A structured observation distribution for generative biological sequence prediction and forecasting

Eli N. Weinstein [1] Debora S. Marks [2]

Abstract

Generative probabilistic modeling of biological sequences has widespread existing and potential application across biology and biomedicine, from evolutionary biology to epidemiology to protein design. Many standard sequence analysis methods preprocess data using a multiple sequence alignment (MSA) algorithm, one of the most widely used computational methods in all of science(Van Noorden et al., 2014). However, as we show in this article, training generative probabilistic models with MSA preprocessing leads to statistical pathologies in the context of sequence prediction and forecasting. To address these problems, we propose a principled drop-in alternative to MSA preprocessing in the form of a structured observation distribution (the "MuE" distribution). The MuE is a latent alignment model in which not only the alignment variable but also the regressor sequence can be latent. We prove theoretically that the MuE distribution comprehensively generalizes popular methods for inferring biological sequence alignments, and provide a precise characterization of how such biological models have differed from natural language latent alignment models. We show empirically that models that use the MuE as an observation distribution outperform comparable methods across a variety of datasets, and apply MuE models to a novel problem for generative probabilistic sequence models: forecasting pathogen evolution.

1. Introduction

High-throughput sequencing is pervasive across biology and biomedicine, and critical to both past and ongoing discoveries and technological advancements. Analyzing large scale sequence data, making predictions about unobserved or future sequences, and generating new functional sequences, are major and growing challenges with relevance to epidemiology (predicting pathogen evolution), immunology (characterizing antibody repertoires), molecular evolution (mapping substructure within protein families), protein design, and many more subfields of biology and biomedicine. In principal, generative probabilistic modeling enables (a) modular and uncertainty-aware data analysis, (b) formal mathematical statement of underlying assumptions, and (c) generation of new samples, which in the case of sequences can be synthesized and tested in the laboratory (taking advantage of recent rapid progress in high-throughput synthesis) (Kucukelbir et al., 2017; Russ et al., 2020). However, although machine learning and statistics offer an extraordinary array of generative probabilistic models, extending existing methods to apply to biological sequences while accounting for domain-specific prior knowledge is nontrivial.

When analyzing biological sequence data, a standard approach is to preprocess the data before building any models by constructing a multiple sequence alignment (MSA). MSA algorithms are among the most widely used methods in all of science; according to a 2014 analysis, the 10th most cited scientific paper of all time is an MSA algorithm, ahead of all other computational data analysis and statistics papers (Van Noorden et al., 2014; Thompson et al., 1994; 1997). Recent major advances in machine learning and statistical methods for protein structure prediction, variant effect prediction for clinical genetics, protein design, epidemiological tracking, and more have continued to rely on MSAs (Marks et al., 2011; Frazer et al., 2020; Russ et al., 2020; Hadfield et al., 2018). Although MSAs are a powerful tool for understanding sequence evolution, in Section 4.1 of this paper we show that employing MSAs as preprocessing introduces statistical pathologies in the context of generative sequence prediction and forecasting.

As a principled, drop-in alternative to MSA preprocessing, this paper provides a structured observation distribution for biological sequences, the "mutational emission" ("MuE") distribution. Observation distributions are a common general-purpose technique for extending continuous-

[1] Program in Biophysics, Harvard University, Cambridge, MA, USA [2] Department of Systems Biology, Harvard Medical School, Boston, MA, USA. Correspondence to: Eli N. Weinstein <eweinstein@g.harvard.edu>, Debora S. Marks <debbie@hms.harvard.edu>.

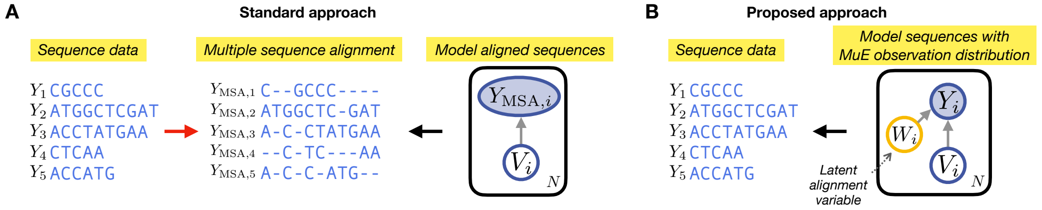

A
Standard approach
Sequence data
Multiple sequence alignment
Model aligned sequences
Y_1 CGCCC
Y_2 ATGGCTCGAT
Y_3 ACCTATGAA
Y_4 CTCAA
Y_5 ACCATG
$Y_{\mathrm{MSA},1}$ C--GCCC----
$Y_{\mathrm{MSA},2}$ ATGGCTC-GAT
$Y_{\mathrm{MSA},3}$ A-C-CTATGAA
$Y_{\mathrm{MSA},4}$ --C-TC---AA
$Y_{\mathrm{MSA},5}$ A-C-C-ATG--
$Y_{\mathrm{MSA},i}$
V_i
N
B
Proposed approach
Sequence data
Model sequences with MuE observation distribution
Y_1 CGCCC
Y_2 ATGGCTCGAT
Y_3 ACCTATGAA
Y_4 CTCAA
Y_5 ACCATG
Y_i
W_i
V_i
N
Latent alignment variable

Figure 1. (A) A standard approach to building biological sequence models is to preprocess the data by constructing an MSA. (B) We propose modifying the model instead of the data using the MuE distribution.

space models to other types of data, perhaps most familiar in the context of generalized linear models, where they are sometimes also referred to as "error", "emission", or "output" distributions. For instance, to predict count data, one might use a Poisson as an observation distribution, or to predict positive continuous data, one might use a Gamma. Good observation distributions account for both the support of the data and common forms of variability or noise in the data. For biological sequences, we propose using the MuE as an observation distribution. The MuE takes the form of a latent alignment model in which the regressor sequence can also be latent (Deng et al., 2018).[1]

The major contributions of the paper are (1) identification of statistical pathologies introduced by widely-used MSA preprocessing methods, (2) a drop-in general purpose alternative, the MuE distribution, (3) a unified and comprehensive theoretical framework for cataloging and rederiving existing biological latent alignment models from the MuE and (4) a novel application of generative probabilistic sequence models enabled by these advancements: forecasting pathogen evolution. At the most practical level, our approach provides a complete recipe for applying one's generative model of choice to biological sequence data while avoiding the pathologies of MSA preprocessing: add a softmax linker function and the MuE to the output of the model.

2. Method

2.1. Background: MSA preprocessing

MSA algorithms are applied to families of evolutionarily related biological sequences (protein, RNA or DNA). Let $\{Y_1, \ldots, Y_N\}$ be a dataset of N such sequences, which may each be different in length, and let $\mathcal{B}$ denote the alphabet (e.g. $\mathcal{B} = \{A, T, G, C\}$ for DNA). The principal goal of an MSA algorithm is to infer sites in each input sequence that are likely to be related to one another, meaning that they descend from a common ancestor. To represent the result of this inference, MSA algorithms convert the sequence dataset into an N by J matrix, adding gap symbols "$-$" such that sites in the same matrix column are those inferred to be related (Figure 1A). Mathematically, MSA algorithms can be summarized as nonlinear functions f_{MSA} that take in datasets of sequences and return processed datasets, $f_{\mathrm{MSA}}(\{Y_1, \ldots, Y_N\}) = \{Y_{\mathrm{MSA},1}, \ldots Y_{\mathrm{MSA},N}\}$; for each $i \in \{1, \ldots, N\}$, we have $Y_{\mathrm{MSA},i} \in (\mathcal{B} \cup \{-\})^J$. Note J itself will depend on the input dataset.

MSA preprocessing is useful in that it (a) converts the data into a matrix, and (b) adjusts for common sources of variability in biological sequence data, in particular insertion and deletion mutations. MSA preprocessing makes building statistical models of sequences more straightforward. For instance, starting from an arbitrary generative model $p_\theta(v)$ describing continuous matrices $v \in \mathbb{R}^{J\times(B+1)}$, where $B := |\mathcal{B}|$, one general strategy is to employ a softmax linker function and a categorical observation distribution $(\mathrm{softmax}(v)_j := \exp(v_{j,b})/\sum_{b'} \exp(v_{j,b'})$ for $j \in \{1, \ldots, J\})$. The complete approach (Figure 1A) is,

$$\begin{aligned}&\text{Preprocess: } \{Y_{\mathrm{MSA},1}, \ldots Y_{\mathrm{MSA},N}\} := f_{\mathrm{MSA}}(\{Y_1, \ldots, Y_N\}),\\&\text{Model: } V_i \sim p_\theta(v) \quad Y_{\mathrm{MSA},i} \sim \mathrm{Categorical}(\mathrm{softmax}(V_i)).\end{aligned} \tag{1}$$

This method allows, for example, the application of generative image models (such as variational autoencoders) to biological sequence data (Riesselman et al., 2018). However, as we describe in depth in Section 4.1, MSA preprocessing introduces substantial problems: each row of the output matrix $Y_{\mathrm{MSA},i}$ depends via f_{MSA} on the entire input dataset $\{Y_1, \ldots, Y_N\}$ and we cannot know ahead of time how future raw data Y_{N+1} will change preprocessed past data $Y_{\mathrm{MSA},i\le N}$. This makes likelihood-based model evaluation on newly observed or heldout data ill-defined.

2.2. The mutational emission distribution

As a drop-in alternative to MSA preprocessing, we introduce the "mutational emission" ("MuE") distribution. The MuE can be used in place of the Categorical observation distribution in Equation 1,

$$\text{Model: } V_i \sim p_\theta(v) \quad Y_i \sim \mathrm{MuE}(\mathrm{softmax}(V_i), c, \ell, a^{(0)}, a^{(t)}), \tag{2}$$

where c, ℓ, $a^{(0)}$, and $a^{(t)}$ are parameters of the MuE and $v \in \mathbb{R}^{M\times D}$ (here the dimensions of v are hyperparame-

[1] We will refer to biological alignments (diagrammatic representations of relatedness between sequences) as "multiple sequence alignments" (Durbin et al., 1998). We will refer to machine learning alignments (latent variables which indicate which positions in one sequence generate which positions in another sequence) as "latent alignments" (Deng et al., 2018).

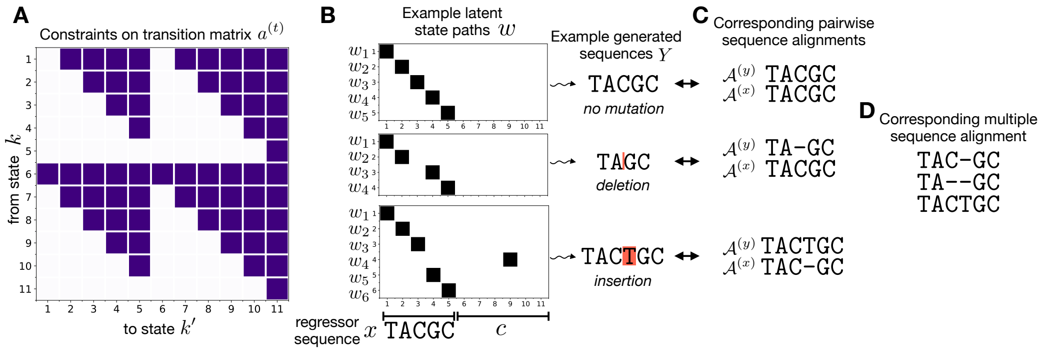

A
Constraints on transition matrix $a^{(t)}$
from state k
to state k'
1 2 3 4 5 6 7 8 9 10 11
B
Example latent state paths w
w_1 w_2 w_3 w_4 w_5 w_6
Example generated sequences Y
TACGC
no mutation
TAGC
deletion
TACTGC
insertion
regressor sequence x TACGC
c
C
Corresponding pairwise sequence alignments
$\mathcal{A}^{(y)}$ TACGC
$\mathcal{A}^{(x)}$ TACGC
$\mathcal{A}^{(y)}$ TA-GC
$\mathcal{A}^{(x)}$ TACGC
$\mathcal{A}^{(y)}$ TACTGC
$\mathcal{A}^{(x)}$ TAC-GC
D
Corresponding multiple sequence alignment
TAC-GC
TA--GC
TACTGC

Figure 2. (A) Condition 2.2 allows only the positions of $a^{(t)}$ in dark purple to be non-zero. (B) Example latent state paths w taken by the Markov model in the MuE, and sequences Y that they can generate, given x is a one-hot encoding of the DNA sequence TACGC. Rows correspond to positions $1, \dots, L$, columns correspond to latent states $1, \dots, K$. (C) w defines a pairwise alignment between X and Y via Definition 4.3. (D) The collection of w values describe a multiple sequence alignment of the generated sequences Y (Section 4.2).

ters rather than dimensions of the input data). The MuE avoids the pathologies of MSA preprocessing by directly generating complete, variable-length sequences (Figure 1B). We refer generically to models that use a MuE observation distribution, such as Equation 2, as "MuE observation" models. In the limiting case where $X_i := \mathrm{softmax}(V_i)$ is a one-hot encoding of a sequence (i.e. $X_{i,m,d} \in \{0,1\}$ and $\sum_d X_{i,m,d} = 1$), the MuE can be interpreted biologically as generating a mutant Y_i of the sequence X_i, with some probability of insertion, deletion and substitution mutations controlled by the parameters c, $a^{(0)}$, $a^{(t)}$ and ℓ (Section 2.3). A latent variable W_i in the MuE determines which positions in the regressor X_i generate which positions in Y_i, and can be interpreted in terms of a latent MSA (Section 4.2). Intuitively, the MuE "adds in", through a generative process, the same mutations that MSA algorithms are intended to "filter out" of the data via preprocessing.

The MuE is a hidden Markov model (HMM) with block-structured emission and transition matrices. Let Δ_D denote the $D-1$ dimensional probability simplex, $\Delta_D := \{v : v \in \mathbb{R}^D, v_d \geq 0, \sum_{d=1}^{D} v_d = 1\}$.

Definition 2.1 (MuE) $\mathrm{MuE}(x, c, \ell, a^{(0)}, a^{(t)})$ *is an HMM with $K = 2M+1$ latent states. The initial probability of each latent state is given by $a^{(0)} \in \Delta_K$, the latent state transition matrix is $a^{(t)} \in (\Delta_K)^K$, and the emission matrix is $\tilde{x} \in (\Delta_B)^K$. The matrices have block structure*

$$\tilde{x} := \begin{bmatrix} x \\ c \end{bmatrix} \cdot \ell, \quad a^{(t)} := \begin{bmatrix} A^{(1,1)} & A^{(1,2)} \\ A^{(2,1)} & A^{(2,2)} \end{bmatrix},$$

where $x \in (\Delta_D)^M$, $c \in (\Delta_D)^{M+1}$, $\ell \in (\Delta_B)^D$, $A^{(1,1)} \in \mathbb{R}^{M \times M}$, and $A^{(2,2)} \in \mathbb{R}^{(M+1)\times(M+1)}$. The transition matrix must satisfy Condition 2.2.

Condition 2.2 (Biological latent alignments) *Entries of $A^{(1,1)}, A^{(1,2)}, A^{(2,1)}$ and $A^{(2,2)}$ below the main diagonal must be zero. Entries of $A^{(1,1)}$ and $A^{(1,2)}$ on the main diagonal must also be zero.*

Condition 2.2, an upper triangular restriction, is illustrated in Figure 2A and justified in depth in Section 4.2.

2.3. Biological interpretation of the MuE

To describe the biological interpretation of the MuE we consider examples of different latent paths $w = (w_1, \dots, w_L)$ through state space that the MuE can take and the samples $Y \sim p_{\mathrm{MuE}}(y|x,w)$ that these paths will generate (Figure 2B). Assume to start that $D = B$ and $\ell = I_B$, where I_B is the $B \times B$ identity matrix, and consider the limiting case where x is a one-hot encoding of a sequence. *Example 1:* $w = (1, \dots, M)$ (no mutation). The generated Y will be an exact copy of x, i.e. $Y = x$ when Y is represented as a one-hot encoding (Figure 2B top). *Example 2:* $w = (1, \dots, m-1, m+1, \dots, M)$ (deletion). The generated Y will be missing the mth letter of x, that is $Y = (x_1, \dots, x_{m-1}, x_{m+1}, \dots, x_M)$ (Figure 2B middle). *Example 3:* $w = (1, \dots, m, M+m+1, m+1, \dots, M)$ (insertion). The generated Y will have an additional letter inserted after the mth letter of x, that is $Y = (x_1, \dots, x_m, S, x_{m+1}, \dots, x_M)$ where $S \sim \mathrm{Categorical}(c_{m+1})$ (Figure 2B bottom). Condition 2.2 guarantees that the states $k \in \{1, \dots, M\}$ are each visited at most once and in sequential order. Sequences such as $\{1, \dots, m, m, \dots, M\}$ (repeated letter) and $\{1, \dots, m+1, m, \dots, M\}$ (backtracking) are not allowed under Condition 2.2. More general matrices ℓ allow for substitution mutations, with the probability of converting from letter d to b given by $\ell_{d,b}$. For example, if $w = (1, \dots, M)$, then $Y \sim \mathrm{Categorical}(x \cdot \ell)$, that is Y is a mutant of x with substitution probabilities determined by ℓ and no insertion or deletion mutations.

MuE observation models directly generalize models that use MSA preprocessing in the special case where the dataset sequences are all the same length and the MSA algorithm does not add any gap symbols (that is, when $f_{\mathrm{MSA}}(\cdot)$ is the identity). Assume $D = B$, and consider the "no mutation limit" where $\ell = I_B$, $a_1^{(0)} = 1$, and $A_{m,m+1}^{(1,1)} = 1$ for all $m \in \{1, \ldots, M-1\}$. In this case we find, for samples Y of length M, that $Y \sim \mathrm{MuE}(x, c, \ell, a^{(0)}, a^{(t)})$ simplifies to $Y \sim \mathrm{Categorical}(x)$. Thus Equation 2 and Equation 1 become equivalent. In practice, we typically select priors on the MuE to favor the no mutation limit, since it serves as a null hypothesis.

2.4. Inference

The marginal likelihood of the MuE with the latent state variable of the HMM integrated out, $p_{\mathrm{MuE}}(y|x, c, \ell, a^{(0)}, a^{(t)})$, is analytically tractable via the HMM forward algorithm and differentiable. The standard forward algorithm requires $\mathcal{O}(L)$ sequential matrix multiplications, where L is the length of the sequence (typically a few hundred amino acids in our setting), but it can also be parallelized to achieve $\mathcal{O}(\log L)$ time (Särkkä & García-Fernández, 2020; Rush, 2020). Using the MuE marginal likelihood allows inference with automatic differentiation variational inference, stochastic gradient MCMC, and related scalable approximate Bayesian inference algorithms (Section S3.1) (Kucukelbir et al., 2017; Welling & Teh, 2011). We have made available an implementation of the MuE within the probabilistic programming language Edward2, making it straightforward to explore different MuE observation models and inference methods (Section S3.2) (Tran et al., 2018). `https://github.com/debbiemarkslab/MuE`.

3. Related Work

Methods that use MSA preprocessing. MSA preprocessing is widely used as a starting point for biological sequence data analysis, perhaps most commonly in combination with other non-probabilistic analysis methods. One very common class of probabilistic methods that nearly always use MSA preprocessing is phylogenetic models, which are central to evolutionary biology and epidemiology (where they are used to reconstruct disease outbreaks), and widely used in nearly every other area of biology (Hadfield et al., 2018; Felsenstein, 2004). Another is fitness models, including Potts models and variational autoencoder models, which are used to infer the structure of proteins and RNA, predict the functional effects of clinical variants, design new proteins, etc. (Marks et al., 2011; Hopf et al., 2017; Frazer et al., 2020; Russ et al., 2020).

Standard methods that avoid MSA preprocessing. Although MSA preprocessing is problematic from the perspective of probabilistic modeling, the use of probabilistic models to infer multiple sequence alignments – that is, in order to *accomplish* the preprocessing – is standard. Perhaps the most widely used such method is the profile HMM, which, besides being used to infer multiple sequence alignments, is also at the core of modern sequence database search methods and is used to define sequence families, among many other applications (Durbin et al., 1998; Johnson et al., 2010; El-Gebali et al., 2019). In Section 4.2 we show that the MuE distribution generalizes a variety of popular methods including the profile HMM. While connections between various methods have been described before, the generalization offered by the MuE is both unified and comprehensive, delimiting the extent of the model class (Holmes, 2017). Note also that some of these methods can be trained by interpreting an MSA as a point estimate of the latent alignment variable; this is distinct from the more common usage of MSA preprocessing described in Section 4.1 and is not subject to the same pathologies. The most closely related methods to MuE observation models are hybrid profile HMM models; our work goes further by providing a generalized approach to building and inferring such models (Wilburn & Eddy, 2020).

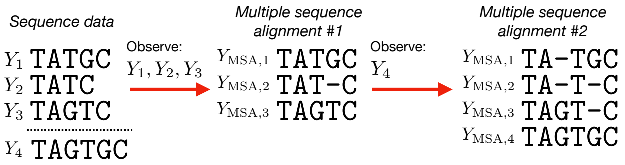

Sequence data
Y_1 TATGC
Y_2 TATC
Y_3 TAGTC
Y_4 TAGTGC
Observe: Y_1, Y_2, Y_3
Multiple sequence alignment #1
$Y_{\mathrm{MSA},1}$ TATGC
$Y_{\mathrm{MSA},2}$ TAT-C
$Y_{\mathrm{MSA},3}$ TAGTC
Observe: Y_4
Multiple sequence alignment #2
$Y_{\mathrm{MSA},1}$ TA-TGC
$Y_{\mathrm{MSA},2}$ TA-T-C
$Y_{\mathrm{MSA},3}$ TAGT-C
$Y_{\mathrm{MSA},4}$ TAGTGC

Figure 3. The multiple sequence alignment of the initial dataset Y_1, Y_2 and Y_3 can change as more data, Y_4, is added.

Recent methods that avoid MSA preprocessing. There has been intense recent interest in applying advances from natural language processing to biological sequences (Rives et al., 2019; Riesselman et al., 2019; Alley et al., 2019). Though natural language models have the correct support (that is, they can generate variable length discrete sequences), given the vast differences between natural language and biological sequences, it is not obvious how to design models with appropriate structure and inductive biases. Our theoretical results in Section 4.3 offer such a design.

4. Theory

4.1. Pathologies in MSA preprocessing

MSA preprocessing is typically applied to static sequence datasets and used for parameter inference problems; its statistical pathologies emerge when when we attempt to predict unobserved or future sequences. To explain these pathologies, we focus on the i.i.d. case.[2] Consider the following modeling assumption, which is nearly universal in the field of statistics:

Assumption 4.1 (I.i.d. data and model) *Let $p_0(x)$ be a probability distribution defined over a space $\mathcal{X}$, i.e. $p_0(x) \in$*

[2] Note that phylogenetic models, although not usually represented as i.i.d., are typically exchangeable and so possess an i.i.d. representation by de Finetti's theorem (Weinstein et al., 2020).

$\mathcal{P}(\mathcal{X})$ where $\mathcal{P}(\mathcal{X})$ is the set of all probability distributions over $\mathcal{X}$. We (1) assume that we observe independently and identically distributed samples $X_1, X_2, \ldots \sim p_0(x)$. In order to describe this process, we introduce a model $\mathcal{M} = \{q(x|\theta) : \theta \in \Theta\}$. We (2) assume $q(x|\theta) \in \mathcal{P}(\mathcal{X})$ for all $\theta \in \Theta$.

Now consider models that use MSA preprocessing and take the following form, of which Equation 1 is a special case:

Preprocess: $\{Y_{\text{MSA},1}, \ldots, Y_{\text{MSA},N}\} := f_{\text{MSA}}(\{Y_1, \ldots, Y_N\})$,

Model: $Y_{\text{MSA},i} \overset{iid}{\sim} p(y_{\text{MSA}})$,

where $p(y_{\text{MSA}}) \in \mathcal{P}((\mathcal{B} \cup \{-\})^J)$. If we attempt to employ Assumption 4.1 to describe the preprocessed data $Y_{\text{MSA},1}, \ldots, Y_{\text{MSA},N}$ we see that it is violated. Part 1 of Assumption 4.1 fails because the preprocessed data cannot consist of independent observations: if a datapoint Y_{N+1} is added to the dataset, then past data, i.e. $Y_{\text{MSA},1}, \ldots, Y_{\text{MSA},N}$, can be altered (Figure 3). For instance, the new sequence may provide additional evidence to the MSA algorithm that sites in previously observed sequences are related to one another. Part 2 of Assumption 4.1 fails because the model is not defined over a space that encompasses future data: if a datapoint Y_{N+1} is added to the dataset, the value of J may change (Figure 3). For instance, the new sequence might be longer than any before. Practically, the failure of MSA models to satisfy both parts of Assumption 4.1 makes rigorous likelihood-based evaluation of their generalization capacity untrustworthy. If we do not know what space future data lives in, or how past data will be altered with future measurements, it is hard to trust that the average log likelihood of our model on a held out test set is genuinely reflective of future model performance. More technically, the violation of Assumption 4.1 causes standard justifications for the use of Bayes factors, heldout likelihood, prequential evaluation, etc. to fail, see e.g. Dawid (1984); Vapnik (1999); Dawid (2011).

Using MSA preprocessing also fails to account for uncertainty in the alignment (Wu et al., 2012; Toth-Petroczy et al., 2016). The goal of an MSA algorithm is to infer related sites among a set of sequences, but the resulting MSA is only a point estimate of this quantity.

4.2. A unified and comprehensive framework

In this section we connect the MuE distribution to previously proposed probabilistic and non-probabilistic methods for inferring biological sequence alignments including MSAs. We start by more formally describing a biological pairwise alignment between two sequences X and Y, and then establish a connection with the latent state variable W in the MuE. Pairwise alignments serve as a diagrammatic representation of how two sequences X and Y may be related via insertion, deletion and substitution mutations.

Definition 4.2 (Biological pairwise alignment) *Let X and Y be sequences of length M and L respectively. A pairwise alignment $\mathcal{A}$ of X and Y with J columns is a matrix $[\mathcal{A}^{(x)}, \mathcal{A}^{(y)}]^\top$, where $\mathcal{A}^{(x)} \in (\mathcal{B} \cup \{-\})^J$ is a column vector of length J consisting of the letters of X, in order, and interspersed with gap symbols; similarly for $\mathcal{A}^{(y)}$. The alignment $\mathcal{A}$ must satisfy the condition that for every $j \in \{1, \ldots, J\}$ either $\mathcal{A}^{(x)}_j \in \mathcal{B}$ or $\mathcal{A}^{(y)}_j \in \mathcal{B}$ or both.*

Let j_l be the column of the alignment $\mathcal{A}$ in which the lth letter of Y falls, i.e. $\mathcal{A}^{(y)}_{j_l} = Y_l$ for $l \in \{1, \ldots, L\}$. Let g_l indicate whether the column j_l in $\mathcal{A}$ contains a gap, i.e. $g_l := \mathbb{I}(\mathcal{A}^{(x)}_{j_l} = -)$, where $\mathbb{I}(\cdot)$ is the indicator function which takes value 1 when the expression is true and 0 otherwise. Given X and Y, the sets $\{j_1, \ldots, j_L\}$ and $\{g_1, \ldots, g_L\}$ together uniquely define an alignment $\mathcal{A}$ (Remark S1.1). We can define a map from the latent state path W to a pairwise alignment $\mathcal{A}$ of X and Y.

Definition 4.3 (From latent states to biological alignments) *Given $W \sim p_{\text{MuE}}(w|X, Y)$, let $g_l := \mathbb{I}(W_l > M)$ and $j_l := W_l - Mg_l + \sum_{l'=1}^{l-1} g_{l'}$, for $l \in \{1, \ldots, L\}$. Note that this map is invertible.*

Under this definition, when $g_l = 0$, the letter Y_l is generated based on a letter X_{W_l} in the MuE, and Y_l and X_m are placed in the same column of the pairwise alignment $\mathcal{A}$; when $g_l = 1$, however, Y_l does not depend on X at all (it depends on c instead) and $\mathcal{A}^{(x)}_{j_l}$ has the gap symbol (Figure 2C).

A zoo of probabilistic and non-probabilistic methods have been proposed for inferring biological sequence alignments from data. Here we show that many of the most widely used methods can be seen as special case examples of the MuE which use Definition 4.3 to convert from W to $\mathcal{A}$.[3]

Proposition 4.4 (Unified) *For different choices of parameters c, ℓ, $a^{(0)}$, and $a^{(t)}$, (1) the Thorne-Kishino-Felsenstein model (Thorne et al., 1991), (2) the profile HMM, and (3) the conditional distribution of a sequence Y given a sequence X under the pair HMM (Durbin et al., 1998) are all special cases of the distribution $\text{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$, with a state-specific probability of the Markov chain terminating at each step. For another choice of parameters, the maximum a posteriori estimator $\hat{w} := \text{argmax}_w p_{\text{MuE}}(Y|X, w)$ corresponds to the Needleman-Wunsch alignment.*

See Section S1.1 for a proof. In the context of the profile HMM, point estimates of the latent state paths $W_1, \ldots, W_N$ associated with each observed sample $Y_1, \ldots, Y_N$ are used to construct a multiple sequence alignment of the dataset by

[3] So far we have not specified a model for the length L of the sequence Y. In the following proposition, we assume that there is some probability of the latent Markov chain terminating after each step l, and that this probability depends on the current state W_l.

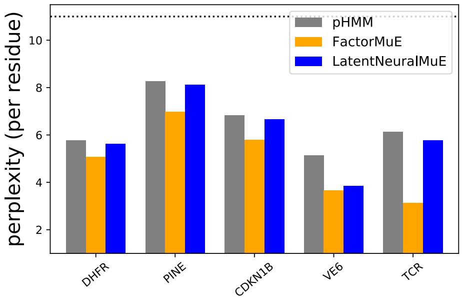


Figure 4. Predictive performance on a randomly heldout test set. Dotted line marks theoretically expected performance of the substitution matrix BLOSUM62 as a reference point (Section S4).

effectively merging pairwise alignments; the same methods can be used for MuE observation models as well (Figure 2D; Section S1.2). MuE observation models can thus also be used as probabilistic methods for inferring MSAs.

The MuE offers not only a unified but also a comprehensive framework in the sense that HMMs which fail to satisfy Constraint 2.2 cannot be interpreted, using Definition 4.3, as biological alignments (proof in Section S1.3):

Proposition 4.5 (Comprehensive) *Consider the setup of Definition 4.3 and assume each latent state $k \in \{1, \ldots, K\}$ of the MuE is Markov accessible under $a^{(0)}$ and $a^{(t)}$ (meaning that it can be reached with non-zero probability). Condition 2.2 is both necessary and sufficient to guarantee that with probability 1, W defines a valid pairwise alignment of X and Y via Definition 4.3.*

4.3. Comparison to natural language models

Latent alignment models are commonly used in natural language processing, often in combination with soft or hard attention methods for inference (Deng et al., 2018). We can compare the MuE directly with a classic latent alignment model for statistical translation. The Vogel et al. (1996) model takes the form of a MuE model where X and Y are sentences in different languages, except that Condition 2.2 is violated (Section S1.4). As a result latent alignments are allowed to "double back" and rearrange the ordering of words in the regressor sentence X to generate Y. Beyond biology, restrictions like Condition 2.2 may be of interest in application areas with structured text, e.g. symbolic math or software testing.

5. Experiments

5.1. Predictive performance

In this section we empirically compare the predictive performance of MuE observation models to a standard method with the same latent alignment structure, namely the profile HMM (pHMM) (Proposition 4.4). We examined five datasets of protein families, ranging in size from 1,000 to 10,000 sequences (Section S6). Four were taken from non-redundant sequence databases: sequences similar to dihydrofolate reductase (DHFR; a widely conserved enzyme), serine recombinase (PINE; a tool for genomic engineering), cyclin dependent kinase inhibitor 1B (CDKN1B/p27; a cell cycle inhibitor) and the human papillomavirus E6 protein (VE6; an oncogenic viral protein) (Hopf et al., 2017; Toth-Petroczy et al., 2016; Tamarozzi & Giuliatti, 2018). Two of these datasets (CDKN1B and VE6) consist of proteins with regions classified as "disordered" for which MSAs are typically considered especially untrustworthy. The final dataset consists of human T-cell receptor (TCR) sequences from a healthy donor obtained using single cell sequencing, for which multiple sequence alignments are similarly considered untrustworthy (Section S7).

We extended probabilistic PCA and VAE models using the MuE observation distribution; we refer to these models as "FactorMuE" and "LatentNeuralMuE" respectively (model architectures are detailed in Section S2). We used stochastic variational inference, estimating the ELBO gradient using automatic differentiation, the reparameterization trick, and an inference network, and optimizing with Adam (Section S3) (Kucukelbir et al., 2017; Kingma & Welling, 2013; Rezende et al., 2014; Kingma & Ba, 2015). We evaluated model performance on a randomly held out 10% of sequences, quantified in terms of per residue (that is, per letter) perplexity (Section S4). The results, summarized in Figure 4, show that FactorMuE models offer a consistent and large improvement over the standard pHMM model in every dataset, with an average change in perplexity of -1.50 (log Bayes factor $> 10^3$ across all datasets). Particularly dramatic improvements are seen in the TCR dataset, where perplexity falls by more than 3 (log Bayes factor $> 10^4$). Meanwhile, the more complex LatentNeuralMuE model also improves over the pHMM in each dataset and overall (average perplexity change -0.42), but underperforms relative to the simpler FactorMuE model.

5.2. Learning complex biological structure

We examined in further detail the FactorMuE model learned from the TCR dataset. T-cell receptors are made up of two separate amino acid chains, α and β, which each develop according to a complex process of genome rearrangement termed V(D)J recombination, in which different V, D and J segments in the genome are, with some randomness and additional mutations, joined together with a constant region to produce a complete sequence (Figure 5A). We cross-referenced the latent representations of each sequence recorded in the dataset against supervised annotations of its segment types (Section S6). We found that the latent space is divided evenly in two, with one side containing TCRα sequences and one side TCRβ sequences (Figure 5B left). Each side contains clusters, which correspond with the type

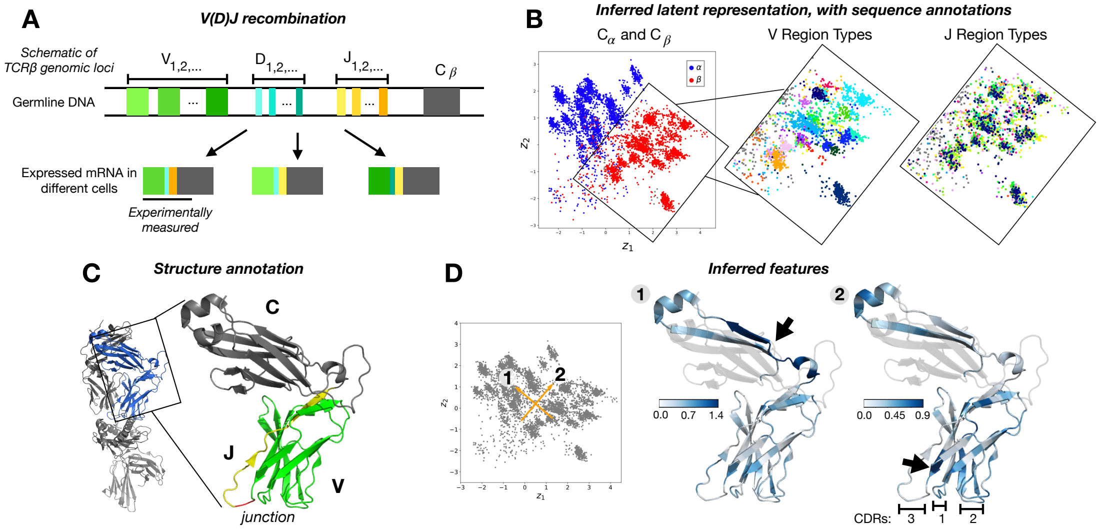


Figure 5. (A) Illustration of the TCRβ genomic locus; the TCRα locus is analogous, with C_α in place of C_β and no D segments (Abbas et al. (2018), Figure 8.7). (B) Inferred latent space representation of the TCR dataset, colored according to supervised annotations. Left: C_α and C_β chains. Middle: V types, $V_2, \ldots V_{30}$ (detailed legend in Figure S5). Right: J subtypes, $J_{\text{1-1},\ldots,\text{2-7}}$ (detailed legend in Figure S5). (C) V (green), J (yellow) and constant C (gray) regions of the TCRβ chain in the reference structure PDB:2BNR, as well as V-J junction nucleotides (red) (Figure S5). (D) Projections ν of latent space vectors (left, in orange) into sequence space. Transparent areas correspond to the portion of the sequence that is not measured in the experiment. Arrows indicate peaks in ν.

of V segment found in each TCR sequence (Figure 5B middle). The shorter J segments are found uniformly distributed across their corresponding α or β half, reflecting their ability to recombine with different V segments (Figure 5B right). See Section S6 for further results.

We next examined features learned by the FactorMuE model. To visualize features we fixed the latent alignment variable w and projected changes in the latent representation into sequence space; this enables a similar model interpretation to MSA preprocessing methods, which also use a fixed (though precomputed) alignment. In particular, we calculated

$$\nu_l := \left[\sum_{b=1}^{B} \left(\mathbb{E}[Y_{l,b} | \hat{w}_{\text{ref}}, z_1] - \mathbb{E}[Y_{l,b} | \hat{w}_{\text{ref}}, z_0] \right)^2 \right]^{1/2} \quad (3)$$

where the expectation is with respect to the variational approximation to the posterior, z_0 and z_1 are the head and tail of a vector in the latent space, $\hat{w}_{\text{ref}}$ is the maximum *a posteriori* estimate of w based on a reference sequence, and $l \in \{1, \ldots, L_{\text{ref}}\}$ where L_{ref} is the length of the reference sequence. We plotted the vector ν on a TCR crystal structure for the reference sequence, and compared to a supervised annotation of the constant, V, D and J segments of the reference sequence (Figure 5CD). Consistent with the annotation of the latent representation, the vector normal to the hyperplane separating TCRα from TCRβ chains in the latent space (vector 1 in Figure 5D) primarily alters the sequence of the constant region, while the orthogonal vector (vector 2 in Figure 5D) primarily determines the sequence of the V segment. Along vector 2, the region of largest variation (the largest peak in ν_l) was the buried C-terminal end of the V segment, corresponding to the start of the CDR3 region, the key specificity-determining region of the receptor. Interestingly, even along vector 1 we observe high values of ν_l in the V segment, suggesting that there are systematic and heterogeneous differences between the V segment sequence distribution used in TCRα chains and in TCRβ chains (see Section S6 for further analysis).

5.3. Evolutionary forecasting

We explored a novel application of generative probabilistic sequence models, evolutionary forecasting, which takes advantage of the capacity of MuE observation models to predict future sequences. Influenza A is responsible for an estimated 500,000 deaths a year and is an ongoing pandemic threat (Iuliano et al., 2018). It is also a model organism for understanding the dynamics of rapidly evolving pathogens, and forecasting its evolution is crucial in preparing vaccines and designing therapeutics (Luksza & Lässig, 2014; Laursen & Wilson, 2013). Previous forecasting methods have focused on predicting the relative fitness of existing strains in future years (Luksza & Lässig, 2014; Bush et al., 1999), or the antigenic properties of newly emerged strains (Neher et al., 2016). We instead predict the full amino acid sequence of the HA1 protein, the primary site of interaction with the immune system (Wiley et al., 1981). From the GISAID database we constructed a training set of influenza A(H3N2) HA1 sequences collected from patient samples from 1968 through 2013, and evaluated our predictions on sequences collected from 2014 through October 2019 (420 out of 2,042 sequences held out, 21% of the dataset) (Sec-

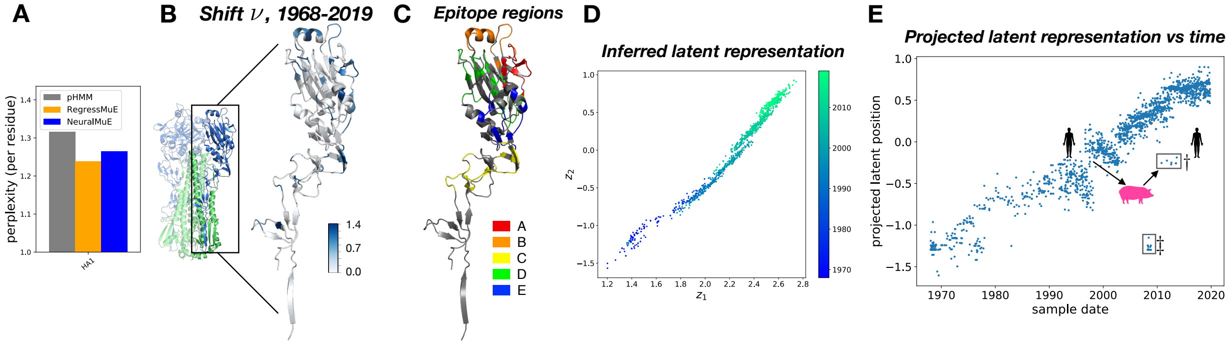


Figure 6. (A) Predictive performance measured by heldout per residue perplexity; models are trained on data from 1968-2013, tested on 2014-2020. (B) Magnitude of the shift in amino acid preference over time ν, for the RegressMuE, projected onto a reference HA1 structure (PDB:4O5N). The full hemagglutinin protein is shown on the left. (C) Classical epitope regions of the HA1 protein. (D) Inferred latent representation from a FactorMuE model, with sequences colored by the time at which the sample was collected (Section S7). (E) Y-axis: orthogonal projection of the latent representation of each sequence onto the least squares fit line relating z_1 and z_2. X-axis: time at which each sample was collected. Two clusters of outliers are marked by † and ‡.

tion S7) (Shu & McCauley, 2017). Insertions and deletions are considered rare, though not absent, in patient samples, so this dataset also offers an opportunity to evaluate MuE observation models in a distinct regime from that considered previously in Section 5.1.

As a benchmark we again used the pHMM, which can capture the observation that there exist key highly variable sites in the HA1 protein, an underlying motivation behind previous prediction methods such as Bush et al. (1999). We then incorporated sequence collection time as a covariate in new MuE observation models, using a linear regression model ("RegressMuE") and a neural network ("NeuralMuE") with MuE observation distributions (Section S2). The pHMM achieves a per residue perplexity of 1.32 and the RegressMuE improves this to 1.24 (log Bayes factor $> 10^3$; Figure 6A). This per residue perplexity difference corresponds to a factor of $\sim 10^{10}$ improvement in per sequence perplexity. The NeuralMuE has similar per residue perplexity (1.26) to the RegressMuE.

Next we investigated in detail what the model can tell us about how HA1 proteins have changed over time. We computed the magnitude of the shift in amino acid preference from 1968 to 2019 inferred by the model, with the latent MuE alignment variable kept fixed (quantified as ν_l, defined analogously to Equation 3 with times t_0 and t_1 replacing latent representations z_0 and z_1) (Figure 6B; Section S7). We found that sites with a large shift are often associated with antigenicity, consistent with the hypothesis that immune evasion is a key driver of influenza evolution. Residues that make up the classical epitope regions A-E of influenza show significantly larger shifts as compared to residues outside these regions (mean ν_l of 0.54 in epitopes A-E versus 0.09 in non-epitope sites, one sided Mann-Whitney U test $p < 10^{-18}$; Figures 6C and S10) (Wiley et al., 1981; Muñoz & Deem, 2005). The same observation holds for residues identified as key determinants of immune escape in recent high-throughput mutational antigenic profiling experiments (mean ν_l of 0.80 in sites with antigenic selection versus 0.24 elsewhere, one sided Mann-Whitney U test $p < 10^{-4}$; Section S7) (Lee et al., 2019).

The latent space representation of the influenza HA1 dataset learned by the FactorMuE model shows the data falling approximately along a line (Figure 6D; Section S7). The position of a sequence along this line is linearly proportional to the time at which the sequence was collected, though this information was not included in the model (correlation coefficient $\rho = 0.94$; Figure 6E) (Novembre & Stephens, 2008). Two clusters of outliers violate the proportionality rule. The first (marked by ‡) originated from mis-annotated entries in the GISAID database (Section S7). The second cluster (marked by †) appears in the early 2010s, but the latent representation of these sequences is close to that of sequences from the mid-1990s to early 2000s. Among this cluster of sequences, the ones that have been fully annotated were all collected from an outbreak in the United States of A(H3N2)v triple-reassortant viruses containing matrix protein genes from pandemic A(H1N1)pdm09. In 1998, A(H3N2)-derived viruses jumped from humans to swine, causing a large outbreak among swine, before recombining with other strains to produce this A(H3N2)v outbreak among humans in the 2010s (Jhung et al., 2013; Skowronski et al., 2012). The epidemiological history is consistent with our unsupervised latent representation, which shows that the cluster of outliers appearing in 2010-2013 most closely matches human samples last seen around 2000.

6. Discussion

MSAs are a powerful tool for understanding sequence evolution, but MSA preprocessing leads to statistical pathologies in generative models. MuE observation models offer a drop-in alternative to MSA preprocessing that does not abandon the underlying biological ideas that have made MSAs so successful. We hope that the MuE will enable rigorous application of a wide variety of new models and methodologies to biological sequence data.

Acknowledgments

We wish to thank Chris Sander, John Ingraham, Elizabeth Wood, Smita Krishnaswamy, Alan Amin, Will Grathwohl, Fritz Obermeyer, and members of the Marks lab for discussion and suggestions. E.N.W. is supported by the Fannie and John Hertz Foundation. D.S.M is supported by the Chan Zuckerberg Initiative.

References

Abbas, A. K., Lichtman, A. H., and Pillai, S. *Cellular and Molecular Immunology*. Elsevier, ninth edition, 2018.

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, 16(12):1315–1322, December 2019.

Bush, R. M., Bender, C. A., Subbarao, K., Cox, N. J., and Fitch, W. M. Predicting the evolution of human influenza A. *Science*, 286(5446):1921–1925, December 1999.

Dawid, A. P. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–292, 1984.

Dawid, A. P. Posterior model probabilities. In Bandyopadhyay, P. S. and Forster, M. R. (eds.), *Philosophy of Statistics*, volume 7, pp. 607–630. North-Holland, Amsterdam, January 2011.

Deng, Y., Kim, Y., Chiu, J., Guo, D., and Rush, A. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems*, pp. 9735–9747, 2018.

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A., Sonnhammer, E. L. L., Hirsh, L., Paladin, L., Piovesan, D., Tosatto, S. C. E., and Finn, R. D. The pfam protein families database in 2019. *Nucleic Acids Res.*, 47(D1):D427–D432, January 2019.

Felsenstein, J. *Inferring phylogenies*. Sinauer associates, Sunderland, MA, 2004.

Frazer, J., Notin, P., Dias, M., Gomez, A., Brock, K., Gal, Y., and Marks, D. Large-scale clinical interpretation of genetic variants using evolutionary data and deep learning. December 2020.

Hadfield, J., Megill, C., Bell, S. M., Huddleston, J., Potter, B., Callender, C., Sagulenko, P., Bedford, T., and Neher, R. A. Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*, 34(23):4121–4123, December 2018.

Holmes, I. H. Solving the master equation for indels. *BMC Bioinformatics*, 18(1):255, May 2017.

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

Iuliano, A. D., Roguski, K. M., Chang, H. H., Muscatello, D. J., Palekar, R., Tempia, S., Cohen, C., Gran, J. M., Schanzer, D., Cowling, B. J., Wu, P., Kyncl, J., Ang, L. W., Park, M., Redlberger-Fritz, M., Yu, H., Espenhain, L., Krishnan, A., Emukule, G., van Asten, L., Pereira da Silva, S., Aungkulanon, S., Buchholz, U., Widdowson, M.-A., Bresee, J. S., and Global Seasonal Influenza-associated Mortality Collaborator Network. Estimates of global seasonal influenza-associated respiratory mortality: a modelling study. *Lancet*, 391(10127):1285–1300, March 2018.

Jhung, M. A., Epperson, S., Biggerstaff, M., Allen, D., Balish, A., Barnes, N., Beaudoin, A., Berman, L., Bidol, S., Blanton, L., Blythe, D., Brammer, L., D'Mello, T., Danila, R., Davis, W., de Fijter, S., Diorio, M., Durand, L. O., Emery, S., Fowler, B., Garten, R., Grant, Y., Greenbaum, A., Gubareva, L., Havers, F., Haupt, T., House, J., Ibrahim, S., Jiang, V., Jain, S., Jernigan, D., Kazmierczak, J., Klimov, A., Lindstrom, S., Longenberger, A., Lucas, P., Lynfield, R., McMorrow, M., Moll, M., Morin, C., Ostroff, S., Page, S. L., Park, S. Y., Peters, S., Quinn, C., Reed, C., Richards, S., Scheftel, J., Simwale, O., Shu, B., Soyemi, K., Stauffer, J., Steffens, C., Su, S., Torso, L., Uyeki, T. M., Vetter, S., Villanueva, J., Wong, K. K., Shaw, M., Bresee, J. S., Cox, N., and Finelli, L. Outbreak of variant influenza A(H3N2) virus in the united states. *Clin. Infect. Dis.*, 57(12):1703–1712, December 2013.

Johnson, L. S., Eddy, S. R., and Portugaly, E. Hidden markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics*, 11:431, August 2010.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kingma, D. P. and Welling, M. Auto-Encoding variational bayes. December 2013.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. Automatic differentiation variational inference. *J. Mach. Learn. Res.*, 18(14):1–45, January 2017.

Laursen, N. S. and Wilson, I. A. Broadly neutralizing antibodies against influenza viruses. *Antiviral Res.*, 98(3): 476–483, June 2013.

Lee, J. M., Eguia, R., Zost, S. J., Choudhary, S., Wilson, P. C., Bedford, T., Stevens-Ayers, T., Boeckh, M., Hurt, A. C., Lakdawala, S. S., Hensley, S. E., and Bloom, J. D. Mapping person-to-person variation in viral mutations that escape polyclonal serum targeting influenza hemagglutinin. *Elife*, 8, August 2019.

Luksza, M. and Lässig, M. A predictive fitness model for influenza. *Nature*, 507(7490):57–61, March 2014.

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C. Protein 3D structure computed from evolutionary sequence variation. *PLoS One*, 6(12):e28766, December 2011.

Muñoz, E. T. and Deem, M. W. Epitope analysis for influenza vaccine design. *Vaccine*, 23(9):1144–1148, January 2005.

Neher, R. A., Bedford, T., Daniels, R. S., Russell, C. A., and Shraiman, B. I. Prediction, dynamics, and visualization of antigenic phenotypes of seasonal influenza viruses. *Proc. Natl. Acad. Sci. U. S. A.*, 113(12):E1701–9, March 2016.

Novembre, J. and Stephens, M. Interpreting principal component analyses of spatial population genetic variation. *Nat. Genet.*, 40(5):646–649, May 2008.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Riesselman, A., Shin, J.-E., Kollasch, A., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A., and Marks, D. Accelerating protein design using autoregressive generative models. 2019.

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

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. April 2019.

Rush, A. M. Torch-Struct: Deep structured prediction library. February 2020.

Russ, W. P., Figliuzzi, M., Stocker, C., Barrat-Charlaix, P., Socolich, M., Kast, P., Hilvert, D., Monasson, R., Cocco, S., Weigt, M., and Ranganathan, R. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369:440–445, 2020.

Särkkä, S. and García-Fernández, Á. F. Temporal parallelization of bayesian smoothers. *IEEE Trans. Automat. Contr.*, 66(1):299–306, 2020.

Shu, Y. and McCauley, J. GISAID: Global initiative on sharing all influenza data - from vision to reality. *Euro Surveill.*, 22(13), March 2017.

Skowronski, D. M., Janjua, N. Z., De Serres, G., Purych, D., Gilca, V., Scheifele, D. W., Dionne, M., Sabaiduc, S., Gardy, J. L., Li, G., Bastien, N., Petric, M., Boivin, G., and Li, Y. Cross-reactive and vaccine-induced antibody to an emerging swine-origin variant of influenza a virus subtype H3N2 (H3N2v). *J. Infect. Dis.*, 206(12):1852–1861, December 2012.

Tamarozzi, E. R. and Giuliatti, S. Understanding the role of intrinsic disorder of viral proteins in the oncogenicity of different types of HPV. *Int. J. Mol. Sci.*, 19(1), January 2018.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. CLUSTAL w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22(22):4673–4680, November 1994.

Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F., and Higgins, D. G. The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Res.*, 25(24): 4876–4882, December 1997.

Thorne, J. L., Kishino, H., and Felsenstein, J. An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 33(2):114–124, August 1991.

Toth-Petroczy, A., Palmedo, P., Ingraham, J., Hopf, T. A., Berger, B., Sander, C., and Marks, D. S. Structured states of disordered proteins from genomic sequences. *Cell*, 167 (1):158–170.e12, September 2016.

Tran, D., Hoffman, M., Moore, D., Suter, C., Vasudevan, S., Radul, A., Johnson, M., and Saurous, R. A. Simple, distributed, and accelerated probabilistic programming. In *Neural Information Processing Systems*, 2018.

Van Noorden, B. Y. R., Maher, B., and Nuzzo, R. Nature explores the most-cited research of all time. *Nature*, 514: 550–553, 2014.

Vapnik, V. N. An overview of statistical learning theory. *IEEE Trans. Neural Netw.*, 10(5):988–999, 1999.

Vogel, S., Ney, H., and Tillmann, C. HMM-based word alignment in statistical translation. In *Proc. of the 16th International Conference on Computational Linguistics (COLING '96)*, pp. 836–841, 1996.

Weinstein, E. N., Frazer, J., and Marks, D. S. Deconvolving fitness and phylogeny in generative models of molecular evolution. In *Learning Meaningful Representations of Life Workshop at Neural Information Processing Systems*, 2020.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688. people.ee.duke.edu, 2011.

Wilburn, G. W. and Eddy, S. R. Remote homology search with hidden potts models. June 2020.

Wiley, D. C., Wilson, I. A., and Skehel, J. J. Structural identification of sites of hong kong influenza and their involvement in antigenic variation. *Nature*, 289, 1981.

Wu, M., Chatterji, S., and Eisen, J. A. Accounting for alignment uncertainty in phylogenomics. *PLoS One*, 7 (1):e30288, January 2012.

Algorithm 1: Pairwise alignment construction

```
input  : $(j_1, \dots, j_L)$ and $(g_1, \dots, g_L)$ and $X$ and $Y$
output: $\mathcal{A}$
$n = 0$                    (indexes position in overall alignment.);
$m = 0$                    (indexes position in sequence $X$);
Iterate until each letter in both $X$ and $Y$ has been placed in $\mathcal{A}$;
while $m < M$ or $n < j_L$ do
    $n = n + 1$;
    if $\exists l : n = j_l$ then
        $\mathcal{A}^{(y)}_n = Y_l$          (by definition of $j_l$);
        if $g_l = 1$ then
            $\mathcal{A}^{(x)}_n =$ -          (by definition of $g_l$);
        else
            $m = m + 1$;
            $\mathcal{A}^{(x)}_n = X_m$   (by definitions of $g_l$ and $\mathcal{A}^{(x)}$; letters of $X$ must be in order);
        end
    else
        $\mathcal{A}^{(y)}_n =$ -          (by definition of $j_l$);
        $m = m + 1$;
        $\mathcal{A}^{(x)}_n = X_m$   (by definition of $\mathcal{A}$; each column of $\mathcal{A}$ must have at least one letter);
    end
end
```

S1 Theory

In this section we describe our theoretical results related to biological alignments. It is useful for understanding the following results to have in mind a particular example to illustrate the definitions in the main text.

Sequences	*Pairwise alignment $\mathcal{A}$*	*j and g representation*
$Y =$ `ATG`	$\mathcal{A}^{(y)} =$ `A--TG-`	$(j_1, \dots, j_L) = (1, 4, 5)$
$X =$ `TCTG`	$\mathcal{A}^{(x)} =$ `-TCT-G`	$(g_1, \dots, g_L) = (1, 0, 1)$

It is also useful to define $m_l := W_l - Mg_l$, which indexes the position within the first or second block of states. For the example we have, $(m_1, \dots, m_L) = (1, 3, 4)$.

Remark S1.1. *Given sequences X and Y of length M and L respectively, $(j_1, \dots, j_L)$ and $(g_1, \dots, g_L)$ uniquely define a pairwise alignment $\mathcal{A}$.*

Proof. Applying Definition 4.2 and the definitions of $(j_1, \dots, j_L)$ and $(g_1, \dots, g_L)$ iteratively to each column of the alignment leads to the construction of $\mathcal{A}$ in Algorithm 1. □

S1.1 Proof of Proposition 4.4

To prove the result, we will examine each existing model individually; exact specifications and assumptions for each model are provided in their corresponding section. The probability of the Markov chain terminating given that it is at a state k is denoted $t^{(t)}_k$, and the probability of the

Markov chain terminating initially (that is, of the Markov chain taking zero steps) is denoted $t^{(0)}$. Without loss of generality, we will write transition probabilities $a^{(t)}$ and $a^{(0)}$ without conditioning on the Markov chain not terminating, i.e. $\sum_{k'} a^{(t)}_{k,k'} + t^{(t)}_k = 1$. The conditional transition probability can of course be computed as $a^{(t)}_{k,k'}/(1-t^{(t)}_k)$. In general, we will also index latent states k of the MuE by their corresponding (m, g) value where (in line with the definition of g_l and m_l) $g = \mathbb{I}(k > M)$ and $m = k - Mg$; we will use k and (m, g) interchangeably for any given state.

S1.1.1 Thorne-Kishino-Felsenstein

The Thorne-Kishino-Felsenstein (TKF) model is a continuous-time stochastic process model of sequence evolution that satisfies detailed balance (Thorne et al., 1991).

Statement Let X be a one-hot encoding of the initial sequence. Let $D = B$ and let π be the TKF parameter corresponding to the equilibrium probability of each letter. For all $m \in \{1, ..., M\}$ and $b \in \{1, ..., B\}$, assign

$$c_{m,b} := \pi_b. \tag{S1}$$

Let $\lambda > 0$ and $\mu > 0$ be the TKF indel rate parameters, with $\lambda < \mu$, and let $\tau > 0$ be the divergence time parameter. Define

$$\beta(\tau) := \frac{1 - e^{-(\mu-\lambda)\tau}}{\mu - \lambda e^{-(\mu-\lambda)\tau}}. \tag{S2}$$

Define the transition matrix and termination probability as

$$a^{(t)}_{k,k'} := \begin{cases} [\mu\beta(\tau)]^{m'-m-1+g} e^{-\mu\tau}[1 - \lambda\beta(\tau)] & \text{if } m - g < m' < M + 1 \text{ and } g' = 0 \\ \lambda\beta(\tau) & \text{if } m - g = m' - 1 \text{ and } g' = 1 \\ [\mu\beta(\tau)]^{m'-m-2+g}[1 - e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)] & \text{if } m - g < m' - 1 \text{ and } g' = 1 \\ 0 & \textit{otherwise.} \end{cases} \tag{S3}$$

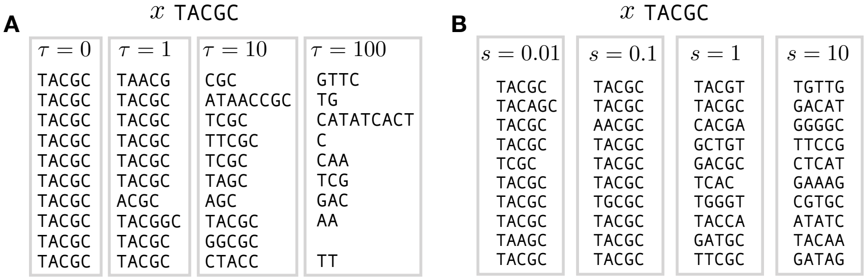
$$t^{(t)}_k := [1 - \lambda\beta(\tau)][\mu\beta(\tau)]^{M-m+g} \tag{S4}$$

The initial transition vector follows the same form, and can be written as $a^{(0)}_k := a^{(t)}_{0,k}$, and the initial termination probability can be written $t^{(0)} := t^{(t)}_0$ (i.e. they match Equations S3 and S4 with $(m, g) = (0, 0)$ plugged in). Let $s > 0$ be the TKF substitution rate parameter and define the substitution matrix

$$\ell_{b,b'} := \begin{cases} e^{-s\tau} + \pi_{b'}(1 - e^{-s\tau}) & \text{if } b = b' \\ \pi_{b'}(1 - e^{-s\tau}) & \text{if } b \neq b' \end{cases} \tag{S5}$$

With these definitions, $Y \sim \text{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$ is the distribution of the Thorne-Kishino-Felsenstein model after the sequence X evolves for time τ. Note that the limit $\tau \to 0$ is the no-mutation limit. Figure S1 illustrates samples from the TKF model with changing parameters.

Proof We will show that the joint probability of W and Y under the MuE distribution is identical to the joint probability of the corresponding alignment pairwise alignment and Y under the TKF model. To start, we systematically enumerate state transitions in the MuE model and compute the corresponding probability factor under the TKF alignment scoring system. Our alignment notation in this section follows the original paper. "X" represents a residue and "–" a gap. "." represents the "immortal link" in the model, the start of the sequence. We use "$" as a termination


A
x TACGC
B
x TACGC

$\tau = 0$	$\tau = 1$	$\tau = 10$	$\tau = 100$
TACGC	TAACG	CGC	GTTC
TACGC	TACGC	ATAACCGC	TG
TACGC	TACGC	TCGC	CATATCACT
TACGC	TACGC	TTCGC	C
TACGC	TACGC	TCGC	CAA
TACGC	TACGC	TAGC	TCG
TACGC	ACGC	AGC	GAC
TACGC	TACGGC	TACGC	AA
TACGC	TACGC	GGCGC	
TACGC	TACGC	CTACC	TT

$s = 0.01$	$s = 0.1$	$s = 1$	$s = 10$
TACGC	TACGC	TACGT	TGTTG
TACAGC	TACGC	TACGC	GACAT
TACGC	AACGC	CACGA	GGGGC
TACGC	TACGC	GCTGT	TTCCG
TCGC	TACGC	GACGC	CTCAT
TACGC	TACGC	TCAC	GAAAG
TACGC	TGCGC	TGGGT	CGTGC
TACGC	TACGC	TACCA	ATATC
TAAGC	TACGC	GATGC	TACAA
TACGC	TACGC	TTCGC	GATAG

Figure S1: **Samples from the Thorne-Kishino-Felsenstein model.** Initial sequence TACGC, $\mu = 0.02$, and $\lambda = 0.01$. A. $s = 0.01$ and varying τ. B. $\tau = 1$ and varying s.

symbol. Following the original paper, we define, for $\nu \in \{1, 2, \ldots\}$,

$$\begin{aligned} p_\nu(\tau) &:= e^{-\mu\tau}[1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \\ p'_0(\tau) &:= \mu\beta(\tau) \\ p'_\nu(\tau) &:= [1 - e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \\ p''_\nu(\tau) &:= [1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \end{aligned} \tag{S6}$$

The TKF model assigns probabilities to a pairwise alignment based on the pattern of residues and gaps; we will break down possible pairwise alignments into chunks corresponding to state transitions under the MuE and compute the probability factor that they contribute under the TKF scoring system. When enumerating transitions in the Markov model we put a "|" symbol to the right of the residue we are transitioning from.

1. Transitioning from a state $(m, 0)$ to a state $(m' > m, 0)$ gives the probability factor $[p'_0(\tau)]^{m'-m-1}p_1(\tau) = [\mu\beta(\tau)]^{m'-m-1}e^{-\mu\tau}[1 - \lambda\beta(\tau)]$ according to the TKF scoring system.

   ```
   X | X ... X X
   X | - ... - X
   ```

2. Transitioning from $(m, 1)$ to $(m' \geq m, 0)$ gives the factor $[p'_0(\tau)]^{m'-m}p_1(\tau) = [\mu\beta(\tau)]^{m'-m}e^{-\mu\tau}[1-\lambda\beta(\tau)]$.

   ```
   - | X ... X X
   X | - ... - X
   ```

3. Transitioning from $(m, 1)$ to $(m, 1)$, situation 1. This gives a factor $\frac{p_{\nu+2}(t)}{p_{\nu+1}(t)} = \lambda\beta(\tau)$.

   ```
   X - ... - | -
   X X ... X | X
   ```

4. Transitioning from $(m, 1)$ to $(m, 1)$, situation 2. This gives a factor $\frac{p'_{\nu+2}(\tau)}{p'_{\nu+1}(\tau)} = \lambda\beta(\tau)$

   ```
   X - ... - | -
   - X ... X | X
   ```

5. Transitioning from $(m, 0)$ to $(m + 1, 1)$. This gives a factor $\frac{p_2(\tau)}{p_1(\tau)} = \lambda\beta(\tau)$.

```
X | -
X | X
```

6. Transitioning from $(m, 0)$ to $(m' > m + 1, 1)$. This gives a factor $[p_0'(\tau)]^{m'-m-2}p_1'(\tau) = [\mu\beta(\tau)]^{m'-m-2}[1 - e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)]$.

```
X | X ... X -
X | - ... - X
```

7. Transitioning from $(m, 1)$ to $(m' > m, 1)$. This gives a factor $[p_0'(\tau)]^{m'-m-1}p_1'(\tau) = [\mu\beta(\tau)]^{m'-m-1}[1-e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)]$.

```
- | X ... X -
X | - ... - X
```

8. Terminating after $(m, 0)$. This gives a factor $[p_0'(\tau)]^{M-m} = [\mu\beta(\tau)]^{M-m}$.

```
X | X ... X $
X | - ... - $
```

9. Terminating after $(m, 1)$. This gives a factor $[p_0'(\tau)]^{M+1-m} = [\mu\beta(\tau)]^{M+1-m}$.

```
- | X ... X $
X | - ... - $
```

10. Initial transition to $(1, 1)$. This gives a factor $p_2''(\tau) = p_1''(\tau)\lambda\beta(\tau) = [1 - \lambda\beta(\tau)][\lambda\beta(\tau)]$.

```
. | -
. | X
```

11. Initial transition to $(m, 0)$. This gives a factor $p_1''(\tau)[p_0'(\tau)]^{m-1}p_1(\tau) = [1-\lambda\beta(\tau)][\mu\beta(\tau)]^{m-1}e^{-\mu\tau}[1-\lambda\beta(\tau)]$.

```
. | X ... X X
. | - ... - X
```

12. Initial transition to $(m > 1, 1)$. This gives a factor $p_1''(\tau)[p_0'(\tau)]^{m-2}p_1'(\tau) = [1-\lambda\beta(\tau)][\mu\beta(\tau)]^{m-2}[1-e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)]$.

```
. | X ... X -
. | - ... - X
```

13. Terminating in the first step. This gives a factor $[p_0'(\tau)]^M = [\mu\beta(\tau)]^M$.

```
. | X ... X $
. | - ... - $
```

Compiling these results yields the probability factors associated with each transition between states

$$\begin{gathered}(m,g) \to (m',g') : \begin{cases} [\mu\beta(t)]^{m'-m-1+g} e^{-\mu t}[1-\lambda\beta(t)] \\ \qquad\quad \text{if } m-g < m' < M+1 \text{ and } g'=0 \\ \lambda\beta(t) \;\; \text{if } m-g = m'-1 \text{ and } g'=1 \\ [\mu\beta(t)]^{m'-m-2+g}[1-e^{-\mu t}-\mu\beta(t)][1-\lambda\beta(t)] \\ \qquad\quad \text{if } m-g<m'-1 \text{ and } g'=1 \\ 0 \qquad\quad \textit{otherwise}\end{cases} \\ (m,g)\to \textit{termination} : [\mu\beta(t)]^{M-m+g}\end{gathered} \tag{S7}$$

And with each initial transition

$$\begin{gathered}\textit{initial} \to (m,g) : \begin{cases} [1-\lambda\beta(t)][\mu\beta(t)]^{m-1}e^{-\mu t}[1-\lambda\beta(t)] & \text{if } 0<m<M+1 \text{ and } g=0 \\ [1-\lambda\beta(t)]\lambda\beta(t) & \text{if } m=1 \text{ and } g=1 \\ [1-\lambda\beta(t)][\mu\beta(t)]^{m-2}[1-e^{-\mu t}-\mu\beta(t)][1-\lambda\beta(t)] & \text{if } 1<m \text{ and } g=1 \\ 0 & \textit{otherwise}\end{cases} \\ \textit{initial} \to \textit{termination} : [1-\lambda\beta(t)][\mu\beta(t)]^{M}\end{gathered} \tag{S8}$$

However, these are unnormalized probability factors, not complete probabilities. Note that every alignment will include a factor $[1-\lambda\beta(t)]$, which in the original TKF description is associated with the initial transition. However, if we instead rearrange this factor and assign it to the final transition we obtain the transition matrix given in Equation S3. We can check that this transition matrix normalized. From a state $(m,0)$, the total outward transition probability is one:

$$\begin{aligned}&\sum_{m'=m+1}^{M}[\mu\beta]^{m'-m-1}e^{-\mu\tau}[1-\lambda\beta]+\lambda\beta+\sum_{m'=m+2}^{M+1}[\mu\beta]^{m'-m-2}[1-e^{-\mu\tau}-\mu\beta][1-\lambda\beta]+[\mu\beta]^{M-m}(1-\lambda\beta)\\ &\qquad=\frac{1-(\mu\beta)^{M-m}}{1-\mu\beta}[1-e^{-\mu\tau}-\mu\beta+e^{-\mu\tau}][1-\lambda\beta]+\lambda\beta+[\mu\beta]^{M-m}(1-\lambda\beta)\\ &\qquad=1-(\mu\beta)^{M-m}[1-\lambda\beta]+[\mu\beta]^{M-m}(1-\lambda\beta)\\ &\qquad=1.\end{aligned} \tag{S9}$$

The same expression holds for the initial transition, plugging in $m=0$. From $(m,1)$, we have

$$\begin{aligned}&\sum_{m'=m}^{M}[\mu\beta]^{m'-m}e^{-\mu\tau}[1-\lambda\beta]+\lambda\beta+\sum_{m'=m+1}^{M+1}[\mu\beta]^{m'-m-1}[1-e^{-\mu\tau}-\mu\beta][1-\lambda\beta]+[\mu\beta]^{M+1-m}(1-\lambda\beta)\\ &\qquad=\frac{1-(\mu\beta)^{M+1-m}}{1-\mu\beta}[1-e^{-\mu\tau}-\mu\beta+e^{-\mu\tau}][1-\lambda\beta]+\lambda\beta+[\mu\beta]^{M+1-m}(1-\lambda\beta)\\ &\qquad=1-(\mu\beta)^{M+1-m}[1-\lambda\beta]+[\mu\beta]^{M+1-m}(1-\lambda\beta)\\ &\qquad=1.\end{aligned} \tag{S10}$$

Conditional on the mth residue of X being aligned to the lth residue of Y (i.e. $w_l = m$), the TKF model specifies that the probability of y_l given x_m is $\sum_{b,b'} x_{m,b}\ell_{b,b'}y_{l,b'}$, which is identical to the probability under the MuE model. In the case where the lth residue of y is aligned to a gap (i.e. $g_l = 1$), the TKF model says the probability of choosing the specific base b is π_b, the equilibrium probability of the base. We can check that the MuE provides the same factor:

$$\begin{aligned} p_{\text{MuE}}(y_{l,b} = 1 | w, x, c, \ell) &= \sum_{b'} c_{m,b'} \ell_{b',b} \\ &= \pi_b e^{-s\tau} + (\pi_b)^2 (1 - e^{-s\tau}) + \sum_{b'' \neq b} \pi_{b''} \pi_b (1 - e^{-s\tau}) \qquad \text{(S11)} \\ &= \pi_b e^{-s\tau} + \pi_b (1 - e^{-s\tau}) = \pi_b. \end{aligned}$$

□

S1.1.2 Pair HMM

The pair HMM model generates pairwise alignments by switching between three states: (1) a state emitting residues in both X and Y (a match state), (2) a state emitting a residue in X and a gap in the alignment of Y, and (3) a state emitting a gap in the alignment of X and a residue in Y (Durbin et al. (1998), Chapter 4.1).

Statement Figure S2 shows a standard pair HMM diagram and state probabilities, with γ the probability of transitioning to a gap state, ϵ the probability of staying in a gap state, and κ the probability of the Markov chain terminating. We assume $1 - 2\gamma - \kappa \geq 0$ and $1 - \epsilon - \kappa \geq 0$. When in a match state, the pair HMM emits letters b and b' in the x and y sequences with probability $\psi_{b,b'}$; otherwise, in gap states, the probability of letter b in the non-gapped sequence is π_b.

Define the MuE transition matrix and termination probability vector as

$$a^{(t)}_{k,k'} := \begin{cases} \frac{1-2\gamma-\kappa}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1 = m' \leq M \text{ and } g = g' = 0 \\ \frac{\gamma\epsilon^{m'-m-2}(1-\epsilon-\kappa)}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1 < m' \leq M \text{ and } g = g' = 0 \\ \frac{\gamma}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1 = m' \leq M \text{ and } g = 0 \text{ and } g' = 1 \\ \frac{\gamma}{\gamma+\kappa} & \text{if } m+1 = m' = M+1 \text{ and } g = 0 \text{ and } g' = 1 \\ \frac{1-\epsilon-\kappa}{1-\kappa} & \text{if } m = m' \leq M \text{ and } g = 1 \text{ and } g' = 0 \\ \frac{\epsilon}{1-\kappa} & \text{if } m = m' \leq M \text{ and } g = g' = 1 \\ \frac{\epsilon}{\epsilon+\kappa} & \text{if } m = m' = M+1 \text{ and } g = g' = 1 \\ 0 & \textit{otherwise} \end{cases} \qquad \text{(S12)}$$

$$t^{(t)}_k := \begin{cases} \frac{\gamma\epsilon^{M-m-1}\kappa}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m < M \text{ and } g = 0 \\ \frac{\kappa}{\gamma+\kappa} & \text{if } m = M \text{ and } g = 0 \\ \frac{\kappa}{\epsilon+\kappa} & \text{if } m = M+1 \text{ and } g = 1 \\ 0 & \textit{otherwise} \end{cases} \qquad \text{(S13)}$$

The initial transition vector is defined by $a^{(0)}_k := a^{(t)}_{0,k}$ and initial termination probability is $t^{(0)} := t^{(t)}_0$. Define the substitution matrix

$$\ell_{b,b'} := \frac{\psi_{b,b'}}{\pi_b} \qquad \text{(S14)}$$

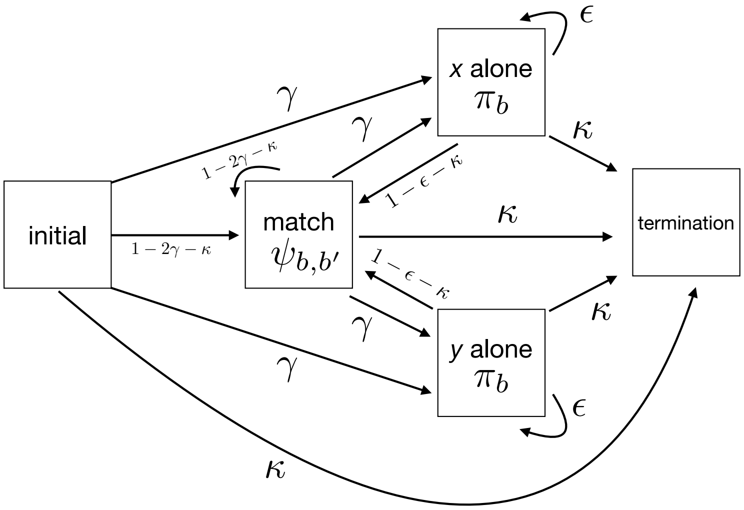


Figure S2: **Pair HMM state diagram.**

for all $b, b' \in \{1, ..., B\}$. Let the rows of the insertion matrix c be

$$c_m := (\ell^{-1})^\top \cdot \pi \tag{S15}$$

where ℓ^{-1} is the inverse of the substitution matrix, which is assumed to be an invertible matrix, and $\top$ indicates the matrix transpose.

With these definitions, $Y \sim \text{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$ is equivalent to the conditional distribution of Y given X under the pair HMM. Note that if $\gamma = 0$ and $\psi = \text{diag}(\pi)$ (the $B \times B$ matrix with diagonal entries π and all other entries 0) then we recover the no-mutation limit of the MuE distribution.

Proof We will show that the joint probability of W and Y under the MuE model is identical to the joint probability of the corresponding alignment and Y under the pair HMM, conditional on X. We start by enumerating all possible transitions between states of the MuE Markov chain and computing their probability under the pair HMM model without conditioning on X. Define $\omega^x_j := \mathbb{I}(\mathcal{A}^{(x)}_j \in \mathcal{B})$ and ω^y likewise. We use ω^x, ω^y notation to represent possible alignments, with the symbol "|" placed to the right of the residue we are transitioning *from*.

1. Transitioning from $(m, 0)$ to $(m + 1 \leq M, 0)$ has probability $1 - 2\gamma - \kappa$.

```
x: 1 | 1
y: 1 | 1
```

2. Transitioning from $(m, 0)$ to $(m' > m+1, 0)$ for $m' < M+1$ has probability $\gamma\epsilon^{m'-m-2}(1-\epsilon-\kappa)$.

```
x: 1 | 1 ... 1 1
y: 1 | 0 ... 0 1
```

3. Transitioning from $(m, 0)$ to $(m + 1, 1)$ has probability γ.

```
x: 1 | 0
y: 1 | 1
```

4. Terminating after $(m < M, 0)$ has probability $\gamma\epsilon^{M-m-1}\kappa$.

   ```
   x: 1 | 1 ... 1 $
   y: 1 | 0 ... 0 $
   ```

5. Terminating after $(M, 0)$ has probability κ.

   ```
   x: 1 | $
   y: 1 | $
   ```

6. Transitioning from $(m, 1)$ to $(m \leq M, 0)$ has probability $1 - \epsilon - \kappa$.

   ```
   x: 0 | 1
   y: 1 | 1
   ```

7. Transitioning from $(m, 1)$ to $(m, 1)$ has probability ϵ.

   ```
   x: 0 | 0
   y: 1 | 1
   ```

8. Terminating after $(M + 1, 1)$ has probability κ

   ```
   x: 0 | $
   y: 1 | $
   ```

9. Transitioning from the initial state to $(1, 0)$ has probability $1 - 2\gamma - \kappa$.

   ```
   x: | 1
   y: | 1
   ```

10. Transitioning from the initial state to $(m > 1, 0)$ for $m < M + 1$ has probability $\gamma\epsilon^{m-2}(1 - \epsilon - \kappa)$.

    ```
    x: | 1 ... 1 1
    y: | 0 ... 0 1
    ```

11. Transitioning from the initial state to $(1, 1)$ has probability γ.

    ```
    x: | 0
    y: | 1
    ```

12. Terminating immediately from the initial state has probability $\gamma\epsilon^{M-1}\kappa$ when $M > 0$.

    ```
    x: | 1 ... 1 $
    y: | 0 ... 0 $
    ```

13. Terminating immediately from the initial state has probability κ when $M = 0$.

    ```
    x: | $
    y: | $
    ```

These transition probabilities were derived without conditioning on the fact that we have observed X, which has length M. To compute this conditional probability, we calculate the probability that the pair HMM generates an alignment with too many or too few X residues starting from each MuE Markov model state.

1. Starting from a state $(m < M, 0)$, the probability of the pair HMM generating an invalid alignment that is too long (rather than transitioning to a valid MuE state) is $\gamma\epsilon^{M-m-1}(1-\epsilon-\kappa)+\gamma\epsilon^{M-m} = \gamma\epsilon^{M-m-1}(1-\kappa)$. The first term is from alignments that use a match state instead of terminating.

   ```
   x: 1 | 1 ... 1 1
   y: 1 | 0 ... 0 1
   ```

   The second term is from alignments that use an x-alone state instead of terminating.

   ```
   x: 1 | 1 ... 1 1
   y: 1 | 0 ... 0 0
   ```

2. Starting from a state $(m < M, 0)$, the probability of generating an invalid alignment that is too short (rather than transitioning to a valid MuE state) is $\kappa + \sum_{m'=m+1}^{M-1} \gamma\epsilon^{m'-m-1}\kappa = \kappa + \gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon}$. The first term is from alignments that immediately terminate.

   ```
   x: 1 | $
   y: 1 | $
   ```

   The second term is from alignments that terminate early after transitioning to the x-alone state.

   ```
   x: 1 | 1 ... 1 $
   y: 1 | 0 ... 0 $
   ```

3. Starting from the state $(M, 0)$, the probability of generating an invalid alignment is $(1-2\gamma-\kappa)+\gamma = 1-\gamma-\kappa$. The first term is from alignments that use a match state instead of terminating.

   ```
   x: 1 | 1
   y: 1 | 1
   ```

   The second term is from alignments that use an x-alone state instead of terminating.

   ```
   x: 1 | 1
   y: 1 | 0
   ```

4. Starting from a state $(m \leq M, 1)$ the probability of generating an invalid alignment that is too short is κ.

   ```
   x: 0 | $
   y: 1 | $
   ```

5. Starting from the state $(M+1,1)$, the probability of generating an invalid alignment that is too long is $1-\epsilon-\kappa$.

   ```
   x: 0 | 1
   y: 1 | 1
   ```

6. Starting from the initial state, the probability of generating an invalid alignment that is too long is $\gamma\epsilon^{M-1}(1-\epsilon-\kappa)+\gamma\epsilon^{M-m} = \gamma\epsilon^{M-1}(1-\kappa)$. The first term is from alignments that use a match state instead of terminating.

   ```
   x: | 1 ... 1 1
   y: | 0 ... 0 1
   ```

   The second term is from alignments that use an x-alone state instead of terminating.

   ```
   x: | 1 ... 1 1
   y: | 0 ... 0 0
   ```

7. Starting from the initial state, the probability of generating an invalid alignment that is too short is $\kappa + \sum_{m'=1}^{M-1} \gamma\epsilon^{m'-1}\kappa = \kappa + \gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon}$ when $M > 0$. The first term is from alignments that immediately terminate.

   ```
   x: | $
   y: | $
   ```

   The second term is from alignments that terminate early after transitioning to the x-alone state.

   ```
   x: | 1 ... 1 $
   y: | 0 ... 0 $
   ```

8. Starting from the initial state, if $M = 0$, then the probability of generating an invalid alignment is $(1-2\gamma-\kappa)+\gamma = 1-\gamma-\kappa$. The first term is from alignments that use a match state.

   ```
   x: | 1
   y: | 1
   ```

   The second term is from alignments that use an x-alone state.

   ```
   x: | 1
   y: | 0
   ```

We can confirm that all possible trajectories of the pair HMM are either valid transitions under the MuE Markov model or produce alignments with too few or too many X residues, by checking that the outward transition probabilities from each state sum to one.

1. From a state $(m < M, 0)$, the total outward transition probability is

$$\begin{aligned}
&(1-2\gamma-\kappa)+\gamma\sum_{m'=m+2}^{M}\epsilon^{m'-m-2}(1-\epsilon-\kappa)+\gamma+\gamma\epsilon^{M-m-1}\kappa+\gamma\epsilon^{M-m-1}(1-\kappa)\\
&+(\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})\\
&=1-\gamma+\gamma(1-\epsilon-\kappa)\frac{1-\epsilon^{M-m-1}}{1-\epsilon}+\gamma\epsilon^{M-m-1}+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon}\\
&=1-\gamma+\gamma(1-\epsilon^{M-m-1})+\gamma\epsilon^{M-m-1}\\
&=1
\end{aligned} \tag{S16}$$

2. From the state $(M, 0)$, the total outward transition probability is

$$\gamma+\kappa+(1-\gamma-\kappa)=1 \tag{S17}$$

3. From a state $(m \leq M, 1)$, the total outward transition probability is

$$(1-\epsilon-\kappa)+\epsilon+\kappa=1 \tag{S18}$$

4. From the state $(M + 1, 1)$, the total outward transition probability is

$$\kappa+\epsilon+(1-\epsilon-\kappa)=1 \tag{S19}$$

5. From the initial state, with $M > 0$, the total outward transition probability is

$$\begin{aligned}
&(1-2\gamma-\kappa)+\sum_{m=2}^{M}\gamma\epsilon^{m-2}(1-\epsilon-\kappa)+\gamma+\gamma\epsilon^{M-1}\kappa+\gamma\epsilon^{M-1}(1-\kappa)+(\kappa+\gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon})\\
&=1-\gamma+\gamma(1-\epsilon-\kappa)\frac{1-\epsilon^{M-1}}{1-\epsilon}+\gamma\kappa\epsilon^{M-1}+\gamma\epsilon^{M-1}(1-\kappa)+\gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon}\\
&=1-\gamma+\gamma(1-\epsilon^{M-1})-\gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon}+\gamma\epsilon^{M-1}+\gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon}\\
&=1
\end{aligned} \tag{S20}$$

6. From the initial state, with $M = 0$, the total outward transition probability is

$$\gamma+\kappa+(1-\gamma-\kappa)=1 \tag{S21}$$

Consolidating transition probabilities and conditioning on the length of X yields the transition matrix Equation S12.

Next we consider sequence emission probabilities, given an alignment. Recall that X and Y are one-hot encodings of sequences.

1. Consider the case that Y_l is aligned to X_m, ie.

```
x: 1
y: 1
```

The conditional probability of $Y_{l,b'} = 1$ given $X_{m,b} = 1$ is, according to the pair HMM, $\psi_{b,b'}/\pi_b$. This matches the conditional probability assigned by the MuE,

$$Y_l \sim \text{Categorical}\Big(\sum_{b''} X_{m,b''}\ell_{b''}\Big) = \text{Categorical}\Big(\frac{\psi_b}{\pi_b}\Big). \tag{S22}$$

2. Consider the case that Y_l is aligned to a gap, ie.

```
x: 0
y: 1
```

The conditional probability of $Y_{l,b}$ given X is just π_b (since X is not informative in this case). This matches the conditional probability assigned by the MuE,

$$Y_l \sim \text{Categorical}\big((\pi^\top \cdot \ell^{-1} \cdot \ell)^\top\big) = \text{Categorical}(\pi). \tag{S23}$$

3. Consider the case that X_m is aligned to a gap, ie.

```
x: 1
y: 0
```

The conditional probability of X_m given X is trivially one, so this term does not contribute to the conditional probability of Y given X under the pair HMM. It also does not contribute to the probability under the MuE.

Thus, term-by-term, the joint probability of W and Y under the proposed MuE distribution matches the joint probability of the corresponding alignment and Y under the pair HMM conditional on X.

□

S1.1.3 Profile HMM

The profile HMM (pHMM) is a widely used model for defining protein sequence families, inferring multiple sequence alignments, and performing database searches (Durbin et al., 1998).

Statement Define the pHMM insertion parameter $r_{m,j} \in [0, 1]$ for all $m \in \{1, ..., M + 1\}$ and $j \in \{0, 1, 2\}$, and the deletion parameter $u_{m,j} \in [0, 1]$ for all $m \in \{1, ..., M\}$ and $j \in \{0, 1, 2\}$. Then define the MuE transition matrix and termination probability

$$a^{(t)}_{k,k'} := \begin{cases} (1 - r_{m+1-g,g})(1 - u_{m+1-g,g}) \\ \qquad \text{if } m + 1 - g = m' \text{ and } g' = 0 \\ (1 - r_{m+1-g,g})u_{m+1-g,g}(\prod_{m''=m+2-g}^{m'-1}[(1 - r_{m'',2})u_{m'',2}])(1 - r_{m',2})(1 - u_{m',2}) \\ \qquad \text{if } m + 1 - g < m' \text{ and } g' = 0 \\ r_{m+1-g,g} \\ \qquad \text{if } m + 1 - g = m' \text{ and } g' = 1 \\ (1 - r_{m+1-g,g})u_{m+1-g,g}(\prod_{m''=m+2-g}^{m'-1}[(1 - r_{m'',2})u_{m'',2}])r_{m',2} \\ \qquad \text{if } m + 1 - g < m' \text{ and } g' = 1 \\ 0 \quad \text{otherwise} \end{cases} \tag{S24}$$

x TACGC

$r = (0,0,0,0,0,0)$ $u = (0,0,0,0,0,0)$	$r = (0,0,0,0,0,0)$ $u = (0,0.5,0,0,0,0)$	$r = (0,0,0,0.4,0,0)$ $u = (0,0,0,0,0,0)$
TACGC	TACGC	TACGTGC
TACGC	TACGC	TACGC
TACGC	TACGC	TACCGC
TACGC	TCGC	TACGC
TACGC	TACGC	TACAGC
TACGC	TCGC	TACGC
TACGC	TACGC	TACCGGC
TACGC	TACGC	TACGC
TACGC	TCGC	TACAAGC
TACGC	TCGC	TACGC

Figure S3: **Samples from the profile HMM.** The regressor sequence $X_{1,...,M}$ is set to TACGC, and we set $r_{m,j=0} = r_{m,j=1} = r_{m,j=2}$ and $u_{m,j=0} = u_{m,j=1} = u_{m,j=2}$ for all m.

$$t_k^{(t)} := \begin{cases} 1 - r_{M+1,g} \\ \qquad \text{if } m - g = M \\ (1 - r_{m+1-g,g})u_{m+1-g,g}(\prod_{m''=m+2-g}^{M}[(1 - r_{m'',2})u_{m'',2}])(1 - r_{M+1,2}) \\ \qquad \text{if } m - g < M \end{cases} \tag{S25}$$

The initial transition vector is given by $a_k^{(0)} := a_{0,k}^{(t)}$ and the initial termination probability is given by $t^{(0)} = t_0^{(t)}$. Let the MuE substitution matrix ℓ be the identity matrix I_B, ie.

$$\ell_{b,b'} := \delta_{b,b'} \tag{S26}$$

for $b, b' \in \{1, ..., B\}$.

With these definitions the profile HMM can be written as $Y \sim \text{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$. Figure S3 illustrates samples from the pHMM. Intuitively, r controls insertion probabilities and u controls deletion probabilities; when $r_{m,j} = 0$ and $u_{m,j} = 0$ for all m and j, we recover the no-mutation limit of the MuE.

Proof This result follows from the relabeling of the profile HMM Markov state architecture with the (m, g) notation (Figure S4). So-called "delete states" in profile HMMs do not generate observations Y_l. To compute the probability of transitioning between two observable states (m, g) and (m', g'), we compute the probability of (1) direct paths between the two states and (2) all possible paths between the two states that go only through deletion states. This yields Equation S24.

The emission probability of each state in the pHMM is set by its associated emission probability vector. Without loss of generality, we can write any emission matrix of the pHMM as $\tilde{x}$ (Definition 2.1) since ℓ is the identity matrix.

□

S1.1.4 Needleman-Wunsch

The Needleman-Wunsch (NW) algorithm is a classic non-probabilistic alignment method (Needleman and Wunsch, 1970).

Summary Let G be the NW gap penalty, which we assume to be negative, and define $u := e^G$.

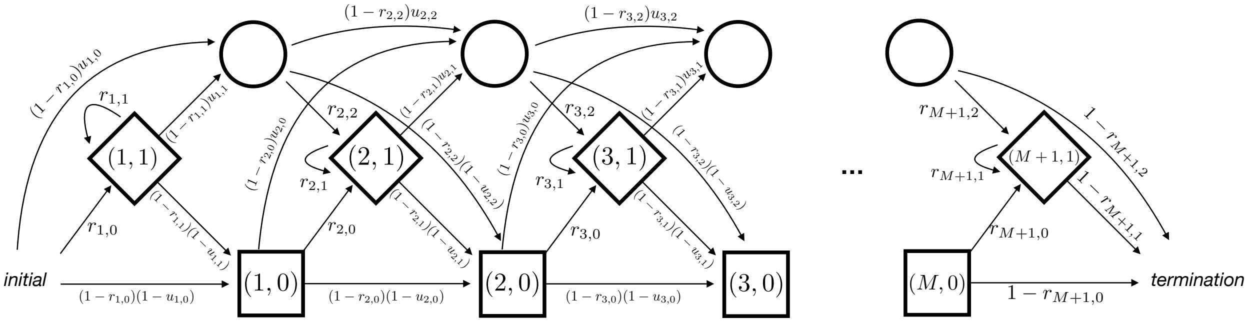


Figure S4: **Profile HMM state architecture.** The conventional profile HMM state architecture labeled with MuE states, using (m, g) notation. Squares indicate "match states", diamonds indicate "insert states", and circles indicate "delete states".

We define the MuE transition matrix and termination probabilities

$$a^{(t)}_{k,k'} := \begin{cases} \frac{1-u}{1+u} u^{m'-m-1+g} & \text{if } m-g < m' < M+1 \text{ and } g' = 0 \\ \frac{1-u}{1+u} u^{m'-m+g} & \text{if } m-g < m' \leq M+1 \text{ and } g' = 1 \\ 0 & \textit{otherwise} \end{cases} \tag{S27}$$

$$t^{(t)}_k := \frac{1+u^2}{1+u} u^{M-m+g} \tag{S28}$$

The initial transition vector is defined by $a^{(0)}_k := a^{(t)}_{0,k}$ and the initial termination probability is $t^{(0)}_k := t^{(t)}_0$. Let $S_{b,b'}$ be the NW similarity matrix, for which we assume that $\sum_{b'} e^{S_{b,b'}} = B$ for all b. We define, for $b, b' \in \{1, ..., B\}$,

$$\ell_{b,b'} := \frac{e^{S_{b,b'}}}{B}. \tag{S29}$$

Finally, for all $m \in \{1, ..., M+1\}$,

$$c_m := (\ell^{-1})^\top \cdot (1/B, ..., 1/B)^\top \tag{S30}$$

where ℓ^{-1} is the inverse of the substitution matrix (assumed to be invertible) and $(1/B, ..., 1/B)^\top$ is a length B column vector. Let X and Y be the sequences to be aligned.

Under the MuE model $Y \sim \text{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$, the maximum *a posteriori* estimator of the alignment variable w given X and Y corresponds to the Needleman-Wunsch pairwise alignment between X and Y. Note that in the limit $G \to -\infty$ and $S_{b,b'} \to -\infty$ for all $b' \neq b$, we recover the no-mutation limit of the MuE distribution.

Proof We can organize the NW scoring system according to transitions in the MuE Markov model. We use ω^x, ω^y notation to represent alignments, with the symbol "|" placed to the right of the residue we are transitioning *from*. We assign l' to be the residue of Y at the column of the alignment corresponding to state k'.

1. Transitioning from $(m, 0)$ to $(m' > m, 0)$ gives a NW score of $(m'-m-1)G + \sum_{b,b'} x_{m',b} S_{b,b'} y_{l',b'}$.

   ```
   x: 1 | 1 ... 1 1
   y: 1 | 0 ... 0 1
   ```

2. Transitioning from $(m, 0)$ to $(m' > m, 1)$ gives a NW score of $(m' - m)G$

```
x: 1 | 1 ... 1 0
y: 1 | 0 ... 0 1
```

3. Transitioning from $(m, 1)$ to $(m' \geq m, 0)$ gives a NW score of $(m' - m)G + \sum_{b,b'} x_{m',b} S_{b,b'} y_{l',b'}$

```
x: 0 | 1 ... 1 1
y: 1 | 0 ... 0 1
```

4. Transitioning from $(m, 1)$ to $(m' \geq m, 1)$ gives a NW score of $(m' - m + 1)G$.

```
x: 0 | 1 ... 1 0
y: 1 | 0 ... 0 1
```

5. Terminating after $(m, 0)$ gives a NW score of $(M - m)G$.

```
x: 1 | 1 ... 1 $
y: 1 | 0 ... 0 $
```

6. Terminating after $(m, 1)$ gives a NW score of $(M - m + 1)G$.

```
x: 0 | 1 ... 1 $
y: 1 | 0 ... 0 $
```

Now we can rewrite the Needleman-Wunsch objective function in terms of these transitions, rather than in terms of gap and insert scoring. In particular, define

$$\Delta(l', m, g, m', g') := \begin{cases} (m' - m - 1 + g)G + \sum_{b,b'} x_{m',b} S_{b,b'} y_{l',b'} & \text{if } m - g < m' < M \text{ and } g' = 0 \\ (m' - m + g)G & \text{if } m - g < m' \leq M \text{ and } g' = 1 \\ -\infty & \textit{otherwise} \end{cases} \tag{S31}$$

Based on the cases outlined above, the NW objective function can now be rewritten as

$$\arg\max_{\vec{m},\vec{g}} \sum_{l=1}^{L} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) + (M - m_L + g_L)G \tag{S32}$$

where we set $m_0 = 0, g_0 = 0$. If we find the solution to this objective function, then follow the mapping from the list of Markov chain states $(m_1, g_1), ..., (m_L, g_L)$ back to an alignment, we obtain the Needleman-Wunsch alignment between sequences x and y.

Now we examine the maximum *a posteriori* estimator of w under the MuE distribution. We have

$$\arg\max_{w} \log p(y, w|x, c, a, \ell) = \arg\max_{w} \left[\log p(\textit{term.}|w_L) + \sum_{l=2}^{L} \log p(y_l, w_l|w_{l-1}) + \log p(y_1, w_1) \right] \tag{S33}$$

where $p(\textit{term.}|w_L)$ is the termination probability after state w_L, which reduces to $p(\textit{term.}|\textit{init.})$ when $L = 0$. Under the given MuE model,

$$p(y_l, w_l|w_{l-1}) = \begin{cases} \frac{1-u}{1+u} u^{m_l - m_{l-1} - 1 + g_{l-1}} \frac{1}{B} \exp(\sum_{b,b'} x_{m_l,b} S_{b,b'} y_{l,b'}) & \text{if } m_{l-1} - g_{l-1} < m_l < M + 1 \text{ and } g_l = 0 \\ \frac{1-u}{1+u} u^{m_l - m_{l-1} + g_{l-1}} \frac{1}{B} & \text{if } m_{l-1} - g_{l-1} < m_l \leq M + 1 \text{ and } g_l = 1 \\ 0 & \textit{otherwise} \end{cases} \tag{S34}$$

$$p(term.|w_L) = \frac{1+u^2}{1+u} u^{M-m_L+g_L} \tag{S35}$$

$$p(y_1, w_1) = \begin{cases} \frac{1-u}{1+u} u^{m_1-1} \frac{1}{B} \exp(\sum_{b,b'} x_{m_1,b} S_{b,b'} y_{1,b'}) & \text{if } m_1 < M+1 \text{ and } g_1 = 0 \\ \frac{1-u}{1+u} u^{m_1} \frac{1}{B} & \text{if } m_1 \leq M+1 \text{ and } g_1 = 1 \\ 0 & otherwise \end{cases} \tag{S36}$$

$$p(term.|init.) = \frac{1+u^2}{1+u} u^M \tag{S37}$$

Now, the maximum *a posteriori* estimator of w can be written as

$$\begin{aligned} \arg\max_w \log p(y, w|x) = \arg\max_{\vec{m},\vec{g}} \Big[L \log(\frac{1-u}{1+u}\frac{1}{B}) + \log(\frac{1+u^2}{1+u}) + \sum_{l=1}^{L} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) \\ + (M - m_L + g_L)G \Big] \\ = \arg\max_{\vec{m},\vec{g}} \Big[\sum_{l=1}^{L} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) + (M - m_L + g_L)G \Big] \end{aligned} \tag{S38}$$

where again $m_0 = 0$ and $g_0 = 0$. This objective function is identical to the NW objective function (Equation S32), so the maximum *a posteriori* estimator of w in the MuE distribution corresponds to the Needleman-Wunsch pairwise alignment of X and Y.

We can confirm that the transition probabilities of the MuE distribution are normalized by considering transitions from state (m, g):

$$\begin{aligned} &\frac{1-u}{1+u} \sum_{m'=m-g+1}^{M} u^{m'-m-1+g} + \frac{1-u}{1+u} \sum_{m'=m-g+1}^{M+1} u^{m'-m+g} + \frac{1+u^2}{1+u} u^{M-m+g} \\ &= \frac{1-u}{1+u} \big[\sum_{m''=0}^{M-m-1+g} u^{m''} + u \sum_{m''=0}^{M-m+g} u^{m''} \big] + \frac{1+u^2}{1+u} u^{M-m+g} \\ &= \frac{1}{1+u} [1 - u^{M-m+g} + u - u^{M-m+g+2}] + \frac{1+u^2}{1+u} u^{M-m+g} \\ &= 1 - \frac{1+u^2}{1+u} u^{M-m+g} + \frac{1+u^2}{1+u} u^{M-m+g} \\ &= 1. \end{aligned} \tag{S39}$$

□

S1.2 Inferring multiple sequence alignments

In this section we describe how MuE observation models can be used to infer multiple sequence alignments. First we define a multiple sequence alignment, analogously to Definition 4.2.

Definition S1.2 (Biological multiple sequence alignment)**.** *Let $Y_1, \ldots, Y_N$ be sequences with lengths $L_1, \ldots, L_N$. A multiple sequence alignment $Y_{\text{MSA}} \in (\mathcal{B} \cup \{-\})^J$ has rows $Y_{\text{MSA},1}, \ldots, Y_{\text{MSA},N}$ each consisting of the letters of Y_i, in order, interspersed with gap symbols. The alignment Y_{MSA} must satisfy the condition that for every $j \in \{1, \ldots, J\}$, there exists some $i \in \{1, \ldots, N\}$ such that $Y_{\text{MSA},i,j} \in \mathcal{B}$.*

Algorithm 2: Multiple sequence alignment construction

```
input  : ${W_{1,1}, \dots, W_{1,L_1}}, \dots, {W_{N,1}, \dots, W_{N,L_N}}$ and $Y_1, \dots, Y_N$
output: $Y_{\mathrm{MSA}}$
Plug in definition of $j_l$ and $g_l$ for each sequence;
for $i \in \{1, 2, \dots, N\}$ do
    for $l_i \in \{1, 2, \dots, L_i\}$ do
        $g_{i,l_i} = \mathbb{I}(W_{i,l_i} > M)$;
        $m_{i,l_i} = W_{i,l_i} - M g_{i,l_i}$;
    end
    $g_{i,L_i+1} = 0$                    (for convenience);
    $m_{i,L_i+1} = 0$                    (for convenience);
end
$n = 0$;
$l_1, l_2, \dots, l_N = 1$;
Iterate through each latent state, assigning letters of $Y_1, \dots, Y_N$ to $Y_{\mathrm{MSA}}$;
for $\tilde{m} \in \{1, 2, \dots, M+1\}$ do
    Place in the same contiguous set of columns letters generated from the same site in $c$;
    while $\exists i : m_{i,l_i} = \tilde{m}$ and $g_{i,l_i} = 1$ do
        $n = n + 1$;
        for $i \in \{1, 2, \dots, N\}$ do
            if $m_{i,l_i} = \tilde{m}$ and $g_{i,l_i} = 1$ then
                $Y_{\mathrm{MSA},i,n} = Y_{i,l_i}$;
                $l_i = l_i + 1$;
            else
                $Y_{\mathrm{MSA},i,n} =$ -;
            end
        end
    end
    Place in the same column letters generated from the same site in $X$;
    if $\exists i : m_{i,l_i} = \tilde{m}$ and $g_{i,l_i} = 0$ then
        $n = n + 1$;
        for $i \in \{1, \dots, N\}$ do
            if $m_{i,l_i} = \tilde{m}$ and $g_{i,l_i} = 0$ then
                $Y_{\mathrm{MSA},i,n} = Y_{i,l_i}$;
                $l_i = l_i + 1$;
            else
                $Y_{\mathrm{MSA},i,n} =$ -;
            end
        end
    end
end
```

Consider models of the form Equation 2, and let W_i be the latent alignment variable associated with sequence Y_i, i.e. $W_{i,1}, \dots, W_{i,L_i}$ is the path through the latent state space that generated Y_i with length L_i. Algorithm 2 constructs a multiple sequence alignment of the dataset $Y_1, \dots, Y_N$ given $W_1, \dots, W_N$. In the case of multiple sequence alignments, as opposed to pairwise alignments, there is no longer a unique alignment given W, since X is not observed; the Algorithm 2 construction is chosen to match a standard construction used for the profile HMM, see Durbin et al. (1998), Chapter 6.5. Note the profile HMM is a special case of Equation 2 with $p_\theta(v) = \delta_{v_0}(v)$ where $\delta_{v_0}(v)$ is the Dirac delta function at v_0, so it is reasonable to apply the same construction to MuE observation models in general.

S1.3 Proof of Proposition 4.5

We require that with probability 1, the set $\{j_1, \ldots, j_L\}$ defined by Definition 4.3 is valid, i.e. it must be ordered such that $j_l < j_{l+1}$ for all $l \in \{1, \ldots, L-1\}$. Plugging in Definition 4.3, this is equivalent to the requirement that

$$m_{l+1} > m_l - g_l, \tag{S40}$$

where recall $m_l := W_l - M g_l$. For this inequality to hold with probability 1 for any sample W, Condition 2.2 is necessary and sufficient. □

S1.4 Vogel et al. Natural Language Translation

The Vogel et al. (1996) translation model takes the same general form as a MuE distribution, with X a sentence in one language and Y a sentence in another language (encoded as sequences of words). In particular, with states k indexed by tuples (m, g), the transition matrix takes the form

$$a^{(t)}_{k,k'} := \begin{cases} \frac{r_{M+m'-m}}{\sum_{m''=1}^{M} r_{M+m''-m}} & \text{if } g = g' = 0 \text{ and } m, m' \leq M \\ 0 & otherwise \end{cases} \tag{S41}$$

where $r \in \mathbb{R}^{2M}_{+}$ is a vector of non-negative weights. The initial transition vector is defined by $a^{(0)}_{k} := a^{(t)}_{0,k}$. The length L of Y is sampled independently of W. We can see that for general r, Condition 2.2 is violated.

S2 Models

In this section we provide a detailed description of the models evaluated in the main text. We parameterize the MuE in each model following the example of the profile HMM (Section S1.1). We introduce parameters $r \in [0,1]^M$ and $u \in [0,1]^M$ and set $u_M := 0$. Intuitively, r_m is the probability of an insertion at position m of x and u_m is the probability of a deletion at position m of x. We have the transition matrix

$$a^{(t)}_{k,k'} = \begin{cases} (1 - r_{m+1-g})(1 - u_{m+1-g}) \\ \qquad \text{if } m + 1 - g = m' \leq M \text{ and } g' = 0 \\ (1 - r_{m+1-g}) u_{m+1-g} (\prod_{m''=m+2-g}^{m'-1} [(1 - r_{m''}) u_{m''}])(1 - r_{m'})(1 - u_{m'}) \\ \qquad \text{if } m + 1 - g < m' \leq M \text{ and } g' = 0 \\ r_{m+1-g} \\ \qquad \text{if } m + 1 - g = m' \leq M \text{ and } g' = 1 \\ (1 - r_{m+1-g}) u_{m+1-g} (\prod_{m''=m+2-g}^{m'-1} [(1 - r_{m''}) u_{m''}]) r_{m'} \\ \qquad \text{if } m + 1 - g < m' \leq M \text{ and } g' = 1 \\ 0 \quad \text{otherwise} \end{cases} \tag{S42}$$

where, as in Section S1.1, we index states k by (m, g) where $g = \mathbb{I}(k > M)$ and $m = k - Mg$. The initial transition vector follows the same form as the transition matrix, and can be written as $a^{(0)}_k = a^{(t)}_{0,k}$. Rather than assign a termination state we assume the length of the sequence Y_i, ie. L_i, is independent of W. For computational convenience, in our experiments we assigned $p(Y_l|W_l = M) = 0$ for all possible Y_l (in general, M will be chosen to be a large value, such that there is a low probability of reaching this state anyway). Since the probability of L_i does not contribute to the per residue perplexity performance metric (Section S4) we do not use an explicit model for L_i.

Note that in our experiments we go slightly beyond the vanilla MuE observation model presented in the main text (Equation 2), and allow the insertion sequence c to also depend on a continuous-space model $p_\theta(v)$.

S2.1 Profile HMM

The profile HMM is

$$Y_i \sim \text{MuE}(x, c, \ell = I_B, a^{(0)}(r, u), a^{(t)}(r, u)) \tag{S43}$$

where $a^{(0)}(r, u)$ and $a^{(t)}(r, u)$ depend deterministically on the parameters r and u according to Equation S42, $D = B$, and I_B is the $B \times B$ identity matrix.

S2.2 RegressMuE

The RegressMuE model uses a linear regression model as the MuE observation's continuous-space vector model. Let $H_{i,1}, ..., H_{i,T}$ be covariates associated with sequence Y_i. Let $\beta^{(x)}_0, ..., \beta^{(x)}_T \in \mathbb{R}^{M\times D}$ be a set of coefficients associated with X, and let $\beta^{(c)}_0, ..., \beta^{(c)}_T \in \mathbb{R}^{(M+1)\times D}$ be a set of coefficients associated with c. Then the RegressMuE is

$$\begin{aligned} V^{(x)}_i &= \beta^{(x)}_0 + \sum_{t=1}^{T} H_{i,t}\beta^{(x)}_t \\ V^{(c)}_i &= \beta^{(c)}_0 + \sum_{t=1}^{T} H_{i,t}\beta^{(c)}_t \\ Y_i &\sim \text{MuE}(x = \text{softmax}(V^{(x)}_i), c = \text{softmax}(V^{(c)}_i), \ell, a^{(0)}(r, u), a^{(t)}(r, u)). \end{aligned} \tag{S44}$$

Note that in this model, unlike the pHMM, the substitution matrix ℓ is not constrained to the identity. When $r_m = q_m = 0$ for all m and $\ell = I_B$, the RegressMuE reduces to a multi-output multinomial logit regression model.

S2.3 FactorMuE

The FactorMuE model is the latent linear version of the RegressMuE. Instead of observing covariates H, we draw a latent variable Z from a standard normal prior,

$$\begin{aligned}
Z_{i,t} &\sim \text{Normal}(0,1)\\
V_i^{(x)} &= \beta_0^{(x)} + \sum_{t=1}^{T} Z_{i,t}\beta_t^{(x)}\\
V_i^{(c)} &= \beta_0^{(c)} + \sum_{t=1}^{T} Z_{i,t}\beta_t^{(c)}\\
Y_i &\sim \text{MuE}(x = \text{softmax}(V_i^{(x)}), c = \text{softmax}(V_i^{(c)}), \ell, a^{(0)}(r,u), a^{(t)}(r,u))
\end{aligned} \tag{S45}$$

S2.4 NeuralMuE

The NeuralMuE model uses a fully connected neural network as the MuE observation's continuous-space vector model. We use a network Γ layers using relu nonlinearities, widths $T_{1:(\Gamma+1)}$, and weights $\beta_{1:(\Gamma+1)}$. Let $H_{i,1:T_{(\Gamma+1)}}$ be a vector of covariates.

$$\begin{aligned}
V_{i,\Gamma+1} &= \beta_{\Gamma+1,0} + \sum_{t=1}^{T_{\Gamma+1}} H_{i,t}\beta_{\Gamma+1,t}\\
V_{i,\Gamma} &= \beta_{\Gamma,0} + \sum_{t=1}^{T_{\Gamma}} \text{relu}(V_{i,\Gamma+1,t})\beta_{\Gamma,t}\\
&\ldots\\
V_{i,1}^{(x)} &= \beta_{1,0}^{(x)} + \sum_{t=1}^{T_1} \text{relu}(V_{i,2,t})\beta_{1,t}^{(x)}\\
V_{i,1}^{(c)} &= \beta_{1,0}^{(c)} + \sum_{t=1}^{T_1} \text{relu}(V_{i,2,t})\beta_{1,t}^{(c)}\\
Y_i &\sim \text{MuE}(x = \text{softmax}(V_{i,1}^{(x)}), c = \text{softmax}(V_{i,1}^{(c)}), \ell, a^{(0)}(r,u), a^{(t)}(r,u))
\end{aligned} \tag{S46}$$

S2.5 LatentNeuralMuE

The LatentNeuralMuE model uses a neural network latent variable model as the MuE observation's continuous-space vector model. It is the latent covariate version of the NeuralMuE, where instead of observing H we draw a latent variable Z from a standard normal prior.

$$\begin{aligned}
Z_{i,t} &\sim \text{Normal}(0,1)\\
V_{i,\Gamma+1} &= \beta_{\Gamma+1,0} + \sum_{t=1}^{T_{\Gamma+1}} Z_{i,t}\beta_{\Gamma+1,t}\\
V_{i,\Gamma} &= \beta_{\Gamma,0} + \sum_{t=1}^{T_{\Gamma}} \text{relu}(V_{i,\Gamma+1,t})\beta_{\Gamma,t}\\
&\dots\\
V_{i,1}^{(x)} &= \beta_{1,0}^{(x)} + \sum_{t=1}^{T_1} \text{relu}(V_{i,2,t})\beta_{1,t}^{(x)}\\
V_{i,1}^{(c)} &= \beta_{1,0}^{(c)} + \sum_{t=1}^{T_1} \text{relu}(V_{i,2,t})\beta_{1,t}^{(c)}\\
Y_i &\sim \text{MuE}(x = \text{softmax}(V_{i,1}^{(x)}), c = \text{softmax}(V_{i,1}^{(c)}), \ell, a^{(0)}(r,u), a^{(t)}(r,u))
\end{aligned} \tag{S47}$$

S2.6 Priors

We place standard normal priors Normal$(0,1)$ over each element of each coefficient matrix β in each model. Recall that each row of the matrix ℓ is constrained to the simplex, $\ell \in \Delta_B$. To enable easy gradient-based optimization and stochastic variational inference (Kucukelbir et al., 2017), we transform an unconstrained parameter $\tilde{\ell} \in \mathbb{R}^{D\times B}$ with a Gaussian prior to the simplex,

$$\begin{aligned}
\tilde{\ell}_{d,b} &\sim \text{Normal}(0,1)\\
\ell_d &:= \text{softmax}(\tilde{\ell}_d).
\end{aligned} \tag{S48}$$

The variables r_m and u_m are constrained to $[0,1]$. This corresponds to the first dimension of a simplex Δ_2, and so we apply the same approach,

$$\begin{aligned}
\tilde{r}_{m,j} &\sim \text{Normal}(\mu_j^{(r)}, 1) \text{ for } j \in \{1,2\}\\
r_m &:= \frac{\exp(\tilde{r}_{m,2})}{\exp(\tilde{r}_{m,1}) + \exp(\tilde{r}_{m,2})}
\end{aligned} \tag{S49}$$

where $\mu^{(r)}$ is a hyperparameter. The variable u_m is handled identically, with prior $\tilde{u}_{m,j} \sim \text{Normal}(\mu_j^{(u)}, 1)$ for $j \in \{1,2\}$.

S3 Inference

S3.1 Stochastic variational inference

Variational inference approximates the posterior distribution $p(\theta|Y_{1:N})$ of a given probabilistic model using a tractable family of distributions $q_\eta(\theta|Y_{1:N})$ parameterized by η (Blei et al., 2017). To form this approximation, variational inference minimizes the Kullback-Leibler (KL) divergence between the two distributions,

$$\eta_0 := \arg\min_\eta \text{KL}(q_\eta(\theta|Y_{1:N})||p(\theta|Y_{1:N})) \tag{S50}$$

This objective can be rewritten as maximizing the evidence lower bound (ELBO),

$$\eta_0 = \arg\max_{\eta} \mathbb{E}_{q_\eta(\theta|Y_{1:N})}[\log p(Y_{1:N}, \theta)] - \mathbb{E}_{q_\eta(\theta|Y_{1:N})}[\log q_\eta(\theta|Y_{1:N})] = \arg\max_{\eta} \text{ELBO}(\eta) \tag{S51}$$

We employ mean-field variational inference for MuE observation models. We use a diagonal Gaussian distribution, with unknown mean and standard deviation, for the variational distribution over the global parameters $\tilde{r}, \tilde{u}, \tilde{\ell}$ and $\tilde{\beta}$. For the local variable z in the FactorMuE and LatentNeuralMuE, we amortize inference using an inference network (also known as an encoder network) (Kingma and Welling, 2013; Rezende et al., 2014). In particular, we set

$$q_{\eta_z}(z_{1:N}|Y_{1:N}) = \prod_{i=1}^{N} q_{\eta_z}(z_i|Y_i) = \prod_{i=1}^{N} \mathcal{N}(z_i|f^{(\mu)}(Y_i;\eta_z), f^{(\sigma)}(Y_i;\eta_z)) \tag{S52}$$

where $\mathcal{N}(z|\mu,\sigma)$ is the probability distribution function of a Gaussian with mean μ and standard deviation σ, and $f^{(\mu)}(Y_i;\eta_z)$ and $f^{(\sigma)}(Y_i;\eta_z)$ are differentiable functions of η_z. We parameterize $f^{(\mu)}$ and $f^{(\sigma)}$ using a neural network,

$$\begin{aligned}
y^{(q)}_{i,l} &= \mathbb{E}_{Y' \sim \text{MuE}(Y_i, c^{(q)}, \ell^{(q)}, a^{(0)}(r^{(q)}, u^{(q)}), a^{(t)}(r^{(q)}, u^{(q)}))}[Y'_l] \\
v^{(q)}_{i,\Gamma^{(q)}+1} &= \beta^{(q)}_{\Gamma^{(q)}+1,0} + \sum_{l=1}^{L^{(q)}} \sum_{b=1}^{B} y^{(q)}_{i,l,b} \beta^{(q)}_{\Gamma^{(q)}+1,l,b} \\
v^{(q)}_{i,\Gamma^{(q)}} &= \beta^{(q)}_{\Gamma^{(q)},0} + \sum_{t=1}^{T_{\Gamma^{(q)}}} \text{relu}(v^{(q)}_{i,\Gamma^{(q)}+1,t}) \beta^{(q)}_{\Gamma^{(q)},t} \\
&\dots \\
f^{(\mu)} &= \beta^{(q,\mu)}_{1,0} + \sum_{t=1}^{T_1} \text{relu}(v^{(q)}_{i,2,t}) \beta^{(q,\mu)}_{1,t} \\
f^{(\sigma)} &= \left|\beta^{(q,\sigma)}_{1,0} + \sum_{t=1}^{T_1} \text{relu}(v^{(q)}_{i,2,t}) \beta^{(q,\sigma)}_{1,t}\right|.
\end{aligned} \tag{S53}$$

where we have introduced the variational parameters $(\beta^{(q)}, c^{(q)}, r^{(q)}, u^{(q)}, \ell^{(q)}) =: \eta_z$. The first layer of the encoder employs the MuE distribution and computes the expected value of mutants of Y_i, at positions $l \in \{1, ..., L^{(q)}\}$; this expected value is a differentiable function of the MuE parameters, and can be tractably computed using the forward algorithm. We use the same parameterization of the MuE distribution as in the models (Section S2), but fix $r^{(q)}_1 = r^{(q)}_2 = ... = r^{(q)}_M$ and $u^{(q)}_1 = u^{(q)}_2 = ... = u^{(q)}_{M-1}$ and $c^{(q)}_1 = c^{(q)}_2 = ... = c^{(q)}_M$. Intuitively, the MuE encoding serves to "smear out" the one-hot encoded sequence Y_i according to learnable indel and substitution probabilities, making it easier for the encoder to learn which sequences are similar, and making each encoded sequence $y^{(q)}_i$ the same length $L^{(q)}$.

To optimize the variational approximation we need to compute the gradient of the ELBO with respect to the variational parameters η. To enable faster optimization we employ stochastic variational inference, approximating the gradient at each update step using a minibatch of data (Ranganath et al., 2014). Let $\phi := (\beta, r, u, \ell)$ be the global parameters of the MuE observation models proposed in Section S2 and let η_ϕ be the parameters of the associated mean-field variational distribution. Then the gradient of the ELBO is

$$\begin{aligned}\nabla_\eta \text{ELBO}(\eta) &= \sum_{i=1}^{N}\Big(\nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)q_{\eta_z}(z_i|Y_i)}[\log p(Y_i|Z_i,\phi)] + \nabla_\eta \mathbb{E}_{q_{\eta_z}(z_i|Y_i)}\Big[\log\frac{p(Z_i)}{q_{\eta_z}(Z_i|Y_i)}\Big]\Big)\\ &\quad +\nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)}\Big[\log\frac{p(\phi)}{q_{\eta_\phi}(\phi)}\Big]\\ &\approx \frac{N}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\Big(\nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)q_{\eta_z}(z_i|Y_i)}[\log p(Y_i|Z_i,\phi)] + \nabla_\eta \mathbb{E}_{q_{\eta_z}(z_i|Y_i)}\Big[\log\frac{p(Z_i)}{q_{\eta_z}(Z_i|Y_i)}\Big]\Big)\\ &\quad +\nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)}\Big[\log\frac{p(\phi)}{q_{\eta_\phi}(\phi)}\Big]\end{aligned} \tag{S54}$$

where $\mathcal{S} \subseteq \{1, ..., N\}$ is the set of datapoint indices making up the minibatch and $|\mathcal{S}|$ is the size of the set $\mathcal{S}$. We estimate the gradient of the first term on the right hand side of this equation using the reparameterization trick Monte Carlo estimator (with a single sample) and automatic differentiation (Kucukelbir et al., 2017; Kingma and Welling, 2013; Rezende et al., 2014). The remaining terms can be computed analytically (see e.g. Kingma and Welling (2013); Rezende et al. (2014)). Note that this approach relies crucially on the fact that the marginal likelihood of the MuE model, $p_{\text{MuE}}(y|x, c, \ell, a^{(0)}, a^{(t)}) = \sum_w p_{\text{MuE}}(y|w, x, c, \ell, a^{(0)}, a^{(t)})$, is a differentiable function of x, c, a and ℓ. We integrate over all possible values of the Markov chain state variable w using the forward algorithm. In our experiments we did not use the parallelized algorithm from Särkkä and García-Fernández (2020), but see the `DiscreteHMM` distribution in Pyro (Bingham et al., 2019) for an example implementation of the parallelized algorithm within an automatic differentiation system.

It is useful in some circumstances to reweight the variational objective to reduce the amount of regularization placed on the local latent variable. In particular, for $\chi \in [0, 1]$, we reweight the ELBO as

$$\begin{aligned}\text{ELBO}_\chi(\eta) &= \sum_{i=1}^{N}\Big(\mathbb{E}_{q_{\eta_\phi}(\phi)q_{\eta_z}(z_i|Y_i)}[\log p(Y_i|Z_i,\phi)] + \chi\mathbb{E}_{q_{\eta_z}(z_i|Y_i)}\Big[\log\frac{p(Z_i)}{q_{\eta_z}(Z_i|Y_i)}\Big]\Big)\\ &\quad +\mathbb{E}_{q_{\eta_\phi}(\phi)}\Big[\log\frac{p(\phi)}{q_{\eta_\phi}(\phi)}\Big].\end{aligned} \tag{S55}$$

We achieved improved training performance by annealing the weight χ from 0 to 1 linearly over the course of an initial time period during training (Bowman et al., 2016). To avoid posterior collapse and produce informative latent representations, we found it useful in certain cases to anneal χ only up to a low value $\chi_0 << 1$ in which case we are approximating the maximum likelihood estimator of z; this annealing schedule was only used for producing data visualizations, rather than prediction of held out data (Section S7) (Alemi et al., 2018).

S3.2 Probabilistic Programming

We implemented the MuE distribution in Edward2, a simple GPU-enabled probabilistic programming language which can use stochastic variational inference as well as MCMC (Tran et al., 2018). (Note that this is not the implementation used for the reported experiments, but instead a more user friendly version of the same code.) It is available at `https://github.com/debbiemarkslab/MuE`. The implementation makes it easy to try out different priors and different continuous-space matrix models, and to build joint models of sequences and experimental measurements, or other data sources. As a brief example, here is real code (not pseudocode) for a version of the FactorMuE with Dirichlet priors on u, r and ℓ. It uses a Laplace prior instead of a Normal prior on the local latent variable, such that the continuous space model is an independent component analysis model (Murphy (2012), Chapter 12.6).

```
# Latent representation.
z = Laplace(0., z_scale, sample_shape=latent_dims, name="z")
# Factors.
bt = Normal(0., bt_scale, sample_shape=[2, latent_dims, latent_length+1,
                                        latent_alphabet_size], name="bt")
# Offset.
b0 = Normal(0., b0_scale, sample_shape=[2, latent_length+1,
                                        latent_alphabet_size], name="b0")
# Ancestral sequence.
vxln = tf.einsum(’j,jkl->kl’, z, bt[0, :, :, :]) + b0[0, :, :]
# Insert biases.
vcln = tf.einsum(’j,jkl->kl’, z, bt[1, :, :, :]) + b0[1, :, :]
# Deletion probability.
u = Dirichlet(u_conc, name="u")
# Insertion probability.
r = Dirichlet(r_conc, name="r")
# Substitution probability.
l = Dirichlet(l_conc, name="l")

# Build the structured HMM parameters given the MuE parameters
# using the mue package.
a0, a, e = mue.make_hmm_params(
            vxln - tf.reduce_logsumexp(vxln, axis=1, keepdims=True),
            vcln - tf.reduce_logsumexp(vcln, axis=1, keepdims=True),
            tf.math.log(u), tf.math.log(r), tf.math.log(l),
            transfer_mats, eps=eps, dtype=dtype)

# Sample data.
x = HiddenMarkovModel(
            tfpd.Categorical(logits=a0), tfpd.Categorical(logits=a),
            tfpd.OneHotCategorical(logits=e), seq_len, name="x")
```

Also included in our package is a utility for building inference networks that use the MuE distribution.

S4 Evaluation

The per residue perplexity of a probabilistic sequence model $p(y)$, over a dataset $Y_{1:N}$, is defined as

$$\Omega := \exp\big(-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\log p(Y_i|L_i)\big). \tag{S56}$$

In evaluating our models, we computed the average log likelihood performance on a heldout test set $Y_\mathcal{T}$ for the model distribution learned from the training set $Y_\mathcal{D}$. More precisely, we use

$$\hat{\Omega} := \exp\big(- \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \frac{1}{L_i} \mathbb{E}_{q(\phi|Y_\mathcal{D})}[\log p(Y_i|L_i, \phi)]\big) \tag{S57}$$

where $q(\phi|y_\mathcal{D})$ is the variational approximation to the posterior distribution from the training dataset and $|\mathcal{T}|$ is the size of the test set. For models with local latent variables z_i, we approximate the marginal likelihood using the ELBO (Blei et al., 2017),

$$\hat{\Omega} \approx \exp\Bigg(- \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \frac{1}{L_i} \big(\mathbb{E}_{q(\phi|Y_\mathcal{D})q(z_i|Y_i)}[\log p(Y_i|L_i, Z_i, \phi)] + \mathbb{E}_{q(z_i|Y_i)}\Big[\log \frac{p(Z_i)}{q(Z_i|Y_i)}\Big]\big)\Bigg). \tag{S58}$$

We use Monte Carlo estimation for the expectations. In comparing between different models p_1 and p_2, we also report the log Bayes factor associated with the held out data, ie. the difference in total log probability of the heldout data between the two models,

$$\log \mathrm{BF}_{1,2} := \sum_{i \in \mathcal{T}} \mathbb{E}_{q_2(\phi|Y_\mathcal{D})}[\log p_2(Y_i|L_i, \phi)] - \sum_{i \in \mathcal{T}} \mathbb{E}_{q_1(\phi|Y_\mathcal{D})}[\log p_1(Y_i|L_i, \phi)] \tag{S59}$$

where q_1 and q_2 are the variational approximations associated with p_1 and p_2. For models with local latent variables, we can use the ELBO approximation as in Equation S58. The Bayes factor provides a measurement of the total evidence in favor of one model versus another.

Per residue perplexity is a useful performance metric for biological sequence models because it is an absolute scale and comparable across datasets as well as models. Since per residue perplexity is not yet widely used in the biological literature, in the interest of making it more interpretable we computed the expected per-residue perplexity for a variety of different protein sequence models, covering different data regimes. In particular, for each model $p(y)$, we examined the expected perplexity in the large data limit, assuming that the model is true,

$$\Omega_0 := \exp\big(- \mathbb{E}_{p(y)}\big[\frac{1}{L} \log p(Y|L)\big]\big). \tag{S60}$$

The expected perplexity is the exponentiated entropy of the model distribution, and so also provides a measurement of sequence diversity under the model. Below, we compute the expected perplexity for distributions ranging from the very high diversity regime (all of evolution) down to the very small diversity regime (human population genetics).

Naive

A naive model assigns an equal probability to each amino acid. In this case the per residue perplexity is

$$\Omega_0 = \exp(-\mathbb{E}[\log(1/20)]) = 20. \tag{S61}$$

Amino acid frequencies

A simple modeling approach is to predict individual amino acids solely based on their naturally occurring frequency across evolution. Using the UniprotKB amino acid frequencies f_b for $b \in \{1, ..., B = 20\}$, we have

$$\Omega_0 = \exp\big(- \mathbb{E}_{Y \sim \text{Categorical}(f)}[\log(f^\top \cdot Y)]\big) = \exp(-\sum_{b=1}^{20} f_b \log f_b) \approx 17.92 \tag{S62}$$

where Y is a one-hot encoding (UniProt Consortium, 2019; Gasteiger et al., 2005).

BLOSUM62

If we are studying specific evolutionary families of proteins, an idealized strategy for building a model is to infer the sequence of the last common ancestor and then predict family members using the standard BLOSUM62 substitution matrix (Henikoff and Henikoff, 1992). The BLOSUM62 matrix is a renormalized copula density, but we can convert it into a mutation probability matrix ℓ by assuming the marginal probability of each amino acid follows the UniprotKB frequency across evolution:

$$\begin{aligned}\log \ell_{b,b'} &= \log p(y_{b'} = 1 | x_b = 1) = \log\left(\frac{f_{b,b'}}{f_b}\right) = \log f_{b'} + \log\left(\frac{f_{b,b'}}{f_b f_{b'}}\right) \\ &= \log f_{b'} + \frac{\log(2)}{2}\text{BLOSUM62}_{b,b'}\end{aligned} \tag{S63}$$

where x is a one-hot encoding of the ancestral amino acid, y is a one-hot encoding the mutated amino acid, and $f_{b,b'}$ is the joint probability of amino acids b and b', where $b, b' \in \{1, ..., B = 20\}$. (The $\log(2)/2$ factor comes from the definition of BLOSUM62.) We renormalize the rows ℓ_b to ensure $\ell_b \in \Delta_B$ (BLOSUM62 uses only small integers, producing non-negligible rounding error). Next, we assume that the ancestral sequence is known exactly, has infinite length, and the frequency of each amino acid within the ancestral sequence matches the UniprotKB overall frequency across evolution. The expected per residue perplexity is then

$$\Omega_0 = \exp(-\mathbb{E}_{X\sim\text{Categorical}(f)}\left[\mathbb{E}_{Y\sim\text{Categorical}(X\cdot\ell)}[\log(X^\top \cdot \ell \cdot Y)]\right]) \approx 11.00. \tag{S64}$$

Human Population Genetics

Finally, we examined a simple model of human population variation. Each human has on average roughly 5 million single nucleotide polymorphisms (SNPs) relative to the reference genome (1000 Genomes Project Consortium et al., 2015). Naively assuming a constant mutation rate over the genome, the probability of a mutation occurring in any particular codon is $q_{\text{codon}} = 1-(1-5/6400)^3$, since there are 6.4 billion total base pairs. If we very naively assume a uniform probability of the codon mutating to any other amino acid, then we can use the substitution matrix ℓ defined by

$$\ell_{b,b'} = \begin{cases} \frac{q_{\text{codon}}}{19} & \text{if } b \neq b' \\ 1 - q_{\text{codon}} & \text{if } b = b'. \end{cases} \tag{S65}$$

If we further very naively assume that there are no correlations among mutations at different genome locations when looking across individuals, then the expected per residue perplexity of the sequence distribution is

$$\Omega_0 = \exp\left(\mathbb{E}_{Y\sim\text{Categorical}(x^\top\cdot\ell)}[\log(x^\top \cdot \ell \cdot Y)]\right) \approx 1.024. \tag{S66}$$

S5 Predictive Performance

Evolutionarily related sequences were collected using jackhmmer (v3.1) from the UniRef100 dataset (date 6/2019) (noa; Suzek et al., 2015). We used seed sequences with Uniprot identifiers DYR_HUMAN (DHFR dataset), PINE_ECOLI (PINE dataset), CDN1B_HUMAN (CDKN1B dataset), and VE6_HPV16 (VE6 dataset). We set a bitscore threshold of 0.5 bits/residue as in Hopf et al. (2017) and ran the jackhmmer search using the API from the EVcouplings package (Hopf et al., 2019). We included the full envelope of the profile HMM hit in the final dataset. The CDN1B dataset had 1,055 sequences and the VE6 dataset 1,609 sequences. We found 32,510 and 79,354 hits respectively for the DHFR and PINE datasets, which we randomly subsampled to 10,000 sequences to create the final datasets. Note that the jackhmmer search algorithm uses a profile HMM to find distant homologs, and thus may bias the dataset to look more like samples from a pHMM; we therefore expect the performance gains from using other MuE observation models, as compared to the pHMM, on these datasets to be smaller (more conservative) than the performance gains that might be achieved on alternative datasets assembled using different search methods. The TCR dataset was not assembled using jackhmmer, see Section S6 for details.

We set the latent alphabet size $D = 25$. In each experiment, we set M to be 10% longer than the longest sequence in the dataset. We used $T = 5$ latent space dimensions in the FactorMuE and layer sizes $T_2 = 5$, $T_1 = 10$ in the LatentNeuralMuE (we found a substantial dropoff in performance when increasing network width or depth). In the recognition network, we set $L^{(q)} = M - 1$. We also used $\Gamma^{(q)} = 0$ (no relu nonlinearities) in the FactorMuE recognition network and $\Gamma^{(q)} = 1$, $T_1 = 10$ in the LatentNeuralMuE recognition network. For the prior on the MuE insertion and deletion parameters we used $\mu^{(r)} = \mu^{(u)} = (100, 1)$ to disfavor indels.

We optimized the variational approximation using Adam (Kingma and Ba, 2015) and a minibatch size of 5. The mean of the variational distribution was initialized at the prior mean, while the variance was initialized to a small random value (the absolute value of a sample from a normal distribution with standard deviation 0.01). We used one Monte Carlo sample to estimate the ELBO gradient at each step. For each model and dataset, we evaluated two different learning rates, 0.1 and 0.01, and three different random restarts, selecting among training runs the parameter values that reached the highest ELBO on the training set for making predictions. For models with local latent variables (the FactorMuE and LatentNeuralMuE), we annealed the ELBO reweighting factor χ from 0 to 1 linearly over the first 2 epochs. We trained for 4 epochs total on the DHFR and PINE datasets, and 7 epochs total on the smaller CDKN1B, VE6 and TCR datasets, which was sufficient for convergence in each model. We estimated the heldout perplexity using one independent Monte Carlo sample per batch. Computations were performed on graphics processing units (NVIDIA Tesla M40, K80 and V100 GPUs), with double precision, and we used gradient accumulation to reduce memory usage. Single training runs ranged from ~30 min. for smaller datasets (CDKN1B and VE6) to ~2.5 hours for larger datasets (DHFR, PINE and TCR).

S6 T-Cell Receptor Analysis

We downloaded a publicly available dataset of 6,327 T-cell receptor (TCR) sequences found in CD8+ cytotoxic T-cells `https://support.10xgenomics.com/single-cell-vdj/datasets/2.2.0/vdj_v1_hs_cd8_t` 10x Genomics (2018). These were sequenced using single cell sequencing of peripheral blood mononuclear cells obtained from an individual healthy donor. Internal stop codons were removed from the sequence. We used the provided CellRanger annotations of chain features. Annotations of the reference structure PDB:2BNR are based on IgBLAST annotations (Ye et al., 2013) of the nucleotide sequence of 1G4 TCRβ obtained from Robbins et al. (2008), and translated from nucleotides into the corresponding positions in the amino acid sequence (Figure S7).

To obtain a latent space representation (Figure 5B), we trained the FactorMuE observation model with $T = 2$ latent dimensions, and chose among training runs based on a randomly held out test set (5% of the data). Hyperparameters were otherwise set as in Section S5. The shift ν is estimated using the variational approximation to the posterior of the FactorMuE (using 10 Monte Carlo samples). $\hat{w}_{\text{ref}}$ is estimated using a single sample from the variational approximation to the

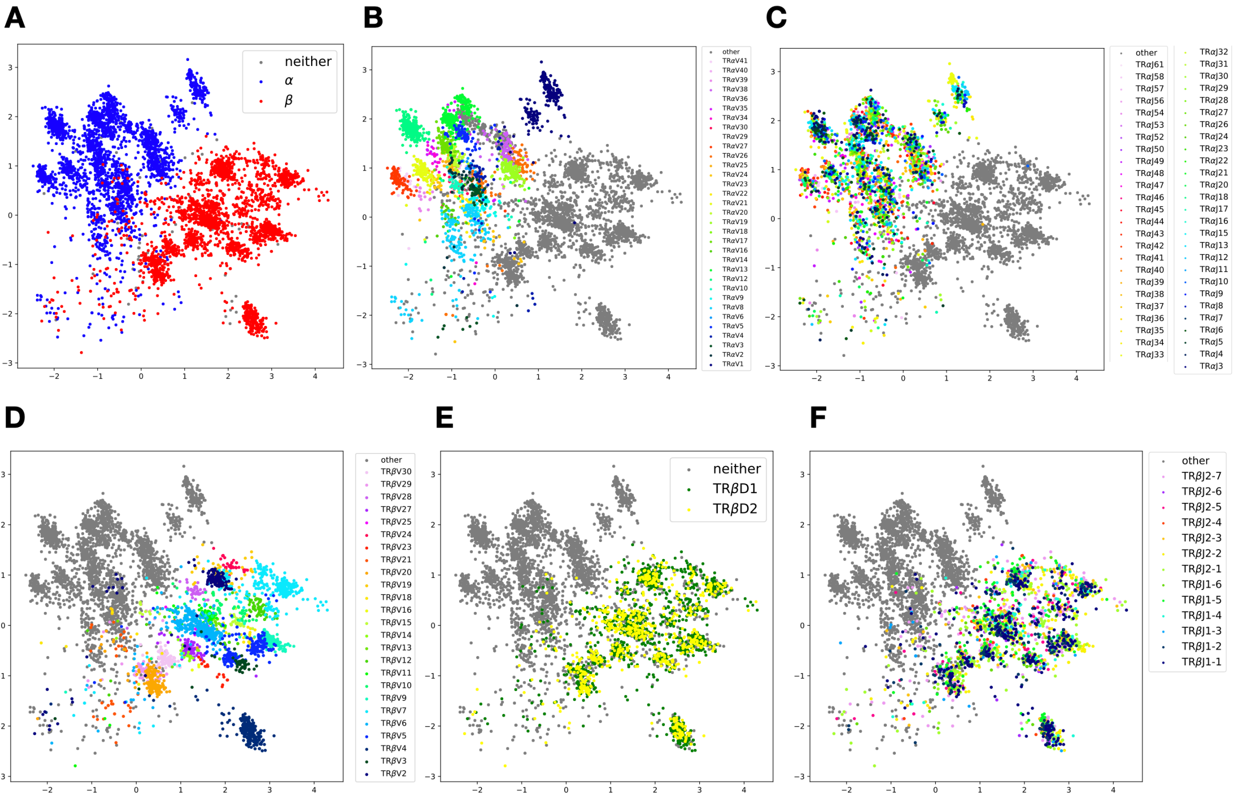


Figure S5: **Latent space representation of human T-cell receptor sequences, colored by supervised annotations.** Annotations from 10x Genomics (2018). (A) C_α versus C_β. (B) α chain V types. (C) α chain J types. (D) β chain V types. (E) β chain D types. (F) β chain J types and subtypes.

posterior and the Viterbi algorithm.

S6.1 Further results

Along feature vector 2 (Figure 5D) we found weak positive correlation between the magnitude of variation and the relative surface accessibility of each site (Spearman correlation $\rho = 0.20$, $p < 0.02$; Figure S6). Along feature vector 1 (Figure 5D) we observed high values of ν_l in the V segment, suggesting that there are systematic and heterogeneous differences between the V segment sequence distribution used in TCRα chains and in TCRβ chains To confirm the observation, we used the RegressMuE model to predict the entire TCR sequence based just on its annotation as TCRα or TCRβ. In particular, as covariate vector H_i we used a one-hot encoding of the chain type annotated by CellRanger; sequences without an annotation were encoded as $(0, 0)$. We computed the regression shift ν_l in the same way as Equation 3, with the covariate H in place of z. Figure S8 plots the shift in amino acid preference between the two chains, showing that at a population level there are key positions within the variable region with substantial differences in preference.

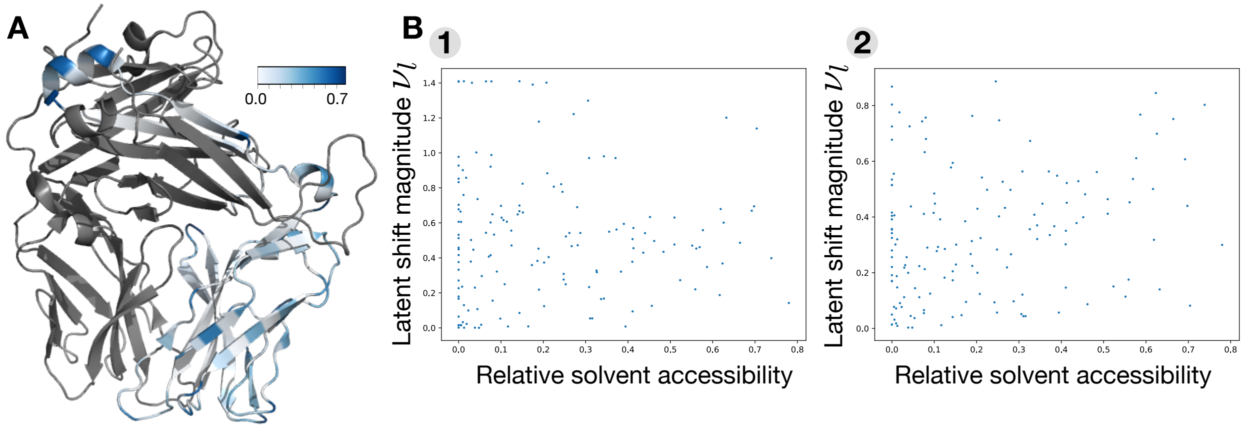


Figure S6: **Comparing MuE observation model features to T-cell receptor relative solvent accessibility.** (A) Relative solvent accessibility of TCRβ from the structure PDB:2BNR (Chen et al., 2005) (the TCRα chain is shown in gray), computed using DSSP (Kabsch and Sander, 1983) and the maximum values in Tien et al. (2013) with the Biopython API (Cock et al., 2009). (B) Residue relative solvent accessibility versus FactorMuE shift magnitude ν_l along vector 1 and vector 2 from Figure 5D. The correlation between the shift along vector 1 and the accessibility is Spearman $\rho = 0.039$, $p = 0.64$.

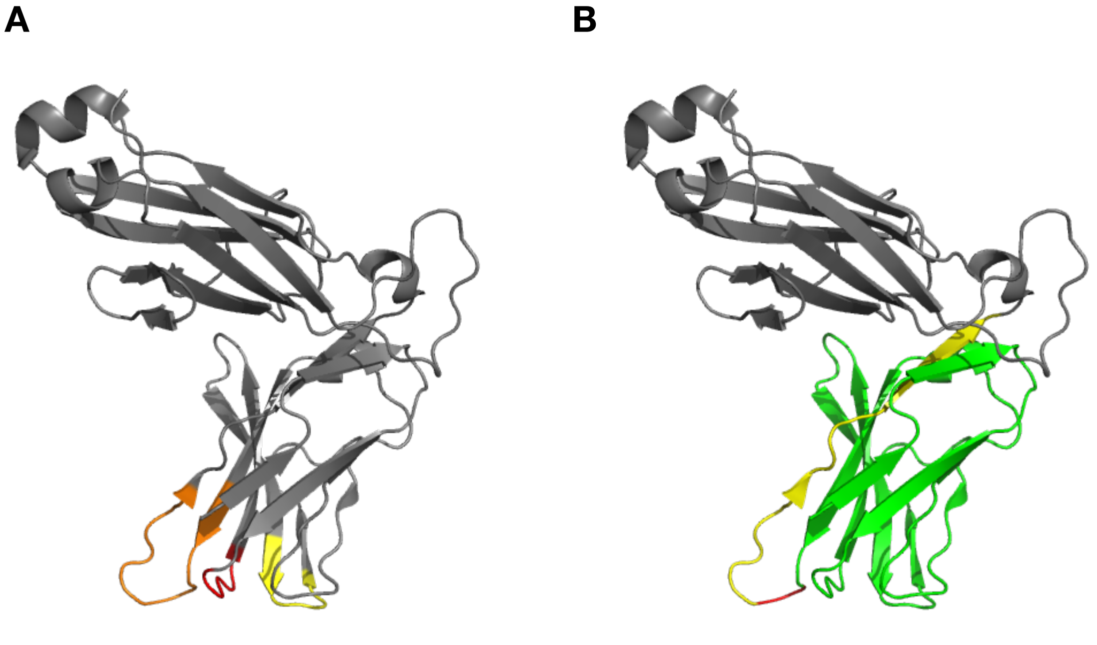


Figure S7: **T-cell receptor structural annotations.** (A) CDR segments of PDB:2BNR chain E (Chen et al., 2005), based on IgBLAST annotations (Ye et al., 2013) of the nucleotide sequence of 1G4 TCRβ obtained from Robbins et al. (2008), and translated from nucleotides into the corresponding positions in the amino acid sequence. CDR1 in red, CDR2 in yellow and CDR3 in orange. (B) V (green), J (yellow) and junction (red) segments of the 1G4 nucleotide sequence, based on the IgBLAST annotations, and translated from nucleotides.

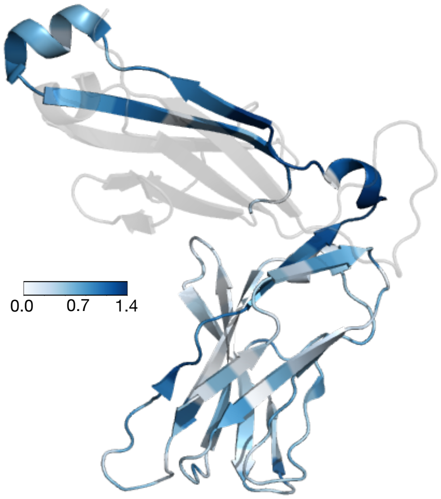


Figure S8: **Shift ν from chain α to chain β sequences learned by the RegressMuE model.** ν_l was computed as in Equation 3, using the chain annotation in place of the latent variable z.

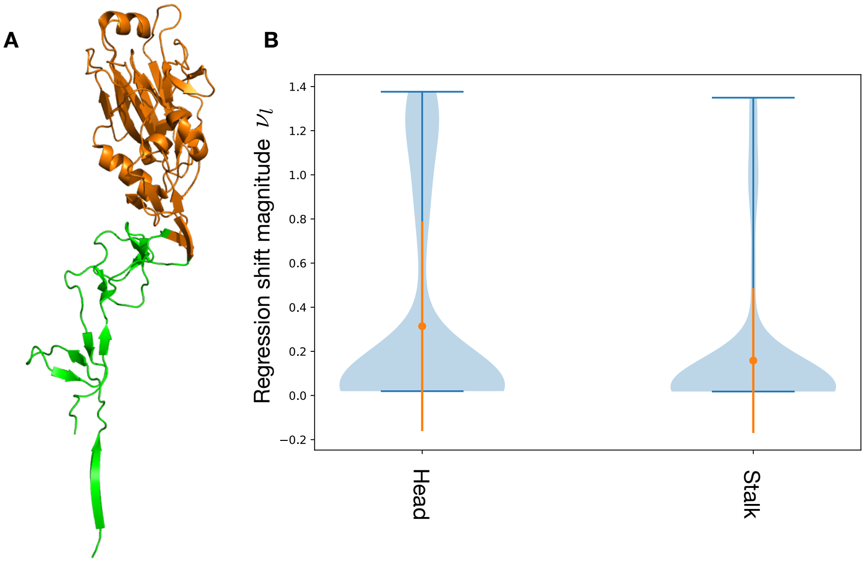


Figure S9: **Comparing RegressMuE model coefficients to HA1 structural domains.** (A) Head (orange) and stalk (green) domains of the HA1 protein (PDB:4O5N); residues between sites 52 and 277 are defined as the head domain, and all others as stalk, following Lee et al. (2018). (B) Violin plots of regression shift ν_l (Equation S67) for residues in the head domain (226 residues) versus the stalk domain (103 residues). Mean and standard deviation are shown in orange.

S7 Influenza Analysis

We downloaded publicly available influenza A(H3N2) HA sequences from GISAID (Shu and McCauley, 2017). We selected only sequences longer than 500 amino acids and with no ambiguous amino acids. Some sequences were labeled at different levels of time resolution, with annotations providing months or years rather than days; we assumed month and/or day were missing at random

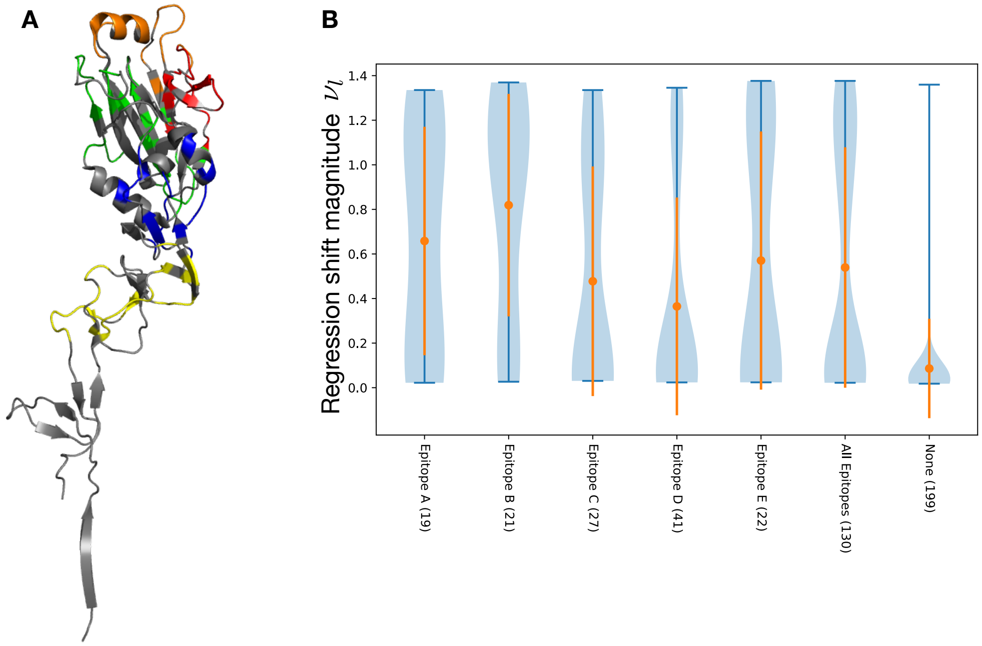


Figure S10: **Comparing MuE observation model regression coefficients to HA1 epitope regions.** (A) Epitope regions A (red), B (orange), C (yellow), D (green), E (blue) (Wiley et al., 1981; Muñoz and Deem, 2005). (B) Violin plots of regression shift ν_l (Equation S67) for residues in each epitope region, for all epitope regions together, and for residues not in any epitope region; the number of residues in each region is shown in parenthesis. Mean and standard deviation are shown in orange.

and imputed them uniformly at random. Following Lee et al. (2018), we randomly subsampled six sequences per month, from 1968 to October 2019, to form the dataset. In the forecasting experiments we removed the mis-annotated data identified in the 2008 outlier cluster marked by ‡ in Figure 6E prior to subsampling (GISAID identifiers `EPI_ISL_24813`, `EPI_ISL_24814`, ..., `EPI_ISL_24867`). Accession numbers for the complete dataset can be found in Supplementary Table 1; our results were stable upon resampling. We extracted only the first 350 amino acids of each HA sequence, covering HA1 in the reference A(H3N2) numbering (Burke and Smith, 2014).

We used $M = 361$ in the MuE distribution. We set the prior on indels to $\mu^{(r)} = \mu^{(u)} = (1000, 1)$. We trained each model for 7 epochs, which was sufficient for convergence. Hyperparameters and training schedule were otherwise set as in Section S5. To produce the latent embedding in Figure 6D, however, we annealed the ELBO weighting χ only up to $\chi_0 = 0.001$ after 7 epochs, providing only very weak prior regularization such that the embedding corresponds to approximately the maximum likelihood estimator of z (and we avoid posterior collapse).

To visualize features, we trained the RegressMuE model on the full time period (1968 to 2019), with 5% of datapoints randomly held out to choose among training runs. We computed the magnitude of the shift in sequence space from time t_0 to time t_1 in the RegressMuE as

$$\nu_l = \left[\sum_{b=1}^{B} \left(\mathbb{E}[Y_{l,b} | \hat{w}_{\text{ref}}, t = 2019] - \mathbb{E}[Y_{l,b} | \hat{w}_{\text{ref}}, t = 1968] \right)^2 \right]^{1/2} \tag{S67}$$

using as reference the HA1 sequence from PDB:4O5N. The expectation is estimated using the variational approximation to the posterior with 10 Monte Carlo samples. $\hat{w}_{\text{ref}}$ is estimated using

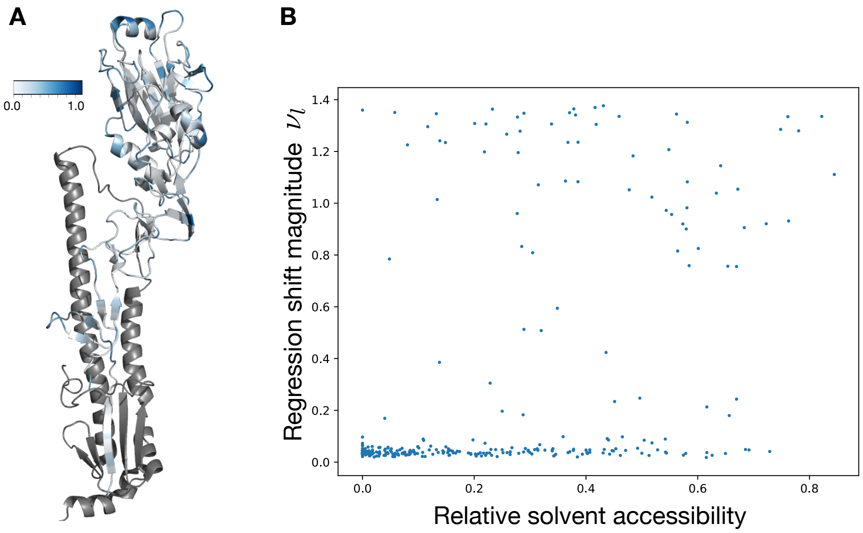


Figure S11: **Comparing MuE observation model regression coefficients to HA1 relative solvent accessibility.** (A) Relative solvent accessibility of the HA1 protein (PDB:4O5N), computed using DSSP (Kabsch and Sander, 1983) and the maximum values in Tien et al. (2013) with the Biopython API (Cock et al., 2009). HA2 protein shown in dark gray. (B) Relative solvent accessibility versus regression shift magnitude ν_l (Equation S67), residue-by-residue. Spearman $\rho = 0.41$, $p < 10^{-13}$.

a single sample from the variational approximation to the posterior and the Viterbi algorithm. In evaluating the association between the shift vector ν_l and epitope regions of HA1, we specifically compared to the 16 sites with clear antigenic selection in at least one human sera identified in Lee et al. (2019).

S7.1 Further results

In addition to the classic epitope regions, we also compared the regression shift ν to the structural domains of the HA1 protein (Figure S9), relative solvent accessibility (Figure S11), and relative amino acid preference in a deep mutational scan evaluating fitness effects of mutations (Figure S12).

The cluster marked ‡ in Figure 6E appears around 2008 but the latent representation of these sequences is close to that of sequences from the late 1960s or 1970s; this cluster comes from an experiment performed in 2008 on 1968 sequences, rather than contemporary patient samples as in the rest of the GISAID dataset.

MuE observation models can be used to generate samples of future sequences, enabling experimental tests of immune response and antibody titer on sequences that are likely to emerge in the future. We generated samples for the year 2024 from the RegressMuE, and confirmed that they are similar to previously observed sequences, as would be expected (Figure S13). In particular, we sampled from

$$\begin{aligned} \phi &\sim q(\phi|Y_{\mathcal{D}}) \\ Y_i &\sim p_{\text{RegressMuE}}(y|\hat{w}_{\text{ref}}, \phi, t = 2024) \end{aligned} \tag{S68}$$

where $q(\phi|Y_{\mathcal{D}})$ is the variational approximation to the posterior over model parameters, under the model trained on the full time period (1968 to 2019), and PDB:4O5N is again used as a reference sequence.

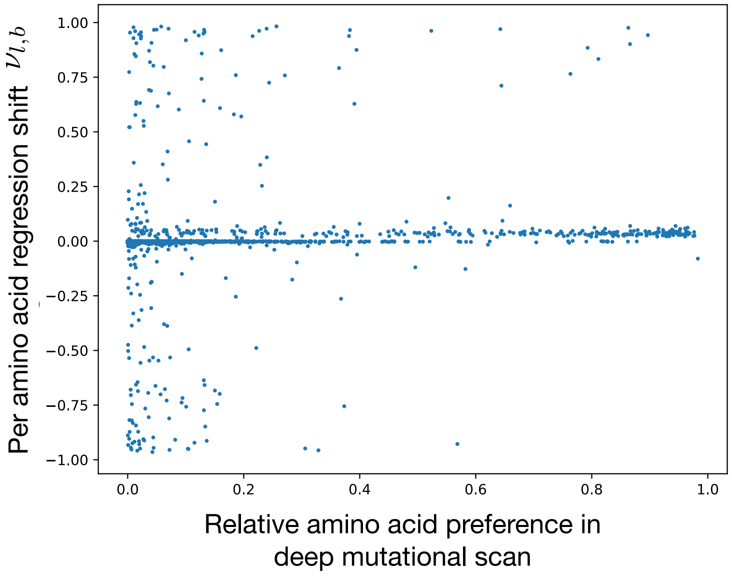


Figure S12:
Comparing MuE observation model regression coefficients to a deep mutational scan of HA. X-axis: regression shift for each amino acid at each position from 1968 to 2019,

$$\nu_{l,b} := \mathbb{E}[y_{l,b}|\hat{w}_{\text{ref}}, t = 2019] - \mathbb{E}[y_{l,b}|\hat{w}_{\text{ref}}, t = 1968]$$

(terms defined as in Equation S67). Y-axis: relative preference for point mutants with amino acid b at position l in the deep mutational scan performed in Lee et al. (2018). Spearman $\rho = 0.08$, $p < 10^{-11}$.

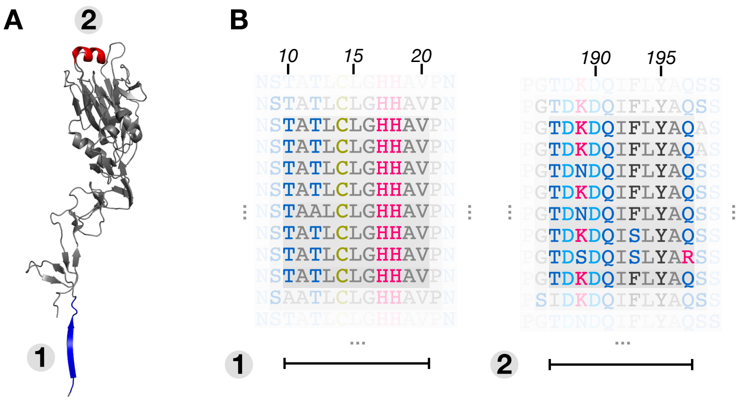


Figure S13: **Generating forecasted samples.** (A) Two locations in the reference structure PDB:4O5N, indicated in blue and red, corresponding to low and high ν_l values (Figure 6B). (B) Segments of sequences sampled from the posterior predictive distribution for the year 2024. The alignment variable w_{ref} is fixed based on the reference (PDB:4O5N), such that segments 1 and 2 correspond to the annotated structural features in A, and the column numbering is standard for influenza A(H3N2) (Burke and Smith, 2014).

S8 Supplementary Code Description

An Edward2 implementation of a MuE distribution and example MuE observation models are provided in `software/MuE`. A copy of Edward2 is also provided in `software/Edward2`, since Edward2 does not yet have a stable release.

References

HMMER. http://hmmer.org/. Accessed: 2020-5-18.

1000 Genomes Project Consortium, A. Auton, L. D. Brooks, R. M. Durbin, E. P. Garrison, H. M. Kang, J. O. Korbel, J. L. Marchini, S. McCarthy, G. A. McVean, and G. R. Abecasis. A global reference for human genetic variation. *Nature*, 526(7571):68–74, Oct. 2015.

10x Genomics. CD8+ T cells isolated from PBMCs of a healthy donor - direct TCR enrichment, Aug. 2018. URL https://support.10xgenomics.com/single-cell-vdj/datasets/2.2.0/vdj_v1_hs_cd8_t?

A. A. Alemi, B. Poole, I. Fischer, J. V. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken ELBO. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20(28):1–6, 2019.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *J. Am. Stat. Assoc.*, 112(518):859–877, Apr. 2017.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 2016.

D. F. Burke and D. J. Smith. A recommended numbering scheme for influenza A HA subtypes. *PLoS One*, 9(11):e112302, Nov. 2014.

J.-L. Chen, G. Stewart-Jones, G. Bossi, N. M. Lissin, L. Wooldridge, E. M. L. Choi, G. Held, P. R. Dunbar, R. M. Esnouf, M. Sami, J. M. Boulter, P. Rizkallah, C. Renner, A. Sewell, P. A. van der Merwe, B. K. Jakobsen, G. Griffiths, E. Y. Jones, and V. Cerundolo. Structural and kinetic basis for heightened immunogenicity of T cell vaccines. *J. Exp. Med.*, 201(8):1243–1255, Apr. 2005.

P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, June 2009.

R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

E. Gasteiger, C. Hoogland, A. Gattiker, S. Duvaud, M. R. Wilkins, R. D. Appel, and A. Bairoch. Protein identification and analysis tools on the ExPASy server. In J. M. Walker, editor, *The Proteomics Protocols Handbook*, pages 571–607. Humana Press, Totowa, NJ, 2005.

S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U. S. A.*, 89(22):10915–10919, Nov. 1992.

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

T. A. Hopf, A. G. Green, B. Schubert, S. Mersmann, C. P. I. Schärfe, J. B. Ingraham, A. Toth-Petroczy, K. Brock, A. J. Riesselman, P. Palmedo, C. Kang, R. Sheridan, E. J. Draizen, C. Dallago, C. Sander, and D. S. Marks. The EVcouplings python framework for coevolutionary sequence analysis. *Bioinformatics*, 35(9):1582–1584, May 2019.

W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, Dec. 1983.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

D. P. Kingma and M. Welling. Auto-Encoding variational bayes. Dec. 2013.

A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference. *J. Mach. Learn. Res.*, 18(14):1–45, Jan. 2017.

J. M. Lee, J. Huddleston, M. B. Doud, K. A. Hooper, N. C. Wu, T. Bedford, and J. D. Bloom. Deep mutational scanning of hemagglutinin helps predict evolutionary fates of human H3N2 influenza variants. *Proc. Natl. Acad. Sci. U. S. A.*, 115(35):E8276–E8285, Aug. 2018.

J. M. Lee, R. Eguia, S. J. Zost, S. Choudhary, P. C. Wilson, T. Bedford, T. Stevens-Ayers, M. Boeckh, A. C. Hurt, S. S. Lakdawala, S. E. Hensley, and J. D. Bloom. Mapping person-to-person variation in viral mutations that escape polyclonal serum targeting influenza hemagglutinin. *Elife*, 8, Aug. 2019.

E. T. Muñoz and M. W. Deem. Epitope analysis for influenza vaccine design. *Vaccine*, 23(9): 1144–1148, Jan. 2005.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–453, 1970.

R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, 2014.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

P. F. Robbins, Y. F. Li, M. El-Gamil, Y. Zhao, J. A. Wargo, Z. Zheng, H. Xu, R. A. Morgan, S. A. Feldman, L. A. Johnson, A. D. Bennett, S. M. Dunn, T. M. Mahon, B. K. Jakobsen, and S. A. Rosenberg. Single and dual amino acid substitutions in TCR CDRs can enhance antigen-specific T cell functions. *J. Immunol.*, 180(9):6116–6131, May 2008.

S. Särkkä and Á. F. García-Fernández. Temporal parallelization of bayesian smoothers. *IEEE Trans. Automat. Contr.*, 66(1):299–306, 2020.

Y. Shu and J. McCauley. GISAID: Global initiative on sharing all influenza data - from vision to reality. *Euro Surveill.*, 22(13), Mar. 2017.

B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, Mar. 2015.

J. L. Thorne, H. Kishino, and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 33(2):114–124, Aug. 1991.

M. Z. Tien, A. G. Meyer, D. K. Sydykova, S. J. Spielman, and C. O. Wilke. Maximum allowed solvent accessibilites of residues in proteins. *PLoS One*, 8(11):e80635, Nov. 2013.

D. Tran, M. Hoffman, D. Moore, C. Suter, S. Vasudevan, A. Radul, M. Johnson, and R. A. Saurous. Simple, distributed, and accelerated probabilistic programming. In *Neural Information Processing Systems*, 2018.

UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.*, 47(D1): D506–D515, Jan. 2019.

S. Vogel, H. Ney, and C. Tillmann. HMM-based word alignment in statistical translation. In *Proc. of the 16th International Conference on Computational Linguistics (COLING '96)*, pages 836–841, 1996.

D. C. Wiley, I. A. Wilson, and J. J. Skehel. Structural identification of sites of hong kong influenza and their involvement in antigenic variation. *Nature*, 289, 1981.

J. Ye, N. Ma, T. L. Madden, and J. M. Ostell. IgBLAST: an immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Res.*, 41(Web Server issue):W34–40, July 2013.