



Modelling Sequence and Structure Towards Functional Protein Design

Citation

Paul, Steffanie Bradley. 2024. Modelling Sequence and Structure Towards Functional Protein Design. Doctoral dissertation, Harvard University Graduate School of Arts and Sciences.

Link

<https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37379828>

Terms of use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material (LAA), as set forth at

<https://harvardwiki.atlassian.net/wiki/external/NGY5NDE4ZjgzNTc5NDQzMGIzZWZhMGFIOWI2M2EwYTg>

Accessibility

<https://accessibility.huit.harvard.edu/digital-accessibility-policy>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#)

HARVARD
Kenneth C. Griffin



GRADUATE SCHOOL
OF ARTS AND SCIENCES

DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the
Division of Medical Sciences
Biomedical Informatics
have examined a dissertation entitled

Modelling Sequence and Structure Towards Functional Protein Design

presented by Steffanie B. Paul
candidate for the degree of Doctor of Philosophy and hereby
certify that it is worthy of acceptance.

Signature: Faisal Mahmood
Faisal Mahmood (Jul 26, 2024 03:05 GMT+9)

Typed Name: Dr. Faisal Mahmood

Signature: Maha Farhat
Maha Farhat (Jul 24, 2024 15:29 EDT)

Typed Name: Dr. Maha Farhat

Signature: Arjun K. Manrai
Arjun K. Manrai (Jul 24, 2024 12:03 EDT)

Typed Name: Dr. Arjun Manrai

Signature: Chang Liu
Chang Liu (Jul 24, 2024 09:11 PDT)

Typed Name: Dr. Chang Liu

Date: July 24, 2024

Modelling Sequence and Structure Towards Functional Protein Design

A DISSERTATION PRESENTED
BY
STEFFANIE B. PAUL
TO
THE DIVISION OF MEDICAL SCIENCES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
BIOMEDICAL INFORMATICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS
JULY 2024

©2024 – STEFFANIE B. PAUL
ALL RIGHTS RESERVED.

Modelling Sequence and Structure Towards Functional Protein Design

ABSTRACT

Millenia of evolutionary experiments have produced an extensive universe of natural macro-molecular machines - proteins - that perform the variety of complex functions needed to make up a cell. In the last few decades, advances in protein engineering technologies, including the adoption of machine learning methods, have enabled us to bend and reform nature's designs towards our human needs. The advent of generative machine learning models trained on evolutionary data has enabled us to leverage nature's experiments along with years of domain knowledge to significantly move the needle on what kinds of proteins and functions we can possibly design. While these models have massive promise, there is still much to understand about i) what these models are learning, ii) how well they are learning, and iii) which models are useful for which design task. This thesis provides tools and insights to the field to shed light on these questions and thus advance our ability to engineer proteins for the functions we want.

We begin with the need to identify what models perform better than others and what biological design tasks they may be useful for. Towards this, chapter 1 details our curation of the largest benchmarking dataset for generative protein models for fitness prediction, which we used to identify functional advantages for particular classes of generative models. This evaluation paradigm relies on functional measurements, which may not be available for any given protein an engineer is interested in. Thus, in chapter 2 we develop novel, statistically motivated kernel-based evaluation metrics that can be used to verify how accurately and reliably a conditional generative model has

learned the distribution of the protein of interest; this provides a practitioner with helpful information about how well their model might perform for their task *a priori*. For highly complex functions in highly local sequence space, we argue that focused experimental data are needed to get engineering gains. In chapter 3 we discuss how machine learning models can improve the efficiency of experimental pipelines and increase our design capabilities, with a case-study on machine learning-assisted antibody optimization.

Contents

TITLE PAGE	i
COPYRIGHT	ii
ABSTRACT	iii
DEDICATION	ix
ACKNOWLEDGEMENTS	x
o INTRODUCTION	i
o.1 The promise of machine learning for functional protein design	i
o.2 The nuances and challenges of antibody design	ii
1 COMBINING SEQUENCE AND STRUCTURE TOWARDS SUPERIOR FITNESS PREDICTION	16
1.1 ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design .	17
1.2 StructSeq: leveraging novel insights on functional advantages of protein generative model classes	33
2 KERNEL-BASED EVALUATION OF CONDITIONAL BIOLOGICAL SEQUENCE MODELS	41
2.1 Preamble	41
2.2 Abstract	42
2.3 Introduction	43
2.4 Problem Setting	45
2.5 Conditional Goodness-of-Fit with ACMMD	46
2.6 Assessing Reliability with ACMMD	51
2.7 Related Work	56
2.8 Experiments	57
2.9 Discussion	65

3	MACHINE LEARNING ENABLES EFFICIENT AND EFFECTIVE AFFINITY MATURATION	67
3.1	Preamble	67
3.2	Abstract	69
3.3	Introduction	69
3.4	Results	73
3.5	Discussion	87
4	CONCLUSION	90
APPENDIX A APPENDIX TO CHAPTER 1		94
A.1	Appendix to ProteinGym	94
A.2	Appendix to StuctSeq	118
APPENDIX B APPENDIX TO CHAPTER 2		120
B.1	Proof of Lemma 2.5.2	120
B.2	Asymptotic distribution of $\widehat{\text{ACMMD}}^2$	124
B.3	Proof of Lemma 2.5.3	125
B.4	Proof of Lemma 2.5.4	127
B.5	Type-I error control of the ACMMD test	127
B.6	Quantile estimation and Decision Rule	127
B.7	Wild-bootstrap and permutation-based approaches are equivalent in the ACMMD test	128
B.8	Level of the ACMMD test	129
B.9	Proofs related to ACMMD-Rel	131
B.10	Differences between the SKCE U-statistics and the ACMMD U-statistic	131
B.11	Proof of Proposition 2.6.2	132
B.12	Proofs regarding the impact of approximate kernels	133
B.13	Proof of Proposition 2.6.4	134
B.14	Proof of Proposition 2.6.3	135
B.15	Additional Details for ACMMD and ACMMD-Rel in the synthetic example	137
B.16	Derivations of ACMMD in the synthetic example	137
B.17	Closed-form ACMMD-Rel evaluation	146
B.18	Additional Experiments	148
B.19	Known Kernels for protein sequences and structures	149
APPENDIX C APPENDIX TO CHAPTER 3		153
C.1	Supplementary methods	153
C.2	Supplementary figures	159
C.3	Supplementary Tables	167

Listing of figures

1.1	ProteinGym benchmarks	21
1.2	Comparing baselines across performance metrics	31
1.3	Spearman differences between Inverse Folding models and top sequence models . .	36
2.1	ACMMD estimates and power rates on synthetic examples	59
2.2	ACMMD estimates and power rates for simulated protein examples	61
2.3	ACMMD-reliability estimates and power rates for simulated protein examples . .	62
2.4	Evolution of ACMMD and ACMMD-rel for varying ProteinMPNN temperature set- tings	64
2.5	ProteinMPNN has varying goodness of fit for different homologous superfamilies	66
3.1	Schematic of our approach	74
3.2	Machine Learning models predict second-round enrichment from first-round data	76
3.3	Enrichment prediction models identify optimizing single substitutions	79
3.4	Experimental validation of predicted optimizing single substitutions	86
A1	Hierarchical clustering of zero-shot models by NDCG performance	110
A2	Spearman comparisons for ProteinMPNN	119
B1	ACMMD estimates and power rates on further synthetic example settings	147
B2	Additional temperature settings for simulated protein examples using ProteinMPNN	148
B3	Goodness of fit and reliability estimates for ProteinMPNN on different homologous superfamilies	149
C1	Affinity maturation of nanobodies that bind b2AR, TMEM192, and RXFP1 . . .	159
C2	Count distributions of the sequencing rounds	160
C3	Enrichment distributions of the FACS rounds	161
C4	Comparison of top singles from a logistic regression model and CNN	162
C5	Heatmaps of scores for all single substitutions	163
C6	Analysis of polyreactivity counter-sort	164
C7	Affinity and Polyreactivity models have low correlation	165
C8	Affinity models suggest different top sequences from the experiment	166

DEDICATED TO LITTLE STEFFAN
WHO ALWAYS DREAMED OF BEING CALLED DR.

Acknowledgments

Firstly, I thank my advisor, Debbie Marks. Thank you for the experience of being in your lab. It has moulded me into the scientist I am today, and for that I will always be grateful.

I thank all the amazing mentors in the lab who have spent their time to teach and guide me. Alan Amin for pushing me to chase statistical rigor in all my work. Pascal Notin for his thoughtful conversations on the world of machine learning. June Shin for introducing me to the world of antibodies. I thank the fantastic PIs and faculty who I have had the pleasure to work with and learn from. Andrew Kruse, Aashish Manglik, Wengong Jin and Chris Sander. I thank my PQE, DAC and defense committee for being willing to advise me through graduate school. Marinka Zitnik, Soumya Raychaudhuri, Kun Hsing-Yu, Andrew Beam, Nick Polizzi, Faisal Mahmood, Raj Manrai, Maha Farhat, Chang Liu.

I thank my undergraduate mentors, Peter Koo and Sean Eddy, for starting me on this path and continuing to offer me encouragement every step of the way. I will ever be in your debt.

I thank all the wonderful collaborators with whom I've had the privilege of producing novel science. Pascal Notin, Aaron Kollasch, Lood van Niekark, Daniel Ritter, Pierre Glaser, Eddie Harvey. A special shout out to Alissa Hummer who has been a great friend and co-conspirator in our machinations about antibodies.

I thank my dear friends in the Marks Lab. You all made it a pleasure and a joy to come to lab everyday and I will miss you all so much. Courtney Shearer, Rose Orenbuch, Fiona Qu, Sam Berry, Sarah Gurev, Ruben Weitzman, Sam Huang, Abigail Jackson, Dinko Franceschi, Tessa Green, Ada Shaw, Noor Youssef, Jake Reardon, Johnny Frazer, Mafalda Dias.

A special thank you to my (work)husband, Han Spinner. If nothing else had come of this PhD, meeting a friend and partner-in-crime like you would have still made it all worth it. I love you!

I thank all the amazing friends I've met in grad school. A special shout-out to Shandu Mulaudzi. If not for whom I might have actually gone to Yale.

Most of all I thank my friends. Helen Murphy, Jules Benthall, Lucien Kleber, Han Spinner, Alex Papale, Fiona Riley (et al). The greatest things I have learned in the past four years, I have learned from you all. How to love and how to be loved. Thank you for holding me through this journey.

And finally, I thank...Steffanie. In a world where trans people are being legislated out of existence, she carved out a safe space for herself and for everyone whom she loves. In a world where brown women are left out of the conversations at the highest levels of science and academia, she planted her flag right at the top and demanded that her voice be heard.

I never thought it would all come to this.

And all I will ever be
is grateful.



Introduction

0.1 THE PROMISE OF MACHINE LEARNING FOR FUNCTIONAL PROTEIN DESIGN

Millenia of evolutionary experiments have produced an extensive universe of natural molecular machines - proteins - that perform the variety of complex functions needed to make up a cell. In the last few decades, advances in protein engineering technologies have enabled us to bend and reform nature's designs towards our human needs. This includes biologic therapeutics that have ad-

vanced our abilities to treat previously untreatable diseases. Now, with the advent of probabilistic machine learning we can leverage evolutionary data to learn from nature's evolutionary experiments to improve our ability to engineer nature towards our will. While these methods hold great promise, there is still much to learn about how best to use them for specific protein engineering tasks. In this thesis, I elucidate different evaluation schemes to help engineers understand: i) what their models are learning, ii) how well they are learning, and iii) which models to use and when. I also develop a case-study in machine learning assisted antibody engineering to demonstrate how targetted deployment of machine learning models can aid in highly specific engineering tasks.

Protein engineering encompasses any task in which we are modulating or designing a protein to perform a desired function. Our motivations for protein engineering arise from a variety of applications, including biologic therapeutics, industrial reagents, and molecular biology tools. For any application, we are often met with a biochemical function (eg. inhibiting a pathogenic protein target, catalyzing a reaction in a manufacturing pipeline) for which we want to design a protein to carry out that function or optimize a protein that already performs that function. Engineering tasks range in the complexity of the desired function and novelty of the protein being developed; highly bespoke functions are more challenging to engineer than general protein properties, and designing a *de novo* protein can be more challenging than optimizing a starting point. We sometimes consider the abstract collection of possible sequences as a high-dimensional "sequence space". This is a useful idea once we start leveraging probabilistic models for engineering protein sequences, as these models learn a probability distribution over sequence space. When considering the diversity goal for an engineering project it is often helpful to consider what part of sequence space we are trying to traverse. When optimizing a starting point protein, we are looking to explore a slice of sequence space local to the original sequence (i.e only changing a relatively small fraction of the residues in the sequence). We can think of *de novo* design as seeking to design proteins in a patch of sequence space that has not been explored before (i.e design sequences that have low sequence identity to anything previously

observed).

For optimization problems, high throughput experimental techniques like deep mutational scanning (DMS)²⁷³ enable assaying a subset of sequence space around a starting candidate (i.e sets of sequences a few mutations away from the starter) for variants that improve the function being measured. While powerful, throughput often limits us to only assaying all few-away mutants (i.e singles or maybe doubles). Directed evolution experiments allow us to assay a more diverse span of sequence space, increasing the likelihood of a more novel, optimized hit¹⁷. However, these experiments often need finetuning of mutation rates and selection constraints to avoid bottlenecking of the variant population.

The goal of *de novo* protein design is to construct novel proteins from scratch, advancing beyond natural evolutionary pathways. In the span of a couple of decades the field moved from proof-of-concept showing that it was possible to decorate a simple alpha helix scaffold with an artificial amino acid sequence that folded and was stable²²⁰, to the *de novo* design of arbitrarily shaped protein scaffolds with favorable protein properties using deep learning models^{117,271}. *De novo* binders have also been designed, illustrating an example of a useful function (engaging in a protein-protein interaction) that is enacted by an artificial sequence^{33,201,20}. However, many approaches are still limited to success on simple helix-based scaffolds. Even within the functional scope of binding proteins, this limits the shapes of binding motifs that can be designed thus limiting the range of targets that can be bound. Having demonstrated that we are capable of designing proteins per say, the field is now moving towards designing proteins to fulfill highly specific functions (eg. catalysis of novel reactions or binding to completely novel epitopes).

Generative machine learning models are poised to be crucial in this next stage of protein engineering¹⁹². As opposed to predictive models that are trained to predict a label given an input, generative models are machine learning models that can be sampled from; this makes them highly useful for design. A major problem for computational sequence design is that the possible search space of

sequences is so large (even a 100 residue long protein has a combinatorial space of 10^{130} possible sequences). Classical design methods involved sampling candidate designs and scoring them with a function that approximates the design success^{151,20}. Even assuming we have access to an accurate way to identify good designs from bad designs (which is an infeasible assumption for most tasks) sampling appropriate candidates to score is challenging, and it is unlikely that the best candidate would be sampled randomly. Protein generative models address this problem by learning a probability distribution over proteins, giving it a notion of what protein choices are good, and more importantly, allowing us to sample directly within the constraint of "likely" proteins.

Borrowing from the genetics jargon, we call this learned likelihood from the model its estimation of the protein's "fitness",⁵³. In natural selection, fitness is any property of a protein (or more specifically, a gene) that promotes its survival across generations. Fitness can reflect a variety of biophysical properties including how well the protein expresses, how thermostable it is, and how well it performs its function. The assumption behind generative modelling is that if one trains a model on a corpus of proteins that are "fit" (i.e possess a certain function or property that has caused them to be carried through evolution), the distribution that that model learns should reflect how likely a given protein is to possess ideal properties.

Indeed, we have observed that the predicted "fitness" from a generative model correlates with a number of different protein properties (we will discuss how we perform this correlation later in this chapter)^{110,207,193}. This correlation means that generative models function as useful "zero-shot" predictors of such properties. Zero-shot prediction refers to a model that can predict a certain property without being trained explicitly on labels relating to that property (i.e supervised prediction). Many groups have leveraged this paradigm to attempt zero-shot engineering of a property of interest using pre-trained generative models^{246,102}.

Depending on the choice of training data and architecture, the model learns different notions of what makes a protein "likely". We can think of our choices of training data and architecture as

instilling "inductive biases" into our model. These biases could or could not be beneficial depending on our task. Thus, it would behoove us to leverage the appropriate inductive bias - amounting to selecting a particular training set or model architecture - for our task of choice.

Models of different protein representations (i.e. sequence or structure) allow us to model and sample proteins at either the sequence or structural level. Initial sequence models relied on multiple sequence alignments (MSAs) to produce uniform length inputs wherein conserved residues are aligned within the same position. Potts models fit to these alignments have been shown to learn interactions between residues that can be used to recapitulate structural contacts¹⁰⁹. Variational autoencoders (VAEs) have also been incredibly powerful for sequence modelling as they can model epistasis above second-order interactions^{207,83}. With the advent of the transformer architecture²⁶⁷, sequence models could be trained without aligning sequences. Large language models (LLMs) based on the transformer architecture have been trained on the entire corpus of known protein sequences contained in Uniref²⁴⁹. These models intuitively learn general rules about protein properties and have been incredibly useful for generating latent representations imbued with evolutionary information across the tree of life^{208,161,190,8}. These models learn an idea of what mutations would be deleterious to a protein, and they have been used to reduce the search space of possible design variants to test¹⁰² and also recommend likely designs with low sequence identity to the original sequence⁸².

Recently, many generative models of protein structure have also been developed; many of which are based on advances in graph neural networks^{134,124}. Diffusion based models^{234,233} have seen the greatest success in this domain, with multiple groups demonstrating the ability to sample a range of possible structural scaffolds^{271,117,259}. Most of these methods only sample the backbone of the structure which contains the essential shape of the protein. Inverse folding models were developed to solve the problem of sampling sequences that would likely fold into an input backbone structure^{112,116,54}. Groups have also found success in optimizing proteins for thermostability and functional activity using inverse folding models^{246,20}.

These modelling strategies can be organized by the different marginal and conditional distributions on protein sequence and structure that they are learning. Let X and Z be random variables for the protein sequence and structure respectively. LLMs trained on the protein universe learn the marginal distribution of protein sequences, $P(X)$. MSA based models learn a distribution conditioned on the homologous family the protein comes from, $P(X|X' \in H(X))$, where $H(X)$ reflects the family a protein belongs to. Generative structure models learn $P(Z)$, while inverse folding models learn the conditional distribution of sequences given a structure, $P(X|Z)$. Indeed, recently the problem of protein folding has also been framed as a generative problem, where models are trained to approximate $P(Z|X)$ ^{125,2}. By framing all these modelling tasks probabilistically, we bake in the ability to get scores for any input sequence or structure using the likelihoods from these models. These scores are what make these generative models useful for zero-shot prediction and design.

The great potential that zero-shot generative modelling presents is that it democratizes the capacity to use these modelling approaches. Instead of having to generate a large corpus of training data, small engineering groups can leverage publically-available pre-trained methods to further their engineering goals. However, these models are not effective for all designs tasks. So how are we to know that a given model may be useful for a task of interest?

Zero-shot predictive ability for generative models ranges depending on the function and protein of interest^{207,193}. This lack of performance could arise from limitations in the training data or the model architecture, or both. For more complex functions that may only be shared amongst a small set of protein families, the general assumption that fitness with respect to a large evolutionary dataset correlates well with our function breaks down. In particular, for novel functions (eg. enzymatic activity on non-natural substrates), if a model has only seen examples of natural-functioning proteins, it may not assign any high likelihood to proteins with novel functions as these were unseen during training. For such cases, rich experimental data that queries the specific function we are interested in may be required. Semi-supervised modelling approaches have also been developed to try and

leverage the best of both worlds¹⁹⁴; combine the rich information from evolutionary datasets with the targetted signal from functional experiments.

To understand what design task a generative model may be effective for, we need relevant evaluation paradigms. The best possible evaluation we can do is to design proteins with our method of choice and to test them experimentally. However, experimental validation is highly resource intensive. Furthermore, often the functions we are most interested in are the most challenging to assay in high-throughput. As more and more methods are published, it quickly becomes intractable to experimentally characterize model performance even for a single function focussing on a single protein family. As such, *in silico* benchmarks are essential for building models with improvements in design performance.

One way to establish performance is to measure the correlation between a model's likelihood (which we will sometimes refer to as the model score) and a functional measurement from an experiment. This tells us how well the model's learned likelihood distribution reflects the distribution of a particular function. If this correlation is high, this suggests that samples from the model (i.e. that have high likelihood) would also possess the function captured by the experimental measurement. Notably, this sort of analysis only tells you how well the model correlates with function for the sequences that were tested and similar such sequences. If the test set contains a fairly narrow slice of sequence space (eg. all singles and doubles of a particular starting protein), we only have confidence that our model has learned the functional distribution within this slice. This is still useful information if we are trying to design similar proteins to those which we have functional data on. Furthermore, if we can demonstrate predictive performance on a wide diversity of functional datasets, encompassing a wide array of protein families and assayed functions, this raises our confidence that our model's distribution matches the functional distribution in *enough* parts of sequence space to be useful for design.

The first chapter of this thesis aims to provide the most holistic evaluation of generative models

via correlation with functional measurements from DMS experiments - which we will refer to as "fitness prediction". We curated ProteinGym, the largest database of DMS studies - over 250 assays - to date and benchmarked over 70 popular zero-shot protein generative models. Performance on these datasets gives us a way to rank these methods for prospective design performance. While even 250 assays does not nearly encompass the entire span of functions and protein families that protein engineers may be interested in, we argue that the benchmark is sufficiently diverse to give us useful information for making decisions for which models to use for engineering tasks. The range of different functions assayed allowed us to identify specific functional advantages that particular model classes have. In particular, we found that sequence models conditioned on structure (inverse folding models) have significant advantages in predicting thermostability. We leverage these advantages to develop an ensemble model of different classes that results in state of the art performance on the ProteinGym benchmark. This work provides insights for engineers on what models they should pick for use in designing their specific function, and is a major step towards understanding how machine learning models are useful for protein design.

Not every protein family that an engineer may be interested in is represented in ProteinGym. Without performance on functional data related to that protein, how can a practitioner gauge how well a model might work? Some insight can be gained by looking for performance on related proteins. For example, if one is trying to engineer a solute-carrier (SLC) protein, performance on another class of transporters like ABC transporters might suggest that the model may also perform well on SLCs. However, for functionally unique families this becomes infeasible. While high throughput data on the protein of interest might not be available, it is likely that sequencing or some structural characterisation of the protein may be available in the large databases that generative models are trained on^{249,23}. Following from our assumption that the distribution of the designed function we are interested in correlates with the fitness distribution for the protein of interest, we would at least want our generative model to have learned the distribution around our protein of

interest well.

Test perplexity, a metric borrowed from the natural language processing world, is commonly used to assess model fit. Test perplexity (as opposed to entropy-based perplexity) is calculated as the exponential of the negative log likelihood. We often look to see if the model's perplexity on a training set is on par with a test set to identify if a model can identify test data as within the same likelihood distribution as its training data. This can be useful for comparing models to each other (with respect to the same dataset). However, when we want to compare a model's fit on different protein families (i.e different subspaces of the training sequence space), perplexity can be misleading. Lower perplexity on one family does not necessarily mean that the model has fit that distribution better than another family with higher perplexity; this could merely point to more training data being available for certain families.

In the second chapter, we leverage new theory on conditional goodness of fit tests to develop statistically motivated evaluation metrics for popular generative model classes that directly measure how well a model has learned a protein distribution using test data. The metrics give us absolute information in that the optimal fit of the model is bounded by the distribution of the data itself, thus we know how far away from ideal our model's fit is. As opposed to perplexity which only gives us performance information relative to other model baselines. These metrics allow us to interrogate a model's performance on specific families. We also develop metrics to measure model reliability - a property formally called calibration - which tells us how well the uncertainty in the model's prediction matches the uncertainty in the actual data. Together, these tools provide ways to ascertain whether a model has learned a particular protein distribution well (which would raise our confidence in its utility for a designing that protein) and whether we can trust the uncertainty in the likelihoods from the model (which allows us to make decisions on whether we think a model prediction should be taken into the wet-lab).

We are far from the point where a single model can be used for all design tasks (if that is even a

relevant goal to work towards). While generative models can likely generalize important protein properties like thermostability and expressibility, more complex functions that are specific to certain families, and novel functions unexplored in nature are more challenging design tasks that often require specific experimental data to guide engineering decisions. Machine learning models can greatly improve engineering pipelines that require experimental data. Often, this involves some amount of supervised or semi-supervised learning. In such settings, model interpretability is especially important to ensure that the model is not learning artifacts from the experiment. Otherwise these artifacts can persist downstream if the model's predictions are used to make engineering decisions.

In the third chapter, we present a case study in leveraging machine learning to improve the efficiency and hit-rate of an antibody affinity maturation pipeline. Affinity maturation involves the optimization of a specific antibody's binding affinity to its epitope. This is an example of a complex function that may not be shared by a large number of proteins in evolutionary databases, and thus requires a different modelling strategy. Low throughput binding assays are required to validate any candidates suggested from an optimization pipeline (experimental or computational); as these are recourse expensive, it is especially important to have pipelines with a high hit rate to avoid wasting recourses validating false positives. In this work, we show how simple but interpretable logistic regression models actually outperform complex neural network architectures in this specific design task. We use models trained on minimal experimental data to suggest optimizing mutations for multiple antibodies, reducing the recourse intensity from the optimization and validation experiments.

All together, this thesis outlines novel tools, insights and engineering practices to further the adoption of generative models in protein engineering and to increase our understanding of what these models are actually learning and what particular design tasks they may be useful for.

0.2 THE NUANCES AND CHALLENGES OF ANTIBODY DESIGN

Antibodies are the fastest growing subtype of therapeutics approved by the FDA¹⁵⁹. In 2021, the global antibody market was valued at 217.3 billion USD and in 2022, 30% of FDA approved drugs were monoclonal antibodies (mAbs)¹⁶⁶. Antibodies are proteins used by the immune system to recognize foreign targets with high specificity and affinity. This high specificity makes them excellent therapeutics as they have low off-target binding reducing adverse effects. Their high affinity and tissue directability allow them to target previously undruggable targets, resulting in breakthrough treatments for many cancers and autoimmune diseases^{199,256}. Due to their therapeutic potential, there is a heightened demand for improved technologies to engineer and optimize antibodies. Furthermore, many antibody drugs are prohibitively expensive. Thus, methods that can improve the recourse efficiency of engineering campaigns are needed as they will reduce the cost of producing the therapeutic, resulting in cheaper drugs for the patient.

The antigen-binding fragment (F_{ab}) of an antibody consists of 4 polypeptide chain: the heavy and light constant chains (C_H and C_L) and the variable chains (V_H and V_L)³. Each variable chain consists of a conserved scaffold of beta sheets - called the framework - and 3 flexible loops referred to as the complementarity-determining regions (CDRs). Of these the CDR-H3 is the most diverse as it spans the intersection of the variable (V), diversity (D), and joining (J) gene segments that recombine to produce a unique V(D)J sequence. The diversity in CDR sequences and the flexibility of the loop structures allow antibodies to accommodate a huge number of epitope shapes, making them extremely useful platforms for engaging any therapeutic target. The CDRs comprise much of the binding interface of an antibody^{144,143}. The diversity of the CDR-H3 means these interfaces can be highly specific to the specific epitope. Camelids (and some sharks) produce single-domain antibodies that lack a light chain. These have been engineered into single chain binders (by removing the constant domains) called nanobodies⁹⁹. These smaller binders allow binding to an extended range

of epitopes.

Starting with a novel target antigen, engineering a novel antibody binder takes place over two stages: the discovery campaign and the affinity maturation campaign²⁹⁴. Discovery involves raising a novel antibody that binds specifically to the target antigen. However, most discovery platforms produce antibodies with sub-par binding affinity for therapeutic and molecular reagent purposes. These starting candidates are then optimized for binding affinity and specificity through affinity maturation.

Traditionally, antibodies were discovered by immunizing an animal model with the target antigen, and leveraging the animal's innate immune system to produce a polyclonal antibody cocktail. The antibodies are extracted from the animal sera and individual binders are isolated into mAbs using the hybridoma technique. Early mouse-raised therapeutics suffered from high immunogenicity due to the patient immune system reacting against the foreign protein material¹⁵⁹. Humanization techniques reduce this immunogenicity by incorporating the binding site from the mouse antibody onto a human germline framework scaffold (by merely swapping in the CDR sequences) to form a chimera²⁰⁶. Humanized mice lines have also been developed wherein the innate mouse immune gene cassettes (encoding the frameworks and V(D)J genes) are exchanged with human genes¹⁴⁸. Consequently, any antibody raised in the model system will resemble a human antibody. Animal models have proven highly effective systems for raising antibodies; most FDA approved mAbs were discovered using animal immunization techniques^{159,166}.

However, these techniques are recourse intensive and afford no control over the types of antibodies produced. To address these limitations, *in vitro* experimental techniques were developed. Large, diverse antibody libraries are presented using yeast- or phage-display technology and screened against a target of interest via magnetic or fluorescent activated cell sorting (MACS or FACS respectively)^{26,147,43}. Multiple rounds of selection are performed to arrive at a single modest affinity binder. Larger and more intelligently-designed libraries increase the success rate of finding a specific

binder to a target. However, the larger the library the more selection steps need to be performed; which also increases the possibility of artifacts expanding in the population (such as polyspecific binders or highly expressing-non binders¹⁰⁰). These techniques have proven very successful and have been onboarded as standard platforms by many biologics pharmaceutical companies.

In vitro technologies still suffer from the inability to control what epitope on the target antigen an antibody will bind to. Recently, much progress has been made towards computational discovery methods that would allow engineers to specify exactly where on an antigen structure an antibody should bind. Some methods take a fragment-screening approach, wherein smaller peptide fragments mined from the protein data bank (PDB) are stitched together to form a novel paratope which is then grafted onto an antibody framework^{6,153}. There has also been much excitement for deep learning based *de novo* design approaches. Here generative neural network models are trained on binding complex structures (antigen-antibody complexes or general protein-protein interactions) and then antibody paratopes are sampled that would likely bind to a specified conformational epitope^{21,123,69}. Recently, Bennett et al.²¹ developed a generative diffusion model finetuned on nanobody binding complexes which was able to generate target-specific small libraries (100 – 10k sequences) using only a structure of the target and the desired VHH framework, from which on-target binders were identified post some experimental screening. Notably, these binders had comparable affinities to nanobodies identified by experimental discovery campaigns. However, many of these designs were to epitopes for which there was pre-existing binder material in the PDB. Thus, further validation is needed to demonstrate the true *de novo* capabilities of the method.

In addition to being able to specify an epitope, computational discovery methods offer the ability to combine the discovery model with computational models for other desirable developability properties (including solubility, immunogenicity¹⁶³, polyreactivity¹⁰⁰) to design binders that would more likely succeed in a lead optimization pipeline for a therapeutic. As most drug candidates fail for not meeting these developability requirements, starting a drug campaign from scratch with these

in mind would likely increase the success rate of therapeutics projects, thus reducing the number of failed attempts needed to produce a new drug, thereby reducing the overall cost of the therapy for the patient.

Following discovery of a candidate binder, the antibody needs to be optimized through affinity maturation. Initially, this was done using rational design principles; if a crystal structure of the binding complex was available, engineers could suggest mutations that were likely to stabilize the interaction. However, as structures are challenging to obtain themselves, high throughput affinity maturation techniques were developed to query multiple mutations simultaneously for binding optimization. This often proceeds by generating a mutational library originating from the starting candidate (by combinatorially mutating residues in the CDR or by using error-prone PCR) and subsequently screening the library to arrive at the highest affinity binder using sorting assays similar to those used in *in vitro* discovery campaigns. Further explication of these methods and a description of the limitations and challenges to address can be found in the introduction to chapter 3.

Antibody engineering presents unique challenges not present in general protein engineering. For one, the paratope of the antibody is highly unstructured before binding. This means that making small changes to the paratope can result in large conformational changes, as opposed to general proteins where the presence of secondary structure constrains the paratope shape somewhat. This flexibility and the diverse CDR sequences that produce it also make it difficult to learn from other similar protein sequences and structures due to low conservation⁵. One way this manifests is in the difficulty of producing alignments of antibody CDRs, which have been crucial for training generative models for protein design. Moreover, this also makes it challenging to leverage the massive expanse of protein data available, requiring specific antibody datasets that thus far are much smaller than general protein databases (contrast the 8200 antibody-antigen binding complexes in the structural antibody database⁶⁶ with the >200k structures deposited in the PDB²³).

Overall, antibody design is a powerful tool with major therapeutic and industrial implications for

which we have developed a wide array of engineering techniques. There are many promising avenues for further development of engineering methods, both experimentally, computationally, and at the intersection of the two. Despite the particular challenges that antibodies present, the potential of this technology promises to meet the demand for more and more antibodies.

1

Combining Sequence and Structure towards Superior Fitness Prediction

1.1 PROTEINGYM: LARGE-SCALE BENCHMARKS FOR PROTEIN FITNESS PREDICTION AND DESIGN

1.1.1 PREFACE TO PROTEINGYM

This chapter presents work from two papers: the first section from a paper published at the Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks¹⁹¹, and the second from a workshop paper presented at the 2023 NeurIPS workshops, Generative AI for Biology and Machine Learning for Structural Biology. My contribution to the first section was in benchmarking the inverse folding methods presented in the paper; contributing to the comparative analysis of the models using different evaluation metrics; and in writing the relevant sections for these results. For the second section, I was first author and I performed all the analyses presented in the paper and also led the paper writing. Multiple other authors contributed to the curation of datasets and benchmarking of other models in ProteinGym (Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekark, Hansen Spinner, Nathan Rollins, Ada Shaw, Ruben Weitzman, Jonathon Frazer, Mafalda Dias, Dinko Franceschi, Rose Orenbuch, Yarin Gal, Debora S. Marks).

The original ProteinGym paper included zero-shot benchmarking on datasets of both substi-

tution and indel variants; supervised benchmarking on these datasets; and an analysis of clinical variant prediction. Only the results on zero-shot prediction of substitution variants was kept in this thesis as this is where my contributions were focussed. Regarding any reference to indels, supervised benchmarking and clinical variant benchmarking please see the full publication ¹⁹¹.

1.1.2 ABSTRACT

Predicting the effects of mutations in proteins is critical to many applications, from understanding genetic disease to designing novel proteins that can address our most pressing challenges in climate, agriculture and healthcare. Despite a surge in machine learning-based protein models to tackle these questions, an assessment of their respective benefits is challenging due to the use of distinct, often contrived, experimental datasets, and the variable performance of models across different protein families. Addressing these challenges requires scale. To that end we introduce ProteinGym, a large-scale and holistic set of benchmarks specifically designed for protein fitness prediction and design. It encompasses both a broad collection of over 250 standardized deep mutational scanning assays, spanning millions of mutated sequences, as well as curated clinical datasets providing high-quality expert annotations about mutation effects. We devise a robust evaluation framework that combines metrics for both fitness prediction and design, factors in known limitations of the underlying experimental methods, and covers both zero-shot and supervised settings. We report the performance of a diverse set of over 70 high-performing models from various subfields (eg., alignment-based, inverse folding) into a unified benchmark suite. We open source the corresponding codebase, datasets, MSAs, structures, model predictions and develop a user-friendly website that facilitates data access and analysis.

1.1.3 INTRODUCTION

Proteins carry out a wide range of functions in nature, facilitating chemical reactions, transporting molecules, signaling between cells, and providing structural support to cells and organisms. This astonishing functional diversity is uniquely encoded in their amino acid sequence. For instance, the number of possible arrangements for a 64-residue peptide chain (20^{64}) is already larger than the estimated number of atoms in the universe. Despite substantial progress in sequencing over the past two decades, we have observed a relatively small, biased portion of that massive sequence space. Consequently, the ability to manipulate and optimize known sequences and structures represents tremendous opportunities to address pressing issues in climate, agriculture and healthcare.

The design of novel, functionally optimized proteins presents several challenges. It begins with learning a mapping between protein sequences or structures and their resulting properties. This mapping is often conceptualized as a “fitness landscape”, a multivariate function that characterizes the relationship between genetic variants and their adaptive fitness. The more accurately and comprehensively we can define these landscapes, the better our chances of predicting the effects of mutations and designing proteins with desirable and diverse properties. Machine learning, by modeling complex, high-dimensional relationships, has emerged as a powerful tool for learning these fitness landscapes. In recent years, a plethora of machine learning methods have been proposed for protein modeling, each promising to offer new insights into protein function and design. However, assessing the effectiveness of these methods has proven challenging. A key issue is their evaluation on distinct and relatively sparse benchmark datasets, while relative model performance fluctuates importantly across experimental assays, as was shown in several prior analyses^{207,146,174}. This situation underscores the importance of scale in the benchmarks used. Larger, more diverse datasets would offer a more robust and comprehensive evaluation of model performance.

To address these limitations, we introduce ProteinGym, a large-scale set of benchmarks specifi-

cally tailored to protein design and fitness prediction. It comprises a broad collection of over 250 standardized Deep Mutational Scanning (DMS) assays which include over 2.7 million mutated sequences across more than 200 protein families, spanning different functions, taxa and depth of homologous sequences. It also encompasses clinical benchmarks providing high-quality annotations from domain experts about the effects of ~ 65 k substitution and indel mutations in human genes (§ 1.1.5).

We have designed ProteinGym to be an effective, holistic, robust, and user-friendly tool. It provides a structured evaluation framework that factors in known limitations of the underlying experimental methods and includes metrics that are tailored to protein design and mutation effect prediction (§ 1.1.6). We report the performance in a unified benchmark of over 70 diverse high-performing models that come from various subfields of computational biology (eg., mutation effects prediction, sequence-based models for de novo design, inverse folding), thereby supporting novel comparisons across. Unlike prior benchmarks, ProteinGym integrates both the zero-shot and supervised settings, leading to new insights (§ 1.1.7). All models are codified with a common interface in the same open-source codebase, promoting consistency and ease of use. Lastly, a dedicated website offers an interactive platform to facilitate comparisons across datasets and performance settings.

1.1.4 RELATED WORK AND BACKGROUND

MULTI-TASK PROTEIN BENCHMARKS In recent years, several benchmarks have been introduced to provide initial means to assess protein model performance across a multitude of tasks of interests, e.g., predicting contacts, structure, thermostability, and fitness. These benchmarks are generally geared towards assessing the quality of learned protein representations, and the extent to which these representations can be broadly leveraged for various tasks. However, for fitness prediction, they all rely on a very limited set of proteins (e.g., 1-3 assays). In comparison, the ProteinGym

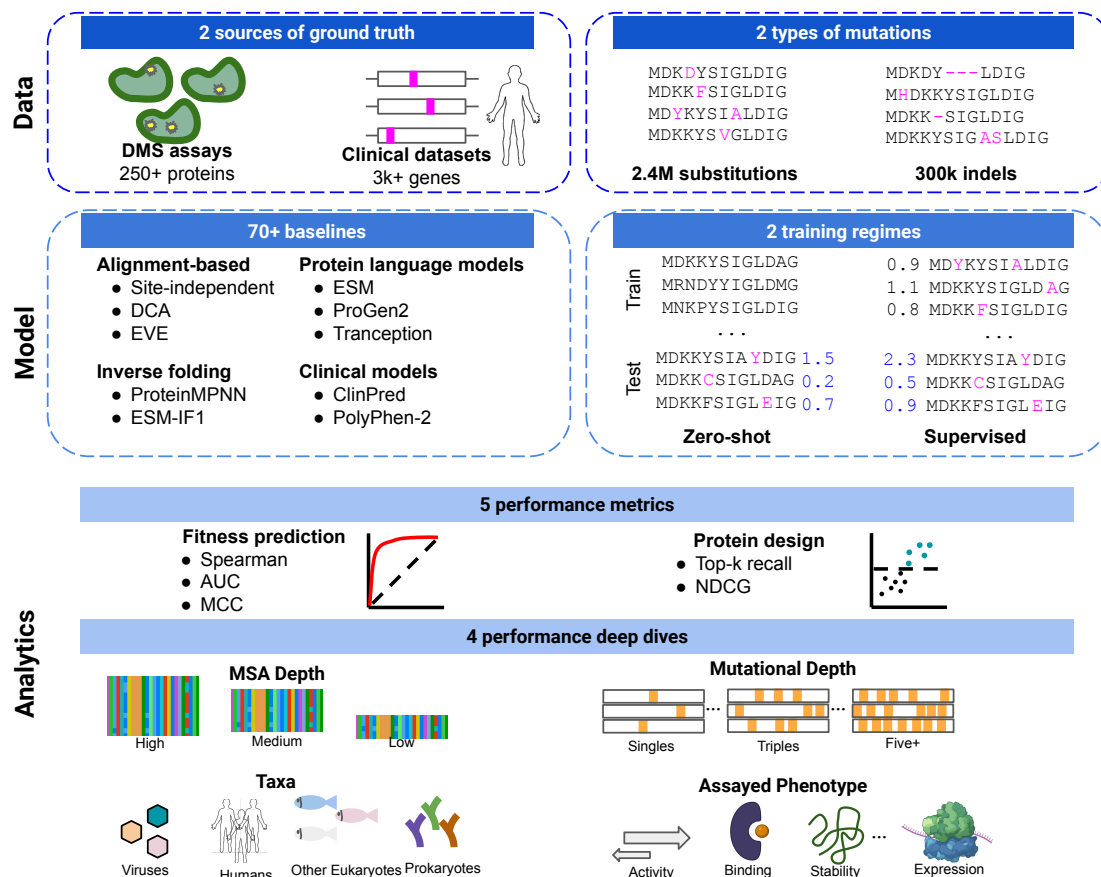


Figure 1.1: ProteinGym benchmarks ProteinGym is comprised of three layers. The data layer encompasses two complementary ground truth labels from DMS assays and clinical annotations from experts. For both, we analyze two types of mutations: substitutions and indels. The model layer is comprised of a diverse set of baselines, tailored to both zero-shot and supervised training regimes. Lastly, the analytics layer includes several performance metrics geared towards fitness prediction or protein design evaluation. Different segmentation variables (e.g., MSA depth, assayed phenotype, taxa) facilitate the comparisons of models across diverse settings

benchmarks focus on a single task – fitness prediction – and encompass *two orders of magnitude more* point mutations assessed and vast diversity of protein families included.

TAPE (Tasks Assessing Protein Embeddings)²⁰⁴ covers five protein prediction tasks, each designed to test a different aspect of protein function and structure prediction (secondary structure, contact, remote homology, fluorescence and stability), and focuses on assessments in the semi-supervised regime via carefully curated train-validation-test splits. ProteinGLUE³⁴ also focuses on assessing the usefulness of learned protein representations on supervised downstream tasks. It is comprised of five different tasks, none directly related to protein fitness: secondary structure, solvent accessibility, protein-protein interactions, epitope region and hydrophobic patch prediction. PEER²⁸⁷ also focuses on multi-task benchmarking, grouped in five categories: protein property, localization, structure, protein-protein interactions and protein-ligand interactions. It contains a richer set of evaluations compared with the prior two benchmarks, and also investigates the multi-task learning setting, but is not designed for thorough fitness prediction benchmarking (3 fitness related assays). The handful of fitness DMS assays from these various benchmarks are all subsumed in ProteinGym.

SINGLE TASK, NON-FITNESS DATASETS & BENCHMARKS Efforts to create fair, large-scale, and comprehensive benchmarks have been a significant focus of computational biologists for certain tasks. Among these, the biennial Critical Assessment of protein Structure Prediction (CASP)¹⁴² is the most renowned. CASP concentrates on protein structure prediction and has set the gold standard in this domain. In parallel to CASP, the Critical Assessment of Functional Annotation (CAFA)²⁹⁷ challenge provides a platform for evaluating protein function classification. The SKEMPI¹⁸² database is specifically designed to aid the evaluation of computational methods predicting the effect of mutations on protein-protein binding affinity. Several datasets have been curated for specific properties of interest across a diverse set of proteins, for instance thermostability^{260,244,40} or solubil-

ity¹⁰⁸.

PROTEIN FITNESS BENCHMARKS Closest to our work are the collections of DMS assays that were curated in Hopf et al.¹¹⁰ (28 substitution assays), and then further expanded upon in Rieselman et al.²⁰⁷ (42 substitution assays) and Shin et al.²²⁷ (4 indel assays). We include all assays related to fitness prediction from these prior works in ProteinGym. FLIP⁵¹ focused on comparing fitness predictors in the semi-supervised setting, developing a robust evaluation framework and curating cross-validation schemes for three assays. MaveDB²¹⁵ is a repository rather than a benchmark, but it compiles a large collection of datasets from multiple variant effect mapping experiments that can be used for benchmarking purposes. An initial prototype of the ProteinGym benchmarks (referred to as ‘ProteinGym vo.1’) was introduced in Notin et al.¹⁹⁰. We have since then significantly expanded the benchmarks in terms of number and diversity of underlying datasets, baselines, evaluation framework and model training regimes (Table A1). This not only enables performance evaluation at an unprecedented scale, but also builds connections between different subfields that are often perceived as separate, as we discuss in the following paragraph.

RELATIONSHIP BETWEEN PROTEIN FITNESS, MUTATION EFFECT PREDICTION AND DESIGN

The protein fitness landscape refers to the mapping between genotype (e.g., the amino acid sequence) and phenotype (e.g., protein function). While it is a fairly broad concept, it should always be thought about in practice within a particular context (e.g., stability at a given temperature in a specific organism). Models that learn the protein fitness landscape have been shown to be effective at predicting the effects of mutations^{83,119,29,193}. But the ability to tell apart the sequences that are functional or not is also critical to protein engineering efforts^{210,289,285,8}. Although typically introduced in the context of de novo protein design¹¹³, inverse folding methods^{116,124,54,85} can also be used for mutation effects prediction (Appendix A.1.4). There is thus a very tight connection

between protein fitness, mutation effect prediction and protein engineering, and the same models can be used for either task depending on context. We seek to illustrate this connection through this work, comparing baselines introduced in different fields (e.g., protein representation learning, inverse folding models, co-evolution models) on the same benchmarks, and including different metrics that are geared more to mutation effect prediction (e.g., Spearman) or design tasks (e.g., NDCG).

1.1.5 PROTEINGYM BENCHMARKS

ProteinGym is a collection of benchmarks (Fig. 1.1) that cover different types of mutation (ie., substitutions vs. indels), ground-truth labels (ie., experimental measurement from DMS vs. clinical annotations), and model training regime (ie., zero-shot vs. supervised).

MUTATION TYPES

We curate benchmarks for two types of protein mutations – *substitutions* and *indels* (insertions or deletions), each with unique implications for the structure, function, and modeling of proteins.

SUBSTITUTIONS Substitution mutations refer to a change in which one amino acid in a protein sequence is replaced by another. Depending on the properties of the substituted amino acid, this can have varied impacts on the protein’s structure and function, which can range from minimal to drastic. The influence of a substitution largely depends on whether it is conservative (i.e., the new amino acid shares similar properties to the original) or non-conservative. In terms of computational modeling, substitutions are the most commonly addressed mutation type, and the majority of mutation effect predictors support substitutions.

INDELS Indel mutations correspond to insertions or deletions of amino acids in protein sequences. While indels can affect protein fitness in similar ways to substitutions, they can also have profound impacts on protein structure by altering the protein backbone, causing structural modifications inaccessible through substitutions alone^{228,257}. From a computational perspective, indels present a unique challenge because they alter the length of the protein sequence, requiring additional considerations in model design and making it more difficult to align sequences. For instance, the majority of models trained on Multiple Sequence Alignments are typically unable to score indels due to the fixed coordinate system they operate within (see § 1.1.6). Furthermore, when dealing with probabilistic models, comparing relative likelihoods of sequences with different lengths results in additional complexities and considerations. Results on indels have been occluded from this thesis, but can be found at the full original publication¹⁹¹.

DATASET TYPES

The fitness of a protein is a measure of how well a protein can perform its function within an organism. Factors that influence protein fitness are diverse and include stability, folding efficiency, catalytic activity (for enzymes), binding specificity and affinity. To properly capture this diversity, we curated a broad set of experimental assays that map a given sequence to phenotypic measurements that are known or hypothesized to be related to its fitness. We focused on two potential sources of ground truth: Deep Mutational Scanning (DMS) assays and Clinical datasets.

DEEP MUTATIONAL SCANNING ASSAYS Modeling protein fitness landscapes presents a challenge due to the complex relationship between experimentally measured protein fitness, the distribution of natural sequences, and the underlying fitness landscape. It is challenging to isolate a singular, measurable molecular property that reflects the key aspects of fitness for a given protein. In developing ProteinGym, we prioritized assays where the experimentally measured property for each

mutant protein is likely to represent the role of the protein in organismal fitness. The resulting compilation of over 250 DMS assays extends over a wide array of functional properties, including ligand binding, aggregation, thermostability, viral replication, and drug resistance. It encompasses diverse protein families, such as kinases, ion channel proteins, G-protein coupled receptors, polymerases, transcription factors, and tumor suppressors. In contrast to most DMS assay collections that focus exclusively on single amino acid substitutions, ProteinGym includes several assays with multiple amino acid variants. Moreover, it spans different taxa (i.e., humans, other eukaryotes, prokaryotes, and viruses), alignment depths, and mutation types (substitutions vs indels). All details about the curation and pre-processing of these DMS assays are provided in Appendix A.1.3.

MODEL TRAINING REGIME

Lastly, we discriminate in our benchmarks between zero-shot and supervised settings. In the supervised regime we are allowed to leverage a subset of labels to train a predictive model, while in the zero-shot setting we seek to predict the effects of mutations on fitness without relying on the ground-truth labels for the protein of interest. These two settings offer complementary viewpoints of practical importance. For instance, in settings where labels are subject to several biases or scarcely available (e.g., labels for rare genetic pathologies), we need methods with robust zero-shot performance. In cases where we seek to design new proteins that simultaneously optimize several properties of interest (e.g., binding affinity, thermostability) and we have collected a sufficiently large number of labels for each target, supervised methods are more appropriate. The need to rely on labels is even more pronounced when we seek to optimize several anti-correlated properties or when evolution is a poor proxy for the property of interest. Predictions obtained in the zero-shot settings may also be used to augment supervised models¹¹¹. The two settings require substantially different evaluation frameworks, which we detail in § 1.1.6.

1.1.6 EVALUATION FRAMEWORK

ZERO-SHOT BENCHMARKS

DMS ASSAYS In the zero-shot setting we predict experimental phenotypical measurements from a given assay, without having access to the labels at training time. Due to the often non-linear relationship between protein function and organism fitness²⁸, the Spearman’s rank correlation coefficient is the most generally appropriate metric for model performance on experimental measurements. We use this metric similarly to previous studies^{110,207,174}. However, in situations where DMS measurements exhibit a bimodal profile, rank correlations may not be the optimal choice. Consequently, for these instances, we supplement our performance assessment with additional metrics, namely the Area Under the ROC Curve (AUC), and the Matthews Correlation Coefficient (MCC), which compare model scores with binarized experimental measurements. Furthermore, for certain goals (e.g., optimizing functional properties of designed proteins), it is more important that a model is able to correctly identify the most functional protein variants, rather than properly capture the overall distribution of all assayed variants. Thus, we also calculate the Normalized Discounted Cumulative Gains (NDCG), which up-weights a model if it gives its highest scores to sequences with the highest DMS value. We also calculate Top K Recall, where we select K to be the top 10% of DMS values. To avoid placing too much weight on properties where we have many assays (e.g., thermostability), we first compute each of these metrics within groups of assays that measure similar functions. The final value of the metric is then the average of these averages, giving each functional group equal weight. We refer to the corresponding value as ‘corrected average’.

BASELINES We implement a diverse set of 50+ zero-shot baselines that may be grouped into alignment-based models, protein language models, inverse folding models and ‘hybrid’ models. These baselines were selected as they were all state-of-the-art at their time of publication accord-

ing to their respective benchmarks; as these benchmarking sets can vary across papers and fields, it is imperative to have a single standardized comparison of these models to each other, which we implement here.

Alignment-based models, such as site-independent and EVmutation models¹¹⁰, DeepSequence²⁰⁷, WaveNet²²⁷, EVE⁸³ and GEMME¹⁴⁶, are trained on Multiple Sequence Alignments (MSAs). Protein language models are trained on large quantities of unaligned sequences across protein families. They include UniRep⁸, the RITA suite¹⁰¹, the ESM1 and ESM2 suite^{208,174,155}, VESPA¹⁶⁵, the CARP suite²⁸⁸ and the ProGen2 suite¹⁸⁸. Inverse Folding models learn sequence distributions conditional on an input structure¹¹⁶. We include here ProteinMPNN⁵⁴ which is trained on structures in the PDB, MIF²⁹⁰ trained on CATH4.2⁵⁵, and ESM-IF1¹¹² which is trained on the PDB and a dataset of AlphaFold2 folded structures. Hybrid models combine the respective strengths of family-specific alignment-based and family-agnostic language models, such as the MSA Transformer²⁰⁵, evotuned UniRep⁸, Tranception¹⁹⁰ and TranceptEVE¹⁹³.

Because of the variable length of sequences subject to insertion or deletion mutations, alignment-based methods with fixed matrix representations of sequences are unable to score indels. However, profile Hidden Markov Model (HMM) and autoregressive models include explicit or implicit probabilities of indels at each position. Both are trained on homologous sequences recovered with an MSA and expanded to include insertions. The masked-marginals heuristic¹⁷⁴ used to predict protein fitness with protein language models trained with a masked-language modeling objective (e.g., ESM-1v, MSA Transformer) does not support indels (see Appendix A.1.4). We thus only report the performance of the following baselines: Tranception¹⁹⁰, TranceptEVE¹⁹³, WaveNet²²⁷, HMM⁶⁸, ProGen2¹⁶¹, UniRep⁸, RITA¹⁰¹ and ProtGPT2⁷⁷.

1.1.7 RESULTS

SUBSTITUTION BENCHMARKS

We follow the experimental protocol described in § 1.1.6 and report our main results on the zero-shot DMS benchmarks in Table 1.1. TranceptEVE emerges as the best overall method across the various settings. One of the key objectives of ProteinGym benchmarks is to analyze performance across a wide range of regimes to guide model selection depending on the objectives of the practitioners. To that end we also provide a performance breakdown across MSA depth, mutational depth and taxa where relevant (see Appendix A.1.5 and supplements). While TranceptEVE tops the ranking across the majority of metrics and settings, GEMME achieves the best performance in several categories, such as assays of viral or non-human eukaryotic proteins, and low and medium depth MSAs. While we report average performance per metric in Table 1.1, the *distribution* of scores across assays is also insightful. For instance, certain models are heavily penalized in aggregate rankings due to very poor performance on a handful of assays (e.g., ESM-1v), such that looking at the median performance in lieu of the average provides a complementary viewpoint. Furthermore, although most models rank similarly under Spearman and NDCG, some have comparatively better performance in one over the other (Fig. 1.2). Superior ranking under NDCG may suggest a model is better at predicting the top end of a score distribution, which may be a desirable feature when using models for design and optimization. Many of the alignment-based methods (e.g. EVmutation, WaveNet) exhibit this behavior (Fig. A1). Models with higher relative Spearman (e.g., ESM-1v and ESM-2) may be more effective for cases where the model needs to learn the full property distribution well, such as with mutation effect prediction.

Model type	Model name	Spearman	AUC	MCC	NDCG	Recall
Alignment-based	Site-Independent	0.359	0.696	0.286	0.747	0.201
	WaveNet	0.373	0.707	0.294	0.761	0.203
	EVmutation	0.395	0.716	0.305	0.777	0.222
	DeepSequence (ensemble)	0.419	0.729	0.328	0.776	0.226
	EVE (ensemble)	0.439	0.741	0.342	0.783	0.230
	GEMME	0.455	0.749	0.352	0.777	0.211
Protein language	UniRep	0.190	0.605	0.147	0.647	0.139
	CARP (640M)	0.368	0.701	0.285	0.748	0.208
	ESM-1b	0.394	0.719	0.311	0.747	0.203
	ESM-2 (15B)	0.401	0.720	0.314	0.759	0.208
	RITA XL	0.372	0.707	0.293	0.751	0.193
	ESM-1v (ensemble)	0.407	0.723	0.320	0.749	0.211
	ProGen2 XL	0.391	0.717	0.306	0.767	0.199
	VESPA	0.436	0.742	0.346	0.775	0.201
Hybrid	UniRep evotuned	0.347	0.693	0.274	0.739	0.181
	MSA Transformer (ensemble)	0.434	0.738	0.340	0.779	0.224
	Tranception L	0.434	0.739	0.341	0.779	0.220
	TranceptEVE L	0.456	0.751	0.356	0.786	0.230
Inverse Folding	ESM-IF1	0.422	0.730	0.331	0.748	0.223
	MIF-ST	0.401	0.718	0.311	0.766	0.226
	ProteinMPNN	0.258	0.639	0.196	0.713	0.186

Table 1.1: Zero-shot substitution DMS benchmark Corrected average of Spearman’s rank correlation, AUC, MCC, NDCG@10%, and top 10% recall between model scores and experimental measurements on the ProteinGym substitution benchmark.

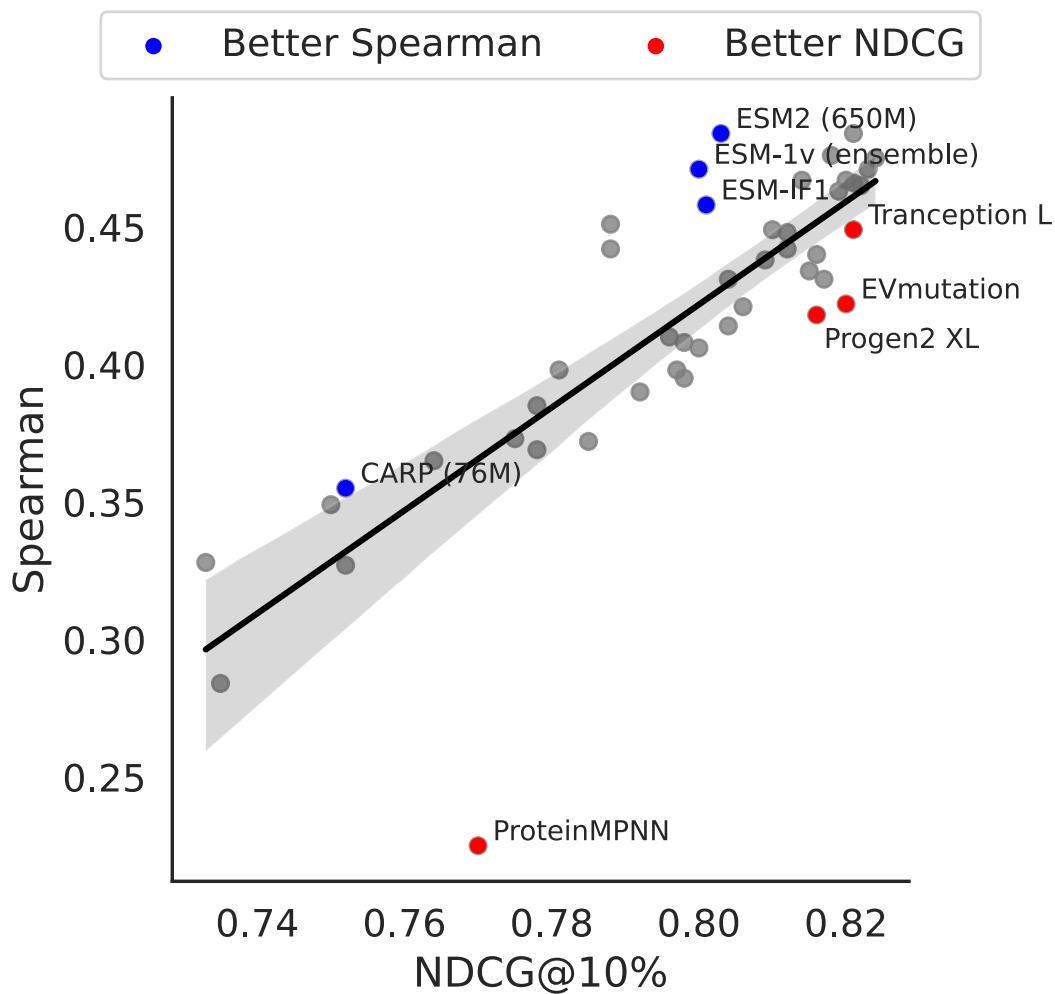


Figure 1.2: Comparing baselines across performance metrics The zero-shot models' median NDCG@10% (x-axis) against median Spearman (y-axis) on the DMS substitutions.

1.1.8 RESOURCES

CODEBASE A key contribution of our work is the consolidation of the numerous baselines discussed in § 1.1.6 in a single open-source GitHub repository. While the main code for the majority

of these baselines is publicly available, it often does not support fitness prediction out-of-the-box or, when it does, the codebase does not necessarily provide all the required data processing logic (e.g., pre-processing of MSAs in MSA Transformer) or handle all possible edge cases that may be encountered (e.g., scoring of sequences longer than context size in the ESM suite). Our [GitHub repository](#) addresses all of these gaps and provides a consistent interface that will aid in the seamless integration of new baselines as they become available.

PROCESSED DATASETS We also make publicly available all processed datasets used in our various benchmarks in a consistent format, including all DMS assays, model scores, ClinVar/gnomAD datasets, predicted 3D structures and Multiple Sequence Alignments required for training and scoring (see Section A.1.3 for more details).

WEBSITE Lastly, we developed a user-friendly [website](#) in which all benchmarks are accessible, with functionalities to support drill analyses across various dimensions (e.g., mutational depth, taxa) and exporting capabilities.

1.1.9 CONCLUSION

ProteinGym addresses the lack of large-scale benchmarks for the robust assessment of models developed for protein design and fitness prediction. It facilitates the direct comparison of methods across several dimensions of interest (e.g., MSA depth, mutational depth, taxa), based on different ground truth datasets (e.g., DMS assays vs Clinical annotations), and in different regimes (e.g., zero-shot vs supervised). We expect the ProteinGym benchmarks and the various data assets we publicly release along with them, to be valuable resources for the Machine Learning and Computational Biology communities, and we plan to continue updating the benchmarks as new assays and baselines become available.

1.2 STRUCTSEQ: LEVERAGING NOVEL INSIGHTS ON FUNCTIONAL ADVANTAGES OF PROTEIN GENERATIVE MODEL CLASSES

1.2.1 PREFACE TO STRUCTSEQ

The following was presented as a workshop paper to the Generative AI for Biology and Machine learning for Structural Biology workshops held at NeurIPS 2023. This work builds off of the curated ProteinGym database described above and performs explorative analyses to identify whether certain generative model classes have predictive advantages for particular biological functions. In ProteinGym we ask the question, "how can we fairly evaluate different generative models for design performance?" and answer it by developing an evaluation paradigm to fairly benchmark popular models. In the following paper we ask the question, "what sorts of biological signal are these models learning and what specific design tasks are they useful for?".

1.2.2 ABSTRACT

Deep generative models of protein sequence and inverse folding models have shown great promise as protein design methods. While sequence-based models have shown strong zero-shot mutation effect prediction performance, inverse folding models have not been extensively characterized in this way. As these models use information from protein structures, it is likely that inverse folding models possess inductive biases that make them better predictors of certain function types. Using the collection of model scores contained in the newly updated ProteinGym, we systematically explore the differential zero-shot predictive power of sequence and inverse folding models. We find that inverse folding models consistently outperform the best-in-class sequence models on assays of protein thermostability, but have lower performance on other properties. Motivated by these findings, we develop StructSeq, an ensemble model combining information from sequence, multiple sequence

alignments (MSAs), and structure. StructSeq achieves state-of-the-art Spearman correlation on ProteinGym and is robust to different functional assay types.

1.2.3 INTRODUCTION

Generative modeling has emerged as a leading approach for mutation effect prediction and protein design. In evolution, protein selection fitness is driven by multiple features including the expressibility and stability of the protein, and the efficacy of its function (e.g. enzymatic activity and binding affinity). Generative sequence models are trained on protein sequences across evolution and capture a rich understanding of protein fitness. Likelihoods from these models have been found to correlate highly with many different aspects of fitness^{223,110,207}. For design, these methods have been used to generate novel sequences and optimize proteins for various functions^{216,255,227,222,24,189,102}. While mutation effect prediction and design have often been treated as separate goals, the distribution learned by these generative sequence models draws a link between these tasks. If the probabilities from a model correlate well with a protein property, designs from that model will be enhanced for that property^{255,127}.

While the sequence contains all the information required to fold most proteins¹⁴, and an ideal model could thus derive its predictions from sequence alone, real-world sequence models underperform due to limited evolutionary data, ill-suited inductive biases, and the inherent performance trade-off between density estimation and fitness estimation²⁷⁶. Therefore, models imbued with task-specific inductive biases may be better suited to predicting specific functions. Furthermore, if protein engineers know which inductive biases correlate with each protein property, they can adapt their design models to promote the properties they are most interested in.

In this work, we delineate the advantages that different generative models have on different functions. We find that, while sequence models outperform inverse folding models on zero-shot prediction of organismal fitness, inverse folding models have superior predictive power on thermostability.

Motivated by this, we develop StructSeq, a hybrid model that builds on the concepts developed in TranceptEVE¹⁹³. We combine global protein sequence information from Tranception¹⁹⁰, family-specific sequence information from EVE⁸³, and structural information from ESM-IF1¹¹². StructSeq has state-of-the-art performance on the ProteinGym DMS substitutions summary benchmark, with top performance for 4/5 of the individual assay function types.

1.2.4 RELATED WORK

Sequence models Generative sequence models are trained on evolutionary datasets of protein sequences. Large protein-language models are trained on repertoires of millions of sequences and aim to learn a general distribution across all proteins^{188,8,190,165,208}. Alignment-based models learn a distribution centered around a particular protein family by training on multiple-sequence alignments (MSAs)^{110,83,146}. These models benefit from the inductive biases afforded by alignments, such as the importance of conserved residues in columns²⁷⁶. Hybrid sequence models combine general protein information with local family information from alignments^{205,193}. This combination makes the log-likelihoods more robust, giving these models top performance on mutation effect prediction¹⁹³.

Inverse folding models condition their sequence likelihoods on a known structure, allowing them to efficiently model interactions that are local in three-dimensional space but distant in two-dimensional residue space^{116,112,54,85,292}. These models take in a protein structure as a graph and leverage graph neural network (GNN) architectures to aggregate information from contacting residues. Some methods construct an input graph using invariant mappings of the residue coordinates^{116,54}, while others leverage SE(3) equivariant GNNs to operate directly on the coordinates themselves^{112,124}. These models learn a structurally informed representation of the residues, which is autoregressively decoded into a sequence. While they are used increasingly for design tasks^{20,127}, inverse folding models have not been extensively characterized for predicting protein fitness, and it is

not known which aspects of protein fitness are best matched by their inductive biases.

1.2.5 INVERSE FOLDING MODELS CONSISTENTLY OUTPERFORM SEQUENCE MODELS ON STABILITY ASSAYS

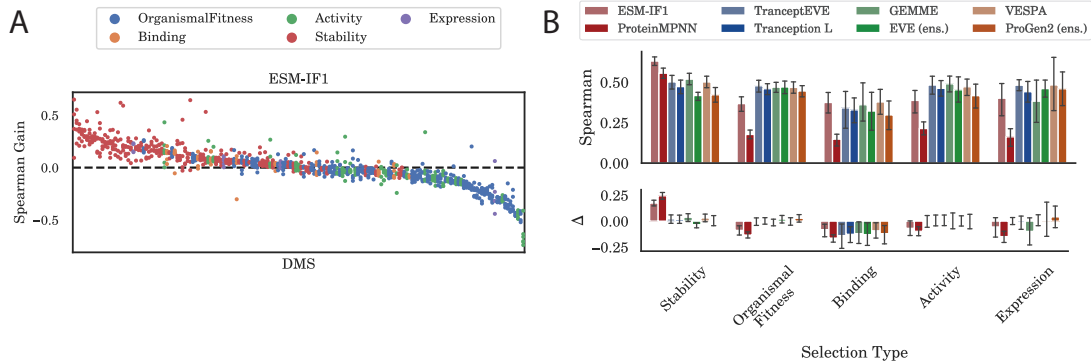


Figure 1.3: Spearman differences between Inverse Folding models and top sequence models (A) The gain in Spearman between ESM-IF1 and the top two sequence models of each sequence model type, calculated for each DMS. We plot the DMSes in order of the median Spearman Gain. The point color corresponds to the type of function assayed in the DMS. (B) Top panel: a barplot of the mean Spearman of each model grouped by function type. Bottom panel: The difference between the Spearman for that function type and the Spearman across all the datasets (Δ). Error bars are SEM for both panels. Models are colored with the primary color corresponding to the model type.

Inverse folding models have been found to have lower performance than sequence models on an extension to the ProteinGym substitutions dataset¹⁹¹. Because inverse folding models have access to structural information, we hypothesized that there might be certain function prediction tasks for which they have an advantage over sequence models.

To investigate this, we looked at the difference in Spearman rank correlation (Spearman Gain) between two inverse folding models (ESM-IF1¹¹² and ProteinMPNN⁵⁴) and the top two models of each sequence model type in the ProteinGym substitutions dataset (Fig 1.3A). These were GEMME¹⁴⁶ and EVE (ens.)⁸³ (alignment-based); VESPA¹⁶⁵ and ProGen2 (ens.)¹⁸⁸ (protein language); and Tranception L¹⁹⁰ and TranceptEVE¹⁹³ (hybrid sequence models). For details on the

DMSes used and on how the models were run, see the appendix. We find that ESM-IF1 and ProteinMPNN consistently outperform the best sequence models on 83/197 and 42/197 of the DMS substitution datasets respectively (Table A9).

We categorized each DMS into 5 function types, corresponding to the kind of function assayed in the DMS ¹³⁰ (See appendix section A.1.3 and A2 for further details on the function type classification). DMSes that could not be clearly put into a category were left out of this analysis (leaving 184 out of 197 assays). Looking at the models' mean Spearman grouped by function type, we find that the sequence models have similar performance across most of the different functions in ProteinGym, while the inverse folding models have much higher Spearman on the protein thermostability assays, and relatively lower performance on other assay types (Fig. 1.3B). ESM-IF1 outperforms the best sequence models on 55/68 of the stability DMSes in ProteinGym (ProteinMPNN outperforms on 32/68) (Table A9). This suggests that inverse folding models have an advantage over sequence models for predicting mutation effects on stability.

Sequence models can learn information about contacting residues that contribute to the overall protein fold from the evolutionary conservation of residue pairs. This makes evolutionary sequence models useful for predicting structure and also identifying mutations that would disrupt protein folds ^{155,164}. However, inverse folding models condition directly on the structure. Thus, it stands to reason that they would have an advantage in predicting the effect of structure-disrupting mutations. In contrast, the selective constraints that direct organismal fitness are probably better represented in evolutionary sequence datasets. Thus, sequence models like TranceptEVE would outperform an inverse folding model on these function types.

We find that ProteinMPNN and ESM-IF1 have similar performance on the stability DMS assays, but ESM-IF1 is superior on the non-stability assays (Fig. A2). ProteinMPNN was trained on only structures in the PDB while ESM-IF1 was also trained using predicted structures from AlphaFold2 ^{54,112,129}. This suggests that, by being exposed to a broader distribution of sequences

across evolution during training, the conditional sequence distribution learned by ESM-IF₁ is more predictive of a wider array of protein functions.

1.2.6 COMBINING SEQUENCE, ALIGNMENTS, AND STRUCTURE GIVES STATE-OF-THE-ART PERFORMANCE

	Function type				Organismal fitness	
	Expression	Binding	Activity	Stability		
#Assays	6	10	31	68	69	
Model name	Spearman					Mean
EVE (ens.)	0.460	0.322	0.453	0.418	0.470	0.425
GEMME	0.382	0.361	0.490	0.518	0.469	0.444
ProGen2 (ens.)	0.459	0.298	0.417	0.422	0.446	0.408
VESPA	0.483	0.378	0.472	0.501	0.468	0.460
ProteinMPNN	0.162	0.148	0.213	0.555	0.177	0.251
ESM-IF ₁	0.401	0.375	0.387	0.630	0.368	0.432
Tranception L	0.441	0.329	0.462	0.473	0.459	0.433
TranceptEVE	0.481	0.341	0.482	0.502	0.478	0.457
StructSeq	0.549	0.399	0.498	0.633	0.468	0.509

Table 1.2: Spearman performance by function type Mean Spearman for each model across the DMS assays grouped under each function type. The number of assays in each group is shown at the top. The top two models of each model type are shown in the order of: Alignment-based models, Protein language models, Inverse folding models, and Sequence hybrid models. The last row contains the performance for our sequence, MSA, and structure ensemble model, StructSeq. We also report the mean of the Spearmans across function types.

Motivated by this finding, we sought to build a model that would have robust predictive power across a wide array of function types.

We develop StructSeq, a sequence-structure hybrid model that combines best-in-class sequence-based and structure-based models. Consider a protein sequence (X), composed of residues $(x_1, x_2, \dots, x_L)$.

We learn a conditional sequence distribution, conditioned on the protein’s structure (Y) and an MSA containing the protein and homologous sequences (M). Based on Notin et al.¹⁹³ we linearly combine sequence information from Tranception with retrieval ($\log P_{TR}(x_i|x_{<i})$), with MSA information from EVE ($\log P_E(x_i|M)$) as in Notin et al.¹⁹³) and structural information from ESM-IF1 ($\log P_I(x_i|Y)$):

$$\log P(x_i|x_{<i}, Y, M) = \alpha_x \log P_{TR}(x_i|x_{<i}) + \alpha_m \log P_E(x_i|M) + \alpha_y \log P_I(x_i|Y, x_{<i})$$

For our experiments, we set $\alpha_x = \frac{1}{2}(1 - \alpha_p)$ and $\alpha_m = \frac{1}{2}\alpha_p$, where α_p is a constant between 0 and 1, that depends on the MSA depth (as described in Notin et al.¹⁹³). We set $\alpha_y = \frac{1}{2}$, so that there is an equal contribution of sequence and structure information to the log-likelihood. Alternatively, to maximize predictive power for a particular property, property-dependent weight sets can be used to upweight the sequence or structure contribution accordingly.

We find that StructSeq has SOTA performance on the DMS assays of each function type in ProteinGym substitutions, except for the organismal fitness assays, for which it has close to top Spearman. Aggregating the mean Spearman for the different selection categories, we find that StructSeq is the best-performing model. Thus, ensembling sequence and structure information produces a model that is robust across multiple protein function types.

1.2.7 CONCLUSION AND FUTURE DIRECTIONS

In this work, we show that inverse folding models are superior zero-shot predictors of protein stability over sequence models. We leverage these findings and present an ensemble method combining sequence, MSAs, and structure which has SOTA summary performance on ProteinGym substitutions and on individual subtypes of DMSes.

As our ensemble approach combines information sources linearly, it is straightforward to up-weight the contribution of the inverse folding model in the fitness predictions and in the designs. This setting would be useful for engineering work in which one is designing de novo stable proteins or optimizing the stability of a protein. As an alternative to ensembling model contributions through their log-likelihoods, it could be of interest to explore learning a joint representation with information flowing from the learned representations of each model sub-unit²⁹⁶. These representations could have utility for downstream design and prediction tasks.

We found that all models have low performance on assays of binding relative to other DMS types (Fig 1.3, Table 1.2). Predicting binding affinity is of great interest for developing therapeutic antibodies and exploring protein-protein interactions. Future work will develop novel methods with higher gains in predicting binding mutation effects.

2

Kernel-Based Evaluation of Conditional Biological Sequence Models

2.1 PREAMBLE

In the previous chapter, we developed an evaluation paradigm to understand how useful our generative models are for zero-shot protein design, and thereafter used this to identify functional advan-

tages for particular model classes; this is crucial information for practitioners to have when selecting a model for a particular design task. However, this paradigm requires functional measurements of sequences within a particular protein family; this may not be readily available for any given protein of interest even if we do have sequencing or structure data for that protein to train a generative model. Suppose a practitioner is working on a new protein family. In that case, they need a way to understand what model would work well for their protein. At the very least we would want to know if our model of choice has learned the distribution around our protein of choice well. One way to estimate a model's fit is using test measures such as perplexity; however, higher perplexities on one protein family does not necessarily indicate a model has fit that distribution worse. To address this, in this chapter, we develop evaluation metrics based on goodness-of-fit tests that directly measure how well a model has learned a distribution relative to a test set. By specifying the test set to a protein family of interest, we can get an absolute measure of how well the model has learned that protein family. Additionally, we develop metrics for model reliability to give practitioners insight into how much they can trust the likelihoods from the model for sampling and scoring.

The following paper was initially published in the forty-first International Conference on Machine Learning (2024). This was a collaboration with Pierre Glaser, Alissa Hummer, Debora Marks and Alan Amin. I co-led the direction of the project and developed the application focus on inverse folding methods as a useful case study of conditional generative models for protein design. I helped design and run the experiments performed in the paper and co-wrote the paper with the first author. Theoretical work was led by P.G. and A.A. and the codebase was developed by P.G.

2.2 ABSTRACT

We propose a set of kernel-based tools to evaluate the designs and tune the hyperparameters of conditional sequence models, with a focus on problems in computational biology. The backbone of

our tools is a new measure of discrepancy between the true conditional distribution and the model’s estimate, called the Augmented Conditional Maximum Mean Discrepancy (ACMMD). Provided that the model can be sampled from, the ACMMD can be estimated unbiasedly from data to quantify absolute model fit, integrated within hypothesis tests, and used to evaluate model reliability. We demonstrate the utility of our approach by analyzing a popular protein design model, ProteinMPNN. We are able to reject the hypothesis that ProteinMPNN fits its data for various protein families, and tune the model’s temperature hyperparameter to achieve a better fit.

2.3 INTRODUCTION

Conditional sequence models constitute one of the most prominent model classes of modern machine learning. Such models have allowed progress in longstanding problems in fields ranging from natural language generation to biomedical applications such as genomics and protein design. Abstracting away the precise nature of the data, the objective common to many of these problems can be summarized as the prediction of high-dimensional discrete-valued sequences, given some possibly high-dimensional input information about the sequence. For example, in protein design, inverse folding models⁵⁴ seek to learn the conditional distribution of amino acid sequences (proteins) that are likely to fold to a given input protein backbone 3D geometry, or *structure*.

In such problems, it is crucial to evaluate the properties of the trained model. Model evaluation can help assess the risk of using the model’s predictions in the real world, such as performing in-vitro experiments (a time-intensive process), guide hyperparameter searches, and deepen one’s understanding of the model’s behavior. Two properties are particularly important to measure: the first one is model *accuracy*, which describes how well the model approximates the true conditional distribution of the target variable given the input. Models with high accuracy have learned the underlying structure of the data, suggesting a high potential value in deploying them in real-world

applications. However, in practice, it is likely that models will not be perfectly accurate. Inaccurate models can still be useful as long as they fall back to conservative guesses (in the extreme case, the prior distribution) when they are uncertain. From a statistical perspective, this property is known as *reliability*^{32,264,278}, and will be the second property of interest in this work.

Given a set of real samples, the standard approach to evaluate models in protein design consists in using log-likelihoods or sequence recovery^{54,112,85}. However, log-likelihoods cannot be used to evaluate reliability, and are only *relative* measures of accuracy: these methods can only be used to compare models and would not alert the practitioner for example if all models make very poor predictions. Instead, to assess how far a model is from being optimally accurate and consistent — and thus the potential value in improving it, by for example collecting more data or increasing its complexity, one should consider *absolute* rather than *relative* metrics, that is, metrics that not only allow one to compare models to each other, but also to evaluate a single model’s performance without any other point of comparison. For these metrics to have practical value, they should come with estimators computable from data samples. These estimators should be efficiently computable, recover the true metric as in the large sample size limit (i.e. be *consistent*), and preferably be centered around the true value of the metric (i.e. be *unbiased*). Finally, to factor out the statistical error coming from estimating these metrics using a finite number of samples, these metrics should be integrable into hypothesis tests built to detect *statistically significant* mismatches between the model and the data.

Contributions In this work, we introduce a set of absolute evaluation metrics for measuring the accuracy and the reliability of conditional sequence models. Both our metrics are grounded in a new measure of divergence between conditional probability distributions, which we call the Augmented Conditional Maximum Mean Discrepancy (ACMMD), which extends the kernel-based conditional goodness-of-fit framework of Jitkrittum et al.¹²⁶, Glaser et al.⁹¹, Widmann et al.²⁷⁸ to the case of sequence-valued variables. We analyze the statistical properties of our proposed metrics, which can be estimated using samples from the data and the model. Under certain conditions, we show that

the ACMMD is able to detect any mismatch between the model and the data. In addition, we integrate the ACMMD into hypothesis tests to detect such mismatches from the model and the data samples. We showcase the utility of our methods by using them in an in-depth analysis of a popular inverse folding model - ProteinMPNN⁵⁴. Our results demonstrate the theoretical properties of our methods, while also providing insight as to how to gauge the certainty and applicability of ProteinMPNN for designing proteins of varying topologies and evolutionary families.

2.4 PROBLEM SETTING

We consider the problem of predicting a discrete sequence-valued variable we are designing $Y \in \mathcal{Y}$, for example a biological sequence, conditionally on a variable $X \in \mathcal{X}$ at our disposal. The predicted sequence Y is allowed to have an arbitrary length, e.g. $\mathcal{Y} = \cup_{\ell=1}^{\infty} \mathcal{A}^{\ell}$, where $\mathcal{A}$ is a finite set. In protein design, X could be the 3D structure of a protein (e.g. $\mathcal{X} = \cup_{\ell=1}^{\infty} \mathbb{R}^{3\ell}$) and Y the sequence of amino acids making up the protein, in which case $\mathcal{A}$ is the set of amino acids. Given a large number of i.i.d measurements $\{X_i, Y_i\}_{i=1}^{N_T}$ from a distribution $\mathbb{P}(X, Y)$, for example pairs of sequences and structures from the Protein Data Bank¹¹⁶, we train a *predictive* model $Q_{\downarrow} : x \mapsto Q_{\downarrow|x}$ that takes in a value x and outputs a distribution on Y , $Q_{\downarrow|x}(Y)$ that attempts to match the true conditional $\mathbb{P}(Y|X = x)$, denoted $\mathbb{P}_{\downarrow|x}(Y)$ in this work. After training, we are interested in quantifying how accurately $Q_{\downarrow}$ approximates $\mathbb{P}_{\downarrow}$ on average across all values of x after training, using a held-out set of samples $\{X_i, Y_i\}_{i=1}^N \sim \mathbb{P}(X, Y)$. Quantifying the accuracy of $Q_{\downarrow}$ is known as the *conditional goodness-of-fit* problem, and we address it in Section 2.5. Furthermore, we will also be interested in quantifying the reliability of $Q_{\downarrow}$, a task which we address in Section 2.6.

2.5 CONDITIONAL GOODNESS-OF-FIT WITH ACMMD

In this section, we propose a metric that quantifies the accuracy of a predictive sequence model. We will show that this metric satisfies many desirable properties: first, it is absolute and able to detect any differences between conditional distributions. Second, it can be unbiasedly and efficiently estimated using samples from the model and the data distribution. Third, it can be used in hypothesis tests to detect statistically significant mismatches from such samples.

2.5.1 THE AUGMENTED CONDITIONAL MMD

We now propose a method to quantitatively evaluate the conditional goodness-of-fit of $Q_{\downarrow}$ to $\mathbb{P}_{\downarrow}$. Our approach consists in constructing a *divergence* $D(\mathbb{P}_{\downarrow}, Q_{\downarrow})$, between the conditional distribution of Y given X and the model $Q_{\downarrow}$. By definition, this divergence should satisfy:

$$\begin{aligned} (i) \quad & D(\mathbb{P}_{\downarrow}, Q_{\downarrow}) \geq 0 \\ (ii) \quad & D(\mathbb{P}_{\downarrow}, Q_{\downarrow}) = 0 \iff \mathbb{P}_{\downarrow|x} = Q_{\downarrow|x}, \mathbb{P}(X)\text{-a.e.} \end{aligned} \tag{2.1}$$

Combined, these two properties ensure that $D(\mathbb{P}_{\downarrow}, Q_{\downarrow})$ is *absolute*, e.g. assigns the known value lowest value 0 to the best possible model, and is able to distinguish any mismatch between the model and the data, which is crucial to prevent blind spots in our evaluation. We borrow the idea of comparing $Q_{\downarrow}$ with $\mathbb{P}_{\downarrow}$ by comparing the joint $\mathbb{P}(X, Y)$ with a joint that keeps the same marginal $\mathbb{P}(X)$ but swaps $\mathbb{P}_{\downarrow}$ with $Q_{\downarrow}$. These two joint distributions are equal if and only if $Q_{\downarrow}$ and $\mathbb{P}_{\downarrow}$ match almost everywhere. To compare these two distributions, we will use the Maximum Mean Discrepancy (MMD)⁹⁷ given by:

$$\text{MMD}(Q_1, Q_2) = \sup_{\substack{f \in \mathcal{H}_{\mathcal{Z}} \\ \|f\|_{\mathcal{H}_{\mathcal{Z}}} \leq 1}} \mathbb{E}_{Q_1}[f(Z)] - \mathbb{E}_{Q_2}[f(Z)]. \tag{2.2}$$

Here, $\mathcal{Z}$ is some measurable space, $\mathbb{Q}_1$ and $\mathbb{Q}_2$ are probability measures on $\mathcal{Z}$, and $\mathcal{H}_{\mathcal{Z}}$ is a reproducing kernel Hilbert space (RKHS) of functions from $\mathcal{Z}$ to $\mathbb{R}$ with kernel $k_{\mathcal{Z}}$ ^{2.2}. Applying this general definition to the case at hand, we obtain a measure of accuracy for $Q_{\downarrow}$, defined below.

Definition 2.5.1 (Augmented Conditional MMD). *Let $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ with law $\mathbb{P}_X \otimes \mathbb{P}_{\downarrow}$. Let $Q_{\downarrow}$ be a conditional probability from $\mathcal{X}$ to $\mathcal{Y}$. We define the Augmented Conditional MMD (ACMMD) between $\mathbb{P}_{\downarrow}$ and $Q_{\downarrow}$ as:*

$$\text{ACMMD}(\mathbb{P}_{\downarrow}, Q_{\downarrow}) := \text{MMD}(\mathbb{P}_X \otimes \mathbb{P}_{\downarrow}, \mathbb{P}_X \otimes Q_{\downarrow}) \quad (2.3)$$

where the MMD is evaluated with a user-specified kernel $k_{\mathcal{X} \times \mathcal{Y}}$ on $\mathcal{X} \times \mathcal{Y}$. Here, $\mathbb{P}_X \otimes \mathbb{P}_{\downarrow}$ is defined by $(X, Y) \sim \mathbb{P}_X \otimes \mathbb{P}_{\downarrow} \iff X \sim \mathbb{P}_X, (Y|X=x) \sim \mathbb{P}_{\downarrow, x}$ and similarly for $\mathbb{P}_X \otimes Q_{\downarrow}$.

Choice of kernel for ACMMD The ACMMD requires specifying a kernel on the joint space $\mathcal{X} \times \mathcal{Y}$. In this work, we will focus on the case where $k_{\mathcal{X} \times \mathcal{Y}}$ is the *tensor product* kernel $k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$ of two kernels $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ on $\mathcal{X}$ and $\mathcal{Y}$ respectively:

$$k_{\mathcal{X} \times \mathcal{Y}}((x, y), (x', y')) = k_{\mathcal{X}}(x, x')k_{\mathcal{Y}}(y, y') \quad (2.4)$$

This choice is popular in practice, and the resulting ACMMD retains its desirable properties, as we show next.

THE ACMMD IS A DIVERGENCE BETWEEN CONDITIONAL PROBABILITIES The ACMMD writes as divergence (which is symmetric, e.g. a distance) between joint distributions, while we seek to use it to compare conditional distributions. The following lemma shows that the same ACMMD can be formulated in alternative manner that highlights its purpose as a conditional distribution comparator.

Lemma 2.5.2. *Under mild integrability conditions, we have:*

$$\text{ACMMD}(\mathbb{P}_\downarrow, Q_\downarrow) = \left\| T_{K_{\mathcal{X}}}(\mu_{\mathbb{P}_\downarrow} - \mu_{Q_\downarrow}) \right\|_{\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}}}$$

Where $\mu_{\mathbb{P}_\downarrow}$ and $\mu_{Q_\downarrow}$ are the conditional mean embeddings¹⁹⁸ of $\mathbb{P}_\downarrow$ and $Q_\downarrow$, $K_{\mathcal{X}}(x, x') := k_{\mathcal{X}}(x, x')I_{\mathcal{H}_{\mathcal{Y}}}$ (here, $I_{\mathcal{H}_{\mathcal{Y}}}$ the identity operator) is an operator-valued kernel with associated vector-valued RKHS $\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}} \subset L^2_{\mathbb{P}_{\mathcal{X}}}(\mathcal{X}, \mathcal{H}_{\mathcal{Y}})$, and $T_{K_{\mathcal{X}}}$ is its associated integral operator from $L^2_{\mathbb{P}_{\mathcal{X}}}(\mathcal{X}, \mathcal{H}_{\mathcal{Y}})$ to $\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}}$. Moreover, if $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are C_0 -universal*, then it holds that:

$$\text{ACMMD}(\mathbb{P}_\downarrow, Q_\downarrow) = 0 \iff \mathbb{P}_{\downarrow|x} = Q_{\downarrow|x}, \quad \mathbb{P}_X\text{-a.e.}$$

The complete statement (with the full set of assumptions, and the definition of integral operators) and its proof can be found in Section B.1. Lemma 2.5.2 shows that the ACMMD can be understood as the result of a two-step procedure, given by (1) computing the conditional mean embedding $\mu_{\mathbb{P}_\downarrow} : x \mapsto \mathbb{E}_{\mathbb{P}_{\downarrow|x}}[k_{\mathcal{Y}}(y, \cdot)|X = x]$ of $\mathbb{P}_\downarrow$ (resp. of $Q_\downarrow$), which is a function from $\mathcal{X}$ to $\mathcal{H}_{\mathcal{Y}}$, and (2) embed the difference of these conditional mean embeddings into the *vector-valued* RKHS $\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}} \subset L^2_{\mathbb{P}_{\mathcal{X}}}(\mathcal{X}, \mathcal{H}_{\mathcal{Y}})$ with kernel $K_{\mathcal{X}}$, before returning its associated RKHS norm. The second part of the lemma gives sufficient conditions for the ACMMD to discriminate between any non ($\mathbb{P}_X$ -a.e) equal conditional distributions, fulfilling the requirements specified in Equation (2.1): these conditions are to use *universal* kernels $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$. Regarding $k_{\mathcal{Y}}$, this requirement is not very restrictive, as many universal kernels on sequences have been shown to be universal¹⁰. The difficulty in finding a universal $k_{\mathcal{X}}$ will depend on the space $\mathcal{X}$ (unspecified in this work) for the problem at hand.

*A kernel k is C_0 -universal if the associated RKHS $\mathcal{H}_k$ is dense in $C_0(\mathcal{X})$, the space of continuous functions on $\mathcal{X}$ vanishing at infinity²³⁷

ESTIMATING THE ACMMD FROM DATA Crucial to this work is the fact that if the model $Q_{\downarrow}$ can be sampled from for any $x \in \mathcal{X}$, ACMMD^2 will admit tractable unbiased estimators. To see this, we first rewrite ACMMD^2 in a form that will make this property apparent.

Lemma 2.5.3. *Let $Z := (X, Y, \tilde{Y})$ the triplet of random variables with law[†] $\mathbb{P}_X \otimes \mathbb{P}_{\downarrow} \otimes Q_{\downarrow}$. Then, under the integrability assumptions of Lemma 2.5.2, we have that:*

$$\text{ACMMD}^2(\mathbb{P}_{\downarrow}, Q_{\downarrow}) = \mathbb{E}_{Z_1, Z_2} [h(Z_1, Z_2)]$$

where Z_1, Z_2 are two independent copies of Z and h is a symmetric function given by:

$$\begin{aligned} h(Z_1, Z_2) &:= k_{\mathcal{X}}(X_1, X_2)g((Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2)) \\ g((Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2)) &:= k_{\mathcal{Y}}(\tilde{Y}_1, \tilde{Y}_2) + k_{\mathcal{Y}}(Y_1, Y_2) \\ &\quad - k_{\mathcal{Y}}(\tilde{Y}_1, Y_2) - k_{\mathcal{Y}}(Y_1, \tilde{Y}_2) \end{aligned}$$

Lemma 2.5.3, proved in Section B.4, expresses ACMMD^2 as a double expectation given two independent samples of $(X, Y, \tilde{Y}) \sim \mathbb{P}_X \otimes \mathbb{P}_{\downarrow} \otimes Q_{\downarrow}$. Leveraging this fact, we can derive an unbiased and consistent estimator for ACMMD^2 .

Lemma 2.5.4. *Let $\{X_i, Y_i, \tilde{Y}_i\}_{i=1}^N \stackrel{i.i.d}{\sim} \mathbb{P}_X \otimes \mathbb{P}_{\downarrow} \otimes Q_{\downarrow}$ be samples from the data and the model. Then an unbiased estimator $\widehat{\text{ACMMD}}^2(\mathbb{P}_{\downarrow}, Q_{\downarrow})$ of $\text{ACMMD}^2(\mathbb{P}_{\downarrow}, Q_{\downarrow})$ is given by:*

$$\frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} h((X_i, Y_i, \tilde{Y}_i), (X_j, Y_j, \tilde{Y}_j)) \quad (2.5)$$

Lemma 2.5.4, proved in Section B.4, shows that it is possible to unbiasedly estimate ACMMD^2 even when the analytical model expectations are intractable, provided that one can sample from

[†]Identifying $Q_{\downarrow}$ with its analogue Markov kernel $\tilde{Q}_{\downarrow}$ from $(\mathcal{X} \times \mathcal{Y}, \mathcal{X} \otimes \mathcal{Y})$ such that $\tilde{Q}_{\downarrow}(x, y)(dy') := Q_{\downarrow|x}(dy')$.

the model. This estimator takes the form of a U-statistics^{2.2.4} Chapter 5 with symmetric probability kernel h , which are well-studied in the statistics literature. In particular, they provide a generic framework to obtain minimal-variance analogues of unbiased estimators^{2.2.4} Chapter 5, p. 176.

$\widehat{\text{ACMMD}}^2$ is a *consistent* estimator of ACMMD^2 : under the integrability assumptions of Lemma 2.5.2 the strong law of large numbers applies^{2.2.4} Section 5.4, Theorem A, and we have:

$$\widehat{\text{ACMMD}}^2(\mathbb{P}_l, Q_l) \xrightarrow[N \rightarrow \infty]{a.s.} \text{ACMMD}^2(\mathbb{P}_l, Q_l)$$

We provide a more detailed characterization of the asymptotic distribution of $\widehat{\text{ACMMD}}^2$ in Section B.2.

2.5.2 TESTING CONDITIONAL GOODNESS-OF-FIT WITH ACMMD

In the limit of infinitely many samples, a positive ACMMD means that the model and the data differ. However, in practice, when only a finite number of samples are available, our estimate $\widehat{\text{ACMMD}}^2$ is only a noisy version of the true ACMMD^2 , meaning we cannot conclude whether the model fits the data by directly inspecting its value. Instead, we need a procedure that accounts for the estimation noise; we achieve this by using the ACMMD as part of a hypothesis test deciding between two different hypotheses:

$$\begin{cases} H_0 : \text{ACMMD}(\mathbb{P}_l, Q_l) = 0 \\ H_1 : \text{ACMMD}(\mathbb{P}_l, Q_l) > 0 \end{cases}$$

In particular, we construct a test that takes as input a sample $\{X_i, Y_i, \tilde{Y}_i\}_{i=1}^N$ from the data and the model and outputs a (binary) decision to reject (or not) the null hypothesis H_0 based on whether $\widehat{\text{ACMMD}}^2(\mathbb{P}_l, Q_l)$ exceeds a certain threshold. Because of the estimation noise arising from the use of finitely many samples, such a test cannot systematically output the right decision. Nonetheless,

we build our test to ensure a *false rejection* (e.g. reject H_0 while $\mathbb{P}_\cdot = Q_\cdot$ a.e) rate of $\alpha \in (0, 1)$, a common practice in statistical testing⁹⁷. To do so, we would like to set the rejection threshold $q_{1-\alpha}$ to be an estimate of the $1 - \alpha$ quantile of the distribution of $\widehat{\text{ACMMD}}^2(\mathbb{P}_\cdot, Q_\cdot)$ under H_0 . However, since $q_{1-\alpha}$ is not available in closed form, we instead compute an estimate $\widehat{q}_{1-\alpha}$ using the wild bootstrap procedure¹⁶. This procedure draws B samples $\{\widehat{\text{ACMMD}}_b^2\}_{b=1}^B$ of the form:

$$\widehat{\text{ACMMD}}_b^2 := \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} W_i^b W_j^b b(Z_i, Z_j) \quad (2.6)$$

where $\{W_i^b\}_{i=1 \dots N}^{b=1 \dots B}$ are i.i.d. Rademacher random variables independent of the data, from which we compute a quantile estimate $\widehat{q}_{1-\alpha}$ of this distribution of samples (see Section B.5 for a precise definition of $\widehat{q}_{1-\alpha}$). Importantly, this procedure guarantees an exact control of the false rejection rate at level α . We prove this fact in Section B.8, where we cast the wild bootstrap procedure as a Monte-Carlo estimation of the distribution of $\widehat{\text{ACMMD}}^2$ when $\mathbb{P}_\cdot = Q_\cdot$, which is valid non-asymptotically. Our test, which we call the ACMMD test, is summarized in Algorithm 1. To the best of our knowledge, this is the first conditional goodness-of-fit test that is applicable to sequence models.

2.6 ASSESSING RELIABILITY WITH ACMMD

In practice, our model $Q_\cdot$ may not fit the data perfectly, and it is important to distinguish (at a given level of inaccuracy) models that remain consistent with their training data from ones that fail more drastically. In this section, we show how the ACMMD can be used to evaluate model reliability, a statistical property capturing model and data consistency.

Problem Setting A model $Q_\cdot$ is said to be reliable^{32,264,278} if the distribution of the target Y given that the model made a specific prediction q is this prediction q itself, e.g. if:

$$q = \mathbb{P}(Y \in \cdot \mid Q_{|X} = q) \quad \mathbb{P}(Q_{|X})\text{-a.e.} \quad (2.7)$$

Algorithm 1 ACMMMD Conditional Goodness-of-fit Test

Input: $\{X_i, Y_i, \tilde{Y}_i\}_{i=1}^N \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}_X \otimes \mathbb{P}_Y \otimes Q_\downarrow$
Parameters: Level α , kernel k_X , kernel k_Y
 // Estimate ACMMMD using Equation (2.5)
 $\widehat{\text{ACMMMD}}^2 \leftarrow \frac{2}{N(N-1)} \sum_{\substack{i,j=1 \\ i < j}}^N b((X_i, Y_i, \tilde{Y}_i), (X_j, Y_j, \tilde{Y}_j))$
 Sample $\{\widehat{\text{ACMMMD}}_b^2\}_{b=1}^B$ using Equation (2.6)
 $\hat{q}_{1-\alpha} \leftarrow \text{approx. } (1 - \alpha)\text{-quantile of } \{\widehat{\text{ACMMMD}}_b^2\}_{b=1}^B$
if $\widehat{\text{ACMMMD}}^2 \leq \hat{q}_{1-\alpha}$ **then**
 Fail to reject H_0
else
 Reject H_0
end if

Here, $Q_{\downarrow X} \in \mathcal{P}(\mathcal{Y})$ (the space of probability distributions on $\mathcal{Y}$) is the random variable obtained by evaluating the model $Q_\downarrow$ at a random value of the input variable X . Reliability differs from accuracy in that it does not require the model to learn all the information between X and Y , but only to make truthful predictions on average — thus, by assessing reliability, one may be able to detect models that hallucinate non-realistic sequences (such as repeats of the same token) in regions of the input space where they are inaccurate, instead of making a conservative guess, such as falling back to the prior distribution. In particular, reliability can be used as an additional criterion to discriminate between models that are equally accurate. From a theoretical perspective, reliability and accuracy can be handled in a unified manner: indeed, Equation 2.7 shows that reliability is defined as an equality between the conditional distribution of Y given a model prediction q , $\mathbb{P}_{\downarrow q}^Q := \mathbb{P}(Y = \cdot | Q_{\downarrow X} = q)$ and a “model” of this conditional distribution mapping $q \in \mathcal{P}(\mathcal{Y})$ to itself, e.g. $Q_{\downarrow}^{\text{Rel}} : q \mapsto Q_{\downarrow q}^{\text{Rel}} = q$. We thus propose to measure reliability using the ACMMMD (a distance between conditional distributions) between $Q_{\downarrow}^{\text{Rel}}$ and $\mathbb{P}_{\downarrow}^Q$.

Definition 2.6.1 (ACMMD for Reliability). *The Augmented Conditional MMD for reliability (ACMMD–Rel) between $\mathbb{P}_|$ and $Q_|$ as:*

$$\begin{aligned} \text{ACMMD–Rel}(\mathbb{P}_|, Q_|) &:= \text{ACMMD}(\mathbb{P}_|^\mathcal{Q}, Q_|^{\text{Rel}}) \\ &= \text{MMD}(\mathbb{P}_{|Q_{|X}} \otimes \mathbb{P}_|^\mathcal{Q}, \mathbb{P}_{|Q_{|X}} \otimes Q_|) \end{aligned} \tag{2.8}$$

where the ACMMD is evaluated with a user-specified kernel $k_{\mathcal{P}(\mathcal{Y}) \times \mathcal{Y}}$ on $\mathcal{P}(\mathcal{Y}) \times \mathcal{Y}$.

As for the ACMMD, we will restrict our attention to the case where $k_{\mathcal{P}(\mathcal{Y}) \times \mathcal{Y}}$ is a tensor product kernel $k_{\mathcal{P}(\mathcal{Y})} \otimes k_{\mathcal{Y}}$ between a kernel on $\mathcal{P}(\mathcal{Y})$ and a kernel on $\mathcal{Y}$. Comparing the ACMMD–Rel with the ACMMD, we see that the former requires specifying a kernel *on the space of probability measures* on sequences $\mathcal{P}(\mathcal{Y})$ instead of a kernel on $\mathcal{X}$. Two important points must be addressed when working with such kernels. First, in order to have $\text{ACMMD–Rel}(\mathbb{P}_|, Q_|) = 0$ if and only if $Q_|$ is reliable, we must find universal kernels defined on $\mathcal{P}(\mathcal{Y})$. Second, as many kernels on probabilities are intractable, we must design an approximation strategy to estimate the ACMMD–Rel from data.

ACMMD–REL CAN DETECT ANY PATTERN OF UNRELIABILITY Our first goal is to ensure that ACMMD–Rel can detect any pattern of unreliability. As ACMMD–Rel is a specific instance of the ACMMD, we can apply Lemma 2.5.2, stating that if the kernels $k_{\mathcal{P}(\mathcal{Y})}$ and $k_{\mathcal{Y}}$ are universal, then

$$\text{ACMMD–Rel}(\mathbb{P}_|, Q_|) = 0 \iff Q_| \text{ is reliable.} \tag{2.9}$$

The task of finding a universal kernel $k_{\mathcal{Y}}$ on $\mathcal{Y}$ was addressed in Section 2.5; thus, it remains to find a universal kernel $k_{\mathcal{P}(\mathcal{Y})}$ on $\mathcal{P}(\mathcal{Y})$. However, to the best of our knowledge, none of the existing kernels defined on probability distributions^{35,250,251,178,91} have been shown to be universal when $\mathcal{Y}$ is the space of arbitrary-length sequences. In the next proposition, we show that many such kernels

can be constructed by following a simple recipe.

Proposition 2.6.2. *Let $k_{\mathcal{Y}}$ be a kernel on $\mathcal{Y}$ vanishing at infinity (on $\mathcal{Y} \times \mathcal{Y}$). Suppose that $k_{\mathcal{Y}}$ has discrete masses, i.e. that $\delta_y \in \mathcal{H}_{\mathcal{Y}}$ for all sequences $y \in \mathcal{Y}$, where δ_y is the Dirac function at y , and let $\sigma > 0$. Then the kernel $k_{\mathcal{P}(\mathcal{Y})}$ on $\mathcal{P}(\mathcal{Y})$ defined as*

$$k_{\mathcal{P}(\mathcal{Y})}(q, q') := e^{-\frac{1}{2\sigma^2} \text{MMD}^2(q, q')}, \quad (2.10)$$

(where the MMD is computed in $\mathcal{H}_{\mathcal{Y}}$) is a C_0 -universal kernel on the space of probability distributions $\mathcal{P}(\mathcal{Y})$ (under the topology of convergence in distribution or Total Variation, which are identical, see⁹).

The proof, provided in Section B.11, relies on an argument similar to prior work for universal kernels on probability measures³⁵, but tailored to the special case of sequences. Proposition 2.6.2 guarantees that any kernel on $\mathcal{Y}$ vanishing at infinity with the discrete mass property¹⁰ can be used to construct a universal kernel on $\mathcal{P}(\mathcal{Y})$. Kernels with discrete masses are studied in detail in¹⁰. In particular, the tilted Exponentiated Hamming Kernel $\frac{1}{|y||y'|} e^{-\lambda d_H(y, y')}$ (where $|y|$ is the length of the sequence y) is a kernel with discrete masses vanishing at infinity on $\mathcal{Y} \times \mathcal{Y}$, and can thus be used to construct a universal kernel on $\mathcal{P}(\mathcal{Y})$.

ESTIMATING ACMMD-REL FROM DATA To estimate ACMMD-Rel from the data $\{X_i, Y_i\}_{i=1}^N$ and samples from the model $\{\tilde{Y}_i \sim Q_{|X_i}\}_{i=1}^N$, one may try to use the general ACMMD estimator proposed in Lemma 2.5.4, which, specialized to the reliability setting, is given by:

$$\frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} b(Z_i, Z_j)$$

$$b(Z_i, Z_j) := k_{\mathcal{P}(\mathcal{Y})}(Q_{|X_i}, Q_{|X_j}) g((Y_i, \tilde{Y}_i), (Y_j, \tilde{Y}_j))$$

This estimator requires evaluating $k_{\mathcal{P}(\mathcal{Y})}(Q_{|X_i}, Q_{|X_j})$ for pairs i, j . Unfortunately, exact evaluation of these quantities for the universal kernels proposed in Proposition 2.6.2 is in general impossible, as $\text{MMD}^2(Q_{|X_i}, Q_{|X_j})$ contains intractable expectations under $Q_{|X_i}$ and $Q_{|X_j}$. However, MMDs can be unbiasedly estimated using samples from $Q_{|X_i}$ and $Q_{|X_j}$ ^{97,221}. Inspired by this fact, we propose the following estimator:

$$\begin{aligned} \widehat{\text{ACMMD-Rel}}^2 &:= \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} \widehat{b}(Z_i, Z_j) \\ \widehat{b}(Z_i, Z_j) &:= \widehat{k}_{ij} \times g((Y_i, \tilde{Y}_i), (Y_j, \tilde{Y}_j)) \end{aligned} \tag{2.11}$$

Here, $\widehat{k}_{ij}$ is an approximation of $k_{\mathcal{P}(\mathcal{Y})}(Q_{|X_i}, Q_{|X_j})$ obtained by drawing R samples $\{\tilde{Y}_i^r\}_{r=1}^R$ and $\{\tilde{Y}_j^r\}_{r=1}^R$ from $Q_{|X_i}$ and $Q_{|X_j}$, and replacing the $\text{MMD}^2(Q_{|X_i}, Q_{|X_j})$ term in $k_{\mathcal{P}}$ by an unbiased estimate $\widehat{\text{MMD}}_{ij}^2$ computed from these samples. The full estimation procedure is provided in Algorithm 2. This additional approximation step has several implications: first, unlike $\widehat{\text{ACMMD}}^2$, $\widehat{\text{ACMMD-Rel}}^2$ is not unbiased. However, the bias of this estimator can be controlled by increasing the number of samples R used to estimate the MMD. Moreover, we show in the next proposition that the estimator $\widehat{\text{ACMMD-Rel}}^2$ is still consistent provided that R is chosen appropriately.

Proposition 2.6.3. *Assume that $k_{\mathcal{Y}}$ is bounded. Then, if $R \equiv R(N)$, with $\lim_{N \rightarrow \infty} R(N) = +\infty$, $\widehat{\text{ACMMD-Rel}}^2$ converges in probability to ACMMD-Rel^2 as $N \rightarrow \infty$.*

TESTING FOR RELIABILITY WITH ACMMD-REL As an ACMMD, ACMMD-Rel has the potential to be used to test whether a model is reliable given some available data: to do so, one can use Algorithm 1, replacing $\widehat{\text{ACMMD}}^2$ by $\widehat{\text{ACMMD-Rel}}^2$, and performing quantile estimation using the $\widehat{b}(Z_i, Z_j)$ instead of the $b(Z_i, Z_j)$. A full description of the algorithm is provided in Appendix B.13. An important question to answer is whether the approximation of using $\widehat{b}$ instead of b affects the false-rejection rate of the test. We show in the next proposition that this is not the case.

Algorithm 2 Estimating ACMMD–Rel

Input: $\{X_i, Y_i, \tilde{Y}_i\}_{i=1}^N \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}_X \otimes \mathbb{P}_Y \otimes Q$, model Q
Parameters: kernel k_Y
for i **in** 1 **to** N **do**
 $[\tilde{Y}_i^r \sim Q_{|X_i} \text{ for } r \text{ in } 1 \text{ to } R]$
end for
for i, j **in** 1 **to** N **do**
 %Use, e.g. Gretton et al.⁹⁷, Equation 4
 $\widehat{\text{MMD}}_{ij}^2 := \text{estimate_mmd}(\{\tilde{Y}_i^r\}_{r=1}^R, \{\tilde{Y}_j^r\}_{r=1}^R)$
 $\hat{b}_{ij} := e^{-\frac{1}{2\sigma^2} \widehat{\text{MMD}}_{ij}^2} \times g((Y_i, \tilde{Y}_i), (Y_j, \tilde{Y}_j))$
end for
return $\frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} \hat{b}_{ij}$

Proposition 2.6.4. Assume that k_Y is bounded, and $k_{\mathcal{P}(\mathcal{Y})}$ is a kernel of the form of Equation 2.10. Then a reliability test using $\hat{b}(Z_i, Z_j)$ instead of $b(Z_i, Z_j)$ to estimate ACMMD–Rel and its $(1 - \alpha)$ –quantile under H_0 has a false-rejection rate of exactly α .

2.7 RELATED WORK

Goodness-of-fit methods The *goodness-of-fit* problem is a well-studied problem in the statistics and machine learning literature, for which many methods were developed^{46,94,95,11,19}. Impressively, these methods can operate directly from the model’s analytical form, without requiring access to samples from the model – which may be hard to generate. In these works, goodness-of-fit is defined as the problem of evaluating the fit of *unconditional* models to their data, which is unlike the conditional goodness-of-fit problem we consider here. Evaluating conditional goodness-of-fit with kernels was recently studied in Jitkrittum et al.¹²⁶. However, the proposed method requires the output space $\mathcal{Y}$ to be a subset of $\mathbb{R}^d$, and is thus unsuitable for conditional sequence models. The

use of conditional goodness-of-fit metrics to evaluate reliability was also done in Glaser et al.⁹¹, in a method also limited to continuous output spaces. Finally, we note that ACMMD–Rel² recovers an existing calibration metric, the Squared Kernel Calibration Error (SKCE) of Widmann et al.²⁷⁸. However, the latter did not study the problems of universality, tractability and test validity in the case of sequence-valued outputs.

Deep Protein Design Models (Deep Learning–powered) conditional probability models have gained significant momentum in computational biology during the last decade. In particular, such models have revolutionized the protein design field^{127,20}. Inverse folding models are trained on protein structures and sequences in the protein data bank (PDB)^{116,112,54}. They condition a sequence distribution on an input protein structure — thus learning what sequences would likely fold into that structure. The designs from these methods have been shown to be highly stable and retain function²⁴⁶. However, many of the leaps made using these models have used small, simple structural scaffolds (like loop-helix-loop motifs)^{20,271}. Protein engineers interested in leveraging these tools for novel scaffolds need to know how accurate and reliable the model is on average. If the model is too imprecise, one might wish to gather more data and train more bespoke models before using the method to design experiments.

2.8 EXPERIMENTS

We now investigate the behavior and utility of the ACMMD and ACMMD–Rel metrics and tests in practice. We start with a synthetic example showing that ACMMD is a natural measure of model distance. We then perform an extended analysis of a state-of-the-art inverse folding model, Protein-MPNN. We show that ACMMD can detect small perturbations in the model, and that it can be used to tune its temperature parameter. Finally, we analyze the absolute performance of Protein-MPNN.

2.8.1 A TOY SYNTHETIC SETTING

We first study the behavior of ACMMD and ACMMD-Rel in a synthetic setting where the data distribution and the model are simple generative models on sequences. We set the input variable X to be a single scalar p drawn from some distribution $\mathbb{P}_X$ with support in $(0.3, 0.5)$. Y is a sequence of arbitrary length with alphabet $\mathcal{A} = \{A, B, \text{STOP}\}$. We set the conditional distribution of Y given p to be:

$$p(y_n | y_{0:n-1}, x = p) = \begin{cases} A & \text{with probability } p \\ B & \text{with probability } p \\ \text{STOP} & \text{with probability } 1 - 2p \end{cases}$$

so long as $y_{n-1} \neq \text{STOP}$. The model distribution $Q_{|}$ is the same as the data, except for the fact that the first factor $Q_{|p}(y_0)$ is perturbed by a parameter Δp :

$$Q_{|p}(y_0) = \begin{cases} A & \text{with probability } p - \Delta p \\ B & \text{with probability } p + \Delta p \\ \text{STOP} & \text{with probability } 1 - 2p \end{cases}$$

We set the kernel on $\mathcal{Y}$ to be the exponentiated Hamming distance kernel $k_{\mathcal{Y}}(y, y') = e^{-d_H(y, y')}$, where $d_H(y, y')$ is the Hamming distance between y and y' , and $k_{\mathcal{X}}$ to be the Gaussian kernel $k_{\mathcal{X}}(p, p') = e^{-\frac{1}{2}(p-p')^2}$. With such choices, it is possible to show that:

$$\text{ACMMD}(\mathbb{P}_{|}, Q_{|}) = C|\Delta p|$$

for some $C > 0$ that does not depend on Δp , and is computable in closed form for discrete priors on X . The proof and expression of C are given in Section B.15. From this expression, we immedi-

ately see that ACMMD is 0 only if $\Delta p = 0$, a manifestation of Lemma 2.5.2 which guarantees that the ACMMD can detect any mismatch between the model and the data when k_X and k_Y are universal. Moreover, in this case, the ACMMD depends monotonically on the shift $|\Delta p|$. Since $|\Delta p|$ represents a natural measure of how different the model is from the data, this fact suggests that the ACMMD is a natural, well-behaved measure of model distance. Additionally, we plot the average rejection rate of the ACMMD test for various number of samples and shifts in Figure 2.1. The results for $\Delta p = 0$ confirm that our test has the correct specified type-I error rate (0.05). Moreover, we see that the power of the test increases with the number of samples, and the shift $|\Delta p|$.

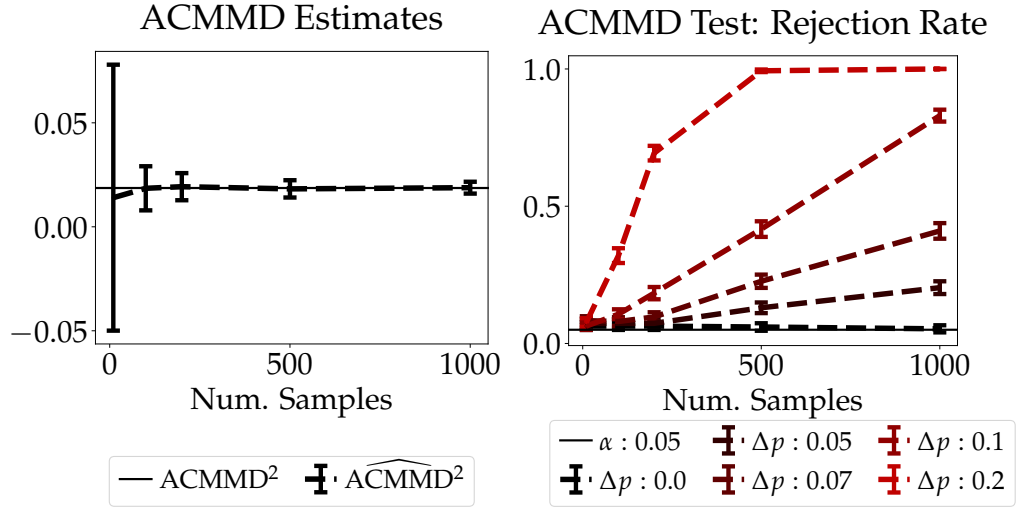


Figure 2.1: Left panel: ACMMD estimates for a fixed shift value $\Delta p = 0.25$ and various number of samples in the synthetic example of Section 2.8.1. The analytic ACMMD value is given by the horizontal line. Right panel: ACMMD test average rejection rate for various number of samples and shifts in the same setting.

2.8.2 ACMMD CASE STUDY: INVERSE FOLDING MODELS

To demonstrate the utility of the ACMMD measures and tests, we apply them to evaluate inverse folding models, a popular model framework used in protein design. Inverse folding models seek a distribution of amino acid sequences that are likely to fold into a given input three-dimensional

structure, as discussed in Section 2.5. We focus our experiments on evaluating ProteinMPNN⁵⁴, a sampleable, commonly used model in this class. The sampling temperature T of ProteinMPNN can also be varied, letting the user control the trade-off between accuracy and diversity of the generated sequences.

Data We leveraged the CATH taxonomy to select a set of diverse (in sequence and structural topologies) protein structures to perform our ACMMD test on. CATH is a taxonomy of protein structures that categorizes proteins according to a hierarchy of structural organization²²⁹. We used the S60 redundancy filtered set which includes proteins that are at least 60% different in sequence identity from each other. Of these, we selected all single domain monomers (proteins where only one topological domain is found in the monomer), and removed any topologies that had fewer than 10 chains in its classification. This left us with 17,540 structures.

Choice of kernel Key to the performance of our metrics is the choice of the kernels $k_{\mathcal{X}}$, $k_{\mathcal{Y}}$ and $k_{\mathcal{P}(\mathcal{Y})}$. For $k_{\mathcal{Y}}$, we propose to use kernels that first embed each element – or *residue* – of a sequence y using an embedding function $\varphi_{\mathcal{Y}} : \mathcal{A} \times \mathcal{Y} \mapsto \mathbb{R}^{d_{\mathcal{Y}}}$, and evaluating a euclidean kernel on $\mathbb{R}^{d_{\mathcal{Y}}}$ on the mean of the resulting embeddings, yielding a kernel of the form:

$$k_{\mathcal{Y}}(y, y') = k_{\mathbb{R}^{d_{\mathcal{Y}}}} \left(\frac{1}{|y|} \sum_{i=1}^{|y|} \varphi_{\mathcal{Y}}(y_i, y), \frac{1}{|y'|} \sum_{i=1}^{|y'|} \varphi_{\mathcal{Y}}(y'_i, y') \right)$$

where we noted $y = (y_1, \dots, y_{|y|})$. As the input space $\mathcal{X}$ is also sequence-valued, we follow the same recipe to construct our a kernel $k_{\mathcal{X}}$, using an embedding function $\varphi_{\mathcal{X}} : \mathbb{R}^3 \times \mathcal{X} \mapsto \mathbb{R}^{d_{\mathcal{X}}}$. Finally, for the kernel on $\mathcal{P}(\mathcal{Y})$, we will use a kernel of the form of Equation (2.10), with kernel $k_{\mathcal{Y}}$ described above to compute the inner MMD. We set our embedding functions $\varphi_{\mathcal{X}}$ and $\varphi_{\mathcal{Y}}$ to a pair of recent pre-trained neural networks that are commonly used for representation learning of protein sequences and structures: Gearnnet²⁹⁵ for structures, and ESM-2¹⁵⁵ for sequences. Such two-step kernels allow us to instill the complex structure present in the distribution of protein structures and

sequences within the ACMMD maximizing the performance and meaningfulness of our evaluation pipeline. Whether Proposition 2.6.2 holds for these kernels is an open question, but we find that they perform well in our experiments.

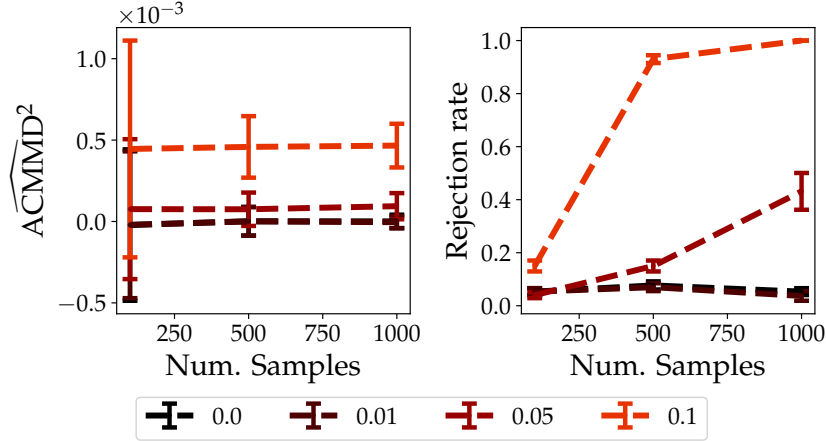


Figure 2.2: Values of $\widehat{\text{ACMMD}}^2$, (left) and of the average rejection rate of the ACMMD test (right) in the setting described in 2.8.2. Each line corresponds to a different value for δT .

THE DISCRIMINATIVE POWER OF ACMMD

We first propose to evaluate the behavior of the ACMMD and its associated test when comparing a known ground truth and a model distribution differing from the ground truth in a controlled manner. To this end, we set the ground truth to be a pre-trained ProteinMPNN model Q_{I}^T with temperature T , and the model to be the same model $Q_{\text{I}}^{T+\delta T}$ with temperature $T + \delta T$. As ProteinMPNN’s probability distribution is a continuous function of T , small changes in T result in small changes in the predicted distribution which will be hard to detect, translating into “low” values for $\widehat{\text{ACMMD}}^2$ relative to larger temperature changes. Conversely, we posit that large changes in T will result in large changes in the model distribution, and will be simpler to detect by the ACMMD. To test these hypotheses, we performed an estimation of $\widehat{\text{ACMMD}}^2$ for a ground truth temperature

$T = 0.1$ (the default in the ProteinMPNN documentation) and $\delta T \in \{0, 0.01, 0.05, 0.1\}$. We used the winged helix-like DNA binding domain superfamily (CATH ID: 1.10.10.10), and performed bootstrap sampling to produce dataset sizes ranging from 100 to 1000, and 100 different random seeds in order to obtain confidence intervals of our estimates.

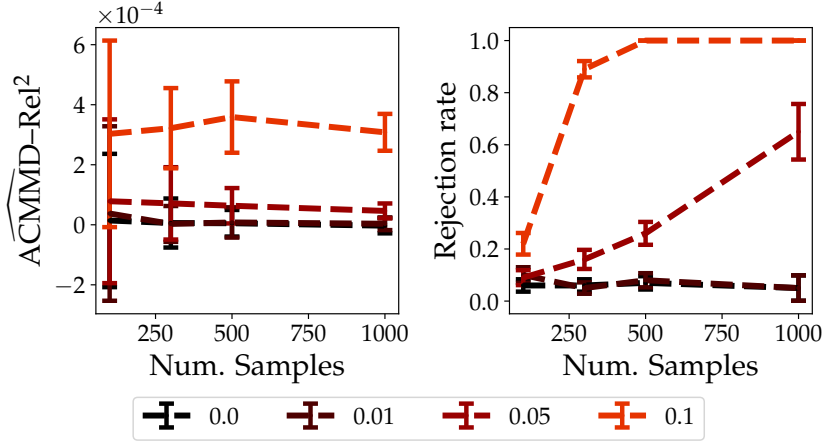


Figure 2.3: Values of $\widehat{ACMMD-Rel}^2$, (left) and of the average rejection rate of the ACMMD-Rel test (right) in the setting described in Section 2.8.2. Each line corresponds to a different value for δT .

The results are shown in Figure 2.2. As expected, $ACMMD^2(Q_{\downarrow}^T, Q_{\downarrow}^{T+\delta T})$ robustly increases with increasing values of δT . Additionally, we performed the ACMMD test of Section 2.5.2 with a target type-I error rate of $\alpha = 0.05$, and 100 permutations to estimate the $1 - \alpha$ quantile of the null distribution for the same values of N and δT , and computed the average rejection rate of the null hypothesis $H_0 : ACMMD^2(Q_{\downarrow}^T, Q_{\downarrow}^{T+\delta T}) > 0$, which, if $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are universal, is equivalent to $H_0 : \delta T = 0$. The results, shown in Figure 2.2 (right), empirically confirm that the ACMMD test controls its type-I error rate and is able to detect differences in temperatures of an order relevant for ProteinMPNN. Similarly, we evaluate the behavior of the ACMMD-Rel, which is used to assess the (lack of) reliability of between model $Q_{\downarrow}^{T+\delta T}$ w.r.t the data $\mathbb{P}_{\downarrow X} \otimes Q_{\downarrow}^T$, the assumption being that $Q_{\downarrow}^{T+\delta T}$ is not reliable when $\delta T \neq 0$. The results are shown in Figure 2.3, and exhibits similar behavior.

EVALUATION OF PROTEINMPNN ON THE CATH DATASET

Now that we have confirmed the discriminative power of the ACMMD on semi-synthetic data, we use our tests to evaluate ProteinMPNN against real-world protein structures and sequences from the CATH dataset. We perform a whole-data evaluation, using samples of 5000 proteins across all families in the dataset. Then we perform a fine-grain evaluation on a subset of CATH superfamilies.

WHOLE-DATA EVALUATION We first study the deviation of ProteinMPNN from the true data by computing $\widehat{\text{ACMMD}}^2$ and estimating its mean and variance by bootstrapping over 10 random seeds. We find that ProteinMPNN with no temperature adjustment ($T = 1.0$) has an $\widehat{\text{ACMMD}}^2$ value of 0.0916 (and a p-value < 0.01). Comparing this to the criterion values obtained on similar dataset sizes in the toy data experiments demonstrates that the model does not fit the test data. This suggests that there is still much room for improvement on solving the inverse folding problem.

ON OPTIMAL TEMPERATURE CHOICES FOR PROTEINMPNN Practitioners vary the sampling temperature as a heuristic method for sampling more certain sequences from ProteinMPNN; lower temperature settings have been found to generate sequences with fewer unrealistic artifacts (e.g. runs of alanines) which fold to more stable structures²⁴⁶. However, the relationship between sampling temperature, model reliability, and design accuracy has not been fully established. To thoroughly evaluate this, designs from different sampling temperatures conditioned on a diverse set of backbone structures would need to be experimentally characterized, which is resource intensive in practice. We leverage the ACMMD to understand at what sampling temperature ProteinMPNN best fits the data, which gives insight as to what temperature is optimum, by computing $\widehat{\text{ACMMD}}^2$ and $\widehat{\text{ACMMD}} - \text{Rel}^2$ for varying temperature values across 10 seeds for each temperature value. The results are shown in Figure 2.4.

First, we observe that reducing the temperature below 1.0 improves both the model’s goodness-

of-fit and its reliability. This corroborates the empirical design success of lowering the sampling temperature, suggesting that greater model fit may increase the quality of samples from the model. The decrease in reliability at higher temperature shows that even though increasing the temperature increases the diversity of the model’s predictions, this diversity does not necessarily capture the one of the data distribution, as for instance the prior would. The optimal temperature from the perspective of goodness-of-fit is 0.4 (which lies outside the suggested temperature range of 0.1-0.3 in the ProteinMPNN documentation⁵⁴). However, we notice that model reliability continues to improve with even lower sampling temperatures while accuracy slightly increases, suggesting a trade-off between reliability and accuracy. Further experiments will determine how this trade-off manifests in the quality of designs from low-temperature settings.

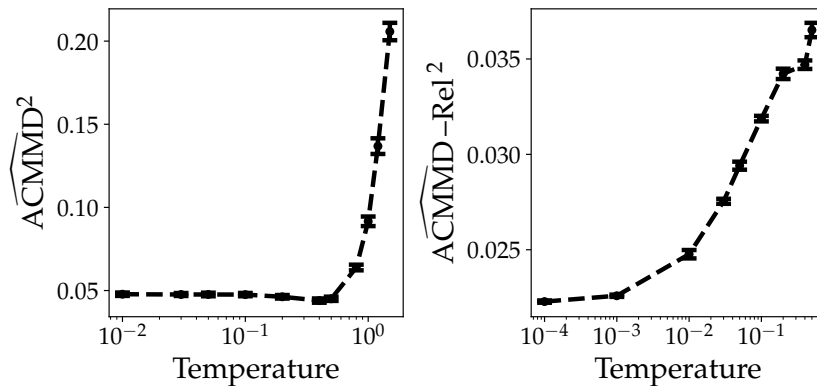


Figure 2.4: Evolution of $\widehat{\text{ACMMD}}^2$ (left) and $\widehat{\text{ACMMD}} - \text{Rel}^2$ (right) between a pre-trained ProteinMPNN model and the CATH S60 reference dataset, for varying temperature values

STRUCTURAL SUPERFAMILY EVALUATION The (H)omologous superfamily tier within the CAT(H) hierarchy groups proteins with the same similar folds and sequence identity. While ProteinMPNN has shown great performance in designing particular structural scaffolds, a practitioner aiming to leverage this model on a yet untested structural family may want some insight as to how well ProteinMPNN may fit the distribution of proteins they are interested in. Thus, we performed

ACMMD evaluation separately on individual superfamilies contained in our dataset to gain insights on what types of structures ProteinMPNN does or does not fit well. We filtered the superfamilies for groupings with at least 500 proteins under a length of 100, yielding 11 families. The results are shown in Figure 2.5. We find that the model fit varies across families and the fit ranking is largely maintained at different temperatures. With no temperature adjustment ($T = 1.0$) the best fit superfamily (lowest $\widehat{\text{ACMMD}}^2$) is the Homeodomain-like proteins (CATH ID: 1.10.10.60). These structures are largely dominated by helical bundles - a class of proteins that ProteinMPNN has demonstrated success on designing^{54,271,20}. While the Immunoglobulin superfamily has the highest fit at lower sampling temperatures, we note that most of an immunoglobulin structure consists of the beta sandwich of the framework, while, for antibody design, engineers are often most interested in the unstructured complementarity determining regions (CDRs) of antibodies^{143,156,123}. As the criterion is calculated across the entire sequence, this may not reflect that ProteinMPNN has learned the distribution of CDR loops well. Further work will extend these tests to focus on subsequences of a domain to answer specific questions of model fit on regions of interest.

2.9 DISCUSSION

Advancing the computational evaluation of conditional sequence models is crucial for accelerating the development of these methods for protein engineering. Given the limitations of current evaluation methods, and leveraging recent advancements in kernel methods for designing tests of goodness-of-fit and calibration, we propose a criterion and its associated test to principledly evaluate protein sequence models for how well they have learned input-conditioned sequence distributions. We discuss the statistical properties of our metrics and develop testing frameworks from them. Finally, we leverage them to investigate the performance of inverse folding models under default and temperature-adjusted settings. We develop novel insights on ideal temperature settings for ProteinMPNN and discuss the trade-off between design accuracy and model calibration that our tests

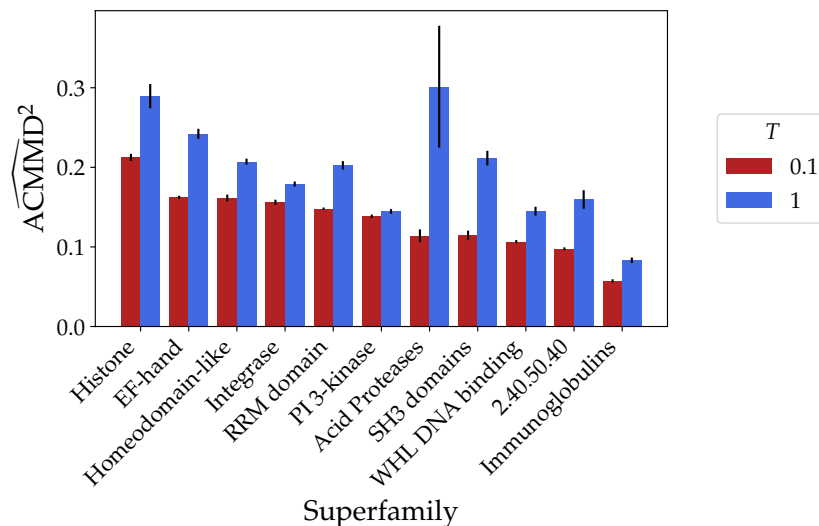


Figure 2.5: Value of $\widehat{ACMMD}^2$ between ProteinMPNN and the CATH S60 reference dataset on a subset of 10 superfamilies for two different temperatures $T = 1.0$ and $T = 0.1$.

demonstrate for lower temperatures. Future work can perform a more fine-grained evaluation, for example investigating which structures in particular cause the model to make unreliable predictions and what features of the model’s predictions do not match the data through the use of witness functions, a by-product of MMDs¹⁵⁷. We also note that protein engineering goals may differ from pure modeling goals, and whether performance under our metrics reflect experimental design success rates requires further investigation to determine. Yet, barring orthogonal *in silico* validation data or experimental testing, our methods offer a powerful framework to test conditional sequence models for desirable statistical properties.

3

Machine Learning Enables Efficient and Effective Affinity Maturation

3.1 PREAMBLE

Thus far, we have considered pre-trained protein generative model and developed tools and paradigms to understand i) what sorts of biological signal the models are learning and ii) how well the models

have learned the distributions they are trained on. These large pre-trained models are likely most useful for modeling protein properties that are shared across a wide array of protein families - i.e signals that the model could learn from large evolutionary datasets. However, for more specific functions, deeper exploration is needed to understand how well a given generative model would perform.

In the following chapter, we present a design campaign for optimizing antibody binding affinity to specific targets for each binder. While there has been some work on deploying pre-trained generative models for optimizing specific binding interactions¹⁰², experimental pipelines for optimizing affinity are well-developed and highly powerful. However, there are still many challenges in the antibody optimization pipeline. In this work, we address some of these challenges. We propose a machine learning framework to use in combination with minimal experimental data to improve the efficiency and downstream hit-rate of affinity maturation campaigns. This work functions as a case-study for how to employ machine learning for design tasks that require bespoke experimental data, and how to understand what elements of a pipeline would most benefit from augmentation with machine learning.

The following work was done in collaboration with Edward P. Harvey, Aaron W. Kollasch, Adama J. Riesselman, James Osei-Owusu, Louis R. Hollingsworth, Irina Shlosman, Biswaranjan Pani, J. Wade Harper, Andrew C. Kruse, and Debora S. Marks. I led the computational work and direction of the project; E.P.H., L.R.H., J.O.O., and I.S. constructed the error prone PCR affinity maturation yeast display libraries; E.P.H., L.R.H., J.O.O., and I.S. performed the MACS and FACS selections; E.P.H. expressed and purified B7 substitution nanobodies; A.C.K., D.S.M., and J.W.H. supervised the project and provided funding.

3.2 ABSTRACT

Antibodies can bind their targets with exquisite potency and selectivity due in part to large antibody-target protein-protein interaction surface areas. Whereas post-immune antibody libraries benefit from *in vivo* affinity maturation, sorting synthetic libraries oftentimes yields low-affinity hits which must be affinity matured prior to further use. Affinity maturation campaigns can be laborious, often requiring multiple selection rounds for each clone to be optimized. Here we deploy machine learning to design optimized binders efficiently and controllably from sparse sequencing data from affinity maturation yeast-display campaigns. We show that machine learning models trained on sequencing data from a single selection round can predict which sequences will be enriched after multiple rounds. We find that linear models outperform deep neural networks and semi-supervised approaches in ranking known affinity-enhancing substitutions. Linear models are also more interpretable, offering insights into residue preferences that can be leveraged for further engineering. We use our approach to optimize nanobody binders to transmembrane protein 192 (TMEM192) and to relaxin family peptide receptor 1 (RXFP1), with the best single substitution improving affinity by over 20-fold.

3.3 INTRODUCTION

The growing need for effective and cheaper antibody therapeutics has necessitated rapid development in technologies for antibody engineering^{294,166}. The antigen-binding properties of an antibody are largely determined by the complementarity determining region (CDR) loops and consequently, these loops are often the focus of engineering efforts to improve antibody binding^{144,143}. A wide variety of antibody formats have been engineered for use as therapeutics and research tools, including single-chain variable fragments (scFvs)¹¹⁴ and monoclonal antibodies (mAbs)¹³⁸. Nanobodies are heavy chain-only VHH domains derived from camelid heavy-chain

only antibodies^{99,184}. Nanobodies are ideal for binding concave epitopes due to lack of steric encumbrance by a paired light chain. Most FDA-approved mAbs were isolated from immunized animal models and then optimized for binding affinity and drug-like properties^{166,159}. In addition to being resource-intensive to acquire, molecules from model animals often first need to be humanized before being used as human drugs. *In vitro* technologies, such as phage or yeast display have allowed for identification of lead molecules from naïve libraries matching human immune repertoires^{26,147,43}; these technologies enabled the development of the first fully human therapeutic mAb, Humira¹³². Using synthetic naïve libraries is especially beneficial when the target antigen shares high homology with a native protein in the animal model, which can limit the prospects for viable binders from being generated by the immune system due to tolerance of self-antigens.

Initial leads from *in vitro* or animal immunization discovery campaigns often need to be optimized for binding affinity in a process called affinity maturation. High throughput display technologies have also been leveraged for affinity maturation, wherein a mutant library of the lead molecule is generated using combinatorial mutagenesis or error-prone PCR (ePCR) and then screened to select for high-affinity mutants²⁹⁴. It is beneficial to generate a large mutant library to maximize chances of finding a high affinity binder; ePCR can produce libraries of 29 mutants from a single lead antibody. However, the scale of these campaigns come with added costs. Larger libraries often require multiple affinity selection rounds to narrow down the pool to a testable number of binders. It is difficult to predict how many affinity selection rounds will be needed and this is variable across antigens and initial leads. Moreover, successive rounds can lead to convergence of the binder pool to a limited number of highly related unique sequences, which can lead to many hits failing subsequent low-throughput evaluation if they are all quite similar to each other in sequence. Recently, more groups have started sequencing the pools after each selection round and using the sequencing counts to nominate binders⁵⁷. This ameliorates the problem of low diversity, as the sequences can be clustered to select for highly distinct hits. However, sequencing abundance is often confounded

by variable expression of the clones and is highly dependent on gating strategies during FACS²⁹⁴. Thus, high sequencing abundance does not correlate strongly with binding affinity as measured by low-throughput binding assays. Methods that reduce the number of selection rounds required for a campaign would greatly improve the efficacy and efficiency of affinity maturation experiments.

A machine learning (ML) method that could predict the affinity landscape of a specific antibody-antigen interaction would have great utility as an optimization tool. The model predictions could be used to select substitutions that would most optimize an initial lead antibody, affording both full control over the diversity of what substitutions are selected and over where they are made. Models trained on a relatively small (104 mutants) dataset of high quality binding affinity data have been shown to be able to design higher affinity antibodies¹⁵², however, acquiring the large number of binding assay measurements to train such models for every new antibody lead candidate is experimentally resource intensive. Instead, as sorting experiments on libraries are significantly higher-throughput than binding-assays, groups have trained models on sequencing data from affinity selection campaigns, as an approximation of binding affinity, and found that optimized designs from these ML models have similar or higher affinity than binders identified experimentally^{156,218}. Notably, both groups trained a model using the enriched sequences from the last FACS round in their campaign to learn the landscape of the highest affinity binders. However, neither demonstrate how useful their methods are when only one or a few affinity selection rounds have been performed and thus both methods necessitate a full campaign before a model can be trained. Pre-trained generative protein models hold much promise as design tools that do not require training on experimental affinity maturation data, and that can optimize a new antibody lead in a zero-shot capacity. Hie et al.¹⁰² showed that large protein language models (PLMs)¹⁵⁵ trained on the protein universe are useful as zero-shot antibody engineering tools. They use the PLM to narrow the search space of single mutants of a lead antibody that are then tested for improved binding affinity and then mutants with high measured affinity are combined manually. However, the PLMs do not consider any infor-

mation about the target antigen, and thus only select mutants that likely do not break the protein's fold. Furthermore, multiple rounds of binding assay experiments are required to identify which substitutions and combinations of substitutions increase affinity. For highly complex binding interactions where epistasis between residues strongly contributes to the affinity landscape, prohibitively large numbers of mutants would need to be tested within this workflow to identify a binder. Pre-trained generative models of structure can consider target antigen information when designing binder sequences^{54,271,246}. However, whether these models are useful for affinity maturation has not been thoroughly investigated. While pre-trained ML models hold promise for optimizing antibodies without data (zero-shot models), a thorough comparison of these methods compared to methods trained with experimental affinity maturation data (supervised models) has not been performed.

Here, we develop a ML framework for designing optimized binders that requires only a single round of FACS affinity sorting. We show that models trained on a single selection round can predict which sequences would be enriched in further rounds retrospectively using completed affinity maturation campaigns. We show that these models identify affinity-enhancing single substitutions high up in their ranking of substitutions predicted to increase affinity. We find that regularized logistic regression models outperform neural network models in identifying single optimizing substitutions; this coupled with the high interpretability of logistic regression models make them powerful tools for optimizing antibodies. We use our framework to optimize 3 nanobody binders and 1 non-nanobody binder and identify single substitutions that confer as high as a 26-fold increase in affinity. Lastly, we perform a systematic comparison of supervised and zero-shot design methods for optimizing a nanobody binder. This investigation sheds insight on how much value experimental affinity maturation data has over zero-shot ML methods.

3.4 RESULTS

3.4.1 AFFINITY MATURATION CAMPAIGNS AND DATA PROCESSING

To benchmark our approach, we used NGS data from a published affinity maturation campaign for nanobody AT110, which binds to the human angiotensin II receptor type 1 (AT1R)²⁷⁹, and two new affinity maturation campaigns for nanobody B7, which binds the human β_2 adrenergic receptor (B2AR), and RX002, which binds to the relaxin family peptide receptor 1 (RXFP1). Each nanobody binder was discovered via *in vitro* yeast-display selection experiments, starting from a diverse, synthetic nanobody library of approximately 5×10^8 unique nanobody clones¹⁷². All nanobodies were optimized via ePCR affinity maturation with library sizes ranging from 3×10^8 to 2×10^9 nanobody variants. We also used data from an affinity maturation campaign of peptidoglycan-synthase activator (LpoB), an activator that binds to bacterial penicillin-binding protein 1b (PBP1b), to demonstrate the use of our method for optimizing non-immunoglobulin binders. In each case, error-prone PCR libraries were generated and cloned into yeast-display vectors. Each library underwent affinity selection via one MACS round and two or three successive FACS rounds (Supplementary Figure C1). Pools were PCR amplified and sequenced after each round. On average we retrieved 242k reads from each sequencing round with the number of unique reads decreasing with further selection rounds (see Supplementary Table C1 for complete sequencing statistics). Although we note that many other studies benefit from much deeper sequencing of their libraries¹⁵⁶, we show here that even with shallower sequencing we are able to extract signal for optimizing substitutions. This predictive ability of sparse sequencing of affinity maturation libraries is noteworthy due to practical considerations including labor and cost savings.

Library amplification during NGS sequencing introduces sequencing errors which result in many sequences with very few counts. We reasoned that many of these low-count sequences constituted noise and would confound the ML models. Thus, for each round, we removed any sequence

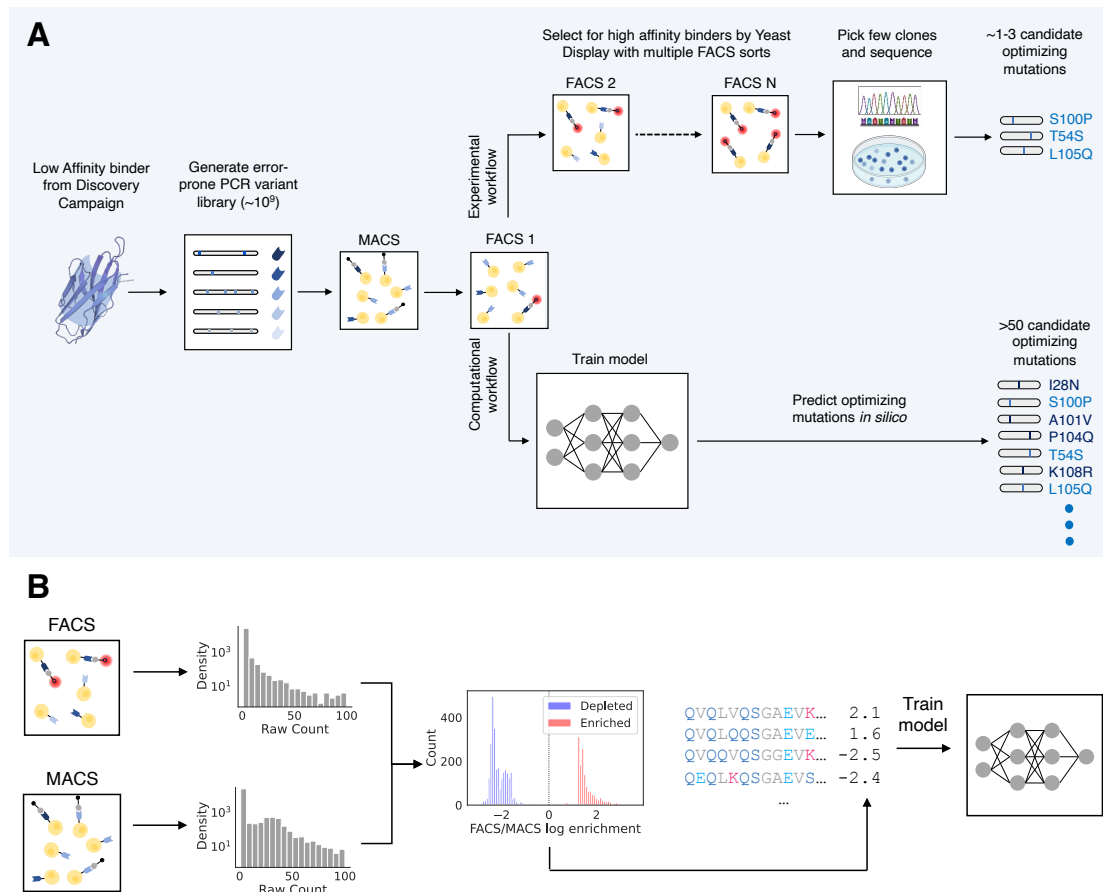


Figure 3.1: Schematic of our approach A) Conventional error-prone PCR affinity maturation proceeds with MACS sorting and multiple FACS sorts to filter the library down to a few number of high affinity mutants. Our approach leverages NGS sequencing from a single round of MACS and FACS, coupled with machine learning models to computationally predict optimizing mutations. The model identifies a much larger set of candidate mutations that are selected for downstream validation. B) Data processing pipeline for generating training data. Sequencing counts from the rounds are filtered and normalized to produce an enrichment ratio that acts as a regression target. Inputs to the models are full VHH sequences represented as one hot encodings.

below a count threshold (set to 5 counts). It has been found that sequencing abundance after enrichment poorly correlates with binding affinity^{294,162}. One cause of this is expression bias from sequences that are highly expressible but have low binding affinity. To account for this, we normalized the abundance after each FACS round by the post-MACS abundance and calculate the resultant log to get a log enrichment ratio (Fig 3.1B). To normalize the abundance, we require read counts of the same sequence in both rounds. However, we find that the overlap of unique reads between sequencing rounds is very low (Supplementary Table C2). To be able to model sequences found in only one round, we impute a floor abundance value (set to 1×10^{-6}) for any sequence below the count threshold of any round and calculate the enrichment ratio with this.

We found that the distributions of log enrichment ratios for the FACS rounds were highly bimodal (Supplementary Fig C2). This arose from the fact that most sequences in the data were only seen in a single sequencing round (Supplementary Table C2). Thus, in addition to the continuous labels of the log enrichment ratios, we also assigned the binary labels “enriched” and “depleted” to any sequence with log enrichment ratio above and below 0, respectively. We aligned each sequence and assigned IMGT numbering to each residue using ANARCI⁶⁵. We only kept IMGT columns that were present in the original nanobody sequence of each campaign. We then one-hot encoded the aligned sequences to produce the inputs to the ML models. Filtering out sequences under the count threshold significantly reduces the data size (Supplementary Table C2). However, we found that the models performed better with the count threshold, and we attribute this to higher data quality post-filtering (Supplementary Table C3).

3.4.2 ML MODELS TRAINED ON ENRICHMENT FROM A SINGLE FACS ROUND CAN PREDICT LATER ROUND ENRICHMENT

To see if ML models trained on early selection rounds could predict the enrichment achieved after many selection rounds, we trained models to predict sequence enrichment after FACS₁, and tested

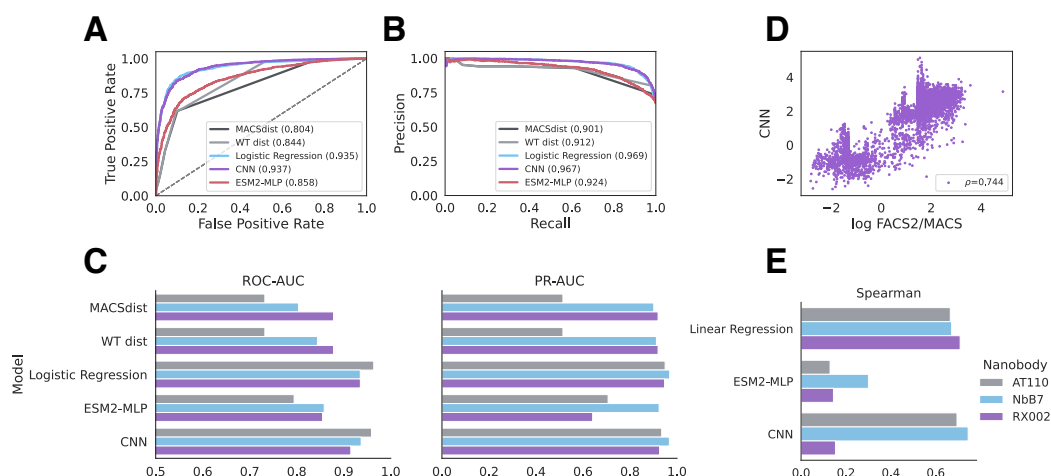


Figure 3.2: Machine Learning models predict second-round enrichment from first-round data. A) ROC-plots and B) Precision-Recall plots for predicting MACS to FACS2 enrichment by models trained on MACS to FACS1 enrichment from the NbB7 dataset. The area under the curve (AUC) for each model for either metric is shown in brackets in the figure legend. C) Bar plots showing the summary performance for the AT110 and NbB7 datasets. ROC-AUC plots are clipped to 0.5. D) Scatterplot of scores from the top regression model (CNN) on the NbB7 dataset. The spearman correlation is shown in the legend. E) Summary bar plots for spearman correlation on both datasets.

their predictive ability of FACS2 enrichment. We trained models both for binary classification of the enrichment label and as regression models on the continuous log enrichment value (Fig 3.2). We trained a variety of models for both regression and binary classification tasks. Linear models with L1 regularization - linear regression and logistic regression for binary classification - have a single weight per amino acid choice for each position in the sequence. Regularization enforces sparsity in the weights such that the non-zero weights in a trained model reflect the most important amino acid substitutions for enrichment/depletion. While linear models are highly interpretable, they cannot model higher-order interactions between residues. Thus, we also trained a convolutional neural network (CNN) which can model nonlinear relationships in the data including residue interactions. Semi-supervised approaches have shown great success in modelling sequence properties by leveraging unsupervised learned representations from a large protein language model (PLM) trained on

protein universe datasets. Thus, we also benchmarked a semi-supervised approach consisting of an MLP top model on top of representations ESM2, a popular transformer-based PLM¹⁵⁵. As the enrichment labels are highly bimodal and mostly reflect whether the sequence was seen before or after enrichment, we wanted to make sure the models were not simply overfitting to which round the sequence came from. Thus, we compared the models to simple distance baselines: i) the Hamming distance from the starting nanobody sequence (WT dist) and ii) the median Hamming distance to all sequences in the MACS round (MACSdist).

The logistic regression and CNN models had similar top performance on both the nanobody B7 and AT110 datasets. Logistic regression is the highest performing binary classification model on the AT110 data, with an area under the receiver operator curve (ROC-AUC) of 0.964 and an area under the precision recall curve (PR-AUC) of 0.949 (Fig 3.2 A-C). For the nanobody B7 data, logistic regression had slightly lower ROC-AUC and slightly higher PR-AUC than the CNN (Table 3.1). The machine learning models all outperform the distance metric baselines, which suggests that the models are learning some affinity property within the sequences rather than which round the sequences came from. Comparable performance of the linear logistic regression model with the CNN suggests that higher-order interactions in the sequence may not be required for prediction on this task. However, as our data sizes are small it may also be there is insufficient data for the neural network models to learn high-order information to improve its prediction.

Of the regression models benchmarked, the CNN has highest Spearman correlation for both AT110 (0.695) and NbB7 (0.744) datasets (Fig 3.2 D-E). However, while the models show high Spearman correlations across the whole dataset, correlation within just the depleted sequences and enriched sequences separately was much worse (-0.052 for depleted sequences and 0.332 for enriched sequences). In fact, the scores from the CNN regression model have an ROC-AUC and PR-AUC of 0.989 and 0.980, respectively, even higher than the CNN trained as a classification model. This suggests that within an enrichment class, the continuous enrichment values may not separate

sequences by affinity with high resolution. This makes sense considering that the FACS gating during affinity selection is stringent and binary, and the abundance of the sequence after or before the gate may not reflect strong differences in binding affinity. This demonstrates the utility of treating prediction of enrichment as a binary classification problem.

Interestingly, for both classification and regression tasks, the semi-supervised approach (ESM2-MLP) had sub-optimal performance, which is logical as the sequences in our datasets are all closely centered around the original nanobody being optimized in the campaign. Thus, evolutionary information from a PLM trained on the protein universe may not be useful for predicting enrichment of these sequences.

Despite the models having high classification performance under ROC-AUC, in comparing scores from a linear regression model against the FACS2/MACS enrichment score, we found that the top binders predicted by the model did not largely overlap with the top binders selected experimental data (Supp Fig C8A). This may be because the top enriched binders are just the most expressible; while we try to account for confounding signal from expressibility of the sequences by normalizing by the MACS sequencing count, due to the very low overlap between rounds (Supp Table C2) most sequence FACS counts are merely normalized by a constant floor value ($1e-6$ in our case). The machine learning models generalize information across all the sequences, and thus may be identifying more reliable affinity information, thus leading them to rank other sequences more highly. Interestingly, the sequences ranked by the model have more substitutions in their CDR₃ (Supp Fig C8B-C), suggesting that the models may be ranking more diverse sequences in their top predictions.

Overall, we find that machine learning models can reliably predict later round enrichment when trained only on a single round of FACS affinity selection, suggesting there is sufficient affinity information in the sequencing data after a single round to predict enriched binders.

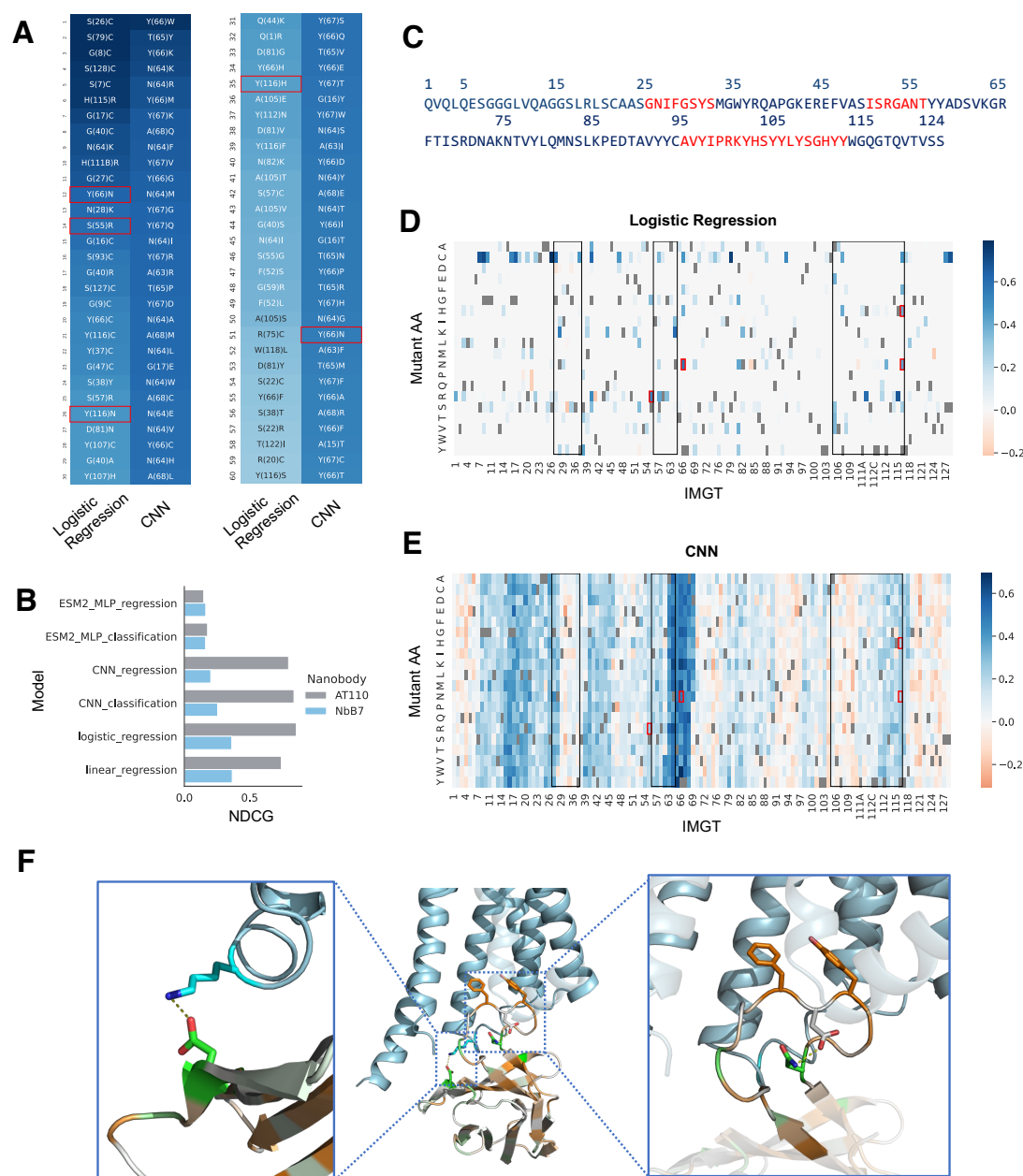


Figure 3.3: Enrichment prediction models identify optimizing single substitutions. A) The top 60 single substitutions for β 2AR binder, NbB7, ranked by model score for the logistic regression and CNN models. Substitutions are colored by model score (darker blue is higher score) and experimentally validated substitutions are boxed in red. B) NDCG ranking performance for machine learning models for the AT110 and NbB7 campaigns. C) Amino acid sequence of NbB7. Indices correspond to residue position in the sequence. CDRs 1-3 are highlighted in red. D-E) Heatmaps showing the gain in score according to the logistic regression model (D) and CNN model (E) for every single substitution of NbB7. Red boxes correspond to the gain/loss in score for experimentally validated single substitutions and black boxes correspond to CDRs 1-3. (F) (middle) the structure of optimized nanobody AT110i1 bound to the intracellular face of AT1R from PDB:6do1. AT1R is colored in blue and the nanobody is colored by the score of the CNN trained on the AT110 FACS1/MACS enrichment with green corresponding to higher scores (predicted gain in affinity) and orange corresponding to lower scores. (left) a close up of a hydrogen bond formed between D58 in the CDR1 and K135 of AT1R. (right) a close up of the interaction between the CDR3 and the lumen of AT1R. A hydrogen bond within CDR3 residues N113 and D108 is shown. Sticks for F107 and Y109 pointing into the lumen are shown. Both hydrogen bonds shown have an inter-atomic distance of 2.9Å.

3.4.3 LINEAR MODELS PERFORM BEST FOR RECOMMENDING SINGLE OPTIMIZING SUBSTITUTIONS

Antibodies are often optimized by identifying single optimizing substitutions that are then combined to produce high-affinity binders. While our methods can be used to design binders multiple substitutions away from the initial lead automatically, it may be beneficial to have optimizing information at the single substitutions level as this can easily be combined with domain knowledge and heuristics for avoiding known undesirable properties that may not be selected against in the sorting experiments. For example, we may want to avoid cysteines to prevent spurious disulfide bonding and tryptophans to avoid non-specific polyreactive binding¹⁰⁰. We also want to avoid the insertion of DG or NG isomerization motifs and NxS/T glycosylation sites. Thus, we evaluated our methods for how well they can identify single optimizing substitutions. For initial *in silico* evaluation, we had access to sets of four validated optimizing single substitutions for NbB7 and AT110 each. We scored all single substitutions for NbB7 and AT110 with our models (Fig 3.3D,E) and evaluated them for how highly they rank the validated substitutions by calculating the normalized discounted cumulative gains (NDCG) of the model's predictions, weighting each validated mutation equally as a hit. Notably, highly ranked substitutions not in our low-N set may still be real optimizing substitutions; the NDCG does not penalize a model for giving high scores to substitutions not in our set, it merely looks for how highly ranked the validated substitutions are. We also calculated the NDCG when only considering the top 100 ranked substitutions (NDCG@100) as we are often only interested in the top predictions from a model.

We found that logistic regression models had the highest NDCG and NDCG@100 on both the NbB7 and AT110 validated sets (Fig 3.3B and Table 3.1). While the logistic regression and CNN models have similar NDCG on the AT110 data, the logistic regression model identifies all four validated substitutions in the predicted top 15 substitutions, whereas the CNN finds all four in the

top 31 substitutions (Supplementary Fig C4A). This suggests the logistic regression model is meaningfully outperforming the CNN in highly ranking the validated substitutions. As this task only considers single substitutions, it stands to reason that a logistic regression model that does not consider inter-residue interactions would have high performance. Our logistic regression models have a single weight for each single mutation away from the initial lead. L1 regularization during training enforces sparsity on these weights and appropriately accounts for correlation between substitutions, meaning that the model identifies the highest confidence optimizing substitutions for a sequence (Fig 3.3D). Contrastingly, a CNN model has much less sparsity in its singles prediction (Fig 3.3E). The model gives high scores for all possible substitutions at certain sites, reflecting that may be learning which sites may be advantageous to mutate without learning the amino acid preferences necessarily. The corollary of this is that if one were to simply select the top singles predictions from the CNN, they would all be clustered around the same sites, leading to low diversity in the selected hits (Supplementary Figure C4B). In this way, the sparsity in a logistic regression model allows for selecting a highly diverse set of hits across the protein, which will increase the chances of the hits passing validation.

3.4.4 MODEL SCORES IDENTIFY STRUCTURALLY IMPORTANT INTERACTING RESIDUES

One of the values of a machine learning model is that it can learn sequence features relating to the specific binding interaction of the nanobody being optimized. To confirm this, we sought to compare the predictions from a model to insights from a structure of a nanobody binding complex. Thus we compared the scores of the CNN trained on the AT110 FACS/MACS enrichment with an available crystal structure for an evolved form of nanobody AT110 (AT110i1) bound to its target, AT1R²⁷⁹. We selected the CNN as it had more signal per site of the nanobody compared to the sparse logistic regression model, enabling better qualitative comparison. We colored each residue in the nanobody structure by the max absolute score per site from the model (Fig 3.3F).

	ROC-AUC	PR-AUC	Spearman	NDCG	NDCG@100
AT110					
Linear/Logistic Regression	0.964	0.949	0.665	0.857	0.857
CNN	0.959	0.935	0.695	0.839	0.839
ESM2-MLP	0.794	0.707	0.129	0.178	0.000
MACSdist	0.732	0.515	-	-	-
WT dist	0.732	0.515	-	-	-
NbB7					
Linear/Logistic Regression	0.935	0.969	0.671	0.363	0.363
CNN	0.937	0.967	0.744	0.255	0.235
ESM2-MLP	0.858	0.924	0.300	0.163	0.000
MACSdist	0.804	0.901	-	-	-
WT dist	0.844	0.912	-	-	-
RX002					
Linear/Logistic Regression	0.935	0.947	0.709	-	-
CNN	0.915	0.926	0.153	-	-
ESM2-MLP	0.855	0.641	0.144	-	-
MACSdist	0.878	0.919	-	-	-
WT dist	0.878	0.919	-	-	-

Table 3.1: Summary performance for enrichment and singles prediction Summary evaluation metrics for the trained machine learning models and sequence distance baselines for the AT110, NbB7 and RX002 campaigns. ROC-AUC, PR-AUC, NDCG and NDCG@100 were evaluated on classification models. MACSdist refers to the median MACS Hamming distance. Spearman was calculated for regression models. NDCG and NDCG@100 are not reported for sequence distance baselines as all single mutants have the same score under these baselines. NDCG values were not calculated for RX002 as we did not have experimentally validated single substitutions *a priori* to validate the RX002 models.

Two of the highest-scoring residues from the models were in sites that were evolved between AT110 and AT110i1 (N58D and Y113N – IMGT no. 66 and 115 respectively); the structure shows the mutated residues. D58 in AT110i1 forms a hydrogen bond with site K135 on AT1R (Fig 3.3F – left), stabilizing the antigen interaction. N113 forms an internal hydrogen bond with D108 (Fig 3.3F – right) which likely stabilizes the conformation of the CDR3 loop. This demonstrates that the model learned AT110 mutations that were important for stabilizing the interaction. The model also assigned low scores to sites 107 and 109 (IMGT no. 112C and 112A respectively), implying that mutating these sites would reduce affinity. In the binding complex, these large hydrophobic residues point into the hydrophobic lumen of AT1R. It is likely that these form hydrophobic interactions further stabilizing the binding complex. This confirms the model prediction that a mutation away from these residues would be deleterious. Notably, the model has high variation in the scores it assigns to CDR3 sites. Rational engineering approaches often begin with mutagenizing all CDR sites as this is where much of the binding interface is for most antibodies^{143,144}. However, as the AT110 example shows, certain sites may be essential for the interaction and need to be held constant. Our modelling approach allows us to identify this granular information so that we focus on sites that would have affinity gains in our engineering. Overall this confirms that our models are learning relevant information relating to the actual biophysical interaction, and demonstrates the utility of our method.

3.4.5 INTERPRETABLE MODELS ALLOW THE SEPARATION OF CONFOUNDING POLYREACTIVITY SIGNAL FROM AFFINITY SIGNAL

For NbB7 we found that many of the top predictions included cysteines which are known to increase the off-target polyreactivity of an antibody (Fig 3.3D)¹⁰⁰. This raised concerns that some of the affinity gains from FACS1 and FACS2 may be coming from an enrichment of polyreactive nanobodies. To confirm this, we performed a third FACS sort that included a polyreactivity

counter-selection along with the affinity selection (referred to as FACS₃). Polyreactivity counter-selection was done using a polyspecificity reagent (PSR) as the fluorescence-tagged antigen. We trained a logistic regression model on the enrichment between FACS₃ and MACS and looked at the weights of the trained model to understand the residue preferences of the FACS₃ enrichment. We found that there were fewer cysteine substitutions in the top FACS₃ model predictions compared to the models trained on FACS₁ and FACS₂ enrichment (Supp Fig C6B-C). This confirmed that some of the enrichment of cysteines in FACS₁ (Fig3D) and FACS₂ (Supp Fig C6A) may have been coming from a population of polyreactive nanobodies in the campaign that were sorted out by the polyreactivity counter-sort. Furthermore, we also found that models trained on the FACS₂ enrichment were less predictive of FACS₃ enrichment than models trained on FACS₁ enrichment by ROC-AUC (Supp Fig C6D). This suggests that there was confounding polyreactivity signal in the selections that was amplified in the library after FACS₂. Models trained on this round may be biased towards polyreactivity signal, thus making them less predictive on FACS₃ which was experimentally depleted for polyreactive binders.

This demonstrates a disadvantage in doing longer affinity maturation campaigns with many FACS selections. If there are confounding binders in a library (eg. highly expressible binders, or very sticky binders), these confounders can be selected for more strongly than on-target, specific binders. Doing many successive sorts risks depleting the library for our on-target binding signal, which would result in many false positives at the end of a campaign. To minimize confounding signal, it may be beneficial to have shorter campaigns which is enabled by our method. Moreover, this use-case exemplifies the utility of having interpretable models. Being able to interpret a model's learned information (eg. by analyzing the weights for each single substitution) allows a practitioner to identify possible confounding signal before hits are validated. Our modelling framework enables one to flexibly combine domain knowledge (eg. about residue biases for polyreactivity) with data-driven insights to effectively optimize a binder. We emphasize here that we were able to identify the

confounding signal even from a model trained on a single, affinity-only FACS round, and the experimental co-affinity sort/polyreactivity counter-sort was only used to validate our hypotheses.

3.4.6 PREDICTED OPTIMIZING SUBSTITUTIONS IMPROVE NANOBODY BINDING AFFINITY

To demonstrate that our method can be used to prospectively identify single optimizing substitutions, we predicted selections using logistic regression models trained on FACS₁ round enrichment. Logistic regression models were selected as they had the highest singles ranking performance across the models (Fig 3.3 and Table 3.1). To demonstrate efficacy of our method on a new nanobody that was not used for computational benchmarking, we trained a logistic regression model on data from a short affinity maturation campaign of nanobody LysoNbV₁ which binds to transmembrane protein 192 (TMEM192). Only a MACS sort and a single FACS sort were performed on a ePCR library seeded by LysoNbV₁. Additionally, activity-modulating binders for RXFP₁ are hypothesized to have great therapeutic potential for treating cardiovascular diseases⁴¹. Thus, we also validated predicted optimizing mutations for RXoo2. For both nanobodies, we scored all possible single substitutions in the CDRs and considered substitutions that were predicted to not increase the nanobody polyreactivity. Test substitutions were selected by going down the ranking from the model scores. If multiple predictions were in the same residue position, the highest scoring substitution was retained and the rest were skipped over. The top nine LysoNbV₁ substitutions were tested and the top RXoo2 substitution was tested (further experimental validation of more RXoo2 substitutions will be included in future versions of this work).

The binding affinities of LysoNbV₁ substitutions were tested using on-yeast display binding assays (see supplementary methods). 2/9 of the tested substitutions had improved binding affinity relative to the WT (Fig 3.4a). Notably, the substitution with the greatest improvement in affinity, S100P, was the top-ranked substitution by the logistic regression model (Fig 3.4b). Remarkably, a double mutant consisting of the 2 optimizing mutations (P98Q and S100P) had reduced binding

affinity. This might arise from the two mutations being so close to each other in the CDR₃. Simultaneous changes to the CDR₃ may alter the binding conformation excessively and thus disrupt the interaction.

The binding affinity of WT RX002 and the top predicted substitution (N27Y) were determined using biolayer interferometry (see supplementary methods). N27Y had markedly improved affinity with respect to the WT (Fig 3.4c), improving the WT affinity from a k_d of $>1\mu\text{M}$ to 47nM , an over 20-fold reduction. The off-rate of the nanobody was also reduced from $8 \times 10^{-3}\text{s}^{-1}$ to $1 \times 10^{-3}\text{s}^{-1}$ by the substitution. Further analysis showed that the N27Y substitution disrupted a NIS glycosylation site in the WT nanobody. We hypothesize that the glycosylated residue may impede interaction with the target, and thus removing it would improve binding.

Overall, these results demonstrate the utility of our methods, and specifically logistic regression models, for identifying single optimizing substitutions.

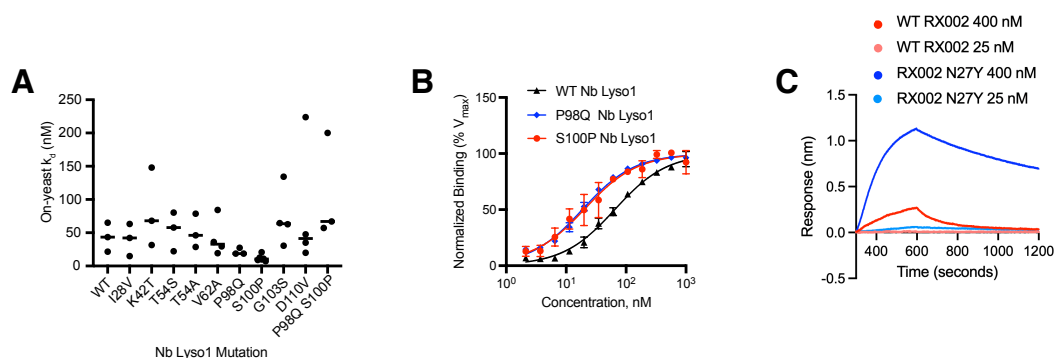


Figure 3.4: Experimental validation of predicted optimizing single substitutions. A) On-yeast k_d values from biological replicates for the top nine selected LysoNbV1 substitutions predicted by the logistic regression model. Bar corresponds to median k_d . B) On-yeast binding curves of the two substitutions that improved LysoNbV1 affinity. Data shown comprise one bioreplicate and are representative of at least $n=3$ independent biological replicates performed on separate days. Error bars represent mean $\pm$ SEM of technical replicates.

3.5 DISCUSSION

The growing demand for antibodies as biologics and molecular tools has necessitated the development of faster and cheaper antibody engineering technologies. Affinity maturation is a key part of the engineering pipeline; however, campaigns are stymied by the need for multiple affinity selection sorts to identify a high quality binder and low hit-rates for binder validation after the campaign. We show that machine learning models trained on a single round of FACS sorting from an affinity maturation campaign can stand in for longer campaigns with multiple FACS sorts. These models are predictive of i) which sequences would have been enriched in a retrospective analysis of completed campaigns (Fig 3.2), and ii) which single optimizing substitutions are found after full campaigns 3.3. Following experimental binding affinity validation, we found that the model's predictions do in fact increase the initial nanobody affinity (Fig 3.4), demonstrating that these models are powerful antibody engineering tools.

While sequencing counts from post-sort sequencing libraries have not shown to be highly predictive of affinity, machine learning models trained on this data are able to generalize across the sequencing information and learn the salient sequence features that enhance affinity. We find that even with a single round, the model has learned sufficient information to identify substitutions that were only found after a full maturation campaign and multiple validation experiments. This affords antibody engineers the ability to run much shorter affinity maturation campaigns and very quickly identify hits from the campaign using computational methods. Furthermore, the highly interpretable weights from L_1 regularized logistic regression models allow engineers to combine prior knowledge about residue preferences with data-driven insights from the model predictions to optimize antibodies flexibly and effectively. For example, we found that our model trained on FACS₁ of the NbB7 affinity maturation campaign overpredicted cysteines at many sites, which are known to increase the polyreactivity of a nanobody (Fig 3.3D, Supp FigC6). We simply ignored these and se-

lected the top non-cysteine mutations from the model which were found to have optimized affinity, while controlling the nanobodies' polyreactivity.

Especially if an engineered antibody is going to be used therapeutically, the polyreactivity of that antibody needs to be minimized to prevent adverse effects from off-target interactions. Affinity and polyreactivity are often correlated (highly sticky antibodies can have high binding affinity) which raises the value in simultaneously optimizing both properties to achieve an optimally specific but high affinity binder. Building computational tools for multi-property optimization is challenging owing to the need for assay data on both properties in the sequence space one wishes to optimize within (eg. for optimizing an initial lead the models would need to be predictive of the sequence space of possible optimizing mutants around the lead). We and other groups¹⁶² enact this by performing joint affinity selection and polyreactivity counterselection experiments on a single library. Towards a more general purpose solution, for cases in which a polyreactivity counter sort was not performed, we leveraged a pretrained polyreactivity model on nanobodies¹⁰⁰ in combination with our trained affinity models. The polyreactivity predictions did not show high correlation with the affinity models for each campaign (Supp Fig C7) which we reasoned stemmed from the difference in training data for the models. The pretrained polyreactivity model was trained on a much more diverse spread of nanobodies from a naïve library¹⁷², while our affinity models are trained on sequences from ePCR which are centered around the initial nanobody in sequence space. The one exception to this was the model for LysoNbV1 which had a spearman correlation of -0.322, significantly higher than the other nanobodies. This suggests that there may have been substantial polyreactivity confounding signal in the LysoNbV1 FACS selections. We opted to use the polyreactivity model as a filter and only selected substitutions that at least retain the initial nanobody specificity for experimental validation. Future work will explore more generalizable models that can be combined across different data sources to construct a single multi-property model to co-optimize antibodies for affinity and specificity.

Finally, while there has been major progress in the field of computational antibody discovery, there is still a way to go before computational methods fully stand in for experimental affinity maturation. Recently, Bennett et al.²¹ developed a generative diffusion model finetuned on nanobody binding complexes which was able to generate target-specific small libraries (100 – 10k sequences) using only a structure of the target and the desired VHH framework, from which on-target binders were identified post some experimental screening. Notably, these binders had comparable affinities to nanobodies identified by experimental discovery campaigns which are insufficient for most therapeutic uses. Thus, affinity maturation experiments are still necessary to optimize leads (whether identified computationally or experimentally). Given that, any efforts to accelerate these experimental pipelines using computational methods are highly needed and will translate to gains for antibody engineers.

Here we show that machine learning models are useful tools in combination with sequencing data from affinity maturation campaigns. Models are able to learn high quality affinity information from even a single FACS round, allowing engineers to run shorter campaigns, extract interpretable insights from their trained models and design optimized nanobodies in a data driven manner.

4

Conclusion

The goal of this thesis is to outline tools, insights and heuristics to help engineers understand how well a generative machine learning model might perform for their design task of choice. We develop different methods for a range of design task complexities and, importantly, for different scenarios in which an engineer may or may not have access to functional measurements to validate their models or the capacity to generate new data for their design task.

In chapter 1, we introduced the largest collection of datasets for benchmarking generative pro-

tein models for mutation effect prediction towards understanding how useful they would be for design. From this wealth of data, we extracted novel insights into functional advantages that different model classes have; specifically, that inverse folding models outperform sequence models for predicting thermostability. These models were trained with structural information from a dataset of proteins that formed defined structures; this suggests that enriching for thermostable proteins during training seems to focus the model's learned distribution to match that of highly stable proteins. This lends credence to our conceptual framework that there are inductive biases from, not only the model architecture, but also the choice of training data that bias a model towards learning certain functions. This work has moved the field toward a more systematic understanding of when and how to use these models. Thus, we are participating in the democratization of the use of these powerful tools; not just putting code in practitioners' hands but also insights into how to use them.

In chapter 2, we developed statistically motivated evaluation metrics to help engineers determine how well a model has learned specific protein distributions. As these metrics are absolute, this tells us how far the performance of our models is from learning the ground truth data distribution, rather than just how much better they are than a baseline. Moreover, we present metrics to assay different aspects of the model's performance: accuracy and reliability. While accuracy tells us in general how well the models are performing, reliability tells us how much we can trust the model's likelihoods and therefore gives us information on the uncertainty of its design. Further work is needed to understand how these novel evaluations compare to more standard metrics (like perplexity) and, more importantly, how the metrics translate to gains in actual design tasks. We develop these tools for engineers to be able to make checks on their model before deploying them in the real world.

In chapter 3, we leverage machine learning to accelerate a specific experimental pipeline, antibody affinity maturation, and we demonstrate how we can use a modelling framework to minimize our data needs and maximize the insight we can get from our data. We show that machine learning models when combined with data from just one selection round produces a greater success rate than

an entire affinity maturation campaign. Moreover, this case study exemplifies how simple models can provide gains on specific tasks where more advanced, general protein models fail. We argue that progress in this field will require flexibility in using a variety of modelling strategies, rather than relying only on large foundational models. More so than any single model, we argue that an engineer requires a thorough understanding of the problem at hand to determine if and where models can usefully be inserted into a design pipeline.

Naturally there are many other outstanding questions relating to evaluating and understanding these models towards their deployment in protein engineering. We posit that the field should take the development of *in silico* evaluation metrics as seriously as it takes the development of new model architectures. Indeed, without proper *in silico* benchmarks to guide our method development we will invariably be solving the wrong problems.

The tools we provide here go a long way in teaching us what different foundation models can and cannot do. We do not deny that foundation models will play a large part in the next stage of computational protein engineering. However, rather than focussing on training bigger models with more and more data types as input, we argue that researchers should consider what kinds of foundation models would have the highest utility for the field.

By understanding what kind of learned information different data types offer during training, we can better design training strategies to build models focussed at particular design tasks. This is crucial as there is no singular hierarchy of design tasks. Instead of having a single model for every problem, we argue that more importantly we require a principled understanding of which models are good for what tasks. Our hope is that these insights will help us move towards being able to design multiple properties simultaneously. This would be a huge boon for developing therapeutics; if we design a protein that fulfills the therapeutic function we need and is an optimized drug-like molecule from the beginning we can save years and millions of dollars in failed therapeutic programs.

Progress in this field will come not just from improvements in the machine learning and modelling, but also technological advancements in our ability to produce large, rich datasets; arguably this is far more important than developing new machine learning architectures. We have seen in chapter 1 how the nature of our training data influences the type of functions we can predict. Thus we should produce training data with our specific functions in mind so that we can train models intentionally. In addition to training data, novel datasets for validation are also in dire need. An evaluation metric is useless without good data to calculate the metric on. Furthermore, a carefully crafted benchmarking dataset will allow us to filter out poorly performing models before even testing their designs. The experiments to produce these datasets should be designed with the specific goal of benchmarking a model in mind. For example, instead of only testing the top end of a model's predictions, we should measure predictions across the score distribution to get a more thorough idea of whether the model's learned distribution actually correlates with the function at hand.

In conclusion, this is a thesis about how to tell if your model is useful for protein engineering and how to use it if it is. We hope these insights will be a boon to engineers and play a part in advancing our ability to leverage computational models for functional protein design.



Appendix to Chapter 1

A.1 APPENDIX TO PROTEINGYM

A.1.1 SOCIAL IMPACT

Protein design holds considerable promise for various fields, ranging from medicine to agriculture, and is likely to have a profound social impact. However, the development of such technology introduces several concerns, particularly relating to the dual use of protein fitness and design models. For

instance, while beneficial for areas like drug design, these models can also be potentially utilized for harmful purposes such as bio-weapon design. Consider a generative model developed for therapeutic purposes: it typically penalizes predicted toxicity. Yet, the logic of this model could be inverted to instead reward for toxicity²⁶². Indeed, any tool or benchmark developed to improve protein design can be manipulated for nefarious objectives. Lastly, protein fitness models will significantly influence the way experiments are conducted. With increased adoption and development of protein design, substantial portions of experimental work can be accelerated, leading to quicker iterations and improved results. Nonetheless, the need for wet lab experimentation remains. These technological advancements will serve to augment, rather than completely supplant, traditional experimental procedures.

Additionally, the American College of Medical Genetics (ACMG) disregards computational prediction of variant effects due to insufficient validation. Consequently, it is essential to create benchmarks using clinical data in order to promote the acceptance of these machine learning methods in medical practice.

A.1.2 LIMITATIONS

DEEP MUTATIONAL SCANS While significant efforts have been dedicated to curating and preprocessing a diverse set of deep mutational scans (DMS), the very nature of these scans imposes biases and limitations to this benchmark:

1. **Measurement noise** Experiments do not have a perfect dynamic range, often imposing a restrictive ceiling and/or floor to the measured response of mutation effects that is not meaningful for protein function and mutation effect prediction. Furthermore, noise is a perennial issue in high-throughput assays, and some assays have poor experimental replicate correlation. Taken together, this means that one cannot expect perfect correlation between exper-

iment and model. Since these considerations affect different proteins to different extents, computing average Spearman correlations across proteins can be misleading.

2. **Bias** There is additional bias in the types of proteins chosen for deep mutational scans. This can be due both to experimental limitations on which proteins' functions can be assayed (for example, disordered proteins are challenging), and to protein prioritisation considerations (for example, viral and cancer-related proteins are over-represented).
3. **Representativeness** No assay is fully representative of the impacts of protein changes on the evolutionary fitness of an organism, which typically involves a convolution of molecular functions across changing environments. In fact, many assays target only a single feature such as expression, binding, or enzymatic activity.
4. **Inconsistent processing** The reported fitness effects from DMS are themselves the result of modeling and analysis of the raw data. The treatment of data is extremely heterogeneous across the community and different analyses can lead to different conclusions on the effect of mutations. For a perfect standardised curation of experimental results one should treat all data with the same approach. We leave this type of analysis for future work.

A.1.3 DATASETS

DMS ASSAYS

EVOLUTION OF PROTEIN FITNESS BENCHMARKS BASED ON DMS ASSAYS As discussed in § 1.1.4, our DMS benchmarks build on several prior works that had compiled a growing library of such assays. We summarize their content in Table A1.

SELECTION AND PROCESSING We focused on several different criteria when determining which DMS assays to include in ProteinGym. These are:

Category	Mut. Type	Metric or Setting	EVmutation	Deep Sequence	ProteinGym v0.1	ProteinGym v1.0
DMS	Sub. Ind.	Assays (mut.) Assays (mut.)	26 (0.1M) 0 (ok)	38 (0.7M) 0 (ok)	87 (1.6M) 7 (270k)	217 (2.4M) 66 (289k)
Clinical	Sub. Ind.	Genes (mut.) Genes (mut.)	0 0	0 0	0 0	2,525 (63k) 1,555 (3k)
Training regime	Sub.	Zero-shot	✓	✓	✓	✓
	Ind.	Zero-shot	-	-	✓	✓
	Sub.	Supervised	-	-	-	✓
	Ind.	Supervised	-	-	-	✓
Baselines	Sub.	Zero-shot	5	3	9	42
	Ind.	Zero-shot	0	0	3	20
	Sub.	Supervised	0	0	0	9
	Ind.	Supervised	0	0	0	3
Metrics	-	Zero-shot	2	3	3	5
	-	Supervised	0	0	0	2

Table A1: Evolution of protein fitness benchmarks ProteinGym v0.1 corresponds to benchmarks in ², while ProteinGym v1.0 corresponds to benchmarks in this paper. The EVmutation benchmark was introduced in ², while the DeepSequence benchmark was developed in ²⁰⁷. Sub., Ind. and mut. are shorthands for substitutions, indels and mutants respectively.

1. The public availability of data
2. The experimental throughput (how many mutations were assayed)
3. The level of noise between experiment replicates
4. The dynamic range of the assay
5. The assay type (selection, enrichment, etc) and whether or not it captures evolutionary constraints.
6. If the assay used amino-acid substitution or indel mutations (no UTR, tRNA, promoter, etc. variants were included).

FINAL LIST OF ASSAYS In-depth metadata about the assays, including the assay type, UniProt ID, MSA start and end positions, mutated positions, and target sequence, is provided under the `reference_files` directory in the codebase. A complete list of included assays is presented at the end of the appendix (See Tables A8 and ??)

PROCESSING OF LARGE THERMOSTABILITY DATASET A large dataset of thermostability assays of 331 natural domains²⁶⁰ contributed 65 assays to our list. We processed these assays as follows:

We used the set of non-redundant natural domains (referred to as Dataset #5 in the original paper). After mapping to UniProt IDs for our DMS id naming convention and removing datasets where none of the tested evolutionary models had a Spearman correlation above 0.2 (suggesting that there is inadequate evolutionary fitness signal in the stability assay, preventing meaningful comparisons between models), we were left with 65 thermostability assays of short domains (40-72 residues long). For substitutions, there was 99+% coverage of each position with 14-19 mutations per position (and 52 of those datasets with multiples), and for indels there was a deletion, Gly and Ala insertion at every position.

CLASSIFICATION OF DMS ASSAYS We grouped the substitution DMS assays into five function types: activity, binding, expression, organismal fitness, and stability, assigning each to a primary class such that the classes are non-overlapping. We provide a brief description of each class in Table A2. We took into account multiple factors to delineate the groups as cleanly as possible, most importantly the type of experiment used (e.g., cell growth, cell sorting, biochemical, stability, etc.). Some assay types presented ambiguities that were resolved as follows:

- **Cell growth** Growth-based assays link the function of their target protein to cellular survival and growth. These assays generally fall under “Organismal fitness”, particularly for complementation assays where mutants are tested for their ability to replace the nat-

ural function of the wild-type target and allow cell growth. However, in some cases, cell growth is artificially linked to phenotypes such as enzymatic activity (e.g., SRC_HUMAN⁷, MET_HUMAN⁷⁴, Q837P4_ENTFA¹⁷³), expression (HXXK4_HUMAN⁸⁷), or binding (RASK_HUMAN²⁷⁷). We recategorized assays that were deemed sufficiently distinct from the protein’s natural function in the cell.

- **Cell sorting** Cell sorting, most commonly fluorescence-activated cell sorting (FACS), artificially selects cells that have been fluorescently labeled according to their function. This class of experiments generally falls into “Activity”, “Binding”, or “Expression”, depending on the choice of labeling target. “Expression” assays label the target protein itself, quantifying its expression levels inside the cell or on the cell surface (e.g., NUD15_HUMAN²⁴⁵, OPSD_HUMAN²⁷⁰). Since protein abundance and cell surface presentation are dependent on protein stability, these assays are often characterized elsewhere as stability assays (e.g., PRKN_HUMAN⁴⁷, PTEN_HUMAN¹⁶⁸). “Activity” assays quantify levels of an enzyme’s product or reporters of enzymatic activity (e.g., OXDA_RHOTO²⁶⁶, AoA247D711_LISMN²³⁸, PPARG_HUMAN²⁶¹). “Binding” assays quantify levels of a binding partner or another reporter of protein binding (e.g., ACE2_HUMAN³⁷, GCN4_YEAST²³⁹).

ACCESS

The following provides more details on the code and data Resources (§ 1.1.8) accompanying this paper.

The open-source codebase containing a framework for scoring all the benchmarks is available via our GitHub repository at: <https://github.com/OATML-Markslab/ProteinGym>. Modifications of certain baselines (e.g. scoring of long sequences beyond the context size in the ESM suite, or pre-processing of MSAs in MSA Transformer) are also released, and all of the model predictions can

Function type	# Assays		Description
	Subs	Indels	
Activity	43	3	Assays that directly or indirectly measure a protein’s catalytic (or otherwise biochemical) activity
Binding	14	0	Assays that measure the affinity or the degree to which a protein binds its target
Expression	17	2	Assays that measure how much the protein is expressed in a cell
Organismal fitness	77	6	Assays that measure how much changes in the protein affect an organism’s growth rate
Stability	66	55	Assays that measure how thermostable a protein is

Table A2: DMS assay function types. The number of substitution and indel assays in each of the 5 function type categories and a general description used to categorize the assays.

be reproduced using this repository. We also include preprocessing code for the clinical data (ClinVar/gnomAD) and DMS assays for reproducibility.

We developed a user-friendly website, <https://www.proteingym.org> containing a leaderboard, detailed results per assay, as well as drill analyses across various dimensions (e.g mutational depth, taxa).

The DMS assays, model scores, Multiple Sequence Alignments, predicted 3D structures, processed ClinVar/gnomAD datasets, and raw files before preprocessing can all be downloaded from our servers (see download instructions on our GitHub repository). Some model checkpoints and other files necessary for scoring (for baselines such as profileHMM, PROVEAN) are also available via our servers, although most model checkpoints such as ESM-1v are available from their respective repositories.

LICENSE

The codebase is open source under the MIT license.

A.1.4 BASELINES

Unless otherwise specified, model scores are calculated by taking the log-ratio of the sequence probabilities between the mutant and wild-type sequences $\log \frac{p(\mathbf{x}_{\text{mut}})}{p(\mathbf{x}_{\text{wt}})}$, following the convention in Hopf et al.¹¹⁰.

ZERO-SHOT BASELINES

ALIGNMENT-BASED MODELS

- **Site-independent Model** We use a site-wise maximum entropy model to infer the contribution of site-specific amino acid constraints without considering explicit epistatic constraints. This model is implemented as referred to in Hopf et al.¹¹⁰.
- **HMM** We use the profile hidden Markov model (HMM) implementation in HMMER⁶⁸. Profile HMMs are frequently used to generate multiple sequence alignments, but also produce log probabilities of sequences that can be used as estimates of fitness for both substitutions and indels⁶⁷.
- **EVMutation** EVMutation¹¹⁰ models pairwise evolutionary couplings between protein sequences using a Potts model (otherwise known as a Markov Random Field).
- **DeepSequence** DeepSequence²⁰⁷ uses a VAE architecture to learn higher-order non-linear evolutionary constraints within each protein family. Mutation effect scores are calculated similarly as EVMutation, as the log-ratio between the mutant and wild-type sequence probabilities $\log \frac{p(\mathbf{x}_{\text{mut}}|\theta)}{p(\mathbf{x}_{\text{wt}}|\theta)}$, but using the VAE evidence lower bound (ELBO) as a proxy for $p(\mathbf{x}|\theta)$.
- **WaveNet** We use a previously published dilated convolutional neural network (dilCNN) based on the WaveNet architecture²²⁷ as an example of a family-specific sequence decoder

capable of handling indels. Due to the expense of training a separate model for each protein, we only evaluate this model against the DMS datasets. Sequence scores are calculated as the difference in (length-normalized) log-likelihoods between the mutant and wild-type sequences.

- **EVE** EVE⁸³ is a Bayesian VAE model architecture for predicting clinical variant effects. The model includes a Gaussian Mixture Model fitted to the background distribution of mutations, in order to provide interpretable protein-specific pathogenicity scores. We use the ClinVar preprocessing pipeline from EVE, and EVE is also used in TranceptEVE¹⁹³.
- **GEMME** GEMME is the Global Epistatic Model for predicting Mutational Effects. It infers the conservation of combinations of mutations across the entire sequence according to an evolutionary tree⁷² and combines it with site-wise frequencies to calculate a combined epistatic sequence score for mutations¹⁴⁶. GEMME intakes multiple sequence alignments of protein families as well as specific mutations to generate scores. To obtain scores, we used the GEMME web-tool hosted at <http://www.lcqb.upmc.fr/GEMME/submit.html> with default parameters.

PROTEIN LANGUAGE MODELS Protein language models are so called because they all use variants of the Transformer²⁶⁷ architecture popularised in natural language processing.

- **UniRep** UniRep⁸ trains a Long Short-Term Memory (LSTM) model on UniRef50²⁴⁹ sequences. It learns how to internally represent proteins by being trained on next amino acid prediction through minimizing cross-entropy loss. While the core model is trained on unaligned sequences, UniRep can also be fine-tuned on sets of homologous sequences from a given family, retrieved with a MSA. This process is called ‘evotuning’ and typically leads to stronger fitness prediction performance.

- **ESM** ESM-1b²⁰⁸ and ESM-1v¹⁷⁴ are protein language models with a Transformer encoder architecture similar to BERT⁵⁸ and trained with a Masked-Language Modeling (MLM) objective on UniRef50 and UniRef90 respectively. We extend the original ESM codebase for these two models to handle sequences that are longer than the model context window (ie., 1023 amino acids), with the approach described in Brandes et al.²⁹ for ESM-1b and in Notin et al.¹⁹⁰ for ESM-1v. We predict fitness for ESM models with the masked-marginal approach introduced in Meier et al.¹⁷⁴, which provides optimal performance on substitutions, but does not support indels.
- **CARP** CARP²⁸⁸ is a protein language model trained with a MLM objective on Uniref50. The architecture leverages convolutions instead of self-attention, leading to computational speedups while maintaining high downstream task performance.
- **RITA** RITA¹⁰¹ is an autoregressive language model akin to GPT2²⁰³, trained on UniRef100²⁴⁹. Four model sizes are available, ranging from 85 million to 1.2 billion parameters. RITA takes unaligned sequences as input, and can score both substitution and indel mutations.
- **ProGen2** ProGen2¹⁸⁸ is an autoregressive protein language model trained on a mixture of UniRef90²⁴⁹ and BFD30²⁴². It follows the standard transformer decoder architecture, and five models of different sizes are available, ranging from 151 million to 6.4 billion parameters. ProGen2 takes unaligned sequences as input, and can score both indel and substitution mutations.
- **VESPA** VESPA¹⁶⁵ is as Single Amino acid Variant (SAV) effect predictor based on a combination the embeddings from the protein language model ProtT5⁷¹, as well as per-residue conservation predictions.

INVERSE FOLDING MODELS Inverse folding models learn the conditional distribution of sequences that are likely to fold to an input protein structure¹¹⁶. Given that there may not be experimentally solved structures for the target sequence of all DMS assays in ProteinGym, we generate input structures using Alphafold2 (AF2)¹²⁹. The inverse folding model in combination with AF2 encompasses an end-to-end scoring pipeline that only requires a protein sequence to score variants. As the sequence representation size is defined by the size of the input structure, the models we benchmark here can only score substitutions.

- **ProteinMPNN** ProteinMPNN⁵⁴ takes in a protein backbone structure and featurizes it as a graph where backbone (N,C,C α) atoms are nodes and edges are determined via euclidian distance cut-offs. The model uses a message passing neural network (MPNN)¹¹⁶ to encode the structure into a latent graph representation. The model then decodes the representation and samples sequences autoregressively.
- **MIF** The Masked Inverse Folding (MIF) and Masked Inverse Folding with Sequence Transfer (MIF-ST) models²⁹⁰ are structured-conditioned protein language models trained with a MLM objective. MIF is trained on CATH4.2⁵⁵, and MIF-ST further augments the MIF model with embeddings from CARP (640M).
- **ESM-IF1** ESM-IF1¹¹² functions similarly to ProteinMPNN but leverages a Geometric Vector Perceptron¹²⁴ (an equivariant message passing module ideal for coordinate data) as the architecture for the structure encoder and sequence decoder.

HYBRID MODELS

- **MSA Transformer** The MSA Transformer²⁰⁵ learns a representation of Multiple Sequence Alignments (MSAs) by training an Axial transformer-based transformer¹⁰⁴ with a MLM objective across a diverse set of 26 million MSAs.

- **Tranception** Tranception¹⁹⁰ combines an autoregressive protein language model with inference-time retrieval from a MSA. We evaluate Tranception Small (85M), Tranception Medium (300M parameters) and Tranception Large (700M parameters) both with and without MSA retrieval. Tranception can score both indel and substitution mutations.
- **TranceptEVE** TranceptEVE augments Tranception with priors for the amino acid distribution at each position based on an ensemble of EVE models for the protein family of interest. The final output log probability is thus a weighted sum of that EVE log prior, the log probability from the autoregressive transformer model in Tranception, as well as site-specific log probabilities obtained from a retrieved MSA (as in the inference-time retrieval procedure described in Tranception). TranceptEVE can score both indels and substitutions.

A.1.5 DETAILED PERFORMANCE RESULTS

DMS SUBSTITUTION BENCHMARKS

ZERO-SHOT Table A3 shows the results for our zero-shot DMS substitutions benchmark. We report Spearman’s rank correlations and bootstrapped standard error estimates for forty baseline models. Table ?? breaks down our substitution DMS by MSA depth, Table A5 by function type, Table ?? by taxa, and Table ?? by mutational depth. To compute the final Spearman’s rank correlation reported in Table A3, we first average all the assays for a particular function type together, resulting in five average values (one each for Activity, Binding, Expression, Organismal Fitness, and Stability). The average of these five numbers is the final reported value.

Table A3: ProteinGym - Zero-shot substitution DMS benchmark Ranking based on corrected average of Spearman's rank correlation between experimental assay measurement and model prediction. The standard error reported corresponds to the non-parametric bootstrap standard error of the difference between the Spearman performance of a given model and that of the best overall model (i.e., TranceptEVE), computed over 10k bootstrap samples from the set of proteins in the ProteinGym DMS substitution benchmark.

Ranking	Model	Type	Spearman	Std. error
1*	TranceptEVE L	Hybrid model	0.456	0.000
1*	TranceptEVE M	Hybrid model	0.455	0.004
1*	GEMME	Alignment-based model	0.455	0.007
4	TranceptEVE S	Hybrid model	0.452	0.004
5	EVE (ensemble)	Alignment-based model	0.439	0.006
6	VESPA	Protein language model	0.436	0.006
7*	Tranception L	Hybrid model	0.434	0.004
7*	MSA Transformer (ensemble)	Hybrid model	0.434	0.009
9	EVE (single)	Alignment-based model	0.433	0.005
10	Tranception M	Hybrid model	0.427	0.005
11	ESM-IF1	Inverse folding model	0.422	0.011
12	MSA Transformer (single)	Hybrid model	0.421	0.009
13	DeepSequence (ensemble)	Alignment-based model	0.419	0.008
14	Tranception S	Hybrid model	0.418	0.006
15	ESM2 (650M)	Protein language model	0.414	0.012
16*	DeepSequence (single)	Alignment-based model	0.407	0.008
16*	ESM-1v (ensemble)	Protein language model	0.407	0.012
18	ESM2 (3B)	Protein language model	0.406	0.011

Table A3: (continued)

Ranking	Model	Type	Spearman	Std. error
19*	MIF-ST	Inverse folding model	0.401	0.010
19*	ESM2 (1.5B)	Protein language model	0.401	0.010
21	EVmutation	Alignment-based model	0.395	0.006
22*	ESM-1b	Protein language model	0.394	0.010
22*	VESPAI	Protein language model	0.394	0.007
24	ProGen2 XL	Protein language model	0.391	0.008
25	ESM2 (1.50M)	Protein language model	0.387	0.013
26	MIF	Inverse folding model	0.382	0.011
27	ProGen2 L	Protein language model	0.380	0.008
28	ProGen2 M	Protein language model	0.379	0.008
29	ProGen2 Base	Protein language model	0.378	0.009
30*	Tranception L no retrieval	Protein language model	0.374	0.008
30*	ESM-1v (single)	Protein language model	0.374	0.013
32	WaveNet	Alignment-based model	0.373	0.012
33	RITA XL	Protein language model	0.372	0.009
34	CARP (640M)	Protein language model	0.368	0.011
35	RITA L	Protein language model	0.365	0.009
36	Site-Independent	Alignment-based model	0.359	0.010
37	RITA M	Protein language model	0.350	0.010
38	Tranception M no retrieval	Protein language model	0.348	0.009
39	Unirep evotuned	Hybrid model	0.347	0.009

Table A3: (continued)

Ranking	Model	Type	Spearman	Std. error
40	ProGen2 S	Protein language model	0.336	0.011
41	CARP (76M)	Protein language model	0.328	0.012
42	ESM2 (35M)	Protein language model	0.321	0.015
43	RITA S	Protein language model	0.304	0.011
44	Tranception S no retrieval	Protein language model	0.303	0.012
45	CARP (38M)	Protein language model	0.279	0.014
46	ProteinMPNN	Inverse folding model	0.258	0.011
47	ESM2 (8M)	Protein language model	0.226	0.015
48	UniRep	Protein language model	0.190	0.016
49	ProtGPT2	Protein language model	0.188	0.011
50	CARP (600K)	Protein language model	0.106	0.016

CLUSTERING ZERO-SHOT SUBSTITUTION MODELS We clustered the zero-shot models using hierarchical clustering on the vector of NDCG metrics for each dataset in the DMS substitutions (Fig. A1). We find that models with the same architecture tend to cluster together (e.g., RITA models), however, there are exceptions (e.g., ESM-2 models). We also observe that the alignment-based models tend to cluster together, suggesting that training on the same MSA may promote similar scoring behavior.

Model type	Model name	Spearman by Function Type (↑)					
		Activity	Binding	Expression	Organismal Fitness	Stability	All
Alignment-based	Site-Independent	0.369	0.345	0.351	0.382	0.358	0.361
	WaveNet	0.379	0.325	0.350	0.365	0.449	0.374
	EVmutation	0.440	0.322	0.382	0.411	0.430	0.397
	DeepSequence (ens.)	0.455	0.368	0.396	0.413	0.476	0.422
	EVE (ens.)	0.464	0.394	0.406	0.447	0.491	0.440
	GEMME	0.482	0.387	0.443	0.452	0.519	0.457
Protein language	UniRep	0.182	0.203	0.230	0.141	0.210	0.193
	CARP (640M)	0.395	0.274	0.419	0.364	0.414	0.373
	ESM-1b	0.428	0.289	0.427	0.351	0.500	0.399
	ESM-2 (15B)	0.405	0.318	0.425	0.388	0.488	0.405
	RITA XL	0.366	0.303	0.416	0.381	0.398	0.373
	ESM-1v (ens.)	0.414	0.320	0.456	0.387	0.500	0.415
	ProGen2 XL	0.402	0.302	0.423	0.387	0.445	0.392
	VESPA	0.468	0.365	0.410	0.440	0.500	0.437
Hybrid	UniRep evotuned	0.355	0.304	0.366	0.346	0.366	0.347
	MSA Transformer (ens.)	0.469	0.343	0.439	0.421	0.495	0.433
	Tranception L	0.465	0.351	0.455	0.436	0.471	0.436
	TranceptEVE L	0.487	0.381	0.456	0.460	0.500	0.457
Inverse Folding	ESM-IF1	0.368	0.392	0.403	0.324	0.624	0.422
	MIF-ST	0.390	0.323	0.432	0.373	0.486	0.401
	ProteinMPNN	0.197	0.165	0.198	0.165	0.566	0.258

Table A5: ProteinGym - Zero-shot substitution DMS benchmark by function type Corrected average of Spearman's rank correlation between model scores and experimental measurements on the ProteinGym substitution benchmark, separated into five functional categories (Activity, Binding, Organismal Fitness, Stability and Expression). 'All' is the average of all the categories.

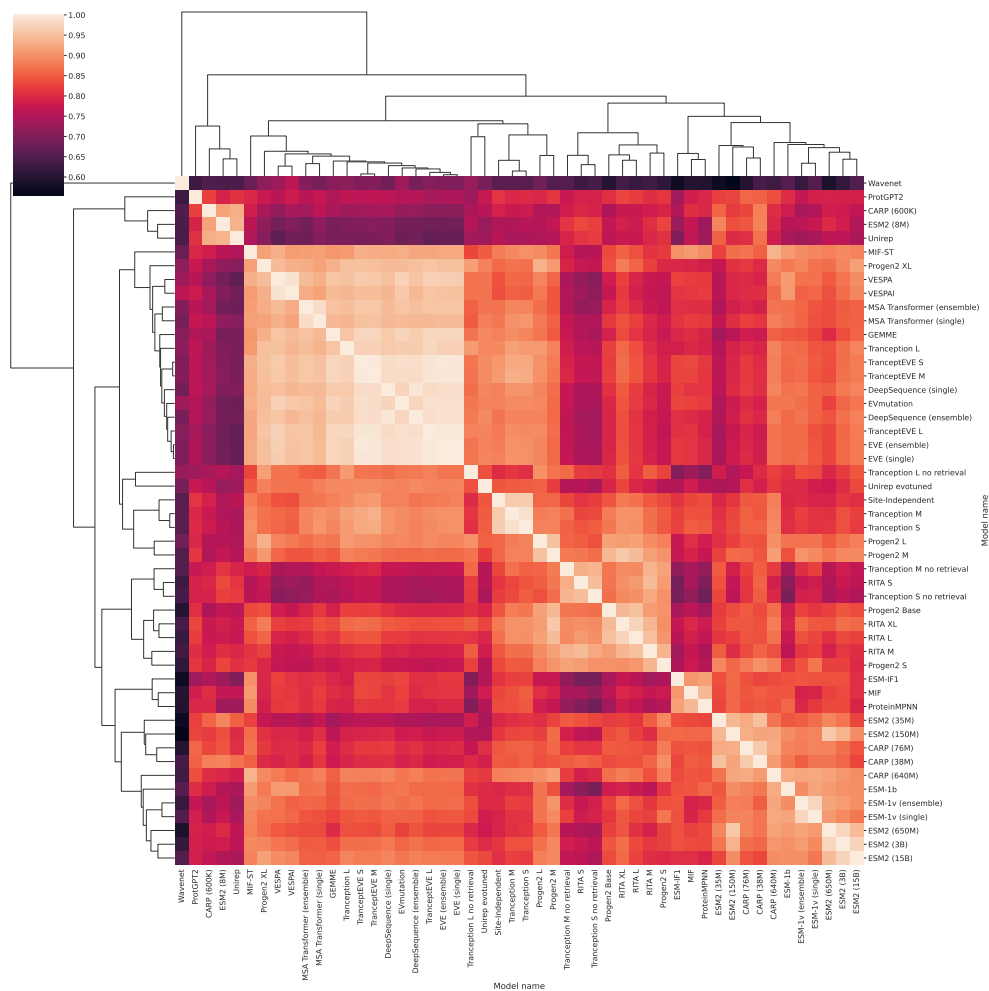


Figure A1: Hierarchical clustering of zero-shot models by NDCG performance Heatmap colored by the Pearson correlation of the NDCG@10% values for each DMS assay for each pair of zero-shot models. Lighter color corresponds to higher correlation. The ordering and dendrogram were produced by hierarchical clustering of the correlation values.

Table A8: List of substitution datasets See the reference file in the GitHub repo for other info (UniProt ID, taxon, DOI, and more assay details).

Dataset	Reference
β -Lactamase	Jacquier et al. ¹¹⁸
β -Lactamase	Stiffler et al. ²⁴³

Table A8: (continued)

Dataset	Reference
β -Lactamase	Firnberg et al. ⁷⁹
β -Lactamase	Deng et al. ⁵⁶
β -Lactamase VIM-2	Chen et al. ³⁹
β -Glucosidase	Romero et al. ²¹¹
AAV	Sinai et al. ²³¹
ACE ₂	Chan et al. ³⁷
ADRB ₂	Jones et al. ¹²⁸
APH(3')II, neo	Melnikov et al. ¹⁷⁷
APP	Seuma et al. ²²⁵
Activation-induced deaminase	Gajula et al. ⁸⁴
Aliphatic amidase	Wrenbeck et al. ²⁸⁰
Alpha-synuclein	Newberry et al. ¹⁸⁵
Amyloid β	Gray et al. ⁹⁶
Amyloid β	Seuma et al. ²²⁶
Ancestral spleen tyrosine kinase	Hobbs et al. ¹⁰⁵
Anti-CRISPR protein AcrIIA ₄	Stadelmann et al. ²³⁸
Antitoxin ParD ₃	Ding et al. ⁶⁰
Antitoxin ParD ₃	Aakre et al. ¹
Arrestin-1	Ostermaier et al. ¹⁹⁷
BRCA ₁	Findlay et al. ⁷⁸
BRCA ₂	Erwood et al. ⁷³
CALM ₁	Weile et al. ²⁷⁵

Table A8: (continued)

Dataset	Reference
CARD ₁₁	Meitlis et al. ¹⁷⁵
CASP ₃	Roychowdhury & Romero ²¹⁴
CASP ₇	Roychowdhury & Romero ²¹⁴
CBS (cystathionine beta-synthase)	Sun et al. ²⁴⁷
CCR ₅	Gill et al. ⁹⁰
CD ₁₉	Klesmith et al. ¹³⁷
CVB ₃ capsid	Mattenberger et al. ¹⁶⁹
CXCR ₄	Gill et al. ⁹⁰
Chalcone synthase	Wrenbeck et al. ²⁸¹
Cytochrome P ₄₅₀ 2C ₉	Amorosi et al. ¹²
Cytochrome P ₄₅₀ 2C ₉	Amorosi et al. ¹²
D-amino acid oxidase	Vanella et al. ²⁶⁶
DHFR reductase	Nguyen et al. ¹⁸⁶
DHFR reductase	Thompson et al. ²⁵³
DNA methylase HaeIII	Rockah-Shmuel et al. ²⁰⁹
Dengue virus NS ₅	Suphatrakul et al. ²⁴⁸
Dlg4, (PSD95_PDZ ₃)	McLaughlin et al. ¹⁷¹
EfrC	Meier et al. ¹⁷³
EfrD	Meier et al. ¹⁷³
EnvZ	Ghose et al. ⁸⁸
ErbB2 membrane domain	Elazar et al. ⁷⁰
EstA	Nutschel et al. ¹⁹⁵

Table A8: (continued)

Dataset	Reference
GAL ₄	Kitzman et al. ¹³⁵
GB _I	Wu et al. ²⁸²
GB _I	Olson et al. ¹⁹⁶
GDI _I	Silverstein et al. ²³⁰
GFP	Sarkisyan et al. ²¹⁹
GMR (aacC _I)	Dandage et al. ⁵²
GRB2-SH ₃	Faure et al. ⁷⁵
Gcn4	Staller et al. ²³⁹
Glucokinase regulatory protein	Gersing et al. ⁸⁷
Glucokinase regulatory protein	Gersing et al. ⁸⁶
Glycophorin A membrane domain	Elazar et al. ⁷⁰
Green fluorescent protein amacGFP	Gonzalez Somermeyer et al. ⁹³
Green fluorescent protein cgreGFP	Gonzalez Somermeyer et al. ⁹³
Green fluorescent protein ppluGFP ₂	Gonzalez Somermeyer et al. ⁹³
HIV env	Duenas-Decamp et al. ⁶⁴
HIV env	Haddox et al. ⁹⁸
HIV env (BF ₅₂₀)	Haddox et al. ⁹⁸
HIV env (BG ₅₀₅)	Haddox et al. ⁹⁸
HIV rev	Fernandes et al. ⁷⁶
HIV tat	Fernandes et al. ⁷⁶
HMG-CoA reductase	Jiang ¹²²
HRAS	Bandaru et al. ¹⁸

Table A8: (continued)

Dataset	Reference
HSP82	Flynn et al. ⁸⁰
HSP82	Mishra et al. ¹⁸¹
Hsp90	Hietpas et al. ¹⁰³
Hydroxymethylbilane synthase	van Loggerenberg et al. ²⁶⁵
IGP dehydratase (HIS ₃)	Pokusaeva et al. ²⁰⁰
InfA	Kelsic et al. ¹³¹
Influenza H ₃ N ₂ neuraminidase	Lei et al. ¹⁵⁰
Influenza M ₁ matrix protein	Hom et al. ¹⁰⁷
Influenza RNA polymerase PB ₁	Li et al. ¹⁵⁴
Influenza hemagglutinin	Thyagarajan & Bloom ²⁵⁴
Influenza hemagglutinin	Doud & Bloom ⁶²
Influenza hemagglutinin	Wu et al. ²⁸⁴
Influenza hemagglutinin	Lee et al. ¹⁴⁹
Influenza neuraminidase	Jiang et al. ¹²¹
Influenza nucleoprotein	Bloom ²⁵
Influenza nucleoprotein	Doud et al. ⁶³
Influenza nucleoprotein	Doud & Bloom ⁶²
Influenza polymerase acidic protein	Wu et al. ²⁸³
Influenza polymerase basic protein 2	Soh et al. ²³²
KCNE ₁	Muhammad et al. ¹⁸³
KCNH ₂	Kozek et al. ¹⁴⁰
KCNJ ₂	Coyote-Maestas et al. ⁴⁸

Table A8: (continued)

Dataset	Reference
KRAS	Weng et al. ²⁷⁷
KRAS	Ursu et al. ²⁶³
L-selectin	Elazar et al. ⁷⁰
LGK (levoglucosan kinase)	Wrenbeck et al. ²⁸¹
LGK (levoglucosan kinase)	Klesmith et al. ¹³⁶
LamB	Andrews & Fields ¹³
Leucine-rich repeat protein SHOC-2	Kwon et al. ¹⁴⁵
MAPK _I	Brenan et al. ³⁰
MET kinase	Estevam et al. ⁷⁴
MPL	Bridgford et al. ³¹
MSH ₂	Jia et al. ¹²⁰
MTHFR reductase	Weile et al. ²⁷⁴
MlaC	MacRae et al. ¹⁶⁰
NPC intracellular cholesterol transporter	Erwood et al. ⁷³
NPC intracellular cholesterol transporter	Erwood et al. ⁷³
NS ₅ A	Qi et al. ²⁰²
NUDT _{I5}	Suiter et al. ²⁴⁵
OCT _I (SLC22A _I)	Yee et al. ²⁹¹
Ornithine transcarbamylase	Lo et al. ¹⁵⁸
p53	Giacomelli et al. ⁸⁹
p53	Kotler et al. ¹³⁹
PAB _I	Melamed et al. ¹⁷⁶

Table A8: (continued)

Dataset	Reference
PPARG	UK Monogenic Diabetes Consortium et al. ²⁶¹
PSD95-PDZ3	Faure et al. ⁷⁵
PTEN	Matreyek et al. ¹⁶⁸
PTEN	Mighell et al. ¹⁷⁹
Parkin	Clausen et al. ⁴⁷
Phosphoserine aminotransferase	Xie et al. ²⁸⁶
Phototropin	Chen et al. ⁴²
Pilin (PilE)	Kennouche et al. ¹³³
Plasminogen activator inhibitor-1	Huttinger et al. ¹¹⁵
Protein phosphatase 1D	Miller et al. ¹⁸⁰
RAF oncogene	Zinkus-Boltz et al. ²⁹⁸
RNAse III (rnc)	Weeks & Ostermeier ²⁷²
Rhodopsin	Wan et al. ²⁷⁰
SARS-CoV-2 Mpro	Flynn et al. ⁸¹
SARS-CoV-2 Spike RBD	Tan et al. ²⁵²
SARS-CoV2 Spike RBD	Starr et al. ²⁴¹
SCN5A	Glazer et al. ⁹²
SOX17	Veerapandian et al. ²⁶⁸
SOX2	Veerapandian et al. ²⁶⁸
SRC	Ahler et al. ⁷
SUMO-conjugating enzyme UBC9	Weile et al. ²⁷⁵
Small ubiquitin-related modifier 1	Weile et al. ²⁷⁵

Table A8: (continued)

Dataset	Reference
Sodium-dependent serotonin transporter	Young et al. ²⁹³
Src	Chakraborty et al. ³⁶
Src	Nguyen et al. ¹⁸⁷
Streptococcus pyogenes Cas9	Spencer & Zhang ²³⁶
TARDBP	Bolognesi et al. ²⁷
TIM Barrel (S. solfataricus)	Chan et al. ³⁸
TIM Barrel (T. maritima)	Chan et al. ³⁸
TIM Barrel (T. thermophilus)	Chan et al. ³⁸
Thiamin pyrophosphokinase 1	Weile et al. ²⁷⁵
Thiopurine S-methyltransferase (TPMT)	Matreyek et al. ¹⁶⁷
Toxin CcdB	Tripathi et al. ²⁵⁸
Toxin CcdB	Adkar et al. ⁴
Tsuboyama multi-DMS	Tsuboyama et al. ²⁶⁰
Ube4b	Starita et al. ²⁴⁰
Ubiquitin	Roscoe et al. ²¹³
Ubiquitin	Roscoe & Bolon ²¹²
Ubiquitin	Mavor et al. ¹⁷⁰
VKORC1	Chiasson et al. ⁴⁴
VKORC1	Chiasson et al. ⁴⁴
YAP1	Araya et al. ¹⁵
Zika virus env	Sourisseau et al. ²³⁵

A.2 APPENDIX TO STUCTSEQ

Function type	ESM-IF1	ProteinMPNN
Activity	9	5
Binding	2	0
Expression	4	2
Organismal fitness	12	2
Stability	55	32
Unannotated	1	1
Total	83	42

Table A9: Top inverse folding model statistic The number of assays for which each inverse folding model had higher Spearman than all 6 best-in-class sequence models analyzed here, grouped by function type. The performances of ESM-IF1 and ProteinMPNN were not compared to each other here.

Model name	Activity	Binding	Expression	OrganismalFitness	Stability
EVE (ens.)	2	1	0	1	0
GEMME	4	1	1	7	3
ProGen2 (ens.)	3	0	0	9	0
VESPA	5	4	1	18	1
ProteinMPNN	0	0	0	0	2
ESM-IF1	9	2	4	12	53
Tranception L	3	1	0	7	4
TranceptEVE	5	1	0	15	5

Table A10: Top model statistics Each row shows the number of assays for which that model was the top performer by Spearman, grouped by assay function type. Here all 8 models were compared against each other.

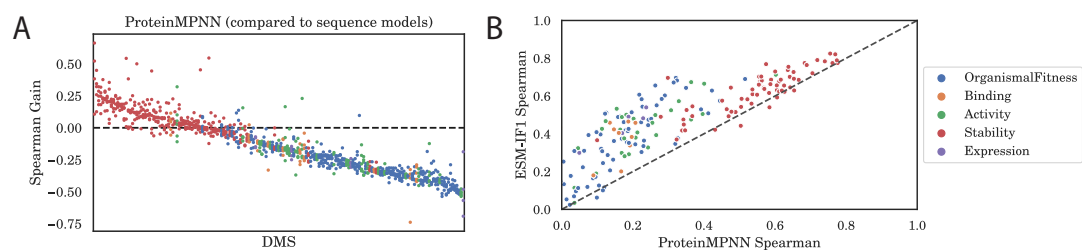


Figure A2: Spearman comparisons for ProteinMPNN (A) The gain in Spearman between ProteinMPNN and the top two sequence models of each sequence model type, calculated for each DMS. We plot the DMSes in order of the median Spearman Gain. (A) The ProteinMPNN Spearman compared to the ESM-IF1 Spearman on the DMS assays. The dotted line corresponds to the identity line. The point color for both panels corresponds to the type of function assayed in the DMS.

B

Appendix to Chapter 2

B.1 PROOF OF LEMMA 2.5.2

Let us first re-state the lemma in its complete form.

Lemma (Complete form of Lemma 2.5.2). *Assume that $\mathcal{X}$ is locally-compact and second countable.*

Moreover, assume that $k_{\mathcal{X} \times \mathcal{Y}} = k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$, and that $k_{\mathcal{X}}, k_{\mathcal{Y}}$ satisfy the integrability conditions

$\mathbb{E}[k_{\mathcal{X}}(X, X)k_{\mathcal{Y}}(Y, Y)] < +\infty$ and $\mathbb{E}[k_{\mathcal{Y}}(Y, Y)] < +\infty$ (and similarly for $\tilde{Y}$). Then,

$$\text{ACMMD}^2(\mathbb{P}_|, Q_|) = \left\| T_{K_{\mathcal{X}}}(\mu_{\mathbb{P}_|} - \mu_{Q_|}) \right\|_{\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}}}^2$$

Where $\mu_{\mathbb{P}_|}$ and $\mu_{Q_|}$ are the conditional mean embeddings¹⁹⁸ of $\mathbb{P}_|$ and $Q_|$, given by: $\mu_{\mathbb{P}_|} : x \mapsto \mathbb{E}_{y \sim \mathbb{P}_{|x}} k_{\mathcal{Y}}(y, \cdot)$ (and similarly for $Q_|$), $K_{\mathcal{X}}(x, x') := k_{\mathcal{X}}(x, x')I_{\mathcal{H}_{\mathcal{Y}}}$ is an operator-valued kernel with associated vector-valued RKHS $\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}} \subset L_{\mathbb{P}_X}^2(\mathcal{X}, \mathcal{H}_{\mathcal{Y}})$, and $T_{K_{\mathcal{X}}}$ is its associated integral operator from $L_{\mathbb{P}_X}^2(\mathcal{X}, \mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}})$ to $\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}}$, defined as

$$T_{K_{\mathcal{X}}}f(x) = \int_{\mathcal{X}} K_{\mathcal{X}}(x, x')f(x')\mathbb{P}_X(dx') \in \mathcal{H}_{\mathcal{Y}}$$

for all $f \in L_{\mathbb{P}_X}^2(\mathcal{X}, \mathcal{H}_{\mathcal{Y}})$ and $x \in \mathcal{X}$. Moreover, if $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are C_0 -universal^{*}

$$\text{ACMMD}(\mathbb{P}_|, Q_|) = 0 \iff \mu_{\mathbb{P}_{|x}} = \mu_{Q_{|x}}, \quad \mathbb{P}_X\text{-a.e.}$$

Proof. Let us introduce the notations used in this proof. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be the sample space, $X(\omega)$, $Y(\omega)$, $\tilde{Y}(\omega)$ being random variables on Ω corresponding to the input, target and the model. When clear, we will identify the measure $\mathbb{P}$ and the push-forwards $Y_{\#}\mathbb{P}$, $\tilde{Y}_{\#}\mathbb{P}$ and drop the dependence of Y , $\tilde{Y}$ on ω . Given $x \in \mathcal{X}$, we write K_x the linear operator from $\mathcal{H}_{\mathcal{Y}}$ to $\mathcal{L}(\mathcal{X}, \mathcal{H}_{\mathcal{Y}})$, the space of linear operators from $\mathcal{X}$ to $\mathcal{H}_{\mathcal{Y}}$, such that $(K_x f)(x') = K_{\mathcal{X}}(x, x')f \in \mathcal{H}_{\mathcal{Y}}$ for all $f \in \mathcal{H}_{\mathcal{Y}}$. When no confusion is possible, we may identify the notations $k_{\mathcal{Y}}(y, \cdot)$ and k_y .

The existence of the conditional mean embeddings $\mu_{\mathbb{P}_|}$ and $\mu_{Q_|}$ is guaranteed by¹⁹⁸ Definition 3.1 under the integrability assumption $\int k_{\mathcal{Y}}(y, y)d\mathbb{P}(y) < +\infty$ and $\int k_{\mathcal{Y}}(\tilde{y}, \tilde{y})d\mathbb{P}(\tilde{y}) < +\infty$. The second integrability assumption $\int k_{\mathcal{X}}(x, x)k_{\mathcal{Y}}(y, y)d(\mathbb{P}_X \otimes \mathbb{P}_|)(x, y) = \int k_{\mathcal{X}} \otimes$

^{*}A kernel k is C_0 -universal if the associated RKHS $\mathcal{H}_k$ is dense in $C_0(\mathcal{X})$, the space of continuous functions on $\mathcal{X}$ vanishing at infinity²³⁷

$k_{\mathcal{Y}}((x, y), (x, y))d(\mathbb{P}_X \otimes \mathbb{P}_Y)(x, y) < +\infty$ guarantees the existence of the mean embedding $\mu_{\mathbb{P}_X \otimes \mathbb{P}_Y}$, defined as:

$$\int k_{\mathcal{X}}(x, \cdot) \otimes k_{\mathcal{Y}}(y, \cdot) d(\mathbb{P}_X \otimes \mathbb{P}_Y)(x, y) \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$$

by⁹⁷ Lemma 3 (and respectively for Q_1). Here, $\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$ is the tensor product Hilbert space of $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$, with kernel $k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$. We actually prove a stronger form of the lemma, given by removing the norm from both hands of the equality and replacing it with a suitable isometric isomorphism $\varphi : \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}} \mapsto \mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}}$

$$\varphi\left(\int k_{\mathcal{X}}(x, \cdot) \otimes k_{\mathcal{Y}}(y, \cdot) d(\mathbb{P}_X \otimes \mathbb{P}_Y)(x, y)\right) = \int K_{\mathcal{X}} \mu_{\mathbb{P}_Y} d\mathbb{P}_X(x)$$

This isometric isomorphism is shown to exist in the “Currying lemma” of Carmeli et al.³⁵, Example 6 regarding tensor product kernels (note that both $\mathcal{X}$ – by assumption – and $\mathcal{Y}$ are locally compact and second-countable). This lemma shows that the mapping:

$$\varphi : \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}} \longrightarrow \mathcal{F}(\mathcal{X}, \mathcal{H}_{\mathcal{Y}})$$

$$f \otimes g \longmapsto \varphi(f \otimes g) = (x \in \mathcal{X} \longmapsto f(x)g \in \mathcal{H}_{\mathcal{Y}})$$

is an isometric isomorphism between $\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$ and $\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}}$. This lemma gives both a representation formula for elements of $\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}}$, and a way to formalize the currying operation, (e.g. the transformation of a function of two variables into a higher-order function of one variable and returning a function of one variable) on tensor-product spaces, since $(f \otimes g)(x, y) = (\varphi(f \otimes g)(x))(y)$. We refer to Carmeli et al.³⁵, Example 6 for a proof. Proceeding with the proof of Lemma 2.5.2, when f and g

are kernel functions $k_{\mathcal{X}}(x, \cdot)$ and $k_{\mathcal{Y}}(y, \cdot)$, the right-hand side of the equality can be related to K_x as

$$\begin{aligned}\varphi(k_{\mathcal{X}}(x, \cdot) \otimes k_{\mathcal{Y}}(y, \cdot))(x') &= k_{\mathcal{X}}(x, x')k_{\mathcal{Y}}(y, \cdot) \\ &= K_x^* K_x k_{\mathcal{Y}}(y, \cdot) \\ &= K_x k_y(x')\end{aligned}$$

where the second to last equality follows from the reproducing property of $\mathcal{H}_{\mathcal{X}, \mathcal{H}_{\mathcal{Y}}}$. Since φ is linear and unitary, it commutes with the mean embedding operation: ⁵⁹ Theorem 36, yielding:

$$\begin{aligned}\varphi\left(\int (k_{\mathcal{X}}(x, \cdot) \otimes k_{\mathcal{Y}}(y, \cdot))d(\mathbb{P}_X \otimes \mathbb{P}_|)(x, y)\right) &= \int \varphi(k_{\mathcal{X}}(x, \cdot) \otimes k_{\mathcal{Y}}(y, \cdot))d(\mathbb{P}_X \otimes \mathbb{P}_|)(x, y) \\ &= \int K_x k_y d(\mathbb{P}_X \otimes \mathbb{P}_|)(x, y)\end{aligned}$$

To complete the proof, it remains to relate the right-hand side to the conditional mean embedding $\mu_{\mathbb{P}_|}$, using

$$\begin{aligned}\int K_x k_y d(\mathbb{P}_X \otimes \mathbb{P}_|)(x, y) &= \iint K_x k_y d\mathbb{P}_X(x) d\mathbb{P}_{|x}(y) \\ &= \int K_x \int k_y d\mathbb{P}_{|x}(y) d\mathbb{P}_X(x) \\ &= \int K_x \mu_{\mathbb{P}_|}(x) d\mathbb{P}_X(x)\end{aligned}$$

as K_x is a bounded linear operator. We thus have that:

$$\varphi\left(\int k_{\mathcal{X}}(x, \cdot) \otimes k_{\mathcal{Y}}(y, \cdot) d(\mathbb{P}_{|X} \otimes \mathbb{P}_|)(x, y)\right) = \int K_x \mu_{\mathbb{P}_|} d\mathbb{P}_X(x)$$

Combining this with the analogue of this result holding for $\mu_{Q_{|}}$ allows to show the stronger form of Lemma 2.5.2. Let us now prove the second part of the lemma. Assume $\text{ACMMD}(\mathbb{P}_|, Q_{|}) = 0$, meaning

$$T_{K_{\mathcal{X}}}(\mu_{\mathbb{P}_|} - \mu_{Q_{|}}) = 0$$

By Carmeli et al. ³⁵, Theorem 2 $K_{\mathcal{X}}$ is a C_0 -universal operator-valued kernel, the operator $T_{K_{\mathcal{X}}}$ is injective. This implies that the conditional mean embeddings of $\mathbb{P}_{|x}$ and $Q_{|x}$ are equal $\mathbb{P}_X$ -almost everywhere. By Park & Muandet ¹⁹⁸, Theorem 5.2 applied to the case where the marginals are equal, and since $k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$ is C_0 -universal, this implies that $\mathbb{P}_{|x} = Q_{|x}$, $\mathbb{P}_X$ -almost everywhere, and in summary, $\text{ACMMD}(\mathbb{P}_{|}, Q_{|}) = 0$ implies $\mathbb{P}_{|x} = Q_{|x}$, $\mathbb{P}_X$ -almost everywhere. To prove the reverse direction, assume that $\mathbb{P}_{|x} = Q_{|x}$, $\mathbb{P}_X$ -almost everywhere. Since Park & Muandet ¹⁹⁸, Theorem 5.2 also prove the reverse direction of the statement relied upon in the previous argument, we have that conversely $\mu_{\mathbb{P}_{|}}(x) = \mu_{Q_{|}}(x)$, $\mathbb{P}_X$ -almost everywhere. By linearity of $T_{K_{\mathcal{X}}}$, we thus have that $T_{K_{\mathcal{X}}}(\mu_{\mathbb{P}_{|}} - \mu_{Q_{|}}) = 0$, and therefore $\text{ACMMD}(\mathbb{P}_{|}, Q_{|}) = 0$. $\square$

B.2 ASYMPTOTIC DISTRIBUTION OF $\widehat{\text{ACMMD}}^2$

As discussed in the main text, it is possible to characterize the asymptotic distribution of $\widehat{\text{NACMMD}}^2$. when $\mathbb{P}_{|} = Q_{|}$, and $\sqrt{N}(\widehat{\text{ACMMD}}^2 - \text{ACMMD}^2)$ when $\mathbb{P}_{|} \neq Q_{|}$. This characterization is given in the next lemma.

Lemma B.2.1. *Assume that the integrability assumptions of Lemma 2.5.2 hold, and that $\mathbb{E}_{Z_1, Z_2} b(Z_1, Z_2)^2 < +\infty$, and that*

- *if $\mathbb{P}_{|x} = Q_{|x}$ $\mathbb{P}(X)$ -a.s, then $\mathbb{E}[\widehat{\text{ACMMD}}] = 0$ and*

$$\widehat{\text{NACMMD}}^2 \xrightarrow{d} \sum_{j=1}^{\infty} \lambda_j (\chi_{1j}^2 - 1)$$

where $\{\chi_{1j}^2\}_{j=1}^{\infty}$ are independent random χ_1^2 variables, and λ_j are the eigenvalues of the operator defined as:

$$\varphi \longmapsto \int b(z, \cdot) \varphi(z) d\mathbb{P}(z)$$

- Assume moreover that $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are C_0 -universal kernels, and that $\sigma_b^2 = 4\mathbb{V}_{Z_2}[\mathbb{E}_{Z_1}b(Z_1, Z_2)] > 0$. Then

$$\sqrt{N}(\widehat{\text{ACMMD}^2} - \text{ACMMD}^2) \xrightarrow{d} \mathcal{N}(0, \sigma_b^2)$$

Proof. Since $\mathbb{E}_{Z_1, Z_2} b(Z_1, Z_2)^2 < +\infty$ we have that $\mathbb{V}_{Z_1, Z_2} \mathbb{E}_{Z_1, Z_2} b(Z_1, Z_2)^2 < +\infty$. Let us define, as in Serfling^{2.2.4}, Section 5.I.5, the function $b(z) = \mathbb{E}_{Z_2} b(z, Z_2)$, and define $\zeta := \mathbb{V}_z b$. For the first point, we will show that if $\mathbb{P}_| = Q_|$, $\mathbb{P}$ -a.s, then $\zeta = 0$, and the result will follow from Serfling^{2.2.4}, Setion 5.5.2. Indeed, noting $k = k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$,

$$\begin{aligned} b(z) &= \mathbb{E}_{Z_2} b(z, Z_2) \\ &= \mathbb{E}_{Z_2} \langle k((x, y), \cdot) - k((x, y'), \cdot), k((X_2, Y_2), \cdot) - k(X_2, \tilde{Y}_2) \rangle \\ &= \langle k((x, y), \cdot), -k((x, y'), \cdot), \mathbb{E}_{z_2} [k((X_2, Y_2), \cdot) - k((X_2, \tilde{Y}_2), \cdot)] \rangle \end{aligned}$$

Where we exchanged the order of integration and inner product, which is possible since $b \mapsto \langle k((x, y), \cdot) - k((x, \tilde{y}), \cdot), b \rangle$ is a bounded linear functional for all $(x, y, \tilde{y})$. Now,

$$\mathbb{E}_{z_2} k((X_2, Y_2), \cdot) - k((X_2, \tilde{Y}_2), \cdot) = \mathbb{E}_{\mathbb{P}_X} [\mathbb{E}_{\mathbb{P}_|} k((X_2, Y_2), \cdot) - \mathbb{E}_{Q_|} k((X_2, \tilde{Y}_2), \cdot)] = 0$$

since $\mathbb{P}_{|x} = Q_{|x}$ $\mathbb{P}_X$ -a.s. Thus, $b(z)$ is a constant function, and $\zeta = 0$. The second case follows by assumption from Serfling^{2.2.4}, Section 5.I.I. $\square$

B.3 PROOF OF LEMMA 2.5.3

Let $\mathcal{H}_{\mathcal{X} \times \mathcal{Y}} := \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$ be the tensor-product RKHS of functions from $\mathcal{X} \times \mathcal{Y}$ with kernel $k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$. The result can be obtained by applying a “coupling” argument, and starting from the

following object:

$$\begin{aligned}\mu_{\mathbb{P}_X \otimes \mathbb{P}_I - \mathbb{P}_X \otimes \mathbb{Q}_I} &:= \int (k((x(\omega), y(\omega)), \cdot) - k((x(\omega), \tilde{y}(\omega)), \cdot)) \, d\mathbb{P}(\omega) \\ &= \mathbb{E}_{x, y, \tilde{y}} [k((x, y), \cdot) - k((x, \tilde{y}), \cdot)]\end{aligned}\tag{B.1}$$

We first show that $\mu_{\mathbb{P}_X \otimes \mathbb{P}_I - \mathbb{P}_X \otimes \mathbb{Q}_I}$ is a well-defined element of $\mathcal{H}_{\mathcal{X} \times \mathcal{Y}}$. Indeed, the following operator

$$T : f \in \mathcal{H} \longmapsto \mathbb{E}_{\tilde{y}} f((x, y)) - f((x, \tilde{y}))$$

satisfies

$$\begin{aligned}|Tf| &\leq \mathbb{E} [|f(x, y)| + |f(x, \tilde{y})|] \\ &\leq \|f\|_{\mathcal{H}_{\mathcal{X} \times \mathcal{Y}}} (\mathbb{E} \sqrt{k((x, y), (x, y))} \\ &\quad + \mathbb{E} \sqrt{k((x, \tilde{y}), (x, \tilde{y}))})\end{aligned}$$

and is bounded thanks to the integrability assumptions of Lemma 2.5.2. Applying the same argument as ⁹⁷ Lemma 3, it follows that the object in Equation (B.1) is well-defined and belongs to $\mathcal{H}_{\mathcal{X} \times \mathcal{Y}}$. Furthermore, by linearity of integration, we have that:

$$\begin{aligned}&\int (k((x(\omega), y(\omega)), \cdot) - k((x(\omega), \tilde{y}(\omega)), \cdot)) \, d\mathbb{P}(\omega) \\ &= \int k((x(\omega), y(\omega)), \cdot) \, d\mathbb{P}(\omega) - \int k((x(\omega'), \tilde{y}(\omega')), \cdot) \, d\mathbb{P}(\omega') \\ &= \mu_{\mathbb{P}_X \otimes \mathbb{P}_I} - \mu_{\mathbb{P}_X \otimes \mathbb{Q}_I}\end{aligned}$$

To conclude, note that:

$$\begin{aligned}
\text{ACMMD}(\mathbb{P}_|, Q_|)^2 &= \left\| \mu_{\mathbb{P}_X \otimes \mathbb{P}_| - \mathbb{P}_X \otimes Q_|} \right\|_{\mathcal{H}_{\mathcal{X} \times \mathcal{Y}}}^2 \\
&= \left\langle \mu_{\mathbb{P}_X \otimes \mathbb{P}_| - \mathbb{P}_X \otimes Q_|}, \mu_{\mathbb{P}_X \otimes \mathbb{P}_| - \mathbb{P}_X \otimes Q_|} \right\rangle_{\mathcal{H}_{\mathcal{X} \times \mathcal{Y}}} \\
&= \left\langle \mathbb{E}_{x, y, \tilde{y}} [k((x, y), \cdot) - k((x, \tilde{y}), \cdot)], \mathbb{E}_{x, y, \tilde{y}} [k((x, y), \cdot) - k((x, \tilde{y}), \cdot)] \right\rangle_{\mathcal{H}_{\mathcal{X} \times \mathcal{Y}}} \\
&= \mathbb{E}_{x_1, y_1, \tilde{y}_1} \mathbb{E}_{x_2, y_2, \tilde{y}_2} b((x_1, y_1, \tilde{y}_1), (x_2, y_2, \tilde{y}_2))
\end{aligned}$$

Where the last equality was obtained by exchanging the order of integration and dot product, possible thanks to the integrability assumptions of Lemma 2.5.2, by using the bilinearity of the inner product and the reproducing property of the kernel k . The symmetry of b in (Z_1, Z_2) follows from the symmetry of $k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$.

B.4 PROOF OF LEMMA 2.5.4

Proof. The proof of the unbiasedness of $\widehat{\text{ACMMD}}^2$ follows by linearity of the expectation, and that each $b((X_i, Y_i, \tilde{Y}_i), (X_j, Y_j, \tilde{Y}_j))$ is an unbiased estimator of $\text{ACMMD}^2(\mathbb{P}_|, Q_|)$. $\square$

B.5 TYPE-I ERROR CONTROL OF THE ACMMD TEST

The goal of this section is to show that the ACMMD test is guaranteed to control its type-I error rate at level α .

B.6 QUANTILE ESTIMATION AND DECISION RULE

We first fully specify the way we compute our quantile estimate $\hat{q}_{1-\alpha}$. Let $b_\alpha := \lceil (1 - \alpha)(B + 1) \rceil$. Given B bootstrap samples $\{\widehat{\text{ACMMD}}_b^2\}_{b=1}^B$ and an $\widehat{\text{ACMMD}}^2$ estimate, we order them in increasing order in a sequence of size $B + 1$, with ties broken arbitrarily. Let $m = \min\{b \in \llbracket 1, B +$

$\mathbb{1} \mid \widetilde{\text{ACMMD}}_b^2 = \text{ACMMD}_{b_\alpha}^2\}$, and $M = \max\{b \in \llbracket 1, B+1 \rrbracket \mid \widetilde{\text{ACMMD}}_b^2 = \text{ACMMD}_{b_\alpha}^2\}$. We set $\widehat{q}_{1-\alpha}$ to be the $(m-1)$ -th element with probability $(b_\alpha - (1-\alpha)(B+1))/(M-m+1)$ (with the convention that the 0-th element is $-\infty$), and the b_α -th element otherwise. The decision rule is then to reject the null hypothesis if $\widetilde{\text{ACMMD}}^2 > q_{1-\alpha}$.

B.7 WILD-BOOTSTRAP AND PERMUTATION-BASED APPROACHES ARE EQUIVALENT IN THE ACMMD TEST

To show that the ACMMD test is guaranteed to control its type-I error rate at level α , we show that the use of a wild bootstrap procedure in the ACMMD test can be cast as a computationally efficient way to approximate the quantiles of the random variable $\widetilde{\text{ACMMD}}^2$ when $\mathbb{P}_{|x} = Q_{|x} \mathbb{P}_X$ -a.e.

Lemma B.7.1. *Let $\{Z_i\}_{i=1}^N$ be i.i.d realizations of $\mathbb{P}_X \otimes \mathbb{P}_Y \otimes Q$, and let $\{W_i^b\}_{i=1 \dots N}^{b=1 \dots B}$ be i.i.d. Rademacher random variables independent of the data. Given a function $\sigma : \llbracket 1, N \rrbracket \mapsto \{-1, 1\}$, define $\{Z_i^\sigma\}_{i=1}^N := \{X_i, Y_i^\sigma, \tilde{Y}_i^\sigma\}_{i=1}^N$, where $(Y_i^\sigma, \tilde{Y}_i^\sigma) = (Y_i, \tilde{Y}_i)$ if $\sigma(i) = 1$, and $(\tilde{Y}_i, Y_i)$ otherwise. Then we have:*

$$\widetilde{\text{ACMMD}}_b^2 = \frac{2}{N(N-1)} \sum_{\substack{i,j=1 \\ i < j}}^N b(Z_i^{\sigma_b}, Z_j^{\sigma_b}) := \text{ACMMD}_{\sigma_b}^2$$

for $\sigma_b(i) := W_i^b$.

The W_i^b should be understood as elements of a random swap σ_b , which for each i , swaps Y_i and $\tilde{Y}_i$ with probability $1/2$.

Proof. Without loss of generality, we fix $i = 1$ and $j = 2$, and fix b , dropping the b index. Note that $b(Z_1, Z_2)$ and $b(Z_1^\sigma, Z_2^\sigma)$ share the same $k_X(X_1, X_2)$. The only differing term is

$$g((Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2)) := k_Y(Y_1, Y_2) + k_Y(\tilde{Y}_1, \tilde{Y}_2) - k_Y(Y_1, \tilde{Y}_2) - k_Y(\tilde{Y}_1, Y_2)$$

and we only need to show that $W_1 W_2 g((Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2)) = g((Y_1^\sigma, \tilde{Y}_1^\sigma), (Y_2^\sigma, \tilde{Y}_2^\sigma))$.

CASE $W^1 = W^2 = 1$ In that case, $Z_1 = Z_1^\sigma$ and $Z_2 = Z_2^\sigma$, and $W_1 W_2 b(Z_1, Z_2) = b(Z_1, Z_2) = b(Z_1^\sigma, Z_2^\sigma)$. by definition of σ .

CASE $W_1 = W_2 = -1$ In that case, we have:

$$g((Y_1^\sigma, \tilde{Y}_1^\sigma), (Y_2^\sigma, \tilde{Y}_2^\sigma)) = k(\tilde{Y}_1, \tilde{Y}_2) + k(Y_1, Y_2)k(\tilde{Y}_1, Y_2) - k(Y_1, \tilde{Y}_2) = b(Z_1, Z_2)$$

implying again $W_1 W_2 b(Z_1, Z_2) = b(Z_1, Z_2) = b(Z_1^\sigma, Z_2^\sigma)$.

CASE $W_1 = 1$ AND $W_2 = -1$ In that case, we have:

$$\begin{aligned} b(Z_1^\sigma, Z_2^\sigma) &= k((X_1, Y_1), (X_2, \tilde{Y}_2)) + k((X_1, \tilde{Y}_1), (X_2, Y_2)) - k((X_1, Y_1), (X_2, Y_2)) - k((X_1, \tilde{Y}_1), (X_2, \tilde{Y}_2)) \\ &= -b(Z_1, Z_2) \end{aligned}$$

meaning again $W_1 W_2 b(Z_1, Z_2) = -b(Z_1, Z_2) = b(Z_1^\sigma, Z_2^\sigma)$, and the last case is proved similarly. □

B.8 LEVEL OF THE ACMMD TEST

We now show that the ACMMD test has the desired type-I error rate.

Lemma B.8.1. *Assume that $\mathbb{P}_{|x} = Q_{|x} \mathbb{P}_X$ -a.s. Then the probability that the ACMMD test rejects the null hypothesis is exactly α .*

The proof consists in 2 steps. First, we show that the decision rule is equivalent to a simpler one. Then, we analyze the latter decision rule.

AN EQUIVALENT DECISION RULE This decision rule is equivalent to the one rejecting H_0 if the position Q (with ties broken uniformly at random) of $\widehat{\text{ACMMD}}^2$ in that sequence satisfies $Q > b_\alpha$, accepting it if $Q < b_\alpha$, and rejecting it with probability $b_\alpha - (1 - \alpha)(B + 1)$ if $Q = b_\alpha$. Indeed, $Q > M \iff \widehat{\text{ACMMD}}^2 > q_{1-\alpha}$ (we always reject), $Q < m \iff \widehat{\text{ACMMD}}^2 \leq q_{1-\alpha}$ (we never reject), and for both rules, when the random position Q is in $\llbracket m, M \rrbracket$, the null is rejected with probability $(b_\alpha - \alpha(B + 1)) / (M - m + 1)$.

ANALYSIS OF THE DECISION RULE We derive the type-I error of our decision rule by analyzing the equivalent, latter one. Our analysis follows a similar argument, in flavor, as Domingo-Enrich et al. ⁶¹, Appendix C. Now, recall that from Lemma B.7.1, the wild bootstrap quantile estimation are draws of $\widehat{\text{ACMMD}}$ on swapped samples Z^σ , e.g. $\{X_i, Y_i^\sigma, \tilde{Y}_i^\sigma\}_{i=1}^N$ parameterized by σ : $\llbracket 1, N \rrbracket \mapsto \{-1, 1\}$ where $Y_i^\sigma = Y_i$ if $\sigma(i) := w_i$ and $Y_i^\sigma = \tilde{Y}_i$ otherwise:

$$(\widehat{\text{ACMMD}}_b^2)_{b=1}^B = (\widehat{\text{ACMMD}}_{\sigma_b}^2)_{b=1}^B$$

using the notation of Lemma B.7.1. Note that σ is a random swap operator such that $\sigma(i) = 1$ with probability 0.5, and $\sigma(i) = -1$ with probability 0.5. If $\mathbb{P}_x = Q_x$ a.e., then since the B swap maps $\sigma_1, \dots, \sigma_B$ are i.i.d. let us note $\widehat{\text{ACMMD}}_{\sigma_0}^2 = \widehat{\text{ACMMD}}^2$, e.g. $\sigma_0(i) = 1$. Then the random sequence $(\widehat{\text{ACMMD}}_{\sigma_b}^2)_{b=0}^B$ is exchangeable. Since Q is the position of $\widehat{\text{ACMMD}}^2$ within that sorted sequence, and that all positions are equally likely under exchangeability, we have:

$$\mathbb{P}[Q < m] = 1 / (B + 1)$$

$$\mathbb{P}[Q > b_\alpha] = (B + 1 - b_\alpha) / (B + 1)$$

$$\mathbb{P}[Q < b_\alpha] = (b_\alpha - 1) / (B + 1)$$

Noting $\Delta((X^i, Y^i, \tilde{Y}^i)_{i=1}^N)$ the event that the null hypothesis is rejected, we have:

$$\begin{aligned}\mathbb{P}[\Delta((X^i, Y^i, \tilde{Y}^i)_{i=1}^N)] &= \mathbb{P}[Q > b_\alpha] + \mathbb{P}[Q = b_\alpha] \mathbb{P}[\text{Reject} | Q = b_\alpha] \\ &= (B + 1 - b_\alpha) / (B + 1) + (b_\alpha - (1 - \alpha)(B + 1)) / (B + 1) = \alpha,\end{aligned}$$

thus showing that the ACMMD test has the desired type-I error rate. $\square$

B.9 PROOFS RELATED TO ACMMD-REL

B.10 DIFFERENCES BETWEEN THE SKCE U-STATISTICS AND THE ACMMD U-STATISTIC

We recall the definition of the SKCE U-statistics estimator from²⁷⁸ Lemma 2:

$$\widehat{\text{SKCE}} = \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} G((Q_{|X_i}, Y_i), (Q_{|X_j}, Y_j)) \quad (\text{B.2})$$

where

$$\begin{aligned}G((q, y), (q', y')) &:= k_{\mathcal{P}(\mathcal{Y}) \times \mathcal{Y}}((q, y), (q', y')) - \mathbb{E}_{Y \sim q} k_{\mathcal{P}(\mathcal{Y}) \times \mathcal{Y}}((q, Y), (q', y')) \\ &\quad - \mathbb{E}_{Y' \sim q'} k_{\mathcal{P}(\mathcal{Y}) \times \mathcal{Y}}((q, y), (q', Y')) + \mathbb{E}_{Y \sim q} \mathbb{E}_{Y' \sim q'} k_{\mathcal{P}(\mathcal{Y}) \times \mathcal{Y}}((q, Y), (q', Y')). \\ &= k_{\mathcal{P}(\mathcal{Y})}(q, q') \times (k_{\mathcal{Y}}(y, y') - \mathbb{E}_{Y \sim q} k_{\mathcal{Y}}(Y, y') - \mathbb{E}_{Y' \sim q'} k_{\mathcal{Y}}(y, Y') + \mathbb{E}_{Y \sim q} \mathbb{E}_{Y' \sim q'} k_{\mathcal{Y}}(Y, Y'))\end{aligned} \quad (\text{B.3})$$

Where the second equality holds when focusing on tensor product kernels. Comparing Equation (B.2) and Equation (B.3) with the expression of the ACMMD U-statistics estimator given in Equation (2.5) and Lemma 2.5.3, we see that the SKCE population criterion equals the ACMMD. However, the SKCE U-statistics estimator is different from the ACMMD U-statistics estimator: while the ACMMD U-statistics only requires samples the conditional distributions $Q_{|X}$, the SKCE

U-statistics contains expectations over the conditional distributions $Q_{|X}$, which are rarely available in practice.

B.11 PROOF OF PROPOSITION 2.6.2

Proof. We will show that the image of $q \mapsto \mu_q$ is compact, and the result will follow from⁴⁵. Let $\mathcal{M}(\mathcal{Y})$ the Banach space of measures of sequences endowed with the total variation norm:

$$\|q\|_{\text{TV}} := q_+(\mathcal{Y}) + q_-(\mathcal{Y})$$

We recall that by the Riesz-Markov theorem, $(\mathcal{M}(\mathcal{Y}), \|\cdot\|_{\text{TV}})$ can be identified with the topological dual of $C_0(\mathcal{Y})$, $(C_0(\mathcal{S})^*, \|\cdot\|_{\text{op}})$ through an isometric isomorphism $q \in \mathcal{M}(\mathcal{Y}) \mapsto \tilde{q} \in C_0(\mathcal{Y})^*$, and for which the following holds:

$$\tilde{q}(f) = \int_{\mathcal{Y}} f d q, \quad \forall f \in C_0(\mathcal{Y}).$$

Let $B := \{q \in \mathcal{M}(\mathcal{Y}) \mid \|q\|_{\text{TV}} \leq 1\}$. As a unit ball, by the Banach-Alaoglu theorem, B is compact under the weak- $\star$ topology and contains all distributions on sequences. We will show that $q \mapsto \mu_q$ is continuous on B and the result will follow. Given that this mapping is linear, it is sufficient to show continuity at 0. Moreover, since $\{B(0_{\mathcal{H}}, r)\}_{r>0}$ is a neighborhood basis of $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$, it suffices to show that there is a neighborhood $\mathcal{V}$ of the null measure in the weak- $\star$ topology such that $\|\int k_{\mathcal{Y}}(y, \cdot) d q(y)\|_{\mathcal{H}}^2 = \int k_{\mathcal{Y}}(y, y') d(q \otimes q)(y, y') < 1$ for all q in $\mathcal{V}$. Since the family

$$\{q \in \mathcal{M}(\mathcal{Y}), \int f_i(x) d q(x) < \varepsilon, \quad i \in 1, \dots, k, f_i \in C_0(\mathcal{Y})\}$$

form a neighborhood basis of the weak- $\star$ topology, we can consider candidates of this form for $\mathcal{V}$. In particular, let us set $\{f_i\} = \{x \mapsto \sqrt{k_{\mathcal{Y}}(x, x)} \in C_0(\mathcal{Y})\}$, since $k_{\mathcal{Y}} \in C_0(\mathcal{Y} \times \mathcal{Y})$, and $\varepsilon = 0.5$. On this neighborhood, we have:

$$\int k(y, y') dq(y) dq(y') \leq \int \sqrt{k(y, y) \times k(y', y')} dq(y) dq(y') \leq 0.5^2 < 1, \quad \forall q \in \mathcal{V}$$

showing the continuity of the map in question. As a consequence, the image of $B_{\mathcal{M}}(S)(0, 1)$ by the map $q \mapsto \mu_q$ is compact, implying from ⁴⁵ that the kernel

$$\tilde{k}(f, g) := \exp\left(-\frac{1}{2\sigma^2} \|f - g\|_{\mathcal{H}}^2\right)$$

is universal on that set. Thus, we have shown that $\tilde{k}$ is universal on $\mathcal{H}$ under the strong topology (e.g. the norm topology in $\mathcal{H}$). This is equivalent to the TV topology of $\mathcal{P}(S)$ since k has discrete masses by proposition 9 of ¹⁰, and thus $k_{\mathcal{P}(\mathcal{Y})}$ is universal on $\mathcal{P}(\mathcal{Y})$. $\square$

B.12 PROOFS REGARDING THE IMPACT OF APPROXIMATE KERNELS

To prove the convergence of the ACMMD-Rel estimator and the validity of its test, we rely on an augmented U-statistics formulation. Let:

$$U := (Q_{|X}, \tilde{Y}^1, \dots, \tilde{Y}^R, \tilde{Y}, Y) \sim \mathbb{P}_{Q_{|X}} \otimes \mathbb{Q}_{|}^{\otimes r} \otimes \mathbb{Q}_{|} \otimes \mathbb{P}_{|}^Q := \mathbb{U}$$

U is the random variable which, for each model $Q_{|X}$, concatenates the synthetic samples $(\tilde{Y}^1, \dots, \tilde{Y}^R)$ used to perform the kernel approximation $\hat{k}_{\mathcal{P}(\mathcal{Y})}(q, q')$, the synthetic sample $\tilde{Y}$ used to evaluate h , and $\tilde{Y}$, a sample from $\mathbb{P}_{|}^Q$ the conditional distribution of Y given $Q_{|X}$. Then, given N realizations

Algorithm 3 ACMMMD–Rel Conditional Goodness of fit Test

Input: $\{X_i, Y_i, \tilde{Y}_i\}_{i=1}^N \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}_X \otimes \mathbb{P}_Y \otimes Q$
Parameters: Level α , kernel k_X , kernel k_Y
 // Estimate ACMMMD–Rel using Algorithm 2 and collect the $\widehat{b}(Z_i, Z_j)$ of Equation (2.11)
 $\widehat{\text{ACMMD}}\text{--Rel}^2, \{b(Z_i, Z_j)\}_{1 \leq i < j \leq N} \leftarrow \text{estimate_acmmd_rel}(\{X_i, Y_i, \tilde{Y}_i\}_{i=1}^N, Q_{|X_i})$
 $[W_i^b \sim \text{Rademacher for } i \in 1, \dots, N \text{ for } b \in 1, \dots, B]$
 $[\widehat{\text{ACMMD}}_b^2 \leftarrow \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} W_i^b W_j^b \widehat{b}(Z_i, Z_j), \text{ for } b \text{ in } 1, \dots, B]$
 // See Section B.6 for how to compute $\widehat{q}_{1-\alpha}$
 $\widehat{q}_{1-\alpha} \leftarrow \text{approx. } (1 - \alpha)\text{-quantile of } \{\widehat{\text{ACMMD}}_b^2\}_{b=1}^B$
if $\widehat{\text{ACMMD}}^2 \leq \widehat{q}_{1-\alpha}$ **then**
 Fail to reject H_0
else
 Reject H_0
end if

$\{U_i\}_{i=1}^N$, of U , the estimator $\widehat{\text{ACMMD}}\text{--Rel}^2$ can be written as a U-statistics on the $\{U_i\}_{i=1}^N$

$$\widehat{\text{ACMMD}}\text{--Rel}^2 = \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} h_a(U_i, U_j)$$

where

$$h_a(U_i, U_j) := \widehat{k}(\{\tilde{Y}_i^r\}_{r=1}^R, \{\tilde{Y}_j^r\}_{r=1}^R) \times (k_Y(Y_i, Y_j) + k_Y(\tilde{Y}_i, \tilde{Y}_j) - k_Y(Y_i, \tilde{Y}_j) - k_Y(\tilde{Y}_i, Y_j))$$

We also will note

$$\text{ACMMD}_a^2 := \mathbb{E}_{U_1, U_2 \sim \mathbb{U} \otimes \mathbb{U}} h_a(U_1, U_2)$$

B.13 PROOF OF PROPOSITION 2.6.4

With this formalism, we now prove that the ACMMMD–Rel test has the specified type-I error rate

of $\alpha \in (0, 1)$, e.g. rejects H_0 when $\mathbb{P}_{|x} = Q_{|x}$ with probability α . Indeed, straightforward adaptations of the arguments in Section B.7 show that that doing a wild bootstrap using the $\hat{b}(Z_i, Z_j)$ is equivalent to estimating

$$\frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} b_a(U_i^\sigma, U_j^\sigma) := \widehat{\text{ACMMD-Rel}}_{\sigma_b}^2$$

where $U_i^\sigma := (Q_{|X_i}, \tilde{Y}_i^1, \dots, \tilde{Y}_i^R, \tilde{Y}_i^\sigma, Y_i^\sigma)$, where $(Y_i^\sigma, \tilde{Y}_i^\sigma) = (Y_i, \tilde{Y}_i)$ if $\sigma(i) = 1$ and $(\tilde{Y}_i, Y_i)$ otherwise. The same argument to show that ACMMD test has the desired type-I error rate follows in this case too: Under $\mathbb{P}_{|x} = Q_{|x}$, the sequence $\{\widehat{\text{ACMMD-Rel}}_{\sigma_b}^2\}_{b=0}^B$ is exchangeable (noting $\widehat{\text{ACMMD-Rel}}_{\sigma_0}^2 = \widehat{\text{ACMMD-Rel}}^2$, e.g. $\sigma_0(i) = 1$ for all $1 \leq i \leq N$), and we can repeat the derivations of the proof of Lemma B.8.1 to show that the ACMMD-Rel test has the desired type-I error rate.

B.14 PROOF OF PROPOSITION 2.6.3

We prove a slightly more general version of the proposition, for kernels of the form $\varphi(d(q, q')^2)$, where φ is a Lipschitz function and d is a distance on $\mathcal{P}(\mathcal{Y})$. Setting $\varphi = e^{-\frac{\cdot}{2}}$, we recover the kernels of Proposition 2.6.3, which include the exponentiated MMD kernel.

Proposition B.14.1. *Assume that $k_{\mathcal{Y}}$ and $k_{\mathcal{P}(\mathcal{Y})}$ is a kernel of the form $k_{\mathcal{P}(\mathcal{Y})}(q, q') = \varphi(d(q, q'))$, for a Lipschitz function φ and a function $d(q, q')$ admitting an unbiased estimator of the form $\hat{d}(\{\gamma_1^r\}_{r=1}^R, \{\gamma_2^r\}_{r=1}^R)$ where $\{\gamma_1^r\}_{r=1}^R$ and $\{\gamma_2^r\}_{r=1}^R$ are R i.i.d samples of q and q' respectively, with variance $O(\frac{1}{R})$ (the bound is uniform in q and q'). Then, assuming $R \equiv R(N)$, with $\lim_{N \rightarrow \infty} R(N) = +\infty$, $\widehat{\text{ACMMD-Rel}}^2$ converges in probability to ACMMD-Rel^2 as $N \rightarrow \infty$.*

Proof. As discussed above, the estimator $\widehat{\text{ACMMD-Rel}}^2$ can be written as a U-statistics on the $\{U_i\}_{i=1}^N$, where $U_i = (Q_{|X_i}, \tilde{Y}_i^1, \dots, \tilde{Y}_i^R, \tilde{Y}_i, Y_i)$, and using the kernel b_a defined as (accounting

approximating k_Y through d directly)

$$b_a(U_i, U_j) := \varphi(\widehat{d}(\{\tilde{Y}_i^r\}_{r=1}^R, \{\tilde{Y}_j^r\}_{r=1}^R)) \times (k_Y(Y_i, Y_j) + k_Y(\tilde{Y}_i, \tilde{Y}_j) - k_Y(Y_i, \tilde{Y}_j) - k_Y(\tilde{Y}_i, Y_j))$$

e.g.

$$\widehat{\text{ACMMD-Rel}}^2 = \frac{2}{N(N-1)} \sum_{i < j} b_a(U_i, U_j)$$

To study the convergence in probability of $\widehat{\text{ACMMD}}_a^2$ to ACMMD_a^2 , we use finite-sample bounds on U -statistics Hoeffding¹⁰⁶:

$$P\left(|\widehat{\text{ACMMD}}_a^2 - \text{ACMMD}_a^2| > \|b_a\|_\infty \sqrt{\frac{\log(2/\delta)}{2\lfloor N/2 \rfloor}}\right) \leq \delta$$

for all $\delta > 0$, where, by assumption, k_Y bounded, and $\widehat{k}_{\mathcal{P}(\mathcal{Y})}$ is of the form $\varphi(\widehat{d}(\{Y_1^r\}_{r=1}^R, \{Y_2^r\}_{r=1}^R))$

for some bounded function φ , implying that b_a is bounded. To show the dependence in R , we

bound the difference ACMMD_a and ACMMD .

$$\begin{aligned} |\text{ACMMD}_a^2 - \text{ACMMD}^2| &= \mathbb{E}_{\mathbb{U}, \mathbb{U}} \left[(\widehat{k}_{\mathcal{P}(\mathcal{Y})}(\{Y_1^r\}_{r=1}^R, \{Y_2^r\}_{r=1}^R) - k_{\mathcal{P}(\mathcal{Y})}(Q_{|X_1}, Q_{X_2})) \times (k_Y(Y_1, Y_2) + k_Y(\tilde{Y}_1, \tilde{Y}_2) \right. \\ &\quad \left. - k_Y(Y_1, \tilde{Y}_2) - k_Y(\tilde{Y}_1, Y_2)) \right] \\ &\leq 4 \|k_Y\|_\infty |\mathbb{E}_{\mathbb{P}_{Q_1} \otimes \mathbb{Q}^{\otimes r} \times \mathbb{P}_{Q_1} \otimes \mathbb{Q}^{\otimes r}} \widehat{k}_{\mathcal{P}(\mathcal{Y})}(\{Y_1^r\}_{r=1}^R, \{Y_2^r\}_{r=1}^R) - k_{\mathcal{P}(\mathcal{Y})}(Q_{|X_1}, Q_{X_2})| \\ &\leq 4 \|k_Y\|_\infty \|\varphi\|_{\text{Lip}} \mathbb{E}_{\mathbb{P}_{Q_1} \otimes \mathbb{Q}^{\otimes r} \times \mathbb{P}_{Q_1} \otimes \mathbb{Q}^{\otimes r}} |\widehat{d}(\{Y_1^r\}_{r=1}^R, \{Y_2^r\}_{r=1}^R) - d(Q_{|X_1}, Q_{X_2})| \\ &\leq 4 \|k_Y\|_\infty \|\varphi\|_{\text{Lip}} \times \\ &\quad \mathbb{E}_{\mathbb{P}_{Q_1} \times \mathbb{P}_{Q_1}} \left[\mathbb{E}_{\mathbb{Q}^{\otimes r} \times \mathbb{Q}^{\otimes r}} |\widehat{d}(\{Y_1^r\}_{r=1}^R, \{Y_2^r\}_{r=1}^R) - d(Q_{|X_1}, Q_{X_2})| \mid Q_{|X_1}, Q_{X_2} \right] \\ &\leq 4 \|k_Y\|_\infty \|\varphi\|_{\text{Lip}} \mathbb{E}_{\mathbb{P}_{Q_1} \times \mathbb{P}_{Q_1}} \left[\sqrt{\mathbb{V}_{\mathbb{Q}^{\otimes r} \times \mathbb{Q}^{\otimes r}} \widehat{d}(\{Y_1^r\}_{r=1}^R, \{Y_2^r\}_{r=1}^R)} \mid Q_{|X_1}, Q_{X_2} \right] \end{aligned}$$

Where the last inequality follows from Jensen's inequality and the unbiasedness of $\widehat{d}$. The result

follows by applying the assumption on the variance of $\widehat{d}$ (a bound which we assume is uniform in $Q_{|X|1}, Q_{|X|2}$). $\square$

The term $\mathbb{V}_{\mathbb{Q}^{\otimes r} \times \mathbb{Q}^{\otimes r}} \left[\widehat{d}(\{Y_1^r\}_{r=1}^R, \{Y_2^r\}_{r=1}^R) | Q_{|X|1}, Q_{|X|2} \right]$ can be more precisely characterized depending on $\widehat{d}$. For instance, we have, when $\widehat{d}$ is a U-statistics (for instance, using the MMD estimator of Gretton et al. ⁹⁷, Lemma 6, Equation 4), that ²²⁴ section 5.2.1 $\mathbb{V}_{\mathbb{Q}^{\otimes r} \times \mathbb{Q}^{\otimes r}} \widehat{d}(\{Y_1^r\}_{r=1}^R, \{Y_2^r\}_{r=1}^R) < \zeta(Q_1, Q_{|X|2})/R$, where $\zeta(Q_{|X|1}, Q_{|X|2}) := \mathbb{V}_{(Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2) \sim Q_{|X|1} \otimes Q_{|X|2}} (\tilde{b}((Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2)))$ and $\tilde{b}((Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2)) = k_Y(Y_1, Y_2) + k_Y(\tilde{Y}_1, \tilde{Y}_2) - k_Y(Y_1, \tilde{Y}_2) - k_Y(\tilde{Y}_1, Y_2)$, which is uniformly bounded by $4 \|k_Y\|_\infty$ for bounded kernels. Putting the two parts together, we thus have that:

$$P \left(\left\{ \widehat{\text{ACMMD}}_a^2 - \text{ACMMD}^2 \right\} > 4 \|k_Y\|_\infty \sqrt{\frac{\log(2/\delta)}{2 \lfloor N/2 \rfloor}} + \frac{16 \|k_Y\|_\infty^2 \|\varphi\|_{\text{Lip}}}{\sqrt{R}} \right) \leq \delta$$

for all $\delta > 0$, showing the convergence in probability of $\widehat{\text{ACMMD}}_a$ to ACMMD.

B.15 ADDITIONAL DETAILS FOR ACMMD AND ACMMD-REL IN THE SYNTHETIC EXAMPLE

B.16 DERIVATIONS OF ACMMD IN THE SYNTHETIC EXAMPLE

We first prove that ACMMD is proportional to Δp .

Lemma B.16.1. *In the setting described in Section 2.8.1, we have*

$$\text{ACMMD}^2(\mathbb{P}_1, Q_1) = C \times \Delta p^2$$

for

$$C := \iint k_X(p, p') 2(1 - e^{-\lambda}) \frac{(1 - 2p)(1 - 2p')}{1 - 4pp'(1 + e^{-\lambda})/2} \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right) d\mathbb{P}_X(p) d\mathbb{P}_X(p')$$

Proof. Recall that we have

$$\text{ACMMD}^2 = \text{MMD}^2(\mathbb{P}_X \otimes \mathbb{P}_|, \mathbb{P}_X \otimes Q_|)^2 = \int k_{\mathcal{X}}(p, p') (T_{11} + T_{22} - 2T_{12}) d\mathbb{P}_X(p) d\mathbb{P}_X(p')$$

where

$$T_{12} = \int k_{\mathcal{Y}}(y, y') p(y|p) q(y'|p') d(y) d(y')$$

and T_{22} and T_{11} are defined similarly. For a sequence y , we define the function len given by $\text{len}(y) := \min \{i \in \mathbb{N} | y_i = \text{STOP}\}$, which intuitively returns the length of the sequence.

COMPUTING T_{ij} As we will see, a lot of the computations are agnostic to whether we are computing T_{11} , T_{22} or T_{12} . Note that the exponentiated hamming distance kernel on $\mathcal{Y}$ writes as a product

$$k_{\mathcal{Y}}(y, y') = e^{-\lambda d_H(y, y')} = e^{-\lambda \sum_{i=0}^{\infty} \delta(y_i \neq y'_i)} = \prod_{i=0}^{\infty} e^{-\lambda \delta(y_i \neq y'_i)} = \prod_{i=0}^{\max(\text{len}(y), \text{len}(y'))} e^{-\lambda \delta(y_i \neq y'_i)}$$

let us define the following events

$$\begin{aligned} F(m) &:= \left\{ \min(\text{len}(y), \text{len}(y')) = m \right\} \\ G(m, \delta m) &:= \left\{ \max(\text{len}(y), \text{len}(y')) = m + \delta m \right\} \end{aligned}$$

which we further break down as

$$\begin{aligned} F_1(m) &= \{\text{len}(y) = m\} \cap \{\text{len}(y') > m\} \\ F_2(m) &= \{\text{len}(y) > m\} \cap \{\text{len}(y') = m\} \\ F_3(m) &= \{\text{len}(y) = m\} \cap \{\text{len}(y') = m\} \\ \implies F(m) &= F_1(m) \cup F_2(m) \cup F_3(m) \end{aligned}$$

For which the following probabilities hold:

$$P(F_1(m)) = P(\text{len}(y) = m) \times P(\text{len}(y') > m) = ((2p)^m \times (1 - 2p)) \times (2p')^{m+1}$$

$$P(F_2(m)) = P(\text{len}(y') = m) \times P(\text{len}(y) > m) = ((2p')^m \times (1 - 2p')) \times (2p)^{m+1}$$

$$P(F_3(m)) = P(\text{len}(y') = m) \times P(\text{len}(y) = m) = ((2p')^m \times (1 - 2p')) \times ((2p)^m \times (1 - 2p))$$

$$P(G(m, \delta m) | F_1(m)) = P(\text{len}(y') = m + \delta m | \text{len}(y) = m, \text{len}(y') > m) = (2p')^{\delta m - 1} \times (1 - 2p') \delta_{(\delta m \geq 1)}$$

$$P(G(m, \delta m) | F_2(m)) = P(\text{len}(y) = m + \delta m | \text{len}(y') = m, \text{len}(y) > m) = (2p)^{\delta m - 1} \times (1 - 2p) \delta_{(\delta m \geq 1)}$$

$$P(G(m, \delta m) | F_3(m)) = \delta(\delta m = 0)$$

Let us note

$$E(m, \delta m, i) := F_i(m) \cap G(m, \delta m)$$

We have that $E(m, \delta m, i) \cap E(m', \delta m', j) = \emptyset$ if $(m, \delta m, i) \neq (m', \delta m', j)$.

$$\Omega = \bigcup_{m=0}^{+\infty} \bigcup_{i=1}^3 \bigcup_{\delta m=0}^{+\infty} E(m, \delta m, i)$$

Using the law of total probability, we have that Thus, using the law of total probability:

$$\begin{aligned}
T_{ij}(p, p') &= \sum_{m=0}^{+\infty} \sum_{i=1}^3 \sum_{\delta m=0}^{+\infty} \mathbb{P}(E(m, \delta m, i)) \mathbb{E}(e^{-\lambda d_H(y, y')} | E(m, \delta m, i)) \\
&= \sum_{m=0}^{+\infty} \sum_{i=1}^3 \sum_{\delta m=0}^{+\infty} \mathbb{P}(F_i(m) \cap G(m, \delta m)) \mathbb{E}(e^{-\lambda d_H(y, y')} | E(m, \delta m, i)) \\
&= \sum_{m=0}^{+\infty} \sum_{i=1}^3 \mathbb{P}(F_i(m)) \sum_{\delta m=0}^{+\infty} P(G(m, \delta m) | F_i(m)) \mathbb{E}(e^{-\lambda d_H(y, y')} | E(m, \delta m, i)) \\
&= \sum_{m=0}^{+\infty} \sum_{i=1}^3 \mathbb{P}(F_i(m)) \mathbb{E}(e^{-\lambda d_H(y_{m:m}, y'_{m:m})} | F_i(m)) \\
&\quad \times \sum_{\delta m=0}^{+\infty} P(G(m, \delta m) | F_i(m)) \mathbb{E}(e^{-\lambda d_H(y_{m+1:m+\max(\delta m, 1)}, y'_{m+1:m+\max(\delta m, 1)})} | E(m, \delta m, i), p, p') \\
&= \sum_{m=0}^{+\infty} \sum_{i=1}^3 \mathbb{P}(F_i(m)) \mathbb{E}(e^{-\lambda d_H(y_{m:m}, y'_{m:m})} | F_i(m)) \times \sum_{\delta m=0}^{+\infty} P(G(m, \delta m) | F_i(m)) e^{-\lambda(\max(0, \delta m-1) + \delta(m>0))} \\
&= \sum_{m=0}^{+\infty} \sum_{i=1}^3 \mathbb{P}(F_i(m)) \mathbb{E}(e^{-\lambda d_H(y_{m:m}, y'_{m:m})} | F_i(m)) \left(\prod_{i=1}^{\max(m-1, 1)} \mathbb{E}(e^{-\lambda \delta(y_i \neq y'_i)} | F_i(m)) \right)^{\delta(m \geq 2)} \\
&\quad \times \sum_{\delta m=0}^{+\infty} P(G(m, \delta m) | F_i(m)) e^{-\lambda(\max(0, \delta m-1) + \delta(m>0))}
\end{aligned}$$

where we break down the factorized hamming distance over the sequence into the sum of the hamming distances over each coordinate, and made use of the fact that

$$d_H(y_{m:m+\delta m}, y'_{m:m+\delta m}) = \max(0, \delta m - 1) + \delta(m > 0)$$

conditioned on $F_i(m)$ and $G(m, \delta m)$. The disjunction of cases is necessary in order to not count the term 0^{th} term twice in the event when $m = 0$. This representation is convenient since whenever

$m \geq 2$, for any $1 \leq i \leq m-1$,

$$P(\delta(y_i, y'_i) = 1 | F_i(m)) = \frac{(p p') + (p p')}{(p + p) \times (p' + p')} = \frac{1}{2} = P(\delta(y_i, y'_i) = 0 | F_i(m))$$

meaning we have

$$\begin{aligned} T_{ij}(p, p') &= \sum_{m=0}^{+\infty} \sum_{i=1}^3 \mathbb{E}(e^{-\lambda \delta(y_0 \neq y'_0)} | F_i(m)) \mathbb{P}(F_i(m)) \left(\frac{1 + e^{-\lambda}}{2} \right)^{\max(m-1, 0)} \\ &\times \sum_{\delta m=0}^{+\infty} P(G(m, \delta m) | F_i(m)) e^{-\lambda(\max(0, \delta m-1) + \delta(m > 0))} \end{aligned}$$

Inserting the relevant event probabilities into the expression for T_{ij} , we have

$$\begin{aligned}
T_{ij}(p, p') &= \sum_{m=0}^{+\infty} \left(\frac{1+e^{-\lambda}}{2} \right)^{\max(m-1, 0)} \\
&\quad \times \left(\mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_1(m)) (2p)^m \times (1-2p)(2p')^{m+1} \times (1-2p')e^{-\lambda\delta(m>0)} \sum_{\delta m=1}^{+\infty} e^{-\lambda(\delta m-1)} (2p')^{\delta m-1} \right. \\
&\quad + \mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_2(m)) (2p')^m \times (1-2p')(2p)^{m+1} \times (1-2p)e^{-\lambda\delta(m>0)} \sum_{\delta m=1}^{+\infty} e^{-\lambda(\delta m-1)} (2p)^{\delta m-1} \\
&\quad \left. + \mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_3(m)) (2p')^m \times (1-2p')(2p)^m (1-2p) \right) \\
&= \sum_{m=0}^{+\infty} \left(\frac{1+e^{-\lambda}}{2} \right)^{\max(m-1, 0)} \\
&\quad \times \left(\mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_1(m)) (2p)^m \times (1-2p)(2p')^{m+1} \times (1-2p')e^{-\lambda\delta(m>0)} \sum_{\delta m=0}^{+\infty} e^{-\lambda\delta m} (2p')^{\delta m} \right. \\
&\quad + \mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_2(m)) (2p')^m \times (1-2p')(2p)^{m+1} \times (1-2p)e^{-\lambda\delta(m>0)} \sum_{\delta m=0}^{+\infty} e^{-\lambda\delta m} (2p)^{\delta m} \\
&\quad \left. + \mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_3(m)) (2p')^m \times (1-2p')(2p)^m (1-2p) \right) \\
&= \sum_{m=0}^{+\infty} \left(\frac{1+e^{-\lambda}}{2} \right)^{\max(m-1, 0)} \\
&\quad \times \left(\mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_1(m)) (2p)^m \times (1-2p)(2p')^{m+1} \times (1-2p') \times \frac{e^{-\lambda\delta(m>0)}}{1-2p'e^{-\lambda}} \right. \\
&\quad + \mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_2(m)) (2p')^m \times (1-2p')(2p)^{m+1} \times (1-2p) \times \frac{e^{-\lambda\delta(m>0)}}{1-2pe^{-\lambda}} \\
&\quad \left. + \mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_3(m)) (2p')^m \times (1-2p')(2p)^m (1-2p) \right)
\end{aligned}$$

Now, some simplification arise when $m \geq 1$. Indeed, in that case, $\mathbb{E}(e^{-\lambda \delta(y_0, y'_0)} | F_i(m))$ is independent of i . Noting $T_{ij}^1(p, p')$ the sum of the terms for $m \geq 1$, we thus have

$$\begin{aligned} T_{ij}^1(p, p') &= \sum_{m=1}^{+\infty} \mathbb{E}(e^{-\lambda \delta(y_0, y'_0)} | F(m)) \left(\frac{1 + e^{-\lambda}}{2} \right)^{m-1} \\ &\quad \times \left((2p)^m \times (1 - 2p)(2p')^{m+1} \times (1 - 2p') \times \frac{e^{-\lambda}}{1 - 2p'e^{-\lambda}} \right. \\ &\quad + (2p')^m \times (1 - 2p')(2p)^{m+1} \times (1 - 2p) \times \frac{e^{-\lambda}}{1 - 2pe^{-\lambda}} \\ &\quad \left. + (2p')^m \times (1 - 2p')(2p)^m(1 - 2p) \right) \end{aligned}$$

Noting A_{ij} the term $\mathbb{E}(e^{-\lambda \delta(y_0, y'_0)} | F(m))$, which is constant for all $m \geq 1$

$$\begin{aligned} T_{ij}^1(p, p') &= A_{ij} \sum_{m=1}^{+\infty} \left(\frac{1 + e^{-\lambda}}{2} \right)^{m-1} \\ &\quad \times \left((2p)^m \times (1 - 2p)(2p')^{m+1} \times (1 - 2p') \times \frac{e^{-\lambda}}{1 - 2p'e^{-\lambda}} \right. \\ &\quad + (2p')^m \times (1 - 2p')(2p)^{m+1} \times (1 - 2p) \times \frac{e^{-\lambda}}{1 - 2pe^{-\lambda}} \\ &\quad \left. + (2p')^m \times (1 - 2p')(2p)^m(1 - 2p) \right) \\ &= A_{ij}(1 - 2p)(1 - 2p')4pp' \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right) \sum_{m=0}^{+\infty} (4pp'(1 + e^{-\lambda})/2)^m \\ &= A_{ij} \times \frac{(1 - 2p)(1 - 2p')4pp'}{1 - 4pp'(1 + e^{-\lambda})/2} \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right) \\ &= C \times A_{ij} \end{aligned}$$

where

$$C(p, p') = \frac{(1 - 2p)(1 - 2p')4pp'}{1 - 4pp'(1 + e^{-\lambda})/2} \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right)$$

is a constant that does not depend on i, j . We compute the $m = 0$ sum, noted $T_{ij}^0(p, p')$. We have

$$\begin{aligned} T_{ij}^0(p, p') = & \left(\mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_1(0)) \times (1 - 2p)(2p') \times (1 - 2p') \times \frac{1}{1 - 2p'e^{-\lambda}} \right. \\ & + \mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_2(0)) \times (1 - 2p')(2p) \times (1 - 2p) \times \frac{1}{1 - 2pe^{-\lambda}} \\ & \left. + \mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_3(0)) \times (1 - 2p')(1 - 2p) \right) \end{aligned}$$

And we need to compute the terms $\mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_i(0))$ individually.

I=1, I=2 For $i = 1$, we must have $y_0 \neq y'_0$, since $y_0 = \text{STOP}$, and $\text{len}(y') > 0$. Thus, $\mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_1(0)) = e^{-\lambda}$. Similarly, $\mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_2(0)) = e^{-\lambda}$.

I=3 In that case, we must have $y_0 = y'_0 = \text{STOP}$, since $\text{len}(y) = \text{len}(y') = 0$. Thus, $\mathbb{E}(e^{-\lambda\delta(y_0 \neq y'_0)} | F_3(0)) = 1$.

Putting this together, we have

$$T_{ij}^0(p, p') = (1 - 2p)(1 - 2p') \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right)$$

With that notation, we have:

$$\begin{aligned} \text{ACMMD}^2(\mathbb{P}_1, \mathbb{Q}_1) &= \int k_{\mathcal{X}}(p, p') C(p, p') (A_{11} + A_{22} - 2A_{12}) d\mathbb{P}_X(p) d\mathbb{P}_X(p') \\ &+ \int k_{\mathcal{X}}(p, p') (T_{11}^0(p, p') + T_{22}^0(p, p') - 2T_{12}^0(p, p')) d\mathbb{P}_X(p) d\mathbb{P}_X(p') \\ &= \int k_{\mathcal{X}}(p, p') C(p, p') (A_{11} + A_{22} - 2A_{12}) d\mathbb{P}_X(p) d\mathbb{P}_X(p') \end{aligned}$$

since T_{ij}^0 does not depend on i, j . We can narrow the variation down even further: by noting $p_{ij}^A = P(\delta(y_i \neq y'_i) = 0 | F(m))$ (resp $p_{ij}^B = P(\delta(y_i \neq y'_i) = 0 | F(0))$), since $\mathbb{E}(e^{-\lambda\epsilon}) = p(\epsilon =$

$0)(1 - e^{-\lambda}) + e^{-\lambda}$ if ε is a Bernoulli random variable,

$$\text{ACMMD}^2(\mathbb{P}_|, \mathbb{Q}_|) = \int k_X(p, p') C(p, p') (1 - e^{-\lambda}) (p_{11}^A + p_{22}^A - 2p_{12}^A) d\mathbb{P}_X(p) d\mathbb{P}_X(p')$$

We now compute the probabilities p_{ij}^A for $i, j \in \{1, 2\}$. In every case, such p_{ij}^A can be written as:

$$p_{ij}^A = \frac{P(\gamma_0 = \gamma'_0 = A) + P(\gamma_0 = \gamma'_0 = B)}{P(\{\gamma_0 \in \{A, B\}\} \cap \{\gamma'_0 \in \{A, B\}\})} \frac{P(\gamma_0 = \gamma'_0 = A) + P(\gamma_0 = \gamma'_0 = B)}{4pp'}$$

and we have

$$\begin{aligned} p_{11}^A &= \frac{pp' + pp'}{4pp'} = \frac{1}{2} \\ p_{22}^A &= \frac{(p + \Delta p)(p' + \Delta p) + (p - \Delta p)(p' - \Delta p)}{4pp'} = \frac{2pp' + 2\Delta p^2}{4pp'} \\ p_{12}^A &= \frac{(p)(p' + \Delta p) + (p)(p' - \Delta p)}{4pp'} = \frac{1}{2} \\ \implies p_{11}^A + p_{22}^A - 2p_{12}^A &= \frac{2pp' + 2\Delta p^2}{4pp'} - \frac{1}{2} = \frac{2\Delta p^2}{4pp'} \end{aligned}$$

PUTTING IT TOGETHER We thus have

$$\text{ACMMD}(\mathbb{P}_|, \mathbb{Q}_|) = \int C(p, p') k(p, p') (1 - e^{-\lambda}) \frac{2\Delta p^2}{4pp'} d\mathbb{P}_X(p) d\mathbb{P}_X(p')$$

Recalling that

$$C(p, p') = \frac{(1 - 2p)(1 - 2p')4pp'}{1 - 4pp'(1 + e^{-\lambda})/2} \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right)$$

yields the desired result. □

B.17 CLOSED-FORM ACMMD-Rel EVALUATION

Assuming the same model, it is also possible to evaluate $\text{ACMMD-Rel}(\mathbb{P}_|, Q_|)$ in closed form. Indeed, ACMMD-Rel becomes a special case of the ACMMD formula given above, with the conditioned variable X set to be the models $Q_{|X}$. It is thus possible to show:

Lemma B.17.1. *We have*

$$\text{ACMMD-Rel}^2(\mathbb{P}_|, Q_|) = C \times \Delta p^2$$

for

$$C = \iint k_{\mathcal{P}(\mathcal{Y})}(q_{|p}, q_{|p'}) 2(1 - e^{-\lambda}) \frac{(1 - 2p)(1 - 2p')}{1 - 4pp'(1 + e^{-\lambda})/2} \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right) d\mathbb{P}_X(p) d\mathbb{P}_X(p')$$

The above lemma leaves the choice of the kernel $k_{\mathcal{P}(\mathcal{Y})}$ open: the tractability of this expression will follow only if such kernel can be tractably computed. In the next lemma, we derive a closed form solution for $k_{\mathcal{P}(\mathcal{Y})}(q, q')$ when $k_{\mathcal{P}(\mathcal{Y})}(q, q') = e^{-\frac{\text{MMD}^2(q, q')}{2\sigma^2}}$, where the MMD is computed with an Exponentiated Hamming kernel on $\mathcal{Y}$.

Lemma B.17.2. *We have*

$$\text{MMD}^2(q_{|p}, q_{|p'}) = T(p, p) + T(p', p') - 2T(p, p')$$

Where

$$\begin{aligned} T(p, p') &= C(p, p')A(p, p') + T^0(p, p') \\ C(p, p') &= \frac{(1 - 2p)(1 - 2p')4pp'}{1 - 4pp'(1 + e^{-\lambda})/2} \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right) \\ A(p, p') &= \frac{2pp' + 2\Delta p^2}{4pp'} \times (1 - e^{-\lambda}) + e^{-\lambda} \\ T^0(p, p') &= (1 - 2p)(1 - 2p') \left(\frac{2p'e^{-\lambda}}{1 - 2p'e^{-\lambda}} + \frac{2pe^{-\lambda}}{1 - 2pe^{-\lambda}} + 1 \right) \end{aligned}$$

Combining the two lemmas allows us to obtain a computable expression for $\text{ACMMD-Rel}(\mathbb{P}_l, Q_l)$.

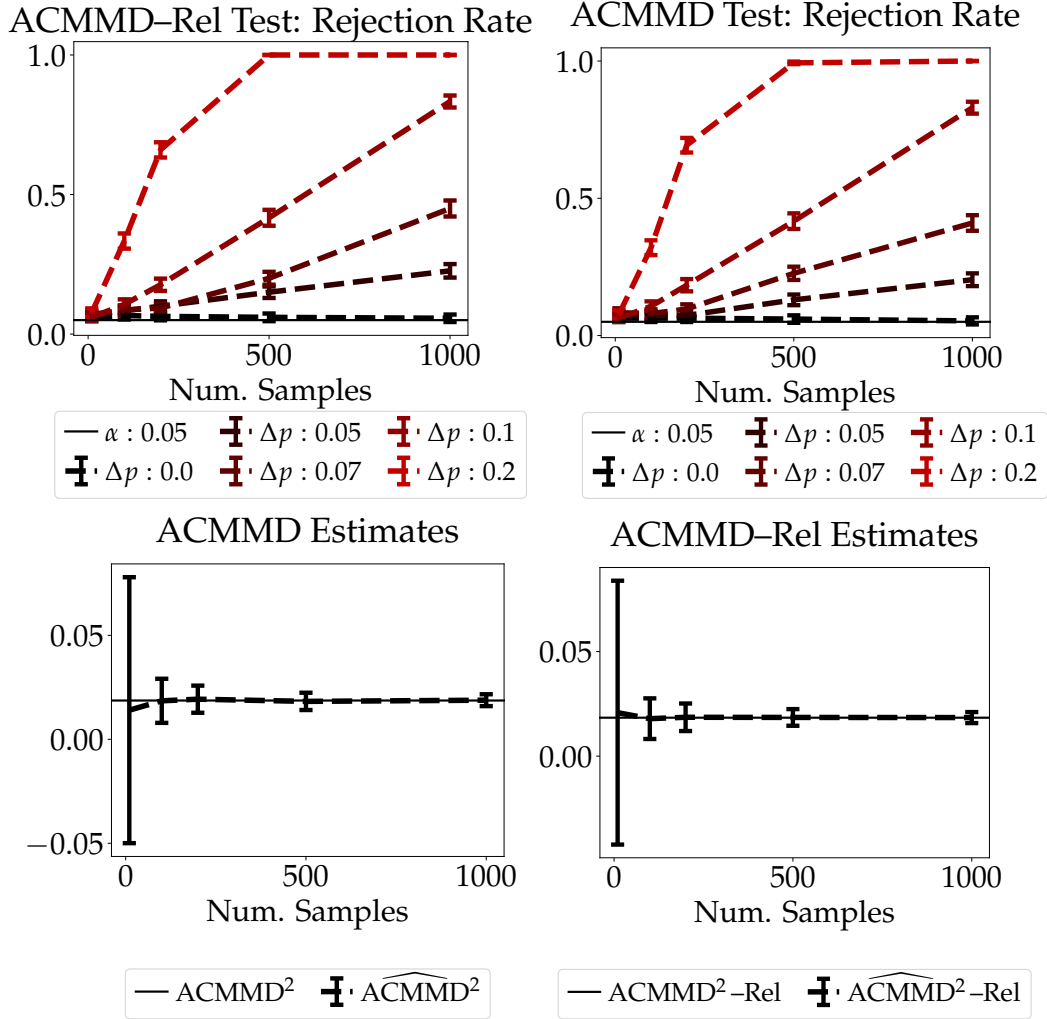


Figure B1: Top left: Rejection Rate of the ACMMD test as a function of the dataset size, for different values of Δp . Top right: Rejection Rate of the ACMMD test as a function of Δp , for different dataset sizes. Bottom left: Estimated ACMMD as a function of the dataset size. Bottom right: Estimated ACMMD-Rel as a function of Δp . To compute these estimates, we use dataset sizes of $\{10, 100, 200, 500, 1000\}$, used $m = 5$ atoms for the prior on p between $p_1 = 0.3, p_2 = 0.45$, used $\lambda = 1, \Delta p = 0.25$, and average over 300 runs. In addition, we plot the true value $\text{ACMMD}(\mathbb{P}_l, Q_l)$ using the closed-form expression derived above.

B.18 ADDITIONAL EXPERIMENTS

B.18.1 ADDITIONAL EXPERIMENTS FOR THE SEMI-SYNTHETIC PROTEINMPNN DATA

In addition to the figures of Section 2.8.2, which use $T = 0.1$ to plot the estimates and rejection rates of ACMMD and ACMMD-Rel on the ProteinMPNN synthetic data, we provide here the same plots for $T = 1.0$ the value used to train ProteinMPNN. We notice that detecting a given change in temperature is slightly simpler for $T = 1.0$ than for $T = 0.1$.

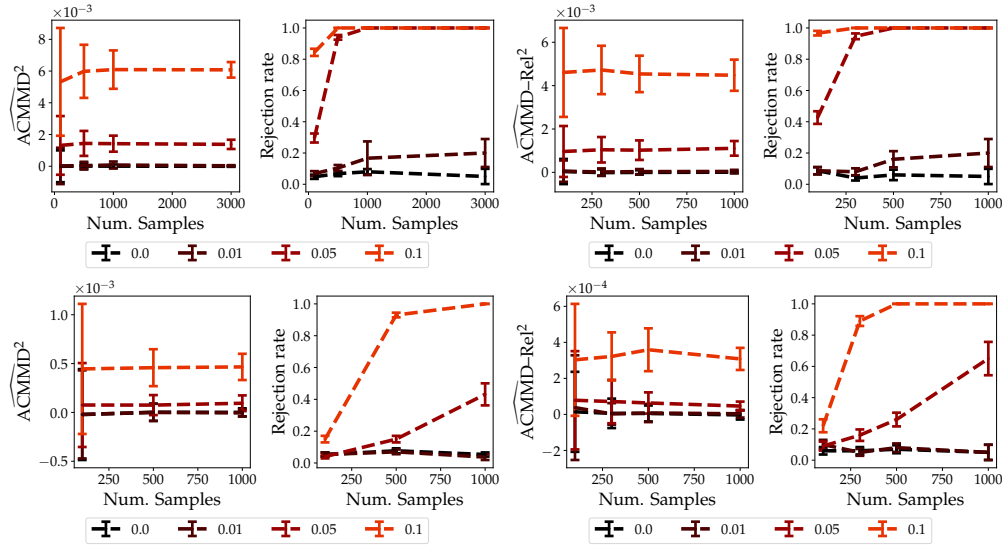


Figure B2: ACMMD and ACMMD-Rel estimates and rejection rate in the semi-synthetic setting of Section 2.8.2. The different lines indicate a different temperature shift between the two MPNN models. Top panel shows uses a base temperature of $T = 1.0$, while the bottom panel uses $T = 0.1$.

B.18.2 ADDITIONAL EXPERIMENTS FOR THE STRUCTURAL SUPERFAMILY EVALUATION

We include in Figure B3 the values of $\widehat{\text{ACMMD-Rel}}^2$ for different superfamilies (which was not included in Section 2.8.2), and compare it with the values of $\widehat{\text{ACMMD}}^2$. In line with the hyperparameter tuning results of Section 2.8.2, we notice that high temperature are highly detrimental

from a reliability perspective. Intuitively, increasing the temperature of ProteinMPNN makes the model “underconfident”. Since a reliable model is neither over- nor underconfident, this decrease of confidence is penalized by $\widehat{\text{ACMMD}}\text{-Rel}$. This also shows that increasing the temperature of a model does not make the model fall back to its prior distribution (otherwise the model would be more reliable). Instead, it just increases the uncertainty of the model in a detrimental fashion.

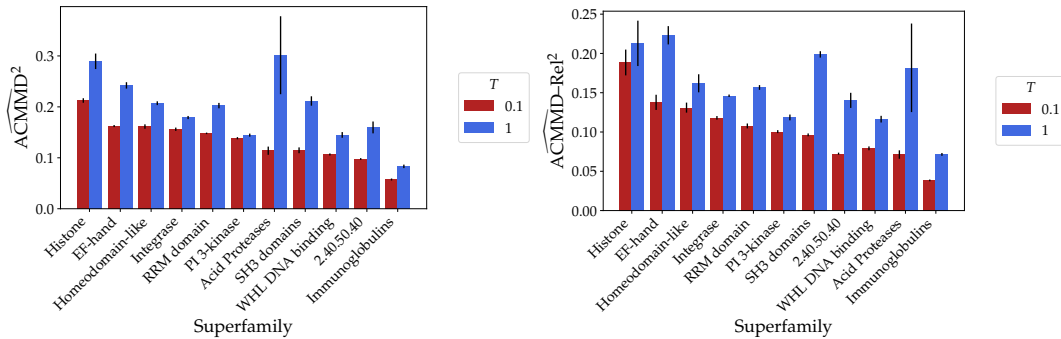


Figure B3: Value of $\widehat{\text{ACMMD}}^2$ (left) and $\widehat{\text{ACMMD}}\text{-Rel}^2$ (right) between ProteinMPNN and the CATH S60 reference dataset on a subset of 10 superfamilies for two different temperatures $T = 1.0$ and $T = 0.1$.

B.19 KNOWN KERNELS FOR PROTEIN SEQUENCES AND STRUCTURES

In the context of inverse folding, computing the ACMMD requires a kernel on sequences k_Y and a kernel on protein structures. This section contains a brief overview of non neural-network based, known kernels for protein sequences and structures. The main desiderata to achieve when choosing kernels for computing goodness-of-fit criterion is to find kernels that are able to detect (up to statistical noise) any deviation from a perfect fit between the model and the data. Such kernels are referred to as *universal*.

THE PROTEIN FORMALISM The most general formalism for the space protein structures is the set of equivalence classes of graphs where the equivalence relationship is defined to the existence of

a graph isomorphism. The need for equivalence classes stems from the fact that different labelling policies exist for a given protein, meaning that a single protein can be associated to multiple graphs. However, this labelling will in practice not be completely arbitrary: first, the set of candidate labelling can be restricted to the ones consistent with covalent bounds. But in the inverse folding problem, the setting is even simpler: the protein structure is restricted to its backbone, which is sequential by nature. This limits the set of covalent-bound consistent labelling policies to two (the forward and the backward one), and my vague understanding is that there is a terminal atom in protein, which suggests the existence of a canonical direction: thus, only one labelling policy remain, and protein structures can thus be associated to the set of atom locations $\bigcup_{i=1}^{+\infty} \mathbb{R}^i$. This set differs from the set of protein sequences $\bigcup_{i=1}^{+\infty} \mathcal{A}$ in that the “alphabet” is the real line instead of a finite set of symbols. The restriction from the space of graphs to the space of variable-length sequences since there it is known that no graph kernels commonly in use are even characteristic¹⁴¹. The space $\bigcup_{i=1}^{+\infty} \mathcal{X}$ (for arbitrary $\mathcal{X}$ have been investigated by the time series community), which have developed a set of kernels to carry out data analysis on it. We provide some background on such kernels below.

BACKGROUND: ALIGNMENT KERNELS FOR REAL-VALUED SEQUENCES OF ARBITRARY LENGTH

Alignment kernels^{50,49,217,269} refer to a diverse set of variety of kernels constitute a family of kernels on $\bigcup_{i=1}^{+\infty} \mathcal{X}^i$ that are computed based on aggregating the similarities between all possible “alignment candidates” between two inputs x_1 and x_2 . There are two main subfamilies of alignment kernels, which both use slightly different alignment definitions: local alignment kernels, and global alignment kernels.

LOCAL ALIGNMENT KERNELS Local alignment kernels^{217,269} are kernels of the form

$$k_{\text{LA}}(x, y) = \sum_{\pi \in \Pi(x, y)} \exp(\beta s(x, y, \pi)) \quad (\text{B.4})$$

Where

$$s(x, y, \pi) = \sum_{i=1}^{|\pi|} s(x_1^{(\pi_1(i))}, x_2^{(\pi_2(i))}) + \sum_{i=1}^{|\pi|-1} g(\pi_1(i+1) - \pi_1(i)) + g(\pi_2(i+1) - \pi_2(i))$$

and $\Pi(x, y)$ is the set of all possible *alignments* of x and y , e.g. the set of all 2-tuple of p -long sequences

$$\pi := ((\pi_1(1), \dots, \pi_1(p)), (\pi_2(1), \dots, \pi_2(p)))$$

where

$$1 \leq \pi_1(1) < \pi_1(2) \cdots < \pi_1(p) \leq n$$

$$1 \leq \pi_2(1) < \pi_2(2) \cdots < \pi_2(p) \leq m$$

Importantly, local alignment kernels involve a gap function, and thus give a specific status to insertions and deletions, unlike global alignment kernels, as we will see below. The local alignment kernel can be seen as computing the (soft) minimum of a discrepancy within the set of all possible alignments. The use of a soft minimum (and not a hard one) is crucial to ensure positive definiteness. Local alignment kernels seem to have been designed initially for finite alphabets target. When $g = 0$, the necessary and sufficient condition on s to ensure that k_{LA} is a positive definite is for s to be a conditionally positive definite kernel[†]. This is in particular verified if $(s(x^i, y^j))_{1 \leq i, j \leq |\mathcal{A}|}$ is positive definite. I need further reading to investigate whether the case of infinite $\mathcal{X}$ was studied.

[†]A kernel is c.p.d if $\sum_{i,j=1}^n c_i c_j s(x^{(i)}, x^{(j)}) \geq 0 \forall c_1, \dots, c_n, c_1 + \dots + c_n = 0$.

GLOBAL ALIGNMENT KERNELS Global alignment kernels^{50,49} also perform a softmin over alignment, but do not incorporate gaps in their score, and use a slightly different notion of alignment, namely:

$$\pi := ((\pi_1(1), \pi_2(1)), \dots, (\pi_1(p), \pi_2(p)))$$

where now, the constraints on π_1 and π_2 are

$$1 = \pi_1(1) < \pi_1(2) \cdots < \pi_1(p) = n$$

$$1 = \pi_2(1) < \pi_2(2) \cdots < \pi_2(p) = m$$

$$\pi_1(i+1) \leq \pi_1(i) + 1 \quad \text{unitary increments}$$

$$(\pi_1(i+1) - \pi_1(i)) + (\pi_2(i+1) - \pi_2(i)) \geq 1 \quad \text{no repetitions}$$

Unlike the previous alignment definition, this one explicitly maps each item in each sequence with another item in the other sequence, and does not try to account for potential gaps. Let us call $\mathcal{A}$ the set of all alignment. The final definition for a global alignment kernel is then:

$$k_{\text{GA}}(x, y) = \sum_{\pi \in \mathcal{A}(x, y)} \exp\left(\sum_{i=1}^{\pi} s(x_1^{(\pi_1(i))}, x_2^{(\pi_2(i))})\right) \quad (\text{B.5})$$

As stated in Cuturi et al.⁵⁰, Theorem 1, k_{GA} will be positive definite if $k(x, y) := \exp(s(x, y))$ is a positive definite kernel such that $\frac{k}{(1-k)}$ is positive definite.



Appendix to Chapter 3

C.1 SUPPLEMENTARY METHODS

C.1.1 GENERATION OF ERROR-PRONE PCR LIBRARIES

Nucleotide errors were introduced into parent nanobody and protein DNA sequences using the GeneMorph II random mutagenesis kit (Agilent) and the AB_MFAL_f and AB_HA_r primers. To achieve DNA pools with different error rates, the amount of template DNA and PCR primer

cycles were varied. The error rates of the resulting PCR pools were checked by Gibson cloning nanobody/protein inserts possessing Gibson overlap overhangs added with the primers into pET26b_3xHA vector linearized with primers and then sanger sequencing 15 random bacterial colonies from the resultant transformations. A second PCR using the AB_lib_fwd and AB_lib_rvs primers was performed on nanobody/protein pools that possessed average error rates greater than 1 to add homology arms needed for yeast homologous recombination. To generate linearized pYDS2.0 vector for homologous recombination, the pYDS2.0 vector was digested with NheI and BamHI (New England Biolabs), which recognize sites immediately preceding and postceding the nanobody/protein insert sequences, by incubating the vector with the restriction digest enzymes for 1 hour at 37°C. Both the double digested vector and nanobody/protein inserts possessing homology arms were PCR purified and then ethanol precipitated to remove salt. First, 0.1x volume 3M sodium acetate pH 5 and 2.5x volume 100% ethanol were incubated with the DNA overnight at -20°C. Following centrifugation at 13,000xRPM for 30 minutes, two sequential washes were performed using 70% ethanol, and DNA pellets were dried at room temperature.

To prepare yeast for library building by homologous recombination, BJ5465 yeast (Addgene) were grown to an OD of 1.5-2, were washed once with water, and then were incubated with 100 mM lithium acetate and 10 mM DTT for 10 minutes at 30°C with shaking. Yeast were washed once more with water and were resuspended in mixture of 5:1 insert to vector in which each 50 mL electroporation reaction contained 5 mg of insert and 1 mg of vector. Electroporations were performed using an ECM 830 Electroporation (BTX-Harvard Apparatus) using square wave pulses at 500V for 15 ms. To obtain estimates of library diversities, dilutions of the transformed yeast were plated on YPAD containing 100 µg/mL nourseothricin to obtain single colony counts. Libraries were frozen using yeast freezing media supplemented with 10% DMSO.

C.1.2 PCR AMPLIFICATION AND NEXT GENERATION SEQUENCING OF NANOBODY SELECTION ROUNDS

Yeast from each nanobody selection round were lysed by microwave and nanobody sequences from the yeast were PCR amplified using primers containing unique barcodes. Briefly, 4e6 yeast were added to multiple PCR tubes and the yeast media was removed by centrifugation. Then, yeast were lysed by microwaving them using the high setting twice for 1 minute each, flipping the PCR strip upside down in between microwave rounds. PCR amplification was conducted using Q5 High-Fidelity Master Mix (New England Biolabs) following the manufacturer's protocol, except for an initial 4 minute 95 °C incubation. Gel electrophoresis was conducted on the resulting nanobody product using a 1.6% agarose gel, and the nanobody band was gel extracted (Qiagen). Amplicon-EZ NGS sequencing was then conducted by Genewiz.

C.1.3 RECOMBINATION NANOBODY EXPRESSION AND PURIFICATION

Parent nanobodies, and nanobodies possessing algorithm predicted substitutions introduced by QuikChange Lightning mutagenesis kit (Agilent), were cloned into pET26b bearing a C-terminal v5 tag and hexahistidine tag and expressed in *Escherichia coli* BL21(DE3) bacteria (Harvey et al 2022). *E. coli* transformed with plasmids were cultured in Terrific Broth (RPI) containing 4% glycerol and 50 mg/mL kanamycin until they reached an OD₆₀₀ of 1.5, were cooled for one hour to 17 °C, and protein expression was induced by 0.2 mM isopropyl B-D-1-thiogalactopyranoside (IPTG, Gold Biotechnology) overnight at 17 °C. Following centrifugation at 4000 RPM, bacterial pellets were resuspended in room temperature SET buffer (200 mM Tris pH 8, 500 mM sucrose, 500 mM EDTA) and mixed for 20 minutes. Afterwards, 2x volume cold water, 5 mM magnesium chloride, and 1 mL benzonase (Sigma Aldrich) were added and the mixture was stirred for 1 hour. Cellular debris was removed by centrifugation at 14,000 x g for 30 minutes, 100 mM sodium chloride was

incubated with the supernatant with stirring for 15 minutes, and then the supernatant was then filtered using glass microfiber filters (Fischer Scientific). Following clarification, the supernatant was passed over Protein A resin equilibrated with protein A wash buffer (10 mM sodium phosphate pH 7.5, 100 mM sodium chloride) and the resin was washed with 20 column volumes of protein A wash buffer. Nanobodies were then eluted with 10 mL of Protein A elution buffer (100 mM sodium phosphate pH 2.5, 100 mM sodium chloride) directly into a column volume of 2M HEPES pH 8. Eluted nanobodies were dialyzed overnight in 20 mM HEPES pH 7.5, 150 mM NaCl, 10% glycerol. Nanobody purity was assessed by SDS-PAGE and Coomassie staining.

C.1.4 NGS PROCESSING

The paired-end sequencing raw data was QCed and filtered using FastQC and Trimmomatic (Bolger, Lohse, and Usadel 2014). Paired ends were joined using fastq-join. Sequences that do not match the primers used during sequencing were discarded. Nucleotide sequences were translated to amino acid sequences using biopython (Cock et al. 2009). Any protein sequences that did not match (allowing for up to 3 mismatches) the known beginning and end of the initial lead sequence were discarded. Lastly, any sequences with major deletions (>10 AAs) with respect to the initial lead sequence were removed.

C.1.5 ENRICHMENT LABEL CONSTRUCTION

Low count sequences were reasoned to be the result of sequencing noise and thus sequences with a count less than 5 were removed from each sequencing round. As the abundance of a sequence in any given round may be inflated due to expression biases of each sequence, we normalized the abundance of a sequence in each FACS round by the sequence abundance of the MACS round and took the natural log of this to construct the enrichment value. If a sequence was not found in both

rounds (which was the case for most sequences), its log enrichment value was imputed to 1×10^{-6} . In addition to the continuous enrichment value, we also binarized the label and set all sequences with enrichment > 0 as “enriched” and with enrichment < 0 as “depleted”.

C.1.6 SEQUENCE PREPARATION

Sequences were aligned using ANARCI (Dunbar and Deane 2016) using IMGT numbering. Only the IMGT columns present in the original lead nanobody were kept as any indels in our sequences probably arose from sequencing errors as our ePCR libraries did not introduce indel variation. The sequences were one-hot-encoded with gaps included (so each site is represented by a 21 length vector). As the elements representing the amino acids for the original lead (WT) are highly correlated in the input, we removed these elements when training logistic/linear regression models to reduce correlation between parameters. In this way each element in the one hot encoding represented a possible mutation from WT.

C.1.7 MACHINE LEARNING MODEL TRAINING

A variety of machine learning models were benchmarked for this study. Logistic regression, linear regression, CNNs and the top model for the ESM2-MLP semi-supervised model were all trained with L1 regularization the respective loss functions, with weights of 2, 0.0005, 0.0001 and 0.0001 respectively. Neural Network models were trained for 1000 epochs with early stopping. Linear and logistic regression models were built using the implementation in scikit-learn. Neural Network models were built using pytorch. All models were trained on CPU. For the ESM2-MLP model, the ESM2 encodings were retrieved using the pre-trained ESM2_t33_650M_UR50D model using the esm codebase (<https://github.com/facebookresearch/esm>). The polyreactivity model (Harvey et al. 2022) from was implemented using the codebase from the paper (<https://github.com/>

debbiemarkslab/nanobody-polyreactivity).

C.I.8 NDCG CALCULATION

Normalized discounted cumulative gains (NDCG) is a metric used to measure the quality of a model's ranking with respect to true scores of the data points (the gains). These scorings can be a true ranking or a binary label. We assigned a gain of 1 to each single substitution in our validated set and a 0 to all others. The NDCG is calculated as:

$$NDCG = \frac{DCG}{IDCG}$$

Where the DCG is the discounted cumulative gains (or the score given the rankings from the model) and the IDCG is the ideal DCG (the score given by the ideal ranking in which data points with the highest gains are at the top of the ranking). These are calculated as such:

$$DCG = \sum_i^N \frac{gains_i}{\log_2(rank_i)}$$

$$IDCG = \sum_i^N \frac{gains_i}{\log_2(ideal_rank_i)}$$

Where the sum is over all single substitutions of the initial lead.

C.2 SUPPLEMENTARY FIGURES

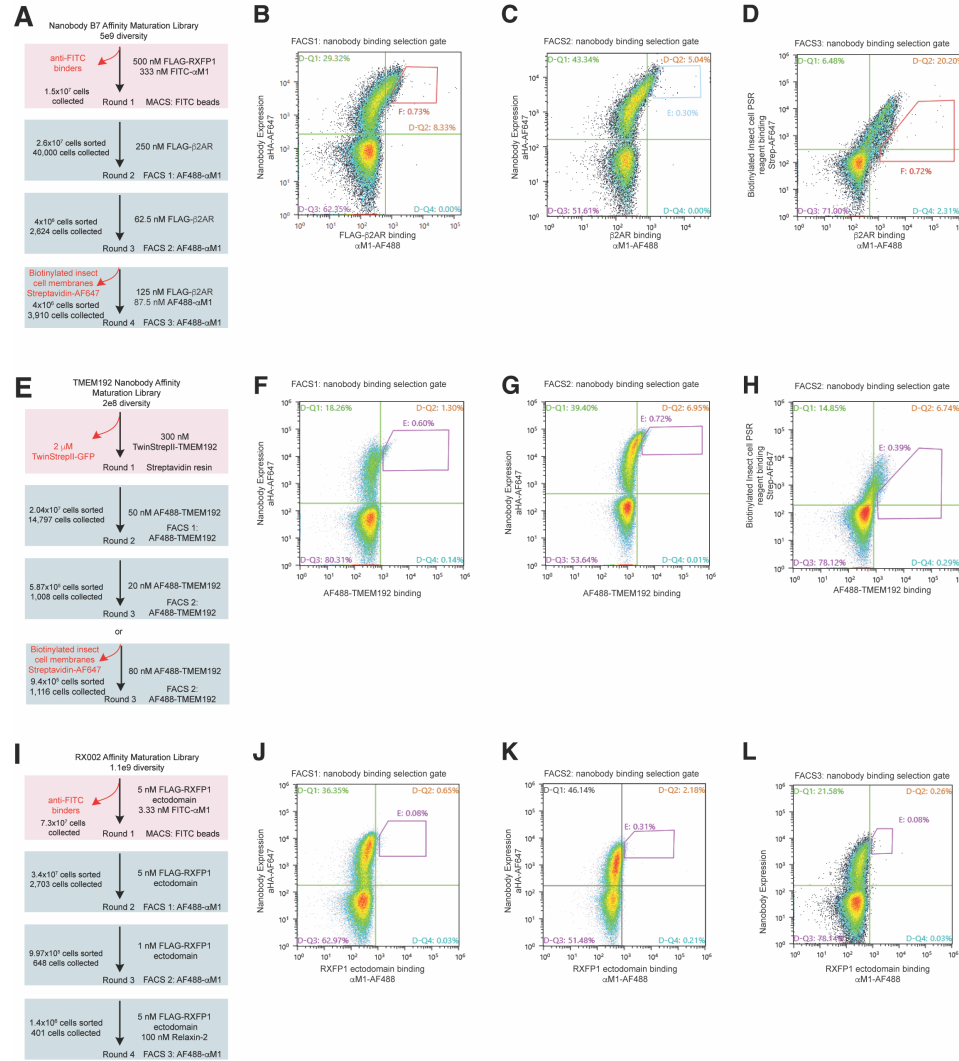


Figure C1: Affinity maturation of nanobodies that bind b2AR, TMEM192, and RXFP1. (A-D) Flowchart and Fluorescence-activated cell sorting (FACS) plots for nanobody B7 affinity maturation to improve binding to b2AR. (E-H) Flowchart and FACS plots for TMEM192 nanobody affinity maturation. (I-L) Flowchart and FACS plots for RX002 nanobody affinity maturation to improve binding to RXFP1. Note: the FACS plots for AT110 can be found in Wingler et al.²⁷⁹.

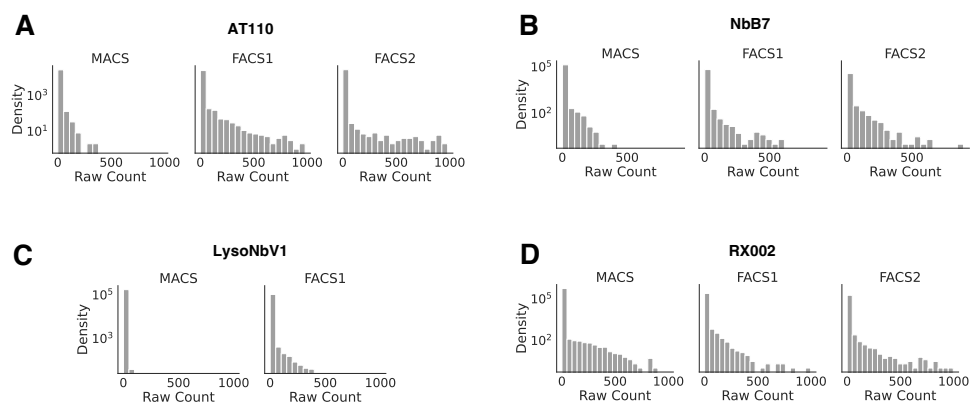


Figure C2: Count distributions of the sequencing rounds. Histograms showing the count distributions of sequences after each sort of the affinity maturation campaigns for A) AT110, B) NbB7 and C) LysoNbV1. The histograms are clipped at 1000 counts to clearly show the increasing skew of the distributions as further sorts are performed.

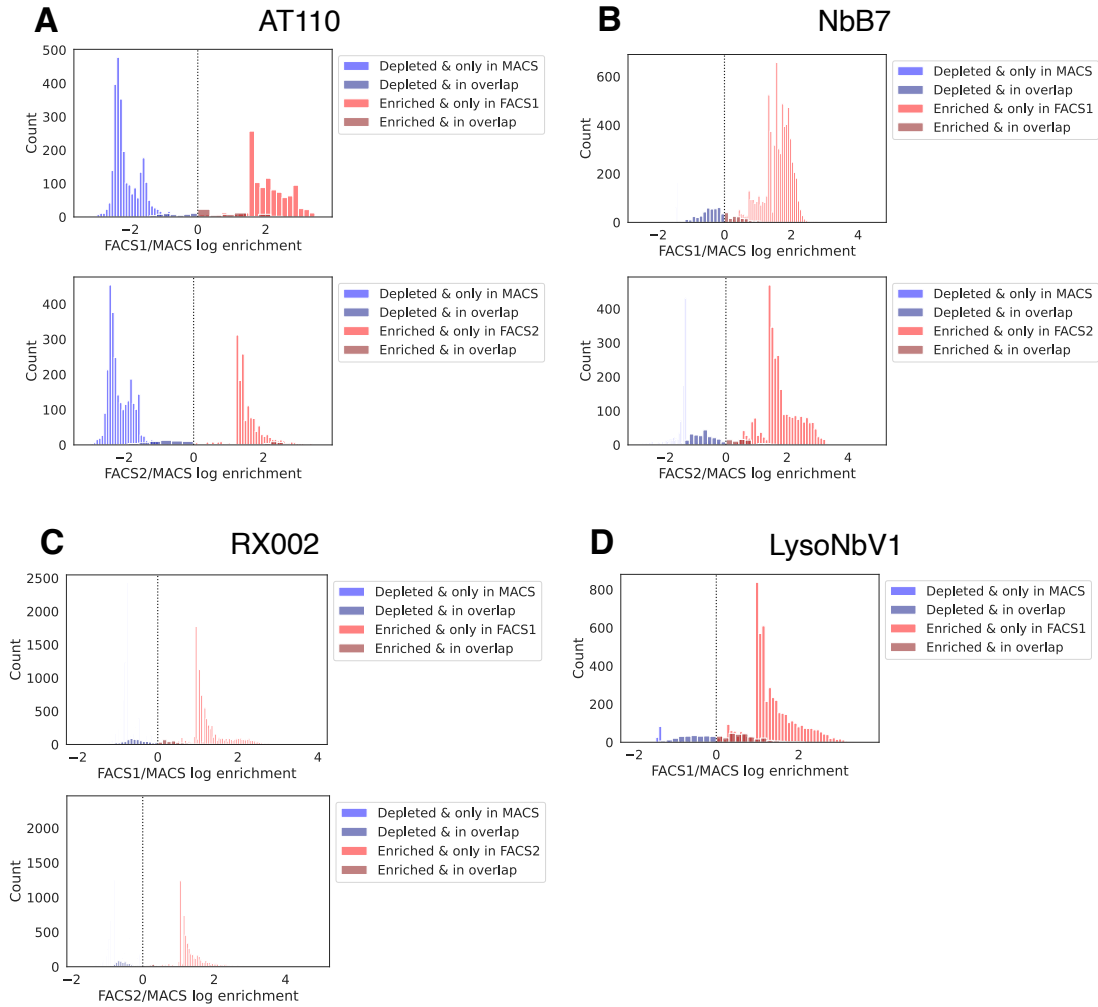


Figure C3: Enrichment distributions of the FACS rounds. Histograms showing the distribution of log-enrichment values after each FACS sort with respect to the MACS sort of the affinity maturation campaigns for A) AT110, B) NbB7 and C) LysoNbV1. The distribution of sequences that are enriched and depleted after FACS are colored in red and blue respectively. Darker bars correspond to the sequences found in both rounds used to calculate the enrichment score. Lighter bars correspond to sequences only in a single round, and thus their score is determined by the sequencing count of the round they were found in and the floor abundance value used (set to 1×10^{-6}).

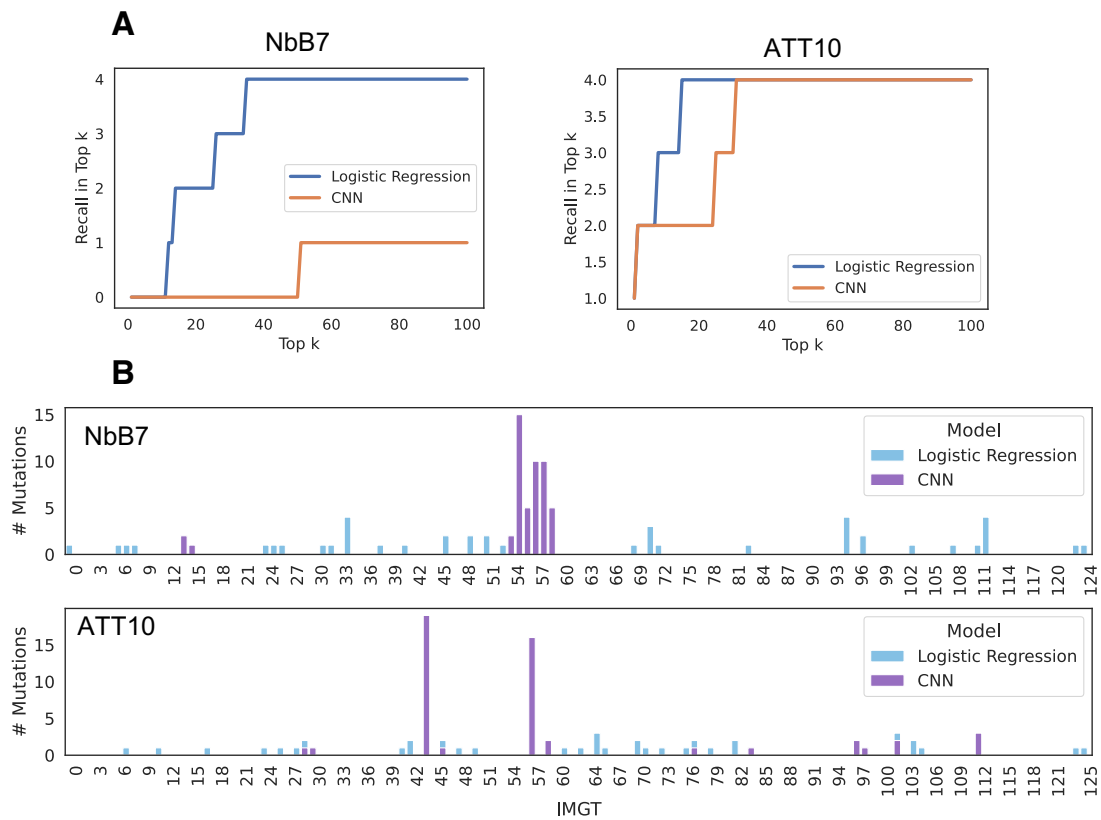


Figure C4: Comparison of top singles from a logistic regression model and CNN. A) The recall of the 4 tested mutants available for NbB7 and ATT10 in the top k ranked single mutations from the Logistic Regression and CNN models for increasing k up to 100. B) The distribution of the IMGT sites of the top 50 mutations predicted by both models for NbB7 and ATT10.

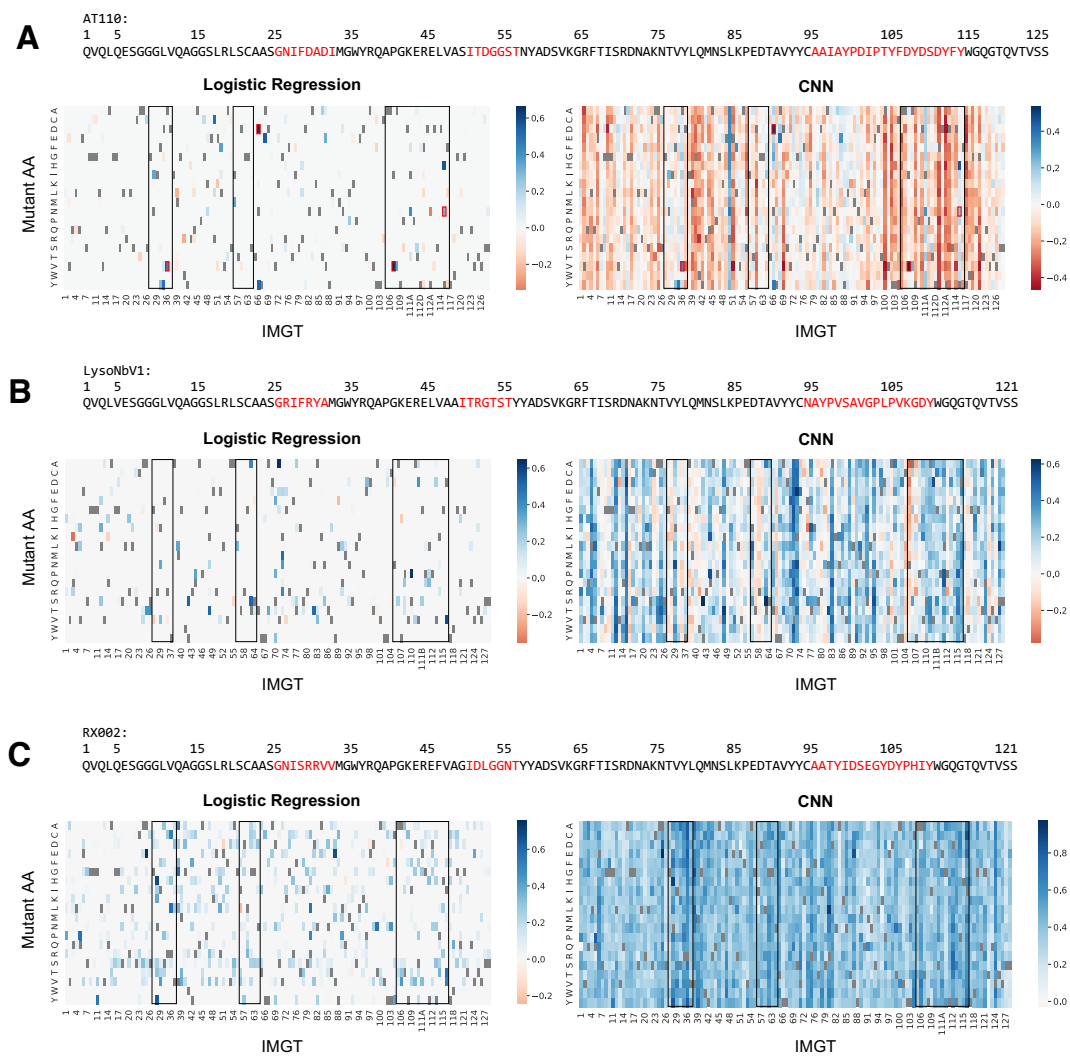


Figure C5: Heatmaps of scores for all single substitutions. Sequence and heatmaps of logistic regression and CNN scores for all single substitutions for A) AT110, B) LysoNbV1, and C) RX002. Red text on the sequence and black boxes on heatmaps correspond to CDRs 1-3. Indices above the sequences are according to residue position in the sequence.

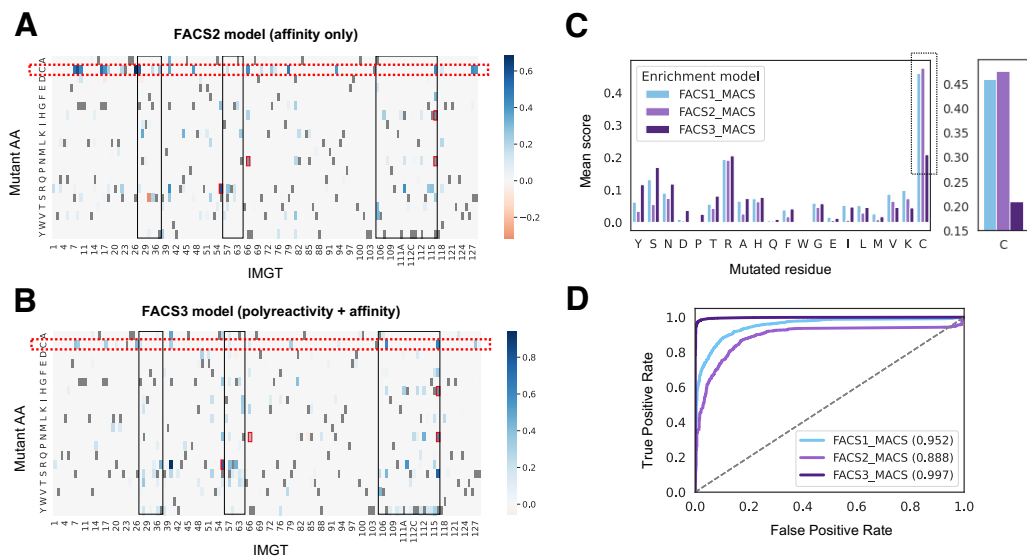


Figure C6: Analysis of polyreactivity counter-sort. Heatmaps of scores for all single substitutions from logistic regression models trained on the A) FACS2/MACS enrichment and B) FACS3/MACAS enrichment for NbB7. The FACS2 sort only selected for affinity to B2AR. The FACS3 sort was a simultaneous affinity selection and polyreactivity counterselection. The scores for mutations to cysteine are highlighted with a dotted red box. Black boxes correspond to CDRs 1-3 and red boxes correspond to experimentally validated substitutions. C) (left) The average score per amino acid from the logistic regression models trained on the NbB7 FACS1, FACS2 and FACS3 enrichment over MACS. (right) A zoomed in view of the scores for cysteine mutations. Zoomed in portion is highlighted in black dotted lined in left panel. D) ROC-plots showing the performance of the logistic regression models trained on each round for their prediction of FACS3/MACS enrichment. The ROC-AUC for each model is shown in brackets in the legend.

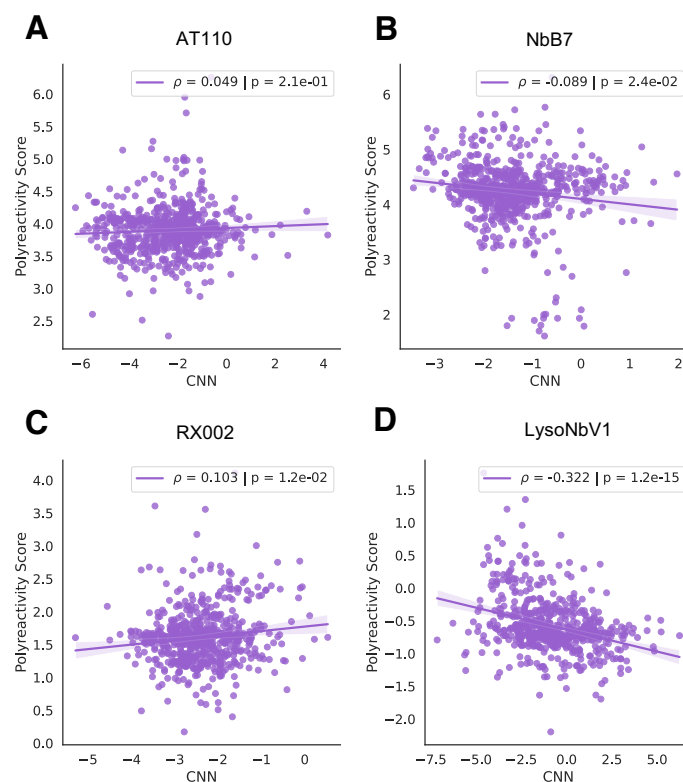


Figure C7: Affinity and Polyreactivity models have low correlation. Scatterplots of the scores from the pretrained polyreactivity model¹⁰⁰ against a CNN trained on the FACS1 enrichment for nanobodies A) AT110, B) NbB7, C) RX002 and D) LysoNbV1. The scores for all single substitutions of the CDRs of each nanobody are plotted. Line of best fit with 95% confidence intervals are plotted and the spearman correlation and p-value for the significance of the correlation are shown in the inset of each plot. Higher Polyreactivity Scores correspond to higher specificity of the nanobody (i.e lower scores mean more polyreactive).

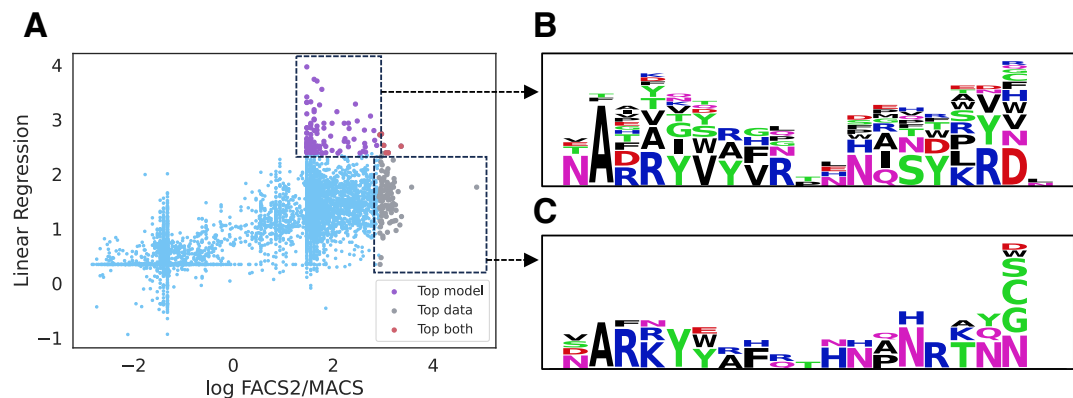


Figure C8: Affinity models suggest different top sequences from the experiment. A) Scatterplot of scores from a linear regression model compared to the continuous FACS2/MACS enrichment score for sequences in the NbB7 campaign. The top 100 sequences by model score and FACS2 enrichment are colored in purple and gray respectively. Sequences in both sets are colored red. B-C) Sequence logos of the CDR3 region for the top 100 sequences under the model score (B) and enrichment from the experiment (C). Logos are plotted by calculating the amino acid frequencies for each site in the CDR3 (excluding gaps). The amino acid corresponding to the WT residue at that site is also omitted from the logo. Thus, the height of each residue corresponds to the relative probability of a substitution to that amino acid, and the height of the whole column corresponds to the probability of any substitution at that site.

C.3 SUPPLEMENTARY TABLES

Nanobody	Round	# Total Reads	# Unique Seq	# Seq above threshold
AT ₁₁₀	MACS	118,429	26,516	2,805
	FACS ₁	148,783	24,648	1,172
	FACS ₂	278,191	27,053	1,539
NbB ₇	MACS	241,225	134,340	1,433
	FACS ₁	230,676	66,641	7,160
	FACS ₂	168,987	37,344	3,131
	FACS ₃	1,521,316	520,440	42,012
LysoNbV ₁	MACS	223,217	174,087	717
	FACS ₁	532,207	105,598	5,542
RX ₀₀₂	MACS	922,141	555,762	8,543
	FACS ₁	554,583	246,068	9,331
	FACS ₂	430,340	180,670	6,126

Table C1: Sequencing dataset sizes Summary count statistics of the sequencing from the affinity maturation campaigns used in this study. We use a threshold of 5 counts for retaining sequences for training machine learning models.

Nanobody	R_0/R_1	$R_0 \cap R_1$	%	$R_0^* \cap R_1^*$	%	$R_0^* \cup R_1^*$	(#depleted, #enriched)
AT110	FACS1/MACS	1173	2.35%	111	2.87%	3866	(2747,1119)
	FACS2/MACS	1166	2.23%	83	1.95%	4261	(2783,1478)
NbB7	FACS1/MACS	5853	3.0%	561	6.98%	8032	(1280,6752)
	FACS2/MACS	2902	1.72%	284	6.64%	4280	(1365,2915)
LysoNbV1	FACS1/MACS	4208	1.53%	515	8.97%	5744	(458,5286)
RX002	FACS1/MACS	12922	1.64%	1016	6.03%	16858	(8154,8704)
	FACS2/MACS	12911	1.78%	798	5.75%	13871	(8202,5669)

Table C2: Overlap statistics and train set sizes R_0 and R_1 describe which two rounds we are considering the overlap between. $R_0 \cup R_1$ is the number of overlapping unique sequences found in both rounds with the % reflecting what this overlap is as a proportion of the union of the sets. R_0 and R_1 describe the sequencing sets after filtering by removing low count sequences. $R_0 \cap R_1$ is the total number of sequences that pass the filtering threshold in either of the rounds. This is the training data size for that round. The number of sequences of each label in each training set is shown in the last column.

Count Threshold	ROC-AUC	NDCG@100
1	0.9253	0.812
2	0.9417	0.855
3	0.9466	0.852
4	0.9507	0.848
5	0.9518	0.857

Table C3: Effect of count threshold on model performance The ROC-AUC of FACS2/MACS enrichment prediction and the NDCG@100 for ranking known optimizing mutations for varying values of the count threshold used to filter out low count sequences. Results here are for Logistic Regression models trained on the FACS1/MACS enrichment from the AT110 campaign. Higher count thresholds increase the performance of the model, likely due to increasing data quality of the training data.

References

- [1] Aakre, C., Herrou, J., Phung, T., Perchuk, B., Crosson, S., & Laub, M. (2015). Evolving New Protein-Protein Interaction Specificity through Promiscuous Intermediates. *Cell*, 163(3), 594–606.
- [2] Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Židek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., & Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*.
- [3] Abul K. Abbas, Andrew H. Lichtman, and Shiv Pillai (2022). *Cellular and Molecular Immunology*, volume 10. Elsevier.
- [4] Adkar, B., Tripathi, A., Sahoo, A., Bajaj, K., Goswami, D., Chakrabarti, P., Swarnkar, M., Gokhale, R., & Varadarajan, R. (2012). Protein Model Discrimination Using Mutational Sensitivity Derived from Deep Sequencing. *Structure*, 20(2), 371–381.
- [5] Adolf-Bryfogle, J., Xu, Q., North, B., Lehmann, A., & Dunbrack, Jr, R. L. (2015). PyIgClassify: a database of antibody CDR structural classifications. *Nucleic Acids Res.*, 43(Database issue), D432–8.
- [6] Aguilar Rangel, M., Bedwell, A., Costanzi, E., Taylor, R. J., Russo, R., Bernardes, G. J. L., Ricagno, S., Frydman, J., Vendruscolo, M., & Sormanni, P. (2022). Fragment-based computational design of antibodies targeting structured epitopes. *Sci Adv*, 8(45), eabp9540.
- [7] Ahler, E., Register, A. C., Chakraborty, S., Fang, L., Dieter, E. M., Sitko, K. A., Vidadala, R. S. R., Trevillian, B. M., Golkowski, M., Gelman, H., Stephany, J. J., Rubin, A. F., Merritt, E. A., Fowler, D. M., & Maly, D. J. (2019). A Combined Approach Reveals a Regulatory Mechanism Coupling Src's Kinase Activity, Localization, and Phosphotransferase-Independent Functions. *Molecular Cell*, 74(2), 393–408.e20.

- [8] Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., & Church, G. M. (2019). Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, 16(12), 1315–1322.
- [9] Amin, A., Weinstein, E. N., & Marks, D. (2021). A generative nonparametric bayesian model for whole genomes. *NeurIPS*, 34, 27798–27812.
- [10] Amin, A. N., Weinstein, E. N., & Marks, D. S. (2023a). Biological sequence kernels with guaranteed flexibility. *arXiv preprint arXiv:2304.03775*.
- [11] Amin, A. N., Weinstein, E. N., & Marks, D. S. (2023b). A kernelized stein discrepancy for biological sequences. In *Proceedings of the 40th ICML*.
- [12] Amorosi, C. J., Chiasson, M. A., McDonald, M. G., Wong, L. H., Sitko, K. A., Boyle, G., Kowalski, J. P., Rettie, A. E., Fowler, D. M., & Dunham, M. J. (2021). Massively parallel characterization of CYP2C9 variant enzyme activity and abundance. *The American Journal of Human Genetics*, 108(9), 1735–1751.
- [13] Andrews, B. & Fields, S. (2020). Distinct patterns of mutational sensitivity for λ resistance and maltodextrin transport in escherichia coli LamB. *Microbial Genomics*, 6(4).
- [14] Anfinsen, C. & Scheraga, H. (1975). Experimental and theoretical aspects of protein folding. volume 29 of *Advances in Protein Chemistry* (pp. 205–300). Academic Press.
- [15] Araya, C. L., Fowler, D. M., Chen, W., Muniez, I., Kelly, J. W., & Fields, S. (2012). A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function. *Proceedings of the National Academy of Sciences*, 109(42), 16858–16863.
- [16] Arcones, M. A. & Giné, E. (1992). On the bootstrap of U and V statistics. *The Annals of Statistics*.
- [17] Arnold, F. H. (2018). Directed evolution: Bringing new chemistry to life. *Angew. Chem. Int. Ed Engl.*, 57(16), 4143–4148.
- [18] Bandaru, P., Shah, N. H., Bhattacharyya, M., Barton, J. P., Kondo, Y., Cofsky, J. C., Gee, C. L., Chakraborty, A. K., Kortemme, T., Ranganathan, R., & Kuriyan, J. (2017). Deconstruction of the Ras switching cycle through saturation mutagenesis. *eLife*, 6, e27810.
- [19] Baum, J., Kanagawa, H., & Gretton, A. (2023). A kernel stein test of goodness of fit for sequential models. In *ICML*.
- [20] Bennett, N. R., Coventry, B., Goreshnik, I., Huang, B., Allen, A., Vafeados, D., Peng, Y. P., Dauparas, J., Baek, M., Stewart, L., DiMaio, F., De Munck, S., Savvides, S. N., & Baker, D. (2023). Improving de novo protein binder design with deep learning. *Nat. Commun.*, 14(1), 2625.

- [21] Bennett, N. R., Watson, J. L., Ragotte, R. J., Borst, A. J., See, D. L., Weidle, C., Biswas, R., Shrock, E. L., Leung, P. J. Y., Huang, B., Goreshnik, I., Ault, R., Carr, K. D., Singer, B., Criswell, C., Vafeados, D., Sanchez, M. G., Kim, H. M., Torres, S. V., Chan, S., & Baker, D. (2024). Atomically accurate de novo design of single-domain antibodies.
- [22] Berlinet, A. & Thomas-Agnan, C. (2011). *Reproducing kernel Hilbert spaces in probability and statistics*.
- [23] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., & Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Res.*, 28(1), 235–242.
- [24] Blazejewski, T., Ho, H.-I., & Wang, H. H. (2019). Synthetic sequence entanglement augments stability and containment of genetic information in cells. *Science*, 365(6453), 595–598.
- [25] Bloom, J. D. (2014). An experimentally determined evolutionary model dramatically improves phylogenetic fit. *Molecular Biology and Evolution*, 31(8), 1956–1978.
- [26] Boder, E. T. & Wittrup, K. D. (1997). Yeast surface display for screening combinatorial polypeptide libraries. *Nat. Biotechnol.*, 15(6), 553–557.
- [27] Bolognesi, B., Faure, A. J., Seuma, M., Schmiedel, J. M., Tartaglia, G. G., & Lehner, B. (2019). The mutational landscape of a prion-like domain. *Nature Communications*, 10(1), 4162.
- [28] Boucher, J. I., Bolon, D. N., & Tawfik, D. S. (2016). Quantifying and understanding the fitness effects of protein mutations: Laboratory versus nature. *Protein Science*, 25(7), 1219–1226.
- [29] Brandes, N., Goldman, G., Wang, C. H., Ye, C. J., & Ntranos, V. (2023). Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, 55(9), 1512–1522.
- [30] Brenan, L., Andreev, A., Cohen, O., Pantel, S., Kamburov, A., Cacchiarelli, D., Persky, N., Zhu, C., Bagul, M., Goetz, E., Burgin, A., Garraway, L., Getz, G., Mikkelsen, T., Piccioni, F., Root, D., & Johannessen, C. (2016). Phenotypic Characterization of a Comprehensive Set of MAPK1 /ERK2 Missense Mutants. *Cell Reports*, 17(4), 1171–1183.
- [31] Bridgford, J. L., Lee, S. M., Lee, C. M. M., Guglielmelli, P., Rumi, E., Pietra, D., Wilcox, S., Chhabra, Y., Rubin, A. F., Cazzola, M., Vannucchi, A. M., Brooks, A. J., Call, M. E., & Call, M. J. (2020). Novel drivers and modifiers of MPL-dependent oncogenic transformation identified by deep mutational scanning. *Blood*, 135(4), 287–292.
- [32] Bröcker, J. (2008). Some remarks on the reliability of categorical probability forecasts. *Monthly Weather Review*.

- [33] Cao, L., Coventry, B., Goreschnik, I., Huang, B., Park, J. S., Jude, K. M., Marković, I., Kadam, R. U., Verschueren, K. H. G., Verstraete, K., Walsh, S. T. R., Bennett, N., Phal, A., Yang, A., Kozodoy, L., DeWitt, M., Picton, L., Miller, L., Strauch, E.-M., DeBouver, N. D., Pires, A., Bera, A. K., Halabiya, S., Hammerson, B., Yang, W., Bernard, S., Stewart, L., Wilson, I. A., Ruohola-Baker, H., Schlessinger, J., Lee, S., Savvides, S. N., Garcia, K. C., & Baker, D. (2022). Design of protein binding proteins from target structure alone. *Nature*.
- [34] Capel, H., Weiler, R., Dijkstra, M., Vleugels, R., Bloem, P., & Feenstra, K. A. (2022). ProteinGLUE multi-task benchmark suite for self-supervised protein modeling. *Scientific Reports*, 12(1), 16047. Number: 1 Publisher: Nature Publishing Group.
- [35] Carmeli, C., De Vito, E., Toigo, A., & Umanitá, V. (2010). Vector valued reproducing kernel Hilbert spaces and universality. *Analysis and Applications*.
- [36] Chakraborty, S., Ahler, E., Simon, J. J., Fang, L., Potter, Z. E., Sitko, K. A., Stephany, J. J., Guttman, M., Fowler, D. M., & Maly, D. J. (2021). Profiling of the drug resistance of thousands of src tyrosine kinase mutants uncovers a regulatory network that couples autoinhibition to catalytic domain dynamics.
- [37] Chan, K. K., Dorosky, D., Sharma, P., Abbasi, S. A., Dye, J. M., Kranz, D. M., Herbert, A. S., & Procko, E. (2020). Engineering human ACE2 to optimize binding to the spike protein of SARS coronavirus 2. *Science*, 369(6508), 1261–1265.
- [38] Chan, Y. H., Venev, S. V., Zeldovich, K. B., & Matthews, C. R. (2017). Correlation of fitness landscapes from three orthologous TIM barrels originates from sequence and structure constraints. *Nature Communications*, 8(1), 14614.
- [39] Chen, J. Z., Fowler, D. M., & Tokuriki, N. (2020a). Comprehensive exploration of the translocation, stability and substrate recognition requirements in VIM-2 lactamase. *eLife*, 9, e56707.
- [40] Chen, T., Gong, C., Diaz, D. J., Chen, X., Wells, J. T., Liu, Q., Wang, Z., Ellington, A., Dimakis, A., & Klivans, A. (2022). HotProtein: A Novel Framework for Protein Thermostability Prediction and Editing.
- [41] Chen, T.-Y., Li, X., Hung, C.-H., Bahudhanapati, H., Tan, J., Kass, D. J., & Zhang, Y. (2020b). The relaxin family peptide receptor 1 (RXFP1): An emerging player in human health and disease. *Mol. Genet. Genomic Med.*, 8(4), e1194.
- [42] Chen, Y., Hu, R., Li, K., Zhang, Y., Fu, L., Zhang, J., & Si, T. (2023). Deep mutational scanning of an Oxygen-Independent fluorescent protein CreiLOV for comprehensive profiling of mutational and epistatic effects. *ACS Synthetic Biology*, 12(5), 1461–1473.
- [43] Cherf, G. M. & Cochran, J. R. (2015). Applications of yeast surface display for protein engineering. *Methods Mol. Biol.*, 1319, 155–175.

- [44] Chiasson, M. A., Rollins, N. J., Stephany, J. J., Sitko, K. A., Matreyek, K. A., Verby, M., Sun, S., Roth, F. P., DeSloover, D., Marks, D. S., Rettie, A. E., & Fowler, D. M. (2020). Multiplexed measurement of variant abundance and activity reveals VKOR topology, active site and human variant impact. *eLife*, 9, e58026.
- [45] Christmann, A. & Steinwart, I. (2010). Universal kernels on Non-Standard input spaces. In *NeurIPS*.
- [46] Chwialkowski, K., Strathmann, H., & Gretton, A. (2016). A kernel test of goodness of fit. In *ICML*.
- [47] Clausen, L., Voutsinos, V., Cagiada, M., Johansson, K. E., Grønbæk-Thygesen, M., Nariya, S., Powell, R. L., Have, M. K. N., Oestergaard, V. H., Stein, A., Fowler, D. M., Lindorff-Larsen, K., & Hartmann-Petersen, R. (2023). A mutational atlas for parkin proteostasis.
- [48] Coyote-Maestas, W., Nedrud, D., He, Y., & Schmidt, D. (2022). Determinants of trafficking, conduction, and disease within a K⁺ channel revealed through multiparametric deep mutational scanning. *eLife*, 11, e76903.
- [49] Cuturi, M. & Blondel, M. (2017). Soft-dtw: a differentiable loss function for time-series. In *ICML*.
- [50] Cuturi, M., Vert, J.-P., Birkenes, O., & Matsui, T. (2007). A kernel for time series based on global alignments. In *ICASSP*.
- [51] Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., & Yang, K. K. (2021). FLIP: Benchmark tasks in fitness landscape inference for proteins.
- [52] Dandage, R., Pandey, R., Jayaraj, G., Rai, M., Berger, D., & Chakraborty, K. (2018). Differential strengths of molecular determinants guide environment specific mutational fates. *PLOS Genetics*, 14(5), e1007419.
- [53] Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. London: Murray. or the Preservation of Favored Races in the Struggle for Life.
- [54] Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N., Leung, P. J. Y., Huddy, T. F., Pellock, S., Tischer, D., Chan, F., Koepnick, B., Nguyen, H., Kang, A., Sankaran, B., Bera, A. K., King, N. P., & Baker, D. (2022). Robust deep learning-based protein sequence design using ProteinMPNN. *Science*, 378(6615), 49–56.
- [55] Dawson, N. L., Lewis, T. E., Das, S., Lees, J. G., Lee, D. A., Ashford, P., Orengo, C. A., & Sillitoe, I. P. W. (2016). Cath: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Research*, 45, D289 – D295.

- [56] Deng, Z., Huang, W., Bakkalbasi, E., Brown, N. G., Adamski, C. J., Rice, K., Muzny, D., Gibbs, R. A., & Palzkill, T. (2012). Deep Sequencing of Systematic Combinatorial Libraries Reveals beta-Lactamase Sequence Constraints at High Resolution. *Journal of Molecular Biology*, 424(3-4), 150–167.
- [57] Desai, A. A., Smith, M. D., Zhang, Y., Makowski, E. K., Gerson, J. E., Ionescu, E., Starr, C. G., Zupancic, J. M., Moore, S. J., Sutter, A. B., Ivanova, M. I., Murphy, G. G., Paulson, H. L., & Tessier, P. M. (2021). Rational affinity maturation of anti-amyloid antibodies with high conformational and sequence specificity. *J. Biol. Chem.*, 296, 100508.
- [58] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.
- [59] Dinculeanu, N. (2000). *Vector integration and stochastic integration in Banach spaces*. John Wiley & Sons.
- [60] Ding, D., Shaw, A., Sinai, S., Rollins, N., Prywes, N., Savage, D. F., Laub, M. T., & Marks, D. S. (2023). Protein design using structure-based residue preferences.
- [61] Domingo-Enrich, C., Dwivedi, R., & Mackey, L. (2023). Compress then test: Powerful kernel testing in near-linear time. *AISTATS*.
- [62] Doud, M. & Bloom, J. (2016). Accurate Measurement of the Effects of All Amino-Acid Mutations on Influenza Hemagglutinin. *Viruses*, 8(6), 155.
- [63] Doud, M. B., Ashenberg, O., & Bloom, J. D. (2015). Site-Specific Amino Acid Preferences Are Mostly Conserved in Two Closely Related Protein Homologs. *Molecular Biology and Evolution*, 32(11), 2944–2960.
- [64] Duenas-Decamp, M., Jiang, L., Bolon, D., & Clapham, P. R. (2016). Saturation Mutagenesis of the HIV-1 Envelope CD4 Binding Loop Reveals Residues Controlling Distinct Trimer Conformations. *PLOS Pathogens*, 12(11), e1005988.
- [65] Dunbar, J. & Deane, C. M. (2016). ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2), 298–300.
- [66] Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., & Deane, C. M. (2014). SABDab: the structural antibody database. *Nucleic Acids Res.*, 42(Database issue), D1140–6.
- [67] Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.
- [68] Eddy, S. R. (2011). Accelerated profile HMM searches. *PLoS Comput. Biol.*, 7(10), e1002195.

- [69] Eguchi, R. R., Choe, C. A., Parekh, U., Khalek, I. S., Ward, M. D., Vithani, N., Bowman, G. R., Jardine, J. G., & Huang, P.-S. (2022). Deep generative design of epitope-specific binding proteins by latent conformation optimization. *bioRxiv*, (pp. 2022.12.22.521698).
- [70] Elazar, A., Weinstein, J., Biran, I., Fridman, Y., Bibi, E., & Fleishman, S. J. (2016). Mutational scanning reveals the determinants of protein insertion and association energetics in the plasma membrane. *eLife*, 5, e12125.
- [71] Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Yu, W., Jones, L., Gibbs, T., Fehér, T. B., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2021). Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *IEEE transactions on pattern analysis and machine intelligence*, PP.
- [72] Engelen, S., Trojan, L. A., Sacquin-Mora, S., Lavery, R., & Carbone, A. (2009). Joint evolutionary trees: a large-scale method to predict protein interfaces based on sequence sampling. *PLoS computational biology*, 5(1), e1000267.
- [73] Erwood, S., Bily, T. M. I., Lequyer, J., Yan, J., Gulati, N., Brewer, R. A., Zhou, L., Pelletier, L., Ivakine, E. A., & Cohn, R. D. (2022). Saturation variant interpretation using CRISPR prime editing. *Nature Biotechnology*, 40(6), 885–895.
- [74] Estevam, G. O., Linossi, E. M., Macdonald, C. B., Espinoza, C. A., Michaud, J. M., Coyote-Maestas, W., Collisson, E. A., Jura, N., & Fraser, J. S. (2023). *Conserved regulatory motifs in the juxtamembrane domain and kinase N-lobe revealed through deep mutational scanning of the MET receptor tyrosine kinase domain*. preprint, Molecular Biology.
- [75] Faure, A. J., Domingo, J., Schmiedel, J. M., Hidalgo-Carcedo, C., Diss, G., & Lehner, B. (2022). Mapping the energetic and allosteric landscapes of protein binding domains. *Nature*, 604(7904), 175–183.
- [76] Fernandes, J. D., Faust, T. B., Strauli, N. B., Smith, C., Crosby, D. C., Nakamura, R. L., Hernandez, R. D., & Frankel, A. D. (2016). Functional Segregation of Overlapping Genes in HIV. *Cell*, 167(7), 1762–1773.e12.
- [77] Ferruz, N., Schmidt, S., & Höcker, B. (2022). ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13.
- [78] Findlay, G. M., Daza, R. M., Martin, B., Zhang, M. D., Leith, A. P., Gasperini, M., Janizek, J. D., Huang, X., Starita, L. M., & Shendure, J. (2018). Accurate classification of BRCA1 variants with saturation genome editing. *Nature*, 562(7726), 217–222.
- [79] Firnberg, E., Labonte, J. W., Gray, J. J., & Ostermeier, M. (2014). A Comprehensive, High-Resolution Map of a Gene's Fitness Landscape. *Molecular Biology and Evolution*, 31(6), 1581–1592.

- [80] Flynn, J. M., Rossouw, A., Cote-Hammarlof, P., Fragata, I., Mavor, D., Hollins, C., Bank, C., & Bolon, D. N. (2020). Comprehensive fitness maps of Hsp90 show widespread environmental dependence. *eLife*, 9, e53810.
- [81] Flynn, J. M., Samant, N., Schneider-Nachum, G., Barkan, D. T., Yilmaz, N. K., Schiffer, C. A., Moquin, S. A., Dovala, D., & Bolon, D. N. (2022). *Comprehensive fitness landscape of SARS-CoV-2 M^{pro} reveals insights into viral resistance mechanisms*. preprint, Molecular Biology.
- [82] Fram, B., Truebridge, I., Su, Y., Riesselman, A. J., Ingraham, J. B., Passera, A., Napier, E., Thadani, N. N., Lim, S., Roberts, K., Kaur, G., Stiffler, M., Marks, D. S., Bahl, C. D., Khan, A. R., Sander, C., & Gauthier, N. P. (2023). Simultaneous enhancement of multiple functional properties using evolution-informed protein design. *bioRxiv*, (pp. 2023.05.09.539914).
- [83] Frazer, J., Notin, P., Dias, M., Gomez, A., Min, J. K., Brock, K., Gal, Y., & Marks, D. S. (2021). Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883), 91–95.
- [84] Gajula, K. S., Huwe, P. J., Mo, C. Y., Crawford, D. J., Stivers, J. T., Radhakrishnan, R., & Kohli, R. M. (2014). High-throughput mutagenesis reveals functional determinants for DNA targeting by activation-induced deaminase. *Nucleic Acids Research*, 42(15), 9964–9975.
- [85] Gao, Z., Tan, C., Chacón, P., & Li, S. Z. (2022). PiFold: Toward effective and efficient protein inverse folding.
- [86] Gersing, S., Cagiada, M., Gebbia, M., Gjesing, A. P., Coté, A. G., Seesankar, G., Li, R., Tabet, D., Stein, A., Gloy, A. L., Hansen, T., Roth, F. P., Lindorff-Larsen, K., & Hartmann-Petersen, R. (2022). *A comprehensive map of human glucokinase variant activity*. preprint, Genetics.
- [87] Gersing, S., Schulze, T. K., Cagiada, M., Stein, A., Roth, F. P., Lindorff-Larsen, K., & Hartmann-Petersen, R. (2023). Characterizing glucokinase variant mechanisms using a multiplexed abundance assay. *bioRxiv*.
- [88] Ghose, D. A., Przydzial, K. E., Mahoney, E. M., Keating, A. E., & Laub, M. T. (2023). Marginal specificity in protein interactions constrains evolution of a paralogous family. *Proceedings of the National Academy of Sciences of the United States of America*, 120(18), e2221163120.
- [89] Giacomelli, A. O., Yang, X., Lintner, R. E., McFarland, J. M., Duby, M., Kim, J., Howard, T. P., Takeda, D. Y., Ly, S. H., Kim, E., Gannon, H. S., Hurhula, B., Sharpe, T., Goodale, A., Fritchman, B., Steelman, S., Vazquez, F., Tsherniak, A., Aguirre, A. J., Doench, J. G.,

- Piccioni, F., Roberts, C. W. M., Meyerson, M., Getz, G., Johannessen, C. M., Root, D. E., & Hahn, W. C. (2018). Mutational processes shape the landscape of TP53 mutations in human cancer. *Nature Genetics*, 50(10), 1381–1387.
- [90] Gill, K. S., Mehta, K., Heredia, J. D., Krishnamurthy, V. V., Zhang, K., & Procko, E. (2023). Multiple mechanisms of self-association of chemokine receptors CXCR4 and CCR5 demonstrated by deep mutagenesis. *bioRxiv*.
- [91] Glaser, P., Widmann, D., Lindsten, F., & Gretton, A. (2023). Fast and scalable score-based kernel calibration tests. In *UAI*.
- [92] Glazer, A. M., Kroncke, B. M., Matreyek, K. A., Yang, T., Wada, Y., Shields, T., Salem, J.-E., Fowler, D. M., & Roden, D. M. (2020). Deep Mutational Scan of an SCN5A Voltage Sensor. *Circulation: Genomic and Precision Medicine*, 13(1), e002786.
- [93] Gonzalez Somermeyer, L., Fleiss, A., Mishin, A. S., Bozhanova, N. G., Igoalkina, A. A., Meiler, J., Alaball Pujol, M.-E., Putintseva, E. V., Sarkisyan, K. S., & Kondrashov, F. A. (2022). Heterogeneity of the GFP fitness landscape and data-driven protein design. *eLife*, 11, e75842.
- [94] Gorham, J. & Mackey, L. (2017). Measuring sample quality with kernels. In *ICML*.
- [95] Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., & Zemel, R. (2020). Learning the stein discrepancy for training and evaluating energy-based models without sampling. In *ICML*.
- [96] Gray, V. E., Sitko, K., Kameni, F. Z. N., Williamson, M., Stephany, J. J., Hasle, N., & Fowler, D. M. (2019). Elucidating the molecular determinants of A β aggregation with deep mutational scanning. *G3*, 9(11), 3683–3689.
- [97] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *JMLR*.
- [98] Haddox, H. K., Dingens, A. S., Hilton, S. K., Overbaugh, J., & Bloom, J. D. (2018). Mapping mutational effects along the evolutionary landscape of HIV envelope. *eLife*, 7, e34420.
- [99] Hamers-Casterman, C., Atarhouch, T., Muyldermans, S., Robinson, G., Hamers, C., Songa, E. B., Bendahman, N., & Hamers, R. (1993). Naturally occurring antibodies devoid of light chains. *Nature*, 363(6428), 446–448.
- [100] Harvey, E. P., Shin, J.-E., Skiba, M. A., Nemeth, G. R., Hurley, J., Wellner, A., Shaw, A. Y., Miranda, V. G., Min, J. K., Liu, C. C., Marks, D. S., & Kruse, A. C. (2022). An in silico method to assess antibody fragment polyreactivity. *Nat. Commun.*, 13(1), 7554.
- [101] Hesslow, D., ed. Zanichelli, N., Notin, P., Poli, I., & Marks, D. S. (2022). RITA: a study on scaling up generative protein sequence models. *ArXiv*, abs/2205.05789.

- [102] Hie, B. L., Shanker, V. R., Xu, D., Bruun, T. U. J., Weidenbacher, P. A., Tang, S., Wu, W., Pak, J. E., & Kim, P. S. (2023). Efficient evolution of human antibodies from general protein language models. *Nat. Biotechnol.*
- [103] Hietpas, R. T., Jensen, J. D., & Bolon, D. N. A. (2011). Experimental illumination of a fitness landscape. *Proceedings of the National Academy of Sciences of the United States of America*, 108(19), 7896–7901.
- [104] Ho, J., Kalchbrenner, N., Weissenborn, D., & Salimans, T. (2019). Axial attention in multi-dimensional transformers. *ArXiv*, abs/1912.12180.
- [105] Hobbs, H. T., Shah, N. H., Shoemaker, S. R., Amacher, J. F., Marqusee, S., & Kuriyan, J. (2022). Saturation mutagenesis of a predicted ancestral Syk-family kinase. *Protein Science*, 31(10).
- [106] Hoeffding, W. (1994). *On sequences of sums of independent random vectors*. Springer.
- [107] Hom, N., Gentles, L., Bloom, J. D., & Lee, K. K. (2019). Deep mutational scan of the highly conserved influenza a virus M1 matrix protein reveals substantial intrinsic mutational tolerance. *Journal of Virology*, 93(13).
- [108] Hon, J., Marusiak, M., Martínek, T., Kunka, A., Zendulka, J., Bednář, D., & Damborský, J. (2020). SoluProt: prediction of soluble protein expression in escherichia coli. *Bioinformatics*, 37, 23 – 28.
- [109] Hopf, T. A., Green, A. G., Schubert, B., Mersmann, S., Schärfe, C. P. I., Ingraham, J. B., Toth-Petroczy, A., Brock, K., Riesselman, A. J., Palmedo, P., Kang, C., Sheridan, R., Draizen, E. J., Dallago, C., Sander, C., & Marks, D. S. (2019). The EVcouplings python framework for coevolutionary sequence analysis. *Bioinformatics*, 35(9), 1582–1584.
- [110] Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P. I., Springer, M., Sander, C., & Marks, D. S. (2017). Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.*, 35(2), 128–135.
- [111] Hsu, C., Nisonoff, H., Fannjiang, C., & Listgarten, J. (2022a). Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7), 1114–1122.
- [112] Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., & Rives, A. (2022b). Learning inverse folding from millions of predicted structures.
- [113] Huang, P.-S., Boyken, S. E., & Baker, D. (2016). The coming of age of de novo protein design. *Nature*, 537, 320–327.

- [114] Huston, J. S., Levinson, D., Mudgett-Hunter, M., Tai, M. S., Novotný, J., Margolies, M. N., Ridge, R. J., Brucoleri, R. E., Haber, E., & Crea, R. (1988). Protein engineering of antibody binding sites: recovery of specific activity in an anti-digoxin single-chain fv analogue produced in *escherichia coli*. *Proc. Natl. Acad. Sci. U. S. A.*, 85(16), 5879–5883.
- [115] Huttinger, Z. M., Haynes, L. M., Yee, A., Kretz, C. A., Holding, M. L., Siemieniak, D. R., Lawrence, D. A., & Ginsburg, D. (2021). Deep mutational scanning of the plasminogen activator inhibitor-1 functional landscape. *Scientific Reports*, 11(1), 18827.
- [116] Ingraham, J., Garg, V. K., Barzilay, R., & Jaakkola, T. (2019). Generative models for Graph-Based protein design.
- [117] Ingraham, J. B., Baranov, M., Costello, Z., Barber, K. W., Wang, W., Ismail, A., Frappier, V., Lord, D. M., Ng-Thow-Hing, C., Van Vlack, E. R., Tie, S., Xue, V., Cowles, S. C., Leung, A., Rodrigues, J. V., Morales-Perez, C. L., Ayoub, A. M., Green, R., Puentes, K., Oplinger, F., Panwar, N. V., Obermeyer, F., Root, A. R., Beam, A. L., Poelwijk, F. J., & Grigoryan, G. (2023). Illuminating protein space with a programmable generative model. *Nature*.
- [118] Jacquier, H., Birgy, A., Le Nagard, H., Mechulam, Y., Schmitt, E., Glodt, J., Bercot, B., Petit, E., Poulain, J., Barnaud, G., Gros, P.-A., & Tenaillon, O. (2013). Capturing the mutational landscape of the beta-lactamase TEM-1. *Proceedings of the National Academy of Sciences*, 110(32), 13067–13072.
- [119] Jagota, M., Ye, C., Rastogi, R., Albors, C., Koehl, A., Ioannidis, N. M., & Song, Y. S. (2022). Cross-protein transfer learning substantially improves zero-shot prediction of disease variant effects.
- [120] Jia, X., Burugula, B. B., Chen, V., Lemons, R. M., Jayakody, S., Maksutova, M., & Kitzman, J. O. (2021). Massively parallel functional testing of MSH2 missense variants conferring Lynch syndrome risk. *The American Journal of Human Genetics*, 108(1), 163–175.
- [121] Jiang, L., Liu, P., Bank, C., Renzette, N., Prachanronarong, K., Yilmaz, L. S., Caffrey, D. R., Zeldovich, K. B., Schiffer, C. A., Kowalik, T. F., Jensen, J. D., Finberg, R. W., Wang, J. P., & Bolon, D. N. (2016). A Balance between Inhibitor Binding and Substrate Processing Confers Influenza Drug Resistance. *Journal of Molecular Biology*, 428(3), 538–553.
- [122] Jiang, R. J. (2019). *Exhaustive Mapping of Missense Variation in Coronary Heart Disease-related Genes*. PhD thesis, University of Toronto.
- [123] Jin, W., Barzilay, R., & Jaakkola, T. (2022). Antibody-Antigen docking and design via hierarchical equivariant refinement. *ICML*.
- [124] Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., & Dror, R. (2020). Learning from protein structure with geometric vector perceptrons.

- [125] Jing, B., Erives, E., Pao-Huang, P., Corso, G., Berger, B., & Jaakkola, T. (2023). EigenFold: Generative protein structure prediction with diffusion models. *arXiv [q-bio.BM]*.
- [126] Jitkrittum, W., Kanagawa, H., & Schölkopf, B. (2020). Testing goodness of fit of conditional density models with kernels. In *UAI*.
- [127] Johnson, S. R., Fu, X., Viknander, S., Goldin, C., Monaco, S., Zelezniak, A., & Yang, K. K. (2023). Computational scoring and experimental evaluation of enzymes generated by neural networks.
- [128] Jones, E. M., Lubock, N. B., Venkatakrishnan, A., Wang, J., Tseng, A. M., Paggi, J. M., Latorraca, N. R., Cancilla, D., Satyadi, M., Davis, J. E., Babu, M. M., Dror, R. O., & Kosuri, S. (2020). Structural and functional characterization of G protein–coupled receptors with deep mutational scanning. *eLife*, 9, e54895.
- [129] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, Russ and, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*.
- [130] Keeling, D. M., Garza, P., Nartey, C. M., & Carvunis, A.-R. (2019). The meanings of ‘function’ in biology and the problematic case of de novo gene emergence. *Elife*, 8.
- [131] Kelsic, E. D., Chung, H., Cohen, N., Park, J., Wang, H. H., & Kishony, R. (2016). RNA Structural Determinants of Optimal Codons Revealed by MAGE-Seq. *Cell Systems*, 3(6), 563–571.e6.
- [132] Kempeni, J. (1999). Preliminary results of early clinical trials with the fully human anti-TNFalpha monoclonal antibody D2E7. *Ann. Rheum. Dis.*, 58 Suppl 1(Suppl 1), I70–2.
- [133] Kennouche, P., Charles-Orszag, A., Nishiguchi, D., Goussard, S., Imhaus, A., Dupré, M., Chamot-Rooke, J., & Duménil, G. (2019). Deep mutational scanning of the *Neisseria meningitidis* major pilin reveals the importance of pilus tip-mediated adhesion. *The EMBO Journal*, 38(22), e102145.
- [134] Kipf, T. N. & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv [cs.LG]*.
- [135] Kitzman, J. O., Starita, L. M., Lo, R. S., Fields, S., & Shendure, J. (2015). Massively parallel single-amino-acid mutagenesis. *Nature Methods*, 12(3), 203–206.
- [136] Klesmith, J. R., Bacik, J.-P., Michalczyk, R., & Whitehead, T. A. (2015). Comprehensive Sequence-Flux Mapping of a Levoglucosan Utilization Pathway in *E. coli*. *ACS Synthetic Biology*, 4(11), 1235–1243.

- [137] Klesmith, J. R., Su, L., Wu, L., Schrack, I. A., Dufort, F. J., Birt, A., Ambrose, C., Hackel, B. J., Lobb, R. R., & Rennert, P. D. (2019). Retargeting CD19 Chimeric Antigen Receptor T Cells via Engineered CD19-Fusion Proteins. *Molecular Pharmaceutics*, 16(8), 3544–3558.
- [138] Köhler, G. & Milstein, C. (1975). Continuous cultures of fused cells secreting antibody of predefined specificity. *Nature*, 256(5517), 495–497.
- [139] Kotler, E., Shani, O., Goldfeld, G., Lotan-Pompan, M., Tarcic, O., Gershoni, A., Hopf, T. A., Marks, D. S., Oren, M., & Segal, E. (2018). A Systematic p53 Mutation Library Links Differential Functional Impact to Cancer Mutation Pattern and Evolutionary Conservation. *Molecular Cell*, 71(1), 178–190.e8.
- [140] Kozek, K. A., Glazer, A. M., Ng, C.-A., Blackwell, D., Egly, C. L., Vanags, L. R., Blair, M., Mitchell, D., Matreyek, K. A., Fowler, D. M., Knollmann, B. C., Vandenberg, J. I., Roden, D. M., & Kroncke, B. M. (2020). High-throughput discovery of trafficking-deficient variants in the cardiac potassium channel KV11.1. *Heart Rhythm*, 17(12), 2180–2189.
- [141] Kriege, N. M., Johansson, F. D., & Morris, C. (2020). A survey on graph kernels. *Applied Network Science*.
- [142] Kryshchak, A., Schwede, T., Topf, M., Fidelis, K., & Moult, J. (2021). Critical assessment of methods of protein structure prediction (CASP)—Round XIV. *Proteins: Structure*, 89, 1607 – 1617.
- [143] Kunik, V. & Ofra, Y. (2013). The indistinguishability of epitopes from protein surface is explained by the distinct binding preferences of each of the six antigen-binding loops. *Protein Engineering, Design & Selection*.
- [144] Kunik, V., Peters, B., & Ofra, Y. (2012). Structural consensus among antibodies defines the antigen binding site. *PLoS Comput. Biol.*, 8(2), e1002388.
- [145] Kwon, J. J., Hajian, B., Bian, Y., Young, L. C., Amor, A. J., Fuller, J. R., Fraley, C. V., Sykes, A. M., So, J., Pan, J., Baker, L., Lee, S. J., Wheeler, D. B., Mayhew, D. L., Persky, N. S., Yang, X., Root, D. E., Barsotti, A. M., Stamford, A. W., Perry, C. K., Burgin, A., McCormick, F., Lemke, C. T., Hahn, W. C., & Aguirre, A. J. (2022). Structure–function analysis of the SHOC2–MRAS–PPP1C holophosphatase complex. *Nature*, 609(7926), 408–415.
- [146] Laine, E., Karami, Y., & Carbone, A. (2019). GEMME: A simple and fast global epistatic model predicting mutational effects. *Mol. Biol. Evol.*, 36(11), 2604–2619.
- [147] Ledsgaard, L., Ljungars, A., Rimbault, C., Sørensen, C. V., Tulika, T., Wade, J., Wouters, Y., McCafferty, J., & Laustsen, A. H. (2022). Advances in antibody phage display technology. *Drug Discov. Today*, 27(8), 2151–2169.

- [148] Lee, E.-C. & Owen, M. (2012). The application of transgenic mice for therapeutic antibody discovery. *Methods Mol. Biol.*, 901, 137–148.
- [149] Lee, J. M., Huddleston, J., Doud, M. B., Hooper, K. A., Wu, N. C., Bedford, T., & Bloom, J. D. (2018). Deep mutational scanning of hemagglutinin helps predict evolutionary fates of human H₃N₂ influenza variants. *Proceedings of the National Academy of Sciences*, 115(35).
- [150] Lei, R., Hernandez Garcia, A., Tan, T. J. C., Teo, Q. W., Wang, Y., Zhang, X., Luo, S., Nair, S. K., Peng, J., & Wu, N. C. (2023). Mutational fitness landscape of human influenza H₃N₂ neuraminidase. *Cell Reports*, 42(1), 111951.
- [151] Leman, J. K., Weitzner, B. D., Lewis, S. M., Adolf-Bryfogle, J., Alam, N., Alford, R. F., Aprahamian, M., Baker, D., Barlow, K. A., Barth, P., Basanta, B., Bender, B. J., Blacklock, K., Bonet, J., Boyken, S. E., Bradley, P., Bystroff, C., Conway, P., Cooper, S., Correia, B. E., Coventry, B., Das, R., De Jong, R. M., DiMaio, F., Dsilva, L., Dunbrack, R., Ford, A. S., Frenz, B., Fu, D. Y., Geniesse, C., Goldschmidt, L., Gowthaman, R., Gray, J. J., Gront, D., Guffy, S., Horowitz, S., Huang, P.-S., Huber, T., Jacobs, T. M., Jeliazkov, J. R., Johnson, D. K., Kappel, K., Karanickolas, J., Khakzad, H., Khar, K. R., Khare, S. D., Khatib, F., Khramushin, A., King, I. C., Kleffner, R., Koepnick, B., Kortemme, T., Kuenze, G., Kuhlman, B., Kuroda, D., Labonte, J. W., Lai, J. K., Lapidoth, G., Leaver-Fay, A., Lindert, S., Linsky, T., London, N., Lubin, J. H., Lyskov, S., Maguire, J., Malmström, L., Marcos, E., Marcu, O., Marze, N. A., Meiler, J., Moretti, R., Mulligan, V. K., Nerli, S., Norn, C., Ó'Conchúir, S., Ollikainen, N., Ovchinnikov, S., Pacella, M. S., Pan, X., Park, H., Pavlovicz, R. E., Pethe, M., Pierce, B. G., Pilla, K. B., Raveh, B., Renfrew, P. D., Burman, S. S. R., Rubenstein, A., Sauer, M. F., Scheck, A., Schief, W., Schueler-Furman, O., Sedan, Y., Sevy, A. M., Sgourakis, N. G., Shi, L., Siegel, J. B., Silva, D.-A., Smith, S., Song, Y., Stein, A., Szegedy, M., Teets, F. D., Thyme, S. B., Wang, R. Y.-R., Watkins, A., Zimmerman, L., & Bonneau, R. (2020). Macromolecular modeling and design in rosetta: recent methods and frameworks. *Nat. Methods*, 17(7), 665–680.
- [152] Li, L., Gupta, E., Spaeth, J., Shing, L., Jaimes, R., Engelhart, E., Lopez, R., Caceres, R. S., Bepler, T., & Walsh, M. E. (2023a). Machine learning optimization of candidate antibody yields highly diverse sub-nanomolar affinity antibody libraries. *Nat. Commun.*, 14(1), 3454.
- [153] Li, T., Pantazes, R. J., & Maranas, C. D. (2014). OptMAVEN—a new framework for the de novo design of antibody variable region models targeting specific antigen epitopes. *PLoS One*, 9(8), e105954.
- [154] Li, Y., Arcos, S., Sabsay, K. R., Te Velthuis, A. J., & Lauring, A. S. (2023b). *Deep mutational scanning reveals the functional constraints and evolutionary potential of the influenza A virus PB1 protein*. preprint, Microbiology.
- [155] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., Dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., & Rives, A.

- (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637), 1123–1130.
- [156] Liu, G., Zeng, H., Mueller, J., Carter, B., Wang, Z., Schilz, J., Horny, G., Birnbaum, M. E., Ewert, S., & Gifford, D. K. (2020). Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics*.
- [157] Lloyd, J. R. & Ghahramani, Z. (2015). Statistical model criticism using kernel two sample tests. *Adv. Neural Inf. Process. Syst.*, 2015-Janua, 829–837.
- [158] Lo, R. S., Cromie, G. A., Tang, M., Teng, K., Owens, K., Sirr, A., Kutz, J. N., Morizono, H., Caldovic, L., Ah Mew, N., Gropman, A., & Dudley, A. M. (2023). The functional impact of 1,570 individual amino acid substitutions in human OTC. *American Journal of Human Genetics*, 110(5), 863–879.
- [159] Lu, R.-M., Hwang, Y.-C., Liu, I.-J., Lee, C.-C., Tsai, H.-Z., Li, H.-J., & Wu, H.-C. (2020). Development of therapeutic antibodies for the treatment of diseases. *J. Biomed. Sci.*, 27(1), 1.
- [160] MacRae, M. R., Puvanendran, D., Haase, M. A. B., Coudray, N., Kolich, L., Lam, C., Baek, M., Bhabha, G., & Ekiert, D. C. (2023). Protein-protein interactions in the mla lipid transport system probed by computational structure prediction and deep mutational scanning. *Journal of Biological Chemistry*, 299(6), 104744.
- [161] Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S., & Socher, R. (2020). ProGen: Language modeling for protein generation.
- [162] Makowski, E. K., Kinnunen, P. C., Huang, J., Wu, L., Smith, M. D., Wang, T., Desai, A. A., Streu, C. N., Zhang, Y., Zupancic, J. M., Schardt, J. S., Linderman, J. J., & Tessier, P. M. (2022). Co-optimization of therapeutic antibody affinity and specificity using machine learning models that generalize to novel mutational space. *Nat. Commun.*, 13(1), 3788.
- [163] Marks, C., Hummer, A. M., Chin, M., & Deane, C. M. (2021). Humanization of antibodies using a machine learning approach on large-scale repertoire data. *Bioinformatics*, 37(22), 4041–4047.
- [164] Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., & Sander, C. (2011). Protein 3D structure computed from evolutionary sequence variation. *PLoS One*, 6(12), e28766.
- [165] Marquet, C., Heinzinger, M., Olenyi, T., Dallago, C., Erckert, K., Bernhofer, M., Nechaev, D., & Rost, B. (2022). Embeddings from protein language models predict conservation and variant effects. *Hum. Genet.*, 141(10), 1629–1647.
- [166] Martin, K. P., Grimaldi, C., Grempler, R., Hansel, S., & Kumar, S. (2023). Trends in industrialization of biotherapeutics: a survey of product characteristics of 89 antibody-based biotherapeutics. *MAbs*, 15(1), 2191301.

- [167] Matreyek, K. A., Starita, L. M., Stephany, J. J., Martin, B., Chiasson, M. A., Gray, V. E., Kircher, M., Khechaduri, A., Dines, J. N., Hause, R. J., Bhatia, S., Evans, W. E., Relling, M. V., Yang, W., Shendure, J., & Fowler, D. M. (2018). Multiplex assessment of protein variant abundance by massively parallel sequencing. *Nature Genetics*, 50(6), 874–882.
- [168] Matreyek, K. A., Stephany, J. J., Ahler, E., & Fowler, D. M. (2021). Integrating thousands of PTEN variant activity and abundance measurements reveals variant subgroups and new dominant negatives in cancers. *Genome Medicine*, 13(1), 165.
- [169] Mattenberger, F., Latorre, V., Tirosh, O., Stern, A., & Geller, R. (2021). Globally defining the effects of mutations in a picornavirus capsid. *eLife*, 10, e64256.
- [170] Mavor, D., Barlow, K., Thompson, S., Barad, B. A., Bonny, A. R., Cario, C. L., Gaskins, G., Liu, Z., Deming, L., Axen, S. D., Caceres, E., Chen, W., Cuesta, A., Gate, R. E., Green, E. M., Hulce, K. R., Ji, W., Kenner, L. R., Mensa, B., Morinishi, L. S., Moss, S. M., Mravic, M., Muir, R. K., Niekamp, S., Nnadi, C. I., Palovcak, E., Poss, E. M., Ross, T. D., Salcedo, E. C., See, S. K., Subramaniam, M., Wong, A. W., Li, J., Thorn, K. S., Conchúir, S. ♦, Roscoe, B. P., Chow, E. D., DeRisi, J. L., Kortemme, T., Bolon, D. N., & Fraser, J. S. (2016). Determination of ubiquitin fitness landscapes under different chemical stresses in a classroom setting. *eLife*, 5, e15802.
- [171] McLaughlin, Jr., R. N., Poelwijk, F. J., Raman, A., Gosal, W. S., & Ranganathan, R. (2012). The spatial architecture of protein function and adaptation. *Nature*, 491(7422), 138–142.
- [172] McMahon, C., Baier, A. S., Pascolutti, R., Wegrecki, M., Zheng, S., Ong, J. X., Erlandson, S. C., Hilger, D., Rasmussen, S. G. F., Ring, A. M., Manglik, A., & Kruse, A. C. (2018). Yeast surface display platform for rapid discovery of conformationally selective nanobodies. *Nat. Struct. Mol. Biol.*, 25(3), 289–296.
- [173] Meier, G., Thavarasah, S., Ehrenbolger, K., Hutter, C. A. J., Hürlimann, L. M., Barandun, J., & Seeger, M. A. (2023). Deep mutational scan of a drug efflux pump reveals its structure-function landscape. *Nature Chemical Biology*, 19(4), 440–450.
- [174] Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., & Rives, A. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*.
- [175] Meitlis, I., Allenspach, E. J., Bauman, B. M., Phan, I. Q., Dabbah, G., Schmitt, E. G., Camp, N. D., Torgerson, T. R., Nickerson, D. A., Bamshad, M. J., Hagin, D., Luthers, C. R., Stinson, J. R., Gray, J., Lundgren, I., Church, J. A., Butte, M. J., Jordan, M. B., Aceves, S. S., Schwartz, D. M., Milner, J. D., Schuval, S., Skoda-Smith, S., Cooper, M. A., Starita, L. M., Rawlings, D. J., Snow, A. L., & James, R. G. (2020). Multiplexed Functional Assessment of Genetic Variants in CARD11. *The American Journal of Human Genetics*, 107(6), 1029–1043.

- [176] Melamed, D., Young, D. L., Gamble, C. E., Miller, C. R., & Fields, S. (2013). Deep mutational scanning of an RRM domain of the *Saccharomyces cerevisiae* poly(A)-binding protein. *RNA*, 19(11), 1537–1551.
- [177] Melnikov, A., Rogov, P., Wang, L., Gnirke, A., & Mikkelsen, T. S. (2014). Comprehensive mutational scanning of a kinase *in vivo* reveals substrate-dependent fitness landscapes. *Nucleic Acids Research*, 42(14), e112–e112.
- [178] Meunier, D., Pontil, M., & Ciliberto, C. (2022). Distribution regression with sliced Wasserstein kernels. In *ICML*.
- [179] Mighell, T. L., Evans-Dutson, S., & O’Roak, B. J. (2018). A Saturation Mutagenesis Approach to Understanding PTEN Lipid Phosphatase Activity and Genotype-Phenotype Relationships. *The American Journal of Human Genetics*, 102(5), 943–955.
- [180] Miller, P. G., Sathappa, M., Moroco, J. A., Jiang, W., Qian, Y., Iqbal, S., Guo, Q., Giacomelli, A. O., Shaw, S., Vernier, C., Bajrami, B., Yang, X., Raffier, C., Sperling, A. S., Gibson, C. J., Kahn, J., Jin, C., Ranaghan, M., Caliman, A., Brousseau, M., Fischer, E. S., Lintner, R., Piccioni, F., Campbell, A. J., Root, D. E., Garvie, C. W., & Ebert, B. L. (2022). Allosteric inhibition of PPM1D serine/threonine phosphatase via an altered conformational state. *Nature Communications*, 13(1), 3778.
- [181] Mishra, P., Flynn, J., Starr, T., & Bolon, D. (2016). Systematic Mutant Analyses Elucidate General and Client-Specific Aspects of Hsp90 Function. *Cell Reports*, 15(3), 588–598.
- [182] Moal, I. H. & Fernández-Recio, J. (2012). SKEMPI: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. *Bioinformatics*, 28 20, 2600–7.
- [183] Muhammad, A., Calandranis, M. E., Li, B., Yang, T., Blackwell, D. J., Harvey, M. L., Smith, J. E., Chew, A. E., Capra, J. A., Matreyek, K. A., Fowler, D. M., Roden, D. M., & Glazer, A. M. (2023). High-throughput functional mapping of variants in an arrhythmia gene, *KCNE1*, reveals novel biology. *bioRxiv*.
- [184] Muyldermans, S. (2013). Nanobodies: natural single-domain antibodies. *Annu. Rev. Biochem.*, 82, 775–797.
- [185] Newberry, R. W., Arhar, T., Costello, J., Hartoularos, G. C., Maxwell, A. M., Naing, Z. Z. C., Pittman, M., Reddy, N. R., Schwarz, D. M. C., Wassarman, D. R., Wu, T. S., Barrero, D., Caggiano, C., Catching, A., Cavazos, T. B., Estes, L. S., Faust, B., Fink, E. A., Goldman, M. A., Gomez, Y. K., Gordon, M. G., Gunsalus, L. M., Hoppe, N., Jaime-Garza, M., Johnson, M. C., Jones, M. G., Kung, A. F., Lopez, K. E., Lumpe, J., Martyn, C., McCarthy, E. E., Miller-Vedam, L. E., Navarro, E. J., Palar, A., Pellegrino, J., Saylor, W., Stephens, C. A., Strickland, J., Torosyan, H., Wankowicz, S. A., Wong, D. R., Wong, G., Redding, S., Chow,

- E. D., DeGrado, W. F., & Kampmann, M. (2020). Robust Sequence Determinants of alpha-Synuclein Toxicity in Yeast Implicate Membrane Binding. *ACS Chemical Biology*, 15(8), 2137–2153.
- [186] Nguyen, T. N., Ingle, C., Thompson, S., & Reynolds, K. A. (2023a). The genetic landscape of a metabolic interaction.
- [187] Nguyen, V., Ahler, E., Sitko, K. A., Stephany, J. J., Maly, D. J., & Fowler, D. M. (2023b). Molecular determinants of hsp90 dependence of src kinase revealed by deep mutational scanning. *Protein Science*, 32(7), e4656.
- [188] Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N., & Madani, A. (2022). ProGen2: Exploring the boundaries of protein language models.
- [189] Norn, C., Wicky, B. I. M., Juergens, D., Liu, S., Kim, D., Tischer, D., Koepnick, B., Anishchenko, I., Foldit Players, Baker, D., & Ovchinnikov, S. (2021). Protein sequence design by conformational landscape optimization. *Proc. Natl. Acad. Sci. U. S. A.*, 118(11).
- [190] Notin, P., Dias, M., Frazer, J., Marchena-Hurtado, J., Gomez, A., Marks, D. S., & Gal, Y. (2022a). Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval.
- [191] Notin, P., Kollasch, A. W., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Orenbuch, R., Gal, Y., & Marks, D. S. (2023a). Proteingym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems*.
- [192] Notin, P., Rollins, N., Gal, Y., Sander, C., & Marks, D. (2024). Machine learning for functional protein design. *Nat. Biotechnol.*, 42(2), 216–228.
- [193] Notin, P., Van Niekerk, L., Kollasch, A. W., Ritter, D., Gal, Y., & Marks, D. S. (2022b). TranceptEVE: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction. *bioRxiv*, (pp. 2022.12.07.519495).
- [194] Notin, P., Weitzman, R., Marks, D. S., & Gal, Y. (2023b). ProteinNPT: Improving protein property prediction and design with non-parametric transformers. *Advances in Neural Information Processing Systems*, 37.
- [195] Nutschel, C., Fulton, A., Zimmermann, O., Schwaneberg, U., Jaeger, K.-E., & Gohlke, H. (2020). Systematically Scrutinizing the Impact of Substitution Sites on Thermostability and Detergent Tolerance for *Bacillus subtilis* Lipase A. *Journal of Chemical Information and Modeling*, 60(3), 1568–1584.
- [196] Olson, C., Wu, N., & Sun, R. (2014). A Comprehensive Biophysical Description of Pairwise Epistasis throughout an Entire Protein Domain. *Current Biology*, 24(22), 2643–2651.

- [197] Ostermaier, M. K., Peterhans, C., Jaussi, R., Deupi, X., & Standfuss, J. (2014). Functional map of arrestin-1 at single amino acid resolution. *Proceedings of the National Academy of Sciences*, 111(5), 1825–1830.
- [198] Park, J. & Muandet, K. (2020). A measure-theoretic approach to kernel conditional mean embeddings. *NeurIPS*.
- [199] Paul, S., Konig, M. F., Pardoll, D. M., Bettegowda, C., Papadopoulos, N., Wright, K. M., Gabelli, S. B., Ho, M., van Elsas, A., & Zhou, S. (2024). Cancer therapy with antibodies. *Nat. Rev. Cancer*, 24(6), 399–426.
- [200] Pokusaeva, V. O., Usmanova, D. R., Putintseva, E. V., Espinar, L., Sarkisyan, K. S., Mishin, A. S., Bogatyreva, N. S., Ivankov, D. N., Akopyan, A. V., Avvakumov, S. Y., Povolotskaya, I. S., Filion, G. J., Carey, L. B., & Kondrashov, F. A. (2019). An experimental assay of the interactions of amino acids from orthologous sequences shaping a complex fitness landscape. *PLOS Genetics*, 15(4), e1008079.
- [201] Polizzi, N. F. & DeGrado, W. F. (2020). A defined structural unit enables de novo design of small-molecule-binding proteins. *Science*, 369(6508), 1227–1233.
- [202] Qi, H., Olson, C. A., Wu, N. C., Ke, R., Loverdo, C., Chu, V., Truong, S., Remenyi, R., Chen, Z., Du, Y., Su, S.-Y., Al-Mawsawi, L. Q., Wu, T.-T., Chen, S.-H., Lin, C.-Y., Zhong, W., Lloyd-Smith, J. O., & Sun, R. (2014). A Quantitative High-Resolution Genetic Profile Rapidly Identifies Sequence Determinants of Hepatitis C Viral Fitness and Drug Sensitivity. *PLoS Pathogens*, 10(4), e1004064.
- [203] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners.
- [204] Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., & Song, Y. S. (2019). Evaluating Protein Transfer Learning with TAPE. arXiv:1906.08230 [cs, q-bio, stat].
- [205] Rao, R., Liu, J., Verkuil, R., Meier, J., Canny, J. F., Abbeel, P., Sercu, T., & Rives, A. (2021). MSA transformer.
- [206] Riechmann, L., Clark, M., Waldmann, H., & Winter, G. (1988). Reshaping human antibodies for therapy. *Nature*, 332(6162), 323–327.
- [207] Riesselman, A. J., Ingraham, J. B., & Marks, D. S. (2018). Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10), 816–822.
- [208] Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.*, 118(15).

- [209] Rockah-Shmuel, L., Tóth-Petróczy, ♦., & Tawfik, D. S. (2015). Systematic Mapping of Protein Mutational Space by Prolonged Drift Reveals the Deleterious Effects of Seemingly Neutral Mutations. *PLOS Computational Biology*, 11(8), e1004421.
- [210] Romero, P. A., Krause, A., & Arnold, F. H. (2012). Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences*, 110, E193 – E201.
- [211] Romero, P. A., Tran, T. M., & Abate, A. R. (2015). Dissecting enzyme function with microfluidic-based deep mutational scanning. *Proceedings of the National Academy of Sciences*, 112(23), 7159–7164.
- [212] Roscoe, B. P. & Bolon, D. N. (2014). Systematic Exploration of Ubiquitin Sequence, E1 Activation Efficiency, and Experimental Fitness in Yeast. *Journal of Molecular Biology*, 426(15), 2854–2870.
- [213] Roscoe, B. P., Thayer, K. M., Zeldovich, K. B., Fushman, D., & Bolon, D. N. (2013). Analyses of the Effects of All Ubiquitin Point Mutants on Yeast Growth Rate. *Journal of Molecular Biology*, 425(8), 1363–1377.
- [214] Roychowdhury, H. & Romero, P. A. (2022). Microfluidic deep mutational scanning of the human executioner caspases reveals differences in structure and regulation. *Cell Death Discovery*, 8(1), 7.
- [215] Rubin, A. F., Min, J. K., Rollins, N. J., Da, E. Y., Esposito, D., Harrington, M., Stone, J., Bianchi, A. H., Dias, M., Frazer, J., Fu, Y., Gallaher, M., Li, I., Moscatelli, O., Ong, J. Y., Rollins, J. E., Wakefield, M. J., Ye, S. ♦., Tam, A. S. P., McEwen, A. E., Starita, L. M., Bryant, V. L., Marks, D. S., & Fowler, D. M. (2021). MaveDB v2: a curated community database with over three million variant effects from multiplexed functional assays. *bioRxiv*.
- [216] Russ, W. P., Lowery, D. M., Mishra, P., Yaffe, M. B., & Ranganathan, R. (2005). Natural-like function in artificial WW domains. *Nature*, 437(7058), 579–583.
- [217] Saigo, H., Vert, J.-P., Ueda, N., & Akutsu, T. (2004). Protein homology detection using string alignment kernels. *Bioinformatics*.
- [218] Saka, K., Kakuzaki, T., Metsugi, S., Kashiwagi, D., Yoshida, K., Wada, M., Tsunoda, H., & Teramoto, R. (2021). Antibody design using LSTM based deep generative model from phage display library for affinity maturation. *Sci. Rep.*, 11(1), 5852.
- [219] Sarkisyan, K. S., Bolotin, D. A., Meer, M. V., Usmanova, D. R., Mishin, A. S., Sharonov, G. V., Ivankov, D. N., Bozhanova, N. G., Baranov, M. S., Soylemez, O., Bogatyreva, N. S., Vlasov, P. K., Egorov, E. S., Logacheva, M. D., Kondrashov, A. S., Chudakov, D. M., Putintseva, E. V., Mamedov, I. Z., Tawfik, D. S., Lukyanov, K. A., & Kondrashov, F. A. (2016). Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603), 397–401.

- [220] Schafmeister, C. E. & Stroud, R. M. (1998). Helical protein design. *Curr. Opin. Biotechnol.*, 9(4), 350–353.
- [221] Schrab, A., Kim, I., Guedj, B., & Gretton, A. (2022). Efficient aggregated kernel tests using incomplete u -statistics. *Advances in Neural Information Processing Systems*, 35, 18793–18807.
- [222] Schubert, B., Schärfe, C., Dönnies, P., Hopf, T., Marks, D., & Kohlbacher, O. (2018). Population-specific design of de-immunized protein biotherapeutics. *PLOS Computational Biology*, 14(3), 1–19.
- [223] Sella, G. & Hirsh, A. E. (2005). The application of statistical physics to evolutionary biology. *Proc. Natl. Acad. Sci. U. S. A.*, 102(27), 9541–9546.
- [224] Serfling, R. J. (2009). *Approximation theorems of mathematical statistics*. John Wiley & Sons.
- [225] Seuma, M., Faure, A. J., Badia, M., Lehner, B., & Bolognesi, B. (2021). The genetic landscape for amyloid beta fibril nucleation accurately discriminates familial Alzheimer’s disease mutations. *eLife*, 10, e63364.
- [226] Seuma, M., Lehner, B., & Bolognesi, B. (2022). An atlas of amyloid aggregation: the impact of substitutions, insertions, deletions and truncations on amyloid beta fibril nucleation. *Nature Communications*, 13(1), 7084.
- [227] Shin, J.-E., Riesselman, A. J., Kollasch, A. W., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A. C., & Marks, D. S. (2021). Protein design and variant prediction using autoregressive generative models. *Nat. Commun.*, 12(1), 2403.
- [228] Shortle, D. & Sondek, J. (1995). The emerging role of insertions and deletions in protein engineering. *Curr. Opin. Biotechnol.*, 6(4), 387–393.
- [229] Sillitoe, I., Bordin, N., Dawson, N., Waman, V. P., Ashford, P., Scholes, H. M., Pang, C. S. M., Woodridge, L., Rauer, C., Sen, N., Abbasian, M., Le Cornu, S., Lam, S. D., Berka, K., Varekova, I. H., Svobodova, R., Lees, J., & Orengo, C. A. (2021). CATH: increased structural coverage of functional space. *Nucleic acids research*.
- [230] Silverstein, R. A., Sun, S., Verby, M., Weile, J., Wu, Y., Gebbia, M., Fotiadou, I., Kitaygorodsky, J., & Roth, F. P. (2021). *A systematic genotype-phenotype map for missense variants in the human intellectual disability-associated gene GDI1*. preprint, Genetics.
- [231] Sinai, S., Jain, N., Church, G. M., & Kelsic, E. D. (2021). *Generative AAV capsid diversification by latent interpolation*. preprint, Synthetic Biology.
- [232] Soh, Y. S., Moncla, L. H., Eguia, R., Bedford, T., & Bloom, J. D. (2019). Comprehensive mapping of adaptation of the avian influenza polymerase protein PB2 to humans. *eLife*, 8, e45079.

- [233] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv [cs.LG]*.
- [234] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv [cs.LG]*.
- [235] Sourisseau, M., Lawrence, D. J. P., Schwarz, M. C., Storrs, C. H., Veit, E. C., Bloom, J. D., & Evans, M. J. (2019). Deep Mutational Scanning Comprehensively Maps How Zika Envelope Protein Mutations Affect Viral Growth and Antibody Escape. *Journal of Virology*, 93(23), e01291–19.
- [236] Spencer, J. M. & Zhang, X. (2017). Deep mutational scanning of *S. pyogenes* Cas9 reveals important functional domains. *Scientific Reports*, 7(1), 16836.
- [237] Sriperumbudur, B., Fukumizu, K., & Lanckriet, G. (2010). On the relation between universality, characteristic kernels and rkhs embedding of measures. In *AISTATS*.
- [238] Stadelmann, T., Heid, D., Jendrusch, M., Mathony, J., Rosset, S., Correia, B. E., & Niopek, D. (2021). *A deep mutational scanning platform to characterize the fitness landscape of anti-CRISPR proteins*. preprint, Synthetic Biology.
- [239] Staller, M. V., Holehouse, A. S., Swain-Lenz, D., Das, R. K., Pappu, R. V., & Cohen, B. A. (2018). A High-Throughput Mutational Scan of an Intrinsically Disordered Acidic Transcriptional Activation Domain. *Cell Systems*, 6(4), 444–455.e6.
- [240] Starita, L. M., Pruneda, J. N., Lo, R. S., Fowler, D. M., Kim, H. J., Hiatt, J. B., Shendure, J., Brzovic, P. S., Fields, S., & Kleit, R. E. (2013). Activity-enhancing mutations in an E3 ubiquitin ligase identified by high-throughput mutagenesis. *Proceedings of the National Academy of Sciences*, 110(14).
- [241] Starr, T. N., Greaney, A. J., Hilton, S. K., Ellis, D., Crawford, K. H., Dingens, A. S., Navarro, M. J., Bowen, J. E., Tortorici, M. A., Walls, A. C., King, N. P., Vesler, D., & Bloom, J. D. (2020). Deep Mutational Scanning of SARS-CoV-2 Receptor Binding Domain Reveals Constraints on Folding and ACE2 Binding. *Cell*, 182(5), 1295–1310.e20.
- [242] Steinegger, M. & Söding, J. (2018). Clustering huge protein sequence sets in linear time. *Nature Communications*, 9(1), 2542.
- [243] Stiffler, M., Hekstra, D., & Ranganathan, R. (2015). Evolvability as a Function of Purifying Selection in TEM-1 beta-Lactamase. *Cell*, 160(5), 882–892.
- [244] Stourac, J., Dbrava, J., Musil, M., Horácková, J., Damborský, J., Mazurenko, S., & Bednář, D. (2020). FireProtDB: database of manually curated protein stability data. *Nucleic Acids Research*, 49, D319 – D324.

- [245] Suiter, C. C., Moriyama, T., Matreyek, K. A., Yang, W., Scaletti, E. R., Nishii, R., Yang, W., Hoshitsuki, K., Singh, M., Trehan, A., Parish, C., Smith, C., Li, L., Bhojwani, D., Yuen, L. Y. P., Li, C.-k., Li, C.-h., Yang, Y.-l., Walker, G. J., Goodhand, J. R., Kennedy, N. A., Klusmann, F. A., Bhatia, S., Relling, M. V., Kato, M., Hori, H., Bhatia, P., Ahmad, T., Yeoh, A. E. J., Stenmark, P., Fowler, D. M., & Yang, J. J. (2020). Massively parallel variant characterization identifies *NUDT15* alleles associated with thiopurine toxicity. *Proceedings of the National Academy of Sciences*, 117(10), 5394–5401.
- [246] Sumida, K. H., Núñez-Franco, R., Kalvet, I., Pellock, S. J., Wicky, B. I. M., Milles, L. F., Dauraras, J., Wang, J., Kipnis, Y., Jameson, N., Kang, A., De La Cruz, J., Sankaran, B., Bera, A. K., Jiménez-Osés, G., & Baker, D. (2024). Improving protein expression, stability, and function with ProteinMPNN. *Journal of the American Chemical Society*.
- [247] Sun, S., Weile, J., Verby, M., Wu, Y., Wang, Y., Cote, A. G., Fotiadou, I., Kitaygorodsky, J., Vidal, M., Rine, J., Ješina, P., Kožich, V., & Roth, F. P. (2020). A proactive genotype-to-patient-phenotype map for cystathionine beta-synthase. *Genome Medicine*, 12(1), 13.
- [248] Suphatrakul, A., Posiri, P., Srisuk, N., Nantachokchawapan, R., Onnoms, S., Mongkol-sapaya, J., & Siridechadilok, B. (2023). Functional analysis of flavivirus replicase by deep mutational scanning of dengue NS5.
- [249] Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., & Consortium, U. (2015). UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6), 926–932.
- [250] Szabó, Z., Gretton, A., Póczos, B., & Sriperumbudur, B. K. (2015). Two-stage sampled learning theory on distributions. In *AISTATS*.
- [251] Szabó, Z., Sriperumbudur, B. K., Póczos, B., & Gretton, A. (2016). Learning theory for distribution regression. *JMLR*.
- [252] Tan, T. J. C., Mou, Z., Lei, R., Ouyang, W. O., Yuan, M., Song, G., Andrabi, R., Wilson, I. A., Kieffer, C., Dai, X., Matreyek, K. A., & Wu, N. C. (2023). High-throughput identification of prefusion-stabilizing mutations in SARS-CoV-2 spike. *Nature Communications*, 14(1), 2003.
- [253] Thompson, S., Zhang, Y., Ingle, C., Reynolds, K. A., & Kortemme, T. (2020). Altered expression of a quality control protease in *E. coli* reshapes the in vivo mutational landscape of a model enzyme. *eLife*, 9, e53476.
- [254] Thyagarajan, B. & Bloom, J. D. (2014). The inherent mutational tolerance and antigenic evolvability of influenza hemagglutinin. *eLife*, 3.
- [255] Tian, P., Louis, J. M., Baber, J. L., Aniana, A., & Best, R. B. (2018). Co-Evolutionary fitness landscapes for sequence design. *Angew. Chem. Int. Ed Engl.*, 57(20), 5674–5678.

- [256] Totewad, N. D., Thamizharasi, M., Ritika, & Agarwal, G. (2022). A review on monoclonal antibodies for the treatment of autoimmune disorders. https://www.researchgate.net/publication/366249675_A_Review_On_Monoclonal_Antibodies_For_The_Treatment_Of_Autoimmune_Disorders. Accessed: 2024-7-9.
- [257] Tóth-Petróczy, A. & Tawfik, D. S. (2013). Protein insertions and deletions enabled by neutral roaming in sequence space. *Mol. Biol. Evol.*, 30(4), 761–771.
- [258] Tripathi, A., Gupta, K., Khare, S., Jain, P. C., Patel, S., Kumar, P., Pulianmackal, A. J., Aghera, N., & Varadarajan, R. (2016). Molecular Determinants of Mutant Phenotypes, Inferred from Saturation Mutagenesis Data. *Molecular Biology and Evolution*, 33(11), 2960–2975.
- [259] Trippe, B. L., Yim, J., Tischer, D. K., Broderick, T., Baker, D., Barzilay, R., & Jaakkola, T. (2022). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *ArXiv*, abs/2206.04119.
- [260] Tsuboyama, K., Dauparas, J., Chen, J., Laine, E., Mohseni Behbahani, Y., Weinstein, J. J., Mangan, N. M., Ovchinnikov, S., & Rocklin, G. J. (2023). Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973), 434–444.
- [261] UK Monogenic Diabetes Consortium, Myocardial Infarction Genetics Consortium, UK Congenital Lipodystrophy Consortium, Majithia, A. R., Tsuda, B., Agostini, M., Gnanapradeepan, K., Rice, R., Peloso, G., Patel, K. A., Zhang, X., Broekema, M. F., Patterson, N., Duby, M., Sharpe, T., Kalkhoven, E., Rosen, E. D., Barroso, I., Ellard, S., Kathiresan, S., O’Rahilly, S., Chatterjee, K., Florez, J. C., Mikkelsen, T., Savage, D. B., & Altshuler, D. (2016). Prospective functional classification of all possible missense variants in PPAR γ . *Nature Genetics*, 48(12), 1570–1575.
- [262] Urbina, F., Lentzos, F., Invernizzi, C., & Ekins, S. (2022). Dual use of artificial-intelligence-powered drug discovery. *Nature Machine Intelligence*, 4(3), 189–191.
- [263] Ursu, O., Neal, J. T., Shea, E., Thakore, P. I., Jerby-Arnon, L., Nguyen, L., Dionne, D., Diaz, C., Bauman, J., Mosaad, M. M., Fagre, C., Lo, A., McSharry, M., Giacomelli, A. O., Ly, S. H., Rozenblatt-Rosen, O., Hahn, W. C., Aguirre, A. J., Berger, A. H., Regev, A., & Boehm, J. S. (2022). Massively parallel phenotyping of coding variants in cancer with perturb-seq. *Nature Biotechnology*, 40(6), 896–905.
- [264] Vaicenavicius, J., Widmann, D., Andersson, C., Lindsten, F., Roll, J., & Schön, T. (2019). Evaluating model calibration in classification. In *AISTATS*.
- [265] van Loggerenberg, W., Sowlati-Hashjin, S., Weile, J., Hamilton, R., Chawla, A., Gebbia, M., Kishore, N., Frésard, L., Mustajoki, S., Pischik, E., Pierro, E. D., Barbaro, M., Floderus, Y., Schmitt, C., Gouya, L., Colavin, A., Nussbaum, R., Friesema, E. C. H., Kauppinen, R.,

- To-Figueras, J., Aarsand, A. K., Desnick, R. J., Garton, M., & Roth, F. P. (2023). Systematically testing human HMBS missense variants to reveal mechanism and pathogenic variation. *bioRxiv*.
- [266] Vanella, R., Küng, C., Schoepfer, A. A., Doffini, V., Ren, J., & Nash, M. A. (2023). Understanding Activity-Stability tradeoffs in biocatalysts by enzyme proximity sequencing.
- [267] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need.
- [268] Veerapandian, V., Ackermann, J. O., Srivastava, Y., Malik, V., Weng, M., Yang, X., & Jauch, R. (2018). Directed evolution of reprogramming factors by cell selection and sequencing. *Stem Cell Reports*, 11(2), 593–606.
- [269] Vert, J.-P., Saigo, H., & Akutsu, T. (2004). Local alignment kernels for biological sequences. *Kernel methods in computational biology*.
- [270] Wan, A., Place, E., Pierce, E. A., & Comander, J. (2019). Characterizing variants of unknown significance in rhodopsin: A functional genomics approach. *Human Mutation*, 40(8), 1127–1144.
- [271] Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., & Baker, D. (2023). De novo design of protein structure and function with RFdiffusion. *Nature*.
- [272] Weeks, R. & Ostermeier, M. (2023). Fitness and functional landscapes of the e. coli RNase III gene rnc. *Molecular Biology and Evolution*, 40(3).
- [273] Wei, H. & Li, X. (2023). Deep mutational scanning: A versatile tool in systematically mapping genotypes to phenotypes. *Front. Genet.*, 14, 1087267.
- [274] Weile, J., Kishore, N., Sun, S., Maaieh, R., Verby, M., Li, R., Fotiadou, I., Kitaygorodsky, J., Wu, Y., Holenstein, A., Bürer, C., Blomgren, L., Yang, S., Nussbaum, R., Rozen, R., Watkins, D., Gebbia, M., Kozich, V., Garton, M., Froese, D. S., & Roth, F. P. (2021). Shifting landscapes of human MTHFR missense-variant effects. *American Journal of Human Genetics*, 108(7), 1283–1300.
- [275] Weile, J., Sun, S., Cote, A. G., Knapp, J., Verby, M., Mellor, J. C., Wu, Y., Pons, C., Wong, C., Lieshout, N., Yang, F., Tasan, M., Tan, G., Yang, S., Fowler, D. M., Nussbaum, R., Bloom, J. D., Vidal, M., Hill, D. E., Aloy, P., & Roth, F. P. (2017). A framework for exhaustively mapping functional missense variants. *Molecular Systems Biology*, 13(12), 957.

- [276] Weinstein, E., Amin, A., Frazer, J., & Marks, D. (2022). Non-identifiability and the blessings of misspecification in models of molecular fitness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems*, volume 35 (pp. 5484–5497).: Curran Associates, Inc.
- [277] Weng, C., Faure, A. J., & Lehner, B. (2022). The energetic and allosteric landscape for KRAS inhibition.
- [278] Widmann, D., Lindsten, F., & Zachariah, D. (2021). Calibration tests beyond classification. In *ICLR*.
- [279] Wingler, L. M., McMahon, C., Staus, D. P., Lefkowitz, R. J., & Kruse, A. C. (2019). Distinctive activation mechanism for angiotensin receptor revealed by a synthetic nanobody. *Cell*, 176(3), 479–490.e12.
- [280] Wrenbeck, E. E., Azouz, L. R., & Whitehead, T. A. (2017). Single-mutation fitness landscapes for an enzyme on multiple substrates reveal specificity is globally encoded. *Nature Communications*, 8(1), 15695.
- [281] Wrenbeck, E. E., Bedewitz, M. A., Klesmith, J. R., Noshin, S., Barry, C. S., & Whitehead, T. A. (2019). An automated Data-Driven pipeline for improving heterologous enzyme expression. *ACS Synthetic Biology*, 8(3), 474–481.
- [282] Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O., & Sun, R. (2016). Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5, e16965.
- [283] Wu, N. C., Olson, C. A., Du, Y., Le, S., Tran, K., Remenyi, R., Gong, D., Al-Mawsawi, L. Q., Qi, H., Wu, T.-T., & Sun, R. (2015). Functional Constraint Profiling of a Viral Protein Reveals Discordance of Evolutionary Conservation and Functionality. *PLOS Genetics*, 11(7), e1005310.
- [284] Wu, N. C., Young, A. P., Al-Mawsawi, L. Q., Olson, C. A., Feng, J., Qi, H., Chen, S.-H., Lu, I.-H., Lin, C.-Y., Chin, R. G., Luan, H. H., Nguyen, N., Nelson, S. F., Li, X., Wu, T.-T., & Sun, R. (2014). High-throughput profiling of influenza A virus hemagglutinin gene at single-nucleotide resolution. *Scientific Reports*, 4(1), 4942.
- [285] Wu, Z., Kan, S. B. J., Lewis, R. D., Wittmann, B. J., & Arnold, F. H. (2019). Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116, 8852 – 8858.
- [286] Xie, M. J., Cromie, G. A., Owens, K., Timour, M. S., Tang, M., Kutz, J. N., El-Hattab, A. W., McLaughlin, R. N., & Dudley, A. M. (2023). Predicting the functional effect of compound heterozygous genotypes from large scale variant effect maps. *bioRxiv*.

- [287] Xu, M., Zhang, Z., Lu, J., Zhu, Z., Zhang, Y., Ma, C., Liu, R., & Tang, J. (2022). PEER: A Comprehensive and Multi-Task Benchmark for Protein Sequence Understanding. *arXiv:2206.02096 [cs]*.
- [288] Yang, K. K., Lu, A. X., & Fusi, N. (2023a). Convolutions are competitive with transformers for protein sequence pretraining. *bioRxiv*.
- [289] Yang, K. K., Wu, Z., & Arnold, F. H. (2018). Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, (pp. 1–8).
- [290] Yang, K. K., Zanichelli, N., & Yeh, H. (2023b). Masked inverse folding with sequence transfer for protein representation learning. *bioRxiv*.
- [291] Yee, S. W., Macdonald, C., Mitrovic, D., Zhou, X., Koleske, M. L., Yang, J., Silva, D. B., Grimes, P. R., Trinidad, D., More, S. S., Kachuri, L., Witte, J. S., Delemotte, L., Giacomini, K. M., & Coyote-Maestas, W. (2023). The full spectrum of OCT1 (SLC22A1) mutations bridges transporter biophysics to drug pharmacogenomics. *bioRxiv*.
- [292] Yi, K., Zhou, B., Shen, Y., Liò, P., & Wang, Y. G. (2023). Graph denoising diffusion for inverse protein folding. *arXiv preprint arXiv:2306.16819*.
- [293] Young, H. J., Chan, M., Selvam, B., Szymanski, S. K., Shukla, D., & Procko, E. (2021). *Deep Mutagenesis of a Transporter for Uptake of a Non-Native Substrate Identifies Conformationally Dynamic Regions*. preprint, Biochemistry.
- [294] Zambrano, N., Froechlich, G., Lazarevic, D., Passariello, M., Nicosia, A., De Lorenzo, C., Morelli, M. J., & Sasso, E. (2022). High-Throughput monoclonal antibody discovery from phage libraries: Challenging the current preclinical pipeline to keep the pace with the increasing mab demand. *Cancers*, 14(5).
- [295] Zhang, Z., Xu, M., Chenthamarakshan, V., Lozano, A., Das, P., & Tang, J. (2023). Enhancing protein language models with structure-based encoder and pre-training. *arXiv preprint arXiv:2303.06275*.
- [296] Zheng, Z., Deng, Y., Xue, D., Zhou, Y., Ye, F., & Gu, Q. (2023). Structure-informed language models are protein designers.
- [297] Zhou, N., Jiang, Y., Bergquist, T., Lee, A. J., Kacsoh, B. Z., Crocker, A., Lewis, K. A., Georghiou, G. E., Nguyen, H. N., Hamid, N. I. B., Davis, L., Dogan, T., Atalay, V., Rifaioğlu, A. S., Dalkiran, A., Cetin-Atalay, R., Zhang, C., Hurto, R. L., Freddolino, P. L., Zhang, Y., Bhat, P., Supek, F., Fernández, J. M., Gemović, B., Perovic, V., Davidovic, R., Sumonja, N., Veljkovic, N., Asgari, E., Mofrad, M. R. K., Profiti, G., Savojardo, C., Martelli, P. L., Casadio, R., Boecker, F., Kahanda, I., Thurlby, N., Mchardy, A., Renaux, A., Saidi, R., Gough, J., Freitas, A. A., Antczak, M., Fabris, F., Wass, M. N., Hou, J., Cheng, J., Wang,

Z., Romero, A. E., Paccanaro, A., Yang, H., Goldberg, T., Zhao, C., Holm, L., Törönen, P., Medlar, A., Zosa, E., Borukhov, I., Novikov, I. B., Wilkins, A. D., Lichtarge, O., Chi, P.-H., Tseng, W.-C., Linial, M., Rose, P. W., Dessimoz, C., Vidulin, V., Dzeroski, S., Sillitoe, I. P. W., Das, S., Lees, J. G., Jones, D. T., Wan, C., Cozzetto, D., Fa, R., Torres, M., Vesztröcy, A. W., Rodriguez, J. M., Tress, M. L., Frasca, M., Notaro, M., Grossi, G., Petrini, A., Ré, M., Valentini, G., Mesiti, M., Roche, D. B., Reeb, J., Ritchie, D. W., Aridhi, S., Alborzi, S. Z., Devignes, M.-D., Koo, D. C. E., Bonneau, R., Gligorijević, V., Barot, M., Fang, H., Toppo, S., Lavezzo, E., Falda, M., Berselli, M., Tosatto, S. C. E., Carraro, M., Piovesan, D., ur Rehman, H., Mao, Q., Zhang, S., Vucetic, S., Black, G. S., Jo, D., Larsen, D. J., Omdahl, A., Sagers, L., Suh, E., Dayton, J. B., McGuffin, L. J., Brackenridge, D. A., Babbitt, P. C., Yunes, J. M., Fontana, P., Zhang, F., Zhu, S., You, R., Zhang, Z., Dai, S., Yao, S., Tian, W., Cao, R., Chandler, C., Amezola, M., Johnson, D., Chang, J.-M., Liao, W.-H., Liu, Y.-W., Pascarelli, S., Frank, Y., Hoehndorf, R., Kulmanov, M., Boudelloua, I., Politano, G., Carlo, S. D., Benso, A., Hakala, K., Ginter, F., Mehryar, F., Kaewphan, S., Björne, J., Moen, H., Tolvanen, M., Salakoski, T., Kihara, D., Jain, A., Smuc, T., Altenhoff, A. M., Ben-Hur, A., Rost, B., Brenner, S. E., Orengo, C. A., Jeffery, C. J., Bosco, G., Hogan, D. A., Martin, M. J., O'Donovan, C., Mooney, S. D., Greene, C. S., Radivojac, P., & Friedberg, I. (2019). The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, 20.

- [298] Zinkus-Boltz, J., DeValk, C., & Dickinson, B. C. (2019). A Phage-Assisted Continuous Selection Approach for Deep Mutational Scanning of Protein–Protein Interactions. *ACS Chemical Biology*, 14(12), 2757–2767.



THIS THESIS WAS TYPESET using $\LaTeX$, originally developed by Leslie Lamport and based on Donald Knuth's $\TeX$. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. The above illustration, "Science Experiment 02", was created by Ben Schlitter and released under [CC BY-NC-ND 3.0](#). A template that can be used to format a PhD thesis with this look and feel has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/Dissertate or from its author, Jordan Suchow, at suchow@post.harvard.edu.