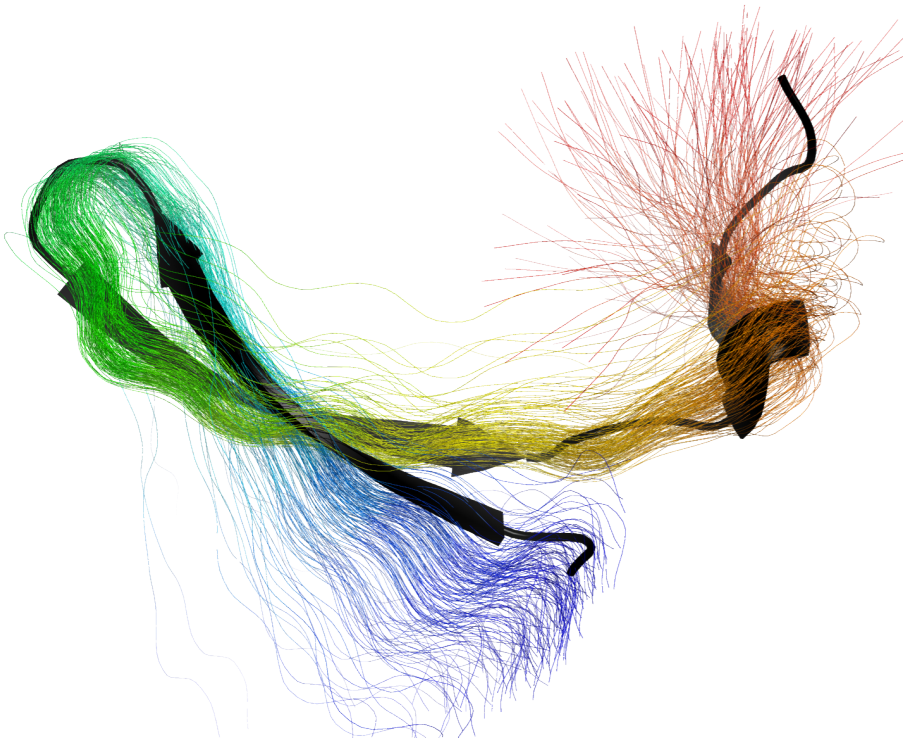
UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE



PhD thesis

Accelerating vaccine development through a deep probabilistic programming approach to protein structure prediction

Christian B. Thygesen

Advisor: Thomas W. Hamelryck

Submitted: September 28, 2022

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen

Abstract

In this thesis, three manuscripts will be presented that focus on utilizing a novel approach to protein structure prediction, subsequently allowing the acceleration of vaccine development.

The first manuscript presents a deep, probabilistic, and generative model of local protein structure. The proposed model represents a means of evaluating the possible conformations that small protein fragments adopt. The model produces fragment libraries at a quality on-par with state-of-the-art models, at a fraction of the run time, without the need for external information and third-party tools to guide the library construction.

In manuscript 2 I use this model of local protein structure to accelerate the vaccine design process. Vaccines typically induce an immune response through the combination of structural B cell epitopes and small linear T cell epitopes. In manuscript 2, I present an approach that uses the local protein structure model to modify the coronavirus spike protein through peptide grafting. We show that the model can adapt the spike protein of SARS-CoV-2, in a manner that preserves the important B-cell epitopes needed to induce an antibody response, while enriching for T cell epitopes that can boost this response. I show that vaccine constructs designed using this model express at a higher level than those designed with a naive approach allowing only small modifications of the spike protein. The vaccine constructs are able to induce an antibody response against the wildtype in immunized mice, indicating proper folding of the modified protein construct.

The model presented in manuscript 1 focuses only on inferring backbone dihedral angles. This focus on internal coordinates limit the ability to model fragments that are larger than 9 amino acids. The third and final manuscript presents a means to alleviate this problem by introducing a novel multi-scale approach employing likelihoods over both internal coordinates as well as reconstructed 3D-coordinates. I show that this change improves the models performance on short fragments while allowing modelling of longer protein fragments as well.

Dansk Resumé

Forudsigelse af hvordan et protein folder er et problem af stor interesse. Proteiner anses for at være livets byggeklodser, da de er centrale for en langt de fleste biologiske processer. Hvis man ved, hvordan et protein folder og fungerer, er man godt på vej til at kunne at manipulere dem for at opnå forskellige mål, såsom at designe en vaccine.

I denne afhandling præsenteres tre artikler med fokus på at udnytte en nyudviklet metode til at forudsige protein strukturer, for at accelerere vaccineudvikling. Den første artikel præsenterer en dyb, probabilistisk og generativ model over lokal protein struktur. Modellen bruges til at evaluere det strukturelle udfaldsrum for mindre fragmenter af proteiner. Modellen er i stand til at generere fragmentbiblioteker hurtigere end eksisterende metoder og af samme kvalitet. Jeg præsenterer hvordan vi kan bruge sådan en model til at modificere proteiner i en større grad end hidtil muligt. Jeg viser, at vi ved hjælp af denne model kan ændre en aminosyresekvens med minimal effekt på den endelige proteinstruktur. Dette gøres med henblik på at designe en vaccine mod SARS-CoV-2, der er potentielt bredere dækkende end eksisterende vaccineprodukter. Til sidst viser jeg, hvordan vi kan forbedre vores lokale proteinstruktur model ved at introducere en ny metode hvorpå man kan træne sådanne modeller. Denne nye metode gør det muligt at træne modellen med sandsynligheder over 3D-koordinater, hvilket tillader, at vi ikke blot kan modellere små fragmenter af proteiner endnu bedre, men at vi samtidig kan bevæge os hen imod større fragmenter.

Preface & Acknowledgements

This thesis is the culmination of three years of exciting collaboration between Assoc. Professor Thomas Hamelryck's probabilistic programming group, Evaxion Biotech, and myself. I carried out the work between October 2019 and October 2022, funded by the Innovation Fund Denmark. The work was supervised by Assoc. Professor Thomas Hamelryck at the University of Copenhagen, Department of Computer Science, as well as Anders Bundgaard Sørensen and Christian Skjødt-Steenmans at Evaxion.

It has been a great opportunity to perform this work under the supervision of talented people from wildly different research areas. Starting out, I knew little about vaccine design and had barely heard of probabilistic programming. Even so, three years later, we managed to write three manuscripts that I am proud of, where one was accepted to a major conference. I would like to thank my three supervisors for being supportive and patient with me through this process.

Anders deserves a special mention as he continued to provide mentorship and advice even after circumstances prevented him from doing so at a daily basis.

Thank you to Christian, Jens, Michael, Gry, and Birgitte at Evaxion for stepping in Anders' shoes and supporting me in the final stage of the PhD. Also to Søren and the rest of Evaxion's lab team, who slaved away to perform experiments for me, that I could not have done myself, even if I had been given a three-year extension.

Finally, I would like to thank my friends and family. Especially my wife, Emilie, and son, Eskild, for their support on the days where nothing worked, and for pretending interest on the days where everything worked.

Contents

Acronyms

APC Antigen Presenting Cell. 23, 24

BCR B Cell Receptor. 23

BIFROST Bayesian Inference for FRagments of protein STructures. 27, 73, 74

Cryo-EM Cryo-Electron Microscopy. 5, 75

CTL Cytotoxic T lymphocyte. 24

DMM Deep Markov Model. 17, 20–22, 27, 73, 74

DPP Deep Probabilistic Programming. 14

ELBO Evidence Lower Bound. 17, 19

FFN Feed Foward Neural Network. 14, 15

GRU Gated Recurrent Unit. 15, 73

HMC Hamiltonian Monte Carlo. 13

HMM Hidden Markov Model. 20, 21

IO-DMM Input-Output Deep Markov Model. 21, 22

IO-HMM Input-Output Hidden Markov Model. 20, 21

KLD Kullback-Leibler Divergence. 19

LSTM Long-Short Term Memory. 15

LVM Latent Variable Model. 17

MHC Major Histocompatibility Complex. 23–25

MLP Multilayer Perceptron. 14

MSA Multiple Sequence Alignment. 9, 11

NMR Nuclear Magnetic Resonance Spectroscopy. 5, 75

NUTS No U-Turn Sampling. 13

PDB Protein Data Bank. 5, 10, 11

PPL Probabilistic Programming Language. 13, 14

RNN Recurrent Neural Network. 14–16

SGD Stochastic Gradient Descent. 16

SVI Stochastic Variational Inference. 13, 17, 19, 27, 73

$\mathbf{T_{FH}}$ Follicular T-helper. 24

$\mathbf{T_{H}}$ T-helper. 24

$\mathbf{T_{reg}}$ Regulatory T. 24

TCR T Cell Receptor. 23

VAE Variational Autoencoder. 17–20, 74

Chapter 1

Introduction

PROTEINS are the building blocks of life. By now, this sentence has become a staple in any protein structure prediction publication. Unsurprisingly, this is because it is true. Proteins are essential components, building blocks, and even motors responsible for wide variety of functions all the way from reproduction to cell death. Protein filaments make up cellular structures, enzymes facilitate chemical reactions and processes, immunoglobulins respond to foreign pathogens, the aptly named transporters move molecules around, and motor proteins allow movement of everything from single cells to entire multicellular organisms.

While proteins make up the biological machinery, they are also the culprits enabling viruses, bacteria, and parasites to infect and spread. Therefore, knowing what these proteins are, and how they function will go a long way towards learning how to protect ourselves from such pathogens.

Another common phrase in structural biology is that *structure is function*. In their simplest form, proteins are long polymers of amino acid residues. As these polymers are produced by a cell, they begin to fold into an endless number of different shapes. This shape dictates how the protein interacts with its surroundings and consequently what functions it can carry out. Thus, learning how a protein folds will tell you a lot about how it functions. Consequently, the problem of protein structure prediction has for many years been of immense interest with recent vast improvements being showed with the application of deep learning [1, 2]. With the advent of AlphaFold [1], the problem has even been stated as solved [3]. However, open questions still remain, relating to protein dynamics as well as the impact of fold-disturbing mutations. These open problems leave some room in the field for research into models with a focus on smaller scales of the protein structure prediction problem.

1.1 Research questions

The purpose of this thesis is to use the multitude of available publicly accessible data on protein structure to accelerate the process of vaccine development. Eventually, we will reach this goal by addressing the following research questions.

> Can we develop a model of local protein structure that properly accounts for the sequence to structure relationship and does not rely on multiple sequence alignments?
>
> How do we efficiently engineer proteins to improve a vaccine product?
>
> Can a model be developed that accounts for multiple levels of uncertainty with regards to the relationship between sequence, backbone dihedral angles and 3D-coordinates?

Chapter 2

Protein Structure Prediction

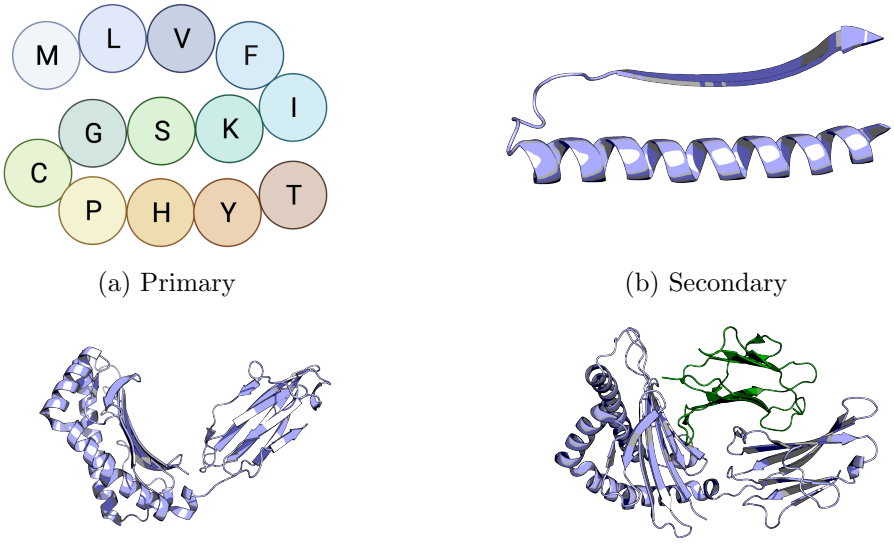
PROTEINS carry out a multitude of functions and therefore are considered the machinery of living organisms. A common, although somewhat simplified, saying on proteins is that structure is function. The 3-dimensional shape of a protein is what endows the protein with the ability to interact with other proteins or small molecules, which activates the protein and triggers a response.

Protein structures are determined experimentally using techniques such as X-ray crystallography [4], Nuclear Magnetic Resonance Spectroscopy (NMR) [5], and Cryo-Electron Microscopy (Cryo-EM) [6]. Techniques like these have been applied to proteins since the 1950's [7], and these experimentally solved structures have been deposited in the Protein Data Bank (PDB) [8]. However, even with modern techniques for experimentally solving protein structures, the throughput is low, and solving even a single protein structure is a long and elaborate process. If information on protein structure is to be used in a high throughput setting, such as modern vaccine development, structures can simply not be solved fast enough. This presents an evident need for computational methods that use a minimal amount of time and resources for determining protein structure.

Enter computational protein structure prediction. The problem can be stated quite simply:

> *Determine the 3-dimensional structure of a protein from its amino acid sequence.*

Even though the problem is clearly and simply stated, it is not a problem that is straight-forward to solve. Historically, the problem has been approached from two general angles; as an energy minimization problem, or by end-to-end deep learning prediction. In 2021, the problem was declared solved with the emergence of Deepmind's AlphaFold2 [1]. This huge achievement changed the protein structure prediction landscape and shifted focus almost entirely towards deep learning approaches.



(a) Primary

(b) Secondary

(c) Tertiary

(d) Quaternary

Figure 2.1: The four different levels of protein structure

The main driver for the development of BIFROST, presented in manuscript 1 (section 5) and subsequently in manuscript 3 (section 7), was to address part of the pipeline for performing energy-based modelling of protein structure. Thus, in addition to a brief overview of the basics of protein structure, this section will focus on introducing the energy-based modelling approach.

2.1 Protein structures

In their simplest form, proteins are linear polymers of amino acid residues, which form the primary structure, or amino acid sequence (figure 2.1a). There are 20 different standard amino acids with a shared backbone structure, but with various physiochemical properties. Small stretches of the amino acid sequence form local substructures, known as secondary structures, which generalize as either α-helices, β-strands, or coils (also called loops) 2.1b. The final fold of a single protein chain is called the tertiary structure 2.1c. In many cases, the functional form of a protein is a complex of multiple identical or different protein chains, which is called the quaternary structure 2.1d. Predicting the structure of protein structure complexes is beyond the scope of this thesis.

2.1.1 Representations of protein structure

Protein structures can be computationally represented in multiple ways. *Cartesian coordinates* are commonly used, as they can be interpreted through visu-

alisation software. Here, the protein chain is represented as a point cloud in a 3-dimensional Cartesian coordinate system, which allows easy visualisation and analysis. Cartesian coordinates maintain the inter-residue distances and thus retain information on relative positioning of each atom. However, 3D-coordinates can be problematic to work with in a machine learning setting, as they are rotationally and translationally variant. The same coordinate sets can be moved arbitrarily along any axis (translation) or arbitrarily rotated without actually changing the structure. Thus, predicting the true set of coordinates or computing the true gradient from a predicted coordinate set is a challenge.

Alternatively, protein structures can be represented by *internal coordinates*. Amino acids share the same basic chemical structure forming a backbone of covalently bonded nitrogen and two carbon atoms, C_α and C. Internal coordinates are the torsion angles around these bonds between the atoms of the protein backbone as well as the lengths of these bonds. The rotations are represented by dihedral angles, ϕ, ψ, and ω (figure 2.2).

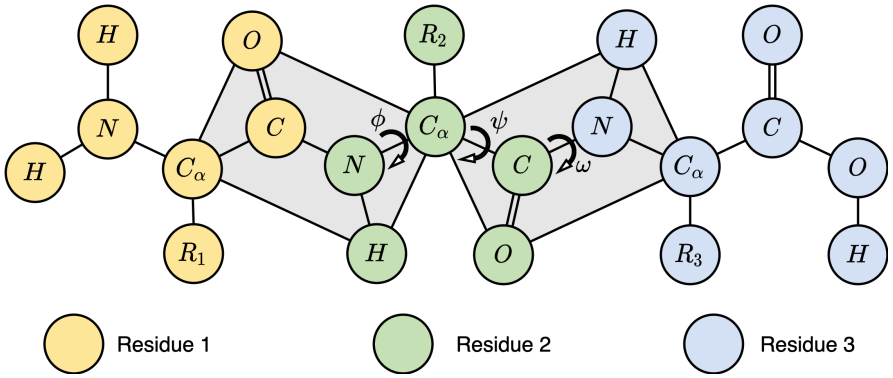
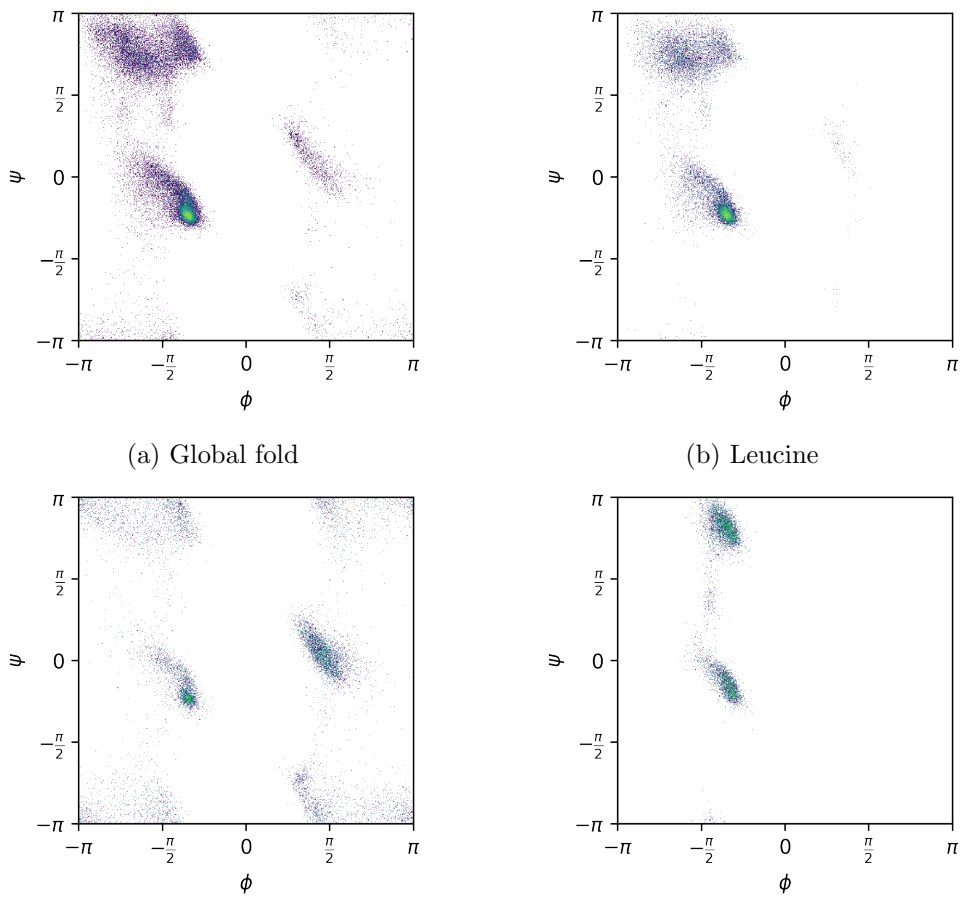


Figure 2.2: Schematic of the protein backbone with ϕ, ψ, and ω dihedral angles. R_i denotes the side chain, which varies between amino acids.

In practice, ϕ and ψ are the dihedral angles of interest, as ω is restricted by the double-bond between a carbon atom, C, and an oxygen-atom, O. This means that ω is restricted to appear in either a cis- ($\sim 0°$) or a trans-orientation ($\sim 180°$), where the cis-orientation is only rarely observed [9]. The distribution of ϕ and ψ angles can be used to represent a protein fold in a Ramachandran-plot [10]. The Ramachandran plot shows that dihedral angles form clusters corresponding to the secondary structure (figure 2.3). As amino acids share the same basic structure, most show the same behavior in Ramachandran space (figure 2.3b). The two exceptions are glycine (figure 2.3c proline (figure 2.3d). As the side chain of glycine consists of a single hydrogen atom, the backbone is flexible, where as the side chain of proline binds the backbone both at the C_α and at the nitrogen atom N, causing the backbone to become more rigid.



(a) Global fold

(b) Leucine

(c) Glycine

(d) Proline

Figure 2.3: The distribution of global ϕ and ψ dihedral angles (a) as well as for individual amino acids shown as Ramachandran plots. Leucine (b) is included to represent the behavior of most amino acids, while glycine (c) and proline (d) are included to show their distinct behavior.

Internal coordinates are a solid alternative to Cartesian coordinates for modelling purposes, as they are rotationally and translationally invariant. Regardless of how the protein is rotated or moved, the dihedral angles will remain unchanged as they represent the relative rotations of the backbone atoms. However, internal coordinates give no insight on the relative positioning of atoms. Dihedral angles are easily calculated from Cartesian coordinates by evaluating the rotation around the bonds between the atoms in the backbone based on the planes formed by three consecutive amino acid residues. However, going the other way is not as straight forward and requires an algorithm that iteratively reconstructs 3D-coordinates one amino acid at a time given the dihedral angles [11, 12]. As a result, the reconstruction of Cartesian

coordinates suffers from an elbow effect, where modest errors in dihedral angle space early in the chain will result in large variations of the reconstructed coordinates downstream.

The third and final approach is to encode the protein structure as a set of pairwise contacts or distances, typically between C_α atoms [13]. These distances are inferred by aligning a multitude of homologous sequences, thus creating a Multiple Sequence Alignment (MSA). From the MSA we can perform co-evolution analyses to evaluate the likelihood that the amino acids at two positions mutate in tandem. We expect that if two residues are in contact in the protein structure, then mutating one of them would negatively affect the structure. Therefore, co-evolving residues are likely to be in close proximity. Many approaches exist for solving this problem including direct coupling analysis [14], inverse covariance [15], pseudolikelihood approaches [16], and of course neural networks [17].

Contact and distance maps are an attractive representation of protein structure, as they are symmetric $L \times L \in \mathbb{R}$ matrices, where L is the number of amino acids in the protein. They are rotationally and translationally invariant, except they do not allow for distinguishing between a structure and it's reflection.

2.2 Energy-based modelling

Energy-based modelling defines the problem of protein structure prediction as an energy minimization problem, based on the assumption that the native fold of a protein will be the fold that minimizes some physical energy function (figure 2.4).

As described in section 2.1, protein structures are shaped by the dihedral angles in the protein backbone. Levinthal's paradox [19] states that even if we assume a simplified problem where each ϕ and ψ dihedral angle can only take on three possible conformations and ω is ignored, the number of possible folds for a hypothetical protein of 100 amino acids would be on the order of 3^{198}. Such an outcome space is too vast to naively sample our way through. Energy-based modelling frameworks, such as Rosetta [20, 21] address this challenge through Monte Carlo sampling guided by a physical energy function. At each iteration, a possible backbone conformation is sampled and the sampled move is then evaluated by the energy function. If the move improves the energy, it is accepted, if not, the move is discarded. The energy function evaluates a range of different parameters, such as the number of salt-bridges, hydrogen bonds, and how well the dihedral angles fit the empirical Ramachandran plot [22]. This Monte Carlo sampling scheme allows a more efficient traversal of the energy landscape.

However, even with an energy function to guide sampling, the number of possible moves is still vast. For this reason, Rosetta samples conformations

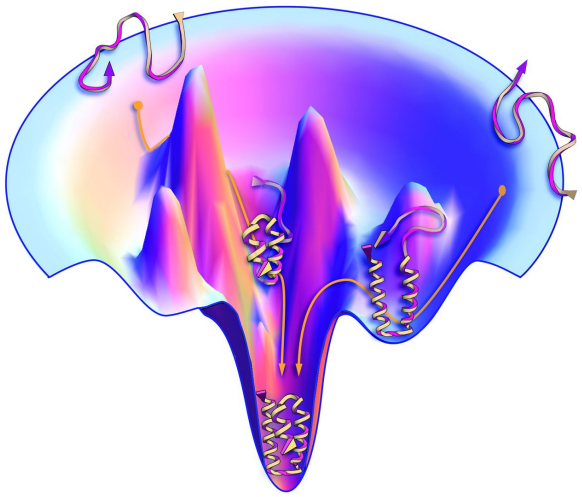


Figure 2.4: A conceptual representation of the energy landscape forming the basic assumption for energy-based modelling. The native fold is observed at the bottom of the funnel formed by the energy landscape. Figure borrowed from [18].

for several amino acids at a time from a library of observed outcomes, known as a fragment library.

2.2.1 Fragment libraries

A fragment library typically consists of 200 different conformations for each overlapping window of 3 and 9 amino acids in the sequence. Fragment libraries are built by splitting the amino acid sequence into overlapping windows of 3 and 9 amino acids (3-mers and 9-mers). The sequence fragments are then searched against a public database of protein structures, such as teh PDB, to find 200 possible structural outcomes for each fragment (figure 2.5).

Fragment libraries allow Rosetta to sample realistic backbone conformations 3 and 9 amino acids at a time, dramatically reducing the space of outcomes. The most widely used approach for generating fragment libraries is the fragment picker [23], which ships the Rosetta software.

Unlike the approach presented manuscript 1, the fragment picker is not a trained model. Rather, it is an algorithm for selecting the best combination of already observed structures that a short amino acid sequence could conform to. The algorithm searches each overlapping fragment against structures in the PDB by matching sequence similarity along with a sequence profile score, as well as similarity to predicted secondary structure, as predicted by a pool of secondary structure prediction tools; e.g. psipred [24], JUFO [25], SAM [26], and NetsurfP [27]. Additionally, fragments are scored by a Ramachandran probability, i.e. how well the backbone dihedral angles fit with the distribution

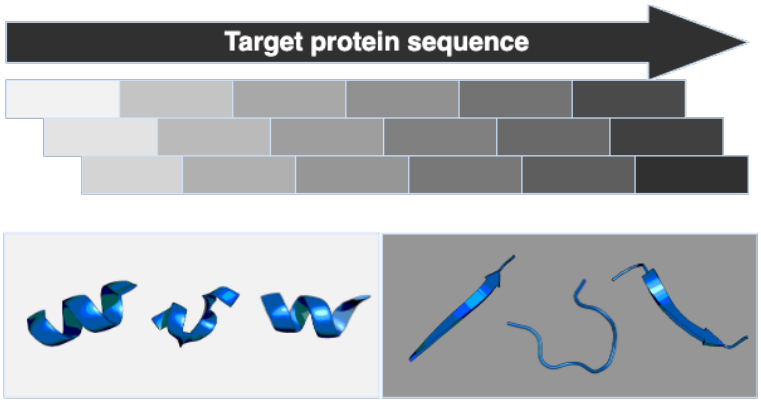


Figure 2.5: Fragment libraries represent each overlapping fragment of a protein sequence with a number possible backbone conformations, typically sampled from the PDB.

of the Ramachandran plot.

Thus, the fragment picker runs a series of third-party tools in order to construct an optimal collection of backbone conformations for each fragment. This is a computationally heavy approach, which does not lend itself to a high-throughput setting.

2.3 Progress in the protein structure prediction landscape

Protein structure prediction is a task that has historically divided the field of structural bioinformatics. Several types of approaches have emerged towards solving the protein folding problem. At one end of the spectrum are the energy-based modelling approaches for simulating the process of folding through sampling, as described in section 2.2. These approaches either simulate the folding process from scratch (*de novo* folding) or starting from known protein structures as templates (homology modelling) [20, 21, 28, 29, 30]. At the heart of these approaches are efficient sampling algorithms, physical energy functions, and template searches. At the opposite end of the spectrum we have the end-to-end prediction approaches [31]. Crudely, end-to-end approaches are able to go from an extended amino acid polymer to a fully folded protein in a single step. Typically, these approaches apply sophisticated neural networks along with clever feature engineering to obtain as much information as possible before making predictions. As it turned out this tribe would be the one supposedly solving the problem with the advent of AlphaFold [1] through the use of MSAs and rotation equivariant transformer neural networks. Multiple

sequence alignments are crucially important in this process, as they provide information on residue co-evolution. This is based on the assumption that residues that evolve and mutate in tandem are more likely to be in close proximity in the protein structure. From this we can predict inter-residue contacts, which greatly limiting the conformational space [16, 32]. Since the publication of AlphaFold, the multiple end-to-end tools have been published publications showing a level of performance, that was unfathomable only a couple of years ago [2, 33, 34, 35].

Chapter 3

Deep Learning and Probabilistic Programming

ORIGINALLY, probabilistic modelling was an art reserved only for those truly gifted in math, probability theory, and coding. If one wanted to address a problem with probabilistic modelling, one would have to manually write programs that (i) define the probabilistic model, (ii) implement home-brewed inference algorithms to deal with the intricacies associated with each specific model, and (iii) write sophisticated sampling algorithms to actually use a model for forecasting and predictions. On top of that comes the computational challenge of fitting a sophisticated model to an arbitrary number of data points.

With the advent of Probabilistic Programming Languages (PPLs) this process was made available to a wider audience. They describe standardised frameworks and libraries for writing probabilistic models in a principled manner, that allows the use of arbitrary inference algorithms, from sampling-based approaches such as Hamiltonian Monte Carlo (HMC) [36, 37] and No U-Turn Sampling (NUTS) [38] to optimization-based approaches such as Stochastic Variational Inference (SVI) [39]. PPLs are developed to extend multiple existing languages such as Python, R or C. Some popular PPLs are Stan [40], PyMC3 [41], Tensorflow Probability [42], Pyro [43], and numpyro [44].

Pyro versus numpyro The models presented in this thesis have been developed using the probabilistic programming frameworks Pyro [43] and NumPyro [44]. Both frameworks are Python libraries providing principled approaches for defining and inferring arbitrary probabilistic models. Both frameworks integrate seamlessly with deep learning frameworks, thus allowing the combination of probabilistic graphical models and neural networks. Pyro utilises the deep learning framework Pytorch [45] for defining neural networks and calculating gradients, whereas numpyro builds on numpy [46] and JAX [47] for calculating gradients, which allows for the use of any JAX-based neural

network library, such as haiku [48].

3.1 Deep probabilistic programming

As mentioned above, PPLs allows the formulation of probabilistic models in the form of code and programs, and provides a unified framework for model inference and sampling. Deep Probabilistic Programming (DPP) languages combine the elegance of probabilistic models with the raw power of artificial neural networks in the same framework, allowing modelling of arbitrarily complex models.

A major obstacle in probabilistic modelling, is the amount of parameters needed to model a phenomenon. In traditional probabilistic modelling, the number of parameters typically scales with the number of data points, which presents a challenge when working with larger data sets. However, using neural networks, we can share parameters between data points (a process called amortization), which means that the number of parameters does not increase with the amount of data. Essentially, DPP frameworks allow describing the relationship between random variables in a model to be approximated by a neural network, thus allowing modelling of arbitrary nonlinear relationships, while allowing amortization of parameters.

3.2 Neural networks and deep learning

Machine learning and especially neural networks [49, 50] are used extensively because of their abstractive power to model complex phenomena. In this thesis they are used to drive the inference of probability distributions in a semi-Bayesian setting.

3.2.1 Neural networks

As mentioned, the *deep* part of Deep Probabilistic Programming (DPP) comes from the use of artificial neural networks to share parameters between data points. In the models implemented here, two general types of neural network architectures were implemented; the standard Feed Foward Neural Network (FFN) along with a subtype of FFN called Recurrent Neural Networks (RNNs).

Feed forward neural networks Fully connected FFNs are the basic class of neural networks, which can be stacked to form a Multilayer Perceptron (MLP) [51] also known as a deep neural network [52]. An FFN layer is composed of a set of *neurons*, each with an associated set of weights, w, and bias b. Stacking these layers results in a deep neural network (figure 3.1).

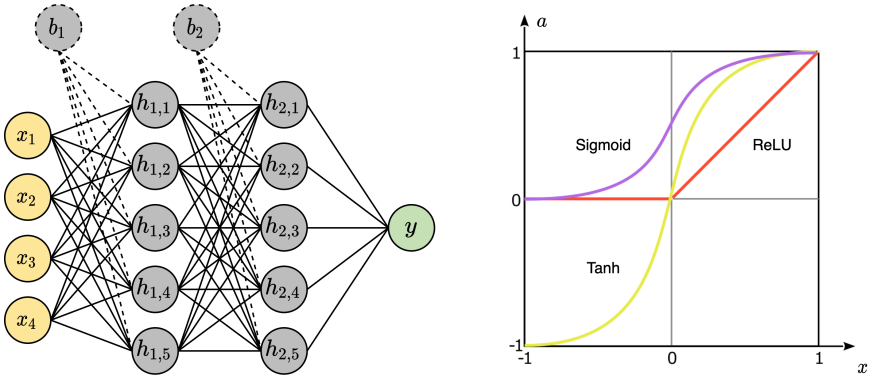


Figure 3.1: A deep neural network (left) is a stack of linear transformations followed by non-linear activation functions (right). Each node represents a neuron and each edge represents a weight.

A single FFN layer can be viewed as a linear transformation of an input vector consisting of n_i values to an output vector of n_o values. Each value in the input vector is connected to each value in the output vector with a tunable weight. Each value in the output vector is then calculated as a weighted sum of the input values followed by a non-linear activation function 3.1.

$$f(\mathbf{x}, \mathbf{W}, \mathbf{b}) = a(\mathbf{x}^T\mathbf{W} + \mathbf{b}) \tag{3.1}$$

where $\mathbf{x}$ is a vector with n_i values, and $\mathbf{W}$ is an $n_i \times n_o$ matrix.

Recurrent neural networks - Sequence models RNNs are a class of feed forward neural networks intended for modelling data with a sequential relationship [53]. An RNN typically performs an iterative parsing of each time step in a series of observations, such as a sequence of words or, relevant for our case, amino acids. Here, we describe the three most often used types of RNNs, the standard RNN, the Gated Recurrent Unit (GRU) [54], and the Long-Short Term Memory (LSTM) [55] networks. Each of the networks are composed of one or more feed forward layers, called a cell. A cell produces a hidden state, h_t, given a previous hidden state, h_{t-1}, and an observation at time t, x_t. The standard RNN and the GRU cells only produce one hidden state (equation 3.2), whereas the LSTM cell produces a memory state, c_t, along with the hidden state (equation 3.3).

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t) \tag{3.2}$$

$$\mathbf{h}_t, \mathbf{c}_t = f(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \mathbf{x}_t) \tag{3.3}$$

The standard RNN cell is the simplest cell, consisting of a single feed forward layer, followed by a tanh activation, whereas the GRU and LSTM

implement logic gates in the form of sigmoid and tanh-activated feed forward layers, determining which of the neurons in the hidden and memory states should be updated (figure 3.2).

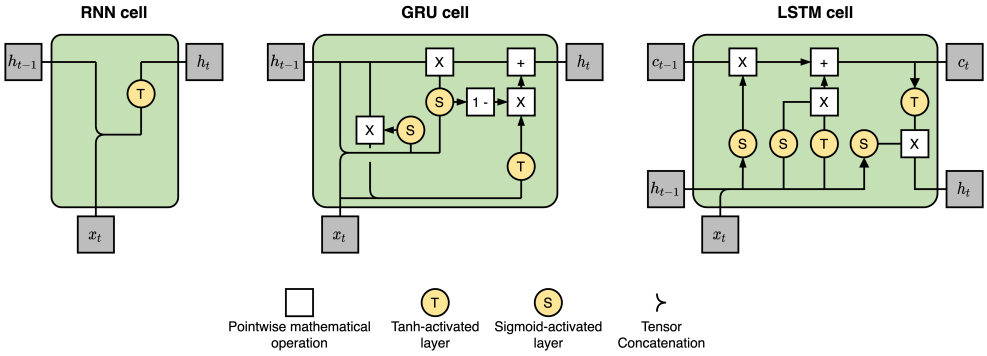


Figure 3.2: Three different types of RNN cell architectures.

3.2.2 Stochastic gradient descent

Neural networks are typically trained on a large data set with a procedure called Stochastic Gradient Descent (SGD) [56]. SGD is composed of three steps: forward propagation, backpropagation and weight update. Consider the neural network predicting the target variable y from an input variable $\mathbf{x}$. In forward propagation, data points are passed through the neural network to perform a prediction, $\hat{y}$ (equation 3.4).

$$\hat{y} = f(\mathbf{x}, \mathbf{W}) \tag{3.4}$$

where $\mathbf{W}$ are the parameters (weights) of the neural network. The predicted target variable, $\hat{y}$, is then compared to the observed output variable with a differentiable loss function, $\boldsymbol{l}$, to obtain an estimate of the predictive error, $\boldsymbol{E}$ (equation 3.5).

$$\boldsymbol{E} = \boldsymbol{l}(\hat{y}, y) \tag{3.5}$$

The loss function is chosen such that if $\hat{y}$ is close to y, the error is small and vice versa. In backpropagation, the derivatives of each weight in the network with regard to the error is calculated to obtain gradients for each weight. In the weight adjustment step, each weight is updated by taking a step in the direction of the gradient calculated in the backpropagation step (equation 3.6.

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \nu \nabla E(\mathbf{w}^{\tau}) \tag{3.6}$$

where τ is the iteration step and ν is size of the step taken in the direction of the gradient, also known as the learning rate.

Repeating these three steps in an iterative manner by passing random samples from the data set will then lead to a minimization of the error function, pushing the neural network toward performing better predictions.

3.3 Generative modelling

The objective of a generative model is to create or synthesise data points with or without conditioning on a certain input. One such class of models are Latent Variable Models (LVMs) [57]. LVMs assume that observed data are noisy observations of some state, which we can not observe, called the latent state. The objective of these models is to generate plausible observations, $\mathbf{x}$, given a latent state, $\mathbf{z}$. Such a model will have the joint density

$$p_\theta(\mathbf{z}, \mathbf{x}) = p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z}) \tag{3.7}$$

where θ are the parameters of the distribution p. The problem to be solved in such a generative model is to infer the posterior probability density

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})} \tag{3.8}$$

Fitting a model like this with continuous latent variables is challenging, as it requires integrating out the latent variables to obtain the marginal likelihood (or evidence) $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})d\mathbf{z}$ which is intractable for continuous latent variables. We will be address this challenge by applying SVI [39] to train the probabilistic, generative models. The problem will be treated as a minimization problem, with the aim of optimizing the Evidence Lower Bound (ELBO) [58]. SVI, ELBO and a simple trick needed to make them work, will be described in more detail below.

First, we will set the stage by introducing the Variational Autoencoder (VAE) [58] to motivate the type of problem to be solved, followed by an introduction of the Deep Markov Model (DMM) [59].

3.3.1 Variational Autoencoders

The most common type of LVM is the VAE [58]. VAEs are composed of an encoder and a decoder step. In the encoding step, a latent state, $\mathbf{z}$, is inferred from observations, $\mathbf{x}$, while in the decoding step an observation is reconstructed from a latent state. These encoding and decoding steps are typically carried out by neural networks (figure 3.3). We let the encoder, called the variational distribution or simply the guide, parameterise a multivariate Gaussian distribution conditioned on an observation $\mathbf{x}$, from which we sample a latent variable $\mathbf{z}$. This latent variable is then passed to the decoder, which we call the model, in order to reconstruct the data point $\hat{\mathbf{x}}$ (figure 3.3). Thus, the model parameterises the distribution $p_\theta(\mathbf{x}|\mathbf{z})$, while the guide parameterises a

distribution approximating the true posterior $q_\phi(\mathbf{z}|\mathbf{x})$, where θ and ϕ are the parameters of the distributions.

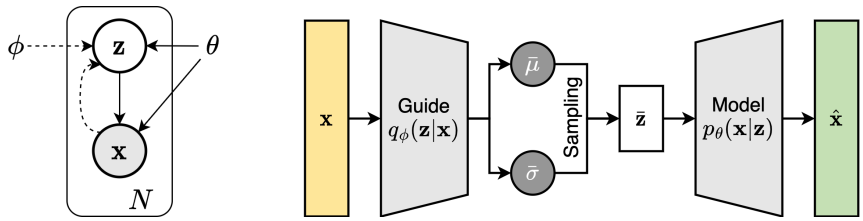


Figure 3.3: The VAE as a directed graphical model (left) and rolled out flowchart (right). The encoder, or guide, parameterises a distribution over the latent state conditioned on an observation ($\mathbf{x}$). A latent variable ($\mathbf{z}$) is sampled from this distribution, and the decoder, or model, reconstructs the observation ($\hat{\mathbf{x}}$) from this latent variable.

A model such as the one depicted in figure 3.3, has a joint probability density which factorises as

$$p(\mathbf{x}, \mathbf{z}) = \prod_{i=1}^{N} p_\theta(\mathbf{x}_i|\mathbf{z}_i)p(\mathbf{z}) \tag{3.9}$$

where N is the number of data points we have available and θ are the model parameters. The objective is to maximise the posterior likelihood of our latent variables given the data

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}, \mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})} \tag{3.10}$$

The denominator of equation 3.10 is the marginal likelihood which can be obtained by marginalizing out the latent variables $\mathbf{z}$:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z})d\mathbf{z} \tag{3.11}$$

such that our posterior can be calculated

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}, \mathbf{z})}{\int p_\theta(\mathbf{x}, \mathbf{z})d\mathbf{z}} \tag{3.12}$$

However, the integral of equation 3.11 is intractable for continuous latent variables, so in order to actually train a model such as this in an optimisation setting, we need to introduce a variational distribution (guide) $q_\phi(\mathbf{z})$, that we can sample from to approximate this density. In a VAE, the guide is given, as it simply boils down to our encoding step.

3.3.2 Stochastic Variational Inference

Suppose one wants to train a VAE on a set of observations $\mathbf{x}$. We assume that the observations $\mathbf{x}$ are generated from an underlying set of unobserved continuous random variables $\mathbf{z}$, which we call latent random variables. A simple graphical model depicting this process is shown in figure 3.3.

In order to fit this generative model, we are interested in obtaining a probability distribution over $\mathbf{x}$ and we want to estimate a model that maximises this probability, which we obtain by marginalizing out the latent variables of the joint probability over $\mathbf{x}$ and $\mathbf{z}$ (equation 3.12).

As mentioned, in order to optimise the parameters of our model to maximise the posterior likelihood $p_\theta(\mathbf{z}|\mathbf{x})$, we need to marginalise out the latents, which is an intractable integral. To counter this, we introduce a variational distribution, that we can sample from, and try to have it approximate the true posterior distribution.

Introducing a variational distribution allows us to define a differentiable loss function - the ELBO. SVI employs stochastic gradient descent to minimise the ELBO loss function.

Evidence Lower Bound - ELBO The ELBO combines two terms; the reconstruction loss, $p_\theta(\mathbf{x}|\mathbf{z})$, and a term for the variational approximation of the latent distribution $\frac{p_\theta(\mathbf{z})}{q_\phi(\mathbf{z})}$. The last term is the Kullback-Leibler Divergence (KLD) between the variational approximation and the prior, i.e. $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))$. In VAEs and similar generative models, the KLD functions as a regulariser, encouraging fitting a valid latent distribution to our prior. The ELBO is interpreted as the lower bound on the evidence, which means that maximizing the ELBO leads to minimizing the KLD between the model and the guide.

Thus, the loss term we will use is given by:

$$\mathrm{ELBO} = -\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log(p_\theta(\mathbf{x}|\mathbf{z})] \tag{3.13}$$

The reparameterisation trick One problem remains with applying SVI: gradients. The added stochastic step of sampling latent variables, makes gradient computation impossible. Thus, in order to circumvent this limitation we apply the reparameterisation trick as proposed in [58]. Simply stated we replace our dependency on the sampled latent variable $\mathbf{z}$ by introducing a deterministic function $g(\epsilon, \mathbf{x})$ such that

$$\mathbf{z} = g_\phi(\epsilon, \mathbf{x}) \tag{3.14}$$

where ϵ is a random variable with independent marginal likelihood $p(\epsilon)$.

For a standard Gaussian distribution the reparameterisation is a simple location-scale transformation

$$\mathbf{z} = \mathbf{z}_\mu + \mathbf{z}_\sigma \epsilon \tag{3.15}$$

where $\mathbf{z}_\mu$ and $\mathbf{z}_\sigma$ are the mean and scale of the distribution over $\mathbf{z}$.

Introducing this rather simple trick means that we have moved the stochastic element to ϵ, which means that we can compute gradients with regard to ϕ.

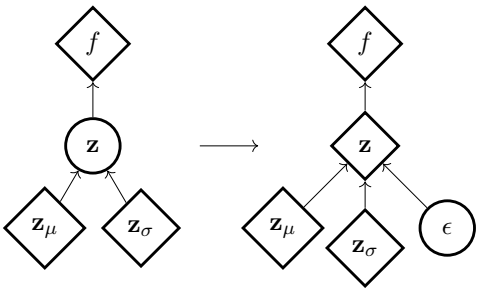


Figure 3.4: The reparameterisation trick allows us to move the stochasticity away from the latent variable. Sampled values are represented by circular nodes, whereas deterministic variables are represented by diamond-shaped nodes.

Thus, combining the introduction of a variational distribution approximating the true posterior with the reparameterisation trick makes it possible to utilize stochastic gradient descent to train a generative latent variable model.

3.3.3 Deep Markov Models

Originally named a hierarchical state space latent variable model [59], the DMM can be seen as a VAE structured like a Hidden Markov Model (HMM). Thus, we can train this model using the same principles as for a VAE.

An HMM is a statistical model for discrete time-series data [60]. It is based on the Markov property, which entails that the latent state ($\mathbf{z}$) of a time-series system depends only on its previous state (equation 3.16.

$$p(\mathbf{z}_t|\mathbf{z}_1, ..., \mathbf{z}_{t-1}) = p(\mathbf{z}_t|\mathbf{z}_{t-1}) \tag{3.16}$$

The HMM thus forms a Markov chain that consists of an initial probability distribution, a number of unobserved latent states along with transition and emission probabilities. The transition probabilities contains the probabilities of the system transitioning from one latent state to another, while the emission probabilities describe the likelihoods of an observation given a latent state. Figure 3.5 shows an HMM for a discrete system with two possible latent states ($k \in [1, 2]$) and two possible observed states (square and circle). The transition between latent states are governed by the transition probabilities represented by solid arrows, while the emission probabilities are represented by dashed arrows. A rolled-out graphical representation of the model is shown to the right in figure 3.5)

HMMs are useful for computing the likelihoods of time-series data, such as speech or text. However, the HMM structure does not necessarily lend itself to supervised learning tasks, where the objective is to correlate a series of observed data points to a series of predictions. A simple extension to the HMM allows us to do exactly this - the Input-Output Hidden Markov Model

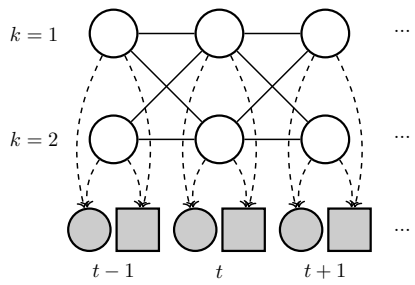


Figure 3.5: A two-state HMM as a graphical model. Unobserved latent states are represented by white nodes, while observed states are represented by grey nodes. Solid arrows represent transition probabilities, dashed arrows represent emission probabilities.

(IO-HMM) [61]. An IO-HMM models the relationship between a series of observations and their impact on the latent states as well as target variables. Given such a sequence of observations $(o_1, ..., o_n)$ and a sequence of target variables $(x_1, ..., x_n)$ we can build a graphical model of an IO-HMM (figure 3.6).

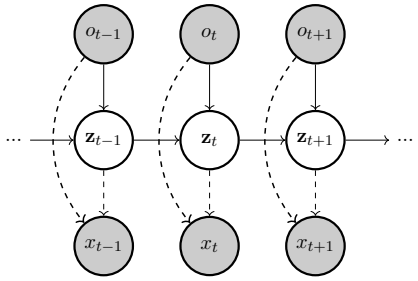
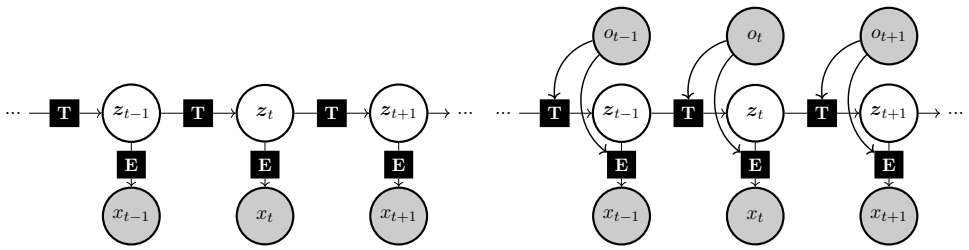


Figure 3.6: A simple graphical representation of an IO-HMM modelling the impact of a series of observed variables on the latent state as well as output variables. Solid arrows denote transition probabilities, dashed arrows represent emission probabilities.

In practical terms, the DMM is a rather straight-forward extension of the HMM. In place of the static transition and emission probabilities of the HMM are neural networks parameterising these distributions, which makes the DMM easily extendable to continuous time-series data (figure 3.7a). For each time step in a sequence, a latent state at time t is conditioned on the latent variable at time $t - 1$ by passing the previous latent state, $\mathbf{z}_{t-1}$ to a transition neural network ($\mathbf{T}$), which parameterises a multivariate gaussian over the latent variable $\mathbf{z}_t$. $\mathbf{z}_t$ is then passed to an emitter neural network ($\mathbf{E}$) which parameterises a distribution from which we can sample a reconstructed observation $\hat{x}_t$.

Similar to the HMM, the DMM is simple to extend to model the impact of a series of observed input variables on the latent state and the predicted output variables. We can call this the Input-Output Deep Markov Model (IO-DMM). In practice, this can be done by conditioning the transition and emission neural networks on the input variables (figure 3.7b).



(a) DMM (b) IO-DMM.

Figure 3.7: The latent state distribution at time i is parameterised by a transition neural network **T** conditioned on the previous latent state $\mathbf{z}_{t-1}$. An observation x_t is generated by passing the latent state at time t to an emitter neural network **E**.

Chapter 4

Vaccine Design

In manuscript 2 we use the tool developed in manuscript 1 to aid the development of a broadly covering SARS-CoV-2 vaccine through a form of protein design. Thus, in this section we will motivate this problem through a brief summary of the main mechanisms of the immune system related to an antiviral response and an efficient vaccine, followed by an introduction to protein design as it relates to manuscript 2.

4.1 Immunology of vaccination

The human immune system is responsible for protecting us against foreign pathogens and malfunctioning cells. It is typically divided into the innate and the adaptive response. The innate response is a broad, naturally occurring immune response capable of immediately protecting the body from a range of pathogens. The adaptive immune response, on the other hand, is a response that develops slower but can provide specific immunity against threats that escape the innate immunity [62]. The adaptive response is composed of a humoral response, relying on B-cells, as well as a cellular response, relying on T-cells [63]. B cells express B Cell Receptors (BCRs), commonly referred to as antibodies, that recognize extracellular foreign proteins, either neutralizing their ability to interact with host cells or flagging them for destruction by other components of the immune system [62, 64]. T cells, on the other hand are divided into two groups based on the receptor they express on the cell surface, namely CD4 or CD8.

Upon viral infection, Antigen Presenting Cells (APCs) of the innate immune response, such as dendritic cells, opsonize virus particles and degrade them. Peptides derived from viral proteins are then presented on the APC's surface bound to proteins called Major Histocompatibility Complex (MHC) class I and II molecules, which are recognized by T Cell Receptors (TCRs) [65]. Peptides presented on MHC class I molecules are recognized by $CD8^+$ T cells, whereas peptides presented on MHC class II molecules are recognized

by CD4$^+$ T cells. This interaction leads to the proliferation of effector T cells. CD8$^+$ T cells become Cytotoxic T lymphocytes (CTLs), which directly kill an infected cell, while CD4$^+$ T cells differentiate to either Regulatory T (T_{reg}) cells, T-helper (T_H) cells, or Follicular T-helper (T_{FH}) cells. T_{reg} cells regulate the immune response, typically to suppress the response against normal host cells, whereas T_H and T_{FH} provide help to the CD8$^+$ T cells and B cells, respectively [62]. T_H cells aid the proliferation of CTLs by interacting with APCs which leads to the release of cytokines stimulating the CTL response (figure 4.1). Eventually, this leads to the generation of memory T cells [66]. T_{FH} cells recognize peptides presented by B cells on MHC class II molecules. This causes the T cells to secrete cytokines, which induces the further proliferation of B cells producing antibodies that recognize the virus. This eventually leads to the establishment of memory B cells, rendering the host immune to subsequent infections (figure 4.1) [67].

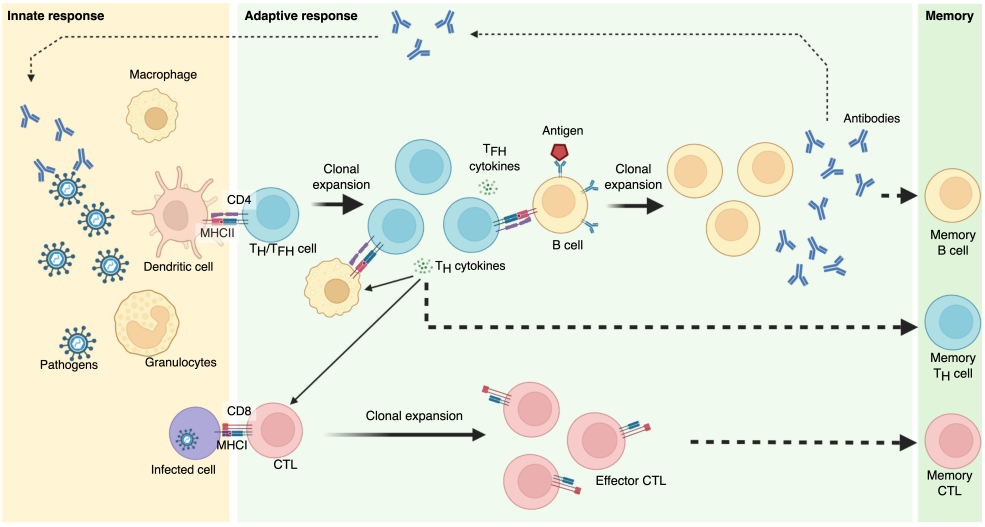


Figure 4.1: The innate and adaptive immune response and how they lead to immunological memory. Pathogens are recognized by the innate immune system. Peptides are presented to T cells by infected cells and APCs, resulting in a stimulation of CTLs as well as T_H and T_{FH} cells. T_H cells provide help to CTLs through the release of cytokines. T_{FH} cells provide B cell help leading to clonal expansion of B cells. Antibodies produced by B cells, recognize and neutralize the pathogen. Eventually, immunological memory is established through the generation of memory B, T_H and CTL cells.

An efficient vaccine is able to elicit the production of both long-lived memory B-cells that produce relevant neutralizing antibodies, as well as memory T-cells, that can recognize and kill infected cells (figure 4.1) [66]. It is therefore essential that a vaccine contains either a strong B cell component in the form of a structural antigen, a strong T cell component in the form of peptides

to be presented on MHC molecules, or both.

4.2 Protein engineering & Peptide grafting

The objective of protein engineering is to modify a protein such that it gains a new function or displays new features. Examples of this could be to increase the solubility or stability of a protein [68, 69], creating multifunctional proteins by fusing two protein domains [70] or increasing the efficacy of a protein by improving it's ability to bind another protein [71, 72]. The different approaches to achieve these goals are diverse, but generalize into directed evolution, semi-rational design, and rational design [73]. All three approaches focus on modifying the DNA sequence encoding the protein, thus endowing it with this novel trait. In directed evolution, a library of variants of the protein are generated through random mutagenesis, and subsequently screened for the desired traits [74]. In rational design, specific modifications are applied to the DNA sequence, e.g. when information on the structure is available to guide the modification [68]. The semi-rational design approach is a combination of the other two, where information on structure is used to generate a library of variants [75].

In manuscript 2 we perform a type of rational design called peptide grafting to increase the number of immune epitopes in a protein. Typically, peptide grafting through sequence engineering is done via helix or loop grafting [76, 77, 71, 72, 78], where an α-helical or a loop (coil) region of a protein is substituted for a novel peptide that conforms to the same secondary structure. This approach theoretically works for any scaffold protein with known surface-exposed loops or α-helices, but requires that the grafted peptide can form a similar secondary structure 4.2. Alternative research directions focus on developing stable and soluble scaffold proteins, into which arbitrary peptides can be grafted [79, 80, 81, 82].

Previous approaches rely on selection of graft sites that are relatively independent from the rest of the structure [83]. This means that grafts should not be performed in regions of the protein that have a high number of inter-residue contacts, as they are likely to stabilize the fold. Effectively, this means that grafting is restricted to surface-exposed loops and helices. In most use cases, this is not a limitation, as most grafting scenarios revolve around adding novel functionalities to the surface of a scaffold protein, such as receptor specificity [78] or increased stability [84].

The approach presented in manuscript 2 represents a universal framework, which allows grafting of arbitrary peptides into arbitrary scaffold proteins.

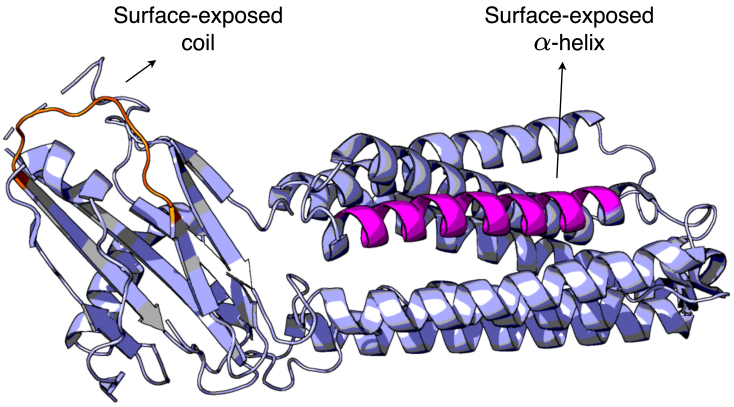


Figure 4.2: Examples of loop-grafting and helix-grafting

Chapter 5

Efficient Generative Modelling of Protein Structure Fragments Using a Deep Markov Model

The following manuscript was published as part of the conference proceedings for the International Conference on Machine Learning (ICML) 2021 and was selected for a contributed talk for the probabilistic programming conference (PROBPROG21).

The paper describes a model called Bayesian Inference for FRagments of protein STructures (BIFROST), a novel approach to the problem of generating fragment libraries for energy-based modelling of protein structures (see section 2.2.1) for details on the fragment library problem). The presented DMM is a model of local protein structure explicitly conditioned on the sequence of amino acids. It focuses solely on sampling ϕ/ψ dihedral angles conditioned on the amino acid sequence. The model was trained with the SVI procedure described in section 3.3.2. We show that the model can generate fragment libraries with a quality on par with the current state of the art, while doing so at a dramatically improved run time. To the extent of my knowledge, this model is the first deep and probabilistic generative model of local protein structure.

Efficient Generative Modelling of Protein Structure Fragments using a Deep Markov Model

Christian B. Thygesen [1 2] **Ahmad Salim Al-Sibahi** [1] **Christian S. Steenmanns** [2] **Lys S. Moreta** [1]
Anders B. Sørensen [* 2] **Thomas Hamelryck** [* 1 3]

Abstract

Fragment libraries are often used in protein structure prediction, simulation and design as a means to significantly reduce the vast conformational search space. Current state-of-the-art methods for fragment library generation do not properly account for aleatory and epistemic uncertainty, respectively due to the dynamic nature of proteins and experimental errors in protein structures. Additionally, they typically rely on information that is not generally or readily available, such as homologous sequences, related protein structures and other complementary information. To address these issues, we developed BIFROST, a novel take on the fragment library problem based on a Deep Markov Model architecture combined with directional statistics for angular degrees of freedom, implemented in the deep probabilistic programming language Pyro. BIFROST is a probabilistic, generative model of the protein backbone dihedral angles conditioned solely on the amino acid sequence. BIFROST generates fragment libraries with a quality on par with current state-of-the-art methods at a fraction of the run-time, while requiring considerably less information and allowing efficient evaluation of probabilities.

1. Introduction

Fragment libraries (Jones & Thirup, 1986) find wide application in protein structure prediction, simulation, design and experimental determination (Trevizani et al., 2017; Chikenji et al., 2006; Boomsma et al., 2012). Predicting the fold of a protein requires evaluating a conformational space that is too vast for brute-force sampling to be feasible (Levinthal, 1969). Fragment libraries are used in a divide-and-conquer approach, whereby a full length protein is divided into a manageable sub-set of shorter stretches of amino acids for which backbone conformations are sampled. Typically, sampling is done using a finite set of fragments derived from experimentally determined protein structures. Fragment libraries are used in state-of-the-art protein structure prediction frameworks such as Rosetta (Rohl et al., 2004), I-TASSER (Roy et al., 2010), and AlphaFold (Senior et al., 2019).

Generally, knowledge-based methods for protein structure prediction follow two main strategies: homology (or template-based) modelling (Eswar et al., 2006; Šali & Blundell, 1993; Song et al., 2013) and *de novo* modelling (Rohl et al., 2004). Both approaches assume that the native fold of a protein corresponds to the minimum of a physical energy function and make use of statistics derived from a database of known proteins structures (Alford et al., 2017; Leaver-Fay et al., 2013). Whereas homology modelling relies on the availability of similar structures to limit the search space, knowledge-based *de novo* protocols require extensive sampling of the conformational space of backbone angles (figure 1).

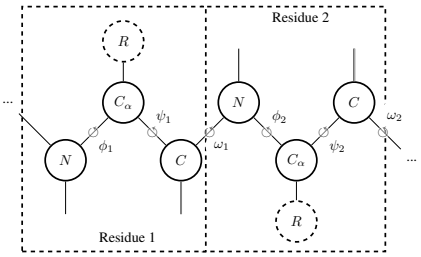


Figure 1. Schematic of the three dihedral angles (ϕ, ψ, and ω) that parameterise the protein backbone. R represents the side chain.

To overcome the shortcomings of either strategy, modelling tools like Rosetta (Rohl et al., 2004) use a combined approach of extensive sampling and prior information. Rosetta employs simulated annealing of backbone conformations

[*]Equal contribution [1]Department of Computer Science, University of Copenhagen, Copenhagen, Denmark [2]Evaxion Biotech, Copenhagen, Denmark [3]Department of Biology, University of Copenhagen, Copenhagen, Denmark. Correspondence to: Christian B. Thygesen <christiank.thygesen@di.ku.dk>, Thomas Hamelryck <thamelry@bio.ku.dk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

according to an energy function (Alford et al., 2017), while reducing the conformational space by sampling fragments of typically 3 or 9 amino acids at a time (Simons et al., 1997).

Fragments are typically extracted from experimentally determined protein structures in the Protein Data Bank (Berman et al., 2000) and used in prediction based on similarities in sequence and sequence-derived features (Gront et al., 2011; Kalev & Habeck, 2011; Santos et al., 2015; De Oliveira et al., 2015; Trevizani et al., 2017; Wang et al., 2019). Generative probabilistic models of protein backbone angles (Hamelryck et al., 2006; Boomsma et al., 2008; Bhattacharya et al., 2016; Edgoose et al., 1998; Li et al., 2008; Lennox et al., 2010) offer an alternative way to construct fragment libraries and aim to represent the associated epistemic and aleatory uncertainty. In this case, epistemic uncertainty is due to experimental errors from the determination of protein structures, while aleatory or inherent uncertainty is due to the dynamic nature, or flexibility, of proteins (Best, 2017).

Here, we present BIFROST - Bayesian Inference for FRagments Of protein STructures - a deep, generative, probabilistic model of protein backbone angles that solely uses the amino acid sequence as input. BIFROST is based on an adaptation of the Deep Markov Model (DMM) architecture (Krishnan et al., 2017) and represents the angular variables (ϕ and ψ) in a principled way using directional statistics (Mardia & Jupp, 2008). Finally, BIFROST makes it possible to evaluate the probability of a backbone conformation given an amino acid sequence, which is important for applications such as sampling the conformational space of proteins with correct statistical weights in equilibrium simulations (Boomsma et al., 2014).

2. Background and related work

Probabilistic, generative models of local protein structure Most generative, probabilistic models of local protein structure are Hidden Markov Models (HMMs) that represent structure and sequence based on the assumption of a Markovian structure (Hamelryck et al., 2012). The first such models did not include the amino acid sequence (Edgoose et al., 1998), discretised the angular variables (Bystroff et al., 2000), or used continuous, but lossy representations (Camproux et al., 1999; Hamelryck et al., 2006), making sampling of conformations with atomic detail problematic. These early models are thus probabilistic but only approximately "generative" at best. TorusDBN (Boomsma et al., 2008) was the first joint model of backbone angles and sequence that properly accounted for the continuous and angular nature of the data. Others introduced richer probabilistic models of local protein structure including Dirichlet Process mixtures of HMMs (DPM-HMMs) Lennox et al. (2010) and Conditional Random Fields (CRFs) (Zhao et al., 2010; 2008). As far as we know, BIFROST is the first deep generative model of local protein structure that aims to quantify the associated aleatory and epistemic uncertainty using an (approximate) Bayesian posterior.

Deep Markov Models The DMM, introduced in (Krishnan et al., 2017), is a generalisation of the variational autoencoder (VAE) (Kingma & Welling, 2014) for sequence or time series data. Related stochastic sequential neural models were reported by Fraccaro et al. (2016) and Chung et al. (2015). Published applications of DMMs include natural language processing tasks (Khurana et al., 2020), inference of time series data (Zhi-Xuan et al., 2020), and human pose forecasting (Toyer et al., 2017). Our application of the DMM and the modifications made to the standard model will be described in section 3.3.

3. Methods

3.1. Data set

BIFROST was trained on a data set of fragments derived from a set of 3733 proteins from the *cullpdb* data set (Wang & Dunbrack, 2005). Quality thresholds were (i) resolution < 1.6Å, (ii) R-factor < 0.25, and (iii) a sequence identity cutoff of 20%. For the purpose of reliable evaluation, sequences with $> 20\%$ identity to CASP13 targets were removed from the dataset.

Fragments containing angle-pairs in disallowed regions of the Ramachandran plot (Ramachandran et al., 1963) were removed using the Ramalyze function of the crystallography software PHENIX (Liebschner et al., 2019). The resulting data set consisted of ~ 186000 9-mer fragments. Prior to training, the data was randomly split into train, test, and validation sets with a $60/20/20\%$ ratio.

3.2. Framework

The presented model was implemented in the deep probabilistic programming language Pyro, version 1.3.0 (Bingham et al., 2019) and Pytorch version 1.4.0 (Paszke et al., 2019). Training and testing were carried out on a machine equipped with an Intel Xeon CPU E5-2630 and Tesla M10 GPU. The model trains on a single GPU and converges after 150 epochs for a total training time of approximately 34 hours.

3.3. Model

BIFROST consists of a DMM (Krishnan et al., 2017) with an architecture similar to an Input-Output HMM (IO-HMM) (Bengio & Frasconi, 1995). The model employs the Markovian structure of an HMM, but with continuous, as opposed to discrete, latent states ($\mathbf{z}$) and with *transition and emission neural networks* instead of transition and emission matrices.

Consequently, the latent states are iteratively transformed using the transition neural network, such that the value of the current latent state depends on the previous state and the (processed) amino acid information at that position (figure 2).

Observed angles (ϕ and ψ) are generated from the latent state sequence by applying an *emitter neural network* at each position (figure 2). Since the backbone angle ω is most often narrowly distributed around $180°$, this degree of freedom is not included in the current version of BIFROST.

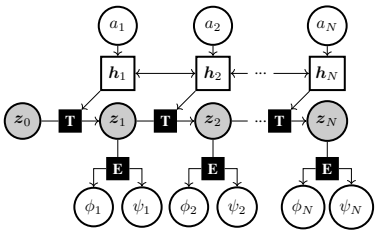


Figure 2. The BIFROST model. Grey nodes are latent random variables, white circular nodes are observed variables, white rectangular nodes represent hidden states from a bidirectional Recurrent Neural Network (RNN) H, and black squares represent neural networks. E and T denote the emitter and the transition network, respectively.

The structure of the model is shown in figure 2. For notational simplicity, the sequence of ϕ and ψ pairs will be denoted by $\mathbf{x}$. The joint distribution of the latent variable $\mathbf{z}$ and the angles $\mathbf{x}$ conditioned on the amino acid sequence $\mathbf{a}$ with length N of the graphical model in figure 2 factorises as

$$p(\mathbf{z}, \mathbf{x}|\mathbf{a}) = \prod_{n=1}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{h}_n(\mathbf{a}))p(\mathbf{x}_n|\mathbf{z}_n) \quad (1)$$

where $\boldsymbol{h}_n(\mathbf{a})$ is the deterministic hidden state generated at position n by a bidirectional RNN H with parameters $\boldsymbol{\theta}_H$ running across the amino acid sequence. The bidirectional RNN incorporates information from amino acids upstream and downstream of position n. The initial latent state $\mathbf{z}_0$ is treated as a trainable parameter and is thus shared for all sequences.

The transition densities are given by a multivariate Gaussian distribution,

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{h}_n(\mathbf{a})) = \mathcal{N}(\boldsymbol{\mu}_T(\mathbf{z}_{n-1}, \mathbf{h}_n(\mathbf{a})), \boldsymbol{\Sigma}_T(\mathbf{z}_{n-1}, \mathbf{h}_n(\mathbf{a}))) \quad (2)$$

where the mean vector ($\boldsymbol{\mu}_T$) and the (diagonal) covariance matrix ($\boldsymbol{\Sigma}_T$) are given by a neural network T parameterised by $\boldsymbol{\theta}_T$.

The emission densities are given by a *bivariate periodic student-T distribution* (Pewsey et al., 2007) (section 3.5) such that

$$p(\mathbf{x}_n|\mathbf{z}_n) = \mathcal{T}(\mathbf{x}_n|\nu_E(\mathbf{z}_n), \boldsymbol{\mu}_E(\mathbf{z}_n), \boldsymbol{\Sigma}_E(\mathbf{z}_n)) \quad (3)$$

where the single, shared degree of freedom (ν_E), the vector of two means ($\boldsymbol{\mu}_E$), and the 2×2 diagonal covariance matrix ($\boldsymbol{\Sigma}_E$) of the distribution are given by a neural network E parameterised by $\boldsymbol{\theta}_E$.

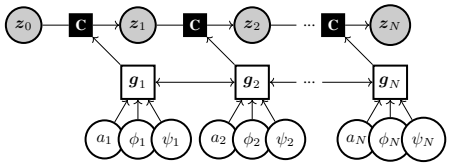


Figure 3. Variational distribution for approximating the posterior. Grey nodes are latent random variables, white circular nodes are observed variables, white rectangular nodes represent hidden states from a bidirectional RNN G, while black squares represent neural networks. C denotes the combiner network.

3.4. Estimation

In order to perform inference of the intractable posterior, we introduce a variational distribution or *guide* q (Kingma & Welling, 2019) (figure 3), which makes use of a *combiner neural network* C parameterised by $\boldsymbol{\zeta}_C$,

$$q(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{a}, \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_C(\mathbf{z}_{n-1}, \mathbf{g}_n(\mathbf{a}, \mathbf{x})), \boldsymbol{\Sigma}_C(\mathbf{z}_{n-1}, \mathbf{g}_n(\mathbf{a}, \mathbf{x}))) \quad (4)$$

where $\boldsymbol{g}_n(\boldsymbol{a}, \boldsymbol{x})$ is the deterministic hidden state generated at position n by a bidirectional RNN G with parameters $\boldsymbol{\zeta}_G$ running across the amino acid sequence $\boldsymbol{a}$ and the angles $\boldsymbol{x}$.

For the parameters of the neural networks $(\boldsymbol{\zeta}_C, \boldsymbol{\zeta}_G, \boldsymbol{\theta}_T, \boldsymbol{\theta}_E, \boldsymbol{\theta}_H)$, point estimates are obtained using Stochastic Variational Inference (SVI), which optimises the Evidence Lower Bound (ELBO) using stochastic gradient descent (SGD) (Kingma & Welling, 2014; 2019). The ELBO variational objective is given by

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\zeta}}(\boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\zeta}}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{a})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}|\boldsymbol{a}) - \log q_{\boldsymbol{\zeta}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{a})\right] \quad (5)$$

where $\boldsymbol{\zeta} = (\boldsymbol{\zeta}_C, \boldsymbol{\zeta}_G)$ and $\boldsymbol{\theta} = (\boldsymbol{z}_0, \boldsymbol{\theta}_T, \boldsymbol{\theta}_E, \boldsymbol{\theta}_H)$ are the parameters of the guide and the model, respectively.

3.5. Periodic student T distribution

As angle-pairs are periodic values, i.e. distributed on a torus (Boomsma et al., 2008), they need to be modelled by an appropriate periodic distribution. Traditionally, angles are assumed distributed according to the von Mises distribution, which is defined by a mean that can be any real number and a concentration parameter, which can be any positive number. SVI showed poor performance when the von Mises distribution was used. Here, we circumvent this by representing the likelihood of the angles by a student T distribution that is wrapped around a circle (Pewsey et al., 2007). This allows for appropriate modelling of the periodicity of the angles, while being more robust with regards to outlier issues than the von Mises distribution due to the wider tails of the T distribution. It should be noted as well that Pewsey et al. (2007) showed that the wrapped student T distribution can approximate the von Mises distribution closely.

3.6. Neural network architecture overview

The overall architecture is based on the originally proposed DMM (Krishnan et al., 2017) with modifications. The main difference is the addition of an RNN H in the model that processes the amino acid sequence $\mathbf{a}$, thus providing explicit conditioning on the amino acid sequence. A similar architecture was used by Fraccaro et al. (2016) for time series. In the guide, a second RNN G is used that processes the angles and the amino acid sequence during training. The initial values for both RNNs are treated as trainable parameters. In addition to the RNNs, the model contains an emitter network E and a transition network T, while the guide relies on a combiner network C.

Emitter architecture The emitter network E parameterises the emission probabilities as stated in equation 3. E is a feed-forward neural network containing two initial layers that branch into three. One output branch is a single layer that outputs the degree of freedom of the Student T distribution, which is shared between the two angles. The other two branches output a mean μ and a standard deviation σ for ϕ and ψ, respectively. Each hidden layer of the neural network contained 200 neurons with rectified linear unit (ReLU) activation. Output layers for μ values had no activation, as the periodic distribution automatically transforms values to a range between $-\pi$ and π. Output layers for σ and degrees of freedom ν used softplus activation to ensure positive, real numbered values. The architecture of E is depicted in figure 4.

Transition and combiner architecture The transition network T and the combiner network C specify the transition densities from the previous to the current latent state of the model (equation 2) and the guide (equation 4), respectively. In the original DMM (Krishnan et al., 2017), C was inspired by the Gated Recurrent Unit (GRU) architecture (Cho et al., 2014), while T was a simple feed forward network. Here, both C and T were based on GRU cells to allow for better horizontal information flow (figure 5).

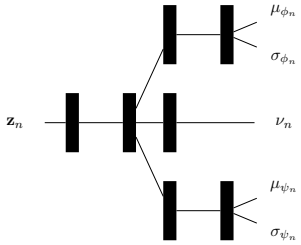


Figure 4. Architecture of the emitter neural network, E. Black rectangles represent ReLU-activated fully connected layers.

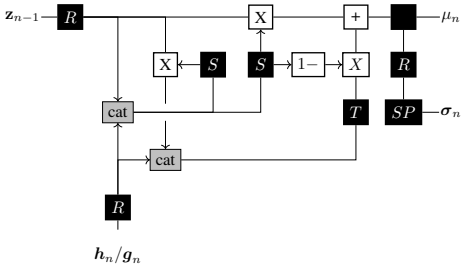


Figure 5. Architecture of the transition T and combiner C neural networks. Black squares represent single neural network layers activated by a ReLU (R), sigmoid (S), tanh (T), softplus (SP) or no activation. White squares represent element-wise mathematical operations. Gray squares represent tensor concatenation. Note that the network takes as input either $\boldsymbol{h}_n$ or $\boldsymbol{g}_n$ obtained from the RNN in the model or the guide, respectively.

The total number of parameters in BIFROST are shown in table 1.

3.7. Hyperparameter optimization

A simple hyperparameter search was performed with the test ELBO as the selection criterion (data not shown). The final model was trained with a learning rate of 0.0003 with a scheduler reducing the learning rate by 90% when no improvement was seen for 10 epochs. Minibatch size was 200. The Adam optimiser was used with a β_1 and β_2 of 0.96 and 0.999 respectively. The latent space dimensionality was 40. All hidden activations (if not specified above) were ReLU activations. We employed norm scaling of the gradient to a norm of 10.0. Finally, early stopping was employed with a patience of 50 epochs.

Neural networks					Z_0	
E	T	C	H	G	p	q
24 805	142 280	142 280	89 200	90 000	40	40
Total parameters: 488 645						

Table 1. Number of parameters in BIFROST. E: Emitter, T: Transition, C: Combiner, H: model RNN, G: guide RNN, p: model, q: guide

3.8. Sampling from the model

The BIFROST model (figure 2) is designed with explicit conditioning on amino acid sequences allowing a simple and efficient ancestral sampling approach that eliminates the need for using the guide for predictions. Thus, the guide is used solely for the purpose of model estimation and is discarded upon sampling.

3.9. Fragment library generation and benchmarking

Fragment libraries are a collection of fragments, consisting of typically 3 or 9 amino acids with known backbone angles. Here, we focus on fragments of nine amino acids. For each fragment in a protein, 200 possible backbone conformations are sampled from BIFROST resulting in a set of $(L-8)\times 200$ fragment candidates, where L is the number of amino acids in the protein. These candidates are compared to the observed fragment by calculating the angular root mean square deviation (RMSD) between the corresponding angles as proposed in Boomsma et al. (2008). The choice of 9-mer fragments and the 200 samples per fragment were made to emulate the default behavior of the Rosetta fragment picker (see below), for fair comparison.

The aggregated quality of fragment libraries are generally represented by two metrics; *precision* and *coverage*. Precision is defined as the fraction of candidates with an RMSD to the observed below a certain threshold, whereas coverage is the fraction of positions covered by at least one candidate with an RMSD below a certain threshold. Evaluating the precision and coverage at increasing thresholds yields two curves, and the quality of the fragment library is quantified by the area under these two curves.

BIFROST was benchmarked against Rosetta's fragment picker (Gront et al., 2011) using the precision and coverage metrics. The fragment picker was run using default parameters, picking 200 fragments per position. Secondary structure predictions were performed using SAM-T08 (Karplus, 2009), PSIPRED (Jones, 1999) and Jufo (Leman et al., 2013). Sequences that were homologous to the targets were excluded (*–nohoms* flag).

Fragment libraries were generated for all available regular (denoted "T") targets from the latest installment of the bi-annual protein structure prediction competition Critical Assessment of Techniques for Protein Structure Prediction (CASP13).

3.10. Runtime comparison

In order to compare the runtime of BIFROST to that of the fragment picker, nine proteins of varying lengths were selected. Both tools generated 200 samples per fragment. The experiment was run on the same 32-core machine for both the fragment picker and BIFROST.

4. Results

To show that the model is able to capture general protein backbone behavior, angles were generated conditioned on the sequences of 5000 previously unseen fragments and compared to the observed angles. The model was able to recreate the observed Ramachandran plots with minimal added noise (figure 6).

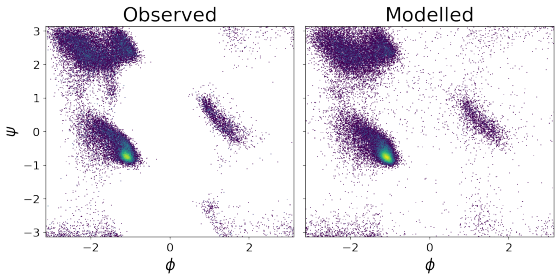


Figure 6. Observed and modelled aggregated Ramachandran plots

While most amino acids show angle distributions similar to the background in figure 6, glycine and proline are exceptions due to the nature of their side chains. The side chain of glycine is a single hydrogen atom, allowing the backbone to be exceptionally flexible, while the side chain of proline is covalently linked to the backbone restraining the conformational space. The modelled distribution of angles for these two unique cases, along with leucine to represent the general case, show that the model is able to capture specific amino acid properties (figure 7).

The left side of figure 8 shows a thin, smoothed coil representation of 100 samples from BIFROST conditioned on example 9-mer fragments that were observed to be either α-helix, β-strand, or coiled. The right side shows distributions of backbone RMSDs of 5000 sampled fragments to the observed structure from BIFROST and as picked by Rosetta's fragment picker.

The RMSDs were generally distributed towards 0Å for the α-helix case, showcasing BIFROSTs ability to predict this

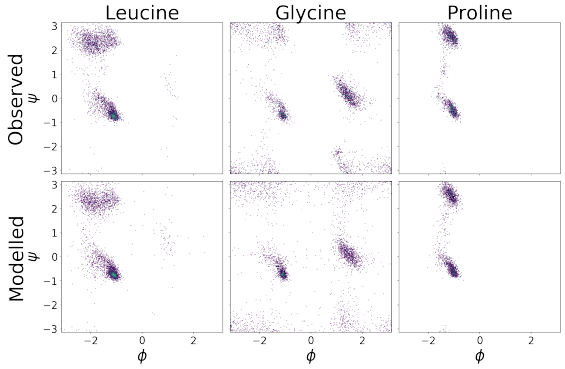


Figure 7. Amino acid specific Ramachandran plots

well defined secondary structure element. The model has more difficulty modelling β-strands and coils. However, the distributions of the RMSDs are nearly identical to those produced by the fragment picker. For coil fragments, the RMSDs were distributed around 3Å reflecting the inherent variability of those fragments.

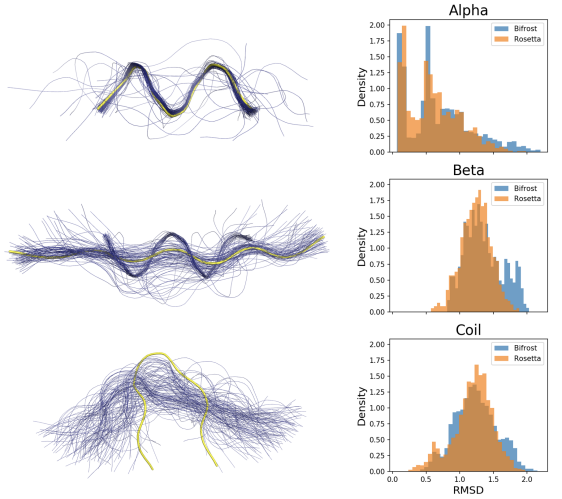


Figure 8. Left: 100 samples of backbone dihedral angles (blue) superimposed on the observed structures (yellow). For clarity, the backbones are represented as thin, smoothed coils instead of traditional cartoon representations. Right: Aggregated RMSDs of BIFROST-sampled conformations and conformations picked by Rosetta's fragment picker for sequences observed as α-helix, β-strand, and coil respectively.

BIFROST was benchmarked against Rosetta's fragment picker (Gront et al., 2011) on all publicly available CASP13 regular targets. BIFROST generated fragment libraries with comparable precision and coverage to the fragment picker (figure 9).

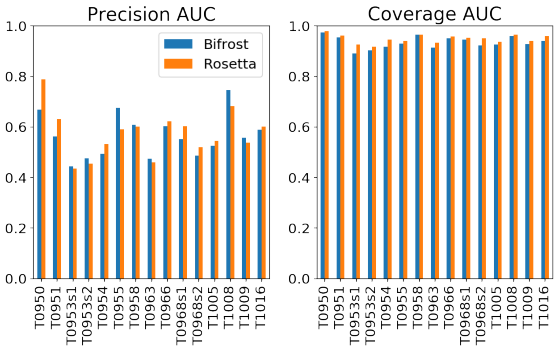


Figure 9. Comparison of fragment libraries generated by BIFROST, relying on just the amino acid sequences, against Rosetta's fragment picker, which uses external information and relies on ensemble predictions of secondary structure.

Finally, BIFROST enables efficient sampling of fragment libraries. The runtime of BIFROST and the fragment picker are compared in figure 10 on a set of nine proteins of varying lengths. Both runtimes roughly scale linearly with protein length, but BIFROST has a smaller constant term than the fragment picker.

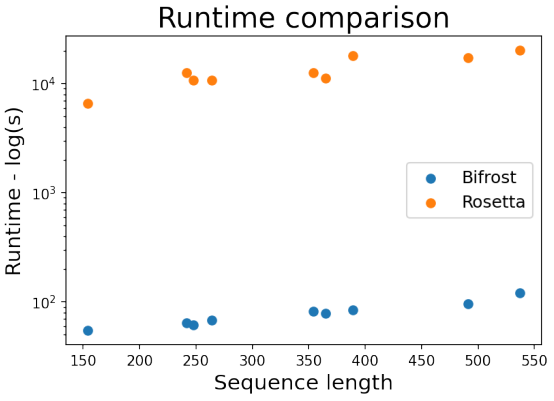


Figure 10. Runtime comparison between Rosetta's fragment picker and BIFROST on a set of nine proteins of varying lengths.

5. Discussion

BIFROST is a deep, generative model of local protein structure conditioned on sequence that provides a probabilistic approach to generating fragment libraries.

The quality of the generated fragment libraries is on par with Rosetta's fragment picker, despite using much less information, such as an ensemble of secondary structure predictors. Due to the probabilistic nature of BIFROST, distributions tend to be slightly wider than those resulting

from picking structural fragments from the PDB based on sequence similarity. This wider distribution plausibly reflects the dynamic nature of protein structure, which is not captured in the experimental data provided by static X-ray structures.

The model was estimated using SVI, relying on the ELBO variational objective. As the ELBO provides a lower bound on the log evidence (Kingma & Welling, 2014), we can evaluate the probability of a specific local structure given the sequence, simply by evaluating the ELBO. Evaluating the probability of fragments is crucial for correct sampling of the conformational space, for example in the case of equilibrium simulations of protein dynamics (Boomsma et al., 2014). The probabilities assigned by BIFROST can be used to decide how often a fragment should be sampled in the folding process. In contrast, existing methods do not provide an explicit measure of fragment confidence.

In this paper the focus was kept on fragments of nine residues for ease of comparison to the fragment picker. However, the DMM architecture of BIFROST allows generation of fragments of arbitrary length but with an observed drop-off in performance as the length of fragments are increased (data not shown).

Existing methods rely heavily on the availability of multiple sequence alignments (MSA) and other information, such as secondary structure predictors. As MSAs are not available for orphan proteins or synthetic proteins, the need for pure sequence based models is evident.

6. Acknowledgements

We acknowledge funding from the Innovation Fund Denmark under the grant "Accelerating vaccine development through a deep learning and probabilistic programming approach to protein structure prediction".

References

Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K., Labonte, J. W., Pacella, M. S., Bonneau, R., Bradley, P., Dunbrack, R. L., Das, R., Baker, D., Kuhlman, B., Kortemme, T., and Gray, J. J. The Rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, jun 2017. ISSN 15499626. doi: 10.1021/acs.jctc.7b00125.

Bengio, Y. and Frasconi, P. An input output HMM architecture. *Neural Information Processing Systems*, pp. 427–434, 1995. ISSN 15322092. doi: 10.1093/europace/euq350.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The Protein Data Bank, jan 2000. ISSN 03051048.

Best, R. B. Computational and theoretical advances in studies of intrinsically disordered proteins. *Current Opinion in Structural Biology*, 42:147–154, feb 2017. ISSN 0959440X. doi: 10.1016/j.sbi.2017.01.006.

Bhattacharya, D., Adhikari, B., Li, J., and Cheng, J. FRAGSION: Ultra-fast protein fragment library generation by IOHMM sampling. *Bioinformatics*, 32(13):2059–2061, 2016. ISSN 14602059. doi: 10.1093/bioinformatics/btw067.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., and Goodman, N. D. Pyro: Deep Universal Probabilistic Programming. *J. Mach. Learn. Res.*, 20: 28:1—-28:6, 2019.

Boomsma, W., Mardia, K. V., Taylor, C. C., Ferkinghoff-Borg, J., Krogh, A., and Hamelryck, T. A generative, probabilistic model of local protein structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105(26):8932–8937, 2008. ISSN 00278424. doi: 10.1073/pnas.0801715105.

Boomsma, W., Frellsen, J., and Hamelryck, T. *Probabilistic models of local biomolecular structure and their applications*. Springer, 2012. doi: 10.1007/978-3-642-27225-7_10.

Boomsma, W., Tian, P., Frellsen, J., Ferkinghoff-Borg, J., Hamelryck, T., Lindorff-Larsen, K., and Vendruscolo, M. Equilibrium simulations of proteins using molecular fragment replacement and NMR chemical shifts. *Proceedings of the National Academy of Sciences of the United States of America*, 111(38):13852–13857, 2014. ISSN 10916490. doi: 10.1073/pnas.1404948111.

Bystroff, C., Thorsson, V., and Baker, D. HMMSTR: A hidden Markov model for local sequence-structure correlations in proteins. *Journal of Molecular Biology*, 301(1):173–190, 2000. ISSN 00222836. doi: 10.1006/jmbi.2000.3837.

Camproux, A. C., Tuffery, P., Chevrolat, J. P., Boisvieux, J. F., and Hazout, S. Hidden Markov model approach for identifying the modular framework of the protein backbone. *Protein Engineering*, 12(12):1063–1073, 1999. ISSN 02692139. doi: 10.1093/protein/12.12.1063.

Chikenji, G., Fujitsuka, Y., and Takada, S. Shaping up the protein folding funnel by local interaction: Lesson from a structure prediction study. *Proceedings of the National Academy of Sciences*, 103(9):3141–3146, feb 2006. ISSN 0027-8424. doi: 10.1073/pnas.0508195103.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1724–1734. Association for Computational Linguistics (ACL), jun 2014. ISBN 9781937284961. doi: 10.3115/v1/d14-1179.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, volume 2015-Janua, pp. 2980–2988, 2015.

De Oliveira, S. H., Shi, J., and Deane, C. M. Building a better fragment library for de novo protein structure prediction. *PLoS ONE*, 10(4), 2015. ISSN 19326203. doi: 10.1371/journal.pone.0123998.

Edgoose, T., Allison, L., and Dowe, D. L. An MML classification of protein structure that knows about angles and sequence. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 585–596, 1998. ISSN 2335-6928.

Eswar, N., Webb, B., Marti-Renom, M. A., Madhusudhan, M., Eramian, D., Shen, M.-y., Pieper, U., and Sali, A. Comparative protein structure modeling using modeller. *Current Protocols in Bioinformatics*, 15(1):5.6.1–5.6.30, sep 2006. ISSN 1934-340X. doi: 10.1002/0471250953.bi0506s15.

Fraccaro, M., Sønderby, S. K., Paquet, U., and Winther, O. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, pp. 2207–2215, 2016.

Gront, D., Kulp, D. W., Vernon, R. M., Strauss, C. E., and Baker, D. Generalized fragment picking in rosetta: Design, protocols and applications. *PLoS ONE*, 6(8): 23294, 2011. ISSN 19326203. doi: 10.1371/journal.pone.0023294.

Hamelryck, T., Kent, J. T., and Krogh, A. Sampling realistic protein conformations using local structural bias. *PLoS Computational Biology*, 2(9):e131, sep 2006. ISSN 1553-7358. doi: 10.1371/journal.pcbi.0020131.

Hamelryck, T., Mardia, K., and Ferkinghoff-Borg, J. (eds.). *Bayesian Methods in Structural Bioinformatics*. Statistics for Biology and Health. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27224-0. doi: 10.1007/978-3-642-27225-7.

Jones, D. T. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292(2):195–202, 1999. ISSN 00222836. doi: 10.1006/jmbi.1999.3091.

Jones, T. A. and Thirup, S. Using known substructures in protein model building and crystallography. *The EMBO journal*, 5(4):819–22, apr 1986. ISSN 0261-4189.

Kalev, I. and Habeck, M. HHfrag: HMM-based fragment detection using HHpred. *Bioinformatics*, 27(22): 3110–3116, 2011. ISSN 13674803. doi: 10.1093/bioinformatics/btr541.

Karplus, K. SAM-T08, HMM-based protein structure prediction. *Nucleic Acids Research*, 37(SUPPL. 2):492–497, 2009. ISSN 03051048. doi: 10.1093/nar/gkp403.

Khurana, S., Laurent, A., Hsu, W.-N., Chorowski, J., Lancucki, A., Marxer, R., and Glass, J. A Convolutional Deep Markov Model for Unsupervised Speech Representation Learning. In *Proceedings of Interspeech*, jun 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

Kingma, D. P. and Welling, M. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237. doi: 10.1561/2200000056.

Krishnan, R. G., Shalit, U., and Sontag, D. Structured inference networks for nonlinear state space models. In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 2101–2109, sep 2017.

Leaver-Fay, A., O'Meara, M. J., Tyka, M., Jacak, R., Song, Y., Kellogg, E. H., Thompson, J., Davis, I. W., Pache, R. A., Lyskov, S., Gray, J. J., Kortemme, T., Richardson, J. S., Havranek, J. J., Snoeyink, J., Baker, D., and Kuhlman, B. Scientific benchmarks for guiding macromolecular energy function improvement. In *Methods in Enzymology*, volume 523, pp. 109–143. Academic Press Inc., 2013. ISBN 9780123942920. doi: 10.1016/B978-0-12-394292-0.00006-0.

Leman, J. K., Mueller, R., Karakas, M., Woetzel, N., and Meiler, J. Simultaneous prediction of protein secondary structure and transmembrane spans. *Proteins: Structure, Function and Bioinformatics*, 81(7):1127–1140, jul 2013. ISSN 08873585. doi: 10.1002/prot.24258.

Lennox, K. P., Dahl, D. B., Vannucci, M., Day, R., and Tsai, J. W. A Dirichlet process mixture of hidden Markov models for protein structure prediction. *Annals of Applied Statistics*, 4(2):916–942, 2010. ISSN 19326157. doi: 10.1214/09-AOAS296.

Levinthal, C. How to fold graciously. *Mössbauer Spectroscopy in Biological Systems Proceedings*, 24(41):22–24, 1969. ISSN 1041-1135. doi: citeulike-article-id:380320.

Li, S. C., Bu, D., Xu, J., and Li, M. Fragment-HMM: A new approach to protein structure prediction. *Protein Science*, 17(11):1925–1934, 2008. ISSN 09618368. doi: 10.1110/ps.036442.108.

Liebschner, D., Afonine, P. V., Baker, M. L., Bunkóczi, G., Chen, V. B., Croll, T. I., Hintze, B., Hung, L.-w., Jain, S., McCoy, A. J., Moriarty, N. W., Oeffner, R. D., Poon, B. K., Prisant, M. G., Read, R. J., Richardson, J. S., Richardson, D. C., Sammito, M. D., Sobolev, O. V., Stockwell, D. H., Terwilliger, T. C., Urzhumtsev, A. G., Videau, L. L., Williams, C. J., and Adams, P. D. Macromolecular structure determination using X-rays, neutrons and electrons: recent developments in Phenix. *Acta Crystallographica Section D Structural Biology*, 75(10):861–877, oct 2019. ISSN 2059-7983. doi: 10.1107/S2059798319011471.

Mardia, K. V. and Jupp, P. E. *Directional Statistics*. Wiley Series in Probability and Statistics. John Wiley and Sons, Inc., Hoboken, NJ, USA, jan 2008. ISBN 9780470316979. doi: 10.1002/9780470316979.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An imperative style, high-performance deep learning library, 2019. ISSN 23318422.

Pewsey, A., Lewis, T., and Jones, M. C. The wrapped t family of circular distributions. *Australian and New Zealand Journal of Statistics*, 49(1):79–91, 2007. ISSN 1467842X. doi: 10.1111/j.1467-842X.2006.00465.x.

Ramachandran, G. N., Ramakrishnan, C., and Sasisekharan, V. Stereochemistry of polypeptide chain configurations, 1963. ISSN 00222836.

Rohl, C. A., Strauss, C. E., Misura, K. M., and Baker, D. Protein structure prediction using rosetta. *Methods in Enzymology*, 383:66–93, 2004. ISSN 00766879. doi: 10.1016/S0076-6879(04)83004-0.

Roy, A., Kucukural, A., and Zhang, Y. I-TASSER: A unified platform for automated protein structure and function prediction. *Nature Protocols*, 5(4):725–738, 2010. ISSN 17502799. doi: 10.1038/nprot.2010.5.

Šali, A. and Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, 234(3):779–815, dec 1993. ISSN 00222836. doi: 10.1006/jmbi.1993.1626.

Santos, K. B., Trevizani, R., Custodio, F. L., and Dardenne, L. E. Profrager Web Server: Fragment libraries generation for protein structure prediction. In *Proceedings of the International Conference on Bioinformatics and Computational Biology (BIOCOMP)*, pp. 38, 2015.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D. T., Silver, D., Kavukcuoglu, K., and Hassabis, D. Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). *Proteins: Structure, Function and Bioinformatics*, 87(12):1141–1148, 2019. ISSN 10970134. doi: 10.1002/prot.25834.

Simons, K. T., Kooperberg, C., Huang, E., and Baker, D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of Molecular Biology*, 268(1):209–225, 1997. ISSN 00222836. doi: 10.1006/jmbi.1997.0959.

Song, Y., Dimaio, F., Wang, R. Y. R., Kim, D., Miles, C., Brunette, T., Thompson, J., and Baker, D. High-resolution comparative modeling with RosettaCM. *Structure*, 21 (10):1735–1742, 2013. ISSN 09692126. doi: 10.1016/j.str.2013.08.005.

Toyer, S., Cherian, A., Han, T., and Gould, S. Human pose forecasting via deep Markov models. In *DICTA 2017 - 2017 International Conference on Digital Image Computing: Techniques and Applications*, volume 2017-Decem, pp. 1–8. Institute of Electrical and Electronics Engineers Inc., jul 2017. ISBN 9781538628393. doi: 10.1109/DICTA.2017.8227441.

Trevizani, R., Dio, F. L. C., Santos, K. B. D., and Dardenne, L. E. Critical features of fragment libraries for protein structure prediction. *PLoS ONE*, 12(1), 2017. ISSN 19326203. doi: 10.1371/journal.pone.0170131.

Wang, G. and Dunbrack, R. L. PISCES: Recent improvements to a PDB sequence culling server. *Nucleic Acids Research*, 33(SUPPL. 2), 2005. ISSN 03051048. doi: 10.1093/nar/gki402.

Wang, T., Qiao, Y., Ding, W., Mao, W., Zhou, Y., and Gong, H. Improved fragment sampling for ab initio protein structure prediction using deep neural networks. *Nature Machine Intelligence*, 1(8):347–355, aug 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0075-7.

Zhao, F., Li, S., Sterner, B. W., and Xu, J. Discriminative learning for protein conformation sampling. *Proteins: Structure, Function and Genetics*, 73(1):228–240, oct 2008. ISSN 08873585. doi: 10.1002/prot.22057.

Zhao, F., Peng, J., Debartolo, J., Freed, K. F., Sosnick, T. R., and Xu, J. A probabilistic and continuous model of protein conformational space for template-free modeling. *Journal of Computational Biology*, 17(6):783–798, 2010. ISSN 10665277. doi: 10.1089/cmb.2009.0235.

Zhi-Xuan, T., Soh, H., and Ong, D. Factorized inference in deep Markov models for incomplete multimodal time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10334–10341, apr 2020. ISSN 2374-3468. doi: 10.1609/aaai.v34i06.6597.

Chapter 6

Design of a broad SARS-CoV-2 vaccine with a universal grafting approach using a deep generative model of local protein structure

FOLLOWING up on the proposed model of local protein structure in manuscript 1 [85], we showcase an application of the model that deviates quite a bit from the originally intended use case. While the model originally acted as a probabilistic and generative approach to the fragment library problem, we realized that the model has other use cases. The model produces latent states accounting for both sequence and strcuture. This means, that we can use these latent states as novel similarity measures accounting for both. Up until this point any similarity measure between two sequences would only take into account the amino acid sequence. This essentially means, that two proteins can only be deemed similar if the amino acid sequences are similar. However, suppose one wants to alter a protein sequence without majorly affecting the structure, sequence similarity is just one of the components that should be considered. In manuscript 2, this is exactly what we will attempt to do; modify a known protein (the spike protein of the coronavirus) in order to introduce immune epitopes while preserving the expression and neutralising antibody response towards the wildtype.

In order to do this, we will perform peptide grafting, which is a process of changing an amino acid sequence by replacing part of it with the amino acid sequence of a foreign peptide (see section 4.2). If we rely only on sequence identity for this, we would be restricted to introducing only minor changes to the sequences, thus limiting the number of possible grafts dramatically. With the model described in manuscript 1 it is possible to compare two sequences

in a manner, that takes the structural outcome into account in addition to the sequence. We show, that using such a model allows enriching a protein with foreign much larger and diverse pool of foreign peptides.

The manuscript focuses on producing a vaccine that is as immunogenic as possible. Therfore we graft peptides, that are likely to be presented to the T-cell receptors of the immune system. The pool of peptides, from which we can graft, peptide:MHCII prediction (section 4). This tool is a neural network that is trained on identified MHCII-binding peptides. However, it should be noted that the grafting process described is generally applicable to any pool of peptides.

We show that the vaccine constructs designed using the BIFROST model are equally likely to result in structurally conserved proteins as an approach that relies only on sequence identity, while allowing for introducing larger changes to the amino acid sequence.

As of the time of writing, further experiments are being performed in order to validate the data and hypotheses presented in manuscript 2.

DESIGN OF A BROAD SARS-COV-2 VACCINE WITH A UNIVERSAL GRAFTING APPROACH USING A DEEP GENERATIVE MODEL OF LOCAL PROTEIN STRUCTURE

A PREPRINT

Christian B. Thygesen*
Copenhagen University
Department of Computer Science
&
Evaxion Biotech A/S
ckt@evaxion-biotech.com

Michael S. Klausen
Evaxion Biotech A/S
msk@evaxion-biotech.com

Søren V. Kofoed
Evaxion Biotech A/S
svk@evaxion-biotech.com

Christian S. Steenmanns
Evaxion Biotech A/S
csh@evaxion-biotech.com

Thomas W. Hamelryck
Copenhagen University
Department of Computer Science
thamelry@binf.ku.dk

Anders B. Sørensen
Evaxion Biotech A/S
abs@evaxion-biotech.com

September 28, 2022

ABSTRACT

The need for rapid vaccine development has become evident in the recent COVID-19 pandemic caused by the SARS-CoV-2 virus. Even though vaccines were developed faster than ever significant human suffering and economic costs was observed, while the risk of novel variant or strains that escape immunity still remains.The design of novel broadly protective vaccines could alleviate the impact of such future events, relying on genetic information from multiple variants or strains. Such an approach is supported by the observation that individuals previously infected with SARS-CoV-1 and who received a vaccination against SARS-CoV-2 showed increased cross-protection. Here, we present a novel design approach based on this concept ,to achieve broadly protective vaccines against SARS-CoV-2, by include immune epitopes from across several strains. The platform utilises deep generative neural networks as a similarity function for guiding the modification of a vaccine based on the spike protein through peptide grafting. We show that this novel grafting approach allows us to modify the spike protein to an unprecedented extent, without compromising the structural fold as shown by the vaccine's ability to induce antibodies against the unmodified spike protein. The grafting approach presented here, is applicable to any scaffold protein and allows grafting of arbitrary peptides.

Keywords Deep Probabilistic Programming · Vaccine design · SARS-CoV-2 · Protein structure modelling · Machine Learning · Immunology · Bioinformatics · Peptide grafting

1 Introduction

The COVID-19 pandemic has highlighted the need for rapid vaccine design and development against Coronavirus. While vaccines against the disease were developed at an unprecedented speed [1, 2], the mutational rate of SARS-CoV-2 confers a risk of emergence of novel variants capable of escaping immunity [3, 4]. In addition, previous outbreaks of corona virus causing severe human disease (SARS-CoV-1 and MERS-CoV) and the four common circulating strains

*Corresponding author

(229E, NL63, OC43 and HKU1), underscores the general threat of emergence of novel corona virus strains able to infect humans and the need for broadly protective vaccines .

The generation of memory B-cells producing neutralizing antibodies against the spike protein, responsible for ACE2 receptor binding for human cell entry, was found essential for SARS-CoV-2 protection [5, 6, 7, 8]. Due to the phylogenetic diversity of human corona viruses, belonging to the alpha and beta lineages, little natural immunity was conferred against SARS-CoV-2 from previous infections [9] even for the closely related SARS-CoV-1 found in the Sarbecovirus subclade [10]. A similar lack of cross protective immunity against other strains has been observed in animals vaccinated with SARS-CoV-2 based vaccine designs [2]. However, studies show evidence of cross-protection among individuals previously infected with SARS-CoV-1 and subsequently vaccinated against SARS-CoV-2 [11, 12]. These individuals showed neutralizing antibodies against a range of Sarbecoviruses, indicating the possibility of a vaccine inducing broad protection across a subclade by combining components from multiple viruses in a single vaccine. Evidence of the feasibility for such an approach in a preclinical setting was recently published relying on the combination of multiple receptor binding (RBD) domains from the spike proteins of different strains [13, 14, 15, 16, 17]. While successful, these approaches are limited in the number of RBDs from different strains that can be combined due to size constrains. Additionally, important T cell epitopes located in the spike protein outside the RBD are lost [18].

We propose an adaptable vaccine platform based on a novel peptide grafting approach of immune epitopes using the full length spike protein as scaffold. Previous studies showed that a vaccination regimen consisting of peptide-priming with $CD4^+$ epitopes from the virus proteome induces the antibody-response [19]. However, neutralizing antibodies are only induced for the proteins from which the grafted epitopes originate - known as paired protein specificity [19]. That same study showed, however, that priming with epitopes from proteins not previously recognized by the immune system induced antibodies against these proteins. This indicates that grafting epitopes from a range of different proteins could induce antibody-mediated endocytosis based on recognition of otherwise ignored proteins.

Thus, in this study we focus on enriching the number of $CD4^+$ epitopes. The sequence of the S1 domain containing the RBD is kept intact to ensure the generation of neutralizing antibodies against this domain. The S2 stem domain, on the other hand is unlikely to elicit neutralizing antibodies, as it does not interact with ACE2. Therefore, this domain is enriched for predicted and experimentally validated $CD4^+$ T cell epitopes from a range of different sources.

Previous peptide grafting approaches focus on manipulating the sequence of surface-exposed loops or α helices [20, 21, 22, 23, 24, 25] to avoid abrogating the structural fold. However, T cell epitopes are not constrained to these regions, as they arise from the antigen processing pathway, where a protein is degraded and processed into small peptides to be presented by the MHC complexes [26]. Therefore, strictly seen from the perspective of antigen presentation, epitopes can be grafted at any position in the scaffold protein. We therefore propose a grafting approach fueled by a deep, generative model (BIFROST [27]) deployed as a similarity function for assessing the likelihood that a peptide can be grafted at an arbitrary position in a scaffold protein without affecting the structural fold. We compare the approach to a homology-based method, and show that the generative model allows for grafting of arbitrary peptides that do not need to show high identity to the wildtype sequence of the scaffold protein. We apply the method to design a proof-of-concept plasmid DNA vaccine against SARS-CoV-2 (section 2.3.1) by enriching the spike-protein with known or predicted T cell epitopes, thus increasing the valency of the vaccine. While the strategy presented here focuses on producing SARS-CoV-2 vaccine constructs, the platform is readily adaptable to design epitope-enriched vaccines against any pathogen.

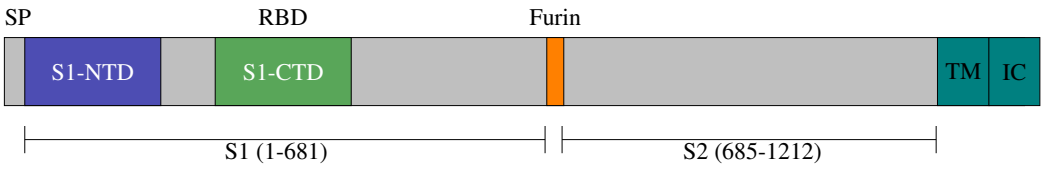
2 Methods

2.1 $CD4^+$ epitope prediction

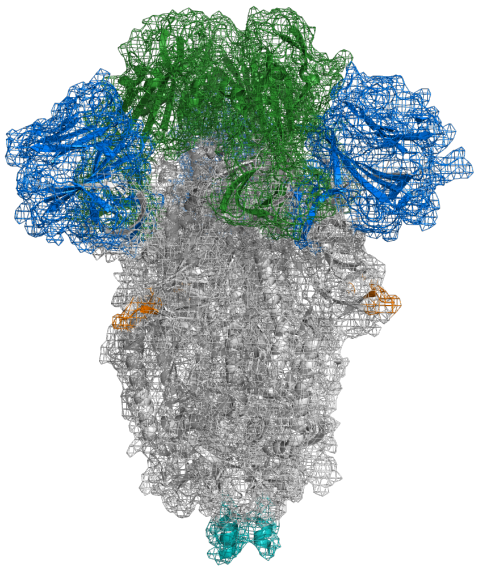
$CD4^+$ epitope predictions were performed using Evaxion's internal peptide:MHC prediction tool against the H2-IAb allele of the mouse strain C57BL/6. For the sake of maintaining a large pool of graft candidates, we consider peptides with a predicted ligand probability higher than 0 positive epitopes. Predicted ligands were sorted by their ligand probability score, in order to obtain a prioritized list of graft candidates.

2.2 Epitope grafting guided by deep generative neural networks

BIFROST [27], a deep generative model of local protein structure, guides the epitope grafting (figure 2). As the model is a latent variable model, similar to a variational autoencoder [29], it infers distributions over latent representations capturing higher-order information. Protein fragments that are similar in terms of backbone conformation and amino acid sequence show similar distributions over their latent representations. Thus, if two fragments are cose in latent space, they are likely to occupy the same structural space.



(a) Schematic representation of the sequence of the spike protein, with important domains highlighted. SP: Signal peptide, RBD: Receptor Binding Domain, TM: Transmembrane region, IC: Intracellular region.



(b) Cartoon representation of the spike protein structure (PDB code 6VXX). Domains and sites of relevance are colored according to the legend in figure 1a. Green: S1-CTD (RBD), blue: S1-NTD, orange: Furin cleavage site, teal: Transmembrane region. Figure produced in pymol [28].

Figure 1: Sequence and structural view of the spike protein. CTD: C-terminal domain, NTD: N-terminal domain.

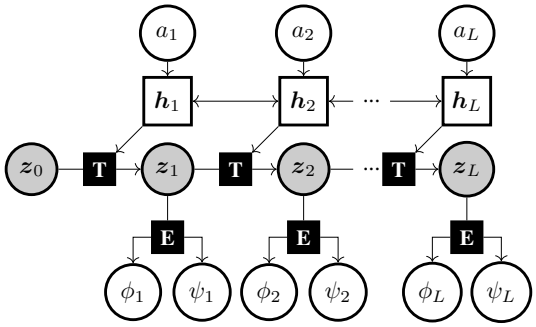


Figure 2: The BIFROST model of local protein structure. The amino acid sequence $(a_1, ..., a_L)$ is processed by a bidirectional recurrent neural network (RNN). A transition network (T) parameterises a distribution over the latent representation at position l given the RNN hidden state at position l and the previous latent state $(\mathbf{z}_{l-1})$. White circular nodes are observed random variables, grey nodes are latent random variables. Black boxes represent neural networks, while white boxes represent hidden states from an RNN.

Grafting is performed by encoding the sequences of epitopes and regions of the scaffold protein into distributions over the latent representations using BIFROST (figure 2). Each distribution is a multivariate Gaussian of shape $L \times D$ where L is the length of the epitope and D is the dimensionality of the latent space. BIFROST produces latent representations as vectors of length 40, i.e. $D = 40$. The similarity between two peptides is defined by the KL-divergence:

$$\text{Similarity} = \text{KL}(p(\mathbf{z}|\mathbf{a}_{\text{target}})||p(\mathbf{z}|\mathbf{a}_{\text{candidate}})) \tag{1}$$

where $p(\mathbf{z}|\mathbf{a}_{\text{target}})$ and $p(\mathbf{z}|\mathbf{a}_{\text{candidate}})$ are distributions over latent states given the amino acid sequence of the target peptide and candidate peptide, respectively.

We empirically determined a threshold for deciding whether a peptide can be grafted by mapping our entire pool of graft candidates against every position in the wildtype spike protein. We observed a small peak was below a KL divergence of 13, which corresponded to the 1st percentile (see supplementary material, section 5.1). Thus, the cutoff value for allowing a graft was set to a KL divergence below 13.

Given a pool of peptides prioritised by predicted MHC ligand probability, we attempt grafting first producing latent state distributions for each position in the scaffold protein ($p(\mathbf{z}|\mathbf{a}_s pike)$). We then produce latent distributions for the peptide ($p(\mathbf{z}|\mathbf{a}_i)$. The KLD between each position in $p(\mathbf{z}|\mathbf{a}_s pike)$ and $p(\mathbf{z}|\mathbf{a}_i)$ is calculated. If a position has a KLD below 13, grafting is performed. This process is repeated until a specified number of grafts are completed (6 or 12). Grafting of a new peptides is only performed if there is a match where a graft has not already been performed. Additionally, grafting is not performed in the S1 domain or the furin cleavage site to preserve structural antibody epitopes from the RBD as well as preserving furin cleavage (figure 1a). The grafting algorithm is summarized in figure 3.

As we hypothesise that the BIFROST approach allows us to graft from arbitrary sources, we this approach grafted epitopes experimentally validated to be immunogenic in C57BL/6 mice from the Immune Epitope Database (`http://www.iedb.org`) [30]. By grafting validated epitopes we are more likely to observe a T cell response against these epitopes in an immunization study in those mice, proving the presence of the epitopes. For a comprehensive overview of the epitope grafts, see table S1.

2.3 Epitope grafting guided by sequence identity

As a baseline for evaluating the novel BIFROST based grafting approach, we used a more classical multiple sequence alignment (MSA) method. We generated multiple sequence alignments of coronavirus and coronavirus-like spike proteins by searching the UniProt database using hidden Markov model (HMM) based approaches HHblits [31] and HMMER [32]. 15-mer candidate epitopes were identified using peptide:MHC prediction. 6 or 12 non-overlapping candidate epitopes were grafted by transferring the sequence from the donor protein to the grafted protein, keeping the MSA-column position constant. Insertions and deletions in the donor protein sequence are disallowed to avoid frame shifts; only residue-to-residue substitutions are allowed. Due to the reliance on sequence identity, the MSA-based approach is limited to graft peptides originating from spike homologs of the betacoronavirus genus.

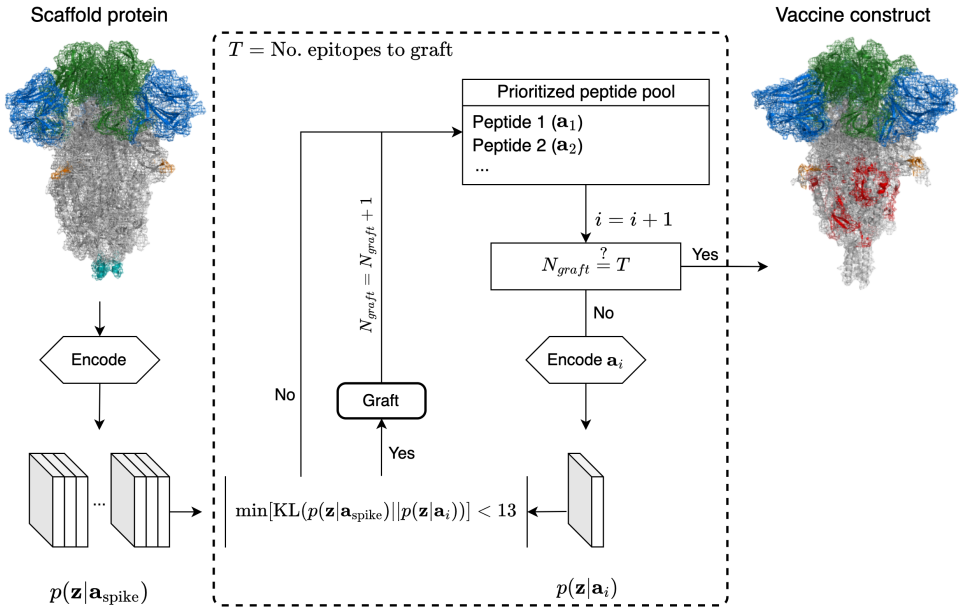


Figure 3: The BIFROST-based grafting algorithm. Epitopes prioritized by MHCII ligand probability are encoded one at a time and compared to every position in the scaffold protein. If a KL-divergence below 13 is observed, the epitope is grafted and the grafted position is blacklisted for further iterations. Grafted positions are highlighted in red in the vaccine construct to the right.

2.3.1 Construct design

To benchmark the grafting approach, we designed three types of constructs. The first group consist of the MSA approach (section 2.3), allowing few amino acid differences compared to the wildtype spike sequence. We allowed a total of 6 or 12 grafts. The second group consists of two constructs designed using the BIFROST-driven approach (section 2.2), also grafting 6 or 12 epitopes. Finally, we evaluated each graft in the MSA constructs using the BIFROST model to identify poor grafts. If an MSA-grafted epitope did not meet the BIFROST threshold, we reverted the grafted sequence to the wildtype (section 2.2).

We base all grafts on the same plasmid backbone based on the pTVG4 plasmid [33] encoding (i) a spike protein with the furin cleavage site knocked [34] and knock-in of two proline residues, K986P & V987P, stabilising the pre-fusion state [35, 36], (ii) a chemokine based antigen presenting cell targeting unit consisting of CCL19 with signal peptide [37], and (iii) a T4 domain inducing spike protein trimerisation [38, 39] (figure 4). CCL19 and spike are separated by a rigid linker (EAAAK), while spike and T4 are separated by a flexible linker (GSGSGS) to allow for proper folding. We included C-terminal his-tags for expression quantification. Synthetic mini genes of the constructs were synthesised and cloned by Aldevron (Fargo, North Dakota) into the plasmid backbone.

2.3.2 Phylogenetic tree construction

We built phylogenetic trees by creating multiple sequence alignments using MAFFT [40] with default parameters, followed by construction of the trees using iqtree [41] with the GTR20 substitution model.

2.4 Expression study

Cell Culturing Adherent HEK293 (ATCC CRL-1573) cells were grown in DMEM (Merk, Catalog # D6546) with 10% FBS, 1% GlutaMAX™ (ThermoFisher, Catalog # 35050061) and penicillin/streptomycin according to the manufacturer's instructions. The cells were maintained in a humidified incubator at 37°C with 5% CO2.

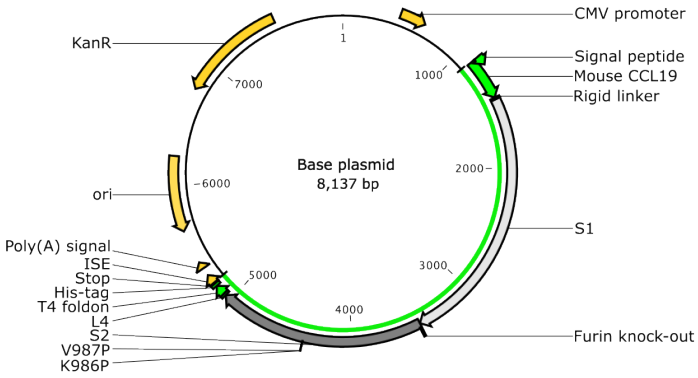


Figure 4: Plasmid used as backbone for grafting. The coding sequence of the vaccine product is highlighted by green. Grafts were performed in the S2 doman (dark grey), but disallowed in S1 (light grey). Components specific to the pTVG4 plasmid are shown in yellow.

Transfection of HEK cells Adherent HEK293 cells were seeded in a poly-L-lysine coated 24 well plate (500.000 cells/well) and transiently transfected the following day using Lipofectamine 3000 (ThermoFisher, Catalog # L3000015). Briefly, 1μg of DNA plasmid, 3μl of Lipofectamine and 2μg of P3000 reagent were diluted in 100μl of OptiMEM (ThermoFisher, Catalog # 31985062) and added to the cells according to the manufacturer's instructions. After 24 hours the medium in the wells was replaced by serum free culture medium and the supernatant collected 48 hours after for further analysis.

CCL19 ELISA Protein expression was assessed using Mouse CCL19/MIP-3 DuoSet antibodies (R&D Systems, Catalog # DY440) to develop a specific sandwich ELISA following the manufacturer's instructions. Briefly, supernatant of HEK293 transfected cells was added to 96 well-plates precoated with capture antibody according to the manufacturer's instructions. The supernatants were incubated for 2h at room temperature, the plates were washed with 0.05% Tween® 20 in PBS and incubated with biotinylated detection antibody for 2h at room temperature, then washed and incubated with streptavidin-conjugated horseradish-peroxidase for 20 minutes at room temperature. After a final wash, captured proteins were detected with 1:1 mixture of Color Reagent A (H2O2) and Color Reagent B (Tetramethylbenzidine) (R&D Systems, Catalog # DY999), and the absorbance measured at 450 nm.

2.5 Immunogenicity study - IgG ELISA & ELISPOT

In vivo setup 6 to 8 weeks old C57BL/6 females were acquired from Janvier Labs (France). All the experiments were conducted under the license 2017-15- 0201-01209 from the Danish Animal Experimentation Inspectorate in accordance with the Danish Animal Experimentation Act (BEK no. 12 of 7/01/2016), which is compliant with the European directive (2010/63/EU).

DNA plasmids were formulated in saline and delivered via electroporation immediately after intramuscular injection (IM). Mice received 2 IM immunizations in left and right tibialis anterior muscles for a final volume of $100\mu l$ per immunization. Dosing occurred on days 0 and 28 of the experiment.

Tail vein blood was sampled on days 13, 20, and 28 and Retro-orbital bleeding was performed at termination for serum collection. Spleens were made to single cell suspension splenocytes.

Anti-RBD and Anti-FL-spike IgG ELISA For the detection of specific antibodies in sera against RBD or the full-length spike construct ELISA was used. MaxiSorp microtiter plates (Thermo Scientific, Catalog # 442404) were coated with $1\text{mg}/\text{mL}$ SARS-CoV-2 recombinant RBD protein (Proteogenix, Catalog # PX-COV-P046) or $2\text{mg}/\text{mL}$ SARS-CoV-2 Spike full-length in trimer (Proteogenix, Catalog # PX-COV-P049-100). The coated plates were incubated with sera from vaccinated mice for 2 hours for the binding of cognate antibodies and non-specific antibodies were

washed off. The total specific IgG was detected by HRP-conjugated polyclonal rabbit anti-mouse IgG (Sigma, Catalog # A9044) using 1-Step Slow TMB-ELISA (Thermofisher, Catalog # 34024) for development. The end result is the number of absorbance values (450 nm) determined at a series of sera dilutions (e.g. 1:100-1:1.000.000) to determine the end-point sera titer.

Evaluation of T cell responses using IFN-γ ELISpot Enzyme-linked immunospot (ELISpot) flat-bottomed, 96-well hydrophobic high protein binding immobilon-P membrane (Merck Millipore, Catalog # MAIPS4510) were coated overnight at 4°C with 5μg/mL murine IFN-γ capture antibody (BD, Catalog # 51-2525KZ) in PBS. To investigate to what extent the different treatment groups harbor immune recognition of the RBD and grafted peptides, 5 x 105 viable splenocytes of single-cell suspensions from individual mice were added to the wells of the coated plates and were stimulated in duplicates with 5μg/mL of each peptide for 20 hours at 37°C with 5% CO2. Cells stimulated with ConA (Concanavalin A) or medium with DMSO alone were used as controls. This was followed by incubation with $2\mu g/mL$ murine IFN-γ detection antibody (BD, Catalog # 51-1818KA) for 2 hours at room temperature, 1:100 dilution of Streptavidin-Horseradish Peroxidase (HRP) (BD, Catalog # 557630) conjugate solution in PBS containing 10% FBS for 1 hour in the dark at room temperature, and then 100μL of AEC substrate solution (BD, Catalog # 551951) for 15 minutes in the dark for spots to develop. ELISPOT plates were air dried and read on a CTL ELISPOT analyzer to count Spot Forming Units (SFUs) in each well. All counts were normalized to SFUs per 106 splenocytes.

3 Results

3.1 Construct design

We designed a total of seven vaccine constructs exhibiting different levels of deviation from the wildtype sequence in terms of amino acid changes. We applied The MSA and BIFROST approaches to design two constructs each where 6 and 12 epitopes were grafted. Based on the assumption that BIFROST can identify graft candidates accounting for more than just the sequence, we applied BIFROST to evaluate each grafted epitope in the MSA top 6/12 constructs. Consequently, two further constructs were designed where grafts identified by BIFROST as poor were undone. We term these constructs MSA + BIFROST (table 1). As the MSA approach relies on sequence identity and HMM models, the resulting constructs shared a high sequence identity with the wildtype, while the BIFROST-based approach allowed a higher level of deviation (figure 5). Naturally, the MSA constructs evaluated by BIFROST (termed MSA + BIFROST) show higher sequence identity to the wildtype compared to the corresponding MSA construct. For a complete overview of grafted epitopes, refer to supplementary table S1.

Table 1: Overview of constructs designed using either MSA or BIFROST-driven grafting approaches.

Construct	No. grafts	Strategy	Candidate pool	No. amino acid differences
Wildtype	0	Wildtype	N/A	0
MSA top 6	6	MSA	Spike homologs	9
MSA top 12	12	MSA	Spike homologs	25
MSA top 6 (S)	5	MSA + BIFROST	Spike homologs	6
MSA top 12 (S)	8	MSA + BIFROST	Spike homologs	14
BIFROST top 6	6	BIFROST	IEDB epitopes	72
BIFROST top 12	12	BIFROST	IEDB epitopes	136

3.2 Expression

We evaluated the expression designed constructs in HEK293 cells as described in section 2.4. Overall, the grafted constructs showed lower expression than the wildtype spike protein. The BIFROST-designed constructs showed the same or higher expression than the MSA approaches grafting the same number of peptides (figure 6). Additionally, the MSA+BIFROST constructs showed an increase in expression by $\sim 25\%$. This is likely in part due to the fact, that the constructs contain fewer grafts.

3.3 Proper folding of modified spike protein indicated by cross-reactive RBD antibody titers to the wildtype protein

We further evaluated the native state of the constructs through their ability to generate cross-reactive antibodies to wildtype RBD or full length spike protein in an ELISA based on sera from mice immunogenicity study. Mice were

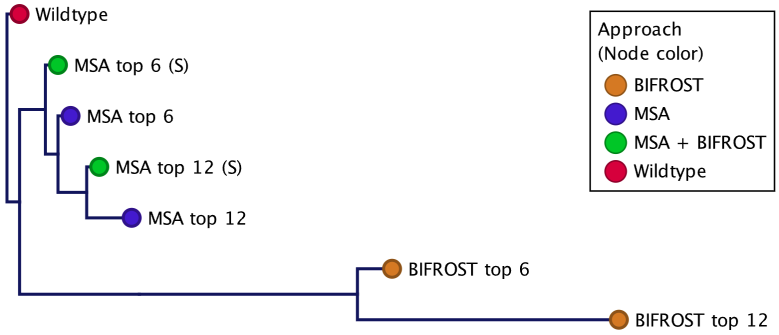


Figure 5: Phylogenetic tree showing the evolutionary distance of constructs from wildtype spike protein.

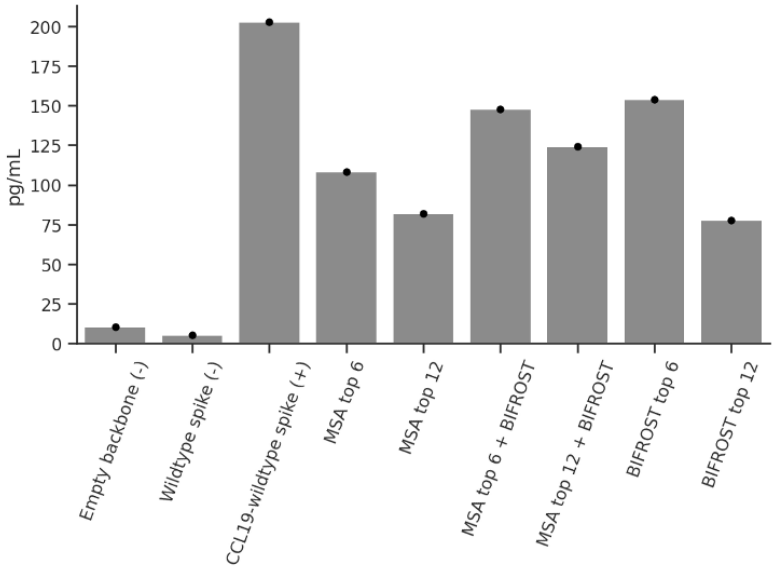
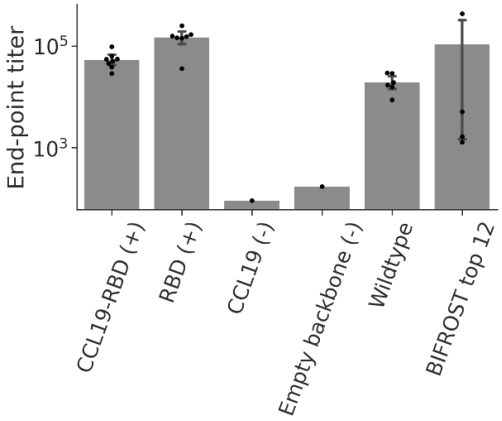


Figure 6: Readout from CCL19 ELISA expression study. Constructs denoted with (-) are negative controls, while (+) indicates positive controls.

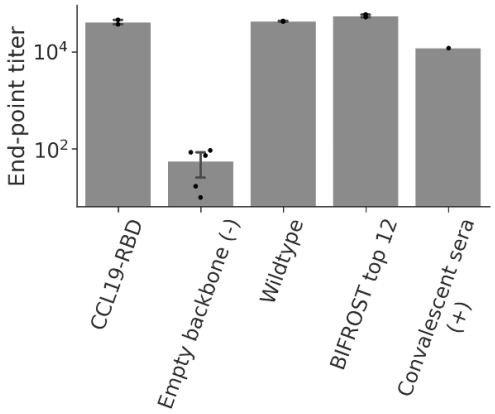
immunized twice with 100μg DNA plasmids encoding RBD, full length spike or BIFROST-grafted spike with 12 epitopes grafted. Immunization with the modified vaccine construct resulted in the generation of IgG antibodies recognizing the wildtype RBD and spike proteins (figure 7a) indicating in vivo expression of a spike protein with a correct overall fold. Additionally, mice immunized with the grafted construct were able to elicit similar antibody titers against the full length wildtype spike 7b as the wild type construct.

3.4 T cell response evaluation

We found that the modified spike protein elicited T cell responses specific for a pool of the grafted epitopes, while retaining a T cell response against epitopes from the RBD. Mice immunized with the wildtype spike or the RBD-based construct only showed a response against the epitopes from the RBD (figure 8a. Additionally, a specific T cell response was mounted towards 10 out of the 12 grafted epitopes 8b.

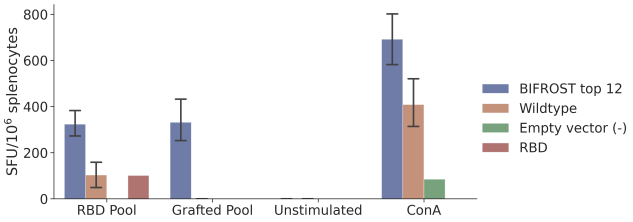


(a) Total IgG Anti-RBD end-point titers

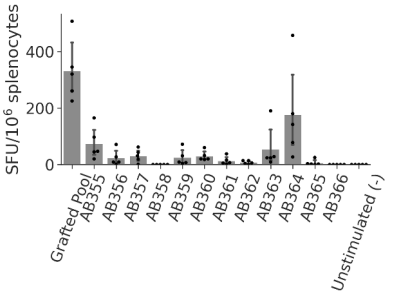


(b) Total IgG Anti-full length spike end-point titers

Figure 7: End-point IgG antibody titers against RBD and full length spike, respectively. (+) indicates positive controls, whereas (-) denotes negative controls.



(a) IFN-γ readout



(b) T cell response against individual grafted epitopes (AB355-366)

Figure 8: Number of IFN-γ producing cells discovered after restimulation with epitopes originating either from the pool of grafted peptides or from the RBD.

3.5 Broad coverage of Betacoronaviruses

From the results above it is evident that the BIFROST-based grafting method allows sourcing of more divergent epitopes, when compared to the MSA approach. To emphasize this, we built a phylogenetic tree of a representative set of genomes from the Betacoronavirus genus (figure 9). The set includes a variety of MERS, HCoV, SARS-CoV-1, and SARS-CoV-2 sequences. We designed three constructs, allowing grafting from a pool of predicted epitopes from the SARS-CoV-2 genome, Sarbecovirus genomes, or the betacoronavirus genus. The intensity of the bands in the three outer rings, corresponds to the number of epitopes in each of the vaccine constructs rediscovered in the specific strain, thus representing immunogenic coverage. Grafting only from SARS-CoV-2 provides coverage to the SARS-CoV-2 clade, and part of the SARS-CoV-1 clade. Grafting epitopes from across Sarbecoviruses and Betacoronavirues increases the epitope coverage to include the entire SARS-CoV-1 clade as well as many of the human Coronaviruses. For the Betacoronavirus grafting pool, we observe coverage against the main MERS-CoV strains as well.

4 Discussion

Due to the high mutational rate exhibited by viruses such as SARS-CoV-2 and the presences of novel strains in various animal reservoirs, there is an evident need for a platform that can rapidly adapt to emerging strains escaping existing immunity. We demonstrated such a platform based on the spike protein of SARS-CoV-2 enriched with $CD4^+$ epitopes. We showed that it is possible to use a universal approach for grafting foreign peptides in a manner that preserves

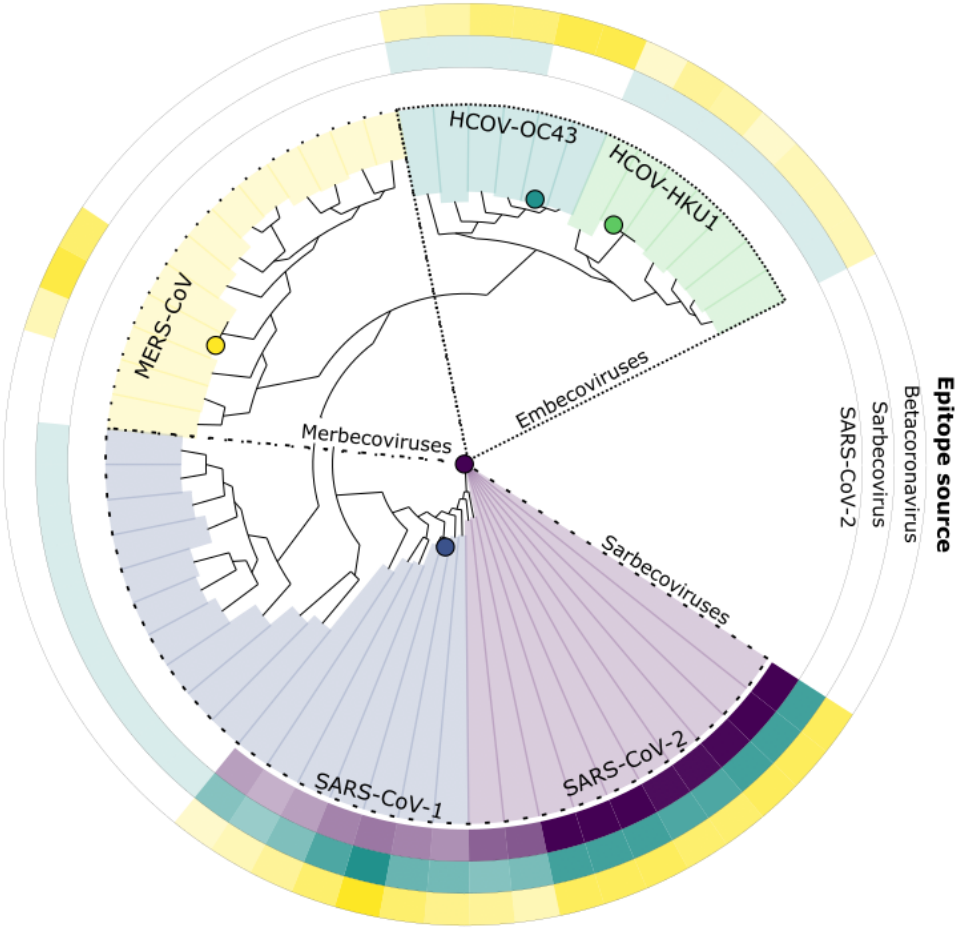


Figure 9: Phylogenetic tree of the betacoronavirus genus. Leaves highlighted by nodes correspond to either SARS-CoV-2, SARS-CoV-1, MERS-CoV, or one of the human coronaviruses HCOV-OC43, HCOV-HKU1. Clade colors correspond to the closest of the five reference strains. The outer rims correspond to three vaccine constructs enriched with epitopes from three different sources. The intensity of the bands in the outer rings correspond to the number of peptide MHCII ligands found in the designed constructs covering a specific strain.

important B-cell epitopes. The proposed approach outperforms a simpler approach relying only on sequence identity, as we observe equal or higher expression levels despite deviating more from the wildtype sequence.

We observed that mice immunized with the heavily modified spike protein, featuring no less than 12 grafted epitopes, were in fact able to cross-reactive antibodies against the wildtype RBD and full length spike. This suggests conservation of the fold of the modified vaccine construct. The level of anti-RBD IgG titers was lower in the grafted construct group than observed for mice immunized with the wildtype spike. We argue that this is in part due to the lower expression observed in the grafted construct, as well as the fact that the epitopes grafted in the BIFROST top 12 construct were

sourced from proteins that did not originate from the spike protein or even the corona virus. We also observed that the grafted construct exhibited the same levels of IgG antibodies specific for the full-length wildtype spike. As we are grafting epitopes from arbitrary foreign sources, we do not expect to see improved neutralizing antibody titers. However, the data presented here indicates that there is some preservation of the fold of the spike protein, as we observe IgG antibodies specific for the wildtype RBD and full length spike protein.

The data presented here are an encouraging proof-of-concept of the feasibility of a grafting-based adaptive vaccine platform. However, it is important to note, that the experiments performed here are "proxies" of measuring the expression (through CCL19-ELISA) and proper fold via antibody titers (through an RBD-ELISA). The next steps for validation of this platform is to perform physiochemical experiments on the proteins modified using the BIFROST model. This could be circular dichroism or even crystallography to confirm that the modified protein in fact folds as expected. The fact that we can show, that immunized mice elicit an immune response towards RBD of the wildtype spike does not necessarily provide final proof that the entire vaccine construct folds as expected.

References

[1] Nicole Lurie, Melanie Saville, Richard Hatchett, and Jane Halton. Developing Covid-19 Vaccines at Pandemic Speed. *New England Journal of Medicine*, 382(21):1969–1973, 5 2020.

[2] Kizzmekia S. Corbett, Darin K. Edwards, Sarah R. Leist, Olubukola M. Abiona, Seyhan Boyoglu-Barnum, Rebecca A. Gillespie, Sunny Himansu, Alexandra Schäfer, Cynthia T. Ziwawo, Anthony T. DiPiazza, Kenneth H. Dinnon, Sayda M. Elbashir, Christine A. Shaw, Angela Woods, Ethan J. Fritch, David R. Martinez, Kevin W. Bock, Mahnaz Minai, Bianca M. Nagata, Geoffrey B. Hutchinson, Kai Wu, Carole Henry, Kapil Bahl, Dario Garcia-Dominguez, LingZhi Ma, Isabella Renzi, Wing-Pui Kong, Stephen D. Schmidt, Lingshu Wang, Yi Zhang, Emily Phung, Lauren A. Chang, Rebecca J. Loomis, Nedim Emil Altaras, Elisabeth Narayanan, Mihir Metkar, Vlad Presnyak, Cuiping Liu, Mark K. Louder, Wei Shi, Kwanyee Leung, Eun Sung Yang, Ande West, Kendra L. Gully, Laura J. Stevens, Nianshuang Wang, Daniel Wrapp, Nicole A. Doria-Rose, Guillaume Stewart-Jones, Hamilton Bennett, Gabriela S. Alvarado, Martha C. Nason, Tracy J. Ruckwardt, Jason S. McLellan, Mark R. Denison, James D. Chappell, Ian N. Moore, Kaitlyn M. Morabito, John R. Mascola, Ralph S. Baric, Andrea Carfi, and Barney S. Graham. SARS-CoV-2 mRNA vaccine design enabled by prototype pathogen preparedness. *Nature*, 586(7830):567–571, 10 2020.

[3] Rachel T. Eguia, Katharine H. D. Crawford, Terry Stevens-Ayers, Laurel Kelnhofer-Millevolte, Alexander L. Greninger, Janet A. Englund, Michael J. Boeckh, and Jesse D. Bloom. A human coronavirus evolves antigenically to escape antibody immunity. *PLOS Pathogens*, 17(4):e1009453, 4 2021.

[4] Aayatti Mallick Gupta, Jaydeb Chakrabarti, and Sukhendu Mandal. Non-synonymous mutations of SARS-CoV-2 leads epitope loss and segregates its variants. *Microbes and Infection*, 22(10):598–607, 11 2020.

[5] Kizzmekia S. Corbett, Martha C. Nason, Britta Flach, Matthew Gagne, Sarah O'Connell, Timothy S. Johnston, Shruti N. Shah, Venkata Viswanadh Edara, Katharine Floyd, Lilin Lai, Charlene McDanal, Joseph R. Francica, Barbara Flynn, Kai Wu, Angela Choi, Matthew Koch, Olubukola M. Abiona, Anne P. Werner, Juan I. Moliva, Shayne F. Andrew, Mitzi M. Donaldson, Jonathan Fintzi, Dillon R. Flebbe, Evan Lamb, Amy T. Noe, Saule T. Nurmukhambetova, Samantha J. Provost, Anthony Cook, Alan Dodson, Andrew Faudree, Jack Greenhouse, Swagata Kar, Laurent Pessaint, Maciel Porto, Katelyn Steingrebe, Daniel Valentin, Serge Zouantcha, Kevin W. Bock, Mahnaz Minai, Bianca M. Nagata, Renee van de Wetering, Seyhan Boyoglu-Barnum, Kwanyee Leung, Wei Shi, Eun Sung Yang, Yi Zhang, John-Paul M. Todd, Lingshu Wang, Gabriela S. Alvarado, Hanne Andersen, Kathryn E. Foulds, Darin K. Edwards, John R. Mascola, Ian N. Moore, Mark G. Lewis, Andrea Carfi, David Montefiori, Mehul S. Suthar, Adrian McDermott, Mario Roederer, Nancy J. Sullivan, Daniel C. Douek, Barney S. Graham, and Robert A. Seder. Immune correlates of protection by mRNA-1273 vaccine against SARS-CoV-2 in nonhuman primates. *Science*, 373(6561), 9 2021.

[6] Edward E. Walsh, Robert W. Frenck, Ann R. Falsey, Nicholas Kitchin, Judith Absalon, Alejandra Gurtman, Stephen Lockhart, Kathleen Neuzil, Mark J. Mulligan, Ruth Bailey, Kena A. Swanson, Ping Li, Kenneth Koury, Warren Kalina, David Cooper, Camila Fontes-Garfias, Pei-Yong Shi, Özlem Türeci, Kristin R. Tompkins, Kirsten E. Lyke, Vanessa Raabe, Philip R. Dormitzer, Kathrin U. Jansen, Uğur Şahin, and William C. Gruber. Safety and Immunogenicity of Two RNA-Based Covid-19 Vaccine Candidates. *New England Journal of Medicine*, 383(25):2439–2450, 12 2020.

[7] Lindsey R. Baden, Hana M. El Sahly, Brandon Essink, Karen Kotloff, Sharon Frey, Rick Novak, David Diemert, Stephen A. Spector, Nadine Rouphael, C. Buddy Creech, John McGettigan, Shishir Khetan, Nathan Segall, Joel Solis, Adam Brosz, Carlos Fierro, Howard Schwartz, Kathleen Neuzil, Lawrence Corey, Peter Gilbert, Holly Janes, Dean Follmann, Mary Marovich, John Mascola, Laura Polakowski, Julie Ledgerwood, Barney S. Graham,

Hamilton Bennett, Rolando Pajon, Conor Knightly, Brett Leav, Weiping Deng, Honghong Zhou, Shu Han, Melanie Ivarsson, Jacqueline Miller, and Tal Zaks. Efficacy and Safety of the mRNA-1273 SARS-CoV-2 Vaccine. *New England Journal of Medicine*, 384(5):403–416, 2 2021.

[8] Fernando P. Polack, Stephen J. Thomas, Nicholas Kitchin, Judith Absalon, Alejandra Gurtman, Stephen Lockhart, John L. Perez, Gonzalo Pérez Marc, Edson D. Moreira, Cristiano Zerbini, Ruth Bailey, Kena A. Swanson, Satrajit Roychoudhury, Kenneth Koury, Ping Li, Warren V. Kalina, David Cooper, Robert W. Frenck, Laura L. Hammitt, Özlem Türeci, Haylene Nell, Axel Schaefer, Serhat Ünal, Dina B. Tresnan, Susan Mather, Philip R. Dormitzer, Uğur Şahin, Kathrin U. Jansen, and William C. Gruber. Safety and Efficacy of the BNT162b2 mRNA Covid-19 Vaccine. *New England Journal of Medicine*, 383(27):2603–2615, 12 2020.

[9] Ren Yang, Jiaming Lan, Baoying Huang, Ruhan A, Mingqing Lu, Wen Wang, Wenling Wang, Wenhui Li, Yao Deng, Gary Wong, and Wenjie Tan. Lack of antibody-mediated cross-protection between SARS-CoV-2 and SARS-CoV infections. *EBioMedicine*, 58:102890, 8 2020.

[10] Alireza Tabibzadeh, Maryam Esghaei, Saber Soltani, Parastoo Yousefi, Mahsa Taherizadeh, Fahimeh Safarnezhad Tameshkel, Mahsa Golahdooz, Mahshid Panahi, Hossein Ajdarkosh, Farhad Zamani, and Mohammad Hadi Karbalaie Niya. Evolutionary study of COVID-19, severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) as an emerging coronavirus: Phylogenetic analysis and literature review. *Veterinary Medicine and Science*, 7(2):559–571, 3 2021.

[11] Chee-Wah Tan, Wan-Ni Chia, Barnaby E. Young, Feng Zhu, Beng-Lee Lim, Wan-Rong Sia, Tun-Linn Thein, Mark I.-C. Chen, Yee-Sin Leo, David C. Lye, and Lin-Fa Wang. Pan-Sarbecovirus Neutralizing Antibodies in BNT162b2-Immunized SARS-CoV-1 Survivors. *New England Journal of Medicine*, 385(15):1401–1406, 10 2021.

[12] Yi-Chun Chen, Sheng-Nan Lu, Huey-Ling You, Chih-Chi Wang, and Ing-Kit Lee. SARS-CoV-2 Antibody Response After ChAdOx1 nCoV-19 Vaccination in Persons With Previous SARS-CoV-1 Infection. *JAMA Internal Medicine*, 182(3):347, 3 2022.

[13] Kevin O. Saunders, Esther Lee, Robert Parks, David R. Martinez, Dapeng Li, Haiyan Chen, Robert J. Edwards, Sophie Gobeil, Maggie Barr, Katayoun Mansouri, S. Munir Alam, Laura L. Sutherland, Fangping Cai, Aja M. Sanzone, Madison Berry, Kartik Manne, Kevin W. Bock, Mahnaz Minai, Bianca M. Nagata, Anyway B. Kapingidza, Mihai Azoitei, Longping V. Tse, Trevor D. Scobey, Rachel L. Spreng, R. Wes Rountree, C. Todd DeMarco, Thomas N. Denny, Christopher W. Woods, Elizabeth W. Petzold, Juanjie Tang, Thomas H. Oguin, Gregory D. Sempowski, Matthew Gagne, Daniel C. Douek, Mark A. Tomai, Christopher B. Fox, Robert Seder, Kevin Wiehe, Drew Weissman, Norbert Pardi, Hana Golding, Surender Khurana, Priyamvada Acharya, Hanne Andersen, Mark G. Lewis, Ian N. Moore, David C. Montefiori, Ralph S. Baric, and Barton F. Haynes. Neutralizing antibody vaccine for pandemic and pre-emergent coronaviruses. *Nature*, 594(7864):553–559, 6 2021.

[14] David R. Martinez, Alexandra Schäfer, Sarah R. Leist, Gabriela De la Cruz, Ande West, Elena N. Atochina-Vasserman, Lisa C. Lindesmith, Norbert Pardi, Robert Parks, Maggie Barr, Dapeng Li, Boyd Yount, Kevin O. Saunders, Drew Weissman, Barton F. Haynes, Stephanie A. Montgomery, and Ralph S. Baric. Chimeric spike mRNA vaccines protect against Sarbecovirus challenge in mice. *Science*, 373(6558):991–998, 8 2021.

[15] Lianpan Dai, Tianyi Zheng, Kun Xu, Yuxuan Han, Lili Xu, Enqi Huang, Yaling An, Yingjie Cheng, Shihua Li, Mei Liu, Mi Yang, Yan Li, Huijun Cheng, Yuan Yuan, Wei Zhang, Changwen Ke, Gary Wong, Jianxun Qi, Chuan Qin, Jinghua Yan, and George F. Gao. A Universal Design of Betacoronavirus Vaccines against COVID-19, MERS, and SARS. *Cell*, 182(3):722–733, 8 2020.

[16] Alexandra C. Walls, Marcos C. Miranda, Minh N. Pham, Alexandra Schäfer, Allison Greaney, Prabhu S. Arunachalam, Mary-Jane Navarro, M. Alejandra Tortorici, Kenneth Rogers, Megan A. O'Connor, Lisa Shireff, Douglas E. Ferrell, Natalie Brunette, Elizabeth Kepl, John Bowen, Samantha K. Zepeda, Tyler Starr, Ching-Lin Hsieh, Brooke Fiala, Samuel Wrenn, Deleah Pettie, Claire Sydeman, Max Johnson, Alyssa Blackstone, Rashmi Ravichandran, Cassandra Ogohara, Lauren Carter, Sasha W. Tilles, Rino Rappuoli, Derek T. O'Hagan, Robbert Van Der Most, Wesley C. Van Voorhis, Jason S. McLellan, Harry Kleanthous, Timothy P. Sheahan, Deborah H. Fuller, Francois Villinger, Jesse Bloom, Bali Pulendran, Ralph Baric, Neil King, and David Veesler. Elicitation of broadly protective sarbecovirus immunity by receptor-binding domain nanoparticle vaccines. *bioRxiv*, page 2021.03.15.435528, 3 2021.

[17] Alexander A. Cohen, Priyanthi N. P. Gnanapragasam, Yu E. Lee, Pauline R. Hoffman, Susan Ou, Leesa M. Kakutani, Jennifer R. Keeffe, Hung-Jen Wu, Mark Howarth, Anthony P. West, Christopher O. Barnes, Michel C. Nussenzweig, and Pamela J. Bjorkman. Mosaic nanoparticles elicit cross-reactive immune responses to zoonotic coronaviruses in mice. *Science*, 371(6530):735–741, 2 2021.

[18] Alba Grifoni, Daniela Weiskopf, Sydney I. Ramirez, Jose Mateus, Jennifer M. Dan, Carolyn Rydyznski Moderbacher, Stephen A. Rawlings, Aaron Sutherland, Lakshmanane Premkumar, Ramesh S. Jadi, Daniel Marrama,

Aravinda M. de Silva, April Frazier, Aaron F. Carlin, Jason A. Greenbaum, Bjoern Peters, Florian Krammer, Davey M. Smith, Shane Crotty, and Alessandro Sette. Targets of T Cell Responses to SARS-CoV-2 Coronavirus in Humans with COVID-19 Disease and Unexposed Individuals. *Cell*, 181(7):1489–1501, 6 2020.

[19] Alessandro Sette, Magdalini Moutaftsi, Juan Moyron-Quiroz, Megan M. McCausland, D. Huw Davies, Robert J. Johnston, Bjoern Peters, Mohammed Rafii-El-Idrissi Benhnia, Julia Hoffmann, Hua Poo Su, Kavita Singh, David N. Garboczi, Steven Head, Howard Grey, Philip L. Felgner, and Shane Crotty. Selective CD4+ T Cell Help for Antibody Responses to a Large Viral Pathogen: Deterministic Linkage of Specificities. *Immunity*, 28(6):847–858, 6 2008.

[20] Conan K. Wang and David J. Craik. Linking molecular evolution to molecular grafting. *Journal of Biological Chemistry*, 296:100425, 1 2021.

[21] Joseph M. Perchiacca, Ali Reza A. Ladiwala, Moumita Bhattacharya, and Peter M. Tessier. Structure-based design of conformation- and sequence-specific antibodies against amyloid β. *Proceedings of the National Academy of Sciences*, 109(1):84–89, 1 2012.

[22] Ali Reza A. Ladiwala, Moumita Bhattacharya, Joseph M. Perchiacca, Ping Cao, Daniel P. Raleigh, Andisheh Abedini, Ann Marie Schmidt, Jobin Varkey, Ralf Langen, and Peter M. Tessier. Rational design of potent domain antibody inhibitors of amyloid fibril assembly. *Proceedings of the National Academy of Sciences*, 109(49):19965–19970, 12 2012.

[23] Pietro Sormanni, Francesco A. Aprile, and Michele Vendruscolo. Rational design of antibodies targeting specific epitopes within intrinsically disordered proteins. *Proceedings of the National Academy of Sciences*, 112(32):9902–9907, 8 2015.

[24] Emiko Mihara, Satoshi Watanabe, Nasir K. Bashiruddin, Nozomi Nakamura, Kyoko Matoba, Yumi Sano, Rumit Maini, Yizhen Yin, Katsuya Sakai, Takao Arimori, Kunio Matsumoto, Hiroaki Suga, and Junichi Takagi. Lasso-grafting of macrocyclic peptide pharmacophores yields multi-functional proteins. *Nature Communications 2021 12:1*, 12(1):1–12, 3 2021.

[25] Samuel K. Sia and Peter S. Kim. Protein grafting of an HIV-1-inhibiting epitope. *Proceedings of the National Academy of Sciences of the United States of America*, 100(17):9756–9761, 8 2003.

[26] Janice S. Blum, Pamela A. Wearsch, and Peter Cresswell. Pathways of Antigen Processing. *Annual Review of Immunology*, 31(1):443–473, 3 2013.

[27] Christian Bahne Thygesen, Ahmad Salim Al-Sibahi, Lys Sanz Moreta, Christian Skjødt Steenmans, Anders Bundgård Sørensen, and Thomas W Hamelryck. Efficient Generative Modelling of Protein Structure Fragments using a Deep Markov Model. *Proceedings of the 38th International Conference on Machine Learning*, PMLR139:10258–10267, 2021.

[28] L L C Schrödinger and Warren DeLano. PyMOL.

[29] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes, 12 2013.

[30] Randi Vita, Swapnil Mahajan, James A Overton, Sandeep Kumar Dhanda, Sheridan Martini, Jason R Cantrell, Daniel K Wheeler, Alessandro Sette, and Bjoern Peters. The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Research*, 47(D1):D339–D343, 1 2019.

[31] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods 2011 9:2*, 9(2):173–175, 12 2011.

[32] Sean R. Eddy. Accelerated Profile HMM Searches. *PLOS Computational Biology*, 7(10):e1002195, 10 2011.

[33] Laura E. Johnson, Thomas P. Frye, Alana R. Arnot, Carrie Marquette, Larry A. Couture, Annette Gendron-Fitzpatrick, and Douglas G. McNeel. Safety and immunological efficacy of a prostate cancer plasmid DNA vaccine encoding prostatic acid phosphatase (PAP). *Vaccine*, 24(3):293–303, 1 2006.

[34] Guido Papa, Donna L. Mallery, Anna Albecka, Lawrence G. Welch, Jérôme Cattin-Ortolá, Jakub Luptak, David Paul, Harvey T. McMahon, Ian G. Goodfellow, Andrew Carter, Sean Munro, and Leo C. James. Furin cleavage of SARS-CoV-2 Spike promotes but is not essential for infection and cell-cell fusion. *PLOS Pathogens*, 17(1):e1009246, 1 2021.

[35] Jesper Pallesen, Nianshuang Wang, Kizzmekia S. Corbett, Daniel Wrapp, Robert N. Kirchdoerfer, Hannah L. Turner, Christopher A. Cottrell, Michelle M. Becker, Lingshu Wang, Wei Shi, Wing-Pui Kong, Erica L. Andres, Arminja N. Kettenbach, Mark R. Denison, James D. Chappell, Barney S. Graham, Andrew B. Ward, and Jason S. McLellan. Immunogenicity and structures of a rationally designed prefusion MERS-CoV spike antigen. *Proceedings of the National Academy of Sciences*, 114(35), 8 2017.

[36] Robert N. Kirchdoerfer, Nianshuang Wang, Jesper Pallesen, Daniel Wrapp, Hannah L. Turner, Christopher A. Cottrell, Kizzmekia S. Corbett, Barney S. Graham, Jason S. McLellan, and Andrew B. Ward. Stabilized coronavirus spikes are resistant to conformational changes induced by receptor recognition or proteolysis. *Scientific Reports 2018 8:1*, 8(1):1–11, 10 2018.

[37] Gunnstein Norheim, Elisabeth Stubsrud, Lise Madelene Skullerud, Branislava Stankovic, Stalin Chellappa, Louise Bjerkan, Katarzyna Kuczkowska, Elisabeth Müller, Monika Sekelja, and Agnete B. Fredriksen. Single dose immunization with a covid-19 dna vaccine encoding a chimeric homodimeric protein targeting receptor binding domain (rbd) to antigen-presenting cells induces rapid, strong and long-lasting neutralizing igg, th1 dominated cd4+ t cells and strong cd8+ t cell responses in mice. *bioRxiv*, page 2020.12.08.416875, 12 2020.

[38] Sarah Güthe, Larisa Kapinos, Andreas Möglich, Sebastian Meier, Stephan Grzesiek, and Thomas Kiefhaber. Very fast folding and association of a trimerization domain from bacteriophage T4 fibritin. *Journal of molecular biology*, 337(4):905–915, 4 2004.

[39] Sebastian Meier, Sarah Güthe, Thomas Kiefhaber, and Stephan Grzesiek. Foldon, the natural trimerization domain of T4 fibritin, dissociates into a monomeric A-state form containing a stable beta-hairpin: atomic details of trimer dissociation and local beta-hairpin stability from residual dipolar couplings. *Journal of molecular biology*, 344(4):1051–1069, 12 2004.

[40] Kazutaka Katoh and Daron M. Standley. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution*, 30(4):772–780, 4 2013.

[41] Lam-Tung Nguyen, Heiko A Schmidt, Arndt Von Haeseler, and Bui Quang Minh. Iq-tree: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular biology and evolution*, 32(1):268–274, 2015.

5 Supplementary material

5.1 Identification of similarity threshold

The entire pool of IEDB epitopes mapped against each position in the scaffold protein. Exact amino acid sequence matches were excluded. The resulting distribution over KL divergence is shown in figure S1. A small peak was observed below a KL divergence of 13. Thus, we set a conservative threshold for considering a peptide and graft site as matches as having a KLD below 13. This ruled out $\sim 99\%$ of peptide/graft site pairs.

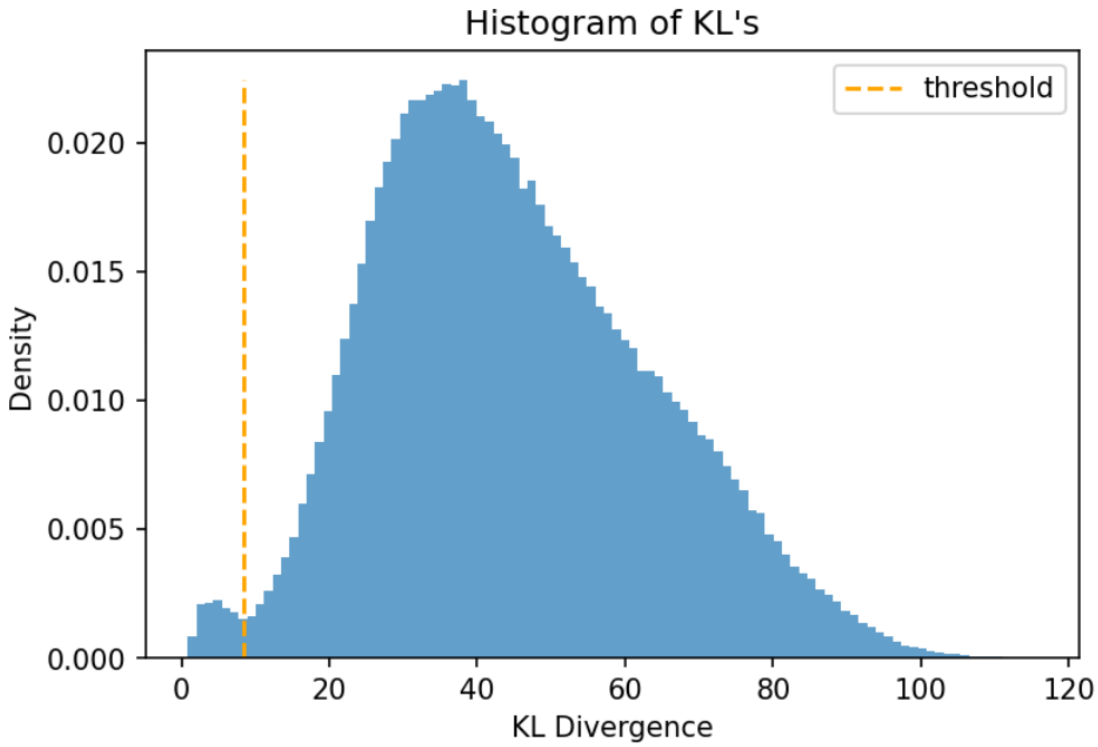


Figure S1: Distribution of KL divergences between every verified C57BL/6 IEDB epitopes and every overlapping position of the spike protein. The decision boundary for classifying peptides as graft matches is shown with the orange dashed line.

5.2 Grafted epitopes

Table S1: List of epitopes grafted with the BIFROST and MSA approach. When relevant, IEDB epitope IDs are listed.

Epitope	IEDB id	Wildtype sequence	Approach
FQDAYNAAGGHNAVF	17460	IIAYTMSLGAENSVA	BIFROST
EQQWNFAGIEAAASA	161623	QLNRALTGIAVEQDK	BIFROST
NEKYAQAYPNVS	43662	QVKQIYKTPPIK	BIFROST
LLTKKQYDKAQASFQ	225401	LLTDEMIAQYTSALL	BIFROST
GYLYIYPSAGNSFDL	23407	GAALQIPFAMQMAYR	BIFROST
HYLVNHPEVLVEASQ	143938	TQNVLYENQKLIANQ	BIFROST
DVNYGYNAATGEYGD	225263	DSLSSTASALGKLQD	BIFROST
VQNRFNSAITNLGNT	70619	LNTLVKQLSSNFGAI	BIFROST
HYFDPKVIPSI	736871	QEKNFTTAPAI	BIFROST
DIYKGVYQFKSV	175783	FVSNGTHWFVTQ	BIFROST
NGVAAPSATSQ	933287	RNFYEPQIITT	BIFROST
VAMIWSVAAVAQTVG	225558	QKEIDRLNEVAKNLN	BIFROST
KNSFAYSNNSIAIPT	N/A	ENSVAYSNNSIAIPT	MSA
TITVTTKIPPASKTK	N/A	TISVTTEILPVSMTK	MSA
LLQYRSSCTQLTRAL	N/A	LLQYGSFCTQLNRAL	MSA
VKPIYKAPPVKDFAG	N/A	VKQIYKTPPIKDFGG	MSA
FPQFFPDPSKPSKRS	N/A	FSQILPDPSKPSKRS	MSA
TSGWTFAAGAALQTP	N/A	TSGWTFGAGAALQIP	MSA
ENQKFIANQFNSAIG	N/A	ENQKLIANQFNSAIG	MSA
QDSFSSTASALGKLQ	N/A	QDSLSSTASALGKLQ	MSA
RPAEIRASANLAATK	N/A	RAAEIRASANLAATK	MSA
GKGYYLMSFPQSAPH	N/A	GKGYHLMSFPQSAPH	MSA
EKNFTTAPAICHDRK	N/A	EKNFTTAPAICHDGK	MSA
SPDVYLGDISGINAS	N/A	SPDVDLGDISGINAS	MSA

Chapter 7

A multiscale deep generative model of protein structure using a directional and a Procrustes likelihood

Manuscript 3 presents a further development of the model described in manuscript 1 [85]. The model in manuscript 1 is majorly limited in performance when the length of the amino acid sequence increases. This is mainly due to the fact, that the model operates purely on internal coordinates, i.e. the dihedral angles of the backbone of the amino acid polymer as described in section 2. When focusing on internal coordinates we introduce a risk of observing "elbow effects". Elbow effects occur when transitioning from internal to Cartesian coordinates, where the 3D-coordinates of the protein backbone are iteratively inferred based on a set of idealized bond lengths and the modelled dihedral angles. Suppose there is a small error in prediction in the dihedral angles of the center-most amino acid. This error may be small in the internal coordinate space but it will propagate through the downstream amino acids causing larger errors in Cartesian space.

The model presented in the following manuscript contains a few improvements on the model from manuscript 1. We introduce a more fitting distribution representing the likelihood over dihedral angles. Additionally, we perform an ablation study to select the best sequence model to represent the amino acid sequence. Finally, we will address the elbow effect by combining the BIFROST model with a probabilistic model of protein superposition - Theseus [86, 87]. Theseus will serve as a likelihood over Cartesian coordinates as well, providing an additional error function to the modelling that takes into account the differences between the 3D-coordinates generated from predicted angles and those of the observed protein structure.

A MULTISCALE DEEP GENERATIVE MODEL OF PROTEIN STRUCTURE USING A DIRECTIONAL AND A PROCRUSTES LIKELIHOOD

A PREPRINT

Christian B. Thygesen[*]
Department of Computer Science
Copenhagen University &
Evaxion Biotech A/S
ckt@evaxion-biotech.com

Ola Rønning
Department of Computer Science
Copenhagen University
ola@di.ku.dk

Christian Skjødt Steenmans
Evaxion Biotech A/S
csh@evaxion-biotech.com

Anders Bundgård Sørensen
Evaxion Biotech A/S
abs@evaxion-biotech.com

Kanti V. Mardia
Department of Statistics
School of Mathematics
University of Leeds
K.V.Mardia@leeds.ac.uk

John T. Kent
Department of Statistics
School of Mathematics
University of Leeds
j.t.kent@leeds.ac.uk

Thomas Hamelryck
Department of Computer Science &
Department of Biology
Copenhagen University
thamelry@bio.ku.dk

September 28, 2022

ABSTRACT

Protein structure prediction has become a paradigm problem in machine learning. Recent methods based on deep learning such as Alphafold [1] and trRosetta [2] have revolutionized the field. However, open problems remain, including modelling of protein folding and dynamics, asessing the impact of mutations and separating aleatory from epistemic uncertainty. These open problems can benefit from a probabilistic approach based on a deep generative model. Due to the presence of rotations as nuisance parameters, probabilistic models of protein structure typically use rotation invariant representations such as dihedral angles or pairwise atom distances instead of 3D coordinates. By combining a deep Markov model with a multiscale likelihood, we can include both dihedral angles and 3D-coordinates. We use the bivariate sine von Mises likelihood for the former, and a Procrustes likelihood that features latent rotations and latent variances of the 3D positions for the latter. The trained model allows a highly efficient prediction-by-sampling regime. We show that the multiscale nature of the model results in better predictions of both dihedral angles and 3D coordinates, reducing the root mean square deviation from 10 Å to 5 Å for protein fragments of length 30. Finally, we briefly discuss how to extend the model to entire proteins.

Keywords Deep Probabilistic Programming · Protein structure · deep Markov model · multiscale modelling

[*]Corresponding author

1 Introduction

In the last few years, the problem of protein structure prediction has seen an increased amount of attention from the machine learning community due to the dramatic improvements shown by deep learning methods such as Alphafold [1], trRosetta [2], and more recently Omegafold [3]. However, studies also showed that their reliance on multiple sequence alignments makes it difficult to predict the effect of mutations [4]. In addition, predicting protein dynamics or the folding process itself remains a challenge [5, 6]. This indicates that the problem is not entirely solved, and that there is room for models that work on the single-sequence level. Probabilistic models of proteins, and more specifically deep generative models can play a role here [7, 8, 9, 10, 11, 12, 13, 14].

Protein structures are generally represented using either internal coordinates, pairwise distances or 3D-coordinates. 3D-coordinates represent the protein structure as a set of Cartesian coordinates of the individual atoms. Thus, they are a high-resolution representation of a protein structure. However, 3D-coordinates are challenging in a machine learning setting, as they can be arbitrarily translated and rotated. Correspondingly, when comparing two sets of 3D-coordinates, we need to account for rotational and translational degrees of freedom. This is done by so-called Procrustes models [15, 16].

There are two ways of avoiding this problem. One is to encode the structure as pairwise amino acid distances. However, pairwise distances are high-dimensional ($L \times L$) and do not account for mirror reflection effects. Alternatively, internal coordinates describe protein structure through bond lengths, bond angles, and backbone dihedral angles. Backbone dihedral angles are the torsion angles associated with the bonds between the N, C_α, and C atoms of the amino acid backbone (figure 1), whereas bond lengths are the lengths of the bonds between these atoms. Current state of the

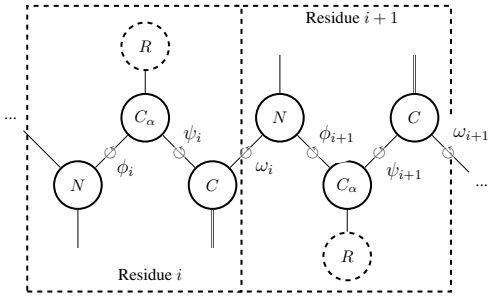


Figure 1: Schematic of the dihedral angles of the protein backbone.

art approaches address the rotation-invariance problem through parameter-heavy rotation-equivariant transformer architectures, making training and inference troublesome without high performance computing setups [2, 1]. Cartesian coordinates can be reconstructed from internal coordinates [17, 18], often assuming idealised values for bond lengths and bond angles. Sequential reconstruction leads to an "elbow-effect": prediction errors early in the sequence of angles lead to large errors in the reconstructed 3D coordinates. Consequently, when representing protein chains, small variances in dihedral angle space may result in large variances in Cartesian coordinate space.

Probabilistic models of proteins structure can be formulated using hidden Markov models [7, 8, 9, 10, 11] or, more recently, deep Markov models [19] that represent sequences of dihedral angles using directional statistics. In addition, VAEs have also been used for this purpose, typically representing a matrix of pairwise distances instead of a sequence of dihedral angles [12, 13, 14]. Here, we combine a deep Markov model with a multiscale likelihood, representing both dihedral angles and 3D coordinates. We show that the resulting deep generative model allows for efficient prediction-by-sampling mediated by dihedral angles, while to a great extent ameliorating the elbow problem.

For the likelihood over the angles, we make use of a probability distribution on the torus, as previously described [19], though we make use of a new distrubution, the sine bivariate von Mises, as described in section 2.3. For the likelihood over the 3D coordinates, we make use of a Procrustes model [20, 15, 16], which following [15] we will call Theseus. The original Theseus model assumes that two protein structures are noisy observations of a latent mean structure, and accounts for the rotation and translation of the structures to infer optimal superposition (figure 2, left). Theseus is a heteroscedastic model that accounts for different variances of the atomic positions. We adapt the model for use as the likelihood of the predicted structure, $\boldsymbol{M}$, given the observed structure, $\boldsymbol{X}$ (see figure 2, right). As we (per construction) can assume that both coordinate sets are centered at $(0, 0, 0)$, the Theseus likelihood introduces two latent variables: a rotation and a vector of variances of the atom positions.

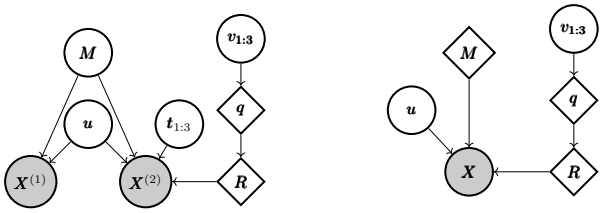


Figure 2: Graphical models illustrating the principles of the Theseus superposition framework [16, 15]. Left: The Theseus model used to superimpose two structures. $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ are $N \times 3 \in \mathbb{R}$ matrices containing the coordinates of the structures to be superimposed. Both $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ are interpreted as noisy observations (controlled by a vector of N variances $\boldsymbol{u}$) of a common underlying latent, mean structure, $\boldsymbol{M}$ (also an $N \times 3 \in \mathbb{R}$ matrix). In addition, one of the structures is rotated (rotation matrix $\boldsymbol{R}$) and translated (vector $\boldsymbol{t}$) with respect to $\boldsymbol{M}$. $\boldsymbol{v}$ and $\boldsymbol{q}$ are quantities used to construct a uniform prior over rotations (see section 2.2). The priors $p(\boldsymbol{t})$ and $p(\boldsymbol{u})$ are normal and half-normal, respectively. $p(\boldsymbol{M})$, $p(\boldsymbol{X}^{(1)} \mid \boldsymbol{M}, \boldsymbol{u})$ and $p(\boldsymbol{X}^{(2)} \mid \boldsymbol{M}, \boldsymbol{u}, \boldsymbol{R}, \boldsymbol{t})$ are interpreted as matrix-normal distributions [15]. Right: the Theseus variant used as likelihood for the 3D coordinates in the Bifrost-Theseus model (see Algorithm 1). In this case, $\boldsymbol{M}$ is a prediction from a neural network and there is only one structure, $\boldsymbol{X}$, abolishing the need for the translation, $\boldsymbol{t}$. $p(\boldsymbol{X} \mid \boldsymbol{M}, \boldsymbol{R}, \boldsymbol{u})$ formulates as a product of multivariate normal distributions.

Multiscale approaches are used for modelling the same phenomena at different levels, or scales, where the different levels affect each other [21], such as in the modelling of nanocomposite materials at both the molecular level as well as how they interact [22]. By combining a deep Markov model [23, 19] model with a Procrustes and an angular likelihood, we can formulate a multiscale model of protein structure.

2 Methods

2.1 Data

We trained the model on the data from [19], consisting of protein fragments derived from $3,733$ protein structures. This data set consisted of a non-redundant, high-resolution set of protein structures from the cullpdb data set [24]. For further details on the data set, see [19]. Three data sets were created, by splitting the protein structures into fragments of 9, 15, and 30 amino acids. We use redundancy reduction, such that no fragments share more than 20% sequence identity.

2.2 Deep Markov model and directional likelihood

Our model is an extension on the deep Markov model (DMM) of local protein structure, Bifrost [19]. The DMM consists of multiple neural networks parameterising latent distributions and distributions over dihedrals. A transition neural network parameterises a multivariate Gaussian distribution over the latent variables at each position in the fragment. An emitter neural networks takes the latent variable as input and parameterises a Sine Bivariate von Mises (SBVM) distribution [25] over the observed angles. This is a fundamental change from the original Bifrost model which used a wrapped student T (Cauchy) distribution for modelling dihedral angles. The SBVM allows us to model the correlation between the ϕ/ψ dihedral angle pairs, as opposed to treating them as independent random variables. A simple graphical representation of the Bifrost model is shown in figure 3.

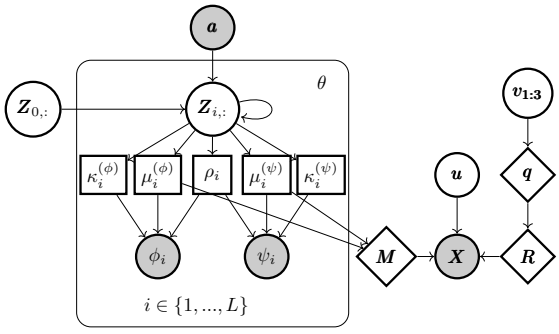


Figure 3: Graphical model of Bifrost-Theseus. Grey nodes are observed variables, white circular nodes are latent variables, white rectangular nodes represent parameters of the SBVM distribution. Diamond-shaped and rectangular nodes are deterministic variables. The square with rounded corners represent a plate, marking conditional independence. Nodes to the right of the plate correspond to variables originating from adding the Theseus layer to the Bifrost model. For details, we refer to Algorithm 1.

2.3 Sine Bivariate von Mises distribution

Each internal coordinate naturally embeds on the 2-torus. To quantify uncertainty in the internal coordinate representation we use a submodel of the Bivariate von Mises [26], known as the sine model [27, 25]. The probability density function of the sine model is given by

$$f(\phi,\psi|\boldsymbol{\mu},\boldsymbol{\kappa},\rho) = C^{-1}\exp(\kappa^{(\phi)}\cos(\phi-\mu^{(\phi)}) + \kappa^{(\psi)}\cos(\psi-\mu^{(\psi)}) + \rho\sin(\phi-\mu^{(\phi)})\sin(\psi-\mu^{(\psi)})), \quad (1)$$

where the $\rho > 0$ parameters control statistical dependence between the angles $\phi, \psi \in [-\pi, \pi)$, $\boldsymbol{\kappa} > 0$ is the concentration, and $\boldsymbol{\mu} \in [-\pi, \pi)^2$ is the point of (toroidal) symmetry for the distribution. The normalization constant C is given by the infinite series

$$\frac{1}{C} = (2\pi)^2 \sum_{i=0}^{\infty} \binom{2i}{i} \left(\frac{\rho^2}{4\kappa^{(\phi)}\kappa^{(\psi)}}\right)^2 I_i(\kappa^{(\phi)})I_i(\kappa^{(\psi)}), \quad (2)$$

where I_i is the $i-$th order modified Bessel function of the first kind.

The sine model is a generalization of the wrapped Cauchy likelihood in [19]. To see this, observe that Equation (1) at $\rho = 0$ corresponds to two independent von Mises distributions, each of which share support with the wrapped Cauchy distribution.

The SBVM distribution is unimodal if and only if $\kappa^{(\phi)}\kappa^{(\psi)} > \rho^2$, otherwise the model is bimodal. Due to this bimodality, sampling from the SBVM can become inefficient near the boundary $\kappa^{(\phi)}\kappa^{(\psi)} = \rho^2$. To overcome this instability, we use a scaled correlation parameter η in place of ρ given by $\rho = \eta\sqrt{\kappa^{(\phi)}\kappa^{(\psi)}}$ where $\eta \in [0, 1]$.

2.4 Procrustes likelihood

We expanded the above described model by introducing a likelihood based on the 3D coordinates. The DMM generates latent representations $\boldsymbol{Z}$, from which we predict the means, concentrations and correlations that parameterise the distribution over the backbone dihedrals. We reconstruct the 3D-coordinates from the predicted means of the dihedral angles, $\boldsymbol{\mu}^{(\phi)}$ and $\boldsymbol{\mu}^{(\psi)}$, using the PNERF algorithm [17, 18, 28]. We assume idealised values for bond lengths and bond angles, as well as trans peptides bonds ($\omega = \pi$). See table S1 for exact idealized values. We treat the reconstructed structure as the latent mean structure ($\boldsymbol{M}$) in the Theseus framework, allowing us to evaluate the likelihood that observed coordinates originate from the distribution defined by this latent mean.
Note that the predicted dihedral angle means could be interpreted as a "denoised" version of the observed dihedral angles, as the reconstructed 3D coordinates make use of ideal bond angles and bond lengths. Thus we can distinguish between aleatory (reconstructed 3D coordinates, $\boldsymbol{M}$) and epistemic uncertainty (observed dihedral angles, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, and coordinates, $\boldsymbol{X}$).
The Theseus likelihood introduces two latent variables per protein structure, a rotation, $\boldsymbol{R}$ and a vector of variances of atom positions, $\boldsymbol{u}$. A uniform prior over rotations is constructed making use of unit quaternions, following [29]. We sample three random variables from a uniform prior on the unit interval. From these, we calculate four deterministic variables defining the unit quaternion ($\boldsymbol{q} = (w, x, y, z)$) (equation 3).

$$v_i \sim U(0,1), i \in 1,2,3 \quad (3a)$$
$$\theta_1 = 2\pi v_2; \theta_2 = 2\pi v_3 \quad (3b)$$
$$r_1 = \sqrt{1-v_1}; r_2 = \sqrt{v_1} \quad (3c)$$
$$\boldsymbol{q} = (w,x,y,z) = (r_2\cos\theta_2, r_1\sin\theta_1, r_1\cos\theta_1, r_2\sin\theta_2) \quad (3d)$$

Finally, from the unit quaternion, a rotation matrix is constructed (equation 4).

$$\boldsymbol{R} = \begin{bmatrix} w^2+x^2-y^2-z^2 & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & w^2-x^2+y^2-z^2 & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & w^2-x^2-y^2+z^2 \end{bmatrix} \quad (4)$$

We treat the reconstructed 3D-coordinates as the underlying mean. We then evaluate the likelihood, that observed coordinates were drawn from a multivariate Gaussian with means at the reconstructed coordinates and a per-datapoint modelled variance. We assume conditional independence between the atoms, so it is sufficient to approximate this matrix normal with a multivariate Gaussian. The variances are drawn from a half-normal distribution.

2.5 Conditional joint distribution

We can express the conditional joint distribution of dihedrals and 3D coordinates given the amino acid sequence as

$$p_{\boldsymbol{\theta}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{X}|\boldsymbol{a}) = \int_{\boldsymbol{Z}} \int_{\boldsymbol{R}} \int_{\boldsymbol{u}} p_{\boldsymbol{\theta}}(\boldsymbol{\phi}, \boldsymbol{\psi}|\boldsymbol{Z}) p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{R}, \text{diag}(\boldsymbol{u})) p(\boldsymbol{u}) p(\boldsymbol{R}) p_{\boldsymbol{\theta}}(\boldsymbol{Z}|\boldsymbol{a}) \, \text{d}\boldsymbol{u} \, \text{d}\boldsymbol{R} \, \text{d}\boldsymbol{Z}, \tag{5}$$

where $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are observed backbone dihedral angles, $\boldsymbol{X}$ represents observed 3D coordinates, $\boldsymbol{a}$ is the amino acid sequence, $\boldsymbol{Z}$ is an $L \times 40$ matrix containing the L latent representations, $\text{diag}(\boldsymbol{u})$ is a diagonal covariance matrix containing the variances of the atom positions, and $\boldsymbol{R}$ is a rotation matrix. The latter two are part of the Procrustes submodel, Theseus. Neural networks with parameters $\boldsymbol{\theta}$ parameterize the conditional distributions in the above. For more details, we refer to the graphical model shown in figure 3 and algorithm 1.

Algorithm 1 The Bifrost-Theseus model. $\boldsymbol{a}$: Amino acid sequence. L: Length of the amino acid sequence. T and E: Transition and emission network of the deep Markov model, respectively. $\mathcal{SBVM}$: Sine Bivariate von Mises distribution. PNERF: Algorithm for reconstructing the 3D-coordinates from a vector of dihedral angles. $\mathcal{N}_+$: Half-normal distribution. $\boldsymbol{Z}$: $L \times 40$ matrix containing the latent vector of the deep Markov model for each amino acid position. $\boldsymbol{X}$ and $\boldsymbol{M}$: $L \times 3 \times 3$ tensors containing 3D coordinates; first dimension: amino acid index; second dimension: atom index (corresponding to C, C_α, N); third dimension: 3D coordinates. $\boldsymbol{X}$ and $\boldsymbol{M}$ contain the coordinates of the observed and the predicted structure, respectively. Both $\boldsymbol{X}$ and $\boldsymbol{M}$ are centered at $(0, 0, 0)$, thus only requiring a rotation matrix $\boldsymbol{R}$ for their superposition used in the calculation of the likelihood. $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are vectors containing the observed dihedral angles.

Input: $\boldsymbol{a} = [a_1, ..., a_L], \boldsymbol{X}, \boldsymbol{\phi}, \boldsymbol{\psi}$ ▷ Amino acid sequence, 3D coordinates, dihedral angle vectors
$\boldsymbol{H} \leftarrow \text{GRU}(\boldsymbol{a})$ ▷ $L \times 100$ matrix of GRU states
$\boldsymbol{Z}_{0,:} \sim \mathcal{MVN}\left(\boldsymbol{\mu}_{0,:}^{(Z)}, \text{diag}\left(\boldsymbol{\sigma}_{0,:}^{(Z)}\right)\right)$ ▷ Latent vector to start the Markov chain
for $i = 1$ to L **do** ▷ Loop over the L amino acids
  $\boldsymbol{\mu}_{i,:}^{(Z)}, \boldsymbol{\sigma}_{i,:}^{(Z)} \leftarrow T(\boldsymbol{Z}_{i-1,:}, \boldsymbol{H}_{i,:})$ ▷ Mean and variances for latent vector at position i
  $\boldsymbol{Z}_{i,:} \sim \mathcal{MVN}\left(\boldsymbol{\mu}_{i,:}^{(Z)}, \text{diag}\left(\boldsymbol{\sigma}_{i,:}^{(Z)}\right)\right)$ ▷ Sample latent vector at position i
  $\mu_i^{(\phi)}, \mu_i^{(\psi)}, \kappa_i^{(\phi)}, \kappa_i^{(\psi)}, \rho_i \leftarrow E(\boldsymbol{Z}_{i,:})$ ▷ Parameters of the SBVM distribution for position i
  $\phi_i, \psi_i \sim \mathcal{SBVM}\left(\mu_i^{(\phi)}, \mu_i^{(\psi)}, \kappa_i^{(\phi)}, \kappa_i^{(\psi)}, \rho_i\right)$ ▷ Likelihood of the dihedral angle pair at position i
$\boldsymbol{M} \leftarrow \text{PNERF}\left(\boldsymbol{\mu}^{(\phi)}, \boldsymbol{\mu}^{(\psi)}\right)$ ▷ Reconstruct the 3D coordinates from the SBVM means
for $j = 1$ to 3 **do**
  $v_j \sim \mathcal{U}(0, 1)$ ▷ $\boldsymbol{v}$ is used for the uniform prior over the rotation
$\boldsymbol{q} \leftarrow \text{Quaternion}(\boldsymbol{v})$ ▷ Quaternion representing the rotation matrix, see Equation 3
$\boldsymbol{R} \leftarrow \text{RotationMatrix}(\boldsymbol{q})$ ▷ Rotation matrix, see Equation 4
for $i = 1$ to L **do** ▷ Loop over L amino acids
  $u_i \sim \mathcal{N}_+(0.1)$
  $\boldsymbol{\Sigma}_i \leftarrow u_i \boldsymbol{I}_3$ ▷ Covariance matrix used for all atoms in amino acid i
  for $j = 1$ to 3 **do** ▷ Loop over $N, C\alpha, C$ atoms of amino acid i
    $\boldsymbol{M}_{i,j,:} \leftarrow \boldsymbol{M}_{i,j,:} - \overline{\boldsymbol{M}_{:,:,1:3}}$ ▷ Center coordinates of $\boldsymbol{M}$ at $(0, 0, 0)$
    $\boldsymbol{X}_{i,j,:} \sim \mathcal{MVN}(\boldsymbol{R}\boldsymbol{M}_{i,j,:}, \boldsymbol{\Sigma}_i)$ ▷ Likelihood of the 3D coordinates of atom j in amino acid i

2.6 Prediction-by-sampling

As we assume that the amino acid sequence is known, a computationally attractive capacity of the model is prediction-by-sampling using the following conditional distribution,

$$p_{\boldsymbol{\theta}}(\boldsymbol{\phi}, \boldsymbol{\psi}|\boldsymbol{a}) = \int_{\boldsymbol{Z}} p_{\boldsymbol{\theta}}(\boldsymbol{\phi}, \boldsymbol{\psi}|\boldsymbol{Z}) p_{\boldsymbol{\theta}}(\boldsymbol{Z}|\boldsymbol{a}) \, \text{d}\boldsymbol{Z}. \tag{6}$$

We only need the Theseus submodel for model training, not for prediction-by-sampling. Using the sampled angles or angle means we can reconstruct the predicted structure (algorithm 2).

Algorithm 2 Sampling 3D coordinates ($\boldsymbol{X}$) and dihedral angles ($\boldsymbol{\phi}, \boldsymbol{\psi}$) conditioned on the amino acid sequence ($\boldsymbol{a}$) from the Bifrost-Theseus model. For notation, see Algorithm 1.

Input: $\boldsymbol{a} = [aa_1, ..., aa_L]$ ▷ Amino acid sequence input
$\boldsymbol{H} \leftarrow \text{GRU}(\boldsymbol{a})$ ▷ $L \times 100$ matrix of GRU states
$\boldsymbol{Z}_{0,:} \sim \mathcal{MVN}\left(\boldsymbol{\mu}_{0,:}^{(Z)}, \text{diag}\left(\boldsymbol{\sigma}_{0,:}^{(Z)}\right)\right)$ ▷ Latent vector to start the Markov chain
for $i = 1$ to L **do** ▷ Loop over the L amino acids
  $\boldsymbol{\mu}_{i,:}^{(Z)}, \boldsymbol{\sigma}_{i,:}^{(Z)} \leftarrow T(\boldsymbol{Z}_{i-1,:}, \boldsymbol{H}_{i,:})$ ▷ Mean and variances for latent vector at position i
  $\boldsymbol{Z}_{i,:} \sim \mathcal{MVN}\left(\boldsymbol{\mu}_{i,:}^{(Z)}, \text{diag}\left(\boldsymbol{\sigma}_{i,:}^{(Z)}\right)\right)$ ▷ Sample latent vector at position i
  $\mu_i^{(\phi)}, \mu_i^{(\psi)}, \kappa_i^{(\phi)}, \kappa_i^{(\psi)}, \rho_i \leftarrow E\left(\boldsymbol{Z}_{i,:}\right)$ ▷ Parameters of the SBVM distribution for position i
  $\phi_i, \psi_i \sim \mathcal{SBVM}\left(\mu_i^{(\phi)}, \mu_i^{(\psi)}, \kappa_i^{(\phi)}, \kappa_i^{(\psi)}, \rho_i\right)$ ▷ Sample the dihedral angles at position i
$\boldsymbol{X} \leftarrow \text{PNERF}\left(\boldsymbol{\mu}^{(\phi)}, \boldsymbol{\mu}^{(\psi)}\right)$ ▷ Reconstruct the 3D coordinates from the SBVM means

2.7 Neural network architectures

Our DMM is a variant of [19]. The model and guide share the same overall topology, with a few key differences. Both the model and the guide employ a bidirectional recurrent neural network (RNN), parsing the amino acid sequence to produce hidden states. The model only uses the amino acids, while the guide also uses the observed dihedral angles. The latent states are parameterised by a neural network, that takes as input the previous latent variable along with the hidden state from the RNN at each time step. In the model we call this the transition network, while in the guide we call it the combiner network. The transition and combiner networks are identical in structure and are both Gated Recurrent Unit (GRU) [30] cells. The model employs one further neural network - the emitter - a feed forward network parameterising a Sine Bivariate von Mises distribution over dihedrals conditioned on the latent state. A rolled-out graphical representation of the model is shown in figure 4.

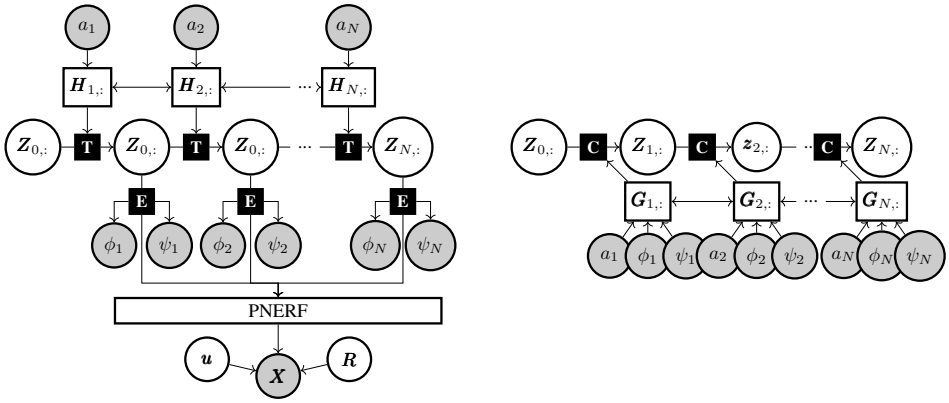


Figure 4: The Bifrost-Theseus model (left) and variational distribution (right). White circular nodes are latent random variables, grey nodes are observed variables, white rectangular nodes represent hidden states from a bidirectional GRU $\boldsymbol{H}$, and black squares represent feed forward neural networks. **E**, **T**, and **C** denote the emitter transition and combiner networks, respectively. For clarity, we do not show all random variables, and the deterministic and random variables pertaining to construction of the rotation matrix are collapsed to a single node, $\boldsymbol{R}$. For the full graphical model, refer to figure 3.

2.7.1 Training

For model estimation, we introduce a variational distribution, q, approximating the true posterior (algorithm 3). The variational distribution mimics the DMM behavior, inferring latent states from the amino acids and ϕ/ψ dihedral angles. Whereas the Bifrost-specific parameters are all amortised, the Theseus-specific parameters are not. Specifically, we use one pair of MAP estimates for the rotation $\boldsymbol{R}$ and variances $\boldsymbol{u}$ per fragment.

Algorithm 3 The Bifrost-Theseus variational distribution. $\delta(.)$ is the Dirac delta distribution and MAP indicates a maximum a posteriori point estimate.

Input: $\boldsymbol{a} = [a_1, ..., a_L], \boldsymbol{\phi}, \boldsymbol{\psi}$ ▷ Amino acid sequence, 3D coordinates, dihedral angle vectors
$\boldsymbol{G} \leftarrow \text{GRU}(\boldsymbol{a}, \boldsymbol{\phi}, \boldsymbol{\psi})$ ▷ $L \times 100$ matrix of GRU states
$\boldsymbol{\mu}_{0,:}^{(Z)} \sim \delta\left(\boldsymbol{\mu}_{0,:}^{(Z,\text{MAP})}\right)$ ▷ Mean and variances for latent vector at position 0
$\boldsymbol{\sigma}_{0,:}^{(Z)} \sim \delta\left(\boldsymbol{\sigma}_{0,:}^{(Z,\text{MAP})}\right)$
$\boldsymbol{Z}_{0,:} \sim \mathcal{MVN}\left(\boldsymbol{\mu}_{0,:}^{(Z)}, \text{diag}\left(\boldsymbol{\sigma}_{0,:}^{(Z)}\right)\right)$ ▷ Latent vector to start the Markov chain
for $i = 1$ to L **do** ▷ Loop over L amino acids
  $\boldsymbol{\mu}_{i,:}^{(Z)}, \boldsymbol{\sigma}_{i,:}^{(Z)} \leftarrow C(\boldsymbol{Z}_{i-1,:}, \boldsymbol{G}_{i,:})$ ▷ Mean and variances for latent vector at position i
  $\boldsymbol{Z}_{i,:} \sim \mathcal{MVN}\left(\boldsymbol{\mu}_{i,:}^{(Z)}, \text{diag}\left(\boldsymbol{\sigma}_{i,:}^{(Z)}\right)\right)$ ▷ Sample latent vector at position i
$\boldsymbol{u} \sim \delta\left(\boldsymbol{u}^{(\text{MAP})}\right)$ ▷ Variances of the atom positions
$\boldsymbol{v} \sim \delta\left(\boldsymbol{v}^{(\text{MAP})}\right)$ ▷ Rotation

Since an untrained Bifrost model will produce random stuctures, the error from the Theseus submodel is be unreasonably high early on, rendering training numerically unstable. Accordingly, we apply three training stages to estimate the final model. First, we train the Bifrost model without applying the Theseus error until convergence. For the Theseus submodel, we need to infer an optimal rotation for each fragment in the training data set. Hence, the second training pass employs a warm-up phase of the Theseus parameters, while freezing the Bifrost parameters. This phase uses the hyperparameters proposed in [16], i.e. an AdagradRMSprop optimizer [31]. After the warm-up-phase we unfreeze the Bifrost parameters, and start training the neural networks using both the dihedral angle and Theseus likelihoods. The Theseus warm-up phase consisted of 50 epochs, while the full training run was allowed to continue until convergence (table S2).

2.8 Model selection

We perform three levels of comparison for model architectures. The first level is the model's ability to recreate a Ramachandran plot. The Ramachandran number [32] summarizes ϕ/ψ pairs to a single number between 0 and 1 [32] (equation 7).

$$\mathcal{R}(\phi, \psi) = \frac{\phi + \psi + 2\pi}{4\pi} \tag{7}$$

We compare the distribution of inferred Ramachandran numbers to the observed distribution through the Jensen-Shannon divergence (JSD) [33], which is a symmetrical interpretation of the KL divergence (KLD) [34]. In practice, we turn generated Ramachandran numbers into histograms and compare to histograms of the observed Ramachandran numbers. We use JSD for this specific task for two reasons. Firstly, it provides a simple metric between 0 and 1, where 0 is maximally identical and 1 is maximally different. Secondly, JSD does not require absolute continuity, as KLD does. This means that even if the model generates histograms where certain bins are empty, JSD returns a real number, whereas KLD is not possible to compute. The second level of evaluation is the ability to recreate the dihedral angles of the specific protein structure, which we assess through RMSD between the predicted and observed dihedral angles. The third and final level assesses the quality of reconstructed 3D-coordinates compared to the observed 3D-coordinates in terms of RMSD between superimposed predicted and observed structures.

3 Results and discussion

3.1 Choice of directional distribution

The original Bifrost approach, modelled dihedral angles as a wrapped student T distribution. This distribution approximates the von Mises distribution with fatter tails [19]. However, this distribution assumes conditional independence of the ϕ/ψ dihedral angles, which may not hold. We replace the wrapped T with the Sine Bivariate von Mises distribution accounting for ϕ/ψ dependence [25]. Table 1 shows the performance of the original Bifrost with the new SBVM likelihood approach. The SBVM approach showed marginal improvement in terms of JSD of Ramachandran numbers, dihedral angle RMSD as well as 3D RMSD.

ϕ/ψ distribution	WST	SBVM
JSD	0.233	**0.202**
ϕ/ψ RMSD	0.782	**0.737**
3D RMSD	2.497	**2.227**

Table 1: Comparison of the impact of dihedral angle likelihoods in terms of JS divergence of Ramachandran number distributions (JSD) and median angular and 3D RMSDs. The metrics indicating best performance are highlighted in bold. WST: Wrapped student T, SBVM: Sine Bivariate von Mises.

3.2 Choice of sequence model

Bifrost uses an RNN to infer hidden states of the amino acid sequence (see figure 4). We investigated the impact of replacing this with more sophisticated networks, the gated recurrent unit (GRU) [30] and the long-short term memory (LSTM) [35] networks. The standard RNN showed the worst performance, while the GRU showed the best performance across all three metrics (table 2). The Ramachandran plots produced by the GRU were qualitatively similar to the observed Ramachandran plot, reducing the number of means predicted in the disallowed region in the middle of the Ramachandran plot (figure 5).

Sequence model	RNN	GRU	LSTM
JSD	0.202	**0.171**	0.182
ϕ/ψ RMSD	0.737	**0.657**	0.741
3D RMSD	2.227	**1.628**	1.736

Table 2: Sequence model comparison in terms of JS divergence of Ramachandran number distributions and median angular and 3D RMSD. The metrics indicating best performance are highlighted in bold.

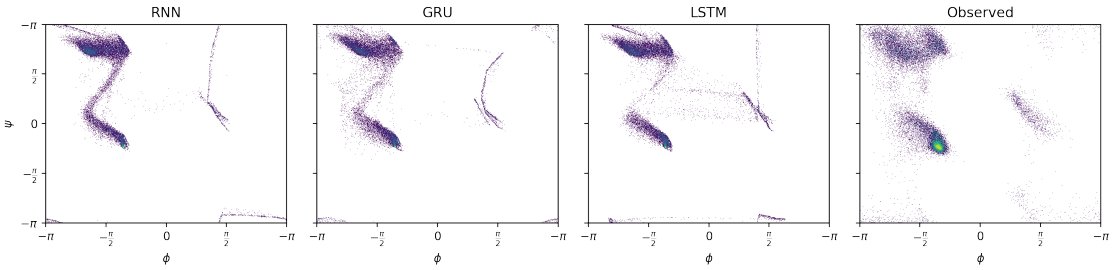


Figure 5: Ramachandran plots produced by Bifrost-predicted dihedral angle means for the different sequence models.

3.3 Extending Bifrost with the Theseus likelihood

In order to address the short-comings of Bifrost-modelled means to resemble Ramachandran behavior, we fine tuned the model applying likelihoods over 3D-coordinates as well as dihedrals. The resulting model showed better JSD as well as angle RMSD while improving RMSD over reconstructed 3D-coordinates (table 3a). We observed clear improvement in the Ramachandran plot as well, completely abolishing predictions in the disallowed region, while trimming the individual clusters to resemble the observed distribution (figure 6).

We observed that the Theseus likelihood improves the predictive performance, significantly reducing the RMSDs of the reconstructed structures (figure 7). RMSD values generally increase as the fragment length increases. However, the models trained with the Theseus likelihood still outperformed the models trained without it while retaining the improvement in terms of JSD and KLD (table 3b and 3c). In figure 8 we show representative examples of sampled structures from models trained with and without the Theseus likelihood. More examples are shown in figure S1.

4 Discussion

Here, we presented a novel multiscale approach for training a model of protein structure by employing likelihoods over both internal coordinates and 3D coordinates. The model is based on a deep Markov model (DMM) [23] using a directional and a Procrustes likelihood. We introduce a new likelihood over dihedral angles through the Sine Bivariate von Mises distribution (SBvM) [25], allowing us to treat the ϕ, ψ angle pairs as dependent and correlated. We observed that the model was unable to learn fragment structure when training a model from scratch with both internal- and 3D-coordinate likelihoods while also suffering from numerical instability (Data not shown). We hypothesised that this was due to the error from the Theseus likelihood being too large for the initial near-random dihedral angle predictions.

Table 3: Comparison between Bifrost with and with out finetuning using Theseus likelihoods in terms of JS divergence of Ramachandran number distributions, median RMSDs between predicted and observed ϕ/ψ angles and 3D-coordinate RMSDs. The metrics indicating best performance are highlighted in bold.

	- Theseus		+ Theseus	
	μ	Sampled	μ	Sampled
JSD	0.171	0.130	**0.102**	0.107
ϕ/ψ RMSD	0.657	0.671	**0.556**	0.563
3D RMSD	1.628	1.923	**1.159**	1.172

(a) 9-mer fragments

	- Theseus		+ Theseus	
	μ	Sampled	μ	Sampled
JSD	0.176	0.133	**0.107**	0.113
ϕ/ψ RMSD	0.856	0.867	**0.460**	0.467
3D RMSD	4.617	5.265	**2.362**	2.553

(b) 15-mer fragments

	- Theseus		+ Theseus	
	μ	Sampled	μ	Sampled
JSD	0.177	0.140	0.113	**0.104**
ϕ/ψ RMSD	0.852	0.863	**0.402**	0.411
3D RMSD	9.816	10.242	**5.243**	5.539

(c) 30-mer fragments

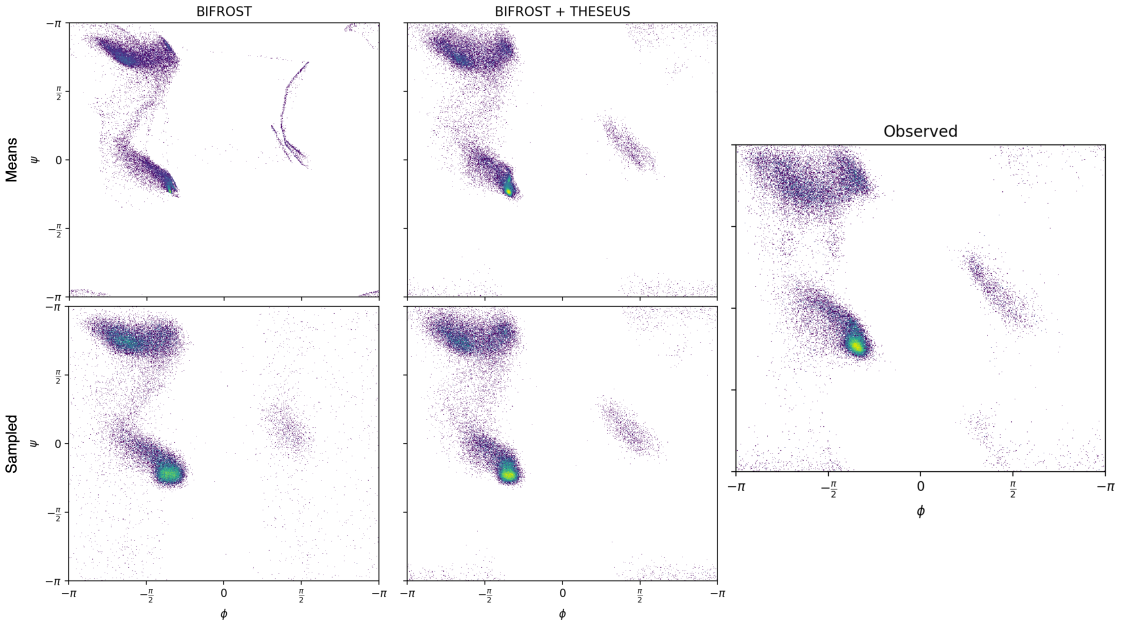


Figure 6: Ramachandran plots of dihedral angle means generated by the pretrained Bifrost model, and the same Bifrost model finetuned with the Theseus loss (left, top row). Left, bottom row: sampled angles. right: Observed angles.

We alleviated the numerical instability by initializing with the weights of a model trained only with the dihedral angle likelihood. Additionally, all parameters are amortised in the standard Bifrost model, but for the Theseus likelihood, per-datapoint rotations and variances need to be learned. Therefore, we applied a warm-up phase of these distribution parameters while freezing all neural network parameters. The combination of these steps reduced the final training stage to a fine tuning stage, where both likelihoods were combined to achieve better performance in terms of 3D-coordinate RMSDs.

We showed, that the this multiscale approach allows us to model longer fragments. However, training the model on fragments longer than the lengths reported, results in an impractical increase in inference time. This is a result of the

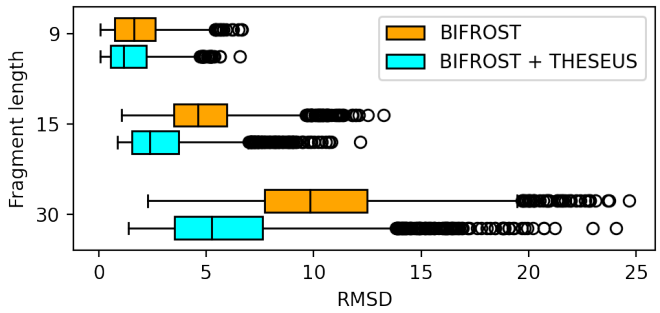


Figure 7: Distributions of RMSDs between reconstructed and observed coordinates of Bifrost trained with or without the Theseus likelihood for different fragment lengths.

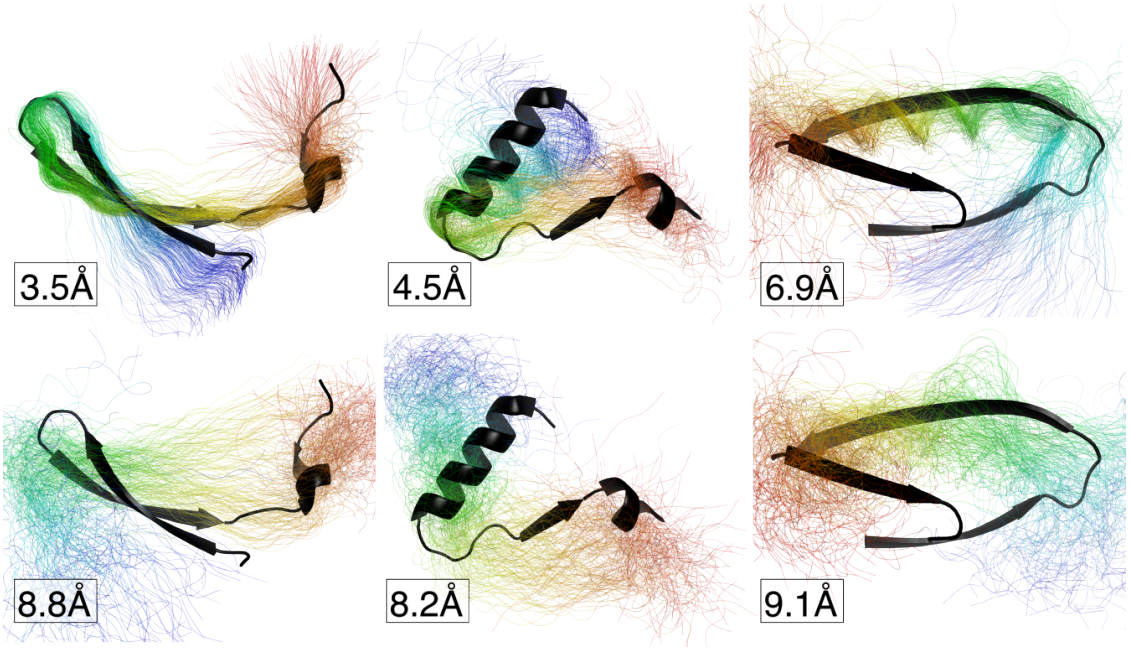


Figure 8: Comparison of the predictions (i.e. posterior distributions) obtained from Bifrost (bottom) and Bifrost-Theseus (top) for identical 30mer fragments. The black cartoon representations show the observed structure, while the thin lines represent 200 samples conditioned on the amino acid sequences. Their color varies from blue at the start of the fragment (N-terminus) to red at the end (C-terminus). Examples are taken from the 25th percentile, median and 75th of the distribution of 30mer RMSDs for Bifrost with Theseus in figure 7. The average RMSDs are shown next to the structures. Graphics produced in PyMol [36]

model being a deep Markov model employing a for loop over the amino acid sequence. We got around this limitation by drastically scaling down the size of the training data sets for 15- and 30-mer fragments. The run time limitation could be alleviated more generally. Either by moving to GPU-first or functional programming frameworks [37] or by implementing a model, that does not require sequential processing.

We assume that predicted means are idealized values. We argue that these are a "denoised" representation of the protein structure, while enforcing a good fit with the observed structure through the Theseus likelihood. This confines predicted dihedral angle means to values that correspond to reasonable geometry, which in turn results in better Ramachandran representations. Dihedral angles should fluctuate around means that are geometrically sound, which we enforce with this new likelihood. Thus, we can make the case that this change allows us to account for a heuristic interpretation of both epistemic and aleatory uncertainty.

The results presented here, show that it is possible to train a deep, generative, probabilistic model directly on 3D-coordinates in a rotationally and translationally invariant manner using the Theseus superposition model.

References

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A.A. Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[2] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R Glassman, Andy DeGiovanni, Jose H Pereira, Andria V Rodrigues, Alberdina A. Van Dijk, Ana C Ebrecht, Diederik J Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K Rathinaswamy, Udit Dalwadi, Calvin K Yip, John E Burke, K Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[3] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, page 2022.07.21.500999, 7 2022.

[4] Gwen R. Buel and Kylie J. Walters. Can AlphaFold2 predict the impact of missense mutations on structure?, 1 2022.

[5] John Ingraham, Adam Riesselman, Chris Sander, and Debora Marks. Learning protein structure with a differentiable simulator. In *International Conference on Learning Representations*, 2018.

[6] Kiersten M Ruff and Rohit V Pappu. AlphaFold and implications for intrinsically disordered proteins. *Journal of Molecular Biology*, 433(20):167208, 2021.

[7] Wouter Boomsma, Kanti V. Mardia, Charles C. Taylor, Jesper Ferkinghoff-Borg, Anders Krogh, and Thomas Hamelryck. A generative, probabilistic model of local protein structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105(26):8932, 7 2008.

[8] Kristin P Lennox, David B Dahl, Marina Vannucci, Ryan Day, and Jerry W Tsai. A Dirichlet process mixture of hidden Markov models for protein structure prediction. *The annals of applied statistics*, 4(2):916, 2010.

[9] T Edgoose, Lloyd Allison, and David L Dowe. An MML classification of protein structure that knows about angles and sequence. In *Pac Symp Biocomput*, volume 3, pages 585–96, 1998.

[10] Christopher Bystroff, Vesteinn Thorsson, and David Baker. HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins. *Journal of molecular biology*, 301(1):173–190, 2000.

[11] Anne-Claude Camproux, P Tuffery, JP Chevrolat, JF Boisvieux, and S Hazout. Hidden Markov model approach for identifying the modular framework of the protein backbone. *Protein engineering*, 12(12):1063–1073, 1999.

[12] Zeming Lin, Tom Sercu, Yann LeCun, and Alexander Rives. Deep generative models create new and diverse protein structures. In *Machine Learning for Structural Biology Workshop, NeurIPS*, 2021.

[13] Namrata Anand and Possu Huang. Generative modeling for protein structures. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[14] Raphael R Eguchi, Christian A Choe, and Po-Ssu Huang. Ig-VAE: Generative modeling of protein structure by direct 3D coordinate generation. *PLoS computational biology*, 18(6):e1010271, 2022.

[15] Douglas L. Theobald and Phillip A. Steindel. Optimal simultaneous superpositioning of multiple structures with missing data. *Bioinformatics*, 28(15):1972–1979, 8 2012.

[16] Lys Sanz Moreta, Ahmad Salim Salim Al-Sibahi, Thomas Hamelryck, Douglas Theobald, Basile Nicolas Rommes, William Bullock, and Andreas Manoukian. A Probabilistic Programming Approach to Protein Structure Superposition. *bioRxiv*, page 575431, 2019.

[17] Jerod Parsons, J Bradley Holmes, J Maurice Rojas, Jerry Tsai, and Charlie EM Strauss. Practical conversion from torsion space to Cartesian space for in silico protein synthesis. *Journal of computational chemistry*, 26(10):1063–1068, 2005.

[18] Mohammed AlQuraishi. Parallelized natural extension reference frame: parallelized conversion from internal to Cartesian coordinates. *Journal of computational chemistry*, 40(7):885–892, 2019.

[19] Christian Bahne Thygesen, Ahmad Salim Al-Sibahi, Lys Sanz Moreta, Christian Skjødt Steenmans, Anders Bundgård Sørensen, and Thomas W Hamelryck. Efficient Generative Modelling of Protein Structure Fragments using a Deep Markov Model. *Proceedings of the 38th International Conference on Machine Learning*, PMLR139:10258–10267, 2021.

[20] John C Gower and Garmt B Dijksterhuis. *Procrustes problems*, volume 30. OUP Oxford, 2004.

[21] Jacob Fish, Gregory J. Wagner, and Sinan Keten. Mesoscopic and multiscale modelling in materials. *Nature Materials 2021 20:6*, 20(6):774–786, 5 2021.

[22] Q. H. Zeng, A. B. Yu, and G. Q. Lu. Multiscale modeling and simulation of polymer nanocomposites. *Progress in Polymer Science*, 33(2):191–269, 2 2008.

[23] Rahul G. Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 2101–2109, 9 2017.

[24] Guoli Wang and Roland L Dunbrack. PISCES: Recent improvements to a PDB sequence culling server. *Nucleic Acids Research*, 33(SUPPL. 2), 2005.

[25] Ola Ronning, Christophe Ley, Kanti V Mardia, and Thomas Hamelryck. Time-efficient Bayesian Inference for a (Skewed) Von Mises Distribution on the Torus in a Deep Probabilistic Programming Language. In *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–8. IEEE, 9 2021.

[26] Kanti V Mardia. Statistics of Directional Data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 37(3):349–371, 1975.

[27] Harshinder Singh, Vladimir Hnizdo, and Eugene Demchuk. Probabilistic Model for Two Dependent Circular Variables. *Biometrika*, 89(3):719–723, 2002.

[28] Mohammed AlQuraishi. End-to-end differentiable learning of protein structure. *Cell systems*, 8(4):292–301, 2019.

[29] Xavier Perez-Sala, Laura Igual, Sergio Escalera, and Cecilio Angulo. Uniform sampling of rotations for discrete and continuous learning of 2D shape models. *Robotic Vision: Technologies for Machine Learning and Vision Applications*, pages 23–42, 2012.

[30] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, 2014.

[31] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[32] Ranjan Mannige. The BackMAP Python module: how a simpler Ramachandran number can simplify the life of a protein simulator. *PeerJ*, 6:e5745, 10 2018.

[33] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[34] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 3 1951.

[35] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[36] L L C Schrödinger and Warren DeLano. PyMOL.

[37] Troels Henriksen, Niels G. W. Serup, Martin Elsman, Fritz Henglein, and Cosmin E. Oancea. Futhark: Purely Functional GPU-programming with Nested Parallelism and In-place Array Updates. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2017, pages 556–571, New York, NY, USA, 2017. ACM.

5 Supplementary material

Table S1: Idealised values for bond lengths and the ω dihedral angle.

Bond lengths ($\mathring{A}$)			Dihedral angle (radians)
$C-N$	$N-C_\alpha$	$C_\alpha-C$	ω
1.45801	1.52326	1.32868	π

Table S2: The three different stages of model training.

	Bifrost	Theseus	Bifrost & Theseus
Optimizer	ADAM	AdagradRMSprop	ADAM
Epochs	Until convergence	50	Until convergence

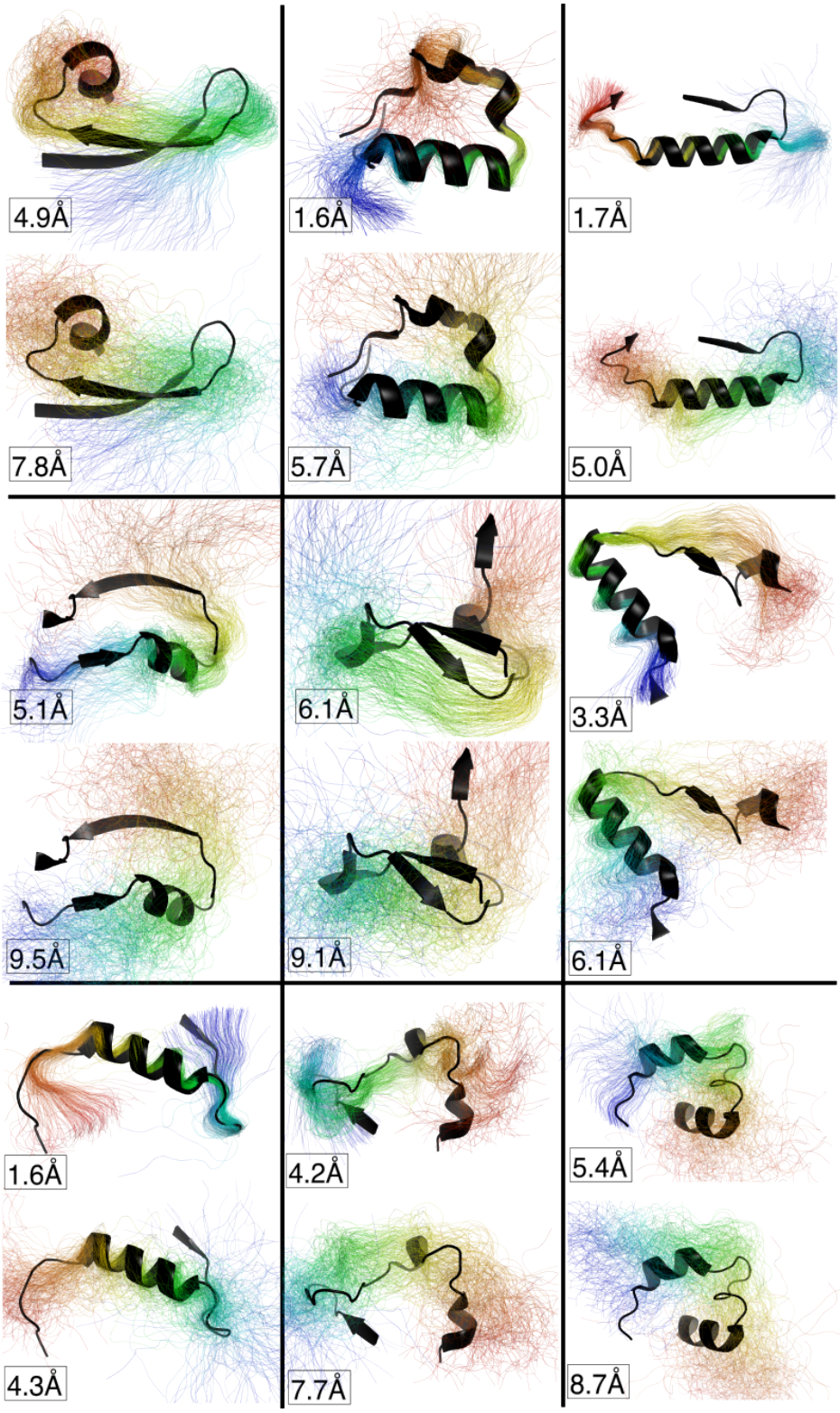


Figure S1: More examples of predictions from models trained with (top) or without (bottom) the Theseus submodel. Mean RMSDs are shown next to the structures.

Chapter 8

Epilogue

8.1 Conclusions

THROUGHOUT this thesis, deep probabilistic programming frameworks were applied to develop a model of local protein structure for use in a vaccine development setting. The model presented in paper 1 [85] was based on the DMM architecture and trained with SVI in the probabilistic programming framework Pyro [43]. The model performed on par with the current state of the art approach of Rosetta's fragment picker [23], while requiring a fraction of the run time and relying only on the amino acid sequence of the small fragments. However, the model focuses entirely on dihedral angles, which means that the model is limited to short fragments, due to the elbow effect of small prediction errors in the dihedral angle space leading to larger errors when reconstructing the 3D-coordinates. In paper 3, I extended the BIFROST model to include a more sophisticated sequence model in the GRU, while also changing the dihedral angle likelihoods to the recently developed Sine Bivariate Von Mises distribution [88]. I addressed the limitations of dihedral angle-based modelling by introducing a likelihood over 3D-coordinates [87] to further regularise the model by accounting for the 3D-coordinate reconstruction error as well. This not only improved the model on short fragments, but also allowed the model to be trained on larger fragments up to 30 amino acids long.

Due to the increased throughput allowed by the BIFROST approach compared to existing methods, it can be applied in a vaccine design setting (paper 2). As the model infers appropriate latent representations of an amino acid sequence and structure, it proved to function well as a similarity function capturing information on both sequence and structure. This means that it is possible to use the model to scan any protein, through the latent state, and identify the similarity between a set of external peptides that could replace parts of the original sequence with minimal impact on the final structure. We used this to enrich the most common SARS-CoV-2 vaccine protein - the spike protein - with epitopes that would be presented by the immune system

to induce the antibody response through the $CD4^+$-signalling pathway. We showed that it is theoretically possible to do so, as the enriched constructs we produced were in fact able to be produced by host cells at a higher rate than constructs designed using a naive sequence similarity-based approach. Additionally, we showed that the presented approach was able to induce a T cell response against the grafted epitopes without compromising the antibody response against the wildtype protein. Unfortunately, due to constraints in time and laboratory resources we did not get a chance to test the vaccine constructs in a challenge model in mice in order to prove that the optimised vaccine constructs could in fact induce an increased protection against SARS-CoV-2.

8.2 Future directions

While the BIFROST model improved dramatically with the addition of the THESEUS-likelihood for 3D-coordinate reconstruction, it still has one major obstacle: time. The fact that the model employs an iterative processing of every single position of an amino acid sequence makes it impractically slow when the proteins to model exceed 30 amino acids. Thus, the model is a ways off from predicting full protein structures. This can be improved in multiple ways. Either the model can be ported to a GPU/TPU-first framework such as numpyro and JAX for a practical improvement in runtime. Alternatively, the sequential DMM could be reduced to a variational autoencoder using a paralellisable sequence model such as the attention-based architecture, the transformer. Another direction for this model could be to move away from the VAE-like architectures and towards diffusion models [89], which have recently showed promising results for the problem of protein structure prediction [34, 35]. At the moment, the BIFROST model focuses purely on dihedral angles and coordinates of the polypeptide backbone, and does not model the side chains. Obviously this is a major limitation for moving towards full protein structure prediction, where the orientation of the side chains play a huge role in forming the fold of the protein. Adding the side chains is not necessarily a massive change to the model, but it requires handling practical challenges, such the variable number of side chain atoms of different amino acids. It would also bring on some further modelling assumptions for the Theseus likelihood on whether to assume uniform variance per amino acid, or whether the coordinates of the side chain atoms should have independent variances. These small hurdles, should, however be rather trivial to implement, given the time.

The universal grafting approach presented in paper 2 showed promise in a protein engineering setting. However, it should be noted that all the experiments carried out to evaluate the modified protein constructs were proximity measures of the real thing. While it is encouraging that we observe that mice immunized with a modified protein could elicit antibodies against the

wildtype protein, it is not final proof that the final fold is unaffected. In order to truly validate this approach for protein engineering, I suggest to design an experiment from scratch. This experiment should focus on a known, stable protein of a moderate size to allow for some grafts to be performed. This protein should then be gradually modified by performing more and more grafts in order to truly understand how far we can push this modification. The final fold of the modified protein constructs should be validated through physiochemical studies such as circular dichroism [90] or actual solving of the structure through Cryo-EM or NMR. The former is yet another approximate measure of the final fold, but closer to a true validation, while the latter may be prohibitively expensive, but would provide final proof-of-concept.

Finally, the developed vaccine constructs should be tested in a mouse challenge model to see if the T cell epitope-enriched vaccine constructs can either induce an increased protection against the virus. Alternatively, this model could be used to induce a novel response against viral proteins, that would otherwise not be recognized by the immune system. However, without a challenge model it is not possible to answer these open questions.

Bibliography

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 8 2021.

[2] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R Glassman, Andy DeGiovanni, Jose H Pereira, Andria V Rodrigues, Alberdina A. Van Dijk, Ana C Ebrecht, Diederik J Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K Rathinaswamy, Udit Dalwadi, Calvin K Yip, John E Burke, K Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373:871–876, 2021.

[3] Robin Pearce and Yang Zhang. Toward the solution of the protein structure prediction problem. *Journal of Biological Chemistry*, 297:100870, 7 2021.

[4] M S Smyth. *x* Ray crystallography. *Molecular Pathology*, 53:8–14, 2 2000.

[5] Dominique Marion. An Introduction to Biological NMR Spectroscopy. *Molecular & Cellular Proteomics*, 12:3006–3025, 11 2013.

[6] Marina Serna. Hands on Methods for High Resolution Cryo-Electron Microscopy Structures of Heterogeneous Macromolecular Complexes. *Frontiers in Molecular Biosciences*, 6:33, 5 2019.

[7] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips. A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. *Nature*, 181:662–666, 3 1958.

[8] H. M. Berman. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 1 2000.

[9] Pierrick Craveur, Agnel Praveen Joseph, Pierre Poulain, Alexandre G. De Brevern, and Joseph Rebehmed. Cis-trans isomerization of omega dihedrals in proteins. *Amino acids*, 45:279–289, 8 2013.

[10] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7:95–99, 1963.

[11] Jerod Parsons, J. Bradley Holmes, J. Maurice Rojas, Jerry Tsai, and Charlie E. M. Strauss. Practical conversion from torsion space to Cartesian space forin silico protein synthesis. *Journal of Computational Chemistry*, 26:1063–1068, 7 2005.

[12] Mohammed AlQuraishi. Parallelized Natural Extension Reference Frame: Parallelized Conversion from Internal to Cartesian Coordinates. *Journal of Computational Chemistry*, 40:885–892, 3 2019.

[13] Michael L Tress and Alfonso Valencia. Predicted residue-residue contacts can help the scoring of 3D models. *Proteins: Structure, Function, and Bioinformatics*, 78(8):1980–1991, 2010.

[14] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[15] David T Jones, Daniel WA Buchan, Domenico Cozzetto, and Massimiliano Pontil. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

[16] Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. Assessing the utility of coevolution-based residue & residue contact predictions in a sequence- and structure-rich era. *Proceedings of the National Academy of Sciences*, 110(39):15674–15679, 2013.

[17] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *bioRxiv*, 2016.

[18] Ken A. Dill and Justin L. MacCallum. The protein-folding problem, 50 years on. *Science*, 338:1042–1046, 11 2012.

[19] Cyrus Levinthal. How to fold graciously. *Mössbauer Spectroscopy in Biological Systems Proceedings*, 24:22–24, 1969.

[20] Carol A. Rohl, Charlie E.M. Strauss, Kira M.S. Misura, and David Baker. Protein Structure Prediction Using Rosetta. *Methods in Enzymology*, 383:66–93, 2004.

[21] Yifan Song, Frank Dimaio, Ray Yu Ruei Wang, David Kim, Chris Miles, TJ Brunette, James Thompson, and David Baker. High-resolution comparative modeling with RosettaCM. *Structure*, 21:1735–1742, 2013.

[22] Rebecca F. Alford, Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, P. Douglas Renfrew, Vikram K. Mulligan, Kalli Kappel, Jason W. Labonte, Michael S. Pacella, Richard Bonneau, Philip Bradley, Roland L. Dunbrack, Rhiju Das, David Baker, Brian Kuhlman, Tanja Kortemme, and Jeffrey J. Gray. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of Chemical Theory and Computation*, 13:3031–3048, 6 2017.

[23] Dominik Gront, Daniel W Kulp, Robert M Vernon, Charlie E.M. Strauss, and David Baker. Generalized fragment picking in rosetta: Design, protocols and applications. *PLoS ONE*, 6:23294, 2011.

[24] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202, 1999.

[25] Julia Koehler Leman, Ralf Mueller, Mert Karakas, Nils Woetzel, and Jens Meiler. Simultaneous prediction of protein secondary structure and transmembrane spans. *Proteins: Structure, Function and Bioinformatics*, 81:1127–1140, 7 2013.

[26] Kevin Karplus. SAM-T08, HMM-based protein structure prediction. *Nucleic Acids Research*, 37:492–497, 2009.

[27] Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjaergaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Soenderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, et al. NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, 2019.

[28] Jianyi Yang, Renxiang Yan, Ambrish Roy, Dong Xu, Jonathan Poisson, and Yang Zhang. The I-TASSER Suite: protein structure and function prediction. *Nature methods*, 12(1):7–8, 2015.

[29] Javier Delgado, Leandro G Radusky, Damiano Cianferoni, and Luis Serrano. FoldX 5.0: working with RNA, small molecules and a new graphical interface. *Bioinformatics*, 35(20):4168–4169, 2019.

[30] Benjamin Webb and Andrej Sali. Comparative protein structure modeling using MODELLER. *Current protocols in bioinformatics*, 54(1):5–6, 2016.

[31] Mohammed AlQuraishi. End-to-end differentiable learning of protein structure. *Cell systems*, 8(4):292–301, 2019.

[32] Jinbo Xu. Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences*, 116(34):16856–16865, 2019.

[33] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, page 2022.07.21.500999, 7 2022.

[34] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-Specific Antibody Design and Optimization with Diffusion-Based Generative Models. *bioRxiv*, page 2022.07.10.499510, 8 2022.

[35] Namrata Anand and Tudor Achim. Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models. *arXiv*, 5 2022.

[36] Michael Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv*, 2018.

[37] David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[38] Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *arXiv*, 11 2011.

[39] Matthew D Hoffman, David M Blei, Chong Wang, John Paisley, Jpaisley@berkeley Edu, and Tommi Jaakkola. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[40] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A. Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76, 2017.

[41] John Salvatier, Thomas V. Wiecki, and Christopher Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2016, 2016.

[42] Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A. Saurous. TensorFlow Distributions. *arXiv*, 11 2017.

[43] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep Universal Probabilistic Programming. *J. Mach. Learn. Res.*, 20:28:1–28:6, 2019.

[44] Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro. *arXiv*, 12 2019.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[46] Charles R Harris, K Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 9 2020.

[47] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[48] Tom Hennigan, Trevor Cai, Tamara Norman, and Igor Babuschkin. Haiku: Sonnet for jax, 2020.

[49] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5:115–133, 12 1943.

[50] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

[51] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[52] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org.

[53] David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. MIT Press, 1987.

[54] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, 2014.

[55] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[56] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv*, 9 2016.

[57] Christopher M Bishop. Latent variable models. In *Learning in graphical models*, pages 371–403. Springer, 1998.

[58] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv*, 12 2013.

[59] Rahul G. Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 2101–2109, 9 2017.

[60] Leonard E. Baum and Ted Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37:1554–1563, 12 1966.

[61] Yoshua Bengio and Paolo Frasconi. An Input Output HMM Architecture. *Neural Information Processing Systems*, pages 427–434, 1995.

[62] Kenneth Murphy, Paul Travers, Mark Walport, and Charles Janeway. *Janeway's Immunology*. Garland Science, 2011.

[63] Jennifer L. Hope and Linda M. Bradley. Lessons in antiviral immunity. *Science*, 371(6528):464–465, 2021.

[64] Jonathan H. Lam, Fauna L. Smith, and Nicole Baumgarth. B Cell Activation and Response Regulation During Viral Infections. *Viral Immunology*, 33:294, 2020.

[65] Janice S. Blum, Pamela A. Wearsch, and Peter Cresswell. Pathways of Antigen Processing. *Annual review of immunology*, 31:443, 3 2013.

[66] Rebecca J. Cox and Karl A. Brokstad. Not just antibodies: B cells and T cells mediate immunity to COVID-19. *Nature Reviews Immunology*, 20:581–582, 10 2020.

[67] Susan L. Swain, K. Kai McKinstry, and Tara M. Strutt. Expanding roles for CD4+ T cells in immunity to viruses. *Nature Reviews. Immunology*, 12:136, 2 2012.

[68] Xiaoyu Wang, Rui Ma, Xiangming Xie, Weina Liu, Tao Tu, Fei Zheng, Shuai You, Jianzhong Ge, Huifang Xie, Bin Yao, and Huiying Luo. Thermostability improvement of a Talaromyces leycettanus xylanase by rational protein engineering. *Scientific Reports*, 7:15287, 12 2017.

[69] Xinhao Ye, Chenming Zhang, and Y.-H. Percival Zhang. Engineering a large protein by combined rational and random approaches: stabilizing the Clostridium thermocellum cellobiose phosphorylase. *Mol. BioSyst.*, 8:1815–1823, 2012.

[70] Emiko Mihara, Satoshi Watanabe, Nasir K. Bashiruddin, Nozomi Nakamura, Kyoko Matoba, Yumi Sano, Rumit Maini, Yizhen Yin, Katsuya Sakai, Takao Arimori, Kunio Matsumoto, Hiroaki Suga, and Junichi Takagi. Lasso-grafting of macrocyclic peptide pharmacophores yields multifunctional proteins. *Nature Communications 2021 12:1*, 12:1–12, 3 2021.

[71] Ali Reza A. Ladiwala, Moumita Bhattacharya, Joseph M. Perchiacca, Ping Cao, Daniel P. Raleigh, Andisheh Abedini, Ann Marie Schmidt, Jobin Varkey, Ralf Langen, and Peter M. Tessier. Rational design of potent domain antibody inhibitors of amyloid fibril assembly. *Proceedings of the National Academy of Sciences*, 109:19965–19970, 12 2012.

[72] Pietro Sormanni, Francesco A. Aprile, and Michele Vendruscolo. Rational design of antibodies targeting specific epitopes within intrinsically disordered proteins. *Proceedings of the National Academy of Sciences*, 112:9902–9907, 8 2015.

[73] Saurabh Bansal and Bishwajit Kundu. Chapter 39 - protein engineering: Methods and applications. In Timir Tripathi and Vikash Kumar Dubey,

editors, *Advances in Protein Molecular and Structural Biology Methods*, pages 641–668. Academic Press, 2022.

[74] Sheryl B. Rubin-Pitel, Catherine M-H. Cho, Wilfred Chen, and Huimin Zhao. Chapter 3 - directed evolution tools in bioproduct and bioprocess development. In Shang-Tian Yang, editor, *Bioprocessing for Value-Added Products from Renewable Resources*, pages 49–72. Elsevier, Amsterdam, 2007.

[75] Roberto A Chica, Nicolas Doucet, and Joelle N Pelletier. Semi-rational approaches to engineering enzyme activity: combining the benefits of directed evolution and rational design. *Current opinion in biotechnology*, 16(4):378–384, 2005.

[76] Joseph M. Perchiacca, Ali Reza A. Ladiwala, Moumita Bhattacharya, and Peter M. Tessier. Structure-based design of conformation- and sequence-specific antibodies against amyloid β. *Proceedings of the National Academy of Sciences*, 109:84–89, 1 2012.

[77] Joseph M. Perchiacca and Peter M. Tessier. Engineering aggregation-resistant antibodies. *Annual Review of Chemical and Biomolecular Engineering*, 3:263–286, 7 2012.

[78] Samuel K. Sia and Peter S. Kim. Protein grafting of an HIV-1-inhibiting epitope. *Proceedings of the National Academy of Sciences of the United States of America*, 100:9756–9761, 8 2003.

[79] Daisuke Fujiwara, Hidekazu Kitada, Masahiro Oguri, Toshio Nishihara, Masataka Michigami, Kazunori Shiraishi, Eiji Yuba, Ikuhiko Nakase, Haeri Im, Sunhee Cho, Jong Young Joung, Seiji Kodama, Kenji Kono, Sihyun Ham, and Ikuo Fujii. A Cyclized Helix-Loop-Helix Peptide as a Molecular Scaffold for the Design of Inhibitors of Intracellular Protein–Protein Interactions by Epitope and Arginine Grafting. *Angewandte Chemie - International Edition*, 55:10612–10615, 8 2016.

[80] Maxim Rossmann, Sandra J. Greive, Tommaso Moschetti, Michael Dinan, and Marko Hyvönen. Development of a multipurpose scaffold for the display of peptide loops. *Protein Engineering, Design and Selection*, 30:419–430, 6 2017.

[81] Lukas Kurt Josef Stadler, Toni Hoffmann, Darren Charles Tomlinson, Qifeng Song, Tracy Lee, Michael Busby, Yvonne Nyathi, Elisenda Gendra, Christian Tiede, Keith Flanagan, Simon J. Cockell, Anil Wipat, Colin Harwood, Simon D. Wagner, Margaret A. Knowles, Jason J. Davis,

Neil Keegan, and Paul Ko Ferrigno. Structure-function studies of an engineered scaffold protein derived from Stefin A. II: Development and applications of the SQT variant. *Protein Engineering, Design and Selection*, 24:751–763, 9 2011.

[82] Sarah K. Madden, Albert Perez-Riba, and Laura S. Itzhaki. Exploring new strategies for grafting binding peptides onto protein loops using a consensus-designed tetratricopeptide repeat scaffold. *Protein Science*, 28:738–745, 4 2019.

[83] Conan K. Wang and David J. Craik. Linking molecular evolution to molecular grafting. *Journal of Biological Chemistry*, 296:100425, 1 2021.

[84] Selin Ece, Serap Evran, Jan-Oliver Janda, Rainer Merkl, and Reinhard Sterner. Improving thermal and detergent stability of bacillus stearothermophilus neopullulanase by rational enzyme design. *Protein Engineering, Design and Selection*, 28(6):147–151, 2015.

[85] Christian Bahne Thygesen, Ahmad Salim Al-Sibahi, Lys Sanz Moreta, Christian Skjødt Steenmans, Anders Bundgård Sørensen, and Thomas W Hamelryck. Efficient Generative Modelling of Protein Structure Fragments using a Deep Markov Model. *Proceedings of the 38th International Conference on Machine Learning*, PMLR139:10258–10267, 2021.

[86] Douglas L. Theobald and Phillip A. Steindel. Optimal simultaneous superpositioning of multiple structures with missing data. *Bioinformatics*, 28(15):1972–1979, 8 2012.

[87] Lys Sanz Moreta, Ahmad Salim Al-Sibahi, Douglas Theobald, William Bullock, Basile Nicolas Rommes, Andreas Manoukian, and Thomas Hamelryck. A probabilistic programming approach to protein structure superposition. In *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–5, 2019.

[88] Ola Ronning, Christophe Ley, Kanti V Mardia, and Thomas Hamelryck. Time-efficient Bayesian Inference for a (Skewed) Von Mises Distribution on the Torus in a Deep Probabilistic Programming Language. *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–8, 9 2021.

[89] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *arXiv*, 6 2020.

[90] Robert W. Woody. Circular dichroism. *Methods in Enzymology*, 246:34–71, 1 1995.