12:** Zero-shot fitness prediction performance of ProGen2 models on wide fitness landscapes

Performance typically improves with model scale, and may lead to emergent zero-shot capabilities for low-homology, highly epistatic landscapes like GB1 (structure with mutation sites shown).

particular for the GB1 library, a challenging low-homology protein mutated at positions with non-linear epistasis, our largest models may exhibit emergent behavior [9] in zero-shot identification of the highest fitness variants.

### Antibody-specific training does not improve fitness prediction

On antibody-specific landscapes, our results again indicate more attention needs to be placed on the distribution of sequences provided to a model during training. We examine the zero-shot fitness prediction of binding ($K_D$) and general properties (expression and melting temperature $T_M$) of antibodies in

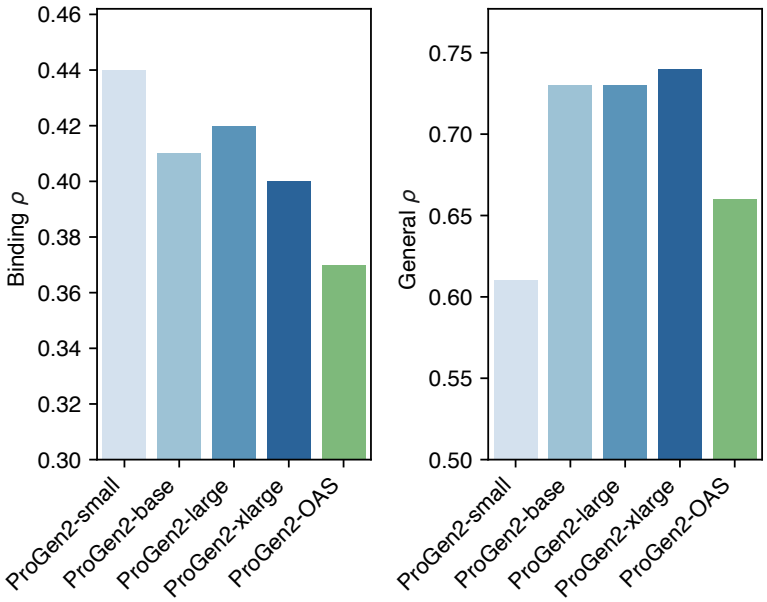


**Figure 5.13:** Zero-shot fitness prediction performance on antibody-specific fitness landscapes

Zero-shot performance of universal ProGen2 models and the antibody-specific ProGen2-OAS for binding datasets and general antibody fitness prediction tasks (e.g., stability and expression). Models trained on broad evolutionary sequence datasets outperform antibody-specific models on both tasks.

Table 5.4. Samples from immune repertoire sequencing studies seem like an intuitive choice for learning powerful representations useful for antibody fitness prediction tasks [36, 37]. However, our ProGen2-OAS model performs poorly as compared to pretrained models trained on universal protein databases (Figure 5.13). Curiously, the binding prediction performance is non-negligible and may be useful in practical antibody engineering campaigns, even though the corresponding antigen is not provided to the model for likelihood calculation.

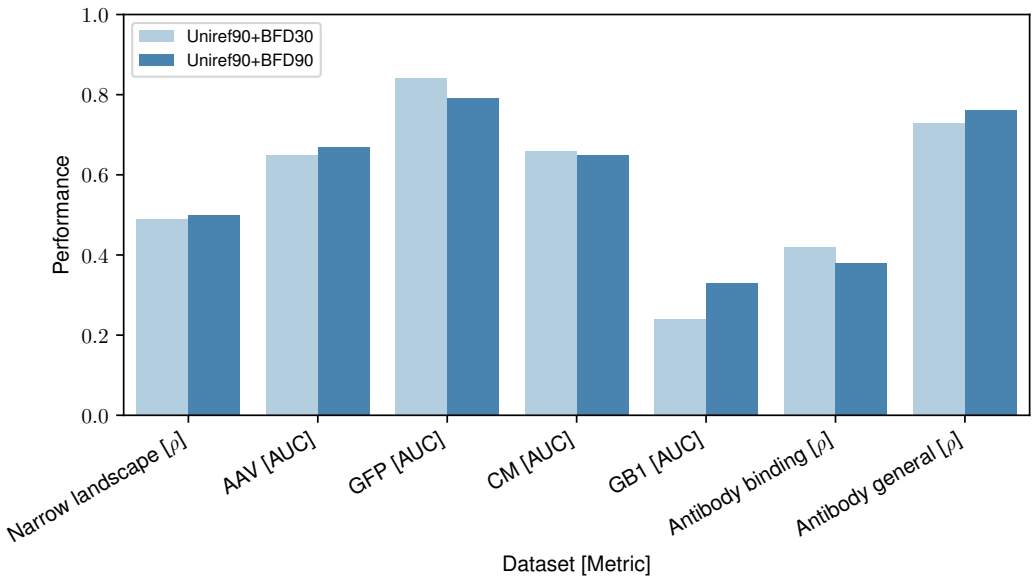


**Figure 5.14:** Zero-shot fitness prediction performance of ProGen2 models trained on alternative data compositions

Comparison of zero-shot fitness prediction performance for 2.7B parameter models trained on Uniref90+BFD30 and Uniref90+BFD90.

## 5.4 Discussion

Protein language models will enable advances in protein engineering and design to solve critical problems for human health and the environment. However, there are many open questions that remain as we begin to realize these advances. In this work, we introduce the ProGen2 suite of models and demonstrate the effectiveness of generative language models for a variety of protein design tasks. Throughout the study, we investigate the impact of increasing model scale for modeling protein sequence landscapes. As model capacity increases, we continue to see improvements in capturing the distribution of natural protein sequences (lower test perplexity). This

**Table 5.2:** Zero-shot fitness prediction on narrow experimentally-measured fitness landscapes

| Model | Average Spearman |
|---|---|
| RITA-XL | 0.443 |
| EVE | 0.511 |
| Tranception (no retrieval) | 0.447 |
| Tranception (retrieval) | 0.503 |
| MSA Transformer | 0.476 |
| ESM-1v (single) | 0.475 |
| ProGen2-small | 0.456 |
| ProGen2-base | 0.505 |
| ProGen2-large | 0.485 |
| ProGen2-xlarge | 0.476 |
| ProGen2-ensemble | 0.518 |

ProGen2-small outperforms an order of magnitude larger RITA-XL and ProGen2-base is the best performing ProGen2 size, indicating larger model capacity does not always translate to improved predictive performance. ProGen2 models outperform or match other baseline methods across a variety of modeling strategies, suggesting the distribution of observed evolutionary sequences provided to the model, along with its inherent biases, likely plays a significant role. The average spearman is reported with data and baselines provided by Hesslow et al. [22].

suggests that current models still underfit the sequence datasets available, and we should expect larger models to deliver further improvements along this axis. Next, we demonstrate the utility of generative language models for creating novel sequences. As shown in prior works [15], pretrained generative models produce diverse sequences spanning the functional and structural space of natural proteins. Sequences from ProGen2 typically adopt natural folds (as predicted by AlphaFold2 [25]) while diverging in sequence space. Further, we show that finetuning ProGen2 models enables a narrowing of the sequence landscape for targeted generation of particular families. Similar approaches have been used to create functional enzymes [16], and are a promising approach for protein design. Finally, we show that the likelihoods

**Table 5.3:** Zero-shot fitness prediction on wider experimental landscapes

| dataset [metric] | ProGen2-small | ProGen2-base | ProGen2-large | ProGen2-xlarge |
|---|---|---|---|---|
| AAV [AUC] | 0.59 | 0.62 | 0.65 | **0.68** |
| GFP [AUC] | 0.51 | 0.64 | **0.84** | **0.84** |
| CM [AUC] | 0.68 | **0.72** | 0.66 | 0.64 |
| GB1 [top100avg] | 0.01 | 0.01 | 0.24 | **0.85** |

Larger model capacity may translate to benefits for landscapes involving higher edit distances or low-homology settings. Particularly for GB1 (a low-homology, epistatic landscape), the largest model may demonstrate emergent behavior in finding top ranked sequences.

learned by large language models like ProGen2 are a useful proxy for protein fitness and are competitive with state-of-the-art methods across a variety of sequence landscapes.

Scaling transformer language models has yielded impressive performance and even emergent capabilities for natural language processing [3, 7]. Several studies have investigated whether these scaling trends apply to protein sequence modeling, and have typically concluded that larger models indeed provide improvements across a variety of tasks [39, 22, 40]. The RITA study found consistent improvements for protein fitness prediction with increasing model capacity up to 1.2B parameters [22]. Similarly, the ESM-2 models (trained for masked language modeling) were better able to predict protein structure in both unsupervised and supervised settings as model sizes were increased up to 15B parameters. In contrast to these results, we show that scaling model capacity is not a panacea for all protein design tasks. While larger ProGen2 models improved zero-shot fitness prediction on broader mutational landscapes, for narrower landscapes composed primarly of amino acid substitutions, we observed a degradation of performance for our largest

**Table 5.4:** Zero-shot fitness prediction on antibody-specific landscapes

| Model | Average Spearman | |
|---|---|---|
| | Binding | General |
| ProGen2-small | 0.44 | 0.61 |
| ProGen2-base | 0.41 | 0.73 |
| ProGen2-large | 0.42 | 0.73 |
| ProGen2-xlarge | 0.40 | 0.74 |
| ProGen2-OAS | 0.37 | 0.66 |

Using redundancy-reduced proteins from immune repertoire sequencing studies, OAS [38], does not lead to better fitness prediction for antibodies. In particular, we examine antibody fitness predictive performance for binding $K_D$ values and general protein properties including expression quality and $T_M$ melting temperatures. The models trained on universal protein databases are better at predicting general properties as compared to binding affinity. Surprisingly, the binding prediction performance is considerably high considering the associated antigen is not provided to the model.

models. The test-max50 and wide fitness landscape results suggest that scale may particularly show advantages for out-of-distribution problems. This is exemplified by the significant advances in zero-shot prediction at larger model scales on the challenging GB1 landscape. Finally, it is worth consideration that fitness as defined as an average spearman correlation coefficient across the multiple experimental datasets in this and other studies comes with its own set of biases and may not be the most reliable criteria for evaluation of models for protein engineering. We refer the reader to prior work from Dallago et al. [41] and Yang et al. [42] for further discussion.

Although pretraining on larger sets of sequences would seem to be an intuitive means of creating broadly useful models, our results suggest that the composition of the pretraining dataset is of critical importance. For zero-shot predictions on narrow fitness landscapes, larger ProGen2 models perform

relatively poorly despite capturing the pretraining sequence distribution better. This indicates a divergence between the two, and could potentially be remedied by identifying a more suitable pretraining corpus. Conversely, for broader mutational landscapes, larger models that better capture the pretraining dataset typically improve zero-shot performance. For the GB1 landscape in particular, pretraining on BFD90 rather than BFD30 yielded significant improvements at the same model scale. Perhaps the most distinctive illustration of the importance of dataset-task alignment is the lackluster zero-shot performance of models pretrained on immune repertoire sequences from the OAS. For both binding and general properties of antibody sequences, ProGen2 models pretrained on universal sets of proteins sequences (rather than just antibodies) significantly outperformed the model pretrained on antibodies alone (even when having fewer parameters). In the case of antibodies, this may be because the selective pressures on natural antibodies diverge from the properties evaluated experimentally (such as thermal stability and binding affinity). More broadly, these results suggest that to improve model performance we must carefully consider the alignment of the pretraining dataset and the downstream task.

## 5.5 Methods

### 5.5.1 Model

The family of ProGen2 models are autoregressive transformers with next-token prediction language modeling as the learning objective trained in various sizes with 151M, 764M, 2.7B, and 6.4B parameters. Table 5.5 summarizes

the model specifications and choice of hyper-parameters for the optimization such models.

### 5.5.2 Data

The standard ProGen2 models are pretrained on a mixture of Uniref90 [43] and BFD30 [44] databases. Uniref90 are cluster representative sequences from UniprotKB at 90% sequence identity. The BFD30 dataset is approximately 1/3 the size of Uniref90, majority from metagenomic sources, commonly not full-length proteins, and clustered at 30% sequence identity. For the ProGen2-BFD90 model, Uniref90 is mixed with representative sequences with at least 3 cluster members after clustering UniprotKB, Metaclust, SRC, and MERC at 90% sequence identity. This BFD90 dataset is approximately twice the size as Uniref90. To train the antibody-specific ProGen2-OAS, we collected unpaired antibody sequences from the Observed Antibody Space (OAS) database [38]. We refer to the supplement for details.

### 5.5.3 Evaluation

Two test sets at differing levels of difficulty were constructed to examine language modeling performance. Test-max90 and Test-max50 correspond to representative sequences from held-out clusters from the Uniref90+BFD30 set of sequences at 90% and 50% sequence identity respectively.

To investigate the properties of sequences generated by the ProGen2 family of models, we sampled complete protein sequences in three settings: universal

generation after pretraining, fold-specific generation after finetuning, and antibody generation after pretraining on only antibody sequences. For universal protein generation, we sampled 5K sequences from the ProGen2-xlarge model. To understand the effects of architecture-specific finetuning on sequence generation, we compared 10K sequences produced by the ProGen2-large model after one and two epochs of finetuning. Antibody sequences were generated using the ProGen2-OAS model after pretraining on a set of variable-fragment sequences from the OAS [38]. Sequences were generated using two prompting strategies: unprompted (52K sequences) and initial-residue prompted (470K sequences).

To assess zero-shot fitness prediction ability, we evaluate on three sets of experimentally-measured protein landscapes: narrow, wide, and antibody-specific. The narrow landscape set is comprised of the Riesselman, Ingraham, and Marks [19] datasets as provided by the authors of Hesslow et al. [22] and generally includes variants that are one or two substitutions away from a given wild-type/natural sequence. The wide landscape set involves larger edit distances and are comprised of the Dallago et al. [41] proteins, chorismate mutase proteins from Russ et al. [11], and the GFP test set proteins from Rao et al. [45]. Lastly, for the antibody-specific landscape, we compiled a dataset consisting of binding, expression, and thermal stability measurements for variants derived from eight distinct antibodies. We refer to the supplement for details.

## 5.6 Appendix

### 5.6.1 Model Parameters

Our models are autoregressive transformers with next-token prediction language modeling as the learning objective. The family of ProGen2 models is trained in various sizes with 151M, 764M, 2.7B, and 6.4B parameters.

The architecture follows a standard transformer decoder with left-to-right causal masking. For the positional encoding, we adopt rotary positional encodings [46]. For the forward pass, we execute the self-attention and feed-forward circuits in parallel for improved communication overhead following [47], that is, $x_{t+1} = x_t + \text{mlp}(\ln(x_t + \text{attn}(\ln(x_t))))$ is altered to $x_{t+1} = x_t + \text{attn}(\ln(x_t)) + \text{mlp}(\ln(x_t))$ for which the computation of self-attention, $\text{attn}()$, and feed-forward, $\text{mlp}()$, with layer-norm, $\ln()$, is simultaneous.

Table 5.5 summarizes the model specifications and choice of hyper-parameters for the optimization such models. The choice of the hyper-parameters was informed by [3], however, the number of layers is reduced with a small number of self-attention heads of relatively high dimensionality to improve overall utilization of the TPU-v3 compute. As explored in [3, 47, 48], these variations introduce insignificant degradation of perplexity for sufficiently large models, while significantly improving computational efficiency.

For the pretraining of the ProGen2 models, Table 5.5 summarizes the hyper-parameters. We adopt the Adam [49] optimizer with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.999, 1\text{e}{-}08)$ and global gradient norm clipping [50] of 0.8 and 1.0. The learning rate function over time follows GPT-3 [3] with warm-up steps and

**Table 5.5:** Model specifications and hyper-parameters

| | Model | | | | |
|---|---|---|---|---|---|
| Hyper-parameter | ProGen2-small | ProGen2-medium | ProGen2-base | ProGen2-large | ProGen2-xlarge |
| Number of params | 151M | 764M | 764M | 2.7B | 6.4B |
| Number of layers | 12 | 27 | 27 | 32 | 32 |
| Number of heads | 16 | 16 | 16 | 32 | 16 |
| Head dimensions | 64 | 96 | 96 | 80 | 256 |
| Context length | 1,024 | 1,024 | 2,048 | 1,024 | 1,024 |
| Batch size | 500k | 500k | 500k | 500k | 1M |
| Learning rate | 6.0e-4 | 2.5e-4 | 2.0e-4 | 0.8e-4 | 0.1e-4 |
| Weight decay | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Grad norm clip | 1.0 | 1.0 | 0.8 | 0.8 | 0.8 |
| Warm-up steps | 3,000 | 3,000 | 10,000 | 10,000 | 10,000 |
| Total steps | 350,000 | 350,000 | 400,000 | 400,000 | 350,000 |

Choice of hyper-parameters for model specification and optimization for the family of ProGen2 causal language models for protein engineering.

cosine annealing.

Notably, the cross-entropy appeared to diverge from the projected power-law relation over time when following standard configurations detailed in [3]. In particular, an increasing the global norm of the gradient as an indicator for a divergence from the expected log-log linear behavior of cross-entropy over time was observed. Decreasing the learning rate, increasing weight-decay (or equivalently $\ell_2$-regularization under re-parameteriztation) and decreasing the gradient norm clipping factor resulted in a near-constant global norm of the gradient which stabilized training.

For the finetuning of the ProGen2 models, the training is continued from a converged model. The state of the optimizer is re-initialized such Adam's moving averages for the first and second moment estimators are set to zero. The learning rate decay function is adjusted such that initial learning-rate is decreased by a factor of 5. The finetuning covers at most two epochs over the finetuning dataset to avoid over-fitting.

### 5.6.2 Training Data

The standard ProGen2 models are pretrained on a mixture of Uniref90 [43] and BFD30 [44] databases. Uniref90 are cluster representative sequences from UniprotKB at 90% sequence identity. The BFD30 dataset is approximately 1/3 the size of Uniref90, majority from metagenomic sources, commonly not full-length proteins, and clustered at 30% sequence identity. For the ProGen2-BFD90 model, Uniref90 is mixed with representative sequences with at least 3 cluster members after clustering UniprotKB, Metaclust, SRC, and MERC at 90% sequence identity. This BFD90 dataset is approximately twice the size as Uniref90.

To train the antibody-specific ProGen2-OAS, we collected unpaired antibody sequences from the Observed Antibody Space (OAS) database [38]. OAS is a curated collection of 1.5B antibody sequences from eighty immune repertoire sequencing studies, which contains heavy and light chain sequences from six species (humans, mice, rats, camel, rabbit, and rhesus). The sequences in OAS possess a significant degree of redundancy, due both to discrepancies in the sizes of its constituent studies, as well as the innate biological redundancy of antibody sequences within organisms. To reduce this redundancy, we clustered the OAS sequences at 85% sequence identity using Linclust [44], yielding a set of 554M sequences for model training. Alignment coverage in Linclust was calculated with respect to the target sequence ("cov-mode 1"), with all other parameters set to their default values.

All samples are provided to the model with a 1 or 2 character token concatenated at the N-terminal and C-terminal side of the sequence. Each sequence is

then provided as-is and flipped. For a given batch, proteins are concatenated with others to fill the maximum token length during training.

### 5.6.3 Evaluation Methods

Two test sets at differing levels of difficulty were constructed to examine language modeling performance. Test-max90 and Test-max50 correspond to representative sequences from held-out clusters from the Uniref90+BFD30 set of sequences at 90% and 50% sequence identity respectively.

To investigate the properties of sequences generated by the ProGen2 family of models, we sampled complete protein sequences in three settings: universal generation after pretraining, fold-specific generation after finetuning, and antibody generation after pretraining on only antibody sequences. For universal protein generation, we sampled 5,000 sequences from the ProGen2-xlarge model. To understand the effects of architecture-specific finetuning on sequence generation, we compared 10,000 sequences produced by the ProGen2-large model after one and two epochs of finetuning. In both generation settings, we varied the sampling temperature and nucleus sampling probability to produce a diverse set of sequences. Structures were predicted for a subset of generated sequences using AlphaFold2 [25], and the similarity to known structures in the PDB was measured with Foldseek [27].

Antibody sequences were generated using the ProGen2-OAS model after pretraining on a set of variable-fragment sequences from the OAS [38]. Sequences were generated using two prompting strategies: unprompted (52K sequences) and initial-residue prompted (470K sequences). For initial-residue

prompting, we began generation with a three-residue sequence motif commonly observed in human heavy chain sequences (EVQ). For both prompting strategies, we generate a diverse set of sequences by varying the sampling temperature and nucleus sampling probability. Structures for all generated antibody sequences were predicted using IgFold [51]. To investigate the therapeutic developability of generated antibody sequences, aggregagation propensity [33] and solubility [34] were calculated for all sequences.

To assess zero-shot fitness prediction ability, we evaluate on three sets of experimentally-measured protein landscapes: narrow, wide, and antibody-specific. The narrow landscape set is comprised of the Riesselman, Ingraham, and Marks [19] datasets as provided by the authors of Hesslow et al. [22] and generally includes variants that are one or two substitutions away from a given wild-type/natural sequence. The wide landscape set involves larger edit distances and are comprised of the Dallago et al. [41] proteins, chorismate mutase proteins from Russ et al. [11], and the GFP test set proteins from Rao et al. [45].

Lastly, for the antibody-specific landscape, we compiled a dataset consisting of binding, expression, and thermal stability measurements for variants derived from eight distinct antibodies. We collected expression and antigen-binding enrichment measurements for variants of the anti-VEGF g6 antibody from a DMS study [52]. From a second DMS study, we collected binding enrichment measurements for variants of the d44 anti-lysozyme antibody [53]. Binding affinity ($K_D$) and thermal stability measurements ($T_M$) for the remaining six antibodies (C143, MEDI8852UCA, MEDI8852, REGN10987, S309, and

mAb114) were drawn from a recent study on antibody affinity maturation using pretrained language models [54]. We combined measurements for the mAb114 and mAb114UCA antibodies from the original study into a single fitness dataset because the parent sequences shared significant overlap.

### 5.6.4 Sequence Generation

To investigate the properties of sequences generated by the ProGen2 family of models, we sampled complete protein sequences in three settings: universal generation after pretraining, fold-specific generation after finetuning, and antibody generation after pretraining on only antibody sequences. For universal protein generation, we sampled 5,000 sequences from the ProGen2-xlarge model. A diverse set of sequences was sampled using a Cartesian product of temperature ($T \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$) and nucleus sampling ($P \in \{0.5, 0.7, 0.9, 1.0\}$) parameters. To understand the effects of architecture-specific finetuning on sequence generation, we compared the sequences produced by the ProGen2-large model after one and two epochs of finetuning. Using a similar strategy as for universal protein generation, 10,000 sequences were generated using a Cartesian product of temperature ($T \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$) and nucleus sampling ($P \in \{0.7, 0.9, 1.0\}$) parameters for both model checkpoints. The structures of all generated sequences were predicted with AlphaFold2 [25]. For universal generations from the pretrained model, structures were predicted using ColabFold [55] with twelve recycles (other parameters set to their default values). For generations after finetuning, structures were predicted using DeepMind's implementation of

AlphaFold2 with single-sequence inputs (no MSAs), structural templates from the PDB [26], and only one recycle. All structures were predicted with the full five-model ensemble (using the pTM models) and the top-ranked structures for each sequence were considered for structural analysis. Similarity of predicted structures to observed proteins in the PDB was measured by calculating the TMscore [56] using Foldseek [27]. For universal generations, we report the sequence identity against the most structurally similar protein reported by Foldseek. For finetuned generations, we calculated the sequence identity against the finetuning dataset using MMseqs2 [30].

Antibody sequences were generated using the ProGen2-OAS model after pretraining on a set of variable-fragment sequences from the OAS. We evalauted sequences generated by the model with and without initial-residue prompting. A set of 52K unprompted sequences was generated using sampling parameters from a Cartesian product of temperature ($T \in \{0.2, 0.4, 0.6\}$) and nucleus sampling probability ($P \in \{0.5, 0.7, 0.9, 1.0\}$). An additional 470K full-length sequences were generated by initializing the sequence with a three-residue motif commonly observed in human heavy chain antibody sequences (EVQ). Prompted sequences were similarly generated using a Cartesian product of temperature ($T \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$) and nucleus sampling ($P \in \{0.5, 0.7, 0.9, 1.0\}$) parameters. The sequence identity of generated sequences against the training dataset was calculated with MMseqs2 [30]. IgFold [51] was used to predict structures for all generated antibody sequences. The full four-model ensemble of IgFold models was used for predictions, with PyRosetta [57] refinement applied to model outputs. Aggregation propensities

of generated sequences were measured by calculating the SAP score [33] of the predicted structures. Solubility profiles were calculated based on sequence using the public CamSol-intrinsic [34] web server.

# References


[1] Frances H Arnold. "Design by directed evolution". In: *Accounts of Chemical Research* 31.3 (1998), pp. 125–131.

[2] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". In: *arXiv preprint arXiv:2205.11487* (2022).

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[5] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. "Pretrained transformers as universal computation engines". In: *arXiv preprint arXiv:2103.05247* (2021).

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[7] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361* (2020).

[8] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. "Training Compute-Optimal Large Language Models". In: *arXiv preprint arXiv:2203.15556* (2022).

[9] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. "Emergent Abilities of Large Language Models". In: *arXiv preprint arXiv:2206.07682* (2022).

[10] Yosephin Gumulya, Jong-Min Baek, Shun-Jie Wun, Raine ES Thomson, Kurt L Harris, Dominic JB Hunter, James BYH Behrendorff, Justyna Kulig, Shan Zheng, Xueming Wu, et al. "Engineering highly functional thermostable proteins using ancestral sequence reconstruction". In: *Nature Catalysis* 1.11 (2018), pp. 878–888.

[11] William P Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, et al. "An evolution-based model for designing chorismate mutase enzymes". In: *Science* 369.6502 (2020), pp. 440–445.

[12] Donatas Repecka, Vykintas Jauniskis, Laurynas Karpus, Elzbieta Rembeza, Irmantas Rokaitis, Jan Zrimec, Simona Poviloniene, Audrius Laurynenas, Sandra Viknander, Wissam Abuajwa, et al. "Expanding functional protein sequence spaces using generative adversarial networks". In: *Nature Machine Intelligence* 3.4 (2021), pp. 324–333.

[13] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. "Protein design and variant prediction using autoregressive generative models". In: *Nature Communications* 12.1 (2021), pp. 1–11.

[14] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. "ProGen: Language modeling for protein generation". In: *arXiv preprint arXiv:2004.03497* (2020).

[15] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. "A deep unsupervised language model for protein design". In: *bioRxiv* (2022).

[16] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. "Large language models generate functional protein sequences across diverse families". In: *Nature Biotechnology* (2023), pp. 1–8.

[17] Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. "Generative language modeling for antibody design". In: *bioRxiv* (2021).

[18] Thomas A Hopf, John B Ingraham, Frank J Poelwijk, Charlotta PI Schärfe, Michael Springer, Chris Sander, and Debora S Marks. "Mutation effects predicted from sequence co-variation". In: *Nature Biotechnology* 35.2 (2017), pp. 128–135.

[19] Adam J Riesselman, John B Ingraham, and Debora S Marks. "Deep generative models of genetic variation capture the effects of mutations". In: *Nature Methods* 15.10 (2018), pp. 816–822.

[20] Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K Min, Kelly Brock, Yarin Gal, and Debora S Marks. "Disease variant prediction with deep generative models of evolutionary data". In: *Nature* 599.7883 (2021), pp. 91–95.

[21] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. "Language models enable zero-shot prediction of the effects of mutations on protein function". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 29287–29303.

[22] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. "RITA: a Study on Scaling Up Generative Protein Sequence Models". In: *arXiv preprint arXiv:2205.05789* (2022).

[23] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan Gomez, Debora S Marks, and Yarin Gal. "Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval". In: *arXiv preprint arXiv:2205.13760* (2022).

[24] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. "Improving language models by retrieving from trillions of tokens". In: *arXiv preprint arXiv:2112.04426* (2021).

[25] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[26] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. "The Protein Data Bank". In: *Nucleic Acids Research* 28.1 (2000), pp. 235–242.

[27] Michel van Kempen, Stephanie Kim, Charlotte Tumescheit, Milot Mirdita, Johannes Söding, and Martin Steinegger. "Foldseek: fast and accurate protein structure search". In: *bioRxiv* (2022).

[28] Tony E Lewis, Ian Sillitoe, Natalie Dawson, Su Datt Lam, Tristan Clarke, David Lee, Christine Orengo, and Jonathan Lees. "Gene3D: extensive prediction of globular domains in proteins". In: *Nucleic Acids Research* 46.D1 (2018), pp. D435–D439.

[29] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla SM Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, et al. "CATH: increased structural coverage of functional space". In: *Nucleic Acids Research* 49.D1 (2021), pp. D266–D273.

[30] Martin Steinegger and Johannes Söding. "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets". In: *Nature Biotechnology* 35.11 (2017), pp. 1026–1028.

[31] Tobias H Olsen, Iain H Moal, and Charlotte M Deane. "AbLang: an antibody language model for completing antibody sequences". In: *Bioinformatics Advances* 2.1 (2022), vbac046.

[32] Matthew IJ Raybould, Claire Marks, Konrad Krawczyk, Bruck Taddese, Jaroslaw Nowak, Alan P Lewis, Alexander Bujotzek, Jiye Shi, and Charlotte M Deane. "Five computational developability guidelines for therapeutic antibody profiling". In: *Proceedings of the National Academy of Sciences* 116.10 (2019), pp. 4025–4030.

[33] Naresh Chennamsetty, Vladimir Voynov, Veysel Kayser, Bernhard Helk, and Bernhardt L Trout. "Prediction of aggregation prone regions of therapeutic proteins". In: *The Journal of Physical Chemistry B* 114.19 (2010), pp. 6614–6624.

[34] Pietro Sormanni, Francesco A Aprile, and Michele Vendruscolo. "The CamSol method of rational design of protein mutants with enhanced solubility". In: *Journal of Molecular Biology* 427.2 (2015), pp. 478–490.

[35] Eli N Weinstein, Alan Nawzad Amin, Jonathan Frazer, and Debora Susan Marks. "Non-identifiability and the Blessings of Misspecification in Models of Molecular Fitness". In: *Advances in Neural Information Processing Systems*. 2022.

[36] Jinwoo Leem, Laura S Mitchell, James HR Farmery, Justin Barton, and Jacob D Galson. "Deciphering the language of antibodies using self-supervised learning". In: *Patterns* (2022), p. 100513.

[37] Jeffrey A Ruffolo, Jeffrey J Gray, and Jeremias Sulam. "Deciphering antibody affinity maturation with language models and weakly supervised learning". In: *arXiv preprint arXiv:2112.07782* (2021).

[38] Tobias H Olsen, Fergus Boyles, and Charlotte M Deane. "Observed Antibody Space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences". In: *Protein Science* 31.1 (2022), pp. 141–146.

[39] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. "ProtTrans: towards cracking the language of Life's code through self-supervised deep learning and high performance computing". In: *arXiv preprint arXiv:2007.06225* (2020).

[40] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. "Language models of protein sequences at the scale of evolution enable accurate structure prediction". In: *bioRxiv* (2022).

[41] Christian Dallago, Jody Mou, Kadina E Johnston, Bruce J Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. "FLIP: Benchmark tasks in fitness landscape inference for proteins". In: *bioRxiv* (2021).

[42] Kevin K Yang, Alex X Lu, and Nicolo K Fusi. "Convolutions are competitive with transformers for protein sequence pretraining". In: *bioRxiv* (2022).

[43] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches". In: *Bioinformatics* 31.6 (2015), pp. 926–932.

[44] Martin Steinegger and Johannes Söding. "Clustering huge protein sequence sets in linear time". In: *Nature Communications* 9.1 (2018), pp. 1–8.

[45] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. "Evaluating protein transfer learning with TAPE". In: *Advances in Neural Information Processing Systems* 32 (2019).

[46] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. "Roformer: Enhanced transformer with rotary position embedding". In: *arXiv preprint arXiv:2104.09864* (2021).

[47] Ben Wang and Aran Komatsuzaki. *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model*. https://github.com/kingoflolz/mesh-transformer-jax. 2021.

[48] Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. "A conversational paradigm for program synthesis". In: *arXiv preprint arXiv:2203.13474* (2022).

[49] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations*. 2015. URL: http://arxiv.org/abs/1412.6980.

[50] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 1310–1318.

[51] Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. "Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies". In: *bioRxiv* (2022).

[52] Patrick Koenig, Chingwei V Lee, Benjamin T Walters, Vasantharajan Janakiraman, Jeremy Stinson, Thomas W Patapoff, and Germaine Fuh. "Mutational landscape of antibody variable domains reveals a switch modulating the interdomain conformational dynamics and antigen binding". In: *Proceedings of the National Academy of Sciences* 114.4 (2017), E486–E495.

[53] Shira Warszawski, Aliza Borenstein Katz, Rosalie Lipsh, Lev Khmelnitsky, Gili Ben Nissan, Gabriel Javitt, Orly Dym, Tamar Unger, Orli Knop, Shira Albeck, et al. "Optimizing antibody affinity and stability by the automated design of the variable light-heavy chain interfaces". In: *PLOS Computational Biology* 15.8 (2019), e1007207.

[54] Brian L Hie, Duo Xu, Varun R Shanker, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, and Peter S Kim. "Efficient evolution of human antibodies from general protein language models and sequence information alone". In: *bioRxiv* (2022).

[55] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. "ColabFold: making protein folding accessible to all". In: *Nature Methods* (2022), pp. 1–4.

[56] Yang Zhang and Jeffrey Skolnick. "Scoring function for automated assessment of protein structure template quality". In: *Proteins: Structure, Function, and Bioinformatics* 57.4 (2004), pp. 702–710.

[57] Sidhartha Chaudhury, Sergey Lyskov, and Jeffrey J Gray. "PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta". In: *Bioinformatics* 26.5 (2010), pp. 689–691.

# Chapter 6

# Generative language modeling for antibody design

Adapted from Richard W Shuai*, Jeffrey A Ruffolo*, and Jeffrey J Gray. "Generative Language Modeling for Antibody Design". *bioRxiv* (2021). Reproduced with permission. *Joint first authors.

## 6.1 Abstract

Discovery and optimization of monoclonal antibodies for therapeutic applications relies on large sequence libraries, but is hindered by developability issues such as low solubility, low thermal stability, high aggregation, and high immunogenicity. Generative language models, trained on millions of protein sequences, are a powerful tool for on-demand generation of realistic, diverse sequences. We present Immunoglobulin Language Model (IgLM), a deep generative language model for creating synthetic libraries by re-designing variable-length spans of antibody sequences. IgLM formulates antibody design as an autoregressive sequence generation task based on text-infilling in

natural language. We trained IgLM on 558M antibody heavy- and light-chain variable sequences, conditioning on each sequence's chain type and species-of-origin. We demonstrate that IgLM can generate full-length heavy and light chain sequences from a variety of species, as well as infilled CDR loop libraries with improved developability profiles. IgLM is a powerful tool for antibody design and should be useful in a variety of applications.

## 6.2 Introduction

Antibodies have become popular for therapeutics because of their diversity and ability to bind antigens with high specificity [1]. Traditionally, monoclonal antibodies (mAbs) have been obtained using hybridoma technology, which requires the immunization of animals [2]. In 1985, the development of phage display technology allowed for in vitro selection of specific, high-affinity mAbs from large antibody libraries [3, 4, 5]. Despite such advances, therapeutic mAbs derived from display technologies face issues with developability, such as poor expression, low solubility, low thermal stability, and high aggregation [6, 7]. Display technologies rely on a high-quality and diverse antibody library as a starting point to isolate high-affinity antibodies that are more developable [8]. Synthetic antibody libraries are prepared by introducing synthetic DNA into regions of the antibody sequences that define the complementarity-determining regions (CDRs), allowing for human-made antigen-binding sites. However, the space of possible synthetic antibody sequences is very large (diversifying 10 positions of a CDR yields $20^{10} \approx 10^{13}$

possible variants). To discover antibodies with high affinity, massive synthetic libraries on the order of $10^{10}$–$10^{11}$ variants must be constructed, often containing substantial fractions of non-functional antibodies [8, 2].

Recent work has leveraged natural language processing methods for unsupervised pretraining on massive databases of raw protein sequences for which structural data are unavailable [9, 10, 11]. These works have explored a variety of pretraining tasks and downstream model applications. For example, the ESM family of models (trained for masked language modeling) have been applied to representation learning [9], variant effect prediction [12], and protein structure prediction [13]. Autoregressive language modeling, an alternative paradigm for pretraining, has also been applied to protein sequence modeling. Such models have been shown to generate diverse protein sequences, which often adopt natural folds despite diverging significantly in residue makeup [14, 15]. In some cases, these generated sequences even retain enzymatic activity comparable to natural proteins [16]. Autoregressive language models have also been shown to be powerful zero-shot predictors of protein fitness, with performance in some cases continuing to improve with model scale [17, 15].

Another set of language models have been developed specifically for antibody-related tasks. The majority of prior work in this area has focused on masked language modeling of sequences in the Observed Antibody Space (OAS) database [18]. Prihoda et al. developed Sapiens, a pair of distinct models (each with 569K parameters) for heavy and light chain masked language modeling [19]. The Sapiens models were trained on 20M and 19M heavy

and light chains respectively, and shown to be effective tools for antibody humanization. Ruffolo et al. developed AntiBERTy, a single masked language model (26M parameters) trained on a corpus of 558M sequences, including both heavy and light chains [20]. AntiBERTy has been applied to representation learning for protein structure prediction [21]. Leem et al. developed AntiBERTa, a single masked language model (86M parameters) trained on a corpus of 67M antibody sequences (both heavy and light). Representations for AntiBERTa were used for paratope prediction. Olsen et al. developed AbLang, a pair of masked language models trained on 14M heavy chains and 187K light chains, for sequence restoration [22]. For sequence generation, autoregressive generative models have been trained on antibody sequences and used for library design [23, 24]. Akbar et al. [23] trained an LSTM for autoregressive generation of CDR H3 loops and conducted an in silico investigation of their potential for binding antigens. Shin et al. [24] experimentally validated a set of nanobody sequences with generated CDR3 loops and showed promising improvements to viability and binding discovery when compared to traditional approaches, despite the library being over 1000-fold smaller. However, because this generative model was unidirectional, it could not be used to directly re-design the CDR3 loop *within the sequence*, and instead had to be oversampled to produce sequences matching the residues following the loop.

Here, we present Immunoglobulin Language Model (IgLM), a generative language model that leverages bidirectional context for designing antibody sequence spans of varying lengths while training on a large-scale natural antibody dataset. We show that IgLM can generate full-length antibody

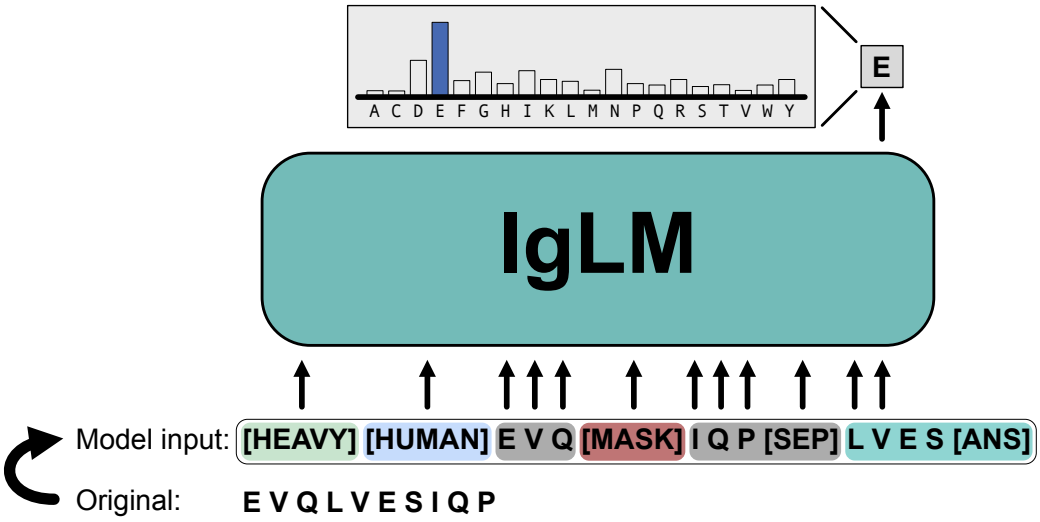


**Figure 6.1:** Overview of IgLM model for antibody sequence generation

sequences conditioned on chain type and species-of-origin. Furthermore, IgLM can diversify loops on an antibody to generate high-quality libraries that display favorable biophysical properties while resembling human antibodies. IgLM should be a powerful tool for antibody discovery and optimization.

## 6.3 Results

### 6.3.1 Immunoglobulin language model

Our method for antibody sequence generation, IgLM, is trained on 558 million natural antibody sequences for both targeted infilling of residue spans, as well as full-length sequence generation. IgLM generates sequences conditioned on the species-of-interest and chain type (heavy or light), enabling controllable generation of antibody sequences.

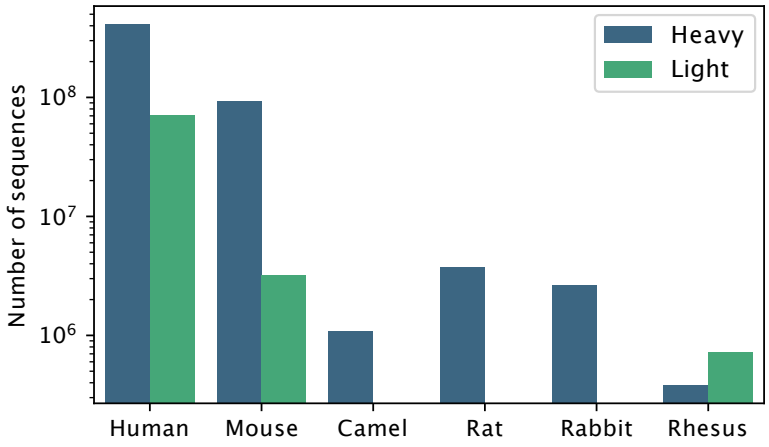


**Figure 6.2:** Distribution of sequences in clustered OAS dataset for various species and chain types

### Infilling language model

Design of antibody libraries typically focuses on diversification of the CDR loop sequences in order to facilitate binding to a diverse set of antigens. Existing approaches to protein sequence generation (including antibodies) typically adopt left-to-right decoding strategies. While these models have proven effective for generation of diverse and functional sequences, they are ill-equipped to re-design specific segments of interest within proteins. To address this limitation, we developed IgLM, an infilling language model for immunoglobulin sequences. IgLM utilizes a standard left-to-right decoder-only transformer architecture (GPT-2), but it is trained for infilling through rearrangement of sequences. Specifically, we adopt the infilling language model formulation from natural language processing [25], wherein arbitrary-length sequence segments (spans) are masked during training and appended to the end of the sequence. By training on these rearranged sequences, models

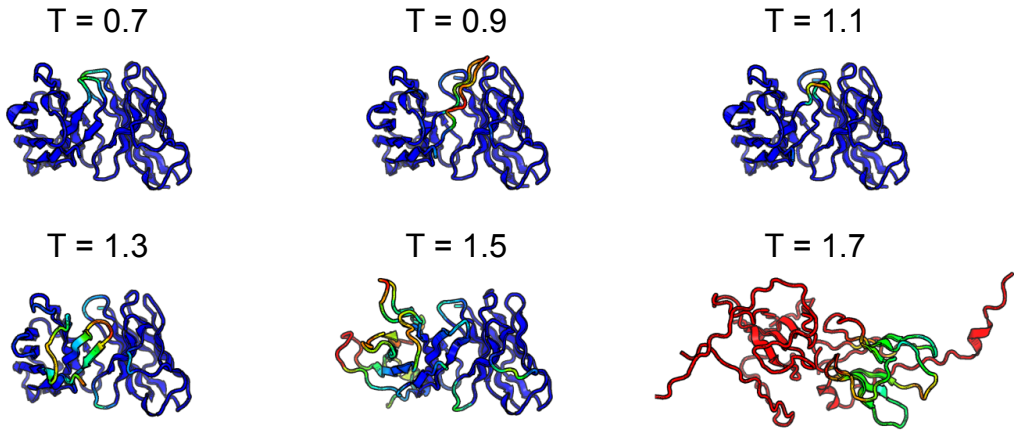


**Figure 6.3:** Effect of increased sampling temperature for full-length generation

Structures at each temperature are predicted by AlphaFold-Multimer and colored by prediction confidence (pLDDT), with blue being the most confident and red being the least.

learn to predict the masked spans conditioned on the surrounding sequence context.

To train IgLM, we collected antibody sequences from the Observed Antibody Space (OAS) [18]. The OAS database contains natural antibody sequences from six species: human, mouse, rat, rabbit, rhesus, and camel. To investigate the impacts of model capacity, we trained two versions of the model: IgLM and IgLM-S, with 13M and 1.4M trainable parameters, respectively. Both IgLM models were trained on a set of 558M non-redundant sequences, clustered at 95% sequence identity. During training, we randomly masked spans of ten to twenty residues within the antibody sequence to enable diversification of arbitrary spans during inference. Additionally, we conditioned sequences on the chain type (heavy or light) and species-of-origin.

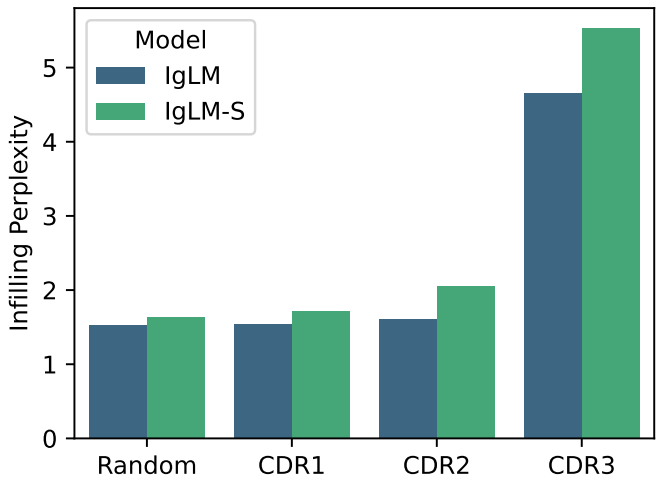


**Figure 6.4:** Infilling perplexity for IgLM heldout test dataset

Infilling capabilities of IgLM and IgLM-S are evaluated on a heldout test dataset. Infilling perplexity is measured for for CDR loops and random spans of 10-20 residues within sequences.

Providing this context enables controllable generation of species-specific antibody sequences. An example of training data construction is illustrated in Figure 6.1. Unless otherwise specified, we use the larger IgLM model for all experiments.

### IgLM generates foldable antibody sequences

As an initial validation of the antibody sequence generation capabilities of IgLM, we conducted a small scale investigation of full-length generation. Specifically, we investigated the impacts of sampling temperature for tuning the diversity of generated sequences. Sampling temperature values above one effectively flatten the amino acid distribution at each step of generation, resulting in more diverse sequences, while temperature below one sharpens

the distribution at each position, resembling a greedy decoding strategy. We generated a set of full-length sequences at temperatures ranging from 0.7 to 1.7, providing the model with human heavy and human light conditioning tags. Because IgLM was trained for sequence infilling, generated sequences contain discontinuous segments of sequence segments, which we simply reordered to produce full-length antibodies. Generated heavy and light chain sequences were paired according to sampling temperature and their structures were predicted using AlphaFold-Multimer [26]. In general, IgLM generates sequences with correspondingly confident predicted structures at lower temperatures (up to 1.3), before beginning to degrade in quality at higher temperatures (Figure 6.3).

**Language modeling evaluation**

We evaluated IgLM as a language model by computing the per-token perplexity for infilled spans within an antibody, which we term the *infilling perplexity*. Because the infilled segment is located at the end of the sequences, computing the infilling perplexity is equivalent to taking the per-token perplexity after the [SEP] token. We compared the infilling perplexity of IgLM and IgLM-S on a heldout test dataset of 30M sequences (Figure 6.4). Results are tabulated by CDR loop, as well as for spans selected randomly within the antibody sequence. As expected, we observe greater perplexity for the CDR loops than the randomly chosen spans, which include the highly conserved framework regions. The CDR3 loop, which is the longest and most diverse, has the highest infilling perplexity. When we compare IgLM and IgLM-S, we observe that IgLM has a lower infilling perplexity for all CDR loops, indicating that the

larger IgLM model (with ten times more parameters) is better at modeling the diversity of antibody sequences.

The diversity of antibody sequences varies by species and chain type. For example, heavy chains introduce additional diversity into their CDR3 loops via D-genes, while some species (e.g., camels) tend to have longer loops. To investigate how these differences impact the performance of IgLM in different settings, we also tabulated the heldout set infilling perplexity by species and chain type. In general, both IgLM models achieve low infilling perplexity for random spans across all species (Figure 6.21). For CDR1 and CDR2 loop infilling, perplexity values are typically lower for human and mouse antibodies, which are disproportionately represented in the OAS database. In general, both models still perform better on these loops than the more challenging CDR3 loops, regardless of species. One exception is for rhesus CDR2 loops, on which IgLM-S performs considerably worse than the larger IgLM model. This appears to be due to poor fitting of rhesus CDR L2 loops, as reflected in the similarity high infilling average perplexity observed when tabulated by chain type (Figure 6.22). The highest infilling perplexity is observed for camel CDR3 loops, which tend to be longer than other species. Across all species and chain types, the larger IgLM model achieves lower infilling perplexity than IgLM-S, suggesting that further increasing model capacity would yield additional improvements.

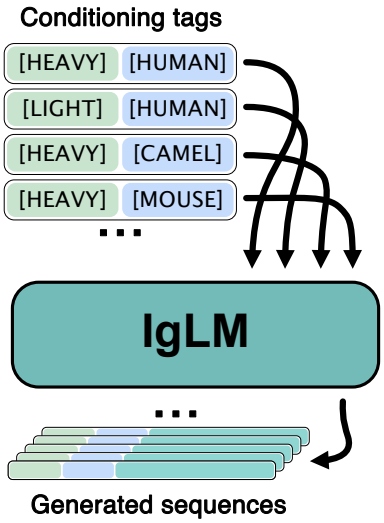


**Figure 6.5:** Diagram of procedure for generating full-length antibody sequences given a desired species and chain type

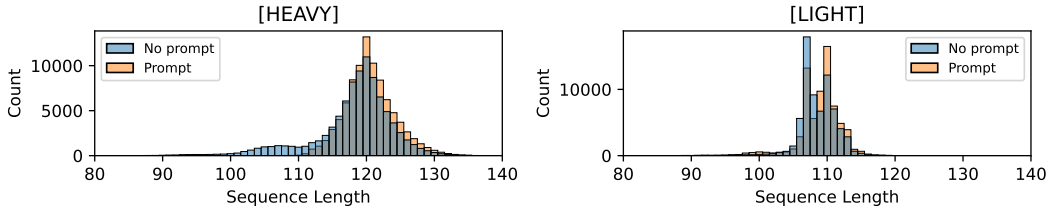


**Figure 6.6:** Effect of residue prompting on full-length sequence generation

engths of generated heeavy and light with and without initial three residues provided (prompting).

### 6.3.2 Controllable generation of antibody sequences

Having demonstrated that IgLM can generate well-formed full-length sequences, we next considered the controllability of IgLM for generating antibody sequences with specific traits. Controllable generation utilizes conditioning tags to provide the model with additional context about the expected sequence.

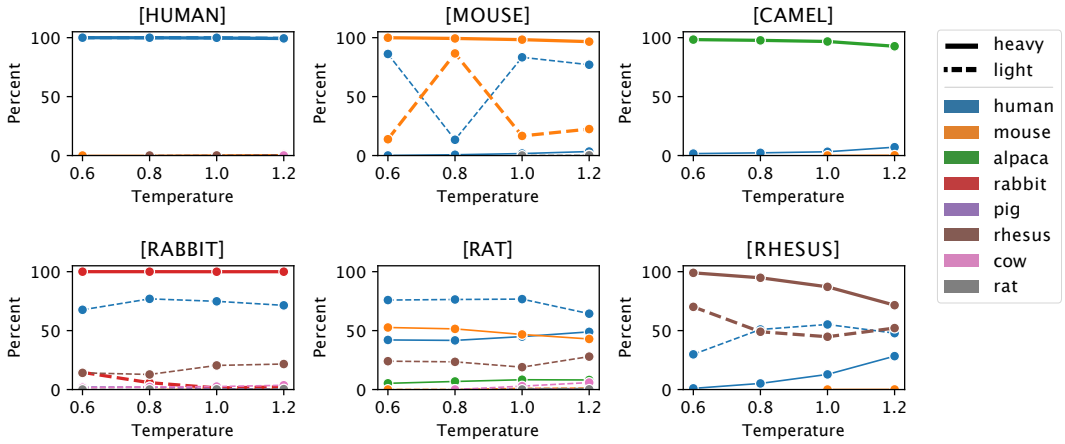


**Figure 6.7:** Adherence of generated sequences to species conditioning tags

Each plot shows the species classifications of antibody sequences generated with a particular species conditioning tag (indicated above plots). Solid and dashed lines correspond to sequences generated with heavy- and light-chain conditioning, respectively.

### Generating species- and chain-controlled sequences

To evaluate the controllability of IgLM, we generated a set of 220K full-length sequences utilizing all viable combinations of conditioning tags, as well as a range of sampling temperatures (Figure 6.5). For every species (except camel), we sampled with both heavy and light conditioning tags. For camel sequence generation, we only sampled heavy chains, as they do not produce light chains. To produce a diverse set of sequences for analysis, we sampled using a range of temperatures ($T \in \{0.6, 0.8, 1.0, 1.2\}$). Sampling under these conditions resulted in a diverse set of antibody sequences. However, we observed that the sequences frequently featured N-terminal truncations, a common occurrence in the OAS database used for training [22]. For heavy chains, these N-terminal deletions appeared as a left-shoulder in the sequence

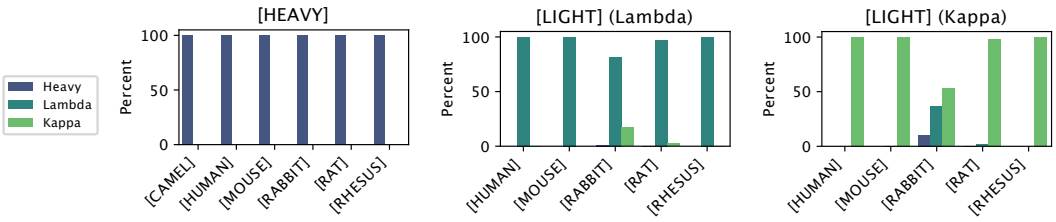


**Figure 6.8:** Adherence of generated sequences to chain conditioning tags

Top plot shows the percentage of heavy-chain-conditioned sequences classified as heavy chains, for each species conditioning tag. Lower plots show the percentage of light-chain-conditioned sequences, further divided by whether initial residues were characteristic of lambda or kappa chains, classified as lambda or kappa chains.

length distribution (Figure 6.6, left) with lengths ranging from 100 to 110 residues. For light chains, we observed a population of truncated chains with lengths between 98 and 102 residues (Figure 6.6, right). To address truncation in generated sequences, we utilized a prompting strategy, wherein we initialize each sequence with a three-residue motif corresponding to the species and chain type tags. Specific intialization sequences are documented in Table 6.3. For both heavy and light chains, prompting with initial residues significantly reduced the population of truncated sequences (Figure 6.6). For the following analysis, we consider only sequences generated with prompting.

### Adherence to conditioning tags

To evaluate the effectiveness of contrallable generation, we considered the agreement between the provided conditioning tags and the sequences produced by IgLM. For each generated sequence, we classified the species (according to V-gene identity) and chain type using ANARCI [27]. We note that the species classes provided by ANARCI diverge in some cases from

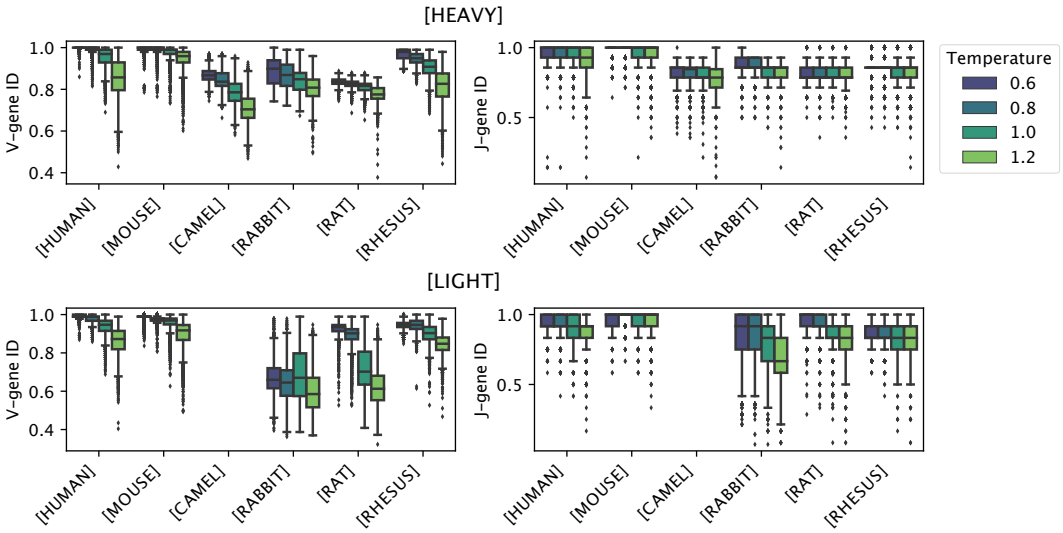


**Figure 6.9:** Sampling temperature controls mutational load on generated sequences

Effect of sampling temperature on germline identity for generated heavy and light chain sequences. As sampling temperature increases, generated sequences diverge from the closest germline V- and J-gene sequences.

those provided by the OAS database, but there was a suitable corresponding class for each conditioning token (e.g., alpaca for [CAMEL]). In Figure 6.7, we show the makeup of sequences for each species conditioning tag, according to sampling temperature. In each plot, the percentage of heavy and light chain sequences classified as each species are indicated by solid and dashed lines, respectively. For most species (human, mouse, camel, rabbit, rhesus), IgLM is able to successfully generate heavy chain sequences at every temperature. The exception to this trend is rat sequences, for which we were unable to produce any sequences that ANARCI classified as belonging to the intended species.

The ability to generate sequences is not directly explained by prevalence in the training dataset, as the model is trained on an order of magnitude more rat heavy chain sequences than rhesus (Table 6.2). IgLM is generally

less effective at generating light chain sequences for most species. With the exception of human light chains, all species have a large proportion of sequences classified as belonging to an unintended species (typically human). For mouse and rhesus light chains, IgLM generates the correct species in 34.89% and 88.14% of cases, respectively (Table 6.4). For rabbit and rat light chains, IgLM was not exposed to any examples during training. Interestingly, despite having seen no such sequences during training, IgLM is capable of generating sequences classified by ANARCI as rabbit light chains for 6.89% of samples (1,120 sequences). The majority of these sequences are cases where the model has instead generated a rabbit heavy chain. However, for 35 of these 1,120 cases, IgLM has produced rabbit light chain sequences. We further investigated the plausibility of these sequences by aligning to the nearest germline sequences assigned by ANARCI with Clustal-Omega [28]. The sequences appear to align well to rabbit germlines, though with conderable mutations to the framework regions (Figure 6.23). To investigate the structural viability of the generated rabbit light chain sequences, we predicted structures with IgFold [21]. All structures were predicted confidently in the framework residues, with the CDR loops being the most uncertain (Figure 6.24). Although rare (35 sequences out of 20,000 attempts), these results suggest that IgLM is capable of generating rabbit light chain sequences despite having never observed such sequences during training. This may be achieved by producing a consensus light chain, with some rabbit-likeness conferred from the heavy chain examples.

We next evaluated the adherence of IgLM-generated sequences to chain

type conditioning tags. In Figure 6.8, we show the percentage of sequences classifed by ANARCI as heavy or light for each conditioning tag. Light chains are further divided into lambda and kappa classes. When conditioned towards heavy chain generation, IgLM effectively produces heavy chains for all species. For light chains, we observe a similar trend, with IgLM producing predominantly light chain sequences for all species. Only for rabbit sequences do we observe a population of heavy chains when conditioning for light chains. As noted above, these are cases where IgLM has instead produced a rabbit heavy chain. When generating light chain sequences, we provide initial residues characteristic of both lambda and kappa chains in equal proportion (Figure 6.3). For most species (except rabbit), the generated sequences are aligned with light chain type indicated by the initial residues. However, as noted above, many of the light sequences for poorly represented species are human-like, rather than resembling the desired species. Interestingly, these results suggest that the chain type conditioning tag is a more effective prior for IgLM than species.

**Sampling temperature controls mutational load**

Increasing sampling temperature has the effect of flattening the probability distribution at each position during sampling, resulting in a greater diversity of sequences. We evaluated the effect of sampling temperature on the diversity of generated sequences by measuring the fractional identity to the closest germline sequences using ANARCI [27]. In Figure 6.9, we show the germline identity for V- and J-genes for each species and chain type. At the lowest sampling temperature ($T = 0.6$), IgLM frequently recapitulates germline

sequences in their entirety for some species (human, mouse, rhesus). As temperature increases, sequences for every species begin to diverge from germline, effectively acquiring mutations. Interestingly, J-gene sequences typically acquire fewer mutations than V-genes for both heavy and light chains. This is likely a reflection of the concentration of CDR loops within the V-gene (CDR1 and CDR2). Only a portion of the CDR3 loop is contributed by the J-gene, with the remaining sequence being conserved framework residues.

### 6.3.3 Therapeutic antibody diversification

Diversification of antibody CDR loops is a common strategy for antibody discovery or optimization campaigns. Through infilling, IgLM is capable of replacing spans of amino acids within antibody sequences, conditioned on the surrounding context. To demonstrate this functionality, we generated infilled libraries for a set of therapeutic antibodies and evaluated several therapeutically relevant properties.

#### Infilled libraries for therapeutic antibodies

To evaluate the utility of infilling with IgLM for diversifying antibody sequences, we created infilled libraries for 49 therapeutic antibodies from Thera-SAbDab [29]. For each antibody, we removed the CDR H3 loop (according to Chothia definitions [30]) and generated a library of infilled sequences using IgLM (Figure 6.10). To produce diverse sequences, we used a combination of sampling temperatures ($T \in \{0.8, 1.0, 1.2\}$) and nucleus sampling probabilities

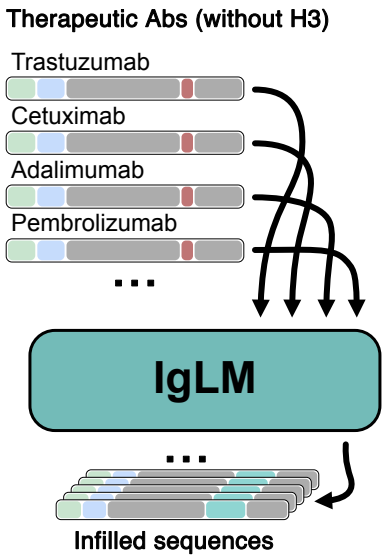


**Figure 6.10:** Procedure for generating therapeutic antibody libraries by infilling CDR H3 loops

($P \in \{0.5, 0.75, 1.0\}$). Nucleus sampling effectively clips the probability distribution at each position during sampling, such that only the most probable amino acids (summing to $P$) are considered. For each of the 49 therapeutic antibodies, we generated one thousand infilled sequences for each combination of $T$ and $P$, totaling nine thousand variants per parent antibody. In Figure 6.13, we show predicted structures (using IgFold [21]) for a subset of ten infilled loops derived from the trastuzumab antibody. The infilled loops vary in length and adopt distinct structural conformations. Across the infilled libraries, we see a variety of infilled CDR H3 loop lengths, dependent on the parent antibody's surrounding sequence context (Figure 6.11). The median length of infilled loops across antibodies ranges from 11 to 16 residues. Interestingly, we observe little impact on the length of infilled loops when varying the sampling temperature and nucleus probabilities (Figure 6.12).

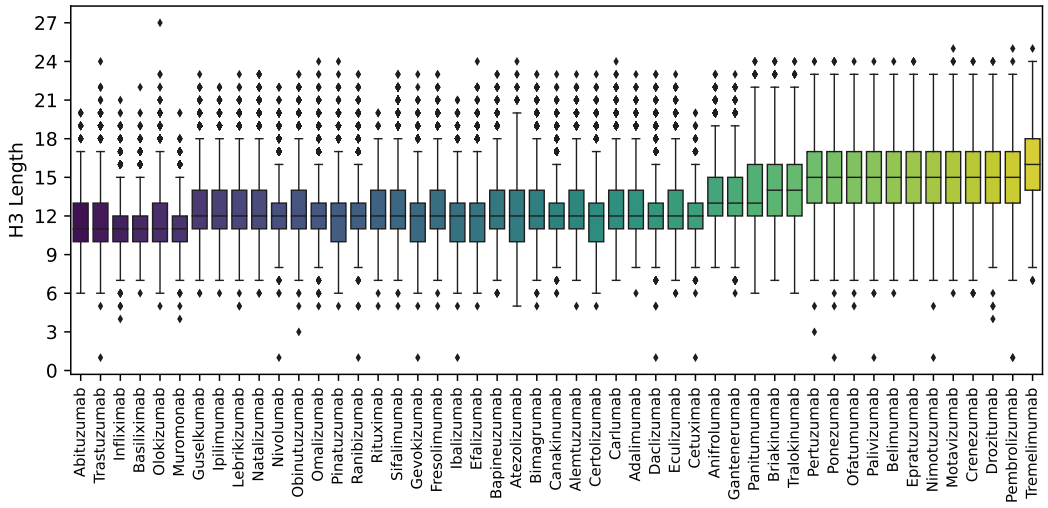


**Figure 6.11:** Distribution of infilled CDR H3 loop lengths for 49 therapeutic antibodies

The distributions of infilled loop lengths vary considerably over the 49 therapeutic antibodies. Because IgLM is trained on natural antibody sequences, we hypothesized that the model may be performing a sort of germline matching, wherein sequences with similar V- and J-genes lead to similar distributions of loop lengths. To test this, we identified the closest germline genes for each antibody with ANARCI [27]. We then group parent antibodies according to common V- and J-gene groups and compared the distributions of infilled loop lengths for each group (Figure 6.14). While there may be some tendency for similar V- and J-genes to lead to similar distributions of infilled loop lengths, we observe considerable variation. This suggests that IgLM is not purely performing germline matching, but rather is considering other properties of the parent antibody.

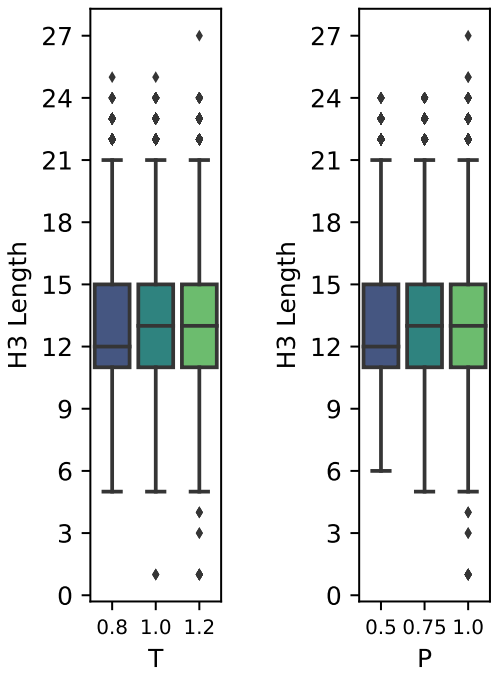


**Figure 6.12:** Effect of sampling parameters on generated CDR H3 loop lengths

Relationship between sampling temperature ($T$) and nucleus probability ($P$) and length of infilled CDR H3 loops.

### Infilling generates diverse loop sequences

Diverse loop libraries are essential for discovering or optimizing sequences against an antigen target. To evaluate the diversity of infilled loops produced by IgLM, we measured the pairwise edit distance between each loop sequence and its closest neighbor amongst the sequences generated with the same sampling parameters. We then compared the diversity of sequences according to loop length and choice of sampling parameters (Figure 6.15). Generally, we observe that generated loops are more diverse at longer lengths, as expected given the increased combinatorial complexity available as more residues are

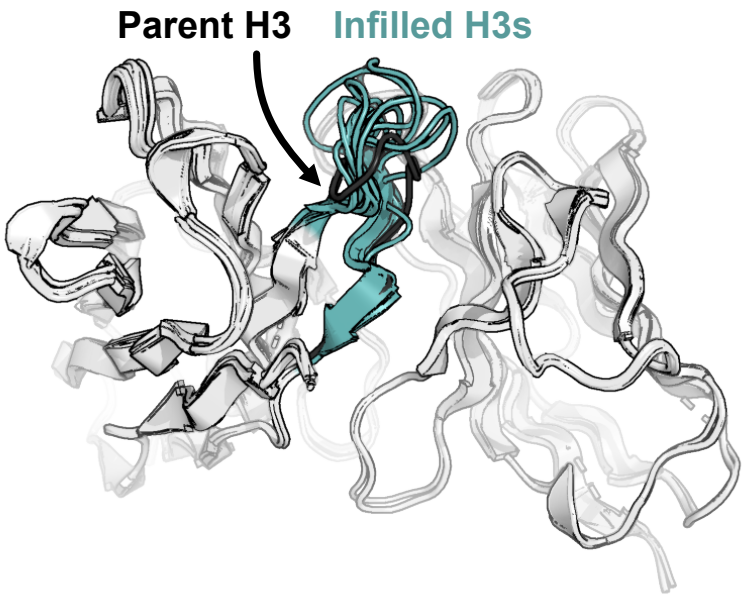


**Figure 6.13:** Structural diversity of infilled CDR H3 loops for trastuzumab

Infilled CDR H3 loops for trastuzumab therapeutic antibody adopt diverse lengths and conformations. Structures for infilled variants are predicted with IgFold.

added. Increasing both sampling temperature and nucleus probability results in a greater diversity of sequences. However, these parameters affect the relationship between length and diversity in distinct ways. For a given loop length, increasing temperature produces more variance in the pairwise edit distance, while increases to nucleus probability provides a more consistent increase in diversity across loop lengths. Indeed, the marginal distribution of pairwise edit distance as nucleus probability is increased produces a much larger shift (Figure 6.15B, marginal) than that of temperature (Figure 6.15A, marginal). In practice, a combination of sampling parameters may be suitable for producing a balance of high-likelihood (low temperature and low nucleus probability) and diverse sequences sequences.

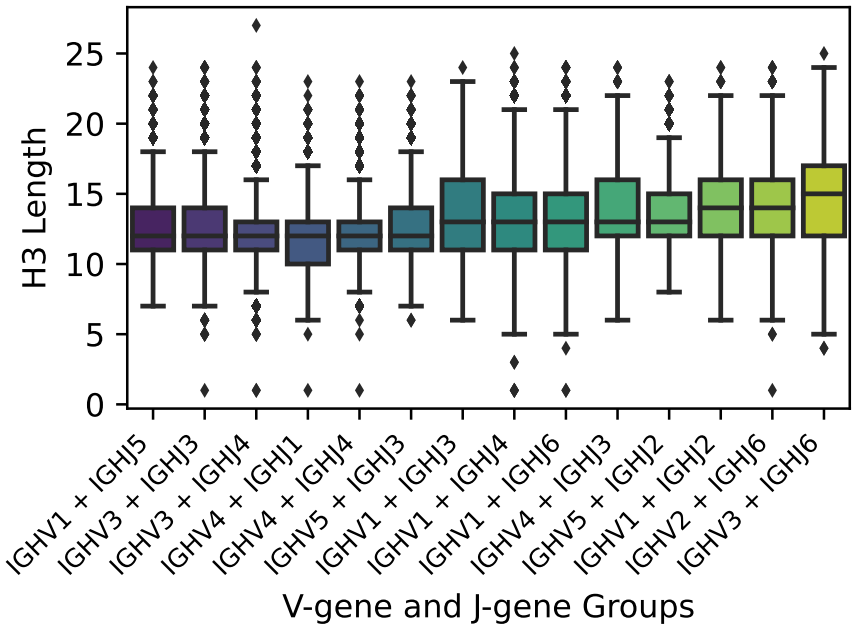


**Figure 6.14:** Germline composition partially determines infilled loop length

Distribution of infilled CDR H3 loop lengths for therapeutic antibodies grouped by nearest germline gene groups.

### Infilled loops display improved developability

Developability encompasses a set physiochemical properties – including aggregation propensity and solubility – that are critical for the success of a therapeutic antibody. Libraries for antibody discovery or optimization that are enriched for sequences with improved developability can allieviate the need for time-consuming post-hoc engineering. To evaluate the developability of sequences produced by IgLM, we used high-throughput computational tools to calculate the aggregation propensity (SAP score [31]) and solubility (CamSol Intrinsic [32]) of the infilled therapeutic libraries. As a precursor to

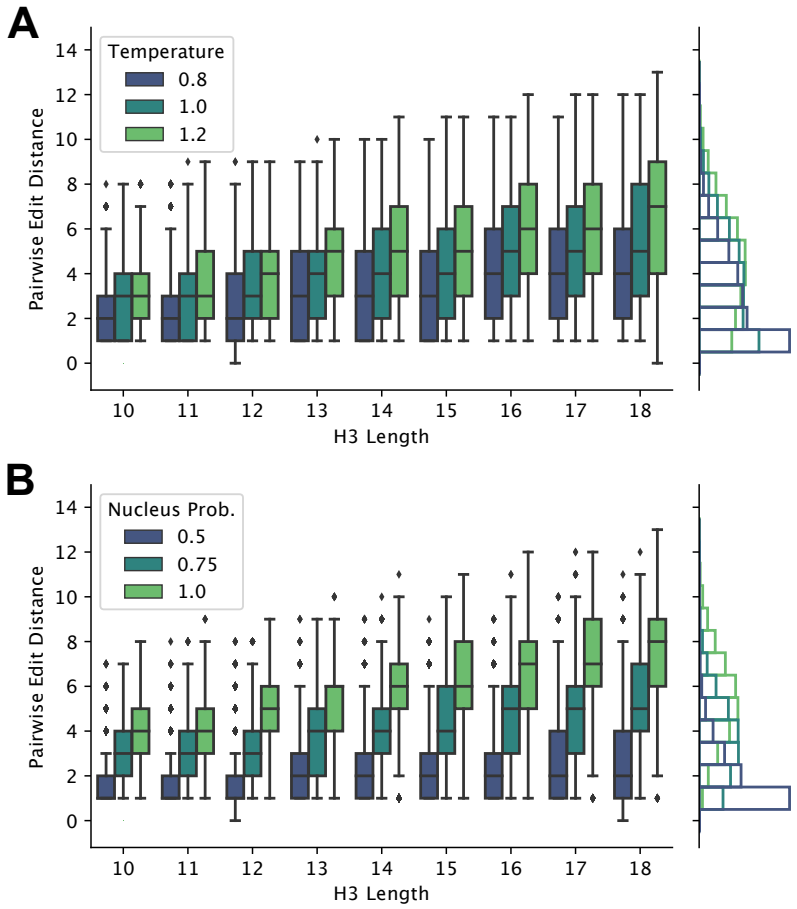


**Figure 6.15:** Effect of sampling parameters on infilled CDR H3 loop lengths

(A-B) Effect of sampling temperature ($T$) and nucleus probability ($P$) on diversity of infilled CDR H3 loops for lengths between 10 and 18 residues. Pairwise edit distance measures the minimum edits between each infilled loop to another in the same set of generated sequences (i.e., within the set of sequences produced with the same $T$ and $P$ parameters). For both parameters, less restrictive sampling produces greater infilled loop diversity.

calculation of aggregation propensity, we used IgFold [21] to predict the structures of the infilled antibodies (including the unchanged light chains). We then compared the aggregation propensities and solubility values of the infilled sequences to those of the parent antibodies. For aggregation propensity, we

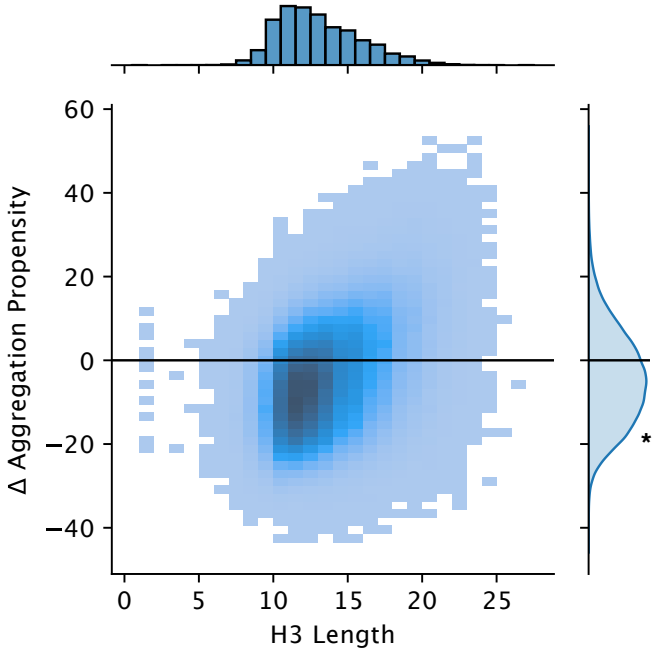


**Figure 6.16:** Change in predicted aggregation propensity of infilled sequences relative to their parent antibodies

Infilled sequences display reduced aggregation propensity (negative is improved), particularly for shorter loops. Asterisks indicate statistical significance ($p < 0.001$) from a one-sample t-test.

observed a significant improvement (negative is better) by infilled sequences over the parent antibodies (Figure 6.16). Similarly for solubility, infilled sequences tend to be more soluble than their parent antibodies (Figure 6.17). In both cases, the largest improvements tend to correspond to the shorter loops. Further, we observe a positive correlation between improvements to aggregation propensity and solubility (Figure 6.18). These results suggest that infilling can be used to generate libraries enriched for sequences with improved developability.

We next investigated whether choice of sampling parameters affects the

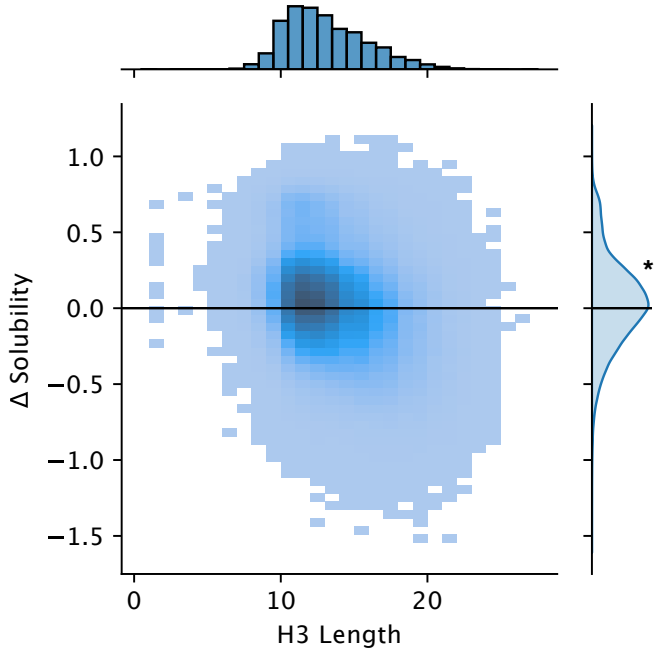


**Figure 6.17:** Change in predicted solubility of infilled sequences relative to their parent antibodies

Infilled sequences display increased solubility (positive is improved). Asterisks indicate statistical significance ($p < 0.001$) from a one-sample t-test.

developability of infilled sequences. When we compared the aggregation propensity and solubility of infilled sequences according to the sampling temperature and nucleus sampling probability, we found marginal practical differences (Figure 6.25). This is likely explained by the relative consistency of infilled loop lengths across sampling parameters (Figure 6.12). These results suggest that developability should not be a concern when tuning the diversity of a generated library.

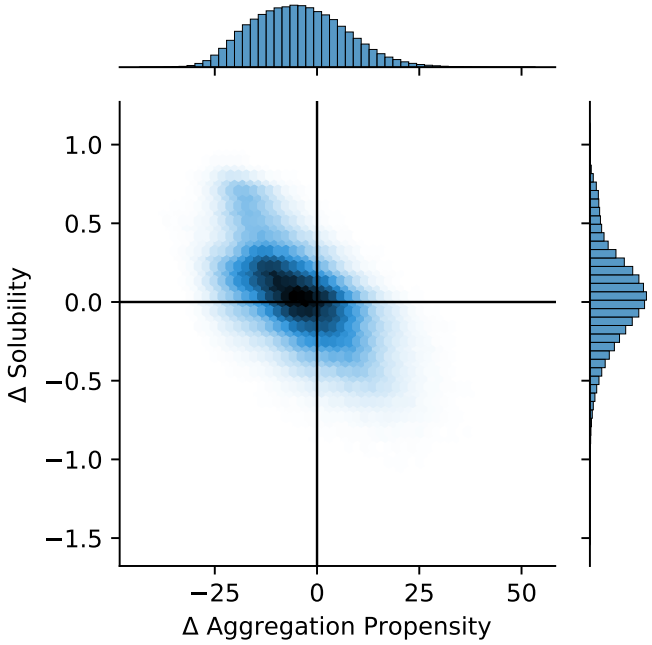


**Figure 6.18:** Relationship between predicted changes in aggregation propensity and solubility for infilled sequence libraries

### Infilled loops are more human-like

Therapeutic antibodies must be human-like to avoid provoking an immune response and to be safe for use in humans. To evaluate the human-likeness of infilled sequences, we calculated the OASis identity (at medium stringency) [19]. OASis divides an antibody sequence into a set of 9-mers and calculates the fraction that have been observed in human repertoires. Thus, higher OASis identity indicates a sequence that is more similar to those produced by humans. When compared to their respective parent antibodies, sequences infilled by IgLM were typically more human-like (Figure 6.19A). This is expected, given that IgLM is trained on natural human antibodies. We also investigated the impact of sampling parameters on the human-likeness of infilled sequences.

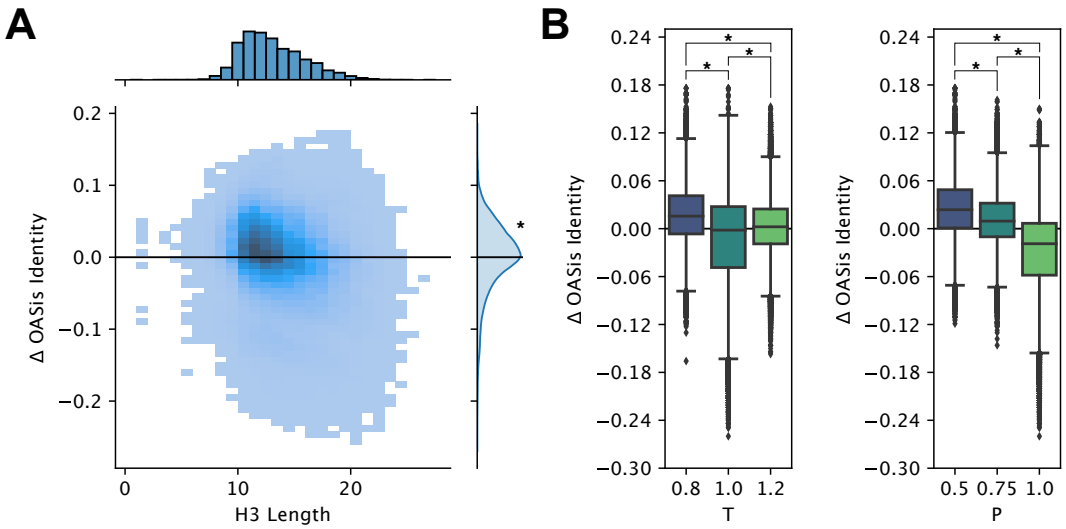


**Figure 6.19:** Change in humanness of infilled sequences relative to their parent antibodies

Asterisks indicate statistical significance ($p < 0.001$) from a one-sample t-test (A) or a two-sample t-test (B). (A) Change in humanness of infilled sequences relative to their parent antibodies. Humanness is calculated as the OASis identity of the heavy chain sequence, with positive larger values being more humanlike. (B) Relationship between sampling temperature ($T$) and nucleus probability ($P$) and change in human-likeness (OASis identity) of infilled heavy chains relative to their parent sequences. (F) Receiver operating characteristic (ROC) curves for human sequence classification methods. The area under the curve (AUC) is shown for each method.

For both sampling temperature and nucleus probability, we find that less restrictive sampling tends to produce less human-like sequences (Figure 6.19B). For practical purposes, this suggests that sampling with lower temperature and nucleus probability may be more suitable when immunogenicity is a concern.

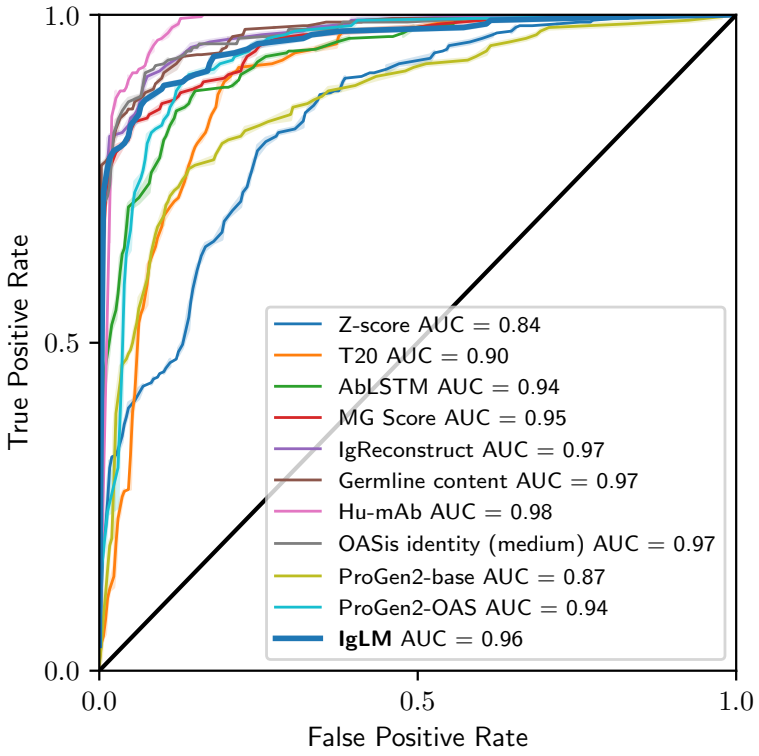


**Figure 6.20:** Evaluation of IgLM for human antibody classification

Receiver operating characteristic (ROC) curves for human sequence classification methods. The area under the curve (AUC) is shown for each method.

### 6.3.4 Sequence likelihood is an effective predictor of humanness

Likelihoods from autoregressive language models trained on proteins have been shown to be effective zero-shot predictors of protein fitness [17, 15]. Antibody-specific language models in particular have been used to measure the "naturalness" of designed sequences [33], a measure related to humanness. To evaluate the effectiveness of IgLM for distinguishing human from non-human antibodies, we utilized the model's likelihood to classify sequences

from the IMGT mAb DB [34]. Sequences in this set span a variety of species (human and mouse) and engineering strategies (e.g., humanized, chimeric, felinized). We considered all sequences not specifically labeled as human to be non-human, and calculated a likelihood (conditioned on human species) for each. All sequences had both a heavy and light chain, for which we calculated separate likelihoods and then multiplied.

We compared the performance of IgLM to that of a number of other methods previously benchmarked by Prihoda et al. [19] using a receiver operating characteristic (ROC) curve (Figure 6.20). The results here for alternative methods are adapted from those presented by Prihoda et al., but with several redundant entries removed to avoid double-counting. We additionally evaluated model likelihoods from ProGen2-base and ProGen2-OAS [15], which are models similar to IgLM that contain significantly more parameters (764M). ProGen2-base is trained on a diverse set of protein sequences, while ProGen2-OAS is trained on a dataset similar to IgLM (OAS clustered at 85% sequence identity). We find that IgLM is competitive with state-of-the-art methods designed for human sequence classification, though not the best. Interestingly, IgLM outperforms ProGen2-OAS (ROC AUC of 0.96 for IgLM vs. 0.94 for ProGen2-OAS), despite having significantly fewer parameters (13M vs. 764M). This may result from the different strategies for constructing training datasets from OAS. By filtering at a less stringent 95% sequence identity, IgLM is likely exposed to a greater proportion of human antibody sequences, which dominate the OAS database. These distinctions highlight the importance of

aligning training datasets with the intended application and suggest that training on only human sequences may further improve performance for human sequence classification.

## 6.4 Discussion

Antibody libraries are a powerful tool for discovery and optimization of therapeutics. However, they are hindered by large fractions of non-viable sequences, poor developability, and immmunogenic risks. Generative language models offer a promising alternative to overcome these challenges through on-demand generation of high-quality sequences. However, previous work has focused entirely on contiguous sequence decoding (N-to-C or C-to-N) [15, 24]. While useful, such models are not well-suited for generating antibody libraries, which vary in well-defined regions within the sequence, and for which changes may be undesirable in other positions. In this work, we presented IgLM, an antibody-specific language model for generation of full-length sequences and infilling of targeted residue spans. IgLM was trained for sequence infilling on 558M natural antibody sequences from six species. During training, we provide the model with conditioning tags that indicate the antibody's chain type and species-of-origin, enabling controllable generation of desired types of sequences.

Concurrent work on autoregressive language models for antibody sequence generation have been trained on similar sets of natural antibody sequences and explored larger model sizes [15]. However, models like ProGen2-OAS are limited in utility for antibody generation and design, as they are

difficult to guide towards generation of specific types of sequences (e.g., species or chain types). Both this work and the ProGen2-OAS paper have utilized prompting strategies to guide model generation towards full-length sequences. While these strategies may help in some cases (particularly to overcome dataset limitations), significantly more residues may need to be provided to guide the model towards a specific sequence type (e.g., human vs rhesus heavy chain). In contrast, by including conditioning information for species and chain type in the model's training, IgLM is able to generate sequences of the desired type without additional prompting. Still, as shown in this work, increasing the capacity of models like IgLM may lead to better performance for sequence infilling (lower perplexity) and scoring (better likelihood estimation), a promising direction for future work.

IgLM's primary innovation is the ability to generate infilled residue spans at specified positions within the antibody sequence. In contrast to traditional generative language models that only consider preceding the residues, this enables IgLM to generate within the full context of region to be infilled. We demonstate the utility of infilling by generating libraries for 49 therapeutic antibodies. We found that IgLM was capable of generating diverse CDR H3 loop sequences, and that diversity was largely tunable by choice of sampling parameters. Further, the infilled libraries possessed desirable developability traits (aggregation propensity, solubility) while being more human-like on average than their parent sequences. Notably, IgLM achieves these improvements over antibodies that are already highly optimized, as all of the parent sequences have been engineered for mass-production and use in humans.

Although we focused on antibody loop infilling in this work, similar strategies may be useful for proteins generally. For example, a universal protein sequence infilling model may be applicable to redesign of contiguous protein active sites or for generating linkers between separate domains for protein engineering.

## 6.5 Methods

### 6.5.1 Infilling formulation

Designing spans of amino acids within an antibody sequence can be formulated as an infilling task, similar to text-infilling in natural langauge. We denote an antibody sequence $A = (a_1, ...a_n)$, where $a_i$ represents the amino acid at position $i$ of the antibody sequence. To design a span of length $m$ starting at position $j$ along the sequence, we first replace the span of amino acids $S = (a_j, ...a_{j+m-1})$ with a single [MASK] token to form a sequence $A_{\setminus S} = (a_1, ...a_{j-1}, [\texttt{MASK}], a_{j+m}, ...a_n)$. To generate reasonable variable-length spans to replace $S$ given $A_{\setminus S}$, we seek to learn a distribution $p(S|A_{\setminus S})$.

We draw inspiration from the Infilling by Language Modeling (ILM) framework proposed for natural language infilling [25] to learn $p(S|A_{\setminus S})$. For assembling the model input, we first choose a span $S$ and concatenate $A_{\setminus S}$, [SEP],$S$, and [ANS]. We additionally prepend conditioning tags $c_c$ and $c_s$ to specific the chain type (heavy or light) and species-of-origin (e.g., human, mouse, etc.) of the antibody sequence. The fully formed sequence of tokens $\mathbf{X}$ for IgLM is:

$$\mathbf{X} = (c_c, c_s, a_1, ...a_{j-1}, [\texttt{MASK}], a_{j+m}, ...a_n, [\texttt{SEP}], a_j, ...a_{j+m-1}, [\texttt{ANS}]) \tag{6.1}$$

We then train a generative model with parameters $\theta$ to maximize $p(\mathbf{X}|\theta)$, which can be decomposed into a product of conditional probabilities:

$$\max_{\grave{}} p(\mathbf{X}|\grave{}) = \max_{\grave{}} \prod_i p(\mathbf{X}_i|\mathbf{X}_{<i}, \grave{}) \tag{6.2}$$

### 6.5.2 Model implementation

The IgLM model uses a modified version of the GPT-2 Transformer decoder architecture [35] as implemented in the HuggingFace Transformers library [36]. We trained two models, IgLM and IgLM-S, for sequence infilling. Hyperparameter details are provided in Table 6.1.

**Table 6.1:** IgLM model hyperparameters.

| | IgLM | IgLM-S |
|---|---|---|
| Number of layers | 4 | 3 |
| Embedding dimension | 512 | 192 |
| Hidden dimension | 512 | 192 |
| Attention heads | 8 | 6 |
| Feed-forward dimension | 2048 | 768 |
| Total parameters | 12,889,600 | 1,439,616 |

### 6.5.3 Antibody sequence dataset

To train IgLM, we collected unpaired antibody sequences rom the Observed Antibody Space (OAS) [18]. OAS is a curated set of over one billion unique

antibody sequences compiled from over eighty immune repertoire sequencing studies. After removing sequences indicated to have potential sequencing errors, we were left with 809M unique antibody sequences. We then clustered these sequences using LinClust [37] at 95% sequence identity, leaving 588M non-redundant sequences. The distribution of sequences corresponding to each species and chain type are documented in Figure 6.2 and Table 6.2. The dataset is heavily skewed towards human antibodies, particularly heavy chains, which make up 70% of all sequences. We held out 5% of sequences as a test set to evaluate model performance. Of the remaining sequences, we used 558M sequences for training and 1M for validation.

### 6.5.4 Model training

During training, for each sequence $A = (a_1, ..., a_n)$ we chose a mask length $m$ uniformly at random from $[10, 20]$ and a starting position $j$ uniformly at random from $[1, n - m + 1]$. We prepended two conditioning tags $c_c$ and $c_s$ denoting the chain type and species-of-origin of each sequence as annotated in the OAS database. Models were trained with a batch size of 512 and 2 gradient accumulation steps using DeepSpeed [38, 39]. Training required approximately 3 days when distributed across 4 NVIDIA A100 GPUs.

## 6.6 Appendix

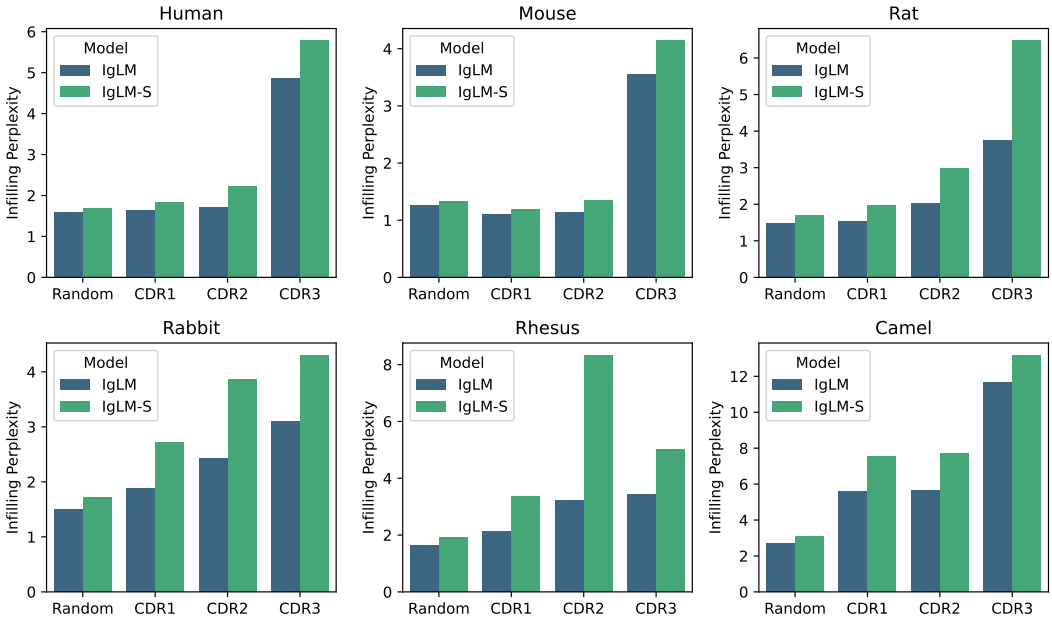


**Figure 6.21:** Infilling perplexity for IgLM and IgLM-S on heldout test dataset of 30M sequences, divided by species-of-origin

Values are reported for CDR loops and random spans of 10-20 residues within sequences.

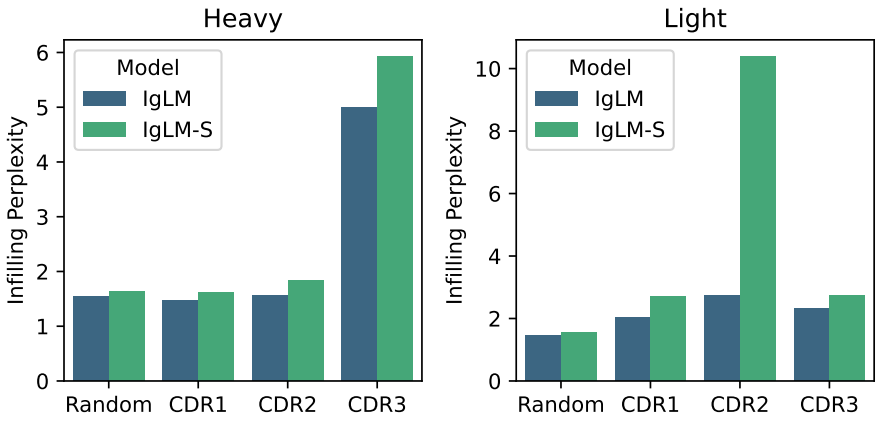


**Figure 6.22:** Infilling perplexity for IgLM and IgLM-S on heldout test dataset of 30M sequences, divided by chain type

Values are reported for CDR loops and random spans of 10-20 residues within sequences.

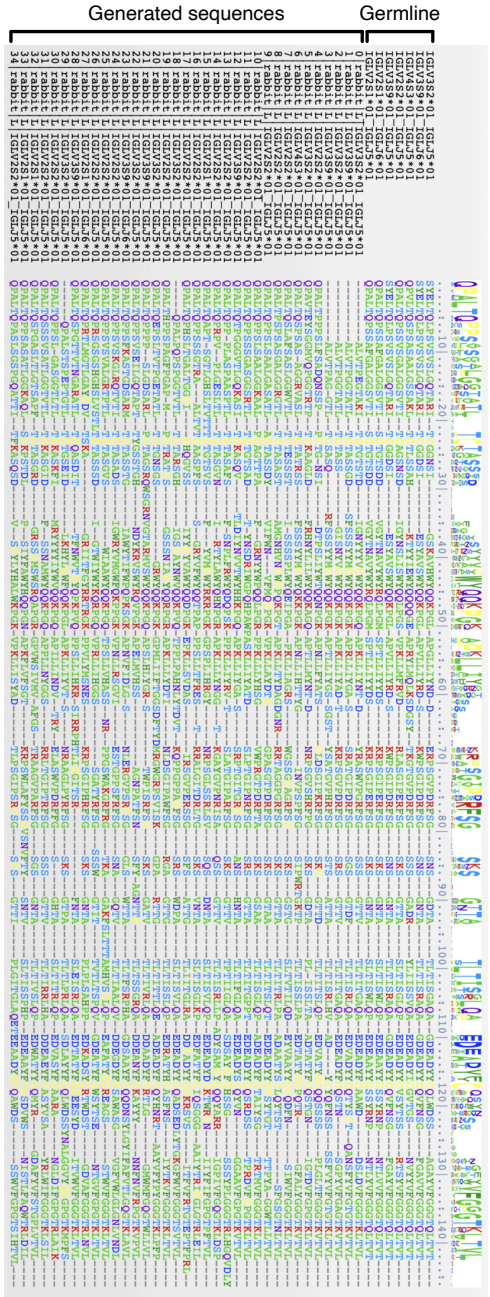


**Figure 6.23:** Alignment of generated rabbit light chain sequences with the closest germline sequences

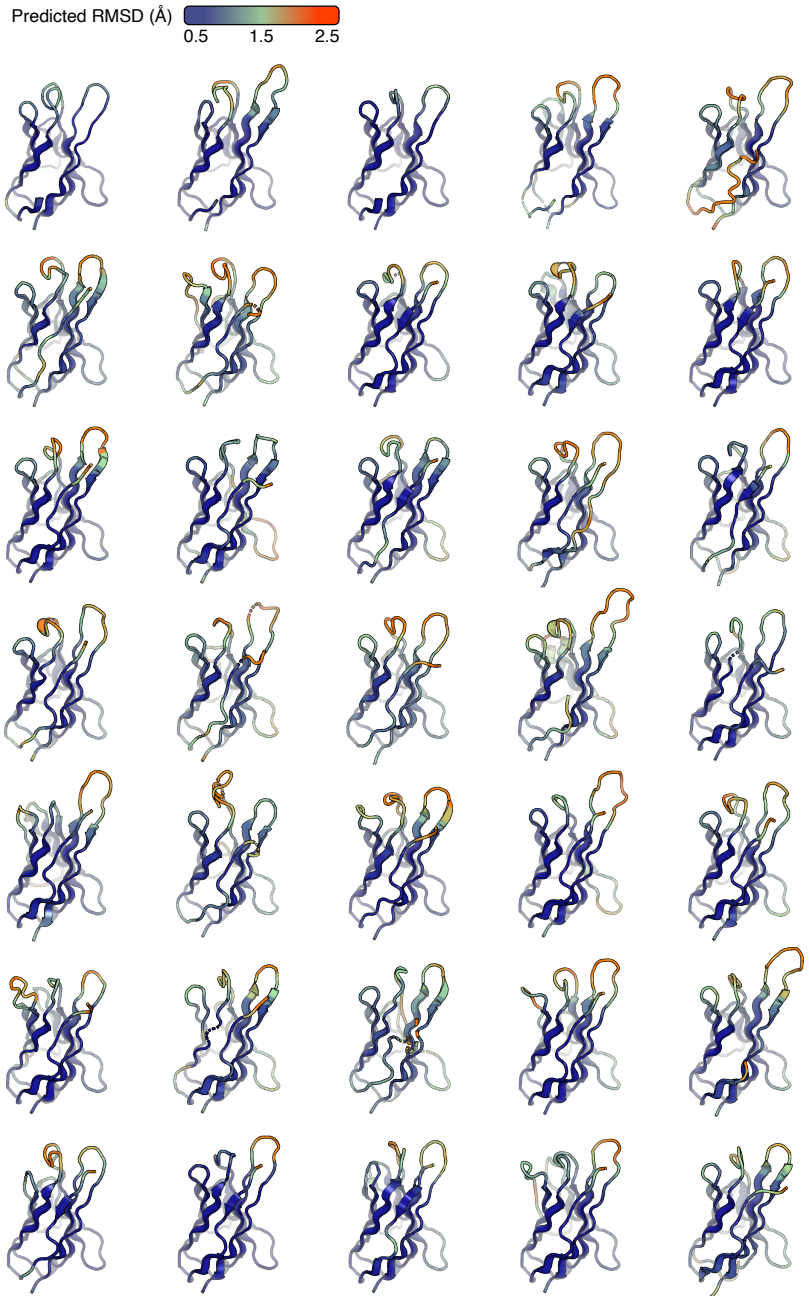


**Figure 6.24:** Prediction of generated rabbit light chain sequences by IgFold

Structures are colored by predicted RMSD from IgFold.

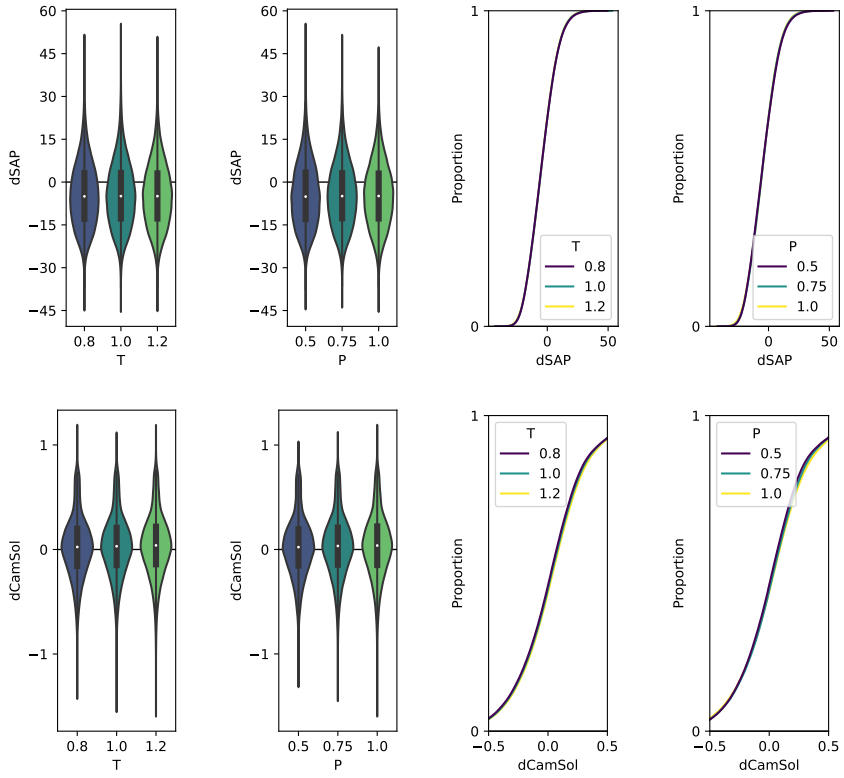


**Figure 6.25:** Impact of sampling parameters on developability of infilled libraries

Library properties are largely unaffected by choice of sampling temperature ($T$) and nucleus sampling probability ($P$).

**Table 6.2:** Distribution of sequences in clustered OAS dataset

| Species | Heavy chains | Light chains | Total |
|---|---|---|---|
| Human | 412,807,447 | 70,584,881 | 483,392,328 |
| Mouse | 93,360,086 | 3,198,407 | 96,558,493 |
| Camel | 1,091,641 | - | 1,091,641 |
| Rat | 3,700,086 | 0 | 3,700,086 |
| Rabbit | 2,644,903 | 0 | 2,644,903 |
| Rhesus | 381,021 | 719,674 | 1,100,695 |
| **Total** | **513,985,184** | **74,502,962** | **588,488,146** |

**Table 6.3:** Full-length sequence generation parameters

| Description | Chain token | Species token | Initial residues | Number generated |
|---|---|---|---|---|
| Human heavy chain | [HEAVY] | [HUMAN] | EVQ | 20,000 |
| Human light chain (lambda) | [LIGHT] | [HUMAN] | QSA | 10,000 |
| Human light chain (kappa) | [LIGHT] | [HUMAN] | DIQ | 10,000 |
| Mouse heavy chain | [HEAVY] | [MOUSE] | QVQ | 20,000 |
| Mouse light chain (lambda) | [LIGHT] | [MOUSE] | QAV | 10,000 |
| Mouse light chain (kappa) | [LIGHT] | [MOUSE] | DIV | 10,000 |
| Camel heavy chain | [HEAVY] | [CAMEL] | QVQ | 20,000 |
| Rabbit heavy chain | [HEAVY] | [RABBIT] | QEQ | 20,000 |
| Rabbit light chain (lambda) | [LIGHT] | [RABBIT] | QPA | 10,000 |
| Rabbit light chain (kappa) | [LIGHT] | [RABBIT] | ALV | 10,000 |
| Rat heavy chain | [HEAVY] | [RAT] | EVQ | 20,000 |
| Rat light chain (lambda) | [LIGHT] | [RAT] | QAV | 10,000 |
| Rat light chain (kappa) | [LIGHT] | [RAT] | GIQ | 10,000 |
| Rhesus heavy chain | [HEAVY] | [RHESUS] | QVQ | 20,000 |
| Rhesus light chain (lambda) | [LIGHT] | [RHESUS] | QSV | 10,000 |
| Rhesus light chain (kappa) | [LIGHT] | [RHESUS] | AIQ | 10,000 |
| **Total** | | | | **220,000** |

**Table 6.4:** Adherence to species conditioning tags for full-length generation

| Chain token | Species token | T = 0.6 | T = 0.8 | T = 1.0 | T = 1.2 | Overall |
|---|---|---|---|---|---|---|
| [HEAVY] | [HUMAN] | 99.98% | 99.94% | 99.66% | 99.30% | 99.72% |
| [HEAVY] | [MOUSE] | 99.94% | 99.34% | 98.32% | 96.62% | 98.55% |
| [HEAVY] | [CAMEL] | 98.36% | 97.72% | 96.76% | 92.72% | 96.39% |
| [HEAVY] | [RABBIT] | 100.00% | 100.00% | 99.96% | 99.98% | 99.98% |
| [HEAVY] | [RAT] | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| [HEAVY] | [RHESUS] | 99.00% | 94.82% | 87.14% | 71.58% | 88.14% |
| [LIGHT] | [HUMAN] | 100.00% | 99.98% | 99.92% | 99.40% | 99.82% |
| [LIGHT] | [MOUSE] | 13.84% | 86.62% | 16.64% | 22.46% | 34.89% |
| [LIGHT] | [RABBIT] | 14.56% | 5.84% | 1.37% | 2.22% | 6.89% |
| [LIGHT] | [RAT] | 0.00% | 0.00% | 0.02% | 0.15% | 0.04% |
| [LIGHT] | [RHESUS] | 70.14% | 49.02% | 44.80% | 51.15% | 54.03% |

Percentage of matches between conditioning tag used for generation and species classification from ANARCI for each generation configuration.

# References


[1] Masami Suzuki, Chie Kato, and Atsuhiko Kato. "Therapeutic antibodies: their mechanisms of action and the pathological findings they induce in toxicity studies". In: *Journal of Toxicologic Pathology* 28.3 (2015), pp. 133–139.

[2] Sachdev S Sidhu and Frederic A Fellouse. "Synthetic therapeutic antibodies". In: *Nature Chemical Biology* 2.12 (2006), pp. 682–688.

[3] John McCafferty, Andrew D Griffiths, Greg Winter, and David J Chiswell. "Phage antibodies: filamentous phage displaying antibody variable domains". In: *Nature* 348.6301 (1990), pp. 552–554.

[4] George P Smith. "Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface". In: *Science* 228.4705 (1985), pp. 1315–1317.

[5] Andrew D Griffiths, Samuel C Williams, Oliver Hartley, IM Tomlinson, P Waterhouse, William L Crosby, RE Kontermann, PT Jones, NM Low, and TJ al Allison. "Isolation of high affinity human antibodies directly from large synthetic repertoires." In: *The EMBO Journal* 13.14 (1994), pp. 3245–3260.

[6] Adriana-Michelle Wolf Pérez, Pietro Sormanni, Jonathan Sonne Andersen, Laila Ismail Sakhnini, Ileana Rodriguez-Leon, Jais Rose Bjelke, Annette Juhl Gajhede, Leonardo De Maria, Daniel E Otzen, Michele Vendruscolo, et al. "In vitro and in silico assessment of the developability of a designed monoclonal antibody library". In: *MAbs*. Vol. 11. 2. Taylor & Francis. 2019, pp. 388–400.

[7] Tushar Jain, Tingwan Sun, Stéphanie Durand, Amy Hall, Nga Rewa Houston, Juergen H Nett, Beth Sharkey, Beata Bobrowicz, Isabelle Caffry, Yao Yu, et al. "Biophysical properties of the clinical-stage antibody landscape". In: *Proceedings of the National Academy of Sciences* 114.5 (2017), pp. 944–949.

[8] Juan C Almagro, Martha Pedraza-Escalona, Hugo Iván Arrieta, and Sonia Mayra Pérez-Tapia. "Phage display libraries for antibody therapeutic discovery and development". In: *Antibodies* 8.3 (2019), p. 44.

[9] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021).

[10] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. "ProtTrans: towards cracking the language of Life's code through self-supervised deep learning and high performance computing". In: *arXiv preprint arXiv:2007.06225* (2020).

[11] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. "ProGen: Language modeling for protein generation". In: *arXiv preprint arXiv:2004.03497* (2020).

[12] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. "Language models enable zero-shot prediction of the effects of mutations on protein function". In: *bioRxiv* (2021).

[13] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. "Language models of protein sequences at the scale of evolution enable accurate structure prediction". In: *bioRxiv* (2022).

[14] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. "ProtGPT2 is a deep unsupervised language model for protein design". In: *Nature Communications* 13.1 (2022), pp. 1–10.

[15] Erik Nijkamp, Jeffrey Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. "ProGen2: exploring the boundaries of protein language models". In: *arXiv preprint arXiv:2206.13517* (2022).

[16] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. "Large language models generate functional protein sequences across diverse families". In: *Nature Biotechnology* (2023), pp. 1–8.

[17] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. "RITA: a Study on Scaling Up Generative Protein Sequence Models". In: *arXiv preprint arXiv:2205.05789* (2022).

[18] Aleksandr Kovaltsuk, Jinwoo Leem, Sebastian Kelm, James Snowden, Charlotte M Deane, and Konrad Krawczyk. "Observed Antibody Space: a resource for data mining next-generation sequencing of antibody repertoires". In: *The Journal of Immunology* 201.8 (2018), pp. 2502–2509.

[19] David Prihoda, Jad Maamary, Andrew Waight, Veronica Juan, Laurence Fayadat-Dilman, Daniel Svozil, and Danny A Bitton. "BioPhi: a platform for antibody design, humanization, and humanness evaluation based on natural antibody repertoires and deep learning". In: *MAbs*. Vol. 14. 1. Taylor & Francis. 2022, p. 2020203.

[20] Jeffrey A Ruffolo, Jeffrey J Gray, and Jeremias Sulam. "Deciphering antibody affinity maturation with language models and weakly supervised learning". In: *arXiv preprint arXiv:2112.07782* (2021).

[21] Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. "Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies". In: *bioRxiv* (2022), pp. 2022–04.

[22] Tobias H Olsen, Iain H Moal, and Charlotte M Deane. "AbLang: an antibody language model for completing antibody sequences". In: *Bioinformatics Advances* 2.1 (2022), vbac046.

[23] Rahmad Akbar, Philippe A Robert, Cédric R Weber, Michael Widrich, Robert Frank, Milena Pavlović, Lonneke Scheffer, Maria Chernigovskaya, Igor Snapkov, Andrei Slabodkin, et al. "In silico proof of principle of machine learning-based antibody design at unconstrained scale". In: *Mabs*. Vol. 14. Taylor & Francis. 2022, p. 2031482.

[24] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. "Protein design and variant prediction using autoregressive generative models". In: *Nature Communications* 12.1 (2021), pp. 1–11.

[25] Chris Donahue, Mina Lee, and Percy Liang. "Enabling language models to fill in the blanks". In: *arXiv preprint arXiv:2005.05339* (2020).

[26] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew W Senior, Timothy Green, Augustin Žídek, Russell Bates, Sam Blackwell, Jason Yim, et al. "Protein complex prediction with AlphaFold-Multimer". In: *bioRxiv* (2021).

[27] James Dunbar and Charlotte M Deane. "ANARCI: antigen receptor numbering and receptor classification". In: *Bioinformatics* 32.2 (2016), pp. 298–300.

[28] Fabian Sievers and Desmond G Higgins. "Clustal Omega, accurate alignment of very large numbers of sequences". In: *Multiple Sequence Alignment Methods*. Springer, 2014, pp. 105–116.

[29] Matthew IJ Raybould, Claire Marks, Alan P Lewis, Jiye Shi, Alexander Bujotzek, Bruck Taddese, and Charlotte M Deane. "Thera-SAbDab: the therapeutic structural antibody database". In: *Nucleic Acids Research* 48.D1 (2020), pp. D383–D388.

[30] Cyrus Chothia and Arthur M Lesk. "Canonical structures for the hypervariable regions of immunoglobulins". In: *Journal of Molecular Biology* 196.4 (1987), pp. 901–917.

[31] Naresh Chennamsetty, Vladimir Voynov, Veysel Kayser, Bernhard Helk, and Bernhardt L Trout. "Prediction of aggregation prone regions of therapeutic proteins". In: *The Journal of Physical Chemistry B* 114.19 (2010), pp. 6614–6624.

[32] Pietro Sormanni, Francesco A Aprile, and Michele Vendruscolo. "The CamSol method of rational design of protein mutants with enhanced solubility". In: *Journal of Molecular Biology* 427.2 (2015), pp. 478–490.

[33] Sharrol Bachas, Goran Rakocevic, David Spencer, Anand V Sastry, Robel Haile, John M Sutton, George Kasun, Andrew Stachyra, Jahir M Gutierrez, Edriss Yassine, et al. "Antibody optimization enabled by artificial intelligence predictions of binding affinity and naturalness". In: *bioRxiv* (2022).

[34] C Poiron, Y Wu, C Ginestoux, F Ehrenmann, P Duroux, and MP Lefranc. "IMGT/mAb-DB: the IMGT® database for therapeutic monoclonal antibodies". In: *Poster no101* 11 (2010).

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[36] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. "Transformers: State-of-the-art natural language processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2020, pp. 38–45.

[37] Martin Steinegger and Johannes Söding. "Clustering huge protein sequence sets in linear time". In: *Nature Communications* 9.1 (2018), pp. 1–8.

[38] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. "Zero: Memory optimizations toward training trillion parameter models". In: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2020, pp. 1–16.

[39] Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. "Zero-offload: Democratizing billion-scale model training". In: *arXiv preprint arXiv:2101.06840* (2021).

# Chapter 7

# Discussion and Conclusion

Antibodies are critical immune molecules that recognize and facilitate neutralization of a broad range of pathogens. The specific binding of antibodies that is enabled by the hypervariable loops has made them an effective tool for therapeutic and diagnostic applications. However, despite the biological and medical importance of antibodies, they remain difficult to model and challenging to design. Application of machine learning to the vast space of protein data explored by nature has driven advances in structure prediction and design. However, many of these advances are enabled by capturing and reproducing broad evolutionary patterns. Antibodies, which evolve rapidly in response to antigenic pressure within individuals, are not subject to the same evolutionary pressures. As such, existing tools are poorly suited to the challenges of antibody structure prediction and design.

## 7.1 My contributions

I began my doctoral studies at the dawn of a machine learning revolution in protein modeling. Inspired by the progress achieved in protein structure prediction [1] and sequence modeling [2], yet clear-eyed of their limitations, I set out to develop a set of antibody-specific tools to improve our understanding of these critical moleucles. In this dissertation, I presented a series of machine learning tools designed to model increasingly complex aspects of antibody structure, with increasing accuracy and speed. Next, I presented two projects aimed at generation of antibody sequences (and proteins more broadly).

Antibody structure prediction is a challenging problem. Despite the similar high-level goal of accurately predicting three-dimensional atomic coordinates, the distinct structural nuances of antibodies highlight many of the limitations of current generalist tools. Foremost is the irregularity of structural variation. For antibodies, the tertiary fold is entirely pre-determined, save for a set of variable-length loops. This stands in stark contrast to the more general problem of protein structure prediction, where the tertiary fold is largely unknown. Further, the co-evolutionary signals driving general protein structure predictors are largely irrelevant to antibodies, in part because of their distinct evolutionary environment, but also because of their function as binders to a complementarity antigen. To begin addressing these shortcomings, I developed three tools for antibody structure prediction: DeepH3 [3], DeepAb [4], and IgFold [5].

Prior to DeepH3 [3], the state-of-the-art methods for prediction of CDR H3 loops relied on grafting (which is limited by poor templates) [6, 7] or extensive

sampling (which is limited by poor energy functions and high computational cost) [8]. With DeepH3, I aimed to address the latter scenario by learning a sequence specific energy function for CDR H3 loops. DeepH3 formulated this task as the prediction of inter-residue distances and orientation potentials, which were used in substitue of the Rosetta energy function for structure scoring. By making more effective use of previously solved structures, DeepH3 was able to significantly outperform RosettaAntibody [8] at scoring structural decoys. This improvement also had the indirect effect of reducing the computational cost of structure prediction (by an order of magnitude), as the improved scoring function allowed for more efficient identification of native-like loops.

The limited scope of DeepH3, in that it only rebuilt CDR H3 loops given the surrounding $F_V$ structure, made it reliant on other tools for complete functionality. With DeepAb [4], I extended the model to full $F_V$ structure prediction. DeepAb built on a similar ResNet [9] architecture to DeepH3, but increased the number of output potentials predicted and incorporated novel architectural improvements to increase accuracy and interpretability. The first such improvement was the use of pretraining on natural paired antibody sequences from the Observed Antibody Space (OAS) database [10]. Embeddings from a sequence-to-sequence encoder-decoder model exposed the model to nearly two orders of magnitude more antibody sequences than there were structures for training. The second improvement involved incorporation of an efficient attention mechanism to improve prediction of output features. Not only did addition of this attention layer improve accuracy, it also allowed

for the identification of important residues for predicting CDR H3 loops. Together, these improvements allowed DeepAb to significantly outperform its contemporaries across all six CDR loops. Most notably, DeepAb reduced the RMSD of CDR H3 loop structure predictions by 0.5 Å, a 20% improvement over the next best method.

While DeepAb established deep learning as a compelling alternative to traditional approaches for antibody structure prediction, it was ultimately quite rigid in its application. DeepAb was unable to incorporate known structural information into its predictions, was opaque with respect to expected accuracy on individual outputs, and performed poorly on nanobodies (single chain antibodies). To address these shortcomings, and to create a more universal antibody structure predictor, I developed IgFold [5]. IgFold was designed to be a Swiss Army Knife for antibody structure prediction. The foundation of IgFold is AntiBERTy [11], a transformer encoder model that I trained on 558 million natural antibody sequences. Using embeddings from AntiBERTy, IgFold directly predicts the backbone atomic coordinates of antibody $F_V$ structures. This end-to-end training regime allows IgFold to predict structures significantly faster than DeepAb, while improving accuracy over alternative methods (including AlphaFold [12, 13]). IgFold is able to incorporate known structural information into its predictions in the form of templates, making it a more flexible tool for antibody engineering. Along with its predictions, IgFold provides a per-residue confidence score that can be used to identify regions of uncertainty. Finally, IgFold is trained on a mixture of paired and single-chain structures, extending its use beyond conventional antibodies.

If antibody structure prediction represented the first thrust of my graduate research, protein language models would make up the second. Protein sequences are a very natural modality for training and studying language models (perhaps better than natural human language). This is because proteins, as physical objects, are constrained by the laws of physics. One of the key innovations of modern language modeling is the attention operation [14], which allows the model to learn relationships between pairs of tokens (e.g., amino acids or words). While for natural language these relationships carry abstract meaning, in proteins they are often physical (and even functional) relationships. In the final two projects described in this dissertation, I presented my work on a pair of generative protein language models: ProGen2 [15] and IgLM [16].

In a collaboration with Salesforce Research, I contributed to the development of ProGen2 [15]. ProGen2, or rather the suite of ProGen2 models, sought to investigate the impacts of training increasingly massive language models for protein generation (including the largest such model to date, with 6.4 billion parameters). Among these models is ProGen2-OAS, a model trained on immune repertoire sequences for generation of antibodies. We showed that ProGen2 can effectively generate protein sequences that span the diversity of natural folds, while diverging significantly in sequence. Through finetuning of ProGen2, we demonstrate deliberate generation of a single protein architecture. Beyond generation, we demonstrate that autoregressive models are effective predictors of protein fitness, a broad notion of protein functions and properties ranging from enzymatic activity to thermal stability. In contrast to

prior work in the field, we reported a divergence in the benefits of scaling protein language models for fitness prediction. Specifically, we found that larger models were more effective at predicting fitness for mutational landscapes with high sequence diversity, while smaller models were more effective at ranking the fitness of more similar sequences. For antibody specific tasks, we showed that pretraining on more antibodies (in the form of immune repertoires) does not result in improved antibody fitness prediction (for binding, stability, or expression). These results highlight the importance of considering the specific task at hand when designing protein language models.

Aside from scaling model parameters, another means of producing useful language models is specialization. IgLM [16] was designed specifically for antibody sequence generation. During training, IgLM is provided with conditioning tokens indicating the species and chain type that should be generated. In practice, this enables controllable generation of sequences from several species. Going beyond traditional N- to C-terminal generation, IgLM learns to fill in internal segments of antibodies (such as CDR loops) through rearrangement of sequence segments during training. This infilling capability has strong implications for antibody library design, as it enables IgLM to diversify specified segments of antibody sequence while conforming to the surrounding context and the natural sequence space of human antibodies. I validated the utility of this approach by creating infilled libraries for 49 clinically approved therapeutic antibodies and conducting extensive computational validation of their properties. The resulting libraries were shown to be diverse in sequence, have improved developability profiles (lower aggregation propensity, higher

solubility), and be more human-like than their parents. These results are particularly promising given that the parent sequences have already undergone extensive optimization on their route to clinical use.

## 7.2 Future directions

Looking forward, I believe there is incredible potential for machine learning to continue to transform antibody modeling and design, as well as the study of proteins more broadly. I will focus on three areas that I believe will particularly impactful in the coming years: flexible structural modeling, escaping the bottlenecks of co-evolution, and generative modeling of proteins.

### 7.2.1 Flexible structural modeling

The current paradigm in protein structure prediction views proteins as static objects, void of the dynamics and flexibility that are intrinsic to their function. While this paradigm has served us well, it is increasingly clear that it places an upper limit on the utility of our models. For example, using current tools it is not possible to sample the conformational space of a protein or to predict the effect of a mutation on protein dynamics. While some of these limitations can be partially addressed by manipulating existing models like AlphaFold2 [17, 18], these are not ideal solutions. In the case of antibodies, dynamics are of particular importance for long CDR loops, which can undergo significant conformational change [19]. Perhaps not by coincidence, these are the types of loops that are most difficult to predict with current methods. In the future, I believe that we must transition towards models that predict a distribution over

protein structures. This will allow us to directly model the conformational space of proteins, and to sample from it. I foresee two main obstacles to this transition in the short term: the need for large datasets of conformationally diverse protein structures and the need for new ways of representing proteins. The first of these challenges may be indirectly addressed by recently released large-scale datasets of predicted protein structures [20, 21], which likely contain a significant amount of conformational diversity (though this remains to be proven). The second challenge is more fundamental, and will require new ways of representing proteins that are more flexible than the current atom-centric approaches. A promising step in this direction is the development of a decorrelated representation space for protein structure generation [22].

### 7.2.2 Escaping the bottlenecks of co-evolution

Many, if not all, of the significant advances in protein structure prediction over the last decade have involved improved harnessing of co-evolutionary information. Beginning with statistical analysis of residue co-variation [23, 24] and culminating in the development of deep learning models that explicitly take sequence alignments as input [12], the powerful co-evolutionary signal has ushered in an era of highly accurate protein structure prediction. However, the co-evolutionary signal is not without its limitations. For example, it is absent in proteins without known homologs. In addition, co-evolutionary models are not well suited to the prediction of protein dynamics, which confound residue co-variation. Finally, and closer to the theme of this dissertation, such models are generally unable to predict antibody-antigen complexes [13],

which have a disjoint co-evolutionary signal between binding partners. One response to these issues has been the development of models that take only a single sequence as input and rely on a pretrained language model to inform structure prediction [21, 25]. While this is a promising direction, such models are likely front-loading the process of learning co-evolutionary dependencies during pretraining rather than obtaining any distinct insights into the sequence-structure relationship. As an alternative, I believe that we must seek an escape from the bottleneck of co-evolution. This will require a shift in focus from the co-evolutionary signal to the underlying sequence-structure relationship. While historically building such models has been infeasible, the large-scale databases [20, 21] mentioned above may provide utility here as well. In particular, due to the size of these datasets, it should be possible to train models on clustered (yet still large) sets of proteins that share little co-evolutionary information. This would allow us to emphasize the relationship between sequence and structure with more examples than the PDB [26] has to offer, hopefully without the shortcut of co-evolutionary information. To benchmark such a model, antibody-antigen complexes could be used as a test set, as they are known to be difficult for co-evolutionary models [13] due to the one-sidedness of the co-evolutionary signal (antibodies mature in response to mostly static antigen, while antigen evolves outside the context of the individual immune response).

### 7.2.3 Generative modeling of proteins

The final area I will discuss is generative modeling of proteins. Protein design is, at its core, a task in sampling new proteins to meet some specification. Traditionally, this specification took the form of a particular fold designed by experts [27]. With the development of differentiable models for sequence-to-structure prediction, this satisfaction of this specification could be achieved through hallucination or gradient-based sequence optimization [28, 29, 30]. However, these approaches are not ideal, as they operate outside the trained context of their respective models and frequently produce unrealistic protein designs. In recent years, numerous approaches have been proposed for direct generative modeling of proteins [31, 32, 33]. These approaches largely focus on capturing the distribution of protein sequence (e.g., language models) or structure (e.g., diffusion models), but not both. The utility of language models has been illustrated throughout this dissertation, and they will undoubtedly continue to be a powerful tool for protein design. In a recent application, language models were used to generate lysozymes that matched the functional activity of natural proteins, while sharing only 31.4% sequence identity to any previously observed protein [34]. Current protein language models are largely identical to their counterparts in natural language processing. While these off-the-shelf models are in many ways well-suited for protein tasks, consideration of the ways in which they are not would be a worthy endeavor. Diffusion models have recently shown immense promise across a diverse array of design tasks [35, 22, 36]. These models seek to learn the function $\nabla \log P(x)$ (denoted the score) for a data distribution, which can be used to

generate new samples through solution of stochastic differential equations [37]. Successful applications of protein diffusion models have shown promise for unconditional generation of protein monomers and complexes [22], design and scaffolding of binding motifs [36], and antigen-specific antibody design [38]. While these models are promising, one major limitation is the separation of sequence and structure. In the future, I believe that we must seek to unify the generative process over sequence and structure, rather than treat one as a latent variable of the other.

# References


[1] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792 (2020), pp. 706–710.

[2] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118.

[3] Jeffrey A Ruffolo, Carlos Guerra, Sai Pooja Mahajan, Jeremias Sulam, and Jeffrey J Gray. "Geometric potentials from deep learning improve prediction of CDR H3 loop structures". In: *Bioinformatics* 36.Supplement_1 (2020), pp. i268–i275.

[4] Jeffrey A Ruffolo, Jeremias Sulam, and Jeffrey J Gray. "Antibody structure prediction using interpretable deep learning". In: *Patterns* 3.2 (2022), p. 100406.

[5] Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. "Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies". In: *bioRxiv* (2022), pp. 2022–04.

[6] Jinwoo Leem, James Dunbar, Guy Georges, Jiye Shi, and Charlotte M Deane. "ABodyBuilder: Automated antibody structure prediction with data–driven accuracy estimation". In: *MAbs*. Vol. 8. 7. Taylor & Francis. 2016, pp. 1259–1268.

[7] Dimitri Schritt, Songling Li, John Rozewicki, Kazutaka Katoh, Kazuo Yamashita, Wayne Volkmuth, Guy Cavet, and Daron M Standley. "Repertoire Builder: high-throughput structural modeling of B and T cell receptors". In: *Molecular Systems Design & Engineering* 4.4 (2019), pp. 761–768.

[8] Brian D Weitzner, Jeliazko R Jeliazkov, Sergey Lyskov, Nicholas Marze, Daisuke Kuroda, Rahel Frick, Jared Adolf-Bryfogle, Naireeta Biswas, Roland L Dunbrack Jr, and Jeffrey J Gray. "Modeling and docking of antibody structures with Rosetta". In: *Nature Protocols* 12.2 (2017), pp. 401–416.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[10] Aleksandr Kovaltsuk, Jinwoo Leem, Sebastian Kelm, James Snowden, Charlotte M Deane, and Konrad Krawczyk. "Observed Antibody Space: a resource for data mining next-generation sequencing of antibody repertoires". In: *The Journal of Immunology* 201.8 (2018), pp. 2502–2509.

[11] Jeffrey A Ruffolo, Jeffrey J Gray, and Jeremias Sulam. "Deciphering antibody affinity maturation with language models and weakly supervised learning". In: *arXiv preprint arXiv:2112.07782* (2021).

[12] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[13] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew W Senior, Timothy Green, Augustin Žídek, Russell Bates, Sam Blackwell, Jason Yim, et al. "Protein complex prediction with AlphaFold-Multimer". In: *bioRxiv* (2021).

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[15] Erik Nijkamp, Jeffrey Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. "Progen2: exploring the boundaries of protein language models". In: *arXiv preprint arXiv:2206.13517* (2022).

[16] Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. "Generative Language Modeling for Antibody Design". In: *bioRxiv* (2021).

[17] Diego Del Alamo, Davide Sala, Hassane S Mchaourab, and Jens Meiler. "Sampling alternative conformational states of transporters and receptors with AlphaFold2". In: *Elife* 11 (2022), e75751.

[18] Hannah K Wayment-Steele, Sergey Ovchinnikov, Lucy Colwell, and Dorothee Kern. "Prediction of multiple conformational states by combining sequence clustering with AlphaFold2". In: *bioRxiv* (2022), pp. 2022–10.

[19] Monica L Fernández-Quintero, Johannes Kraml, Guy Georges, and Klaus R Liedl. "CDR-H3 loop ensemble in solution–conformational selection upon antibody binding". In: *MAbs*. Vol. 11. 6. Taylor & Francis. 2019, pp. 1077–1088.

[20] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models". In: *Nucleic Acids Research* 50.D1 (2022), pp. D439–D444.

[21] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. "Evolutionary-scale prediction of atomic level protein structure with a language model". In: *bioRxiv* (2022), pp. 2022–07.

[22] John Ingraham, Max Baranov, Zak Costello, Vincent Frappier, Ahmed Ismail, Shan Tie, Wujie Wang, Vincent Xue, Fritz Obermeyer, Andrew Beam, et al. "Illuminating protein space with a programmable generative model". In: *bioRxiv* (2022), pp. 2022–12.

[23] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. "Protein 3D structure computed from evolutionary sequence variation". In: *PLOS One* 6.12 (2011), e28766.

[24] Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. "Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era". In: *Proceedings of the National Academy of Sciences* 110.39 (2013), pp. 15674–15679.

[25] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. "High-resolution de novo structure prediction from primary sequence". In: *bioRxiv* (2022), pp. 2022–07.

[26] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. "The Protein Data Bank". In: *Nucleic Acids Research* 28.1 (2000), pp. 235–242.

[27] Po-Ssu Huang, Scott E Boyken, and David Baker. "The coming of age of de novo protein design". In: *Nature* 537.7620 (2016), pp. 320–327.

[28] Ivan Anishchenko, Samuel J Pellock, Tamuka M Chidyausiku, Theresa A Ramelot, Sergey Ovchinnikov, Jingzhou Hao, Khushboo Bafna, Christoffer Norn, Alex Kang, Asim K Bera, et al. "De novo protein design by deep network hallucination". In: *Nature* 600.7889 (2021), pp. 547–552.

[29] Christoffer Norn, Basile IM Wicky, David Juergens, Sirui Liu, David Kim, Doug Tischer, Brian Koepnick, Ivan Anishchenko, Foldit Players, David Baker, et al. "Protein sequence design by conformational landscape optimization". In: *Proceedings of the National Academy of Sciences* 118.11 (2021), e2017228118.

[30] SP Mahajan, JA Ruffolo, R Frick, and JJ Gray. "Hallucinating structure-conditioned antibody libraries for target-specific binders." In: *Frontiers in Immunology* 13 (2022), pp. 999034–999034.

[31] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. "ProtGPT2 is a deep unsupervised language model for protein design". In: *Nature Communications* 13.1 (2022), p. 4348.

[32] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. "ProGen: Language modeling for protein generation". In: *arXiv preprint arXiv:2004.03497* (2020).

[33] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. "Rita: a study on scaling up generative protein sequence models". In: *arXiv preprint arXiv:2205.05789* (2022).

[34] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. "Large language models generate functional protein sequences across diverse families". In: *Nature Biotechnology* (2023), pp. 1–8.

[35] Namrata Anand and Tudor Achim. "Protein structure and sequence generation with equivariant denoising diffusion probabilistic models". In: *arXiv preprint arXiv:2205.15019* (2022).

[36] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. "Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models". In: *bioRxiv* (2022), pp. 2022–12.

[37] Yang Song and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems* 32 (2019).

[38] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. "Antigen-specific antibody design and optimization with diffusion-based generative models". In: *bioRxiv* (2022), pp. 2022–07.