THE UNIVERSITY *of* EDINBURGH

AI-driven design of enzyme replacement therapies

Evgenii Lobzaev

Doctor of Philosophy

CDT Biomedical Artificial Intelligence

School of Informatics

The University of Edinburgh

2024

Abstract

Artificial intelligence (AI) and Machine Learning (ML) have become pivotal technologies in the 21st century, revolutionizing many industries, including retail, finance, manufacturing and healthcare among others. The role of AI and ML in biology and medicine is equally profound, with significant research efforts highlighting their potential. In protein engineering, AI and ML have been used to predict protein structure, function, and interactions, as well as to design novel proteins with desired characteristics.

In this work, I focused on the development of computational methods that should facilitate the design of novel therapeutics for Lysosomal storage disorders (LSDs), specifically targeting Fabry disease. Fabry disease, a rare genetic disorder, affects multiple parts of the body, including kidneys, heart, and skin. The treatment of Fabry disease is largely based on the administration of Enzyme replacement therapies (ERTs), which are recombinant α-galactosidase (AGAL) enzymes that replace the missing or defective enzyme in the patient's body. Despite the availability of three approved ERTs for Fabry disease in Europe, limitations such as immunogenicity, high cost, and limited efficacy, call for the development of novel ERTs.

First, I developed a baseline Variational autoencoder (VAE) model that effectively learns evolutionary constraints from a small set of homologous sequences. The model was validated on mutation effect prediction task and showed comparable performance to the state-of-the-art methods, while being smaller. It was then used to generate a library of AGAL enzyme variants which maintained biochemical and structural properties of the wild-type enzyme, while avoiding deleterious mutations. This showcased how the model can be used to generate diverse set of potential ERT candidates for further experimental validation.

Designing sequences with enhanced properties is both challenging and desirable. In the second part of this work, I developed a generative model that learns sequence-to-free-energy relationship from a small set of biophysical simulations and can be used

to generate novel and stable variants of a protein. The model was validated both computationally and experimentally on 40 AI-designed variants of semi-essential *E. coli* phosphotransferase N-acetyl-L-glutamate kinase (*Ec*NAGK) protein, crucial for cell survival. Results of these experiments demonstrate how the model can be used for the library design of thermodynamically stable AGAL variants.

Immunogenicity is a major concern in the development of protein therapeutics. Epitopes, parts of a protein that are recognized by the immune system, are the main cause of immunogenicity. These epitopes need to be modified or masked in order to reduce the immunogenicity of a therapeutic protein. In the third part of this work, I proposed a novel generative model that combines sequence and structure information to generate protein variants with modified epitopes. By assessing the model's performance, enhanced through pretraining on a broad dataset of protein structures and sequences, then finetuning on a targeted dataset of AGAL homologous sequences and their structures, and evaluating the impact of structural data, the study explores the advantages over a sequence-only modeling approach in epitope redesign problem.

Lay summary

Artificial intelligence (AI) and Machine Learning (ML) are 21st century technologies revolutionizing many industries. In biology and medicine, AI and ML have been used to predict protein structure and function, protein interactions, and to design novel proteins and medicines with desired characteristics.

My research is focused on the development of AI-based computational methods that should help to create new treatments for a group of inherited diseases known as Lysosomal storage disorders (LSDs), focusing particularly on Fabry disease. This rare genetic disease affects various parts of the body, including kidneys, heart, and skin and generally treated by the administration of Enzyme replacement therapies (ERTs), which are synthetic α-galactosidase (AGAL) enzymes that replace the missing or malfunctioning enzyme in patients. However, currently available treatments have several drawbacks, such as immune reaction, limited effectiveness, and high cost. That is why there is a pressing need for new and improved ERTs.

In the first part of my research I developed a generative model that can understand the evolutionary relationships between and within provided protein sequences. Once trained, this model was shown to be effective at predicting the impact of mutations on protein function for a set of well-studied proteins. The model achieved comparable performance to the state-of-the-art methods, while being substantially smaller. We then used this model to generate new AGAL enzyme variants that were distinct from the wild-type enzyme on a sequence level, but maintain similar biochemical and structural properties. This demonstrates that the model can be used in real-world applications to generate diverse sets of AGAL mutants for further experimental validation.

In the second part of my research I focused on developing a generative model that can produce novel protein sequences with enhanced properties, specifically focusing on the stability of a protein. The model was used to design 40 new variants of the protein *E. coli* phosphotransferase N-acetyl-L-glutamate kinase (*Ec*NAGK), crucial for cell survival, which were then experimentally tested. These results showed how

the model can be used to design new and stable AGAL variants, with stability being a very important property for therapeutic proteins.

One of the most important problems in the development of therapeutic proteins is avoiding immune reactions when the medicine is administered. The immune system recognizes particular parts of a protein, called epitopes, and fights against them similar to how it fights against a virus. To make a protein safer for patients, we need to change or hide these epitopes. The third part of my research was focused on developing a generative model that uses both protein sequence and structure information to create versions of a protein with modified epitopes that are less likely to cause an immune reaction. The model gets better by learning first from a wide range of protein sequences and structures and then focusing on the specific proteins related to the AGAL enzyme with the objective of reducing the immunogenicity of the enzyme by changing its epitopes. By doing an array of diverse experiments, we assessed the benefits of adding structural information to the model for the epitope modification problem.

Acknowledgements

I would like to acknowledge the support of my main supervisor, Professor Giovanni Stracquadanio, whose constant supervision and guidance made this highly interdisciplinary project concerning the development of novel enzyme replacement therapies for the treatment of lysosomal storage disorders possible. Without him it would be impossible to join together different pieces of knowledge from different fields, such as AI, microbiology and biochemistry.

I would also like to thank past and current members of the Stracquadanio Lab with whom I closely worked during the course of my PhD: Dr. Michael A. Herrera, Dr. Ginevra Camboni, Dr. Louise Holyoake and Martyna Kasprzyk, for their help in conducting experiments and providing evidence that my AI models work; Anima Sutradhar, for the discussions about microbiology and RNA-sequencing.

Finally, I would like to thank my family for their support and encouragement throughout my studies.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Evgenii Lobzaev)*

Table of Contents

Acronyms **1**

1 Introduction **5**

1.1 Background 5

1.2 Thesis Outline 11

2 Dirichlet latent modelling for protein sequence generation **13**

2.1 Introduction 13

2.2 Methods 16

2.3 Results 21

2.4 Discussion 35

3 Protein engineering using variational free energy approximation **38**

3.1 Introduction 38

3.2 Methods 40

3.3 Results 48

3.4 Discussion 60

4 Epitope recoding using multimodal generative deep learning **64**

4.1 Introduction 64

4.2 Methods 66

4.3 Results 70

4.4 Discussion 79

5 Conclusions **83**

A Supplementary materials for chapter 2 **88**

B Supplementary materials for chapter 3 **91**

C Supplementary materials for chapter 4 **104**

Bibliography **117**

Acronyms

E. coli	Escherichia coli
*Ec*NAGK	*E. coli* phosphotransferase N-acetyl-L-glutamate kinase
DEEPSEQUENCE	DeepSequence
TDVAE	Temporal Dirichlet Variational Autoencoder
TGVAE	Temporal Gaussian Variational Autoencoder
ADA	anti-drug antibody
AFDB	AlphaFold Protein Structure Database
AGAL	α-galactosidase
AI	Artificial intelligence
APC	Antigen-presenting cell
BCR	B-cell receptor
CASP	Critical Assessment of Structure Prediction
CAT	concordance at the top
CDF	Cumulative Density Function
DE	Directed evolution
DL	Deep learning

ELBO	Evidence Lower BOund
eNMA	ensemble Normal Mode Analysis
ERT	Enzyme replacement therapy
FCNN	Fully connected Neural Network
GAN	Generative adversarial network
GFP	Green fluorescent protein
GRU	Gated Recurrent Unit
HMM	Hidden Markov model
JUMP	Joint Universal Modular Plasmids
KL divergence	Kullback-Liebler divergence
LLM	Large language model
LM	Language model
LSD	Lysosomal storage disorder
LSTM	Long Short Term Memory
MC	Monte Carlo
MDH	Malate dehydrogenase
MHC	Major histocompatibility complex
ML	Machine Learning
MSA	Multiple Sequence Alignment
MSE	Mean Squared Error

NAG	N-acetyl-L-glutamate
NLP	Natural language processing
NN	Neural Network
PCA	Principal Component Analysis
PDF	Probability Density Function
pLDDT	predicted local distance difference test
PLM	Protein Language Model
POE	prior optimization of free energy
PREVENT	PRotein Engineering by Variational frEe eNergy approximaTion
RMSD	Root Mean Square Deviation
RMSE	Root Mean Square Error
RMSF	Root Mean Square Fluctuation
RNN	Recurrent Neural Network
SA	Simulated annealing
SNE	seeded non optimal free energy ranking
SRT	Substrate reduction therapy
SVI	Stochastic Variational Inference
SVM	Support vector machine
TCN	Temporal convolutional network
TCR	T-cell receptor
VAE	Variational autoencoder

VI	Variational Inference

Chapter 1

Introduction

1.1 Background

Lysosomal storage disorders and Fabry disease

Enzymes, a subset of proteins, serve as biological catalysts facilitating cellular reactions. These proteins, when functionally compromised, are implicated in the onset of several life-threatening diseases. A number of these enzymes are responsible for the regulation of various bioactive lipids, including sphingolipids, which are crucial for the structural integrity of cellular membranes and for facilitating intracellular and extracellular signaling pathways. Sphingolipid metabolism dysregulation, which can be caused by the deficiency of the enzymes responsible for the degradation of sphingolipids, results in a group of diseases known as Lysosomal storage disorders (LSDs). These disorders are characterized by the aberrant processing and degradation of substrates, impaired transport of lipids and excessive accumulation of non-degraded or partially degraded macromolecules inside lysosomes, which leads to cellular damage, cell death as well as organ dysfunction and degeneration [Marques and Saftig, 2019]. The clinical manifestations of LSDs are highly variable, ranging from mild to severe, depending on the residual enzymatic activity, rate of accumulation of macromolecules and particular genetic mutations involved, and are often associated with a significant

reduction in the quality of life. As a group of diseases, LSDs are quite common, with 1 in $5,000$ incidence rate. However, individual LSDs are rare, with incidence rates of 1 in $50,000$ to 1 in $250,000$ [Meikle et al., 1999]. Although there are several advancements in the treatment of LSDs using Enzyme replacement therapy (ERT), Substrate reduction therapy (SRT) or Gene therapy, most disorders cannot be cured and are managed symptomatically [Platt et al., 2018; Leal et al., 2020].

Fabry disease is one of the most common LSDs, which is caused by the deficiency of the enzyme α-galactosidase (AGAL) that is critical in the degradation of a sphyngolipid known as globotriaosylceramide (Gb3). The condition results in the systemic accumulation of Gb3 within lysosomes, impacting various tissues and organs, including kidneys, heart, and skin [Mehta and Hughes, 1993]. The pathological manifestations of Fabry disease are variable, ranging from skin lesions, renal dysfunction, to cardiomyopathy and neurological complications, profoundly impacting patient quality of life and longevity [Waldek et al., 2009; Burlina et al., 2011; Putko et al., 2015; Kim et al., 2016; Akhtar and Elliott, 2018]. The cornerstone of Fabry disease treatment is ERT, which involves the intravenous infusion of the recombinant version of the deficient or malfunctioning AGAL enzyme. This approach is aimed at supplementing the defective native enzyme, thus reducing the accumulation of Gb3 and alleviating the symptoms of the disease. However, the efficacy of ERT is variable, with some patients showing a suboptimal response to the treatment. Factors influencing the response to ERT include individual variations in dose amounts, the development of IgG antibodies against the recombinant enzyme and different responses of different disease phenotypes to the ERT treatment [Wanner et al., 2018]. Moreover, the financial implications of ERT are significant, with the most recent Pegunigalsidase alfa ERT costing on average £$118,187$ per year [Katsigianni and Petrou, 2022; National Institute for Health and Care Excellence (NICE), 2023]. This imposes a heavy burden on the healthcare system and society, especially in developing countries, and highlights the need for more cost-effective and efficacious therapies for Fabry disease.

Classical approaches to protein engineering

ERTs is an example of a protein engineering problem, a scientific endeavor aimed at designing novel proteins with desired properties. Protein engineering integrates principles and ideas from various scientific disciplines, such as biochemistry, molecular biology, computational biology, biophysics and, very recently, AI. The most traditional and widely used approach to protein engineering is Directed evolution (DE). DE is an iterative process of creating a large number of protein variants and selecting the ones with the desired properties. In this approach a starting gene is mutagenized to create a library of variants, which is screened for enzymes with an improvement of the sought-after property [Arnold, 1996]. Due to the incremental nature of enhancements achieved in each mutation cycle, multiple iterations are typically necessary to achieve significant improvements. Identification and isolation of promising candidates from a large pool of candidates in each cycle is the major challenge in DE approach. In contrast, the natural evolution or enzyme redesign approach leverages the inherent catalytic versatility of enzyme families. This method starts with a template enzyme that possesses a baseline level of activity for the target reaction. Under the natural evolution framework, protein engineering is aimed at redesigning the template enzyme by incorporating information from naturally occurring enzymes, typically within the same family as the template and known to catalyze the target reaction more efficiently. The main limitation of this approach is its reliance on the range of reactions naturally catalyzed by enzymes within a specific family, which may restrict potential modifications.

An alternative and more computational approach to protein engineering is the rational design protocol, which significantly relies on calculations from physics and requires a knowledge of a protein structure and its catalytic mechanism [Marshall et al., 2003]. This method begins with the modelling of the active site of an enzyme with the goal of stabilizing the transition state of the target reaction. Subsequent steps involve designing a protein scaffold around the modeled active site using various com-

putational tools for this purpose. The process concludes with targeted mutagenesis of amino acid residues to optimize the sequence and structure of the engineered protein, particularly focusing on the functionality of the active site [Kiss et al., 2013; Kuhlman and Bradley, 2019].

AI and Machine Learning in protein engineering

Machine Learning (ML), a transformative interdisciplinary field that integrates principles from computer science, statistics, and mathematics, has long been used in various scientific domains. Classical ML methods have been used for decades to solve various problems in biology, such as protein structure classification using Support vector machines (SVMs) or Random forests [Cai et al., 2001; Jain and Hirst, 2010], Hidden Markov models (HMMs) for sequence alignment and motif discovery [Durbin et al., 1998; Pachter et al., 2001] and functional genomics among others [Caudai et al., 2021]. The rapid advancements in DL models and architectures, coupled with the development of specialized hardware suitable for training these models, has further revolutionized the field of computational biology outperforming traditional ML methods in terms of performance and accuracy [Dahl et al., 2014]. A landmark achievement in the field of computational biology was the development of AlphaFold2 by DeepMind, a DL model for protein structure prediction, which demonstrated remarkable accuracy in the Critical Assessment of Structure Prediction (CASP) competition [Jumper et al., 2021] and set up a new standard for protein structure prediction.

Advancements in Natural language processing (NLP), an area of AI concerned with giving the computers the ability to understand and use human language, have also been transformative for the field of computational biology. It has been shown that Language models (LMs), when trained on extensive datasets, can learn various NLP tasks, such as machine translation, question answering and summarization in a self-supervised manner [Radford et al., 2019]. When scaled up, these models, known as Large language models (LLMs), can achieve state-of-the-art results on these tasks

[Brown, 2020]. Inspired by these results, LLMs have been adapted to biological applications, such as protein modelling and different DNA sites predictions [Ji et al., 2021; Lin et al., 2023; Nguyen et al., 2024]. For instance, it was shown that structural representations of proteins, called contact maps, emerge in the trained protein LLMs [Rives et al., 2021]. This discovery led to the development of a new model, called ESM-Fold, that can predict protein structure directly from its amino acid sequence without relying on any structural templates or evolutionary information, which is a common practice in the field and was used in AlphaFold2 model [Lin et al., 2023].

The rapid development of various generative models, such as Generative adversarial networks (GANs) [Goodfellow et al., 2014], Variational autoencoders (VAEs) [Kingma, 2013] or Diffusion models [Sohl-Dickstein et al., 2015; Ho et al., 2020] has enabled a new approach to protein engineering, both in terms of sequence and structure. For example, a new iteration of AlphaFold models, called AlphaFold3, relies on diffusion-based modules and is able to predict joint structures of complexes, including proteins, nucleic acids and small molecules better than specialized models [Abramson et al., 2024]. Conditional generation, a technique where a generative model is conditioned on some input data, to generate a new sample with desired properties, presents a formidable potential for the creation of novel protein sequences with desired characteristics. While an array of advanced generative models exists, focusing on either sequence or structure aspect of protein design, most of them are validated only *in silico* and lack experimental validation. Experimentally verified models primarily focus either on relatively short proteins under 200 amino acids or highly diverse protein families, such as Malate dehydrogenase (MDH) or Green fluorescent protein (GFP) which could tolerate a large number of mutations without losing their function. As such, empirical evidence demonstrates that conditional generation facilitates the design of novel proteins across distinct protein families through the application of control tags [Keskar et al., 2019; Madani et al., 2023]. In another study, conditioning the latent space of a VAE model on the solubility of the protein was shown to improve the solubility of the

generated sequences [Hawkins-Hooker et al., 2021].

Thesis as part of a big interdisciplinary project

This dissertation forms a part of an extensive interdisciplinary project dedicated to developing novel ERTs for the treatment of Fabry disease. My contributions, which are presented in this thesis, are focused on the development of novel generative DL methods. They are designed to facilitate and streamline the construction of screening libraries of AGAL enzymes for any downstream experimental validation. The thesis is divided into three main parts, each of which is a separate chapter and touches upon a different aspect of the project. Specifically, we first developed a baseline generative DL model based on VAE architecture that effectively learns evolutionary constraints from a set of homologous sequences for a highly conserved AGAL protein. We then focused on a conditional generative model, which was shown to be capable of generating sequences with desired thermostability. This model was validated in *E. coli* on a conditionally essential gene argB that encodes *E. coli* phosphotransferase N-acetyl-L-glutamate kinase (*Ec*NAGK), which is a part of the arginine biosynthesis pathway in *E. coli*. Finally, we evaluated the rationality and necessity of combining sequence and structural information in a single model with the objective of recoding epitopes of AGAL to potentially reduce immunogenic response of the recombinant enzymes. We showed that structural information can be beneficial in the epitope redesign problem, obtaining a better performance than with sequence-only models. In the next section, we present and discuss the detailed overview of the thesis, introducing the reader to the main concepts and ideas of the work and providing a motivation for the research conducted.

1.2 Thesis Outline

This thesis is centered on the development of innovative ERTs for Fabry disease utilizing generative DL techniques. The focus is on the computational design of therapeutical human AGAL enzymes with improved properties, such as thermostability and reduced immunogenicity. The selected candidates from the designed AGAL libraries undergo experimental validation in *Pichia pastoris* and results are compared with the wild-type enzyme. Each chapter is devoted to a progressively more complex approach of the *in silico* design of new AGAL enzymes. Chapter 2 introduces a foundational VAE model that effectively learns evolutionary constraints from a small set of homologous sequences for a human AGAL protein. The model employs Temporal convolutional networks (TCNs) for the effective processing of long protein sequences in the dataset and utilizes a Dirichlet distribution for the latent space. Dirichlet distribution acts as a conjugate prior for categorical distribution used in modelling amino acid sequences. Our model's efficacy was confirmed on standard benchmark datasets for mutation effect variant prediction tasks, demonstrating comparability to state-of-the-art models in performance, while being significantly more compact. Additionally, a thorough hyperparameter analysis was performed, demonstrating the superiority of Dirichlet distribution over Gaussian distribution for the latent space. Ultimately, we demonstrated that our model can be used for the generation of novel AGAL sequences and, to the best of our abilities, compared the model performance with the DE study performed on the same AGAL protein. Similar to the mutation effect prediction task, we showed that Dirichlet latent space closer approximates the results of DE.

Chapter 3 is focused on a conditional generative model, which was shown to be capable of generating sequences with desired properties. A prevalent objective in protein engineering, particularly important in the context of ERT, is the enhancement of enzyme thermostability. Given that recombinant AGAL enzymes are administered intravenously to patients, their stability is critical for maintaining their efficacy. Therefore, augmenting enzyme thermostability is a desirable property. In our generative model,

we simultaneously mapped sequence data with thermostability information through a common latent space. This mapping allowed us to efficiently explore local minimas in the latent space, which, by design, correspond to sequences with higher degree of stability. We validated our model by designing and testing 40 mutated variants of a semi-essential protein *Ec*NAGK. The semi-essential nature of this protein allowed us to rapidly and efficiently test the cell viability with a mutated argB gene in a live-dead assay. We showed a 85% success rate in the lab results, with some protein sequences having up to 9 mutations, and showing near wild-type levels of cell growth rate.

Chapter 4 is dedicated to the redesign of epitopes of AGAL to reduce immunogenic response of recombinant enzymes. In this chapter, we evaluated the rationality and necessity of combining sequence and structural information in a single model for the task of epitope redesign. We also evaluated the effect of pre-training and fine-tuning a generalist model on a specific set of proteins. By computationally evaluating the immunogenicity of designed sequences, we showed that structural information significantly outperforms sequence-only models in the epitope redesign problem.

Chapter 2

Dirichlet latent modelling for protein sequence generation

Disclaimer

This chapter is based on a manuscript accepted at *Nature Communications*. The manuscript is titled *Dirichlet latent modelling enables effective learning and sampling of the functional protein design space* and is authored by Evgenii Lobzaev and Giovanni Stracquadanio.

2.1 Introduction

Recent advances in DNA synthesis and sequencing technologies coupled with high-throughput, automated experimental screening platforms are enabling the engineering of proteins with desired function and properties suitable to address biotechnology and biomedical challenges [Huang et al., 2016].

Nonetheless, the protein design space is exponentially large and with most regions harboring non-functional biomolecules. Therefore, there has been a strong interest in developing methods, both experimental and computational, to explore the neighbor-

hood of known functional proteins to identify new variants likely to be functional with the aim of reducing the burden of downstream experimental validation. Experimental methods are usually based on the Directed evolution (DE) framework [Romero and Arnold, 2009], a process of mutation and selection that allows to explore variants of a known protein in an unbiased fashion. However, in a DE campaign, only a fraction of the variants are functional, which makes the whole process expensive and difficult to scale for large molecules.

The development of biophysical models of protein folding, instead, have boosted the use of computational approaches to rationally design proteins and variant libraries [Kiss et al., 2013]. However, these models can only approximate the physical principles underpinning protein folding, and thus are limited in designing proteins characterized by complex properties, like flexibility, which are not directly related to their thermodynamic properties.

High-order information underlying complex protein features can now be learned by leveraging peta-scale information available for known protein sequences in public databases [Rives et al., 2021]. Machine Learning (ML) and, more recently Deep learning (DL), have been able to exploit this information and are propelling a paradigm shift in protein engineering. While initial ML models required experimental data to identify beneficial mutations, recent auto-regressive models, like Variational autoencoders (VAEs) [Kingma, 2013; Wu et al., 2021], have achieved state-of-the-art performances in mutation effect prediction [Hopf et al., 2017; Riesselman et al., 2018; Frazer et al., 2021] and variant library design tasks [Shin et al., 2021; Repecka et al., 2021; Giessel et al., 2022] using only sequence information. VAEs achieve these results by mapping known protein families or homologs into parametric probabilistic distributions, which can then be used to either design new proteins or evaluate the likelihood of a variant to belong to the input sequence dataset, and hence retaining their characteristic features. From a biotechnology point-of-view, VAEs have the potential to generate large libraries of functional variants to screen, compared to randomized experimental

approaches, like DE, where most variants are non-functional.

Interestingly, while there have been a lot of efforts in developing efficient architectures to learn distribution parameters and design new sequences, protein families have always been modelled with a standard multivariate Gaussian distribution. However, decades of research in homology search have shown that the Dirichlet distribution is significantly more powerful at modelling amino acid frequencies and relationships, and in finding remote homologous sequences [Sjölander et al., 1996], suggesting that using this distribution could substantially improve current auto-regressive models performances in generating large library of diverse protein variants.

Here we hypothesized that modelling protein families using a Dirichlet distribution will enable auto-regressive models to i) identify substantially different sequences with similar biochemical and structural features to those of a known wildtype protein and ii) to capture biological and fitness constraints while being computationally tractable. To prove that, we developed a new model, called Temporal Dirichlet Variational Autoencoder (TDVAE), which maps protein homologs on a Dirichlet distribution and uses Temporal convolutional networks (TCNs) [Bai et al., 2018] to learn distribution parameters and sample new protein sequences from it.

We then assessed the performances of TDVAE as a mutation effect prediction tool on an extensive dataset of mutagenesis experiments, and showed that it achieves comparable to state-of-the-art results while being 90% smaller than the current best unsupervised model. We also performed an extensive hyperparameter analysis to show the robustness of our model to parameter settings and its superior performances to the same architecture when modelling the latent space as a canonical multivariate Gaussian distribution. Finally, we used TDVAE to design a library of human α-galactosidase (AGAL) variants, a complex lysosomal enzyme whose inactivation causes Fabry disease and with a well characterized mutational landscape [Platt et al., 2018]. Our results show that TDVAE generates a diverse library of variants while retaining biochemical and structural properties of the human enzyme and avoiding pathogenic mutations.

Moreover, TDVAE identifies mutational hotspots associated with improved enzymatic activity and biochemical properties, while not requiring any experimental information.

Taken together, TDVAE provides a new effective and efficient platform to design libraries of functional proteins using only sequence information.

2.2 Methods

A deep generative learning framework for protein engineering

Protein engineering requires learning how to sample the protein design space to identify amino acid sequences associated with a desired catalytic function. Here we hypothesize that the design space has a statistical structure, whose functional form and corresponding parameters are unknown but can be learned from known sequences readily available in protein databases.

We hereby assume that the probability of observing a protein sequence x depends on a latent random variable z, such that the joint probability distribution of x and z can be factorized according to equation 2.1, where $p_\theta(x|z)$ and $p_\theta(z)$ are parametric distributions.

$$p_\theta(x,z) = p_\theta(x|z)p_\theta(z) \tag{2.1}$$

Thus, a protein sequence can be considered the result of a generative process, which involves sampling a random variable $\hat{z}$ from $p_\theta(z)$, which in turn is used to build a sequence $\hat{x}$ by sampling from the conditional probability distribution $p_\theta(x|\hat{z})$; in our case, $\hat{z}$ can be thought as a random variable encoding properties specific to the proteins in the training dataset, such as function or amino acid composition.

However, learning the parameters θ of this class of models is usually intractable, since we cannot evaluate or differentiate the marginal likelihood $\int p_\theta(x|z)p_\theta(z)dz$ and the posterior probability $p_\theta(z|x) := \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$. Here we addressed these issues by using a Variational autoencoder (VAE) architecture, where Neural Networks (NNs) are

used to approximate $p_\theta(x|z)$ and $p_\theta(z|x)$, and Stochastic Variational Inference (SVI) to learn model parameters [Kingma, 2013]. Specifically, in a VAE framework, $p_\theta(z|x)$ is approximated by a parametric recognition model, $q_\phi(z|x)$, which acts as a probabilistic encoder taking in input a sequence x and returning a distribution over the possible value of z. Thus, $p_\theta(x|z)$ acts as a probabilistic parametric decoder, which takes in input a sample z and returns a distribution over the possible values of x.

Here we argued that the ability of VAEs to effectively sample the protein design space and generate new functional variant depends on the parametric family used to model the latent space, and the use of robust Neural Networks (NNs) for computing $p_\theta(z|x)$ and $q_\phi(z|x)$.

VAEs have traditionally assumed the prior distribution of the latent space to be continuous, ultimately leading to the ubiquitous use of a Gaussian prior distribution. However, protein sequences inherently follow a multivariate multinomial distribution, which represents the probability of observing a given amino acid at any given position. Importantly, the conjugate prior of the multinomial distribution is the Dirichlet distribution, thus using a Dirichlet latent space represents a necessary step to build effective VAEs for sequence modelling. Moreover, from an engineering perspective, the design space is expected to be highly multimodal, as a result of the biophysical forces controlling protein folding; this multimodality cannot be modelled by a multivariate Gaussian distribution [Joo et al., 2020], thus we hypothesized that modelling the latent space using a Dirichlet prior could be beneficial in generating functional proteins.

Computing $p_\theta(z|x)$ and $q_\phi(z|x)$ over sequences is usually intractable, thus a plethora of NNs have been proposed as robust approximations, including Recurrent Neural Network (RNN) [Mikolov et al., 2011], Long Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] and Gated Recurrent Unit (GRU) [Cho et al., 2020]; however, as the length of the sequences increases, their ability to learn long-range relationships between amino acids decreases [Pascanu et al., 2013], a drawback that makes them unsuitable to handle long amino acid sequences. Moreover, these architectures are

computationally expensive to train [Wu, 2016], as they cannot be readily parallelized, and thus unsuitable to scale over large sequence datasets.

Therefore, we used an alternative architecture for both the encoder and decoder, called Temporal convolutional network (TCN), which overcomes these limitations and can be efficiently trained [Bai et al., 2018]. TCNs take sequences in input and return new ones of the same length. Sequences of the same length are easily obtained by using a standard 1-dimensional convolutional layer with zero-padding to keep the output of the subsequent layers equal to the input length. To condition the probability of a residue on the previously observed ones, TCNs use causal convolution, where the residue at position t is obtained by applying convolution only with elements at positions $t, \ldots, 0$ in the previous layers. However, obtaining an effective memory usually requires stacking multiple convolutional layers, thus making vanilla TCNs inefficient to train. The problem has been recently addressed by using dilated convolution. Let x be a sequence of length n and f a kernel of size k, d a dilation factor that represents the distance between two elements of the input that are used to produce one element of the output. Then the dilated convolution F is computed according to the equation 2.2.

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-di} \tag{2.2}$$

Stacking multiple temporal convolution layers with exponentially increased dilation factors, allows obtaining full sequence coverage while keeping the number of layers logarithmic in sequence length. Specifically, to reduce the number of hyperparameters to optimize, we determined the total number of stacked layers according to equation 2.3.

$$n = \left\lceil \log_2 \left[\frac{(L-1)}{2(k-1)} + 1 \right] \right\rceil \tag{2.3}$$

Here, L is the length of the longest sequence and assuming dilation factor to be 2^i, where i is the layer index, starting from 0.

Variational inference of model parameters

In a Variational autoencoder framework, NNs are used to compute the variational parameters ϕ for a fixed family of probability distributions $q_\phi(z|x)$ and model parameters θ for conditional likelihood $p_\theta(x|z)$. Here, we use SVI to find an approximate solution to the problem of maximizing the marginal likelihood $\int p_\theta(x|z)p_\theta(z)dz$ by maximizing the Evidence Lower BOund (ELBO) w.r.t both model parameters θ and variational parameters ϕ as follows:

$$\mathcal{L}(\phi,\theta) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p_\theta(z)) \rightarrow \max_{\theta,\phi} \tag{2.4}$$

In equation 2.4 KL is the Kullback-Liebler divergence, and the expected conditional likelihood $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ is optimized with respect to maximum log-likelihood, as the probability distribution $p_\theta(x|z)$ over the amino acid space is categorical.

Depending on the choice of parametric families for $q_\phi(z|x)$ and $p_\theta(x|z)$ computing the expected conditional likelihood can be challenging, is often intractable and requires numerical approximations, whereas KL divergence can be computed analytically. Ultimately, we need to compute a gradient of the expected conditional likelihood w.r.t parameters ϕ: $\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$. However, in this case, the gradient computation cannot be moved under the expectation operator, since the expectation is done w.r.t $q_\phi(z|x)$. Nonetheless, for many parametric distributions, a number of low variance gradient estimators have been proposed, such as those based on pathwise derivatives, alternatively known as reparametrization trick [Kingma, 2013; Rezende et al., 2014], but they cannot be applied to the Dirichlet distribution, unless Gaussian-based approximations are used at the cost of losing the characteristic properties of the Dirichlet distribution [Joo et al., 2020]. The general idea of the generalized reparametrization trick that applies to majority of continuous distributions is based on implicit differentiation that results in $\nabla_\phi z$ term that can be computed using only Probability Density Function

(PDF) $q_\phi(z)$ and derivatives of Cumulative Density Function (CDF) $\nabla_\phi F(z|\phi)$ or its numerical approximation as follows:

$$\begin{aligned} \nabla_\phi \mathbb{E}_{q_\phi(z)}[f(z)] &= \mathbb{E}_{q_\phi(z)}[\nabla_z f(z) \nabla_\phi z] \\ \nabla_\phi z &= -\frac{\nabla_\phi F(z|\phi)}{q_\phi(z)} \end{aligned} \tag{2.5}$$

The exact derivation of $\nabla_\phi z$ for Dirichlet distribution are provided in [Figurnov et al., 2018; Jankowiak and Obermeyer, 2018], with the latter being an official implementation of PyTorch library.

Data collection

Homologous sequences for the human α-galactosidase (AGAL) enzyme where obtained by searching the UNICLUST30 database using HHBLITS [Remmert et al., 2012] with parameters `-Z 10000 -B 10000 -e 0.001 -all` and the filtered to retain only sequences with 80% query coverage and 50% query identity (`-id 100 -cov 80 -qid 50`). The final dataset consisted of 1,746 sequences, from which we removed gaps and insertions.

AGAL library analysis

Biochemical properties of the variants in the AGAL library were all computed using the `ProtParam` module of the BIOPYTHON package [Cock et al., 2009]. Structures for selected variants of the library where predicted using ESMFOLD [Lin et al., 2023] and successively relaxed by energy minimization as implemented in the OPENMM package [Eastman et al., 2017] using the Amber14 force field and adding a harmonic potential energy term to restrain Cα atoms position. The downstream normal mode analysis was performed using the BIO3D package [Grant et al., 2021].

Data Availability

The data supporting the findings of this study are available on Zenodo at:
`https://doi.org/10.5281/zenodo.13269310`.

Code Availability

The software is available at the following url:
`https://licensing.edinburgh-innovations.ed.ac.uk/product/proton`.

2.3 Results

We assessed the performances of our approach by using TDVAE as a mutation effect prediction and as a protein engineering tool, comparing and contrasting experimental results with state-of-the-art methods and experimental data. Here we hypothesized that a model able to predict mutation effect, should be sufficiently powered to design functional protein variants by learning sequence features associated with known functional homologous sequences.

TDVAE performance on predicting protein mutation effects

We first assessed TDVAE performances as a mutation effect prediction model using 19 widely used mutagenesis datasets [Hsu et al., 2022a]. Each dataset consists of a library of experimentally tested variants of a given wildtype protein, with each variant annotated with a fitness score quantifying different phenotypes, ranging from enzyme activity to cell viability, normalized and log transformed such that the wildtype fitness is 0. Out of 19 mutational sets 16 contain single-point mutations, whereas the remaining 3 contain mutants with multiple mutations with respect to the wildtype. The input information provided to a model is a Multiple Sequence Alignment (MSA) generated as part of the EVMUTATION pipeline [Hopf et al., 2017], which contains homologs

of the wildtype sequence. Given a mutagenesis dataset and a model which outputs an effect score measuring the functional impact of one or more mutations, the model performances are then reported in terms of Spearman's correlation between the experimental fitness scores and the predicted mutation effect scores.

A recent comparative analysis of state-of-the-art models for mutation effect prediction showed that the unsupervised probabilistic model DEEPSEQUENCE [Riesselman et al., 2018] consistently reported the best performance across all proteins in our dataset. DEEPSEQUENCE outputs a mutation effect score for a given variant sequence as the ratio $S_m = \log[p(x_{mut}|\theta)/p(x_{wt}|\theta)]$, where p is replaced by the Evidence Lower BOund (ELBO); in practice, S_m represents the log-likelihood of a variant x_{mut} relative to the wildtype sequence. This score has been shown to be predictive of mutation effects and can be learned without fitting the model to experimental data [Hopf et al., 2017].

Here we compared TDVAE performances against DEEPSEQUENCE, using the available PYTORCH implementation; specifically, we used the proposed Bayesian decoder and the default parameters as reported in the original study [Riesselman et al., 2018; Frazer et al., 2021]. To perform a fair comparison and quantify the contribution of our Dirichlet latent space modelling, we adapted TDVAE to use the same sparse one-hot encoding layer as in DEEPSEQUENCE, to avoid any representational bias. We then used a single block of stacked dilated causal 1D convolutional layers with kernel size 3, intermediate channel size of 128, 20% dropout for regularization, and a 50-dimensional latent space, with a symmetric Dirichlet distribution with $\alpha = 1.0$ as prior, whereas the number of causal convolutional layers is computed according to Eqn.2.3. Unlike DEEPSEQUENCE, we did not use Variational Inference (VI) on the decoder weights or any structured parametrization in the final layer, but again relied on a single block dilated convolutional layer [Shin et al., 2021]. For each dataset, we trained each model 5 times with 5 different random seeds using a mini batch of 256 sequences, and then computed 2,000 ELBO samples per mutant to estimate model-specific mu-

tation effect scores to be correlated with experimental data. Since these models are usually computationally taxing to train, we also tested a version of TDVAE, dubbed LOW MEM, which uses a mini batch of only 4 sequences, as an alternative that can be readily adopted in consumer GPU hardware.

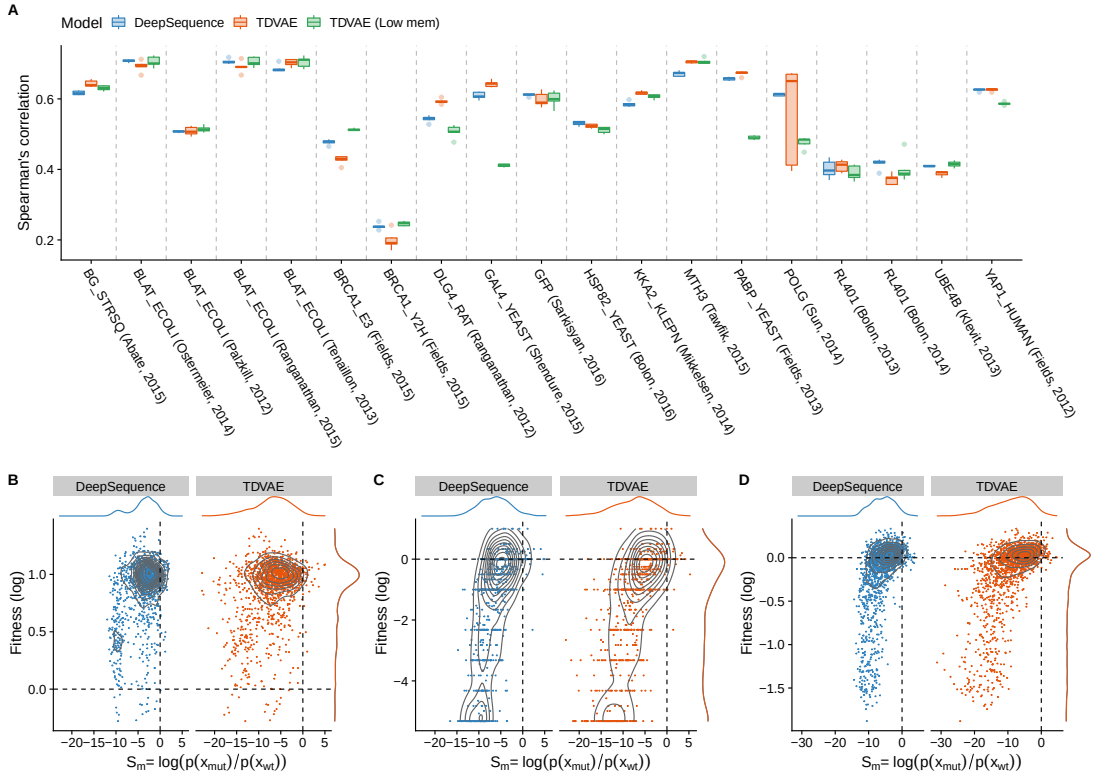

Figure 2.1: **Performance on mutation effect prediction.** A) For each protein dataset, we report the performance of TDVAE and DEEPSEQUENCE in terms of Spearman's rank correlation computed over 5 independent runs and using the best parameters associated with the best validation loss. We denote as TDVAE 'Low mem' the TDVAE model trained using a mini-batch size of 4 sequences. B) Correlation between evolutionary scores computed by DEEPSEQUENCE and TDVAE for the *BRCA1* dataset, C) the *BLAT* (Tenaillon, 2013) dataset, and D) the *DLG* dataset; the dashed lines represent the wildtype fitness level in the experimental data (y-axis) and as predicted by the models (x-axis).

Experimental results showed that, in 17 out of 19 datasets, TDVAE performed better than DEEPSEQUENCE with up to 6% increase in Spearman's correlation, as for

the *POLG* dataset (see Figure 2.1A); specifically, the best performance was obtained by TDVAE on 8 datasets and by TDVAE LOW MEM on 9 datasets. When compared the robustness of models' performance, that is the best performance on average across 5 independent runs, performances were comparable; specifically, TDVAE achieved the best average performance on 7 datasets, TDVAE LOW MEM on 6 datasets and DEEPSEQUENCE on the remaining 6 datasets. It is important to note that differences in performances are limited, albeit improvements of up to 5% in Spearman's correlation were found depending on the protein, e.g. *DLG4*. Interestingly, out of the 19 datasets, the LOW MEM version generally performed poorly on only 3 datasets, i.e. the difference in correlation is more than 10% compared to the best model, namely *GAL4*, *POLG* and *PABP*, suggesting that even a less optimal training process can still produce satisfactory results and can also be beneficial as smaller batches might act as regularizer during training.

Differences in performance between the models could not be apportioned to either the size of the MSA, the number of mutations or number of homologs used to train the model. However, we observed that both DEEPSEQUENCE and TDVAE had a similar performance trend on the same datasets, suggesting that mutation effect prediction might vary significantly depending on the specific protein and phenotype studied. This becomes apparent when analyzing the *BRCA1* dataset (see Figure 2.1B), which is the one where both models had their worst performance; specifically, they both report most mutations as detrimental rather than beneficial. Conversely, on the *BLAT* dataset (see Figure 2.1C), which is the one with the highest correlation for both models, and on the *DLG4* dataset (see Figure 2.1D), where we observed the largest TDVAE improvement, both models accurately capture the fitness variability in the dataset.

While we showed that TDVAE achieves comparable to state-of-the-art performance, we wanted to test whether these results could be related to our model being more complex than DEEPSEQUENCE. Thus, we computed the number of model parameters for both models on each of the 14 distinct proteins in our benchmark dataset.

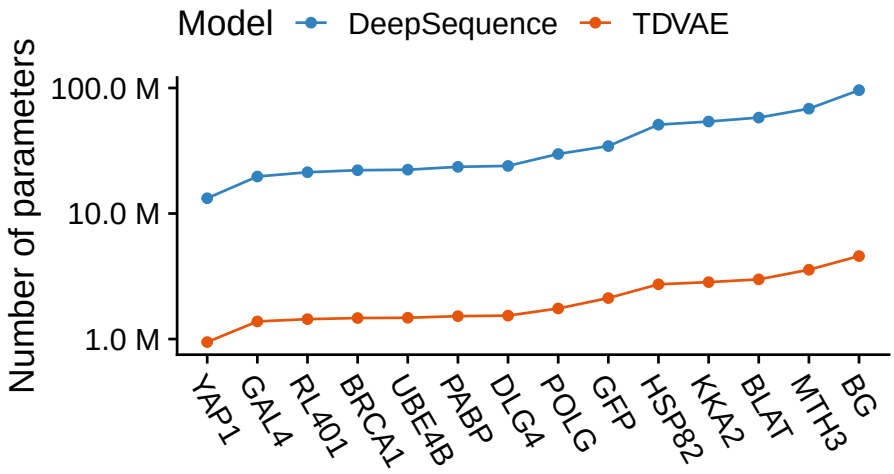

Figure 2.2: **Analysis of TDVAE and DEEPSEQUENCE model complexity**. The x-axis reports the protein analyzed in our benchmark dataset, while the y-axis the number of parameters of each model. Here we found that TDVAE has approximately 94% less parameters than DEEPSEQUENCE on average.

Here we found TDVAE to be $\sim 94\%$ smaller than DEEPSEQUENCE on average, ranging from $\sim$945K parameters for the *YAP1* dataset to $\sim$4.6M parameters for *BG* dataset (see Figure 2.2), suggesting that TDVAE performance cannot be apportioned to model over-parametrization but rather to the different modelling of the latent space.

Taken together our results suggest that TDVAE is a robust model for mutation effect prediction, which achieves good performance while being substantially less complex than other approaches.

TDVAE outperforms Gaussian latent modelling and is robust to parameters settings choices

In our previous experiments we used a TDVAE configuration comparable to the one used by DEEPSEQUENCE, and showed it obtains state-of-the-art performance while requiring the learning of a significantly smaller number of parameters. We then decided to test how it compares with the same architecture when the latent space is modelled

using a canonical Gaussian distribution, a model we dubbed Temporal Gaussian Variational Autoencoder (TGVAE), and whether TDVAE performances were sensitive to parameters choice. To do that, we trained both TDVAE and TGVAE using different configurations; specifically, we varied the number of blocks of stacked dilated convolutions, the intermediate channel size, the latent space dimensionality, and the kernel size. We also tested two scenarios with sub-optimal training procedures, following up on the satisfactory TDVAE LOW MEM performances in the previous experiments, by using i) a mini-batch size of only 8 sequences, and ii) training for 40,000 batch updates. Taken together, we constructed 9 different configurations.

We conducted the hyperparameters analysis by training each model configuration on 4 proteins (8 mutagenesis datasets total), namely *YAP1*, *DLG4*, *BRCA1*, and *BLAT*, which vary with respect to the size of the MSA size and initial model performances, in order to work on a small but diverse subset of proteins. Finally, for each model, we computed the best and average performances over 3 independent runs.

Experimental results showed that TDVAE is more robust than TGVAE to parameters settings, i.e. performs better on average, for 5 out the 8 datasets, ranging from 3.10% for YAP1 to 0.02% for the BLAT (Ranganathan, 2015) dataset. Importantly, average performance improvement for TDVAE can be as high as 8.04% for BRCA-e3 dataset, while performing worse by less than 0.5% on the other 3 datasets on average (see Supplementary Table A.1). Similar trend was observed when considering the best absolute performance of each model, where TDVAE outperforms TGVAE in 6 out of 8 datasets considered regardless of the configurations used, with average improvement of $\approx 2\%$, ranging from 0.21% for the DLG4 dataset to 6.42% for the BRCA1-e3 dataset (see Supplementary Table A.2).

Taken together, our analysis shows that TDVAE performs robustly with respect to parameters settings, and that, in general, the use of the Dirichlet distribution is associated with better predictive power, albeit the extent of improvement depends on the specific protein studied.

TDVAE generates variants of the human α-galactosidase enzyme with wildtype properties

We have already shown that our model can robustly learn the fitness landscape of a protein in unsupervised fashion, thus we hypothesized that TDVAE should be able to generate new variants that are similar at sequence and structural level with respect to a target wildtype protein. However, evaluation of proteins designed by generative models is usually difficult *in silico* because experimental mutagenesis datasets usually encompass single locus mutations, whereas generative models can design very diverse proteins with a potentially high number of mutations. Therefore, as a testbed, we looked for protein coding genes associated with Mendelian diseases and with a well-characterized mutational landscape, such that we can have a more unbiased approach to evaluate *in silico* whether the designed variants could be functional.

Here we focused on the human α-galactosidase (AGAL) lysosomal enzyme (Uniprot ID: P06280), a 429 amino acid long homodimeric protein responsible for hydrolyzing the terminal α-galactosyl moieties from glycolipids and glycoproteins [Kornreich et al., 1989]. Inherited loss of function mutations in the *GLA* gene, which encodes this enzyme, leads to accumulation of partially metabolised glycosphingolipids, particularly globotriaosylceramide (Gb3) and globotriaosylsphingosine (lyso-Gb3) in multiple cells. Progressive accumulation of glycosphingolipids, a condition known as Fabry disease, ultimately leads to organ damage, particularly heart and kidney, and premature death [Platt et al., 2018]. Enzyme replacement therapies (ERTs), which consist in the infusion of a recombinant version of the AGAL enzyme, are the current standard of care. However, current ERTs have poor catalytic activity, are unstable in blood, and often are immunogenic [Parenti et al., 2021; Xu et al., 2015]. Generating large variant libraries of AGAL enzymes to screen for desirable therapeutic properties represent an attractive approach to address these issues. However, the α-galactosidase (AGAL) mutational landscape, one the most well characterised among all lysosomal

storage diseases, consists of 216 known validated single point mutations covering more than 50% of the protein sequence, suggesting that designing new recombinant AGAL enzymes is a challenging task.

Library design

To design new AGAL variants with TDVAE, we obtained homolog sequences from the UNICLUST30 database using HHBLITS (see Methods), which gave us an initial dataset of 1,746 sequences. We then further processed this dataset to remove sequences with non-canonical amino acids (35 in total), and then split the remaining sequences into training and validation datasets using a 90/10 ratio. We then used TDVAE in a configuration similar to the one used for mutation effect prediction, albeit we introduced a 32-dimensional embedding layer to better deal with variable length sequences. We then used a 32-dimensional Dirichlet latent space, in order to work with a compact latent space, whereas we used a 128-dimensional channel size for the causal convolutional layers and a 20% dropout to prevent over-fitting. Finally, we trained the model for 5,000 epochs using mini-batches of 256 sequences.

We first tested whether the AGAL sequence landscape could be effectively mapped to a Dirichlet latent space. To do that, we analyzed the empirical standard deviation of each latent space component, defined as $\mathrm{diag}(\mathrm{cov}_x[\mathbb{E}_{q_\phi(z|x)}[z]])$ [Burda et al., 2015]. We considered a component as being active if its standard deviation was greater than 0.03, which represents the expected standard deviation if a model assigns random values to a latent component. Here we found TDVAE effectively uses all the 32 components, albeit with a different relative importance, suggesting that different embeddings are used to encode different sequences (see Figure 2.3A). We further validated this finding by performing Principal Component Analysis (PCA) of the 32-dimensional expected value parameter of the variational posterior distribution, $q_\phi(z|x)$, for the sequences in the training set. PCA revealed distinct clusters associated with the 3 largest classes found in the training set (mammals, birds and fish), with the emergence of a core

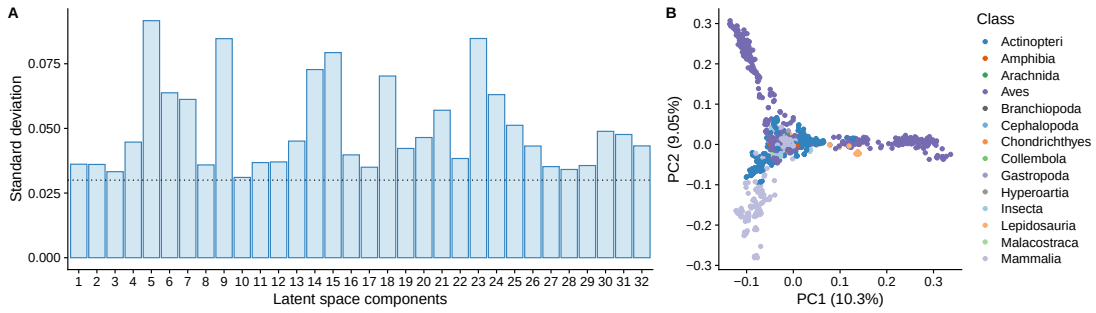

Figure 2.3: **TDVAE learns the landscape of α-galactosidase (AGAL) homologs**. A) Activation plot for the 32-dimensional latent space learned by TDVAE. The x-axis represents each latent component and y-axis the standard deviation of the corresponding value learned during training; the dashed line represents the expected standard deviation if the model assigns random values to each component. B) Principal component analysis of the $\mathbb{E}_{q_\phi(z|x)}[z]$ embeddings generated by TDVAE for the sequences in the training set and labelled according to the corresponding lineage class.

common to all embeddings which is consistent with sequences being all members of the α-galactosidase family (see Figure 2.3B).

Sequence analysis

After training the model, we generated variants of the wildtype human AGAL enzyme, by first passing the wildtype sequence in input to our encoder to obtain the parameters of the associated region of the Dirichlet latent space, and then sampled 20,000 independent latent vectors to be decoded into sequences by picking the most likely amino acid at each position. Generated sequences were further processed using BLASTP by filtering out those with E-value$>$0.001 and query coverage below 75%. All 20,000 samples passed these filters and were then considered for downstream analysis.

Variants have an average of ≈48 mutations (see Figure 2.4A), albeit not localized at random but clustered in specific regions of the enzyme (see Figure 2.4B). The most variable region is located at the N-terminus (1-32 residues), where the signaling peptide is encoded: this is expected as the signaling peptides change significantly across

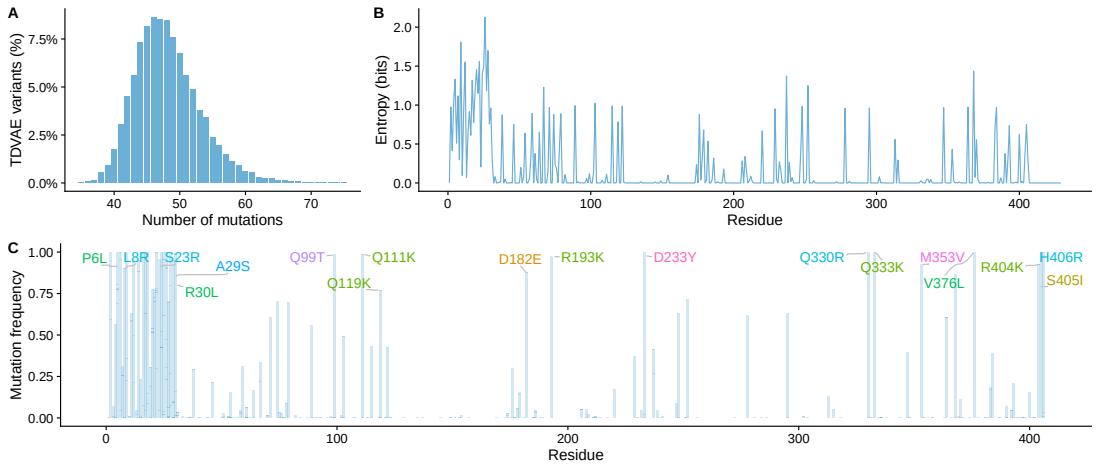

Figure 2.4: **TDVAE generates a diverse library of AGAL variants**. A) Distribution of the number of mutations per variant in the library of 20,000 sequences generated by TDVAE. B) Residue-level entropy of TDVAE variants. C) Most frequent mutations in TDVAE variants.

species [Li et al., 2009]. Importantly, the binding site region, spanning residue 203 to 207, is highly conserved, which confirms that the model is not introducing any obvious inactivating mutations. Using the BLOSUM62 substitution matrix, we analyzed the mutations introduced in at least 75% of the generated sequences: here we found that TDVAE always introduces conservative mutations except in three locus, with the most non-conservative mutation being `D233Y` (see Figure 2.4C), suggesting that our model generates diversity by introducing putative non-detrimental changes.

We then characterized the biochemical properties of our variants and compared them with those of the wildtype enzyme (see Figure 2.5A). Here we found our variants to have similar molecular weight, flexibility and isolectric point compared to the wild-type, albeit they are predicted to be more stable, suggesting that our library of variants retains the biochemical features of the wildtype.

We then looked whether any of the introduced mutations are associated with significant reduction of enzymatic activity and disease phenotypes. The generated library did not contain pathogenic mutations, albeit 2 mutations were found in more than 5% of

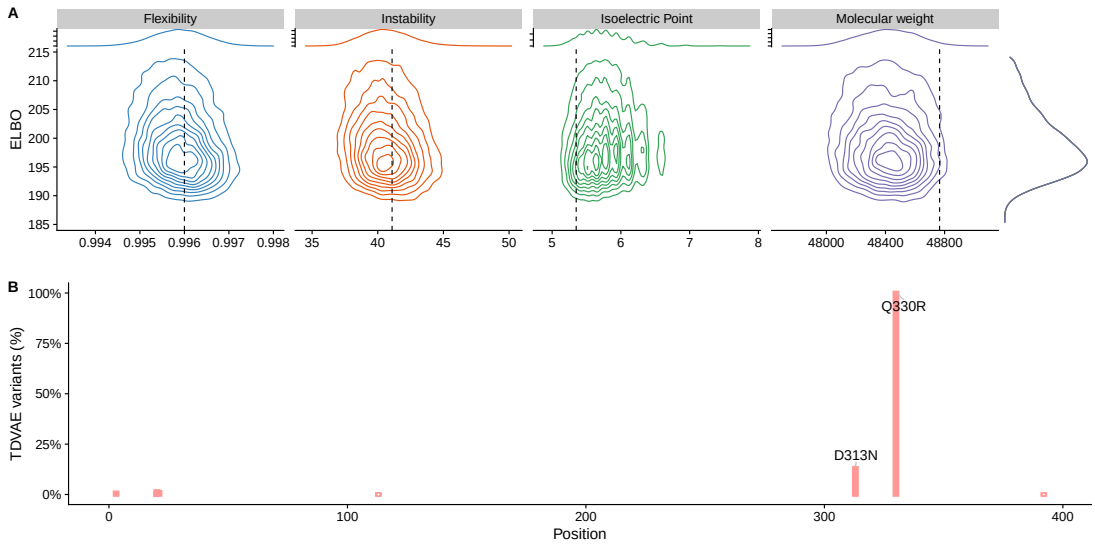

Figure 2.5: **AGAL variants have wiltype like biochemical properties**. A) Analysis of biochemical properties for the TDVAE variant as a function of the associated ELBO. B) Frequency of mutations in the TDVAE variants associated with Fabry disease or changes in enzymatic activity.

the sequences (see Figure 2.5B), respectively `Q330R` and `D313N`; both mutations were found in Fabry patients [Lukas et al., 2016], but the most frequent one, `Q330R`, is not associated with a physiologically relevant accumulation of Gb3, whereas the second one has an unclear phenotype; interestingly, both mutations are reported as 'benign' by POLYPHEN2, a standard tool for assessing the impact of protein mutations.

Taken together, our sequence analysis shows that TDVAE can generate a diverse library of AGAL enzymes, while retaining the functional features of the wildtype sequence.

Structural analysis

We also studied our variant library at the structural level to identify potential functional changes and differences in stability compared to wildtype. To do that, we ranked sequences by their associated ELBO and selected the top 20 sequences from 5 equally spaced deciles as a way to select diverse variants across the library. We predicted the

structure of the 100 selected sequences (see Methods) and used them for downstream comparative and motion analysis using the human wildtype AGAL structure (PDB ID: 1R46) as the reference structure.

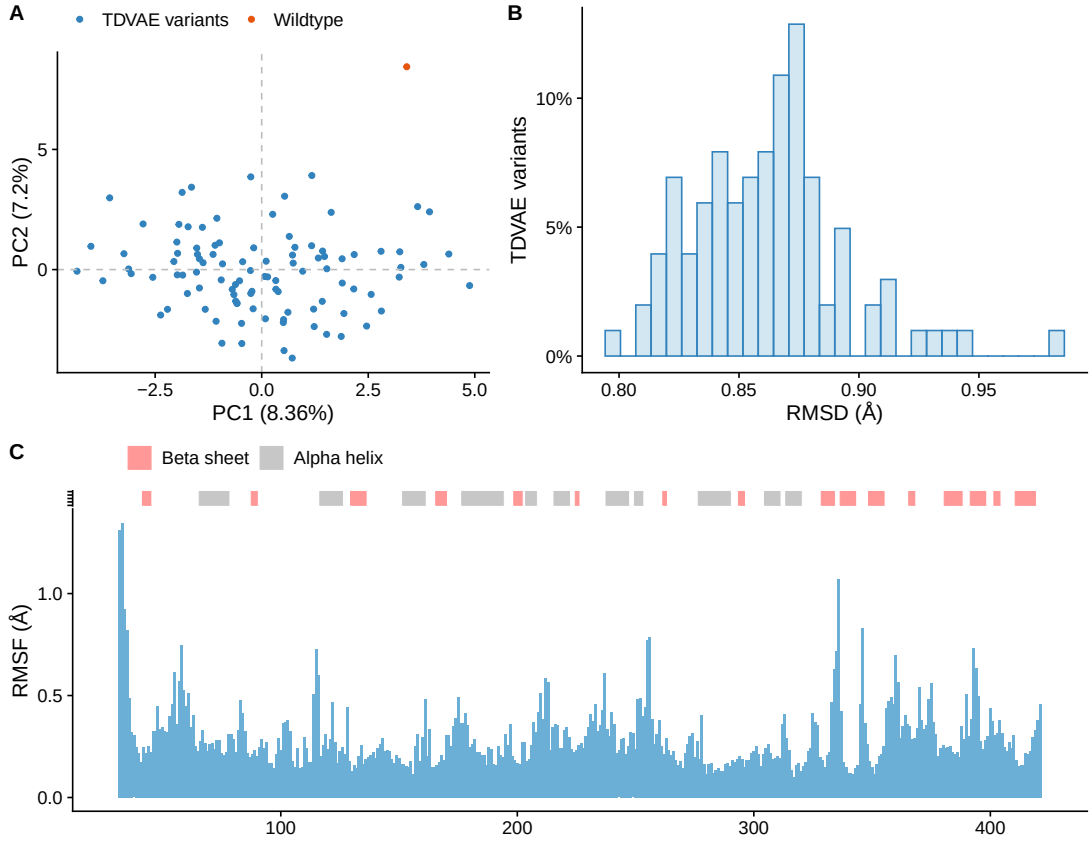

Figure 2.6: **AGAL variants have wildtype like structures.** A) Principal Component Analysis (PCA) analysis of wildtype and variant AGAL structures. B) Distribution of root mean squared deviations (RMSD) of the variants from the wildtype structure (PDB id: 1R46). C) Structural variance analysis of TDVAE variants; the y-axis reports the average root mean square fluctuation (RMSF) per residue.

PCA showed that TDVAE variants are similar to the known wildtype structure (see Figure 2.6A), which is consistent with the average RMSD being $\sim 0.87\mathring{A}$ (see Figure 2.6B) and the high sequence homology with the wildtype of all the variants. Structural variability in the variants was limited to coiled regions, with root mean square fluctuation (RMSF) less than $1\mathring{A}$ everywhere except at the N-terminus(see Fig-

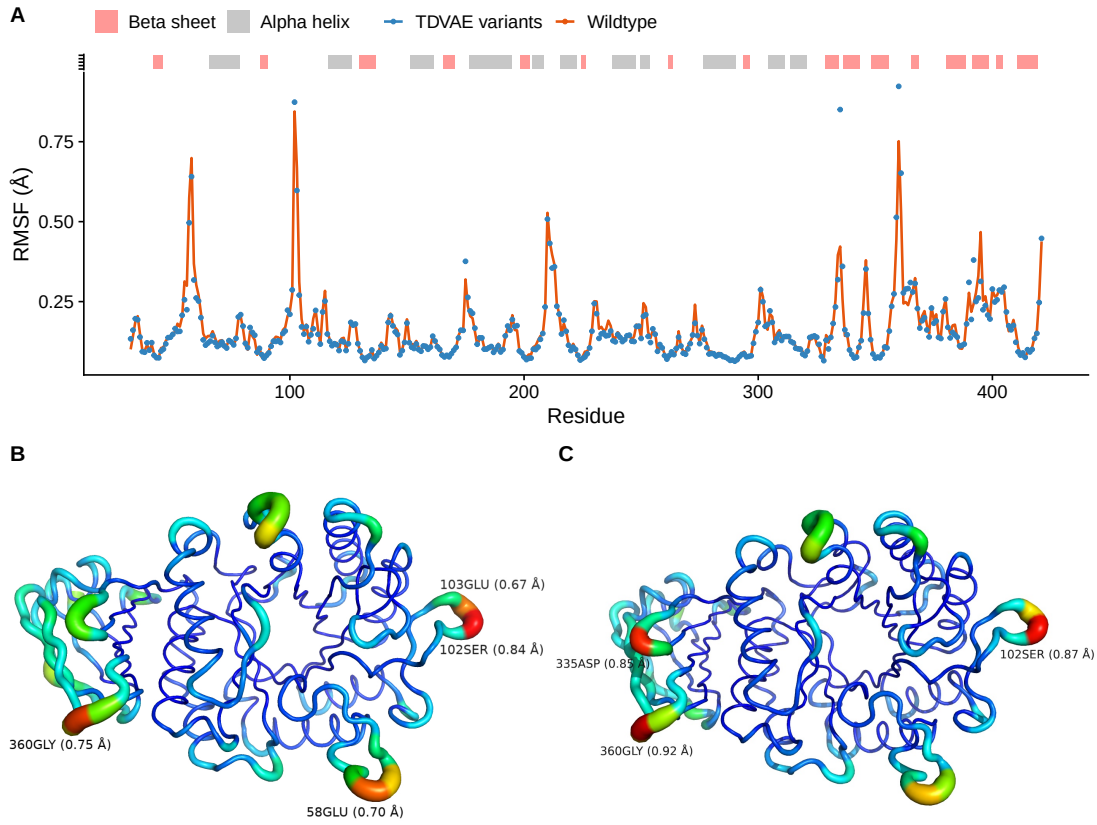

Figure 2.7: **AGAL variants preserve structural flexibility of the wildtype enzyme.** A) ensemble Normal Mode Analysis (eNMA) of the wildtype and TDVAE variants. The RMSF of the variants is averaged per residue across the library. B) RMSF from eNMA analysis for the wildtype structure. C) Average RMSF from eNMA analysis for the variants library projected on the wildtype structure.

ure 2.6C). Finally, we performed ensemble Normal Mode Analysis (eNMA) [Skjærven et al., 2014] to probe large scale motion of our variants. Simulation results showed that the designed variants have a flexibility profile consistent with the wildtype protein (see Figure 2.7A), albeit variants show major instability around the asparagine residue in position 335 (see Figure 2.7C) compared to wildtype (see Figure 2.7B), whereas they proved to be more stable around the glutamic acid at position 58.

Taken together, our analyses confirms that TDVAE can generate a diverse library of putatively active AGAL variants, while retaining the structural and functional prop-

erties of the wildtype enzyme.

Comparison with a directed evolution library

The ability of TDVAE of generating enzymes with wildtype features represents an attractive approach to increase the success rate of variant library screening workflows, where randomized approaches, such as directed evolution, are the industry standard.

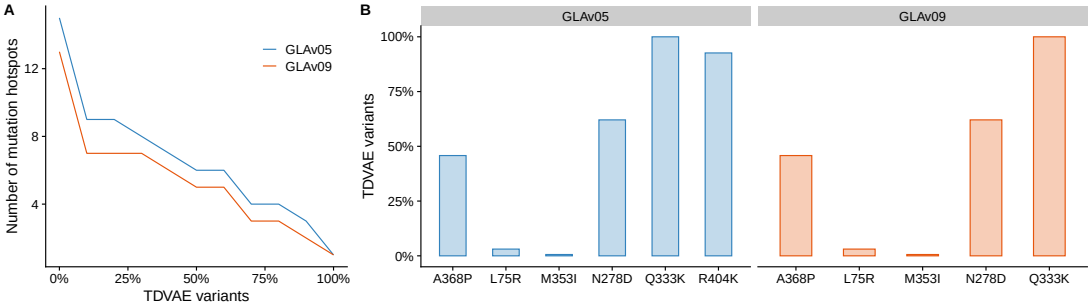

Figure 2.8: **TDVAE identifies beneficial mutational hotspots.** A) Percentage of variants harboring mutations at the mutational hotspots identified in GLAv05 and GLAv09 [Hallows et al., 2023]. B) Percentage of TDVAE variants carrying GLAv05 and GLAv09 beneficial mutations.

To quantify the potential benefit of onboarding our approach, we compared our results with a recent directed evolution study which identified, out of 12,000 screened enzymes, 2 variants, GLAv05 and GLAv09, with higher catalytic activity and stability and lower immunogenicity [Hallows et al., 2023]. These two enzymes identify a total of 19 loci harboring beneficial mutations, which we denote as mutational hotspots. We used this information to assess whether mutations at hotspots were enriched in our library of variants, under the assumption that higher the number of variants harboring mutations at the hotspots and matching the same beneficial mutations, the better the library is. Here we found 2 loci mutated in at least 90% of the sequences and 7 in at least 50% of them (see Figure 2.8A), suggesting that TDVAE identified most of the mutational hotspots in unsupervised fashion from sequence information alone. We then looked at the frequency of the beneficial mutations in GLAv05 and GLAv09, and

found 2 beneficial mutations, `Q333K` and `R404K`, in at least 90% of the variants, and 4 in ∼50% of them (see Figure 2.8B), suggesting that our model has the potential to identify not only the hotspots but also the specific beneficial mutations.

Finally, we used this data to further compare our library with another library generated by TGVAE of the same size using the same training strategy (see Figure 2.9). Interestingly, we found that TGVAE performs worse than TDVAE, with less than 25% of its variants having mutations at the hotspots and less than 1% of the variants having exact GLAv05 and GLAv09 mutations, a significantly worse library compared to the one designed by TDVAE.

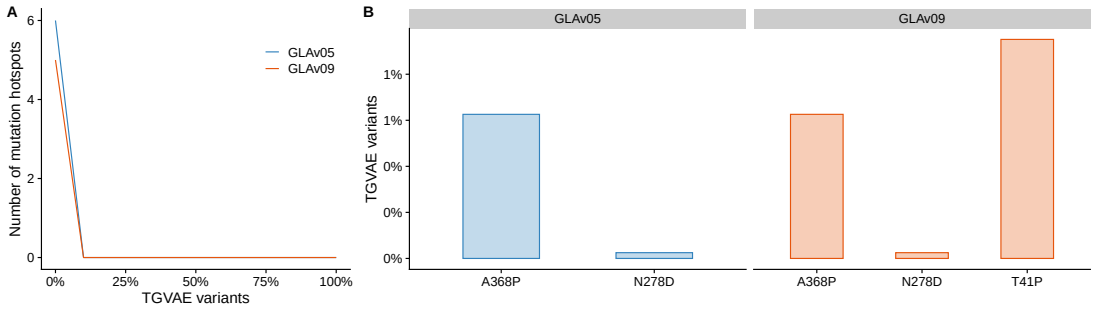

Figure 2.9: **Beneficial hotspots and mutations identified by TGVAE.** A) Percentage of variants harboring mutations at the mutational hotspots identified in GLAv05 and GLAv09 [Hallows et al., 2023]. B) Percentage of TGVAE variants carrying GLAv05 and GLAv09 beneficial mutations.

In conclusion, our results provide strong evidence of the benefits of using TDVAE as a tool to optimize the design of variants library.

2.4 Discussion

Learning how an amino acid sequence can be engineered to obtain a desired biochemical function remains an open challenge in biology, biotechnology and medicine. Recent developments in deep learning are now allowing us to learn the biochemical rules to engineer new functional proteins.

Here we introduced a new deep learning model, called Temporal Dirichlet Variational Autoencoder (TDVAE), to effectively and efficiently predict the effect of amino acid mutations and design new variants of known proteins. Since TDVAE is an unsupervised deep autoregressive model, these tasks are achieved without experimental data but by extracting higher-order information from a functionally related set of sequences. Therefore, TDVAE represent an attractive approach to effectively design large protein libraries to be screened to identify candidates with desired properties. While the use of deep autoregressive models is not new, we introduced Dirichlet latent space modelling as a more accurate approach to estimate amino acid distributions compared to standard Gaussian latent space modelling, and has the potential to identify functional remote homolog proteins.

To prove that, we first used TDVAE as a mutation effect prediction tool, and found that our model achieves excellent performances while being substantially less complex and less taxing to train. We then conducted a hyperparameter analysis to assess the choice of Dirichlet distribution on the performance and showed that Dirichlet distribution is indeed leads to better results when compared to Gaussian distribution. Then, we showed that TDVAE is also able to design variants of a complex protein, like the human α-galactosidase (AGAL) while retaining wildtype like features at sequence, structural and functional level, which are essential to use this enzyme as a potential enzyme replacement therapy for Fabry disease. Remarkably, we showed that our model identified mutational hotspots and beneficial mutations exclusively *in silico*, whereas it required screening more than 12,000 variants using directed evolution [Hallows et al., 2023]. Therefore, our analysis suggests that TDVAE could be a powerful tool to maximize the success rate of protein engineering workflows by designing libraries of functional proteins for massively parallel screening.

Despite our model holds the promise of being an effective protein engineering and analysis tool, we are also aware of its limitation. First, despite generating wildtype like proteins, identifying those with desired characteristics requires downstream post-

processing and experimental validation; extending TDVAE with a conditional generative learning process can address this, albeit at the cost of increased model complexity and computational burden for training and inference. While sequence information has been shown to be sufficient to tackle many protein engineering related tasks, introducing structural information will likely be beneficial, especially in designing enzymes and protein-ligand complexes, where physical constraints are key to obtain the desired function [Hsu et al., 2022b; Anishchenko et al., 2021].

Taken together, we anticipate that the introduction of robust, unsupervised, sequenced-based model like TDVAE will allow us to take advantage of the increasing number of available protein sequences and structures across the kingdom of life, to deepen our understanding of the biochemical rules to engineer functional designer proteins.

Chapter 3

Protein engineering using variational free energy approximation

Disclaimer

This chapter is based on a manuscript accepted at *Nature Communications*. The manuscript is titled *Protein engineering using variational free energy approximation* and is authored by Evgenii Lobzaev, Michael A. Herrera, Martyna Kaspzyk and Giovanni Stracquadanio.

3.1 Introduction

Designing a new protein entails the search for novel, stable and functional amino acid sequences that integrate seamlessly with an existing protein fold, while minimizing the associated free energy of the new configuration [Fleishman and Baker, 2012]. In practice, this requires evaluating thousands of sequence variants [Maynard Smith, 1970], derived from a suitable protein template, for both their thermodynamic and operational fitness, often with respect to a desired downstream application, e.g., biocatalysis or enzyme replacement therapy. This is conventionally achieved experimentally us-

ing Directed evolution (DE) [Arnold, 2018], where thousands of protein variants are synthesized, screened and selected through a selection process that can converge to a sequence optimum that meets specific operational requirements. However, DE is a laborious and expensive process, which requires a high degree of laboratory automation to achieve the throughput required by the industrial biotechnology sector.

Computational methods instead hold the promise to speed up protein engineering by providing a way to screen *in silico* favorable mutations by biophysically modelling thermodynamic changes by amino acid changes [Huang et al., 2016; Kuhlman and Bradley, 2019]. While a plethora of *in silico* biophysical methods have been developed to facilitate the process of protein engineering, they are often constrained by the computational cost of free energy calculations, which in turn limits the exploration of the protein design space. Recently, generative deep learning has emerged as a promising tool for protein engineering, as it can learn the biological properties associated with functional proteins using large sequence and structural datasets [Shin et al., 2021; Anishchenko et al., 2021; Wu et al., 2021; Dauparas et al., 2022; Repecka et al., 2021]. Nonetheless, while the generated sequences broadly resemble those observed in Nature, it remains challenging to obtain functional designs, since most methods cannot condition the sequence generation process towards thermodynamically stable variants.

Here we hypothesized that we could overcome current limitations of generative protein engineering models by learning the sequence-to-free-energy relationship from a small set of biophysical simulations over a library of computationally designed variants using a Variational autoencoder (VAE) [Kingma, 2013]. To this end, we developed the PRotein Engineering by Variational frEe eNergy approximaTion (PREVENT) model, which allows both the controlled generation of variants with minimal free energy and the prediction of the free energy associated with generated variants for downstream experimental prioritization.

We evaluated PREVENT by designing and experimentally testing variants of the archetypal *E. coli* phosphotransferase N-acetyl-L-glutamate kinase (*Ec*NAGK) [Marco-

Marin et al., 2003]. *Ec*NAGK is a small, 258 amino acid long conformationally dynamic homodimer, comprising a bi-domain architecture organized in a Rossmann-like ($\alpha/\beta/\alpha$) sandwich. In Nature, it is primarily responsible for ATP-dependent phosphorylation of N-acetyl-L-glutamate (NAG) in the L-arginine biosynthesis pathway. The intrinsic flexibility of *Ec*NAGK, as evidenced by several crystal structures (PDB: 1GS5, 2WXB, 1OHA, 1OHB, 1OH9), is hypothesized to be crucial for catalysis and hints towards a complex thermodynamical landscape. Using our model, we designed a library of 40 new *Ec*NAGK variants and observed that 85% of the transformed variants could substitute for the wildtype enzyme despite harboring up to 9 mutations compared to the wildtype.

Taken together, our results support a new approach to generative protein design that can dramatically accelerate engineering of novel, functional proteins.

3.2 Methods

In silico mutagenesis and free energy estimation

In order to approximate the *E. coli* phosphotransferase N-acetyl-L-glutamate kinase (*Ec*NAGK) free energy landscape, we generated variants by mutating the wildtype protein uniformly at random, except for the methionine in the first position, and allowing up to 15% of the residues to be mutated, while discarding any duplicated sequences.

We then estimated Gibbs free energy of each variant using the FoldX empirical force field, which has been shown to be a robust tool to estimate the stability of protein and protein complexes [Guerois et al., 2002; Schymkowitz et al., 2005]. Specifically, we used standard FOLDX best practice [Guerois et al., 2002], where we first performed structural relaxation of the wildtype structure, using the REPAIRPDB command, which was then used to estimate the Gibbs free energy (ΔG) of the variants using the BUILDMODEL command using default parameters and averaging the value across 2 runs.

Joint variational inference of protein sequences and free energy

We hypothesized that a generative model could approximate the sequence-to-free-energy function by fitting a latent, multi-variate statistical distribution, whose parameters are unknown but can be estimated from the data. Specifically, we assume that samples of the latent distributions encode both sequence and free energy information, which allow us to explore the protein design space either with respect to the sequence properties or to free energy states.

To do that, we built a new generative protein engineering model, called PRotein Engineering by Variational frEe eNergy approximaTion (PREVENT), by extending the classical Variational autoencoder (VAE) framework [Kingma, 2013], which has been shown to be effective in designing new, functional proteins [Hawkins-Hooker et al., 2021; Shin et al., 2021; Hsu et al., 2022a]. In a classical VAE model, the objective is to maximize the Evidence Lower BOund (ELBO):

$$\mathcal{L}(\phi,\theta) = \mathbb{E}q_\phi(z|x)[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)) \rightarrow \max_{\theta,\phi} \tag{3.1}$$

In equation 3.1 x is an observed random variable (i.e. a protein sequence) and z is latent variable (or encoding) and KL divergence is the Kullback-Leibler divergence. Our model, instead, takes into account two observed variables, that is a sequence x and its associated ΔG value g, which we assume to be conditioned on the latent variable z. Since x and g are conditionally independent given z, the entire probability distribution is factorized as $p(x,g,z) = p(z)p(x|z)p(g|z)$. By assuming that the variational distribution $q_\phi(z|x,g) = q_\phi(z|x)$, since we only model the latent variable dependency on the sequence information, our model can be trained end-to-end by maximizing the modified ELBO:

$$\mathcal{L}(\phi,\theta_1,\theta_2) = \mathbb{E}_{q_\phi(z|x)}[\log p_{\theta_1}(x|z)] + \mathbb{E}_{q_\phi(z|x)}[\log p_{\theta_2}(g|z)] - KL(q_\phi(z|x)||p(z)) \rightarrow \max_{\theta_1,\theta_2,\phi} \tag{3.2}$$

In equation 3.2 $\mathbb{E}_{q_\phi(z|x)}[\log p_{\theta_2}(g|z)]$ term corresponds to the reconstruction error

for ΔG. Specifically, by assuming that $p(g|z) = N(g|\mu_{\theta_2}(z), 1/\sqrt{2})$, this term is proportional to $\mathbb{E}_{q_\phi(z|x)}[-(g-\mu_{\theta_2}(z))^2]$, which is equivalent to the Mean Squared Error (MSE).

To approximate $p_{\theta_1}(x|z)$ and $q_\phi(z|x)$, we used a standard Transformer architecture, a widely used neural network architecture for language processing [Vaswani, 2017]; here we hypothesized that an attention mechanism could better assist in the prediction of ΔG values from a variant than any other architecture, given the high sequence homology in the training set. PREVENT uses 6 layers for the transformer encoder and 4 layers for the decoder with 512 embedding size split across 8 heads. To approximate $p_{\theta_2}(g|z)$, instead, we used a Fully connected Neural Network (FCNN) with 5 layers of progressively decreasing size (from 128 to 1 nodes) and RELU as non-linear activation function, except for the linear transformation in the last layer (see Supplementary Figure B.2).

Finally, since the variance of the free energy can be extremely large between batches and higher free energy values are more likely to appear as most mutations are destabilizing, we developed a weighted random sampler to ensure a uniform coverage of the sequence and free energy space during training. Specifically, a histogram of free energy values was first constructed and then sequences were included in a mini-batch probabilistically at random, with a probability inversely proportional to the number of sequences in their corresponding free energy bin.

Protein engineering using variational energy approximation

Our generative model allows to learn a robust approximation of the sequence-to-free-energy function, which can then be exploited to generate more stable proteins. Specifically, PREVENT implements two free energy based design procedures, namely the seeded non-optimal energy ranking (SNE) and the prior optimization of free energy (POE).

The SNE procedure follows the classical VAE approach, where a seed sequence is

passed in input to the decoder to obtain an embedding z, which in turn is passed in input both to the sequence decoder, to obtain an amino acid sequence, and the energy decoder to estimate the predicted free energy. Specifically, while sequences can be generated using categorical sampling, the expected free energy estimate associated with each generated sequence, instead, is obtained by first passing a generated sequence x in input to the encoder to obtain $q(z|x)$, and then the resulting encoding to the free energy decoder to obtain $\mathbb{E}[g|x]$, that is the expected value of the energy given x, as shown in Eq.3.3. In practice, we estimate the variational posterior free energy expectation $\mathbb{E}[g|x]$ with 300 MC samples as follows:

$$\mathbb{E}[g|x] = \mathbb{E}_{q(z|x)}\Big|\mathbb{E}_{p(g|z)}[g]\Big| = \mathbb{E}_{q(z|x)}[\mu_{\theta_2}(z)] \approx \frac{1}{N}\sum_{n=1}^{N}\mu_{\theta_2}(\tilde{z_n}) \qquad \tilde{z_n} \sim q(z|x) \tag{3.3}$$

While the energy ranking strategy enables the prioritization of variants based on their predicted free energy, it does not inform the sequence generation process. However, since PREVENT also maps free energy information to the latent space, we can perform free energy optimization over the latent space and then use the resulting encoding to sample new sequences; specifically, we hypothesized that this approach would allow us to more efficiently navigate the protein design space and to generate more diverse proteins. To do that, we used a constrained trust region method [Byrd et al., 1999] to minimize the predicted free energy $f_{\theta_2}(z)$, while constraining z within 3 standard deviations from the mean to control sequence diversity. Once an embedding z_{opt} associated with a minimal free energy is found, it is used as input to the sequence decoder to generate new variants, which are then ranked by their expected energy values.

Code Availability

The software is available at the following url:

`https://github.com/stracquadaniolab/prevent-nf`.

Computational structural analysis of generated proteins

To characterize the structural and dynamic properties of the generated variants, we developed a standardized analytical workflow.

First, we obtained structure prediction for each variant using ESMFOLD [Lin et al., 2023] with default parameters, and then performed potential energy minimization to relax the predicted structure as implemented in the OPENMM package [Eastman et al., 2017] using the Amber14 force field, while adding an harmonic potential energy term to restrain Cα atoms position. ensemble Normal Mode Analysis (eNMA) was performed using the BIO3D package [Grant et al., 2006], using the `calpha` force field and by setting the temperature to T=300K. Downstream fluctuation analysis was conducted by generating trajectory using the 7th mode. Finally, we evaluated the likelihood of the variants to be compatible with the known wildtype structure. To do that, we used the ESM inverse folding model (ESM-IF) [Hsu et al., 2022b] using the wildtype *Ec*NAGK structure, and scoring variants using the score $IF(s_{mut}, s_{wt}) = \mathcal{L}(s_{mut}) - \mathcal{L}(s_{wt})$, where $\mathcal{L}$ is the log likelihood of a sequence given the input structure. IF values greater or equal to 0 are indicative of highly structurally compatible variants.

Experimental materials

BW25113 Δ*argB* Keio knockout strain was purchased from Horizon Discovery Ltd. Basic parts for plasmid creation were kindly supplied by Dr. Marcos Valenzuela-Ortega. Polymerases, RNA miniprep kits, restriction enzymes and master mixes for cloning were purchased from New England Biolabs. Invitrogen E-Gel (1%, with SYBR Safe) agarose gels were used for DNA electrophoresis. Invitrogen NuPAGE (4-12% Bis-Tris) gels and 1X MES NuPAGE SDS Running buffer were used for SDS-PAGE. 5X M9 minimal salts base was purchased from Formedium. All other chemical reagents were purchased from Merck or Fisher Scientific. The *Ec*NAGK variant library was assembled and prepared by Neochromosome Inc. The High-throughput

transformations and agar plate spotting were achieved using an Opentrons OT-2 robot equipped with a thermocycler module, a single channel p20 pipette and a multi-channel p300 pipette. Where stated, the working concentration of kanamycin or carbenicillin in selective media was 50μ g mL^{-1} and 100μ g $m\,L^{-1}$, respectively.

Construction of the pKCHU-argB vector

pKCHU-argB was assembled via the Joint Universal Modular Plasmids (JUMP) method using pre-domesticated parts and destination vector (see Supplementary Table B.2) [Valenzuela-Ortega and French, 2021]. In brief, the basic parts (2 $fmol\ \mu L^{-1}$ each) and destination vector (2 $fmol\ \mu L^{-1}$) were combined with BsaI (1U) and NEBuilder Master Mix (1X) in a total reaction volume of 15 μL. The parts were assembled using the reported thermocycler parameters for a level 0 JUMP assembly [Valenzuela-Ortega and French, 2020]. The reaction mixture was used to transform commercial NEB5-alpha cells (New England Biolabs) using the heat-shock method. Transformants were selected overnight at 37°C on Yeast Extract-Peptone (YEP) agar plates supplemented with kanamycin. Potential constructs were identified by colony PCR, propagated in YEP-kanamycin medium (10 mL), isolated using a GeneJET Plasmid Miniprep kit (Thermo) and confirmed by Sanger sequencing.

Purification of *Ec*NAGK variant DNA library

The *Ec*NAGK variant DNA library was assembled and prepared commercially using the same basic parts as pKCHU-argB. Dried *E. coli* Top 10 cells, harbouring sequence-perfect clones, were individually reconstituted in yeast-extract-peptone (YEP) medium (200 μL) at 37°C for 6 hours. The reconstituted cells were propagated in fresh YEP medium (10 mL) for a further 18 hours at 37°C. Following biomass harvest by centrifugation (4000 rcf, 15 minutes), the plasmids encoding the *Ec*NAGK variants were isolated using a GeneJET Plasmid Miniprep kit (Thermo). The plasmids were stored

at -20°C in Tris-HCl buffer (0.1 M, pH 8) until required.

Curing, preparation and transformation of chemically-competent *E. coli* BW25113 Δ*argB*

Commercial *E. coli* BW25113 Δ*argB* was cured by Flp-FRT recombination using the curing plasmid pCP20, as per literature protocol [Datsenko and Wanner, 2000]. The loss of both *kanR* and *ampR* from BW25113 ΔargB was confirmed by the restored susceptibility to kanamycin and carbenicillin. Cured BW25113 Δ*arg*B was sub-cultured in YEP media (20 mL, 37°C) until mid-log phase. The biomass was harvested by centrifugation (3000 rcf, 10 minutes, 4C) and washed thrice with sterile, ice-cold CaCl2 (100 mM, 20 mL). The washed cells were resuspended in ice-cold $CaCl_2$ (100 mM, 2 mL), sub-aliquoted as required and transformed fresh using the heat-shock method. pKCHU-argB transformants were selected on YEP-agar (37°C, 18 hours) laced with kanamycin.

Auxotrophic screen on solid minimal media

BW25113 ΔargB transformants (harbouring pKCHU-argB or variants thereof) were propagated in YEP-kanamycin medium (10 mL) overnight with agitation (37°C, 18 hours). Culture samples (4.8×10^8 cells) were diluted in PBS (500 μL, 0.1 mM, pH 7.4), gently harvested and washed a further two times in PBS (200 μL) to remove residual rich media. The washed cells were resuspended in PBS (100 μL) and spotted (5 μL) on solid minimal media containing bacto-agar (1% w/v), M9 minimal salts base (1X), $MgSO_4$ (2 mM), $CaCl_2$ (0.1 mM) and D-glucose (0.4% w/v). Positive control media was prepared by supplementing with L-arginine (100 $\mu g m L^{-1}$). The minimal media plates were incubated (37°C) and visually inspected for growth over 48 hours.

Auxotrophic selection on liquid minimal media

BW25113 Δ*argB* transformants (harbouring pKCHU-argB or variants thereof) were propagated in YEP-kanamycin medium (10 mL) overnight with agitation (37°C, 18 hours). The washed cells were resuspended in PBS (10 mL) and sampled for back-dilution in minimal media (25 mL, OD600 = 0.1) containing M9 minimal salts base (1X), $MgSO_4$ (2 mM), CaCl2 (0.1 mM) and D-glucose (0.4% w/v). The cultures were incubated (37 °C) with rigorous shaking for 12-48 hours. OD_{600} measurements were recorded periodically.

Reverse Transcriptase Quantitative PCR (RT-qPCR) of wildtype constructs

BW25113 Δ*argB* transformants (harbouring pKCHU-argB) were propagated in YEP-kanamycin medium (10 mL) overnight with agitation (37°C, 18 hours). Untransformed BW25113 Δ*argB* was similarly propagated in antibiotic-free YEP. The cells were sub-cultured (OD_{600} = 0.1) in fresh YEP media and harvested by centrifugation (3000 rcf, 12 minutes) at mid-log phase. The cells were resuspended in PBS (1 mL) and pelleted again by microcentrifugation (3000 rcf, 5 min). Total RNA was extracted from the pelleted biomass using a Monarch Total RNA Miniprep kit (New England Biolabs). Per RT-qPCR reaction (20μL), cDNA was synthesised and amplified from total RNA (500 ng) with target-specific primers (400 nM) using a Luna Universal One-Step RT-qPCR kit (New England Biolabs), according to manufacturer protocol. The reactions were performed and analysed in a StepOnePlus RT-qPCR system (Thermo) configured for SYBR Green fluorescence detection. Reverse transcriptase-free and template RNA-free controls were performed in parallel. Relative fold-expression was calculated via the Livak method using *E. coli* 16S rRNA (rrsA) as the reference gene.

3.3 Results

L-arginine is a ubiquitous proteinogenic amino acid necessary for the basic physiological function of all organisms. De novo L-arginine biosynthesis proceeds via a critical L-ornithine precursor in both animals and bacteria. Distinctly from animals, however, bacterial L-ornithine is generated from a cascade of N-acetylated, rather than non-acetylated, intermediate (see Supplementary Figure B.1) [Cunin et al., 1986].

The first committed step in *E. coli* L-ornithine biosynthesis is the transacetylation of L-glutamate using acetyl-CoA to yield NAG, catalyzed by N-acetyl-L-glutamate synthase (NAGS). In the second biosynthetic step, the nascent NAG is phosphorylated to furnish N-acetyl-L-glutamyl 5-phosphate (NAGP); this ATP-dependent conversion is catalyzed by aforementioned *Ec*NAGK, encoded in *E. coli* by the *argB* gene. Importantly, the binding of ATP to *Ec*NAGK triggers a major inter-domain conformational shift that greatly tightens the active site, thereby permitting the facile phosphoryl transfer from ATP to NAG. This reaction is further facilitated by invariant residues Lys8, Gly11, Gly44, Gly45 and Lys217, which work to correctly orient the substrates and stabilize the transition state. Following phosphoryl transfer, the enzyme relaxes into an open conformation and liberates the products NAGP and ADP in preparation for the next catalytic cycle.

Here we used our PREVENT model to replace the wildtype *Ec*NAGK enzyme with new, unseen variants thereof. To do that, we began by performing *in silico* mutagenesis of the *Ec*NAGK wildtype sequence (Uniprot ID: P0A6C8; PDB ID: 1GS5) to build the input dataset required to approximate the sequence-to-energy relationship for this enzyme (see Methods). With our procedure, we generated 117,387 unique variants, with each variant harboring up to 38 mutations ($\approx$ 15% of the residues), and then computed the associated free energy, ΔG, using FOLDX [Guerois et al., 2002]. Each position of the wildtype enzyme, except the first one, is mutated on average in 6.58% of the variants; as expected, most of the introduced mutations are destabilizing, with a significant positive correlation between number of mutations and free energy (see

Figure 3.1A,B). In order to study how PREVENT performances depend on the size of the input mutagenesis dataset, we further subsampled the initial dataset by selecting 25K, 50K and 75K sequences at random, albeit maintaining a ΔG distribution similar to the original dataset.

Using these datasets, we then trained our model with 128-dimensional latent space for 1,400 epochs using mini batches of 256 sequences, and by setting the learning rate to 10^{-4} and the dropout probability to 0.2 for regularization. Moreover, to learn effective latent space encodings, we masked 25% of each sequence residues at random. With this experimental setup, we tested our model on 4 different configurations.

We first assessed model performances using an held-out test set by looking both at the sequence reconstruction error and the Root Mean Square Error (RMSE) of the free energy across the 4 experiments. After attaining the prefixed number of epochs, the reconstruction error was 114.80 (perplexity: 0.66) for the full dataset, suggesting that our model is able to effectively reconstruct sequences from the latent space, a trend that was consistent regardless of the size of the training dataset (see Supplementary Table B.1). We then looked at the accuracy of the predicted free energy, and obtained the best results when using the full dataset (RMSE = 9.27); nonetheless, even when using only 25K sequences, the model achieves comparable performances (RMSE = 12.12). Importantly, while the predicted energy values might differ from those estimated by FOLDX, they are in strongly correlated, with Spearman correlation ranging from $\rho = 0.96$ when training on the full dataset and only dropping to $\rho = 0.92$ when using only 25K sequences, suggesting that predicted values are a robust metric for variants ranking. This is further confirmed by the concordance at the top (CAT) analysis of variants ranked by free energy values; in particular, we found a concordance higher than 80% when looking at the top 1,000 ranked sequences, a trend observed regardless of the size of the mutagenesis dataset used (see Figure 3.1C,D).

We then analyzed the *Ec*NAGK encodings to check whether PREVENT would map different variants to different region of the latent space. To achieve this, we computed

the empirical variance of each latent component, defined as $\mathrm{diag}(\mathrm{cov}_x[\mathbb{E}_{q_\phi(z|x)}[z]])$ [Burda et al., 2015], and considering a component as being active if its standard deviation was greater than 0.1. Here we found that the vast majority (126) of the 128 components to be active, albeit to a different level, suggesting that our model maps variants to different regions of the latent space. We then proceeded with the analysis of the latent space organization by performing Principal Component Analysis (PCA) of the expected value of the variational posterior distribution, $q_\phi(z|x)$, for the sequences in the mutagenesis dataset, which revealed a structural organization, where sequences with lower free energy were located in the central part of the latent space, whereas sequences with higher ΔG values were located in the periphery, remotely resembling a folding funnel (see Figure 3.2).

Taken together, we found that PREVENT is able to effectively reconstruct *Ec*NAGK sequences and the expected free energy from the latent space.

Sequence analysis of the *Ec*NAGK variants generated by PREVENT

We then proceeded to perform sequence analysis of variants generated by the two PREVENT strategies, namely prior optimization of free energy (POE) and seeded non optimal free energy ranking (SNE). We generated 2,000 variants using POE and 30,000 samples for SNE, and then removed those sequences that were i) present in the training set, ii) have mutations in the binding site, or iii) are duplicated.

With these criteria, we obtained 1,282 new variants (66%) for POE and 13,323 variants for SNE (44%), which is expected given that the first strategy freely explores the free energy landscape of *Ec*NAGK, whereas the second generates sequences within the region of the wildtype sequence. Importantly, POE variants are more diverse (see Figure 3.3A) than SNE, with an average of 7.09 and 3.57 mutations respectively.

Further, we analyzed the type of mutations introduced by our sampling strategies by scoring the designed variants using the BLOSUM62 substitution matrix. We chose BLOSUM62 as it is a widely used substitution matrix for general-purpose alignment

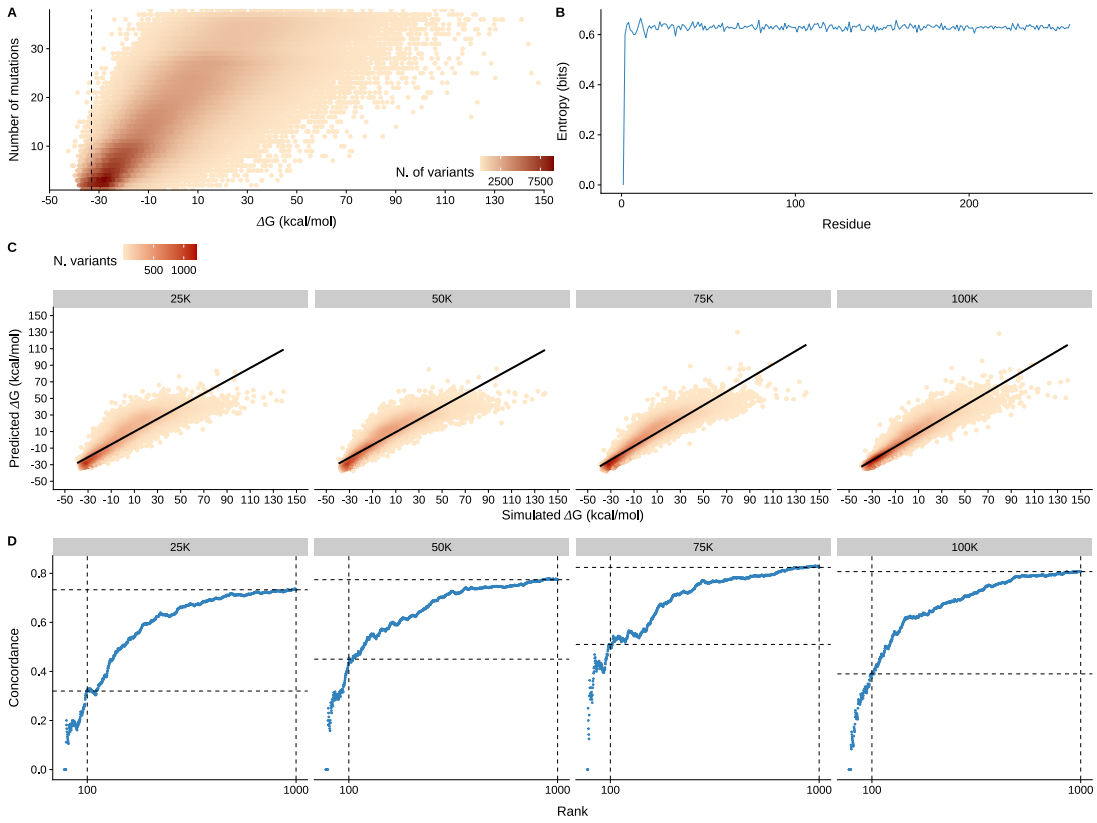

Figure 3.1: **PREVENT performance analysis.** A) Thermodynamic landscape approximation (ΔG) as a function of the number of mutations in *Ec*NAGK variants in the training dataset. Black dashed line denotes the free energy of the wildtype *Ec*NAGK. B) Amino acid entropy of *Ec*NAGK variants in the training dataset. Every position, except the first methionine, is mutated in 6.58% of the generated variant. C) PREVENT estimates of ΔG values against free energy estimates simulated using FoldX for different training set sizes. RMSE is 9.28 for 100K training set, 9.74 for 75K training set, 11.24 for 50K training set and 12.12 for 25K training set. Spearman correlation coefficient is 95.53% for 100K training set, 95.25% for 75K training set, 93.58% for 50K training set and 91.79% for 25K training set. D) concordance at the top (CAT) analysis of sequences ranked using free energy information. PREVENT achieves a concordance 75% with FoldX simulations regardless of the size of the training set, albeit moderate differences in ranking top sequences are observed when considering the top 100 and 1000 variants.

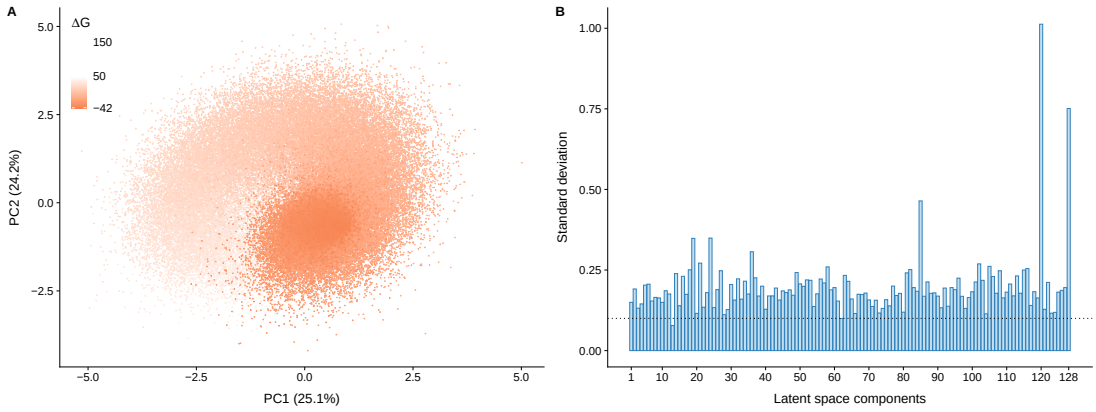

Figure 3.2: **PREVENT latent space analysis.** A) Principal component analysis of the $\mathbb{E}_{q_\phi(z|x)}[z]$ embeddings generated by PREVENT for the sequences in the training set and color coded according to the corresponding free energy ΔG values. B) Activation plot for the 128-dimensional latent space learned. The x-axis represents each latent component and y-axis the standard deviation of the corresponding value learned during training; the dashed line represents the expected value of each component under a null model of random value assignment.

of protein sequences, and we did not want to over-penalize introduced mutations by using a more stringent matrix for highly homologous sequences, such as BLOSUM80. We observed that only $\approx 25\%$ of mutations were conservative, as measured by BLOSUM62 substitution matrix; this result suggests that PREVENT might select energetically favorable mutations regardless of evolutionary information (see Figure 3.3B). Interestingly, the distribution of scores, using BLOSUM80 matrix does not change significantly (see Supplementary Figure B.3), with only a few mutations being over-penalized. Finally, while the type of mutations was highly similar across strategies, they are located differently across the wildtype *Ec*NAGK sequence (see Figure 3.3C); in particular, POE predominantly mutates the C-terminus harboring the ATP-binding domain of the protein, which harbors highly non-conservative mutations, such as the tryptophan introduced at position 212 in place of the aspartic acid residue (see Figure 3.3D), whereas SNE introduced mutations almost uniformly across the wildtype

sequence (see Figure 3.3E).

Taken together, we found that PREVENT can generate highly diverse protein variants, which carry not only conservative mutations but also non-conservative ones, which would not be selected by using homology information alone.

Design of a library of *Ec*NAGK variants

We then built a library of *Ec*NAGK variants for experimental validation, in order to estimate the expected fraction of functional variants generated by PREVENT.

To do that, we took the 10 variants with lowest free energy obtained with the POE strategy and the 10 variants with the lowest free energy generated with the SNE strategy. We further augmented this library by adding 10 variants with the highest ELBO, denoted as SNL, and 10 with the highest identity with respect to the wildtype, hereby denoted as SNI, in order to compare our energy-driven design against common generative strategies for protein engineering (see Figure 3.4A). Variants in the library have an average free energy ranging from -29.8 kcal/mol for the SNE group, to -18 kcal/mol for the SNL group, with a number of mutations ranging from 6.8 for the SNE group to 1 for the SNL group, mostly located in the C-terminus of the protein for the variants generated by energy-driven strategies (see Figure 3.4B). Taken together, we screened a library of 40 new and diverse variants with a significant variability in free energy.

Before proceeding with experimental work, we performed a series of quality control steps, first, by comparing the predicted free energy values of each variant to the FOLDX estimates where, as expected, we found a strong correlation between the two estimates (Spearman correlation; $\rho = 0.77$, p-value: 5.7×10^{-9}). We then assessed the structural properties of the variants by predicting their three-dimensional structure using ESMFOLD, performing inverse folding sequence scoring using the wildtype *Ec*NAGK structure and then downstream dynamics analyses using ensemble Normal Mode Analysis (eNMA). The predicted structures have an average pLDDT of 89.64 (see Figure 3.5A), which confirms high accuracy of the predictions, closely re-

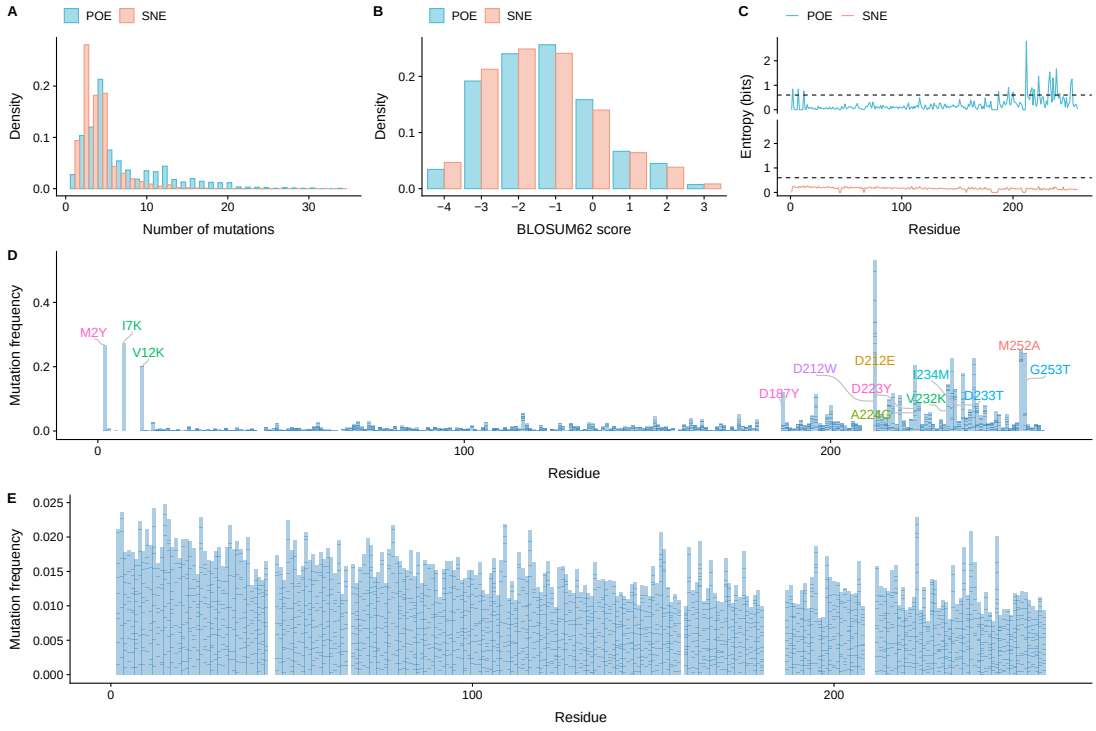

Figure 3.3: **Sequence analysis of PREVENT generated variants.** A) Distribution of the number of mutations in variants generated using the POE and SNE strategies; the POE strategy generates more diversity variants compared to SNE. B) Distribution of the BLOSUM62 substitution matrix scores for the 9,083 mutations introduced by the POE strategy and for the 47,573 mutations introduced by the SNE strategy; the vast majority of the mutations are non-conservative. C) Residue level entropy for variants obtained using POE and SNE strategies; POE generates preferentially adds mutations at the C-terminus of the protein. Most frequent mutations in variants generated by the POE strategy (D) the SNE strategy (E). While SNE variants have a similar mutational profile to the training set, POE variants have a vastly different mutational profile, with surprisingly non-conservative mutations highly represented in all variants.

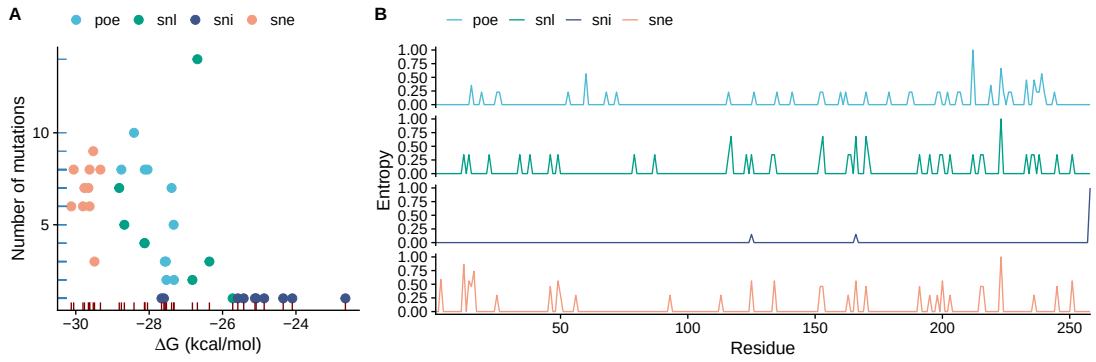

Figure 3.4: **Design of a library of 40 *Ec*NAGK variants using PREVENT.** A) Free energy and number of mutations of each variant colored by design strategy. B) Normalized sequence entropy of variants generated using different design strategies; energy-based strategies (POE, SNE) introduce mutation preferably in the C-terminus of the wildtype protein.

sembling the structure of the wildtype enzyme (average RMSD: 1.70Å). This result was further confirmed when scoring our variants against the wildtype *Ec*NAGK structure, where the average inverse folding score of -0.077 (see Figure 3.5B-C), suggesting that variants are highly consistent with the wildtype fold albeit to a lesser extent than the wildtype sequence (wildtype log-likelihood: -0.991; average variants log-likelihood: -1.069). We then looked at the dynamics of each variant and, in general, we found strong agreement with the wildtype dynamics (average RMSF: 0.15, (see Figure 3.6A)). Nonetheless, we found that the region spanning residues 58 and 65, and 211 and 216, showed increased flexibility, which could potentially lead to phenotypic differences (see Figure 3.6B)).

Taken together, the overall dynamics of the protein is preserved despite the introduced mutations.

Establishing the auxotrophic screen for *Ec*NAGK

The *Ec*NAGK encoded by the *argB* gene is conditionally essential for protein synthesis in the absence of exogenous L-arginine (or a suitable precursor). In this context,

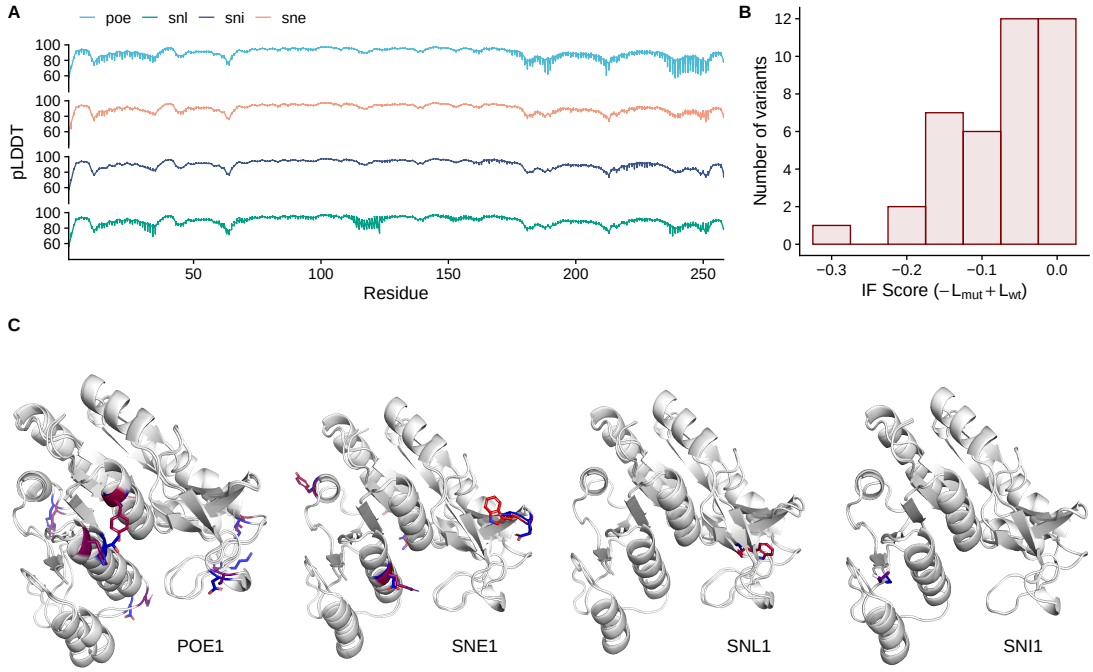

Figure 3.5: **Structural analysis of the *Ec*NAGK variants library designed by PREVENT.** A) Accuracy of the *Ec*NAGK variants structures predicted by ESMFOLD reported as predicted local distance difference test (pLDDT) at the residue level; regions with pLDDT $>$ 70 are expected to be accurately modelled. B) B) Inverse folding scores of the variants library. C) Structure of the top variant for each design strategy, with mutations color coded based on the BLOSUM90 substitution matrix scores.

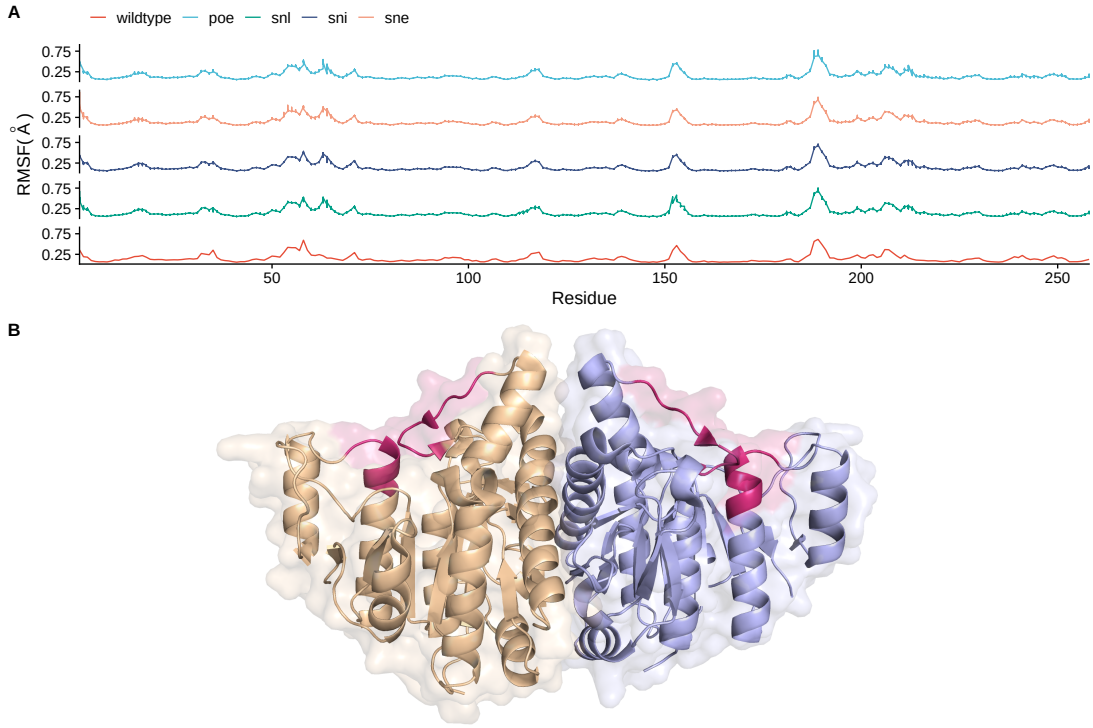

Figure 3.6: **ensemble Normal Mode Analysis (eNMA) of the *Ec*NAGK variants library designed by PREVENT.** A) Residue level Root Mean Square Fluctuation (RMSF) of *Ec*NAGK variants compared to wildtype. The most significant changes with respect to the wildtype are in the regions comprising residues 58 and 65, and 211 and 216. B) Dimer structure of *Ec*NAGK with region of increased flexibility compared to the wildtype colored in red.

we hypothesised that *argB* knockouts could be rescued by expressing *Ec*NAGK (or functional variants) via an auxiliary plasmid, which is supplemented by DNA transformation. Without access to such a plasmid, we reasoned that the auxotroph will not survive in L-arginine-deficient media. Therefore, by using a simple viability screen, it becomes possible to identify functional *Ec*NAGK variants in a manner amenable to high-throughput screening.

To test this hypothesis, we cured (see Supplementary Figure B.4) and transformed commercial BW25113 Δ*argB* using a bespoke constitutive expression vector encoding the wildtype *E. coli argB* gene (pKCHU-argB) and the production of *Ec*NAGK was subsequently confirmed by SDS-PAGE (see Supplementary Figure B.5A-C). As anticipated, L-arginine auxotrophy could be remedied in M9 minimal salts media by either the supplementation of L-arginine or by complementation with pKCHU-argB (see Supplementary Figure B.5D). However, using this plasmid-based expression system, we observed that *argB* knockouts required over 24 hours of incubation to emerge from lag phase. Based on our SDS-PAGE analysis, we hypothesized that the strong *Ec*NAGK overexpression was severely penalizing cell growth in minimal media. By recoding the *argB* coding sequence using *E. coli* codon usage bias data (see Supplementary Figure B.6) [Sharp et al., 1988], we measured an average 25-fold reduction in gene expression by RT-qPCR, which was associated with a drastic improvement in growth rate using both solid and liquid minimal media (see Supplementary Figure B.7). Given the clear benefit to cell viability, we decided to backtranslate all our variants using the same codon usage bias strategy.

Screening the *Ec*NAGK variant library

A sequence-perfect library of 40 *Ec*NAGK variants cloned into our pKCHU expression vector was used to transform BW25113 Δ*argB*. To this end, we developed a robust and high-throughput heat-shock transformation protocol using an Opentrons OT-2 robot, which allowed us to reliably and reproducibly transform *E. coli* using both

pET23b-EFGP and pKCHU-argB with minimal user input (see Supplementary Figure B.8). Whilst many *Ec*NAGK variants yielded colonies overnight following heat-shock transformation, several more variants exhibited poorer transformation efficiency and required up to 2 consecutive nights of incubation to develop colonies (see Supplementary Figure B.9). Despite numerous attempts, 7 *Ec*NAGK variants (namely poe4, snl3, snl6, snl7, sni10, sne1, sne3) did not produce viable transformants. The observed array of cell viability suggests that the *Ec*NAGK variants are not tolerated equally, and some variants may considerably affect cell fitness.

The remaining 33 viable variants were studied by auxotrophic selection, as previously outlined. Remarkably, 22 of the 33 viable *Ec*NAGK variants could quickly recover the BW25113 Δ*argB* phenotype in standard M9 minimal salts solid media without L-arginine supplementation. Interestingly, these variants include the top-ranked viable candidates in each model category (poe1, snl1, sni1, sne2). A further 6 variants (poe5, sni9, sne5, sne7, sne8, sne9) facilitated slow-to-intermediate recovery, thus bringing the total count of active variants to 28. The remainder (poe2, poe3, poe8, snl5, snl10) did not rescue the phenotype after 48 hours. Conversely, all cells carrying our variants could proliferate readily and rapidly by supplementing the minimal media with L-arginine (see Figure 3.7A).

We then checked whether using the predicted energy information for designing or prioritizing variants was associated with either a higher number of functional variants or more diverse sequences. 14 out of the 28 viable variants were obtained by using either POE (6 variants) or SNE (8 variants); as expected, since they carry only 1 mutation, 9 out of 10 SNI variants recovered the phenotype, whereas only 5 out of the 10 SNL variants were functional.

Interestingly, the functional variants designed using free energy information (POE, SNE) harbor, on average, significantly more mutations (5.64) compared to the 1.64 mutations found on variants selected by sequence only information (Student t-test; t: 4.9173, df: 22.177, p-value = 3.151×10^{-5}), suggesting that exploiting free energy in-

formation could yield more diverse library with minimal impact on the overall number of functional proteins obtained in a screen.

We subsequently quantified the growth supported by the top candidates for each design strategy (namely poe1, snl1, sni1, and sne2) in a cell density time-course assay. We observed that variants poe1, sni1, and sne2 enabled similar rates of exponential growth compared to the wildtype *Ec*NAGK (see Figure 3.7B-C), albeit cells expressing sne2 exhibited a 23-34% shorter lag time compared to all other experiments (see Supplementary Table B.5). Conversely, cells expressing variant snl1 doubled 39% slower on average than those expressing the wildtype enzyme. Notably, variant snl1 features a radical and sterically-intrusive Gly123Trp mutation in an otherwise highly-conserved position on β7, which likely destabilizes the β6-β7 hairpin in the NAG-binding subdomain (see Supplementary Figure B.10) [Ben Chorin et al., 2020]. Furthermore, the expression of variant snl1 was discovered to be 5-fold greater than the wildtype by RT-qPCR analysis (see Supplementary Figure B.11). Therefore, we propose that the observed growth penalty may be attributed to the divergent nature of the Gly123Trp mutation and/or less favorable transcription regulation, which is mediated by our gene re-coding strategy.

In conclusion, 85% of the designed and transformed *Ec*NAGK variants enabled survival on auxotrophic media, with 67% permitting similar recovery levels to the wildtype sequence.

3.4 Discussion

Computational protein engineering has been limited by the complexity and poor understanding of the function driving the folding of a polypeptide chain into a thermodynamically stable structure.

Here we addressed these problems by building a generative deep learning model, called PREVENT, which uses variational inference to approximate the free energy

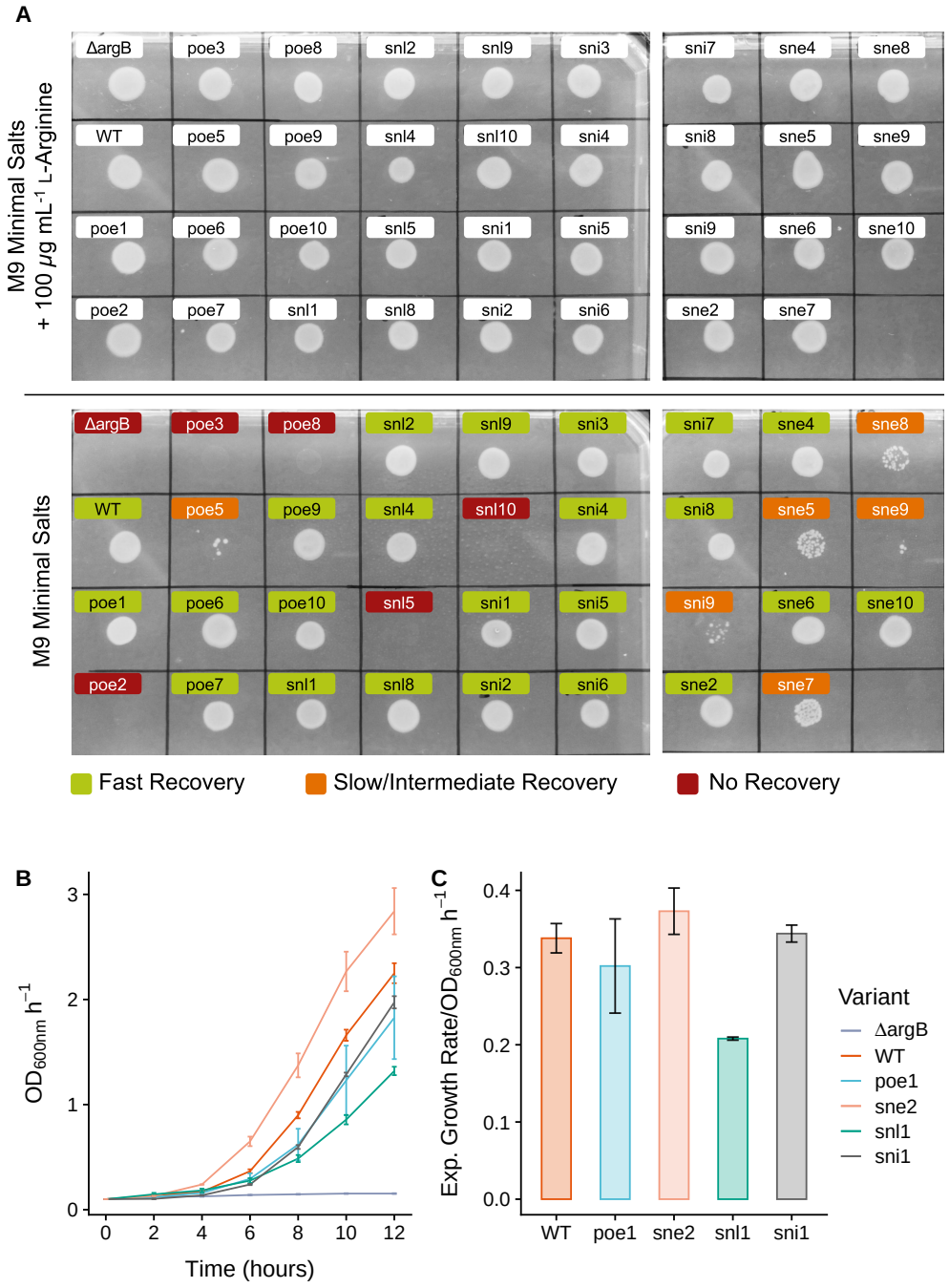

Figure 3.7: **Functional analysis of the *Ec*NAGK variants designed by PREVENT.** A) Top plate: perfect recovery of transformed *Ec*NAGK variants, including wildtype and Δ*argB*, in L-arginine rich media. Bottom plate: different recovery rates of transformed *Ec*NAGK variants in the minimal media. B) Growth curves of BW25113 (*argB*) and *Ec*NAGK transformants grown in auxotrophic M9 minimal salts media. Optical density (OD) measurements were recorded every 2 hours for up to 12 hours. Error bars represent the standard error of three biological replicates. C) The average rate of exponential growth supported by wildtype *Ec*NAGK and variants. Error bars represent the standard error of three biological replicates.

landscape of a target protein by learning favorable mutations from small number of biophysical simulations. Our method enables generation of new protein sequences and simultaneous prediction of the associated free energy and, more importantly, enables free energy minimization over a mathematically tractable thermodynamic space, which enables the rapid identification of diverse putatively functional variants.

We used our model to redesign the *E. coli* phosphotransferase N-acetyl-L-glutamate kinase (*Ec*NAGK), a key enzyme of the L-arginine pathway characterized by a highly dynamic structure, which is likely to be associated with multiple stable thermodynamic states. Computational results showed that PREVENT can accurately approximate the free energy landscape of this protein and generate new and diverse variants, which retain wildtype structural features, irrespective of the size of the training set used. We then built a library of *Ec*NAGK variants and experimentally shown that 85% of the generated variants were functional, albeit not all variants could completely recover the wildtype growth phenotype. Importantly, design or selection of variants using predicted free energy information allowed us to obtain a significantly more diverse set of functional proteins compared to sequence only selection criteria, further validating our approach.

While PREVENT allowed redesigning a complex enzyme, we are also aware of its limitations. Currently, our free energy estimates are obtained by FOLDX, a well-known and validated tool, which has been shown to be reliable mostly for small and medium size proteins, mostly due to the intrinsic difficulty of modelling proteins in unfolded state [Schymkowitz et al., 2005]. Understanding how the accuracy of the free energy calculations will impact our model will require further analysis using different biophysical engines, including neural network potentials [Maksymenko et al., 2023], and proteins from different families. Moreover, while the use of free energy information for protein engineering is reasonable, the limited accuracy of current estimation methods suggests that relative changes, $\Delta\Delta G$, in free energy compared to the wildtype protein could be a more robust and accurate way to exploit energy information,

especially for large proteins.

Taken together, our work provides a new framework to integrate generative deep learning with biophysical information, which can substantially improve both the number and diversity of functional proteins obtained compared to more lengthy and expensive molecular biology approaches. It is also important to note that while PREVENT learns the sequence-to-free-energy relationship of a protein, it can learn relationships with any other biochemical property, thus representing a generic framework for phenotype-driven protein design.

Chapter 4

Epitope recoding using multimodal generative deep learning

4.1 Introduction

Human adaptive immune response is aimed at recognizing and neutralizing foreign pathogens. Adaptive immune response is mediated by lymphocytes, such as T-cells and B-cells, that are responsible for the humoral and cell-mediated immunity. Both T-cells and B-cells recognize specific regions of the pathogen, called epitopes and trigger the immune response aimed at neutralizing the pathogen. The recognition of the epitopes and the subsequent response differs between T-cells and B-cells. B-cells recognize B-cell epitopes, which are typically linear or conformational, with the help of B-cell receptor (BCR) and trigger the production of antibodies that neutralize the pathogen. T-cells recognize T-cell epitopes, which are typically linear, by T-cell receptors (TCRs) with the help of Major histocompatibility complex (MHC) molecules of Antigen-presenting cells (APCs) and kills the infected cells with the help of cytotoxic T-cells (CD8+) and helper T-cells (CD4+) [Zubler, 2001; Janeway et al., 2001; Sanchez-Trincado et al., 2017]. Knowing these epitopes is crucial for the development of vaccines and therapeutics [Lin et al., 2008; Jensen et al., 2018].

Immunogenicity is a major concern in the development of protein therapeutics. Repeated administration of protein therapeutics can lead to the development of anti-drug antibodies (ADAs), which can reduce the efficacy of the drug and cause adverse effects. For example in Fabry disease, classical male patients, lacking any endogenous AGAL protein, are under a high risk for developing anti-drug antibodies against AGAL ERTs, Agalsidase-alfa and Agalsidase-beta [Schiffmann et al., 2001; Banikazemi et al., 2007]. A solution to this problem is to develop a novel ERT that is not recognized by the immune system. One solution, as shown in Pegunigalsidase alfa study, is a pegylation, which acts as a physical barrier for ADAs, and cross-linked homodimerization of the AGAL enzyme, which suggests less immunogenicity compared to the currently available ERTs due to prolonged stability and increased half-life of the enzyme in plasma [Lenders et al., 2022]. Alternative deimmunization approach is to eliminate known epitopes from the protein sequence by humanizing them substantially [King et al., 2014]. This approach has been shown to be effective for deimmunization of P99 β-lactamase used in cancer therapy and antibacterial drug lysostaphin [Salvat et al., 2017; Zhao et al., 2015].

Here, we propose a novel AI-driven approach to the AGAL epitope redesign problem [Scharnetzki et al., 2020]. Specifically, we propose to use a multimodal deep learning model to generate novel sequences with modified epitopes. We hypothesized that a combination of two different modalities, such as sequence and structure, can be used to generate novel sequences with modified epitopes that are similar to the original sequence in terms of structure and properties. We first pretrained our generative model on a large and diverse dataset of protein structures and sequences, such as CATH 4.2, followed by finetuning on a smaller dataset of AGAL homologous sequences and structures. We then masked out most immunogenic epitopes in the AGAL sequence and used the finetuned model to complete the sequence by redesigning the epitopes. Finally, we compared the performance of the model with and without the structural information in the encoder and assessed the quality of the generated sequences in terms

of immunogenicity and compatibility with the wild-type structure.

4.2 Methods

Data collection

For the model pretraining, we used the CATH 4.2 dataset, which partitions all available protein domain structures up to 500 amino acids with 40% non-redundancy into training (18,204), validation (608) and test (1,120) sets by their CATH classification (class, architecture, topology/fold, homologous superfamily) [Ingraham et al., 2019]. For finetuning, we used AlphaFold Protein Structure Database (AFDB) cluster for AGAL enzyme (Uniprot ID: P06280, AFDB cluster ID: R1EKQ7) obtained by clustering the whole AFDB by MMSeqs2, with each cluster maintaining a maximum sequence identity of 50% and achieving a 90% bi-directional sequence overlap with the longest sequence of the cluster representative [Jumper et al., 2021; Varadi et al., 2022; Steinegger and Söding, 2018]. Furthermore, we filtered the AFDB dataset by removing sequences that are longer than 500 amino acids and outside eukaryotic taxonomy. We also masked out backbone coordinates if pLDDT score was less than 70. The resulting dataset consisted of 2,465 sequences and structures.

Multimodal generative modelling

We hypothesized that a combination of two different modalities, such as sequence and structure, used in the VAE encoder could lead to a better performance both in terms of training and generating novel sequences with modified epitopes compared to the model that uses sequence information only. In this case two observed variables, sequence x and structure s, act as conditional variables for the variational posterior distribution $q_\phi(z|x,s)$. As before, the model is trained end-to-end by maximizing the Evidence

Lower BOund (ELBO) defined as follows:

$$\mathcal{L}(\phi,\theta) = \mathbb{E}_{q_\phi(z|x,s)}[\log p_\theta(x|z)] - \beta KL(q_\phi(z|x,s)||p(z)) \rightarrow \max_{\theta,\phi} \tag{4.1}$$

We used a cyclical linear annealing schedule for the KL term, starting from 0.01 and increasing it to 1.0 with 4 cycles over the training time [Fu et al., 2019]. For validation, we always used the full KL term.

Since we are conditioning the latent space on the protein backbone structure, which is a 3D object, we need to ensure invariance to the global rotation and translation of the protein backbone. In other words, we need to make sure that $q_\phi(z|x,s) = q_\phi(z|x,T(s))$ where $T(s)$ is a roto-translation of the protein backbone. To achieve that, we used special GVP–GNN (Geometric Vector Perceptron - Graph Neural Network) encoder layers capable of processing structural information in an equivariant manner. For GVP–GNN protein backbone structures are represented as proximity graphs, where each node represents an amino acid of a protein sequence. Node features are a combination of scalar- and vector-valued structural information, such as torsion angles, distances and orientations of the amino acids in the protein sequence, while the edge features encode relationship between spatially neighboring amino acids by using CA–CA vectors and its distances [Jing et al., 2020]. By construction, the input features for GVP–GNN are translation invariant. Each GVP–GNN layer is rotation invariant for scalar features and equivariant for vector features, that is for any vector feature v and any rotation R, $f(Rv) = Rf(v)$, where f is the GVP-GNN layer. To achieve the desired invariance of the vector features, they are projected to a local reference frame of the protein backbone based on N, CA, C atom positions of the amino acids [Jumper et al., 2021; Hsu et al., 2022b]. These projected, rotation invariant, vector features are then combined with the scalar features and dense amino acid embeddings and passed through the TCN layers to obtain the parameters of variational posterior distribution $q_\phi(z|x,s)$. The decoder part of the model is responsible for reconstructing the input sequence from the latent representation using TCN layers as in Chapter 2.

For fair comparison, we also trained two more VAE models: one that uses only

sequence information and another one that uses only structural information in the encoder, effectively making an inverse-folding VAE model in the latter case.

Training and finetuning experiments

For each model type, we defined a baseline set of parameters in small and large model configuration, making sure that any model can be trained on a single GPU with a reasonable batch size (see Supplementary Table C.1). We then varied a small number of hyperparameters, that we deemed to be the most important for the model performance, such as masking type, masking probability and aggregation of input modalities in the VAE encoder (see Supplementary Table C.2). For a standard, sequence-only, VAE, we tried both span and random masking as well as combination of forward and reverse sequences in the input. In case of the structurally augmented VAE, in addition to various masking regimes, we tried different ways of combining sequence and structural information in the encoder, namely, summation and concatenation. For the inverse-folding VAE model, we varied the depth of the GVP-GNN layers and did not use any additional masking, beyond present in the dataset, for the structural information. The maximum number of training epochs was set to 300 with early stopping based on the validation loss. We used the Adam optimizer with a learning rate of 10^{-4} and a batch size of 64. Perplexity per amino acid, defined as the exponentiated average negative log-likelihood of a protein sequence, was also computed and reported. We then selected few best models based on the validation loss and used them for the finetuning on the AGAL dataset. The finetuning is performed for 100 epochs with a similar masking regime as in the pretraining phase. No validation set was used, and the model snapshots were taken every 20 epochs.

Sampling and epitope design

A seed sequence and its backbone structure are passed through the encoder part of the model to obtain the parameters of the variational posterior distribution $q_\phi(z|x,s)$. A random sample z is obtained from this distribution and passed through the VAE decoder to obtain a new sequence, using either greedy selection or categorical sampling in each position for the sequence generation.

An epitope design is performed by providing a model with a seed sequence and structure, as well as the list of epitopes to be modified. Within each of provided epitopes, a controlled number of amino acids are randomly masked out. This partially masked sequence and wild-type structure are then passed through the model to complete the partially masked input. These designed sequences are then used as an input for the BEPIRED 3.0 tool to identify potential discontinuous and linear B-cell epitopes [Clifford et al., 2022]. We used default settings recommended by the BEPIRED 3.0 web server and compared the average result for generated sequences with the wild-type sequence.

In addition to the B-cell epitope probability prediction, we computed the humanization score for each generated sequence. First, we used the whole human proteome [Frankish et al., 2020] and calculated all its possible 9-mers. We further excluded all 9-mers that are present in the wild-type AGAL sequence. The resulting set was then used as a proxy of a Fabry disease patient's proteomic landscape. 9-mers are of paramount importance in immunology, particularly due to their prevalence as the most common length of epitopes recognized by T-cells. The assumption is that the more human-like the sequence is, the less likely it is to be recognized by the immune system. Humanization score for each generated sequence is computed by counting the number of its 9-mers (k_j) that are present in the Fabry disease patient's proteome (S) and normalizing by the number of 9-mers in the sequence (K_i) (see Equation 4.2). This very conservative score tells us how much human-like the generated sequence is, with a score of 1 indicating that the sequence is fully human-like.

$$Score_i = \frac{\sum_{k_j \in K_i} \mathbb{1}(k_j \in S)}{|K_i|} \tag{4.2}$$

4.3 Results

Model comparison by pretraining on CATH 4.2 dataset

For each model type and model configuration, we trained a total of 144 experiments on the CATH 4.2 dataset. We first identified that structure-to-sequence VAE model with any configuration performs much worse than other models, regardless of the model size, and therefore we excluded it from the further analysis, even though we observed a small improvement in the validation perplexity when scaling up the model (see Supplementary Table C.6). For standard, sequence-to-sequence, VAE models we observed that the large model configuration outperforms the small model configuration in terms of the validation perplexity (see Supplementary Table C.3). Using sequence in both forward and reverse direction as an input to the encoder always improves the validation perplexity, without increasing the number of parameters or the training set size. Finally, no conclusion can be made about the best masking type. Only in high masking regime, the model with span masking performs better than the model with random masking.

Next, we trained a structurally augmented version of the sequence-to-sequence VAE model. As in case of the standard VAE model, the large model configuration (see Supplementary Table C.5) outperforms the small model configuration (see Supplementary Table C.4) in terms of the validation perplexity and span masking gives better results than random masking in high masking regime. Furthermore, the concatenation of input modalities in the encoder is always preferred to the element-wise summation.

In order to compare the performance of standard and structurally augmented VAE

Model type	Masking probability	Sequence only			Sequence & structure			Difference
		Model ID	**# parameters**	**Avg PPL (val)**	**Model ID**	**# parameters**	**Avg PPL (val)**	
small	0.1	experiment-2	6 382 487	3,7592	experiment-24	7 392 444	3,7454	-0,0138
	0.05	experiment-14	6 382 487	3,8265	experiment-8	7 392 444	3,8684	0,0420
	0.1	**experiment-16**	6 382 487	4,0676	**experiment-32**	7 392 444	4,0143	-0,0533
	0.2	experiment-18	6 382 487	4,7451	experiment-36	7 392 444	4,6782	-0,0669
	0.4	experiment-10	6 382 487	5,6149	experiment-19	7 393 244	5,5265	-0,0885
large	0.1	experiment-2	21 896 887	1,6280	experiment-24	25 894 012	1,6205	-0,0075
	0.05	experiment-14	21 896 887	1,7066	experiment-7	25 895 612	1,73874	0,0321
	0.1	experiment-16	21 896 887	1,9071	experiment-11	25 895 612	1,91722	0,0102
	0.2	experiment-8	21 896 887	2,3922	experiment-15	25 895 612	2,26542	-0,1268
	0.4	**experiment-10**	21 896 887	3,0968	**experiment-19**	25 895 612	2,78736	-0,3095

Table 4.1: **Sequence-to-sequence and structurally augmented sequence-to-sequence model performance comparison.** For each masking probability, we selected best model configuration from sequence-to-sequence and structurally augmented sequence-to-sequence VAEs and trained them 5 times. The table shows the average perplexity on the validation set for each model configuration. Highlighted models were selected for sampling analysis.

models, we selected the best model configuration from each masking regime and model size and trained them 5 times with different initializations, resulting in additional 100 experiments. Interestingly, we did not observe significant difference in terms of the validation perplexity between standard and structurally augmented VAE models, with the only exception being large model with the highest masking regime. For further assessment of the model performance and, more importantly, the added value of the structural information in the encoder, we selected 4 models, 2 small and 2 large from the same masking regime, namely, 0.1 and 0.4, hereafter referred to as small and large sequence-to-sequence and structurally augmented sequence-to-sequence models (see Table 4.1).

We then selected 7 proteins for sampling procedure, 6 proteins of different length from the CATH 4.2 test set as well as AGAL enzyme from the AFDB cluster. First, we observed that greedy selection of the most likely amino acid is not giving enough diversity in the generated sequences, which is consistent with how CATH 4.2 datasets

were constructed, and therefore we used categorical sampling for the rest of experiments. We then sampled $20,000$ sequences for each protein, removed possible duplicates and queried the remaining ones against the wild-type sequence using BLASTP. We removed sequences above 0.001 e-value threshold and below 25% query coverage and computed various pairwise statistics, describing the quality of generated sequences (see Supplementary Table C.7).

Ideally, we would like to see generated sequences to be similar to the wild-type, that is to have similar length, identity and high coverage. While all assessed models have similar identities of generated sequences, regardless of the protein size, larger models tend to generate sequences with higher coverage and length, therefore making them more similar to the wild-type sequence. As for the use of the structural information, no significant difference can be observed between sequence-to-sequence and structurally augmented sequence-to-sequence model samples, as measured by the bitscore statistic. For 3 proteins, namely *1b6q*, *1sji* and *agal*, we visually assessed the quality of generated sequences. We generated MSA for each protein and calculated entropy and coverage for each position in the alignment corresponding to the wild-type sequence (see Figure 4.1).

We observed that a small protein *1b6q* has a good coverage and a relatively low entropy across the sequence length, which is indicative of the high quality of generated sequences. However, generated sequences are typically longer than the wild-type sequence. Generally, not much difference can be observed between our models, with smaller models showing a bit more diversity in certain positions.

For larger proteins, such as *1sji* and AGAL, generated sequences are of lower quality than for the small protein, such as *1b6q*. Although generated sequences typically have similar lengths to the wild-type sequence, except samples of the small structurally augmented model, the entropy and coverage analysis suggests that only parts of generated sequences are similar to the wild-type sequence.

Taken together our pretraining results suggest few things. First, the perplexity

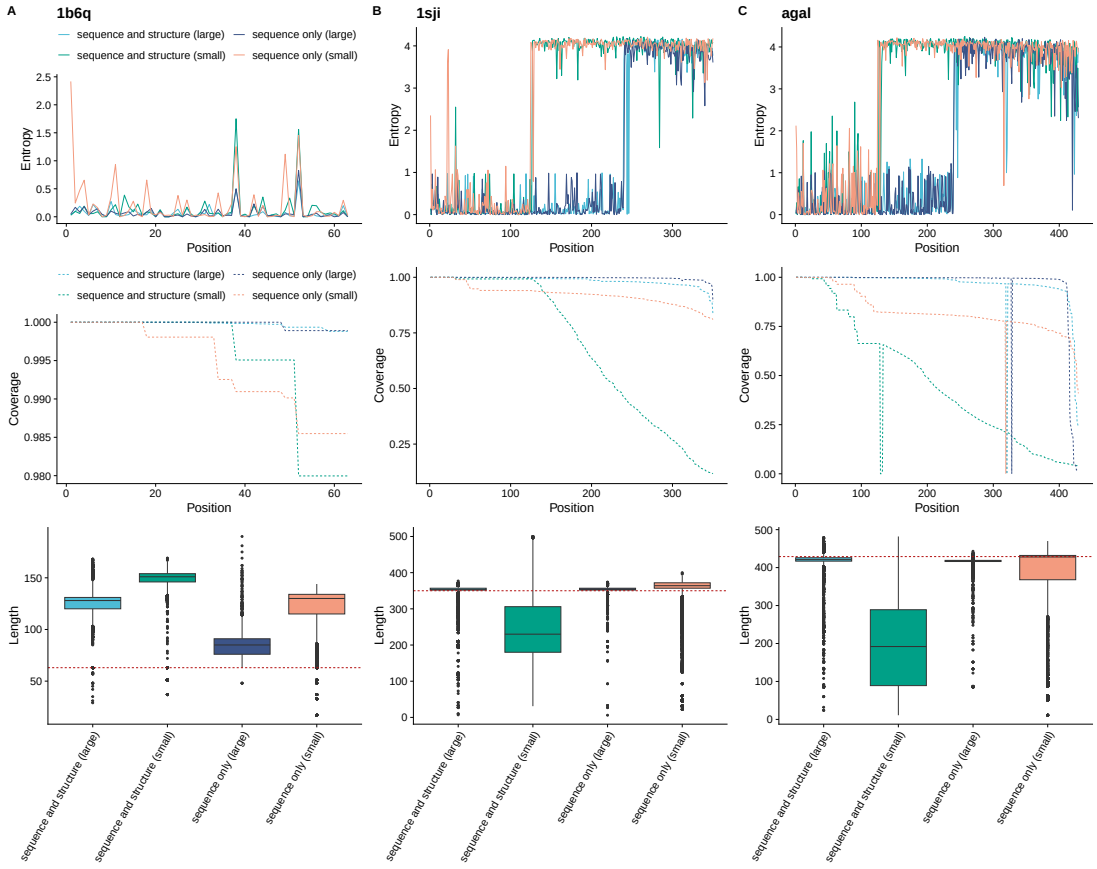

Figure 4.1: **Sampling results for selected proteins.** For 3 selected proteins, *1b6q* (left column), *1sji* (central column) and *agal* (right column), entropy (top row) and coverage (middle row) for each position of the wild-type sequence was calculated for each model. In addition, distribution of samples' length is shown (bottom row) with dashed line representing the wildtype length.

metric is not sufficient to assess the model performance as we can observe that models with similar perplexity values (~ 4 vs ~ 3) can generate sequences of different quality. Second, the benefit of the structural information in the encoder is not clear as we did not observe significant difference in the quality of generated sequences. Finally, while training on a large and diverse dataset of protein structures and sequences is beneficial for assessing differences in model performance and can be used as a filtering metric for the model selection, it is not enough for high-quality sequence generation, especially for larger proteins, that can be used in downstream applications, such as library preparation.

Epitope recoding by finetuning on AGAL dataset

After pretraining analysis, we took large VAE models and finetuned them on the AGAL dataset for 100 epochs, using the same masking regime as in the pretraining phase and taking model snapshots every 20 epochs. Interestingly, we observed that the structurally augmented sequence-to-sequence model learns significantly faster than the sequence-to-sequence model (see Supplementary Figure C.1). Next, we compared how our models perform in the epitope design problem. We selected 7 non-overlapping AGAL epitope regions, spanning in total 134 amino acids or 31% of AGAL, (see Supplementary Table C.8) that were shown to be the most recognized by the immune system [Scharnetzki et al., 2020] and randomly and independently masked out up to 10 amino acids within each epitope, obtaining a total of $10,000$ partially masked AGAL sequences. We took two snapshots (at epoch 20 and 100) of each model and used them to complete partially masked sequences 5 times each by passing them through VAEs and selecting the most likely amino acid at each sequence position.

Variants generated by the structurally augmented sequence-to-sequence model have a slightly lower average number of mutations per sequence (28), compared to the sequence-to-sequence model (33) (see Figure 4.2A), which is consistent with sampling statistics (see Supplementary Table C.9). Interestingly, the structurally aug-

mented model always generates sequences of the same length as the wild-type sequence, whereas the sequence only model occasionally generates sequences that are slightly shorter than the wild-type sequence. The sequence only model tend to generate more variability across the whole sequence length and the area around the binding site region (residues $203-207$) is more conserved in the structurally augmented model. As observed previously, there is much more variability in the N-terminus (1-32 residues). We analyzed mutations introduced in at least 75% of generated sequences using the BLOSUM62 substitution matrix. We found that sequence only model introduces more potentially deleterious mutations (`C63P`, `F145Q`, `T158W`, `F248S`, `Q330R`, `G346D`, `Q416R`,), which account for almost 65% of the most frequent mutations, whereas the structurally augmented model has a slightly smaller number of the most frequent mutations (55%) being potentially deleterious (`M42G`, `Q57F`, `Q330R`, `G346D`,`Q416R`) (see Figure 4.2C).

After 100 finetuning epochs, results between two models are much more comparable, with both models generating sequences of the same length as the wild-type sequence. The sequence only model still has a slightly higher average number of mutations per sequence and more variability across the sequence length (see Supplementary Figure C.2A,B). Only two mutations are introduced in at least 75% of the generated sequences by both models, namely `G346D` and `Q416R`, both having BLOSUM62 score of -1 (see Supplementary Figure C.2C).

We then assessed if any of mutations introduced in generated sequences are associated with Fabry disease or changes in enzymatic activity. Both models introduced a very frequent `Q330R` mutation, described previously, and much less frequent `A31V` and `A20D` mutations in the signalling peptide region, classified as 'probably' and 'possibly damaging' by the POLYPHEN2 tool (see Figure 4.3A). In terms of comparison with the directed evolution study [Hallows et al., 2023], no significant difference between our models was observed, although the structurally augmented model introduced beneficial mutations at mutational hotspots, identified as GLAv05 and GLAv09, more often

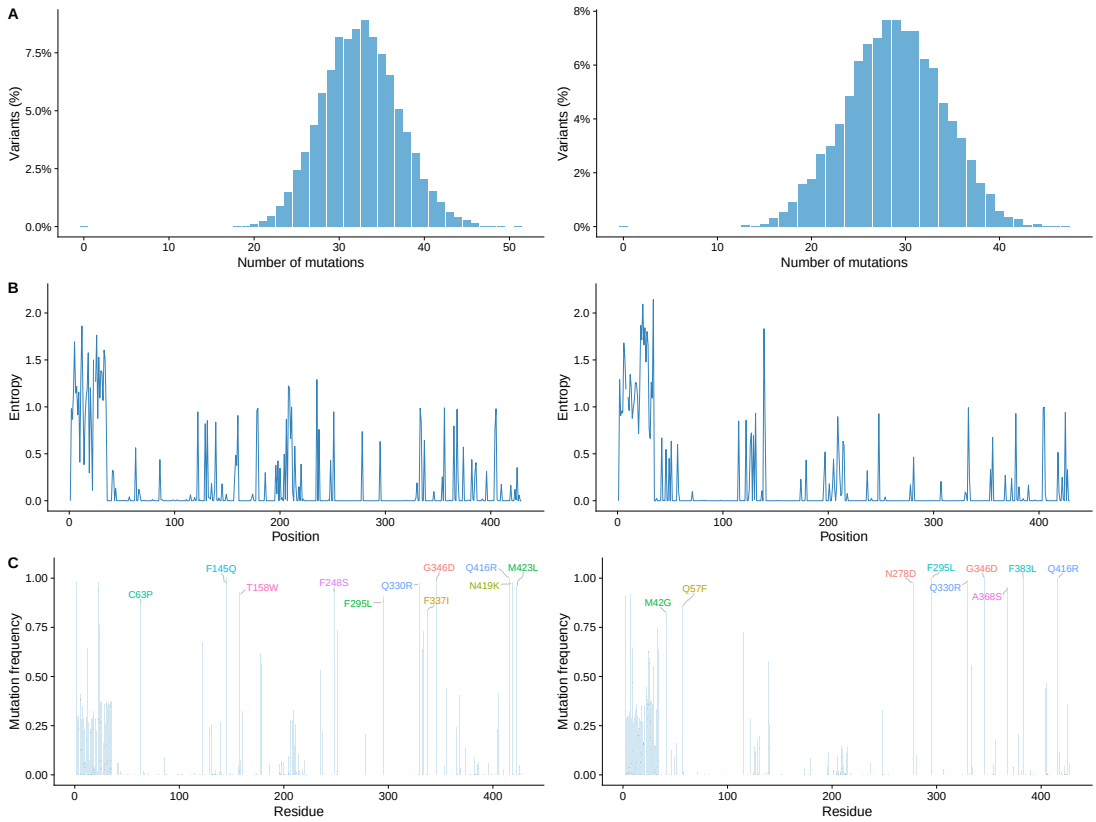

Figure 4.2: **Mutation analysis of the sequence only and structurally augmented VAEs after** 20 **finetuning epochs.** Comparison of sequence only (left column) and structurally augmented (right column) VAEs in terms of library diversity. A) Histogram of the number of mutations per generated sequence. B) Residue-level entropy of the generated sequences. C) Most frequent mutations.

(see Figure 4.3B,C). Structurally augmented model tends to generate 2.5 times more human-like sequences (446 out of $10,063$), as measured by the humanization score, compared to the sequence only model (174 out of $10,009$). Moreover, the structurally augmented model tends to 'humanize' sequences more than the sequence only model as measured by the range of the humanization scores (see Figure 4.4C).

Unsurprisingly, after 100 finetuning epochs, we observe less difference between samples from both models. Only two mutations, `A31V` and `L3V`, associated with Fabry disease, are present in generated sequences with less than 10% frequency (see Supplementary Figure C.3A). When it comes to the directed evolution study comparison, both models tend to have lower frequency of beneficial mutations at mutational hotspots, identified as GLAv05 and GLAv09, than after 20 finetuning epochs. However, the structurally augmented model still has a slight advantage over the sequence only model (see Supplementary Figure C.3B,C).

Next, we checked how likely generated sequences are to possess B-cell epitopes. Even though sampling results between models are quite similar, the structurally augmented model generates less immunogenic sequences, in some areas significantly reducing the B-cell epitope probability (see Figure 4.4A,B). As before, after 100 finetuning epochs, the difference between models is less pronounced (see Supplementary Figure C.4A,B). In terms of humanization score, both models fail to 'humanize' generated sequences, with the sequence only model generating only 1 sequence with non-zero score and the structurally augmented model generating 4 sequences.

Finally, we assessed how compatible generated sequences are with the wild-type AGAL structure (PDB ID: 1R46). For this task, we considered only full length sequences with non-zero humanization scores. We eliminated the signalling peptide region (1-32) and calculated the IF score for each sequence against the wild-type structure. We found that the structurally augmented model generates sequences that are significantly more compatible with the wild-type structure (see Figure 4.5). IF scores were not computed for the sequences generated after 100 finetuning epochs, as the

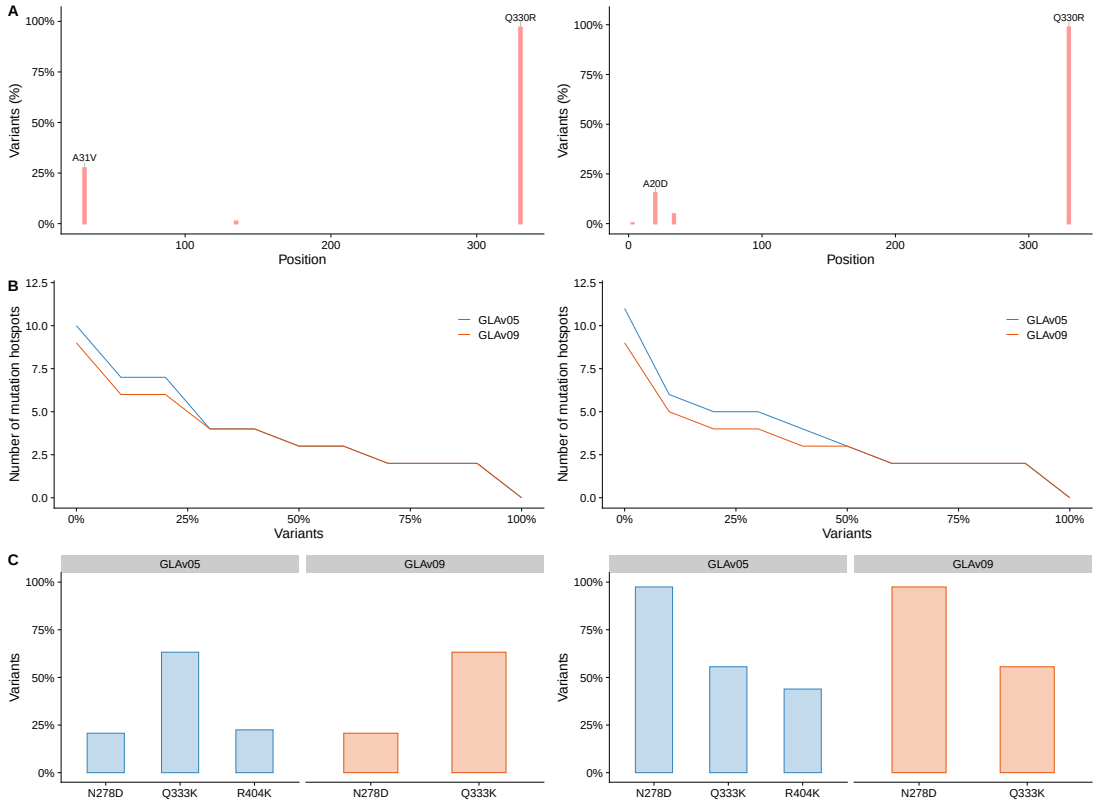

Figure 4.3: **Pathogenic and beneficial mutations comparison after** 20 **finetuning epochs.** Comparison of sequence only (left column) and structurally augmented (right column) VAEs in terms of pathogenic and beneficial mutations. A) Frequency of mutations in the designed variants associated with Fabry disease or changes in enzymatic activity. B) Percentage of variants harbouring mutations at the mutational hotspots identified as GLAv05 and GLAv09. C) Percentage of variants carrying GLAv05 and GLAv09 beneficial mutations.

humanization score was zero for the majority of sequences.

Despite no significant differences observed in the pretraining phase, our finetuning results showed clear distinction between the sequence only and structurally augmented models in terms of the quality of generated sequences. Although the output from both models shares certain attributes, such as similar identity to the wild-type or coverage, the structurally augmented model excels in generating more sequences with fewer potential immunogenic epitopes and higher human-like content. This result suggests that structurally augmented VAE output is more suitable for the downstream tasks of library design. For practical applications, it is important to consider the finetuning period length. We have shown that the difference between models is more pronounced after 20 finetuning epochs compared to 100 epochs, which suggests that both models may overfit to the AGAL dataset after a longer finetuning period.

4.4 Discussion

Designing a novel ERT that is not recognized by the immune system is a challenging task. Here we addressed this problem by using a multimodal generative DL model to redesign the AGAL enzyme epitopes. We first pretrained the model on a large and diverse dataset of protein structures and sequences, such as CATH 4.2, followed by finetuning on a smaller dataset of AGAL homologous sequences and structures from the AFDB. Surprisingly, despite no clear difference between the sequence only and structurally augmented VAE results in the pretraining phase, our finetuning results showed significant difference in the quality of redesigned AGAL sequences. The structurally augmented model generated significantly more sequences that are more human-like, more compatible with the wild-type fold and contain less potential immunogenic epitopes.

This study also has some practical implications. First, we showed that the use of a single metric, such as perplexity, is not sufficient to fully assess the model performance,

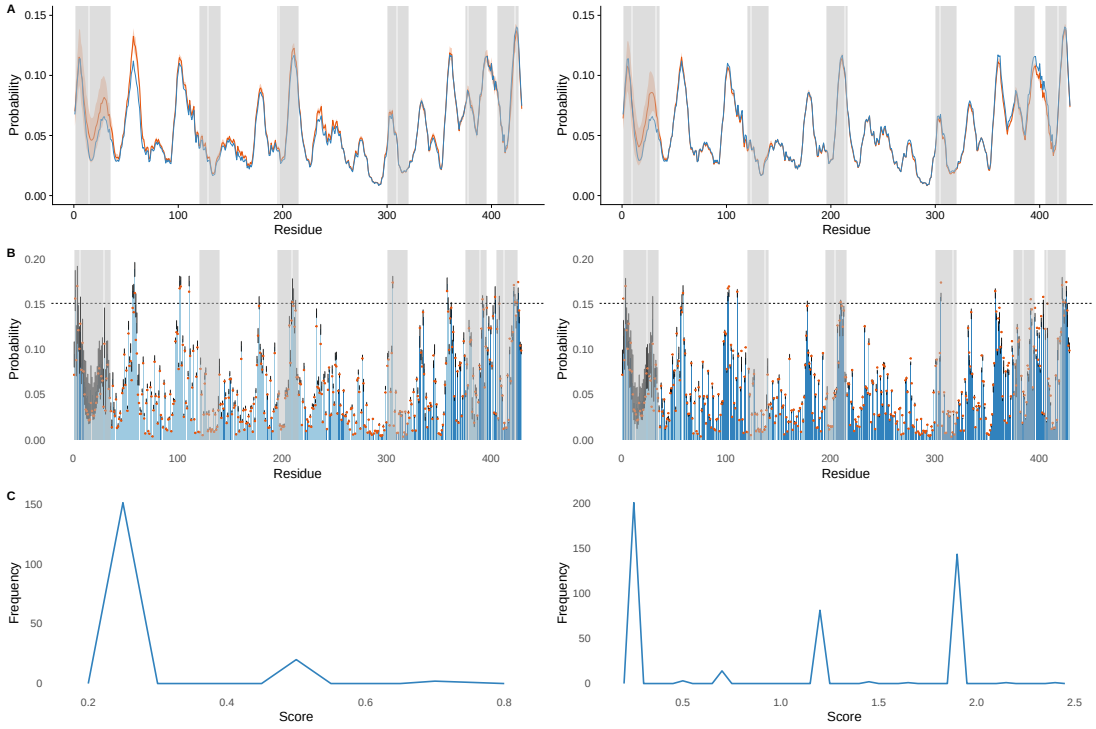

Figure 4.4: **Immunogenicity analysis of generated sequences after** 20 **finetuning epochs.** Comparison of sequence only (left column) and structurally augmented (right column) VAEs in terms of immunogenicity comparison. A) Linear B-cell epitopes' probabilities. Blue line represents WT sequence, red line represents the average of generated sequences and the shaded area represents the standard deviation. The grey area represents positions of the AGAL epitopes that were masked out. B) Discontinuous B-cell epitopes' probabilties. Dots represent the WT sequence, each column represents the average of generated sequences and error bars represent the standard deviation. The darker columns represent positions where the average of generated sequences is lower than the WT sequence. The dotted line represents the threshold of 0.151 recommended by the BEPIRED 3.0 tool. The grey area represents positions of AGAL epitopes that were masked out. C) Humanization scores for generated sequences. The higher the score, the more human-like the sequence is.

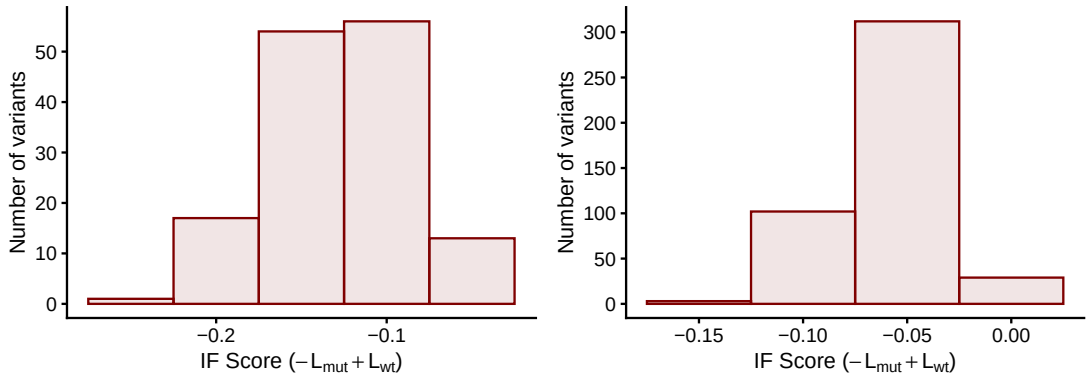

Figure 4.5: **Structurally augmented model generates sequences more compatible with the wild-type structure.** Comparison of sequence only (left column) and structurally augmented (right column) VAEs in terms of compatibility with the wild type structure. For each model, we selected sequences with non-zero humanization scores and full length, resulting in 141 and 446 sequences for the sequence only and structurally augmented model, respectively. We removed the signalling peptide region (1-32) and calculated the IF score for each sequence against the wild-type structure (PDB ID: 1R46).

even though it is a good filtering metric for the model selection. Second, we showed that the benefit of the structural information in the encoder may not be clear at early stages of the experiment, such as pretraining, but it can have a significant impact on downstream tasks, such as finetuning and epitope design in our case.

While our structurally augmented model showed promising results, we are aware of its limitations. First, of all the model is not natively aware of the epitopes and therefore it is not guaranteed that redesigned sequences are not immunogenic. An alternative way to address this problem is to force the model to avoid known epitope generation by using, for example, NLP unlikelihood training technique [Welleck et al., 2019] or by adding some regularization term to the loss function. It would be interesting to see how adding the structural information would affect the model performance in this case. Next, the post-sampling epitope analysis is done using the BEPIRED 3.0 tool, which is suitable for B-cell epitope prediction. However, it is also necessary to assess the T-cell epitopes and additional tools, such as NETMHCIIPAN will be required for additional filtering [Nilsson et al., 2023].

Taken together, our results suggest that masked multimodal VAE outperforms the sequence only VAE in the epitope design problem. The structurally augmented model therefore can be used for the AGAL library design with the objective of reducing the immunogenicity of a variant enzyme and increasing the efficacy of the therapy.

Chapter 5

Conclusions

This thesis investigates the use of AI in protein design, a challenging problem, which has been a subject of research for decades and has a wide range of applications in medicine, industry and biotechnology. While traditional protein design methods, such as Directed evolution (DE), often yield desired results, they are very time-consuming, expensive and require a substantial experimental infrastructure. While we do not expect AI to be a remedy that will immediately give us a perfect protein for whatever application we consider, we believe that AI can significantly accelerate the process of protein design, making it more efficient and cost-effective.

In this work, we focused on the development of computational methods that should facilitate the protein design process, specifically focusing on the design of novel therapeutics for a rare genetic disorder from a group of Lysosomal storage disorders (LSDs), Fabry disease. We fully understand that computational results, no matter how plausible, should be validated experimentally, and while we were focusing on the computational part of the problem, our experimental colleagues were working on the design of the experimental pipeline for the validation of the computational results, a feat that is not less challenging than the computational part of the problem, given the complicated nature of the protein of interest.

In chapter 2 we presented a baseline VAE model that learns evolutionary con-

straints from a small set of homologous sequences. We validated the model on the mutation effect prediction task and showed that it performs comparably to the state-of-the-art methods, while being significantly smaller. In the same task, we justified the use of the Dirichlet latent space, which was shown to be more effective in capturing evolutionary constraints than the standard, commonly used, Gaussian latent space. Unsurprisingly, the Dirichlet latent space was also better at picking up evolutionary constraints for AGAL protein, when we compared our generated samples to published DE results for the same protein. We showed that the model can be used to generate a diverse set of potential ERTs candidates, which maintain biochemical and structural similarities to the wild-type enzyme. Finally, we used the model to design and order a library of 48 AGAL enzyme variants, which are currently being validated experimentally. We already have some preliminary results, showing that some AI-designed variants have twice as much activity as the wild-type enzyme. Once the experimental validation on the first library is complete, we will present our findings in a separate publication.

While chapter 2 was focused on the design of the AGAL enzyme variants using only evolutionary information, in chapter 3 we developed a generative model that focused on a single protein and its mutants in attempt to tractably approximate and explore the free energy landscape of the protein for finding stable mutants. We focused on the *Ec*NAGK protein, a highly conserved non-trivial protein, involved in the L-arginine biosynthesis pathway. We showed that the model can effectively learn the sequence-to-free-energy relationship and use it to generate novel and stable variants of the protein. We selected 40 designed variants for experimental validation and showed that exploring free energy landscape leads to the discovery of stable protein variants with higher number of mutations, when compared to the variant selection based on the likelihood of the model. We achieved a high success rate of 85% in the experimental validation of designed variants. While the whole study was done on a protein, unrelated to the ERT problem, the high success rate of the current experimental validation

gives us confidence that this model can also be used to design an ERT library. Currently, we are working on more robust biophysical simulations, that will allow us to approximate the free energy landscape of the much more complex AGAL protein, and we are planning to use the updated pipeline to design another library of AGAL enzyme variants.

Immunogenicity is a major concern in the development of protein therapeutics. In chapter 4 we tried to lay the groundwork for the development of a generative model that tackles the problem of creating protein variants with modified epitopes, that should reduce the immunogenicity of the protein. We also tested whether the use of the structural information in the model can improve the performance of the model, when compared to the sequence-only model. Until the final stage of the computational part of the study, that is epitope recoding, we did not observe any significant improvement in the model's performance when adding the structural information. However, the epitope redesign part of the study showed that having structural information can be very beneficial. Designed sequences were shown to be less immunogenic and more compatible with the native protein structure, when compared to the sequences from the sequence-only model. While the study is still in its early stages, we believe the that the use of the structural information in the model can be very beneficial, and we are planning to continue its development. Moreover, in addition to B-cell epitope prediction, we need to incorporate the T-cell epitope recognition into the study. Finally, when the pipeline is ready, we will use it to design a library of AGAL enzyme variants with recoded epitopes, and we will also validate designed variants experimentally.

It is worth noting that models developed in this work explore to a larger extent the self-supervised learning approach. Using this framework we rely on designing a large set of potential protein variants, which we then narrow down to a smaller set of candidates using various filters, such as the likelihood of the model, evolutionary constraints, known mutations, binding sites conservation or tools for the prediction of the immunogenicity or stability. We have shown that this approach can be very

effective in the design of the protein variants for very non-trivial proteins, such as AGAL and *Ec*NAGK.

However, the current models have their limitations that we plan to address in the future. As such, all three models are independent to one another, and it would be beneficial to be able to use them in a more integrated way. It is also desirable to incorporate the experimental results into models in order to create an active learning loop. For example, experimental results from the AGAL library could be used to condition the sampling process of the model from Chapter 2 towards mutants that are more likely to be active by preserving mutations that were shown to be beneficial, effectively creating a more targeted library for the second round of experiments. Alternatively, the same information could be used to condition the structurally augmented model from Chapter 4 towards potentially less immunogenic mutants while maintaining the high activity of an enzyme. Finally, by exploring the latent space of either model in a gradient free manner, we could use Simulated annealing (SA) or Monte Carlo (MC) sampling to try to find variants that satisfy multiple criteria at once, as long as we can define an appropriate objective function.

In this work, we focused on the development of VAE models for protein design. However, there exist other generative models, such as Protein Language Models (PLMs) or diffusion models, that could be used for the same purpose. PLMs, based on transformer architecture, have shown to be very effective in generating functional protein sequences across various protein families. The only downside of these models is that they are computationally expensive and require a lot of data to train from scratch. However, the pre-trained models such as ProGen, could be fine-tuned on a smaller dataset to generate protein sequences from AGAL protein family [Madani et al., 2023]. Diffusion models, on the other hand, have shown to be very effective in generating protein structures satisfying multiple input criteria [Watson et al., 2023]. We hypothesize that diffusion models could be very useful to generate protein libraries for problems where structural information is crucial, such as epitope recoding. We plan to explore these

models in the future and compare their performance to the VAE models developed in this work.

During the course of this work, we have shown that relatively small and simple models, when trained on intelligently designed datasets, can be very effective in generating protein variants with desired properties. We have validated their performance both computationally and experimentally on two non-trivial proteins, AGAL and *Ec*NAGK. Additionally, we have shown that incorporating structural information into the model can be very beneficial for certain tasks, such as epitope recoding, and we plan to further develop this approach. From a practical point of view, we have established several computational pipelines that can already be used to design ERT libraries for various LSDs. Meanwhile, we are developing more sophisticated AI models that should allow us to design more targeted libraries in the future. Finally, we hope that results of this work will encourage more rigorous experimental validation of computational models in the field of protein engineering.

Appendix A

Supplementary materials for chapter 2

Supplementary Tables

Dataset	**Avg diff**	**Max diff**	**Top diff**
BRCA1-y2h	-0.49%	4.87%	2.99%
DLG4	-0.09%	1.45%	-0.37%
BLAT-Ranganathan2015	0.02%	2.69%	-0.50%
BLAT-Ostermeier2014	0.18%	2.83%	-0.67%
BLAT-Palzkill2012	0.72%	2.36%	0.09%
BRCA1-e3	1.14%	9.30%	8.04%
BLAT-Tenaillon2013	1.39%	2.04%	1.83%
YAP1	3.10%	7.05%	1.60%

Table A.1: **Average performance comparison across all scenarios.** Each hyperparameter configuration (scenario) for each dataset was run 5 times and the average result for 5 runs was computed. Average difference is computed as the average of differences between TDVAE and TGVAE results for each dataset. Max difference is the maximum difference between TDVAE and TGVAE results for each dataset. Top difference is the difference between TDVAE and TGVAE for the best performing scenario for each dataset.

Dataset	**Avg diff**	**Max diff**	**Top diff**	**TDVAE win**
BLAT-Ostermeier2014	0.43%	2.78%	0.19%	5
BLAT-Palzkill2012	0.74%	2.21%	0.40%	7
BLAT-Ranganathan2015	0.17%	1.79%	0.67%	4
BLAT-Tenaillon2013	1.32%	2.64%	0.60%	7
BRCA1-e3	1.34%	8.90%	6.42%	7
BRCA1-y2h	-0.94%	4.15%	1.00%	3
DLG4	-0.07%	3.23%	0.21%	5
YAP1	1.72%	5.79%	1.55%	7

Table A.2: **Best performance comparison across all scenarios.** Each hyperparameter configuration (scenario) for each dataset was run 5 times and the best result for 5 runs was computed. Average difference is computed as the average of differences between TDVAE and TGVAE results for each dataset. Max difference is the maximum difference between TDVAE and TGVAE results for each dataset. Top difference is the difference between TDVAE and TGVAE for the best performing scenario for each dataset. TDVAE win represents the number of times TDVAE outperformed TGVAE for each dataset.

Appendix B

Supplementary materials for chapter 3

Supplementary Figures

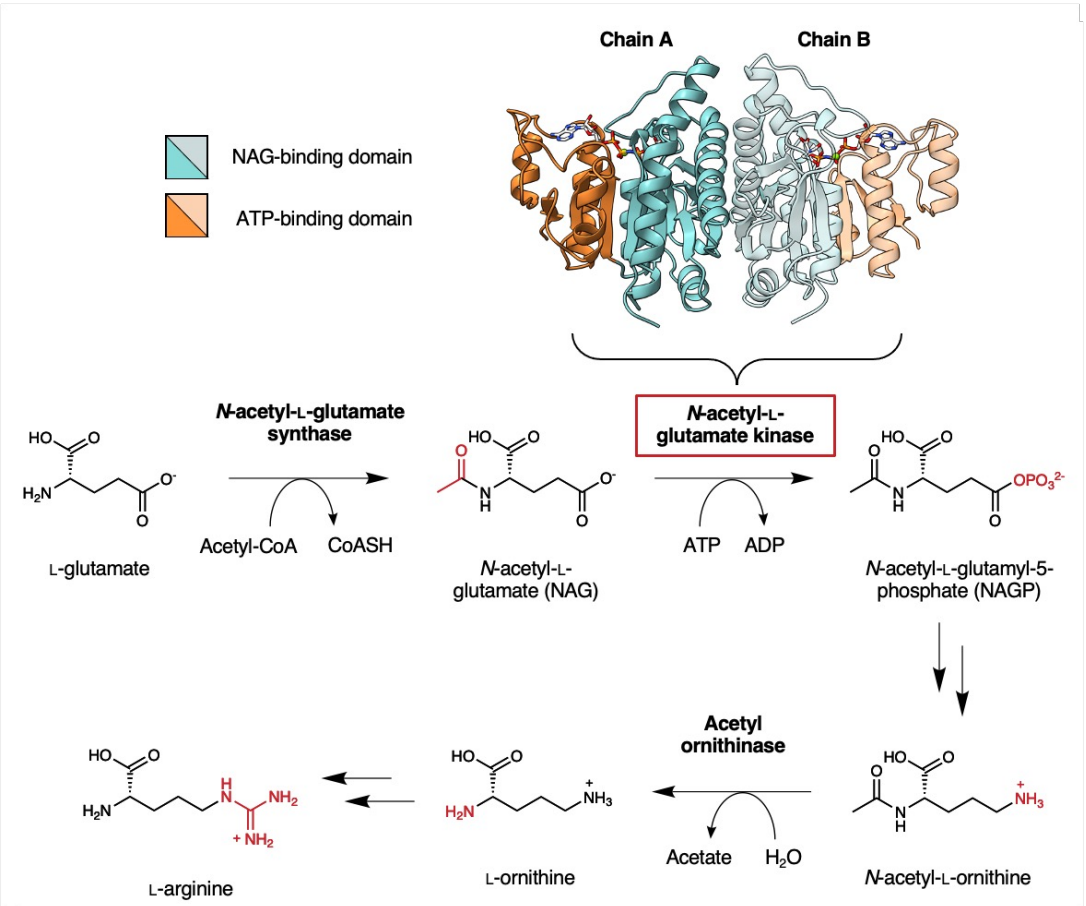

Figure B.1: **Abridged bacterial L-arginine biosynthetic pathway.** The crystal structure of the target *E. coli* phosphotransferase N-acetyl-L-glutamate kinase (*Ec*NAGK) (PDB: 1GS5) is shown.

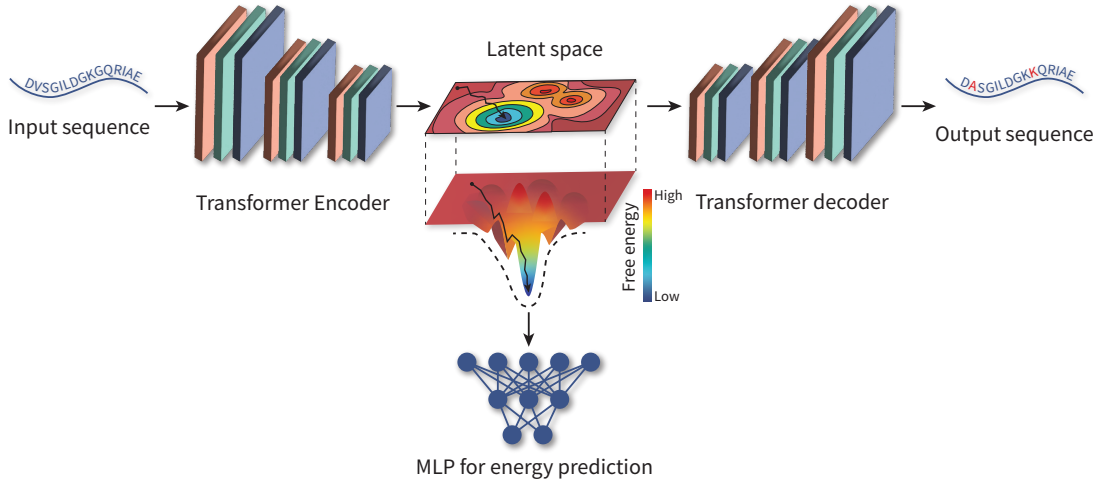

Figure B.2: **PRotein Engineering by Variational frEe eNergy approximaTion (PREVENT).** The model takes in input protein sequences and associated free energy values and uses a transformer encoder to map this information to a latent Gaussian space. Samples from the latent space are then sampled and decoded by transformer decoder, to obtain an amino acid sequence, and a multi-layer perceptron to obtain the expected free energy value.

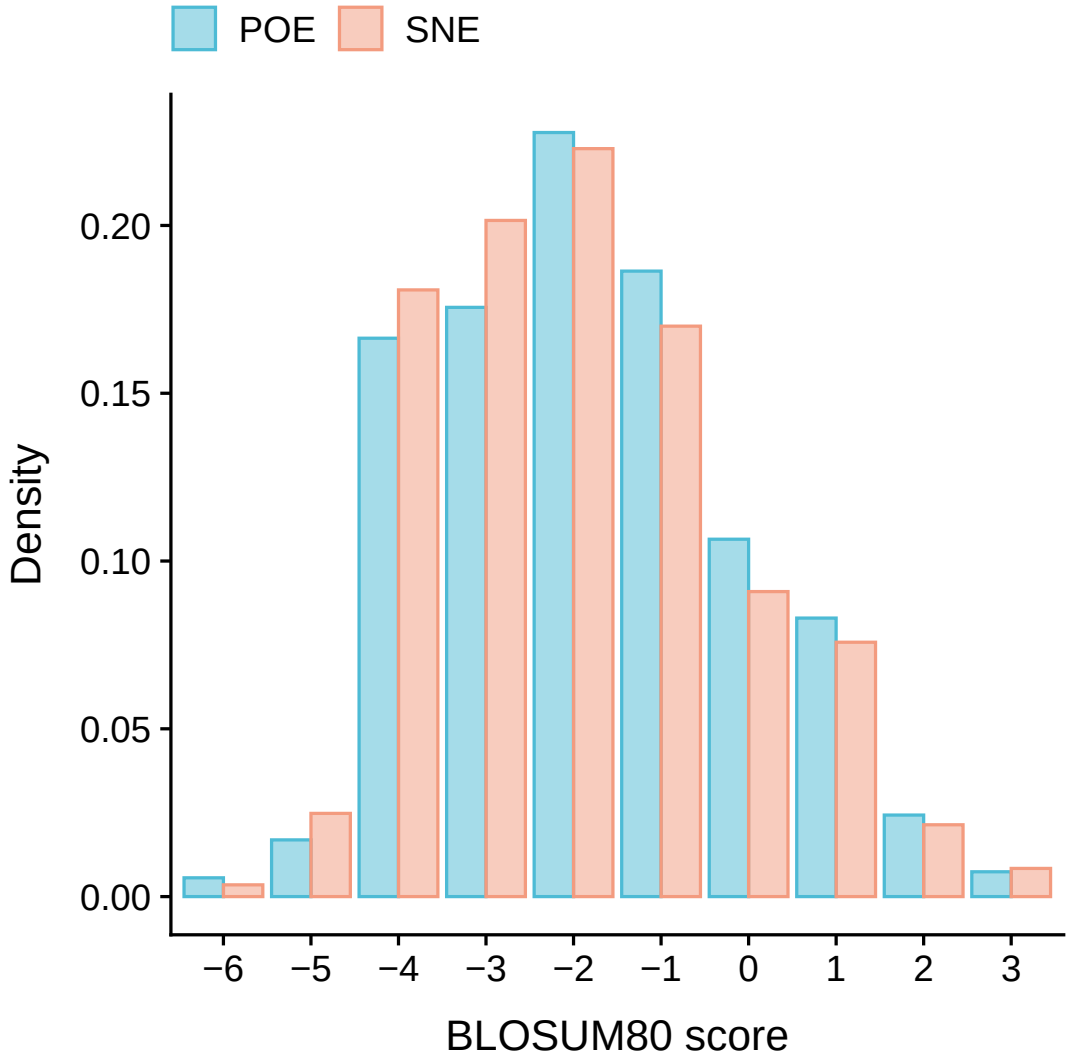

Figure B.3: **Distribution of the BLOSUM80 substitution matrix scores.** Distribution of the BLOSUM80 substitution matrix scores introduced by POE and SNE strategies.

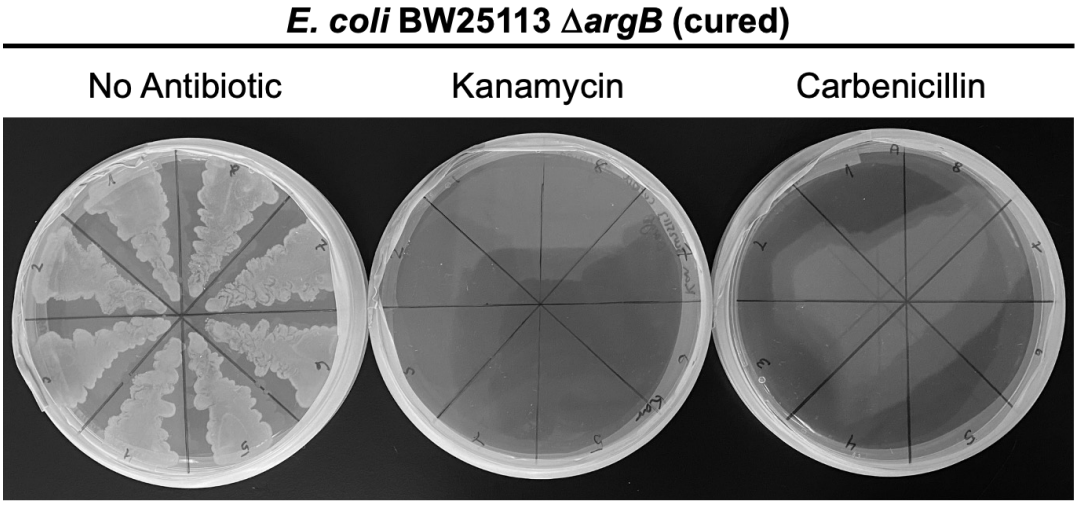

Figure B.4: **Commercial *E. coli* BW25113 ΔargB cured using Flp-FRT recombination.** Eight putatively cured colonies were streaked on selective and nonselective YEP-agar plates to test their re-engineered susceptibility to antibiotics.

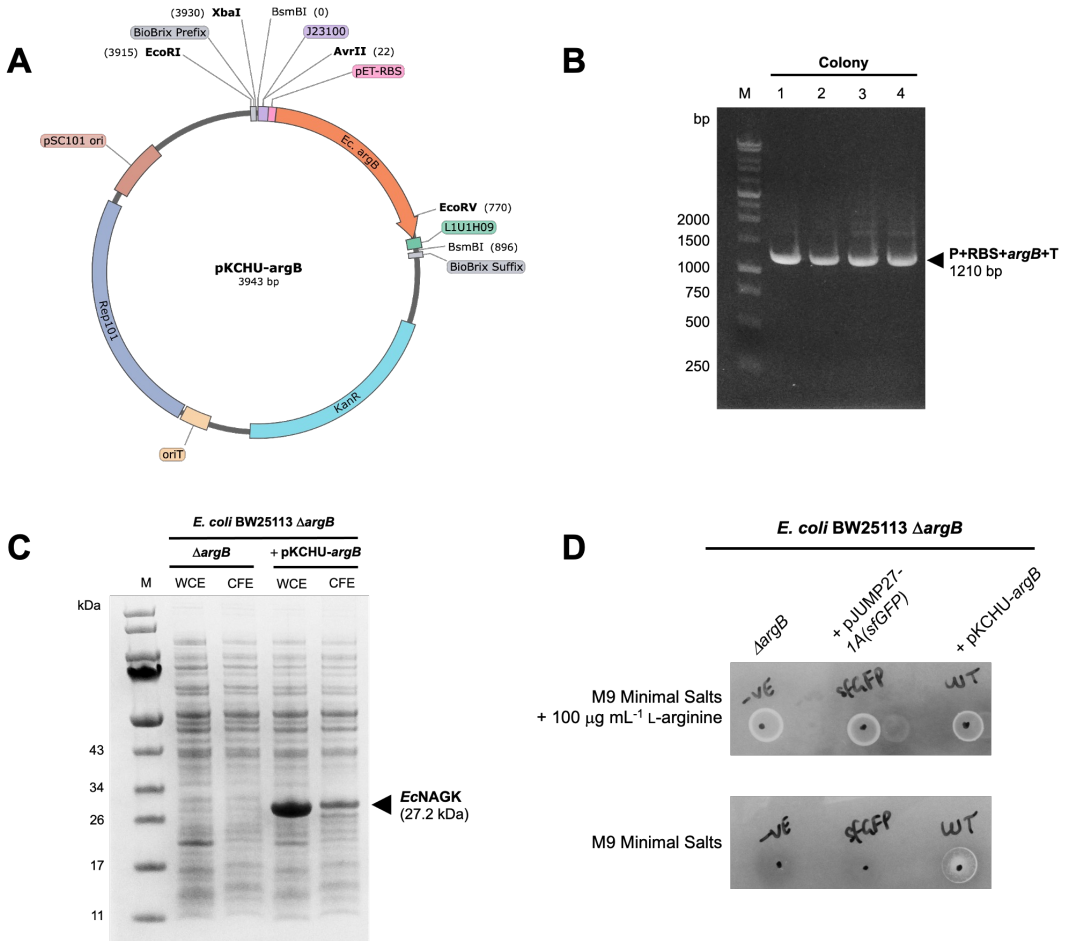

Figure B.5: **Construction and expression of pKCHU-argB.** A) Plasmid map highlighting key features of pKCHU-argB including the promoter (J23100), RBS (pET-RBS) and terminator (L1U1H09). pJUMP27-1A(*sfGFP*) was selected as the destination vector for assembly. B) Colony PCR of NEB5-alpha transformants following the level 0 JUMP assembly of pKCHU-argB. Backbone-specific primers provided complete coverage of the assembled promoter, RBS, CDS and terminator (1210 bp). C) SDS-PAGE analysis of BW25113 ΔargB cultures with and without pKCHU-argB. Whole-cell extracts (WCE) show the total protein content (soluble and insoluble) of the biomass sample. Cell-free extracts (CFE) show soluble protein content of the biomass sample following non-mechanical lysis and lysate clarification. D) Auxotrophic selection of pKCHU-argB transformants on M9 salts minimal media after 48 hours of incubation. Both untransformed and pJUMP27-1A(*sfGFP*)-transformed cells were used as negative controls. A positive control plate containing supplemental L-arginine is also shown.

Figure B.6: **Nucleotide alignment of native and re-coded *E. coli argB* coding sequences.** Visualisation was performed using ESpript 3.0.

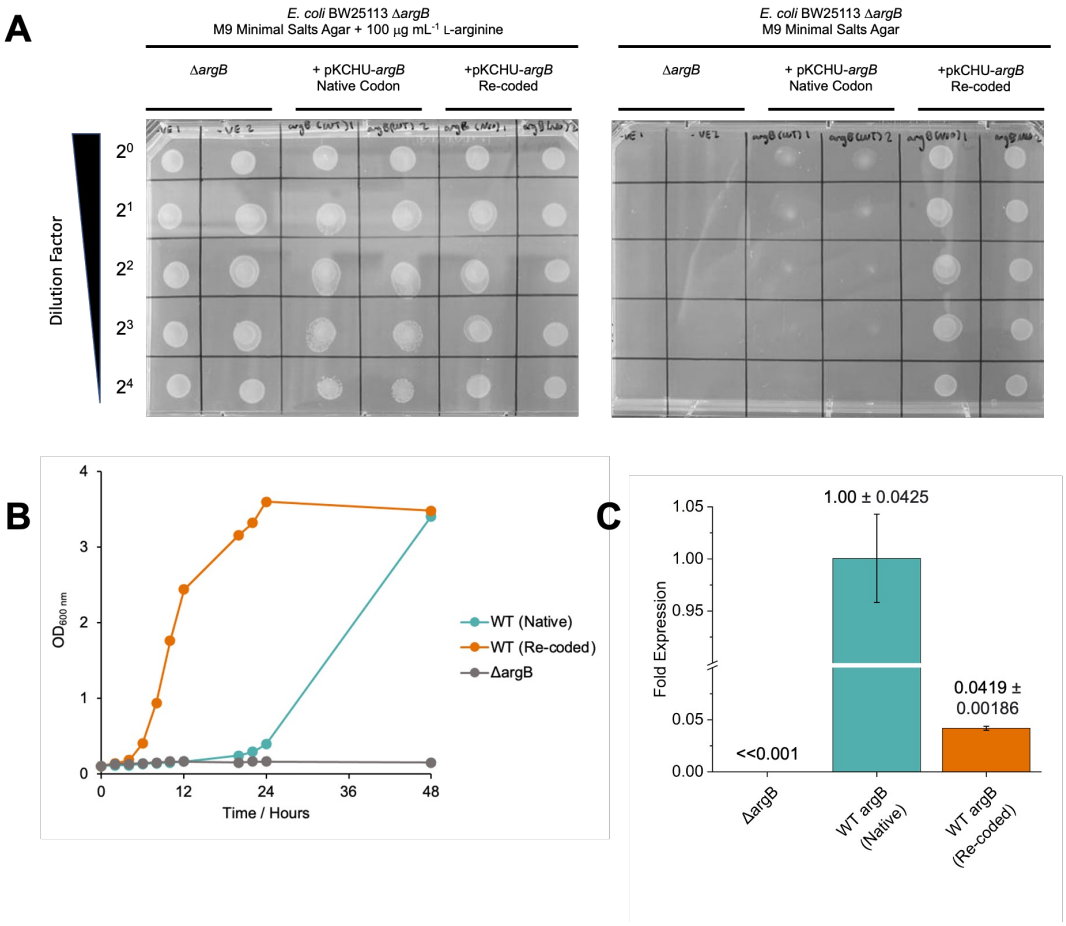

Figure B.7: **Performance of BW25113 ΔargB transformed using native or re-coded pKCHU-*argB* expression constructs.** A) Growth discrepancy between pKCHU-*argB* transformants expressing the native or re-coded *argB* on M9 minimal salts agar. A positive control plate containing supplemental L-arginine is also shown. Images were captured after 24 hours of incubation. Experiments were performed in biological duplicate. B) Growth curve comparison of pKCHU-*argB* transformants expressing native or re-coded *argB* in M9 salts minimal media. C) Relative fold-expression of native and re-coded *argB* determined by RT-qPCR. Error bars represent standard deviation of 3 technical replicates.

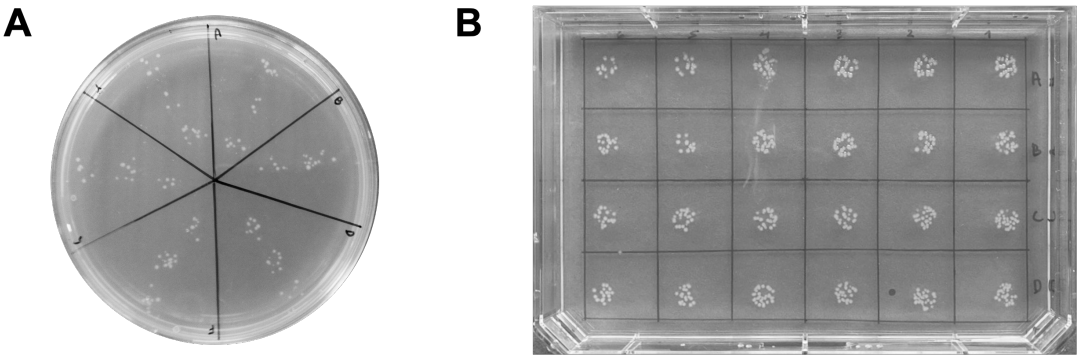

Figure B.8: **Robust *E. coli* transformation using an Opentrons OT-2 robot.** A) E. coli DH5α transformed using pKCHU-*argB* and spotted manually on YEP-kanamycin agar. Each segment represents a single biological replicate. B) *E. coli* DH5α transformed using pET23b-*EGFP* and spotted automatically on YEP-carbenicillin agar. Each spot represents a single biological replicate.

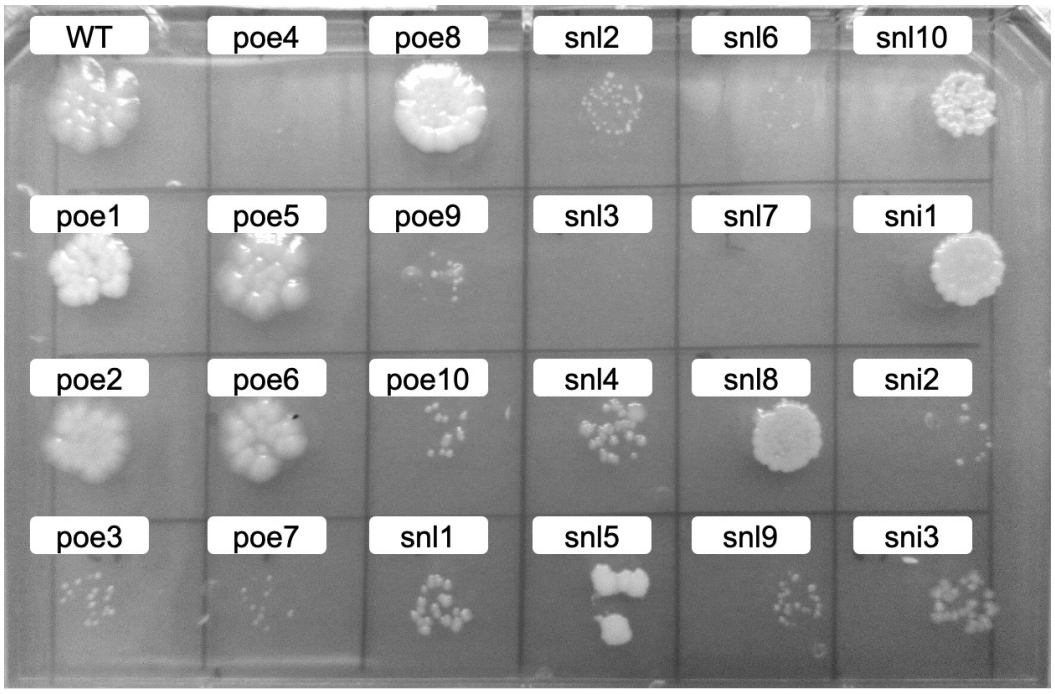

Figure B.9: **Exemplar transformation plate of *Ec*NAGK variants, demonstrating the array of library transformation efficiency.** Image was captured after 48 hours of incubation.

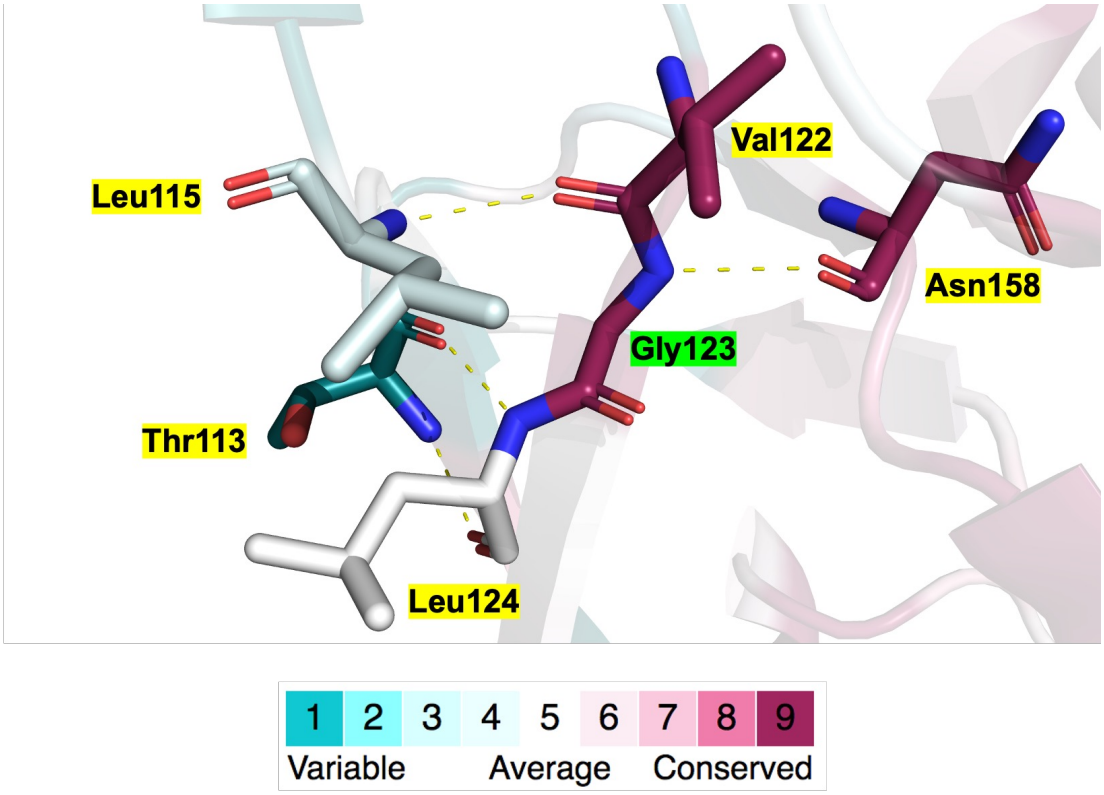

Figure B.10: **Conservation for Gly123.** β-sheet interactions between β6, β7 and β10 in wildtype *Ec*NAGK, with evolutionary conservation score mapping (ConSurf). Gly123 is highlighted in green.

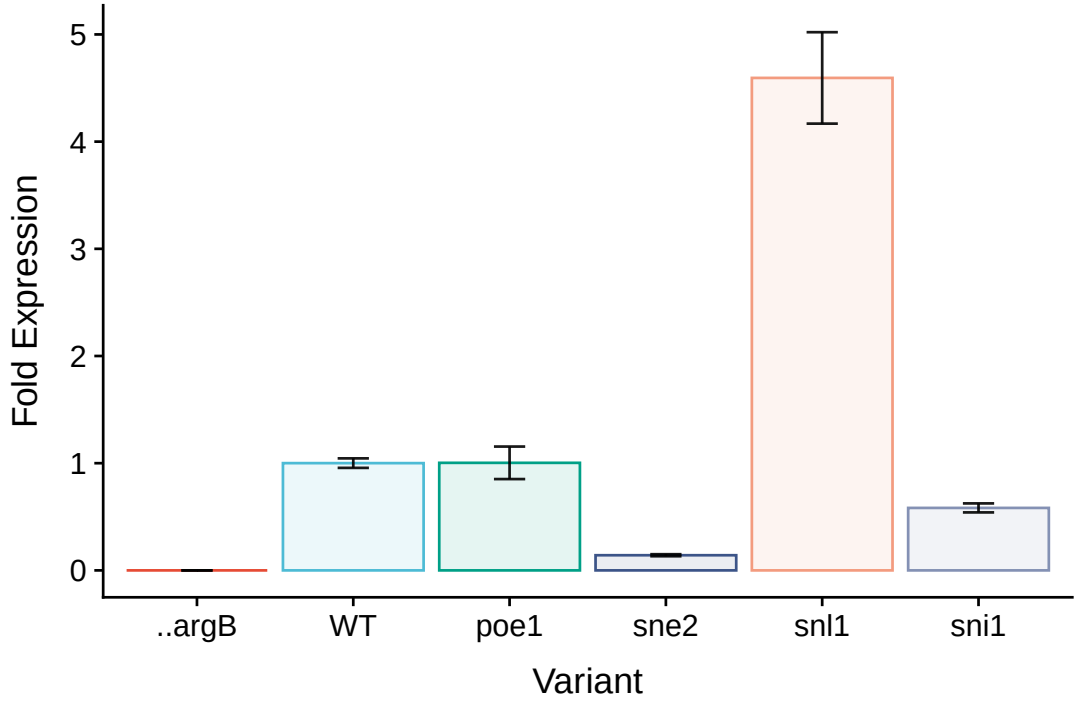

Figure B.11: **RT-qPCR of the top performing candidates from each category.**

Supplementary Tables

Train set size	Reconstuction (PPL)	KL	ELBO	RMSE	Spearman corr
100K	114.80 (0.66)	9.44	124.24	9.27	0.96
75K	115.69 (0.66)	8.94	124.63	9.74	0.95
50K	116.48 (0.66)	8.16	124.64	11.24	0.94
25K	123.42 (0.65)	10.38	133.80	12.12	0.92

Table B.1: **Models performance on test set.** For each size of the training set, upon model convergence, we compute average metrics on the test set.

Part	Part ID	JUMP Part Origin	Description
Promoter	J23100	pJUMP19-J23100_P	Constitutive strong promoter
Ribosome Binding Site	pET-RBS	pJUMP18-RBS-pET_R	pET vector ribosome binding site
Terminator	L1U1H09	pJUMP19-L1U1H09_T	Synthetic terminator
Backbone (destination) vector	pJUMP27-1A(sfGFP)	pJUMP27-1A(sfGFP)	Low copy plasmid with pSC101 origin and superfolder GFP reporter

Table B.2: **Basic parts used for the creation of pKCHU-*argB*.**

Primer	JUMP Primer ID	Sequence
Forward	PS1	AGGGCGGCGGATTTGTCC
Reverse	PS2	GCGGCAACCGAGCGTT

Table B.3: **Sequencing primers used for pKCHU-*argB* expression constructs (wild-type and variant).**

Target	**Primer**	**Sequence (5' → 3')**
rrsA	Forward	TCCAGGTGTAGCGGTGAAAT
	Reverse	TTGAGTTTTAACCTTGCGGC
argB (Native wildtype)	Forward	AGACGAAGGGCAACTGATGA
	Reverse	GCCGCGTTCACTTTCACTAT
argB (Re-coded wildtype + variants)	Forward	GTCCGCTGGTTATCGTTCAC
	Reverse	TAACAGAGTCACCGTCACCC

Table B.4: **RT-qPCR primers used for pKCHU-*argB* expression constructs (native wildtype, re-coded wildtype and variants) and *rrsA*.**

Experiment	**Lag Phase Duration/Hours**	**Standard Error**	**Fold Change**
WT	5.64	0.038	1.00
poe1	6.39	0.304	1.13
snl1	6.22	0.159	1.10
sni1	6.55	0.059	1.16
sne2	4.32	0.192	0.77

Table B.5: **Lag phase duration for selected variants.**

Appendix C

Supplementary materials for chapter 4

Supplementary Figures

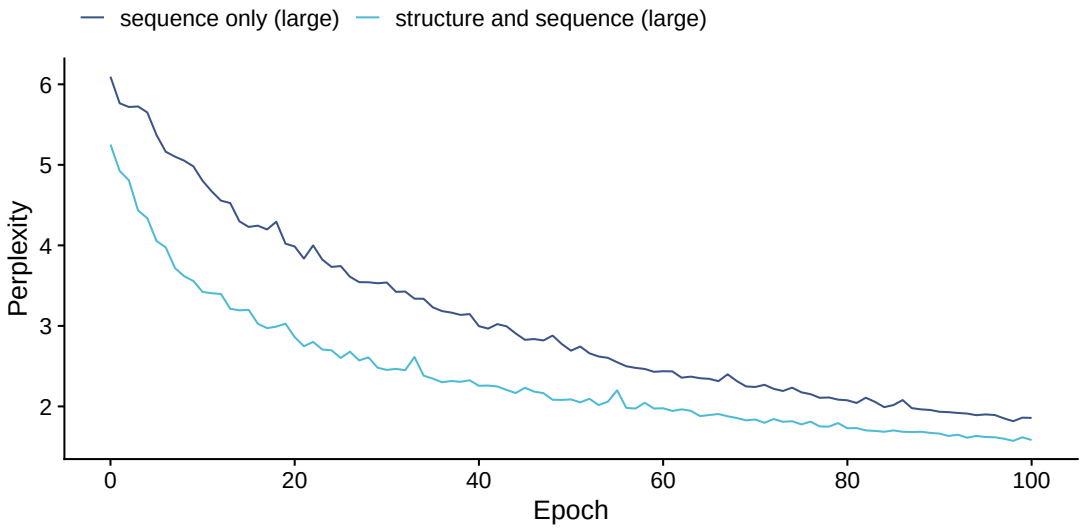

Figure C.1: **Perplexity evolution over finetuning epochs.** The evolution of perplexity on the finetuning dataset is shown for the sequence-to-sequence and structurally augmented sequence-to-sequence models. While both models tend converge to similar perplexity values, the structurally augmented sequence-to-sequence model learns significantly faster.

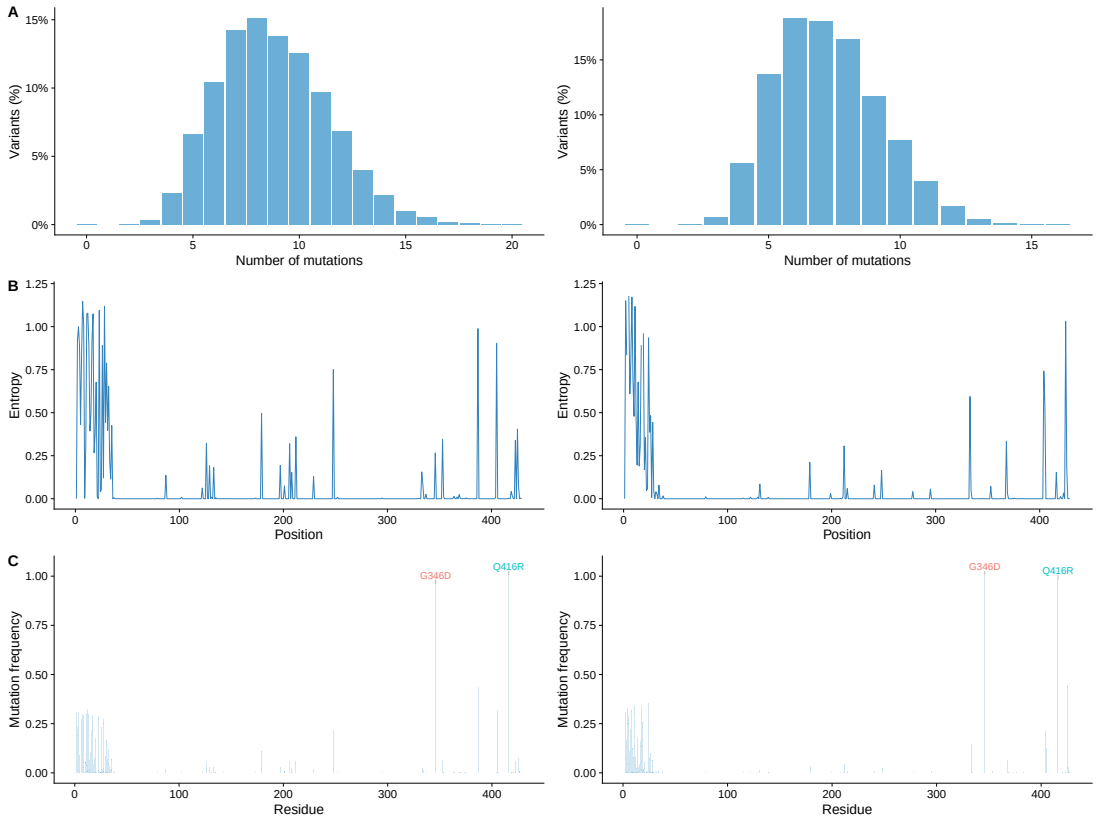

Figure C.2: **Mutation analysis of the sequence only and structurally augmented VAEs after** 100 **finetuning epochs.** Comparison of sequence only (left column) and structurally augmented (right column) VAEs in terms of library diversity. A) Histogram of the number of mutations per generated sequence. B) Residue-level entropy of generated sequences. C) Most frequent mutations.

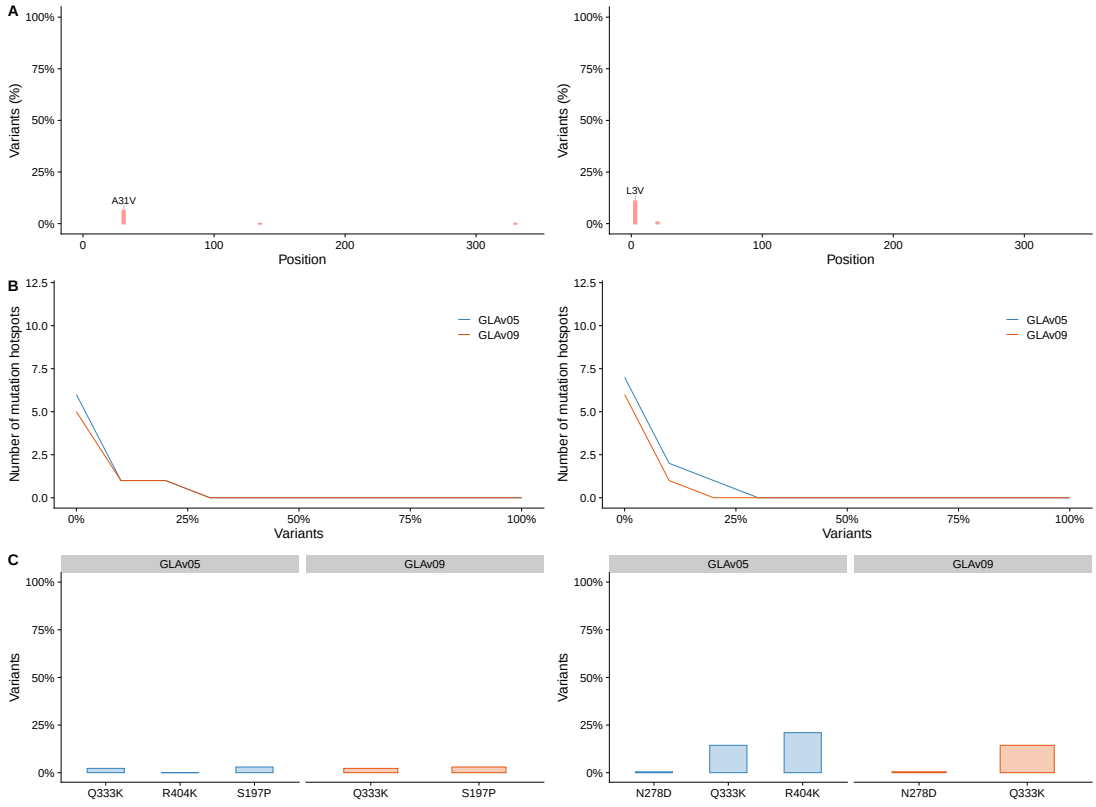

Figure C.3: **Pathogenic and beneficial mutations comparison after** 100 **finetuning epochs.** Comparison of sequence only (left column) and structurally augmented (right column) VAEs in terms of pathogenic and beneficial mutations. A) Frequency of mutations in designed variants associated with Fabry disease or changes in enzymatic activity. B) Percentage of variants harbouring mutations at mutational hotspots identified as GLAv05 and GLAv09. C) Percentage of variants carrying GLAv05 and GLAv09 beneficial mutations.

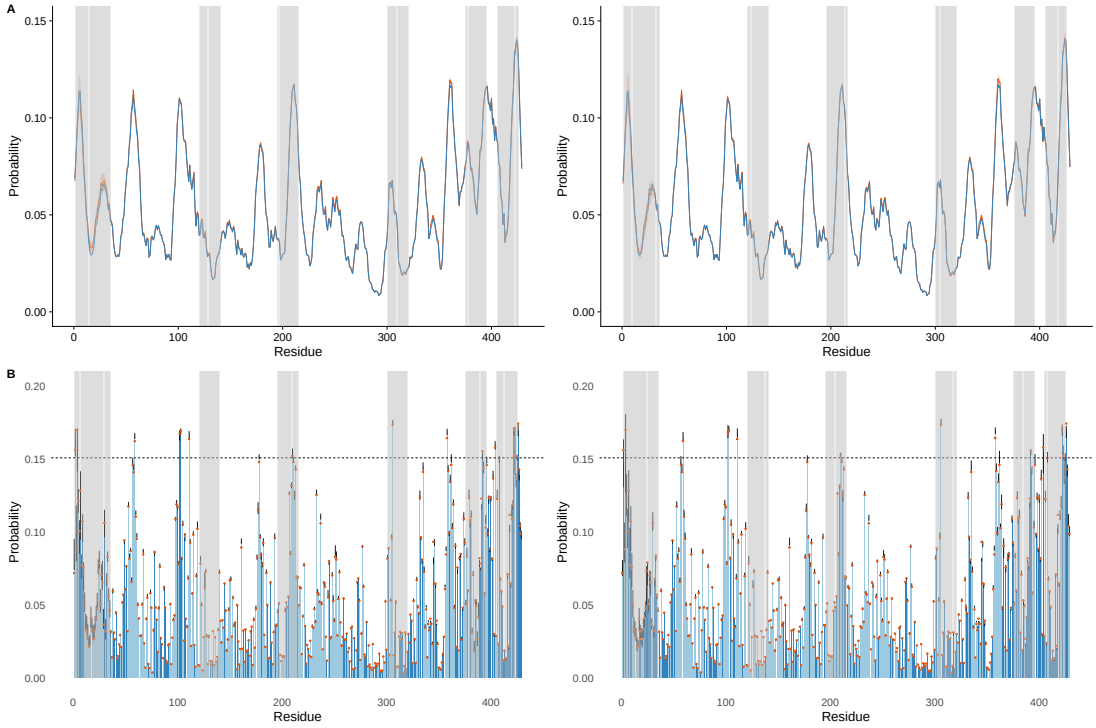

Figure C.4: **Immunogenicity analysis of generated sequences after 100 finetuning epochs.** Comparison of sequence only (left column) and structurally augmented (right column) VAEs in terms of immunogenicity comparison. A) Linear B-cell epitopes' probabilities. Blue line represents WT sequence, red line represents the average of generated sequences and the shaded area represents the standard deviation. The grey area represents positions of the AGAL epitopes that were masked out. B) Discontinuous B-cell epitopes' probabilties. Dots represent the WT sequence, each column represents the average of generated sequences and error bars represent the standard deviation. The darker columns represent positions where the average of the generated sequences is lower than the WT sequence. The dotted line represents the threshold of 0.151 recommended by the BEPIRED 3.0 tool. The grey area represents positions of the AGAL epitopes that were masked out.

Supplementary Tables

Parameters	**Sequence only**		**Sequence & structure**		**Structure only**	
	small	**large**	**small**	**large**	**small**	**large**
AA embedding size	32	64	32	64	32	64
Hidden size	32	64	32	64	128	256
Latent size	128	256	128	256	0.2	0.2
Dropout	0.2	0.2	0.2	0.2	0.2	0.2
# stacked TCN layers	1	1	1	1	1	1
Kernel size	8	16	8	16	8	16
# GVP-GNN layers	NA	NA	3	3	Variable	Variable
Hidden node dim	NA	NA	[128,32]	[256,64]	Variable	Variable
Hidden edge dim	NA	NA	[32,1]	[32,1]	[32,1]	[32,1]
Vector gate	NA	NA	TRUE	TRUE	TRUE	TRUE
Masking	Variable	Variable	Variable	Variable	No	No

Table C.1: **Static model parameters based on encoder type.** For each model category, we provide a static list of parameters for all experiments.

Model type	**Parameters**	**Values**
Sequence only	Masking type	span; random
	Masking %	0.01; 0.05; 0.1; 0.2; 0.4
	Sequence directionality	forward only; forward and reverse
Sequence & structure	Masking type	span; random
	Masking %	0.01; 0.05; 0.1; 0.2; 0.4
	Combination of sequence and structural embeddings	concatenation; element-wise summation
	Add original geometrical features; do diheral embeddings	TRUE; FALSE
Structure only	# GVP-GNN layers	1; 3; 6
	Hidden node dim	[128,32]; [256,64]
	Add original geometrical features; do diheral embeddings	TRUE; FALSE

Table C.2: **Variable model parameters based on encoder type.** For each model category, we provide a list of parameters that vary from experiment to experiment. Cartesian product of all values determine the number of experiments for each model type. For random masking, a given percentage of the sequence is randomly masked out. For span masking, a given percentage of the sequence is masked out in a contiguous manner on average by using geometric distribution. Span masking does not exceed 50% of the sequence length and the minimum span length is 1.

Model type	Experiment	Masking type	Masking %	Sequence features	# parameters	Epoch	ELBO (val)	PPL (val)
small	**experiment-2**	**span**	**0.01**	**Forward & Reverse**	**6 382 487**	**111**	**257,416**	**3,8115**
	experiment-12	random	0.01	Forward & Reverse	6 382 487	102	259,833	3,8373
	experiment-11	random	0.01	Forward only	6 372 215	102	303,194	4,5524
	experiment-1	span	0.01	Forward only	6 372 215	99	303,297	4,5901
	experiment-14	**random**	**0.05**	**Forward & Reverse**	**6 382 487**	**120**	**256,958**	**3,779**
	experiment-4	span	0.05	Forward & Reverse	6 382 487	120	260,695	3,866
	experiment-13	random	0.05	Forward only	6 372 215	177	282,524	4,2363
	experiment-3	span	0.05	Forward only	6 372 215	120	292,037	4,3987
	experiment-16	**random**	**0.1**	**Forward & Reverse**	**6 382 487**	**120**	**271,097**	**4,0738**
	experiment-6	span	0.1	Forward & Reverse	6 382 487	120	276,698	4,2071
	experiment-15	random	0.1	Forward only	6 372 215	183	288,366	4,3719
	experiment-5	span	0.1	Forward only	6 372 215	111	305,64	4,692
	experiment-18	**random**	**0.2**	**Forward & Reverse**	**6 382 487**	**165**	**299,632**	**4,749**
	experiment-8	span	0.2	Forward & Reverse	6 382 487	114	302,076	4,8289
	experiment-17	random	0.2	Forward only	6 372 215	183	316,502	5,087
	experiment-7	span	0.2	Forward only	6 372 215	120	333,167	5,5104
	experiment-10	**span**	**0.4**	**Forward & Reverse**	**6 382 487**	**102**	**334,843**	**5,7386**
	experiment-9	span	0.4	Forward only	6 372 215	96	362,543	6,3141
	experiment-20	random	0.4	Forward & Reverse	6 382 487	120	365,614	6,7844
	experiment-19	random	0.4	Forward only	6 372 215	246	378,888	7,1925
large	**experiment-2**	**span**	**0.01**	**Forward & Reverse**	**21 896 887**	**108**	**92,4689**	**1,6054**
	experiment-12	random	0.01	Forward & Reverse	21 896 887	93	95,853	1,6235
	experiment-1	span	0.01	Forward only	21 823 095	174	129,25	1,8864
	experiment-11	random	0.01	Forward only	21 823 095	183	138,833	1,9672
	experiment-14	**random**	**0.05**	**Forward & Reverse**	**21 896 887**	**165**	**103,656**	**1,7022**
	experiment-4	span	0.05	Forward & Reverse	21 896 887	96	113,001	1,7814
	experiment-13	random	0.05	Forward only	21 823 095	180	131,462	1,9174
	experiment-3	span	0.05	Forward only	21 823 095	168	140,24	1,9963
	experiment-16	**random**	**0.1**	**Forward & Reverse**	**21 896 887**	**123**	**125,498**	**1,9083**
	experiment-6	span	0.1	Forward & Reverse	21 896 887	117	134,825	2,0099
	experiment-15	random	0.1	Forward only	21 823 095	186	150,063	2,122
	experiment-5	span	0.1	Forward only	21 823 095	123	163,828	2,2163
	experiment-8	**span**	**0.2**	**Forward & Reverse**	**21 896 887**	**60**	**171,123**	**2,4307**
	experiment-18	random	0.2	Forward & Reverse	21 896 887	123	174,134	2,468
	experiment-17	random	0.2	Forward only	21 823 095	186	191,492	2,6434
	experiment-7	span	0.2	Forward only	21 823 095	168	193,955	2,6667
	experiment-10	**span**	**0.4**	**Forward & Reverse**	**21 896 887**	**117**	**213,587**	**3,0681**
	experiment-9	span	0.4	Forward only	21 823 095	114	256,515	3,6377
	experiment-20	random	0.4	Forward & Reverse	21 896 887	150	264,371	3,9858
	experiment-19	random	0.4	Forward only	21 823 095	180	282,186	4,2976

Table C.3: **Sequence-to-sequence VAE model performance.** Pretraining results for sequence-to-sequence VAE models. Results are grouped by masking probability and the best performing model is highlighted in bold.

Model type	Experiment	Masking type	Masking %	Features aggregation	Add original geometrical features	Dihedral embedding	# parameters	Epoch	ELBO (val)	PPL (val)
small	**experiment-24**	**random**	**0.01**	**concat**	**FALSE**	**FALSE**	**7 392 444**	**96**	**250,1038**	**3,6337**
	experiment-4	span	0.01	concat	FALSE	FALSE	7 392 444	105	251,946	3,6717
	experiment-3	span	0.01	concat	TRUE	TRUE	7 393 244	111	260,8341	3,8788
	experiment-23	random	0.01	concat	TRUE	TRUE	7 393 244	93	263,3727	3,9049
	experiment-21	random	0.01	sum	TRUE	TRUE	7 382 972	108	275,0119	3,9758
	experiment-2	span	0.01	sum	FALSE	FALSE	7 382 172	111	280,660	4,1517
	experiment-22	random	0.01	sum	FALSE	FALSE	7 382 172	105	290,3904	4,317
	experiment-1	span	0.01	sum	TRUE	TRUE	7 382 972	123	293,0776	4,397
	experiment-8	**span**	**0.05**	**concat**	**FALSE**	**FALSE**	**7 392 444**	**165**	**254,7766**	**3,7354**
	experiment-28	random	0.05	concat	FALSE	FALSE	7 392 444	123	255,3843	3,7508
	experiment-27	random	0.05	concat	TRUE	TRUE	7 393 244	159	256,5532	3,7754
	experiment-7	span	0.05	concat	TRUE	TRUE	7 393 244	120	257,9938	3,8043
	experiment-25	random	0.05	sum	TRUE	TRUE	7 382 972	123	279,1986	4,1026
	experiment-26	random	0.05	sum	FALSE	FALSE	7 382 172	123	283,6013	4,1644
	experiment-6	span	0.05	sum	FALSE	FALSE	7 382 172	120	278,4396	4,3249
	experiment-5	span	0.05	sum	TRUE	TRUE	7 382 972	123	291,1079	5,5622
	experiment-32	**random**	**0.1**	**concat**	**FALSE**	**FALSE**	**7 392 444**	**174**	**269,7627**	**4,0619**
	experiment-31	random	0.1	concat	TRUE	TRUE	7 393 244	123	270,6915	4,0656
	experiment-12	span	0.1	concat	FALSE	FALSE	7 392 444	123	272,0812	4,1016
	experiment-11	span	0.1	concat	TRUE	TRUE	7 393 244	120	272,4649	4,1068
	experiment-29	random	0.1	sum	TRUE	TRUE	7 382 972	186	280,1748	4,1905
	experiment-30	random	0.1	sum	FALSE	FALSE	7 382 172	123	291,3793	4,3692
	experiment-10	span	0.1	sum	FALSE	FALSE	7 382 172	150	293,5545	4,4122
	experiment-9	span	0.1	sum	TRUE	TRUE	7 382 972	120	305,7293	4,6907
	experiment-36	**random**	**0.2**	**concat**	**FALSE**	**FALSE**	**7 392 444**	**168**	**294,3517**	**4,6129**
	experiment-15	span	0.2	concat	TRUE	TRUE	7 393 244	117	293,9747	4,6147
	experiment-16	span	0.2	concat	FALSE	FALSE	7 392 444	102	297,5231	4,6737
	experiment-35	random	0.2	concat	TRUE	TRUE	7 393 244	123	304,3444	4,8778
	experiment-34	random	0.2	sum	FALSE	FALSE	7 382 172	183	309,9515	4,8795
	experiment-14	span	0.2	sum	FALSE	FALSE	7 382 172	96	320,0418	5,052
	experiment-13	span	0.2	sum	TRUE	TRUE	7 382 972	108	332,2865	5,3838
	experiment-33	random	0.2	sum	TRUE	TRUE	7 382 972	60	358,3993	6,1939
	experiment-19	**span**	**0.4**	**concat**	**TRUE**	**TRUE**	**7 393 244**	**117**	**323,9905**	**5,4155**
	experiment-20	span	0.4	concat	FALSE	FALSE	7 392 444	105	326,2264	5,4639
	experiment-18	span	0.4	sum	FALSE	FALSE	7 382 172	123	347,3329	5,937
	experiment-40	random	0.4	concat	FALSE	FALSE	7 392 444	252	346,0853	6,103
	experiment-17	span	0.4	sum	TRUE	TRUE	7 382 972	123	358,6085	6,3141
	experiment-39	random	0.4	concat	TRUE	TRUE	7 393 244	249	352,5478	6,319
	experiment-37	random	0.4	sum	TRUE	TRUE	7 382 972	261	360,672	6,4827
	experiment-38	random	0.4	sum	FALSE	FALSE	7 382 172	261	360,598	6,505

Table C.4: **Structure-enhanced VAE small model performance.** Pretraining results for structure-augmented sequence-to-sequence VAE models. Results are grouped by masking probability and the best performing model is highlighted in bold.

Model type	Experiment	Masking type	Masking %	Features aggregation	Add original geometrical features	Dihedral embedding	# parameters	Epoch	ELBO (val)	PPL (val)
large	**experiment-24**	**random**	**0.01**	**concat**	**FALSE**	**FALSE**	**25 894 012**	**108**	**89,2176**	**1,579**
	experiment-3	span	0.01	concat	TRUE	TRUE	25 895 612	93	92,682	1,6005
	experiment-23	random	0.01	concat	TRUE	TRUE	25 895 612	117	92,1685	1,6046
	experiment-4	span	0.01	concat	FALSE	FALSE	25 894 012	120	97,1828	1,6559
	experiment-21	random	0.01	sum	TRUE	TRUE	25 821 820	117	114,2461	1,6718
	experiment-2	span	0.01	sum	FALSE	FALSE	25 820 220	123	124,4982	1,7482
	experiment-1	span	0.01	sum	TRUE	TRUE	25 821 820	120	124,7512	1,7578
	experiment-22	random	0.01	sum	FALSE	FALSE	25 820 220	117	128,4847	1,7976
	experiment-7	**span**	**0.05**	**concat**	**TRUE**	**TRUE**	**25 895 612**	**93**	**103,6303**	**1,697**
	experiment-28	random	0.05	concat	FALSE	FALSE	25 894 012	96	105,0166	1,7043
	experiment-27	random	0.05	concat	TRUE	TRUE	25 895 612	117	104,2948	1,7052
	experiment-8	span	0.05	concat	FALSE	FALSE	25 894 012	123	107,0183	1,7302
	experiment-25	random	0.05	sum	TRUE	TRUE	25 821 820	117	132,1111	1,8332
	experiment-26	random	0.05	sum	FALSE	FALSE	25 820 220	123	133,6633	1,8581
	experiment-5	span	0.05	sum	TRUE	TRUE	25 821 820	111	137,3097	1,8692
	experiment-6	span	0.05	sum	FALSE	FALSE	25 820 220	123	138,2346	1,8853
	experiment-11	**span**	**0.1**	**concat**	**TRUE**	**TRUE**	**25 895 612**	**120**	**122,3818**	**1,8822**
	experiment-12	span	0.1	concat	FALSE	FALSE	25 894 012	147	126,4297	1,9168
	experiment-32	random	0.1	concat	FALSE	FALSE	25 894 012	117	126,4814	1,9194
	experiment-31	random	0.1	concat	TRUE	TRUE	25 895 612	117	127,8616	1,9343
	experiment-29	random	0.1	sum	TRUE	TRUE	25 821 820	123	150,0732	2,0347
	experiment-30	random	0.1	sum	FALSE	FALSE	25 820 220	117	153,9259	2,0531
	experiment-9	span	0.1	sum	TRUE	TRUE	25 821 820	120	155,2407	2,0872
	experiment-10	span	0.1	sum	FALSE	FALSE	25 820 220	123	156,3651	2,0879
	experiment-15	**span**	**0.2**	**concat**	**TRUE**	**TRUE**	**25 895 612**	**123**	**151,7154**	**2,1937**
	experiment-16	span	0.2	concat	FALSE	FALSE	25 894 012	81	158,4911	2,2522
	experiment-36	random	0.2	concat	FALSE	FALSE	25 894 012	183	163,5238	2,3393
	experiment-13	span	0.2	sum	TRUE	TRUE	25 821 820	183	176,9037	2,391
	experiment-35	random	0.2	concat	TRUE	TRUE	25 895 612	162	168,8039	2,4025
	experiment-14	span	0.2	sum	FALSE	FALSE	25 820 220	123	186,0683	2,4693
	experiment-34	random	0.2	sum	FALSE	FALSE	25 820 220	183	183,4959	2,48
	experiment-33	random	0.2	sum	TRUE	TRUE	25 821 820	168	188,2028	2,5216
	experiment-19	**span**	**0.4**	**concat**	**TRUE**	**TRUE**	**25 895 612**	**123**	**190,9857**	**2,7051**
	experiment-20	span	0.4	concat	FALSE	FALSE	25 894 012	81	197,223	2,7593
	experiment-18	span	0.4	sum	FALSE	FALSE	25 820 220	108	230,0809	3,1117
	experiment-17	span	0.4	sum	TRUE	TRUE	25 821 820	120	230,5317	3,1291
	experiment-37	random	0.4	sum	TRUE	TRUE	25 821 820	123	270,9158	3,9181
	experiment-40	random	0.4	concat	FALSE	FALSE	25 894 012	111	261,9495	3,9285
	experiment-39	random	0.4	concat	TRUE	TRUE	25 895 612	102	275,5329	4,2184
	experiment-38	random	0.4	sum	FALSE	FALSE	25 820 220	81	311,2768	4,4661

Table C.5: **Structure-enhanced VAE large model performance.** Pretraining results for structure-augmented sequence-to-sequence VAE models. Results are grouped by masking probability and the best performing model is highlighted in bold.

Mode type	Experiment	Hidden node dim	# GVP-GNN layers	Add original geometrical features	Dihedral embedding	# parameters	Epoch	ELBO (val)	PPL (val)
small	experiment-1	[128, 32]	1	TRUE	TRUE	6 718 746	15	540.2744	17.272
	experiment-2	[128, 32]	3	TRUE	TRUE	7 382 236	21	539.1515	17.2315
	experiment-3	[128, 32]	6	TRUE	TRUE	8 377 471	21	540.9797	17.3741
	experiment-4	[256, 64]	1	TRUE	TRUE	7 725 530	24	540.0201	17.3208
	experiment-5	[256, 64]	3	TRUE	TRUE	10 355 036	24	539.7643	17.3033
	experiment-6	[256, 64]	6	TRUE	TRUE	14 299 295	24	535.9457	17.0449
	experiment-7	[128, 32]	1	FALSE	FALSE	6 717 946	18	538.2144	17.1225
	experiment-8	[128, 32]	3	FALSE	FALSE	7 381 436	15	538.205	17.1042
	experiment-9	[128, 32]	6	FALSE	FALSE	8 376 671	21	534.8288	16.8521
	experiment-10	[256, 64]	1	FALSE	FALSE	7 724 730	21	537.7892	17.1046
	experiment-11	[256, 64]	3	FALSE	FALSE	10 354 236	18	537.5681	17.0695
	experiment-12	[256, 64]	6	FALSE	FALSE	14 298 495	21	531.9146	16.6917
large	experiment-1	[128, 32]	1	TRUE	TRUE	22 176 890	6	540.4718	17.053
	experiment-2	[128, 32]	3	TRUE	TRUE	22 840 380	6	540.8283	17.1047
	experiment-3	[128, 32]	6	TRUE	TRUE	23 835 615	6	539.8746	16.9471
	experiment-4	[256, 64]	1	TRUE	TRUE	23 190 842	6	540.4792	17.0639
	experiment-5	[256, 64]	3	TRUE	TRUE	25 820 348	9	537.5806	16.8675
	experiment-6	[256, 64]	6	TRUE	TRUE	29 764 607	9	530.578	16.3338
	experiment-7	[128, 32]	1	FALSE	FALSE	22 175 290	6	538.6372	16.9308
	experiment-8	[128, 32]	3	FALSE	FALSE	22 838 780	6	538.2638	16.8961
	experiment-9	[128, 32]	6	FALSE	FALSE	23 834 015	12	535.7665	16.7812
	experiment-10	[256, 64]	1	FALSE	FALSE	23 189 242	12	535.0261	16.7386
	experiment-11	[256, 64]	3	FALSE	FALSE	25 818 748	12	532.8125	16.5719
	experiment-12	[256, 64]	6	FALSE	FALSE	29 763 007	15	521.3619	15.7574

Table C.6: **Structure-to-sequence (Inverse Fold) VAE model performance.** Pre-training results for structure-to-sequence VAE models.

Protein	Metrics	Small model		Large model	
		Sequence only	**Sequence & structure**	**Sequence only**	**Sequence & structure**
1b6q (63AA)	# hits	19569	17670	3663	18473
	avg identity	95,4878	97,1281	98,8953	99,0206
	avg similarity	95,8724	97,143	98,9794	99,0714
	avg bitscore	119,8136	134,5368	129,585	135,0011
	avg query coverage	53,2071	44,4352	76,2869	50,6114
1d1r (116AA)	# hits	18034	15698	1753	3322
	avg identity	95,5168	96,9801	97,8729	97,6644
	avg similarity	95,876	96,9924	98,1003	97,758
	avg bitscore	199,4063	208,9031	222,7261	221,2291
	avg query coverage	92,5427	85,8542	99,8311	99,7158
1aol (226AA)	# hits	19992	17427	19995	19993
	avg identity	84,8807	89,7229	93,5592	93,9597
	avg similarity	85,5399	90,0173	94,0311	93,9935
	avg bitscore	158,9456	146,724	380,0182	382,0936
	avg query coverage	70,4424	70,6555	99,9544	99,803
1ift (263AA)	# hits	19656	19887	20000	19997
	avg identity	91,4462	94,6257	93,52	94,6106
	avg similarity	92,9621	95,4666	94,6526	95,2726
	avg bitscore	235,8432	255,6739	455,7487	465,0784
	avg query coverage	56,8205	55,0556	92,2801	92,4607
1sji (350AA)	# hits	19993	19324	20000	19991
	avg identity	90,2587	92,1685	91,9479	91,4271
	avg similarity	92,1383	93,0186	94,3024	92,3718
	avg bitscore	198,6699	208,6248	420,2625	418,6925
	avg query coverage	41,6129	59,1255	69,7759	72,1725
3qcp (470AA)	# hits	19190	19084	20004	20000
	avg identity	82,3326	84,5771	84,0707	83,8178
	avg similarity	85,1299	85,4082	87,282	86,5407
	avg bitscore	208,3102	206,8835	400,7839	400,6601
	avg query coverage	34,8455	62,8874	50,7577	53,9008
agal (429AA)	# hits	19801	17761	20001	20002
	avg identity	82,2346	87,608	86,4672	85,7176
	avg similarity	85,5217	88,1249	87,7884	87,3397
	avg bitscore	203,8584	192,4352	427,7278	428,8038
	avg query coverage	44,6246	70,6251	59,3636	60,7243

Table C.7: **Sampling results for selected pretrained models.** For sampling procedure 6 proteins of varied length were randomly selected from test set of CATH 4.2 in addition to full-length AGAL protein. These proteins were used to obtain parameters of variational posterior distributions $q_\phi(z|x)$ or $q_\phi(z|x,s)$, from which $20,000$ samples were drawn. BLASTP was used to compare these samples to the wild-type sequences. Duplicated hits as well as hits above 0.001 e-value threshold and below 25% query coverage were discarded.

Epitope	**Start position**	**End position**
QLRNPELHLGCALALRFLA	2	20
LVSWDIPGARALDNG	21	35
ANYVHSKGLKLGIYADVGNK	121	140
RSIVYSCEWPLYMWPFQKPN	196	215
RHISPQAKALLQDKDVIAIN	301	320
VACNPACFITQLLPVKRKLG	376	395
HINPTGTVLLQLENTMQMSL	406	425

Table C.8: **Selected epitopes.** Selected non-overlapping epitopes were used for partial (up to 10 amino acids) masking.

Sampling time	**Metrics**	**Large model**	
		Sequence only	**Sequence & structure**
after 20 epochs	# hits	10009	10063
	avg identity	92,44	93,34
	avg similarity	94,77	95,70
	avg bitscore	816,42	830,77
	avg query coverage	99,95	99,90
after 100 epochs	# hits	9589	7848
	avg identity	97,96	98,31
	avg similarity	98,75	99,00
	avg bitscore	877,40	879,45
	avg query coverage	100,00	99,99

Table C.9: **Sampling results for AGAL with partially masked epitopes.** 7 epitopes were partially masked up to 10 amino acids $10,000$ times randomly and independently and used to obtain parameters of variational posterior distributions $q_\phi(z|x,s)$, from which 5 samples were drawn for each partially masked sequence. BLASTP was used to compare these samples to the wild-type AGAL. Duplicated hits as well as hits above 0.001 e-value threshold and below 25% query coverage were discarded.

Bibliography

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, pages 1–3.

Akhtar, M. and Elliott, P. (2018). Anderson-fabry disease in heart failure. *Biophysical Reviews*, 10(4):1107–1119.

Anishchenko, I., Pellock, S. J., Chidyausiku, T. M., Ramelot, T. A., Ovchinnikov, S., Hao, J., Bafna, K., Norn, C., Kang, A., Bera, A. K., et al. (2021). De novo protein design by deep network hallucination. *Nature*, 600(7889):547–552.

Arnold, F. H. (1996). Directed evolution: creating biocatalysts for the future. *Chemical engineering science*, 51(23):5091–5102.

Arnold, F. H. (2018). Directed evolution: bringing new chemistry to life. *Angewandte Chemie (International Ed. in English)*, 57(16):4143.

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Banikazemi, M., Bultas, J., Waldek, S., Wilcox, W. R., Whitley, C. B., McDonald, M., Finkel, R., Packman, S., Bichet, D. G., Warnock, D. G., et al. (2007). Agalsidase-beta therapy for advanced Fabry disease: a randomized trial. *Annals of internal medicine*, 146(2):77–86.

Ben Chorin, A., Masrati, G., Kessel, A., Narunsky, A., Sprinzak, J., Lahav, S., Ashkenazy, H., and Ben-Tal, N. (2020). ConSurf-DB: An accessible repository for the evolutionary conservation patterns of the majority of PDB proteins. *Protein Science*, 29(1):258–267.

Brown, T. B. (2020). Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

Burlina, A. P., Sims, K. B., Politei, J. M., Bennett, G. J., Baron, R., Sommer, C., Møller, A. T., and Hilz, M. J. (2011). Early diagnosis of peripheral nervous system involvement in Fabry disease and treatment of neuropathic pain: the report of an expert panel. *BMC neurology*, 11:1–11.

Byrd, R. H., Hribar, M. E., and Nocedal, J. (1999). An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900.

Cai, Y.-D., Liu, X.-J., Xu, X.-b., and Zhou, G.-P. (2001). Support vector machines for predicting protein structural class. *BMC bioinformatics*, 2:1–5.

Caudai, C., Galizia, A., Geraci, F., Le Pera, L., Morea, V., Salerno, E., Via, A., and Colombo, T. (2021). AI applications in functional genomics. *Computational and Structural Biotechnology Journal*, 19:5762–5790.

Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2020). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arxiv 2014. *arXiv preprint arXiv:1406.1078*.

Clifford, J. N., Høie, M. H., Deleuran, S., Peters, B., Nielsen, M., and Marcatili, P. (2022). BepiPred-3.0: Improved B-cell epitope prediction using protein language models. *Protein Science*, 31(12):e4497.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., et al. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422.

Cunin, R., Glansdorff, N., Piérard, A., and Stalon, V. (1986). Biosynthesis and metabolism of arginine in bacteria. *Microbiological Reviews*, 50(3):314–352.

Dahl, G. E., Jaitly, N., and Salakhutdinov, R. (2014). Multi-task neural networks for QSAR predictions. *arXiv preprint arXiv:1406.1231*.

Datsenko, K. A. and Wanner, B. L. (2000). One-step inactivation of chromosomal genes in *Escherichia coli* K-12 using PCR products. *Proceedings of the National Academy of Sciences*, 97(12):6640–6645.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. (2022). Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56.

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. J. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., Wiewiora, R. P., Brooks, B. R., and Pande, V. S. (2017). OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):1–17.

Figurnov, M., Mohamed, S., and Mnih, A. (2018). Implicit reparameterization gradients. *Advances in neural information processing systems*, 31.

Fleishman, S. J. and Baker, D. (2012). Role of the biomolecular energy gap in protein design, structure, and evolution. *Cell*, 149(2):262–273.

Frankish, A., Diekhans, M., Jungreis, I., Lagarde, J., Loveland, J., Mudge, J. M., Sisu, C., Wright, J. C., Armstrong, J., Barnes, I., Berry, A., Bignell, A., Boix, C., Carbonell Sala, S., Cunningham, F., Di Domenico, T., Donaldson, S., Fiddes, I., García Girón, C., Gonzalez, J. M., Grego, T., Hardy, M., Hourlier, T., Howe, K. L., Hunt, T., Izuogu, O. G., Johnson, R., Martin, F. J., Martínez, L., Mohanan, S., Muir, P., Navarro, F. C. P., Parker, A., Pei, B., Pozo, F., Riera, F. C., Ruffier, M., Schmitt, B. M., Stapleton, E., Suner, M.-M., Sycheva, I., Uszczynska-Ratajczak, B., Wolf, M. Y., Xu, J., Yang, Y., Yates, A., Zerbino, D., Zhang, Y., Choudhary, J., Gerstein, M., Guigó, R., Hubbard, T. J. P., Kellis, M., Paten, B., Tress, M. L., and Flicek, P. (2020). GENCODE 2021. *Nucleic Acids Research*, 49(D1):D916–D923.

Frazer, J., Notin, P., Dias, M., Gomez, A., Min, J. K., Brock, K., Gal, Y., and Marks, D. S. (2021). Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95.

Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., and Carin, L. (2019). Cyclical annealing schedule: A simple approach to mitigating KL vanishing. *arXiv preprint arXiv:1903.10145*.

Giessel, A., Dousis, A., Ravichandran, K., Smith, K., Sur, S., McFadyen, I., Zheng, W., and Licht, S. (2022). Therapeutic enzyme engineering using a generative neural network. *Scientific Reports*, 12(1):1–17.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Grant, B. J., Rodrigues, A. P. C., ElSawy, K. M., McCammon, J. A., and Caves, L.

S. D. (2006). Bio3d: an R package for the comparative analysis of protein structures. *Bioinformatics*, 22(21):2695–2696.

Grant, B. J., Skjærven, L., and Yao, X.-Q. (2021). The Bio3D packages for structural bioinformatics. *Protein Science*, 30(1):20–30.

Guerois, R., Nielsen, J. E., and Serrano, L. (2002). Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *Journal of molecular biology*, 320(2):369–387.

Hallows, W. C., Skvorak, K., Agard, N., Kruse, N., Zhang, X., Zhu, Y., Botham, R. C., Chng, C., Shukla, C., Lao, J., et al. (2023). Optimizing human α-galactosidase for treatment of Fabry disease. *Scientific Reports*, 13(1):4748.

Hawkins-Hooker, A., Depardieu, F., Baur, S., Couairon, G., Chen, A., and Bikard, D. (2021). Generating functional protein variants with variational autoencoders. *PLOS Computational Biology*, 17(2):1–23.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P., Springer, M., Sander, C., and Marks, D. S. (2017). Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135.

Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. (2022a). Learning protein fitness models from evolutionary and assay-labeled data. *Nature biotechnology*, 40(7):1114–1122.

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A.

(2022b). Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR.

Huang, P.-S., Boyken, S. E., and Baker, D. (2016). The coming of age of de novo protein design. *Nature*, 537(7620):320–327.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. (2019). Generative models for graph-based protein design. *Advances in neural information processing systems*, 32.

Jain, P. and Hirst, J. D. (2010). Automatic structure classification of small proteins using random forest. *BMC Bioinformatics*, 11(1):364.

Janeway, C. A. J., Travers, P., Walport, M., and et al. (2001). *Immunobiology: The Immune System in Health and Disease*. Garland Science, New York, 5th edition.

Jankowiak, M. and Obermeyer, F. (2018). Pathwise derivatives beyond the reparameterization trick. In *International conference on machine learning*, pages 2235–2244. PMLR.

Jensen, K. K., Andreatta, M., Marcatili, P., Buus, S., Greenbaum, J. A., Yan, Z., Sette, A., Peters, B., and Nielsen, M. (2018). Improved methods for predicting peptide binding affinity to MHC class II molecules. *Immunology*, 154(3):394–406.

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. (2021). DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. (2020). Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*.

Joo, W., Lee, W., Park, S., and Moon, I.-C. (2020). Dirichlet variational autoencoder. *Pattern Recognition*, 107:107514.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, 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., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

Katsigianni, E. I. and Petrou, P. (2022). A systematic review of the economic evaluations of Enzyme Replacement Therapy in Lysosomal Storage Diseases. *Cost Effectiveness and Resource Allocation*, 20(1):51.

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. (2019). CTRL: A conditional Transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Kim, J. H., Lee, B. H., Hyang Cho, J., Kang, E., Choi, J.-H., Kim, G.-H., and Yoo, H.-W. (2016). Long-term enzyme replacement therapy for Fabry disease: efficacy and unmet needs in cardiac and renal outcomes. *Journal of Human Genetics*, 61(11):923–929.

King, C., Garza, E. N., Mazor, R., Linehan, J. L., Pastan, I., Pepper, M., and Baker, D. (2014). Removing T-cell epitopes with computational protein design. *Proceedings of the National Academy of Sciences*, 111(23):8577–8582.

Kingma, D. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kiss, G., Çelebi-Ölçüm, N., Moretti, R., Baker, D., and Houk, K. (2013). Computational enzyme design. *Angewandte Chemie International Edition*, 52(22):5700–5725.

Kornreich, R., Desnick, R. J., and Bishop, D. F. (1989). Nucleotide sequence of the human alpha-galactosidase A gene. *Nucleic acids research*, 17(8):3301.

Kuhlman, B. and Bradley, P. (2019). Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, 20(11):681–697.

Leal, A. F., Espejo-Mojica, A. J., Sánchez, O. F., Ramírez, C. M., Reyes, L. H., Cruz, J. C., and Alméciga-Díaz, C. J. (2020). Lysosomal storage diseases: current therapies and future alternatives. *Journal of Molecular Medicine*, 98(7):931–946.

Lenders, M., Pollmann, S., Terlinden, M., and Brand, E. (2022). Pre-existing anti-drug antibodies in Fabry disease show less affinity for pegunigalsidase alfa. *Molecular Therapy - Methods & Clinical Development*, 26:323–330.

Li, Y.-D., Xie, Z.-Y., Du, Y.-L., Zhou, Z., Mao, X.-M., Lv, L.-X., and Li, Y.-Q. (2009). The rapid evolution of signal peptides is mainly caused by relaxed selection on non-synonymous and synonymous sites. *Gene*, 436(1-2):8–11.

Lin, H. H., Zhang, G. L., Tongchusak, S., Reinherz, E. L., and Brusic, V. (2008). Evaluation of MHC-II peptide binding prediction servers: applications for vaccine research. *BMC Bioinformatics*, 9(12):S22.

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., and Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130.

Lukas, J., Scalia, S., Eichler, S., Pockrandt, A.-M., Dehn, N., Cozma, C., Giese, A.-K., and Rolfs, A. (2016). Functional and clinical consequences of novel α-galactosidase A mutations in Fabry disease. *Human mutation*, 37(1):43–51.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos, J. L., Xiong, C., Sun, Z. Z., Socher, R., Fraser, J. S., and Naik, N. (2023). Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106.

Maksymenko, K., Maurer, A., Aghaallaei, N., Barry, C., Borbarán-Bravo, N., Ullrich, T., Dijkstra, T. M., Alvarez, B. H., Müller, P., Lupas, A. N., et al. (2023). The design of functional proteins using tensorized energy calculations. *Cell Reports Methods*, 3(8).

Marco-Marin, C., Ramon-Maiques, S., Tavarez, S., and Rubio, V. (2003). Site-directed mutagenesis of *Escherichia coli* acetylglutamate kinase and aspartokinase III probes the catalytic and substrate-binding mechanisms of these amino acid kinase family enzymes and allows three-dimensional modelling of aspartokinase. *Journal of molecular biology*, 334(3):459–476.

Marques, A. R. and Saftig, P. (2019). Lysosomal storage disorders–challenges, concepts and avenues for therapy: beyond rare diseases. *Journal of cell science*, 132(2).

Marshall, S. A., Lazar, G. A., Chirino, A. J., and Desjarlais, J. R. (2003). Rational design and engineering of therapeutic proteins. *Drug Discovery Today*, 8(5):212–221.

Maynard Smith, J. (1970). Natural selection and the concept of a protein space. *Nature*, 225(5232):563–564.

Mehta, A. and Hughes, D. A. (1993). Fabry disease. *Seattle (WA)*.

Meikle, P. J., Hopwood, J. J., Clague, A. E., and Carey, W. F. (1999). Prevalence of lysosomal storage disorders. *JAMA*, 281(3):249–254.

Mikolov, T., Kombrink, S., Burget, L., Černockỳ, J., and Khudanpur, S. (2011). Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE.

National Institute for Health and Care Excellence (NICE) (2023). Pegunigalsidase alfa

for treating Fabry disease: Technology appraisal guidance. Technology appraisal guidance, National Institute for Health and Care Excellence (NICE).

Nguyen, E., Poli, M., Durrant, M. G., Thomas, A. W., Kang, B., Sullivan, J., Ng, M. Y., Lewis, A., Patel, A., Lou, A., et al. (2024). Sequence modeling and design from molecular to genome scale with Evo. *BioRxiv*, pages 2024–02.

Nilsson, J. B., Kaabinejadian, S., Yari, H., Kester, M. G., van Balen, P., Hildebrand, W. H., and Nielsen, M. (2023). Accurate prediction of HLA class II antigen presentation across all loci using tailored data acquisition and refined machine learning. *Science Advances*, 9(47):eadj6367.

Pachter, L., Alexandersson, M., and Cawley, S. (2001). Applications of generalized pair hidden Markov models to alignment and gene finding problems. In *Proceedings of the fifth annual international conference on Computational biology*, pages 241–248.

Parenti, G., Medina, D. L., and Ballabio, A. (2021). The rapidly evolving view of lysosomal storage diseases. *EMBO molecular medicine*, 13(2):e12836.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR.

Platt, F. M., d'Azzo, A., Davidson, B. L., Neufeld, E. F., and Tifft, C. J. (2018). Lysosomal storage diseases. *Nature Reviews Disease Primers*, 4(1):27.

Putko, B. N., Wen, K., Thompson, R. B., Mullen, J., Shanks, M., Yogasundaram, H., Sergi, C., and Oudit, G. Y. (2015). Anderson-Fabry cardiomyopathy: prevalence, pathophysiology, diagnosis and treatment. *Heart Failure Reviews*, 20(2):179–191.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Remmert, M., Biegert, A., Hauser, A., and Soding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature methods*, 9(2):173–175.

Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Rokaitis, I., Zrimec, J., Poviloniene, S., Laurynenas, A., Viknander, S., Abuajwa, W., et al. (2021). Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. (2018). Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118.

Romero, P. A. and Arnold, F. H. (2009). Exploring protein fitness landscapes by directed evolution. *Nature reviews Molecular cell biology*, 10(12):866–876.

Salvat, R. S., Verma, D., Parker, A. S., Kirsch, J. R., Brooks, S. A., Bailey-Kellogg, C., and Griswold, K. E. (2017). Computationally optimized deimmunization libraries yield highly mutated enzymes with low immunogenicity and enhanced activity. *Proceedings of the National Academy of Sciences*, 114(26):5085–5093.

Sanchez-Trincado, J. L., Gomez-Perosanz, M., and Reche, P. A. (2017). Fundamentals and methods for T-and B-cell epitope prediction. *Journal of immunology research*, 2017(1):2680160.

Scharnetzki, D., Stappers, F., Lenders, M., and Brand, E. (2020). Detailed epitope mapping of neutralizing anti-drug antibodies against recombinant α-galactosidase A in patients with Fabry disease. *Molecular Genetics and Metabolism*, 131(1-2):229–234.

Schiffmann, R., Kopp, J. B., Austin III, H. A., Sabnis, S., Moore, D. F., Weibel, T., Balow, J. E., and Brady, R. O. (2001). Enzyme replacement therapy in Fabry disease: a randomized controlled trial. *JAMA*, 285(21):2743–2749.

Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F., and Serrano, L. (2005). The FoldX web server: an online force field. *Nucleic acids research*, 33(suppl_2):W382–W388.

Sharp, P. M., Cowe, E., Higgins, D. G., Shields, D. C., Wolfe, K. H., and Wright, F. (1988). Codon usage patterns in Escherichia coli, Bacillus subtilis, Saccharomyces cerevisiae, Schizosaccharomyces pombe, Drosophila melanogaster and Homo sapiens; a review of the considerable within-species diversity. *Nucleic acids research*, 16(17):8207–8211.

Shin, J.-E., Riesselman, A. J., Kollasch, A. W., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A. C., and Marks, D. S. (2021). Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):2403.

Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I. S., and Haussler, D. (1996). Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Bioinformatics*, 12(4):327–345.

Skjærven, L., Yao, X.-Q., Scarabelli, G., and Grant, B. J. (2014). Integrating protein structural dynamics and evolutionary analysis with Bio3D. *BMC bioinformatics*, 15(1):1–11.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Steinegger, M. and Söding, J. (2018). Clustering huge protein sequence sets in linear time. *Nature communications*, 9(1):2542.

Valenzuela-Ortega, M. and French, C. (2021). Joint universal modular plasmids (JUMP): a flexible vector platform for synthetic biology. *Synthetic Biology*, 6(1):ysab003.

Valenzuela-Ortega, M. and French, C. E. (2020). Joint universal modular plasmids: a flexible platform for golden gate assembly in any microbial host. *DNA Cloning and Assembly: Methods and Protocols*, pages 255–273.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., et al. (2022). AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444.

Vaswani, A. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Waldek, S., Patel, M. R., Banikazemi, M., Lemay, R., and Lee, P. (2009). Life expectancy and cause of death in males and females with Fabry disease: findings from the Fabry Registry. *Genetics in Medicine*, 11(11):790–796.

Wanner, C., Arad, M., Baron, R., Burlina, A., Elliott, P. M., Feldt-Rasmussen, U., Fomin, V. V., Germain, D. P., Hughes, D. A., Jovanovic, A., et al. (2018). European expert consensus statement on therapeutic goals in Fabry disease. *Molecular genetics and metabolism*, 124(3):189–203.

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., et al. (2023). De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. (2019). Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Wu, Y. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wu, Z., Johnston, K. E., Arnold, F. H., and Yang, K. K. (2021). Protein sequence design with deep generative models. *Current opinion in chemical biology*, 65:18–27.

Xu, S., Lun, Y., Brignol, N., Hamler, R., Schilling, A., Frascella, M., Sullivan, S., Boyd, R. E., Chang, K., Soska, R., et al. (2015). Coformulation of a novel human α-galactosidase a with the pharmacological chaperone AT1001 leads to improved substrate reduction in Fabry mice. *Molecular Therapy*, 23(7):1169–1181.

Zhao, H., Verma, D., Li, W., Choi, Y., Ndong, C., Fiering, S. N., Bailey-Kellogg, C., and Griswold, K. E. (2015). Depletion of T cell epitopes in lysostaphin mitigates anti-drug antibody response and enhances antibacterial efficacy in vivo. *Chemistry & biology*, 22(5):629–639.

Zubler, R. H. (2001). Naive and memory B cells in T-cell-dependent and T-independent responses. In *Springer seminars in immunopathology*, volume 23, pages 405–419. Springer.