Generative probabilistic biological sequence models that account for mutational variability

Eli N. Weinstein
Program in Biophysics
Harvard University
eweinstein@g.harvard.edu

Debora S. Marks
Department of Systems Biology
Harvard Medical School
debbie@hms.harvard.edu

July 31, 2020

Abstract

Large-scale sequencing has revealed extraordinary diversity among biological sequences, produced over the course of evolution and within the lifetime of individual organisms. Existing methods for building statistical models of sequences often preprocess the data using multiple sequence alignment, an unreliable approach for many genetic elements (antibodies, disordered proteins, etc.) that is subject to fundamental statistical pathologies. Here we introduce a structured emission distribution (the MuE distribution) that accounts for mutational variability (substitutions and indels) and use it to construct generative and predictive hierarchical Bayesian models (H-MuE models). Our framework enables the application of arbitrary continuous-space vector models (e.g. linear regression, factor models, image neural-networks) to unaligned sequence data. Theoretically, we show that the MuE generalizes classic probabilistic alignment models. Empirically, we show that H-MuE models can infer latent representations and features for immune repertoires, predict functional unobserved members of disordered protein families, and forecast the future evolution of pathogens.

1 Introduction

High-throughput sequencing has become pervasive across modern biology and biomedicine, and has revealed extraordinary sequence diversity among proteins, RNA, and other genetic elements. Interpreting that diversity, and making predictions about unobserved or future sequences, is an open challenge, with relevance to epidemiology (predicting pathogen evolution), immunology (characterizing antibody repertoires), molecular evolution (mapping substructure within protein families), protein design, and far more. Accomplishing these goals requires tools for working with high-dimensional complex sequence distributions. In principal, generative probabilistic models of biological sequences could enable discovery of rare subpopulations, key sequence features, trends across time, the impact of experimental interventions, etc., and then convert this understanding into predictions of new sequences that could be synthesized and tested in the laboratory.

There are a variety of methods that have seen widespread success on similar challenges in other fields of science, but adapting them to structured data such as biological sequences is non-trivial. In particular, there is an enormous wealth of generative models of continuous-space vectors, such as linear regression, probabilistic PCA [1], non-negative matrix factorization [2], and image neural networks [3]. In order to apply continuous-space vector models to types of data besides continuous-space vectors, statisticians typically rely on an *emission* distribution, e.g. a Poisson distribution for rare count data or a zero-inflated negative binomial distribution for single cell RNA expression data [4, 5, 6]. For instance, if $p(v|\theta)$ is a generative model of continuous-space vectors, with parameter θ, a generative model of count vectors y can be built with a Poisson emission distribution as:

$$\begin{aligned} v &\sim p(v|\theta) \\ y &\sim \text{Poisson}(\lambda = \exp(v)). \end{aligned} \tag{1}$$

While there is often a wide variety of possible choices for an emission distribution, a good emission distribution should not only generate the right type of data, but also capture the variability commonly seen in the data.

To enable the principled application of continuous-space vector models to biological sequences, we propose the "mutational emission", or "MuE" distribution. This new distribution generates unaligned sequence data while explicitly accounting for the kinds of variability commonly seen in biological sequence data, namely substitution, insertion and deletion mutations. Using the MuE as an emission distribution enables the direct application of continuous-space vector models to biological sequences; we term these combined models "hierarchical MuE" or H-MuE models. H-MuE models do not require a multiple sequence alignment of the data, which is often unreliable in practice, especially for disordered proteins, antibodies, promoters, and other genetic elements, and is pathological in theory when the ultimate aim is sequence prediction (Section S2). Instead, H-MuE models represent the multiple sequence alignment implicitly as a latent variable, making it possible to account rigorously for alignment uncertainty. Classical probabilistic alignment models can be re-derived as special cases of the MuE distribution. Moreover, in contrast to alternative models of biological sequence mutation, the likelihood function of the MuE distribution is analytically tractable and differentiable [7]. This helps enable inference of the parameters of H-MuE models from data using Bayes' rule, allowing in particular the use of scalable approximate inference algorithms that rely on automatic differentiation (also known as backpropagation) [8, 9].

We demonstrate empirically how H-MuE models can be applied to large sequence datasets to map the biological diversity found among disordered proteins, viruses, immune receptors, and more. We show that H-MuE models can be used to predict unobserved and future protein sequences, and enable unsupervised learning of sequence subpopulations and sequence features.

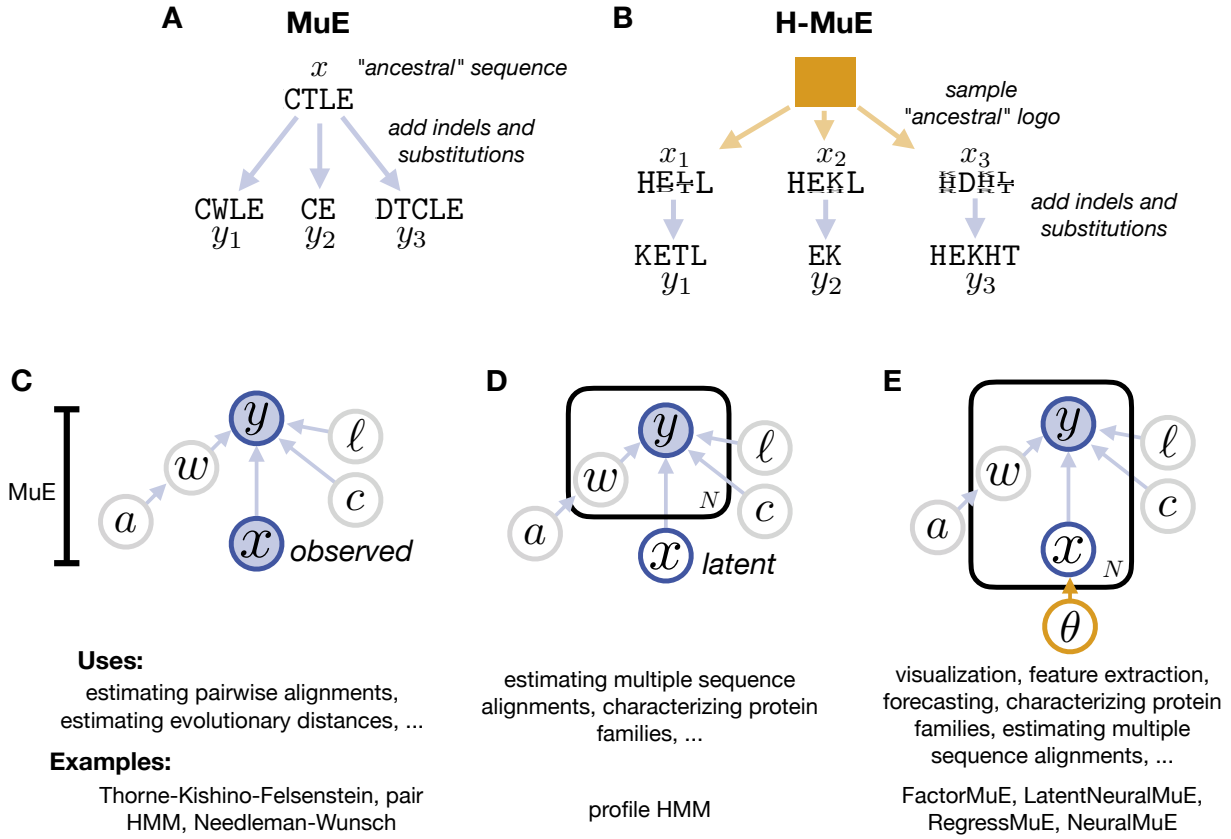


Figure 1: Hierarchical mutational emission (H-MuE) models generalize and extend previous probabilistic mutation models and alignment methods. A. The MuE model generates samples y_i that are mutants of an initial sequence x (see Figure S1 for a detailed illustration). B. In an H-MuE model, v_i is sampled from an initial continuous vector model $p(v|\theta)$ and determines the sequence logo $x_i = \text{softmax}(v_i)$; then y_i is sampled from x_i according to the MuE distribution (see Figure S2 for a detailed illustration). C,D,E. Graphical models of alternative use cases for the MuE distribution, along with examples of models for each use case (based on our theoretical results, Section S4). C. In some situations, sequence x is observed as well as y; estimating the hidden state variable of the MuE distribution can provide an alignment between the two sequences, and estimating the parameters can give insight into their evolutionary relatedness. D. A collection of sequences $y_1, ..., y_N$ can be modeled as mutants of an unobserved "ancestral" sequence or sequence logo. The pHMM fits this graphical model. E. In H-MuE models, each sequence y_i is associated with its own individual "ancestor" x_i, drawn from a population determined by θ.

2 Results

2.1 The MuE distribution

We consider datasets of unaligned biological sequences y_i for $i \in \{1, ..., N\}$, which may be recorded from different species, organisms, cells, etc. In order to model the distribution of sequences, and how this distribution may depend on any covariates, we designed two-

layer generative models. The first layer generates an "ancestral" sequence logo x_i with fixed length; the second step adds mutations, including substitutions, insertions and deletions to generate y_i. The first step relies on a continuous-space matrix model $p(v|\theta)$, where $v \in \mathbb{R}^{M\times D}$, which may be any model of interest and may depend on covariates such as sequence collection time. The second step employs the proposed mutational emission distribution $\text{MuE}(x, c, a, \ell)$ (details in Box 1), which describes a distribution of mutants of x with indel probabilities controlled by the parameter a, substitution probabilities controlled by the parameter ℓ, and insertion sequences controlled by the parameter c (Figure 1A). The complete H-MuE model is:

$$\begin{aligned} v_i &\sim p(v|\theta) \\ y_i &\sim \text{MuE}(x_i = \text{softmax}(v_i), c, a, \ell). \end{aligned} \tag{2}$$

where the softmax linker function sets $x_{i,m,d} = \exp(v_{i,m,d}) / \sum_{d'} \exp(v_{i,m,d'})$ for $m \in \{1, ..., M\}$ and $d, d' \in \{1, ..., D\}$. We will refer to models that fit the form of Equation 2 as "hierarchical MuE" (H-MuE) models, since they generate sequences according to a hierarchical Bayesian model, first employing $p(v|\theta)$ to generate ancestral sequence logos $x_1, ..., x_N$, then employing the MuE distribution to add additional mutations (Figure 1B). More generally, c, a and ℓ may also be made to depend on v_i (Section S5).

Given a dataset of sequences $y_{1:N}$, we would like to infer the parameters of an H-MuE model (in particular, θ, c, a and ℓ) using Bayes' theorem. While exact inference is computationally intractable, we can approximate the posterior distribution using variational inference [10]. We use stochastic black box variational inference with the reparameterization trick, relying on the crucial property of the MuE distribution that its likelihood $p_{\text{MuE}}(y_i|x_i, c, a, \ell)$ is a differentiable function of the parameters x_i, c, a and ℓ [11]. For models with local latent variables, we also use a recognition network to amortize computation across datapoints [12, 13]. When implemented for modern computing hardware (graphics processing units), these techniques together enable scalable approximate inference in H-MuE models for a wide variety of continuous-space models $p(v|\theta)$ (Section S6) [8].

Box 1: MuE Mathematical Details

The MuE is a structured hidden Markov model (HMM), with initial transition vector $a^{(i)}$, transition matrix $a^{(t)}$ and discrete emission matrix $e = (\xi \cdot x + \zeta \cdot c) \cdot \ell$, where ξ and ζ are fixed constants. The matrix ξ has shape $K \times M$, x has shape $M \times D$, ζ has shape $K \times (M+1)$, c has shape $(M+1) \times D$ and ℓ has shape $D \times B$, where M is the length of the ancestral sequence, $K = 2(M+1)$ is the size of the Markov chain state space, D is the alphabet size of the ancestral sequence and B is the alphabet size of the observed sequence (typically $D = B$). Each row of the matrices x, c and ℓ is a vector of discrete probabilities that sum to one, ie. $x_m \in \Delta_{D-1} = \{v \in \mathbb{R}^D : v_d \geq 0, \sum_{d=1}^{D} v_d = 1\}$. The constants ξ and ζ are defined by $\xi_{k,m} = \delta_{k,2m}$ and $\zeta_{k,m} = \delta_{k,2m-1}$, where $\delta_{k,k'}$ is the Kronecker delta, ie. $\delta_{k,k'} = 1$ if $k = k'$ and $\delta_{k,k'} = 0$ if $k \neq k'$. Crucially, the transition matrix $a^{(t)}$ must satisfy the restriction that $a^{(t)}_{k,k'} = 0$ for all k, k' such that (1) state k is accessible from the initial state and (2) $k' + k'\%2 - k + k\%2 \leq 0$, where $\%$ is the modulo (remainder) operation.

To see how the MuE represents alignments, we rewrite $\text{MuE}(x, c, a, \ell)$ as

$$\begin{aligned} w &\sim \text{MarkovModel}(a^{(i)}, a^{(t)}) \\ y &\sim \text{Categorical}\big(w \cdot (\xi \cdot x + \zeta \cdot c) \cdot \ell\big) \end{aligned} \tag{3}$$

where w is a one-hot encoding of the states of a trajectory sampled from the Markov model. w has size $L \times K$, where L, the length of the trajectory, is itself a random variable and determines the length of the sampled sequence y (Section S3.1). We see from Equation 3 that w acts like a matrix of regression coefficients, determining which position in the regressor ancestral sequence x influences each position in the regressand mutated sequence y. In Section S4.1 we formalize a mapping between the variable w and a pairwise alignment between x and y, and prove that the restrictions on the transition matrix $a^{(t)}$ are necessary and sufficient to guarantee that this mapping exists.

2.2 The MuE distribution generalizes probabilistic sequence alignment models and methods

Before employing H-MuE models in practice, we analyzed the MuE distribution theoretically. First we showed that if we choose the parameters a and ℓ such that there is zero probability of insertions, deletions or substitutions, the MuE distribution reduces to a multivariate categorical distribution, a standard emission distribution for pre-aligned sequences (Section S3.2). We used this fact to show that H-MuE models generalize state-of-the-art aligned sequence models [14].

Next, we showed that standard probabilistic mutation and alignment models are also special cases of the MuE distribution. First, we proved that for a particular setting of a and ℓ, the MuE distribution reduces to the transition distribution of the Thorne-Kishino-Felsenstein model, a continuous-time stochastic process model of sequence evolution that includes indels and satisfies detailed balance (Section S4.2) [15]. Second, we showed that

for another setting of a and ℓ, the MuE distribution matches the conditional distribution of y given x under the pair hidden Markov model, a common probabilistic pairwise sequence alignment model (Section S4.3) [16]. These two models have usually been employed in contexts where x is observed (that is, x is a one-hot encoding of a particular sequence) (Figure 1C). Next, we showed that for another setting of a, and with ℓ the identity matrix, the MuE distribution reduces to the profile hidden Markov model (pHMM) (Figure S4.4) [16]. The pHMM has usually been employed in contexts where x is latent and we observe many sequences $y_{1:N}$ (Figure 1D). While none of these previous models have been explicitly used as emission distributions (Figure 1E), they have been highly successful across a range of other biological sequence analysis problems, providing evidence that the MuE model can effectively capture common forms of variability among biological sequences.

Finally, we specifically investigated the connection between the MuE distribution and biological sequence alignments. The MuE distribution is a type of structured hidden Markov model, and, in the context of the previous probabilistic alignment models discussed above, the state variable of the hidden Markov model has been interpreted as an alignment between sequences x and y. Indeed, in addition to probabilistic alignment methods, the MuE is also connected to non-probabilistic alignment methods: we proved that for yet another setting of a and ℓ the maximum *a posteriori* estimator of the state variable corresponds to the Needleman-Wunsch alignment between x and y (Section S4.5). In general, we established necessary and sufficient conditions on the transition matrix of the MuE distribution guaranteeing that the state variable can always be interpreted as an alignment (Section S4.1). As a consequence, any MuE or H-MuE model can be used not only as a generative sequence model but also as a probabilistic sequence alignment method, and Bayesian inference of the H-MuE parameters yields a posterior distribution over multiple sequence alignments of the dataset. Our variational inference procedure for H-MuE models marginalizes out the MuE state variable during training, in effect considering all possible alignments of the dataset to address the problem of alignment uncertainty.

2.3 H-MuE models describe complex sequence diversity

Prediction

We first sought to evaluate the ability of H-MuE models to accurately model the distribution of sequences within protein families and to generate new sequences that satisfy the implicit functional constraints of those families. As a baseline we compared our models to a profile HMM, the most widely-used generative model of unaligned protein sequence families (Section S5.1). We then used the MuE to extend two of the most successful continuous-space vector models, probabilistic PCA and a neural network latent variable model, to unaligned sequences, creating the "FactorMuE" and the "LatentNeuralMuE" respectively (Sections S5.3 and S5.5). These three models – the pHMM, FactorMuE and LatentNeuralMuE – have sequentially increasing model complexity; the FactorMuE and LatentNeuralMuE are capable of representing long-distance epistasis. We measured the capacity of each model to predict unobserved, presumably functional, members of the protein family (Figure 2A). We quantified performance in terms of the heldout per residue perplexity, a metric that is monotonically related to the heldout log likelihood, and is interpretable as the effective number

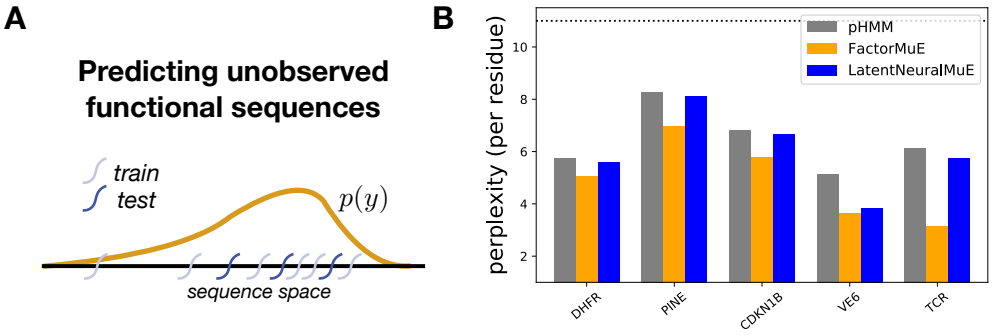


Figure 2: Predicting unobserved functional sequences with H-MuE models. A. Illustration of random train-test data split, used to validate predictions of unobserved members of a sequence family (density estimation). B. Per residue perplexity on the heldout dataset, for the baseline pHMM and two H-MuE models, across five sequence datasets; lower numbers indicate better performance. A BLOSUM62 substitution model that exactly describes the data distribution will have perplexity of 11.0, indicated by a dotted line (Section S7).

of amino acid choices per position. Per residue perplexity ranges from 1 (perfect prediction) to 20 (naive prediction); a BLOSUM62 substitution matrix model, if it exactly describes the data, will have per residue perplexity of 11.0 (Section S7). We applied these models to five datasets of protein families, ranging in size from 1,000 to 10,000 sequences (Section S8). Four were taken from non-redundant sequence databases: sequences similar to dihydrofolate reductase (DHFR; a widely conserved enzyme, often studied with generative models of aligned sequences), serine recombinase (PINE; a tool for genomic engineering), cyclin dependent kinase inhibitor 1B (CDKN1B/p27; a cell cycle inhibitor with a disordered region) and the human papillomavirus E6 protein (VE6; an oncogenic protein with a disordered region) [17, 18, 19]. The final dataset consisted of human T-cell receptor (TCR) sequences from a healthy donor obtained using single cell sequencing (Section S9). We evaluated model performance on a randomly held out 10% of sequences. The results, summarized in Figure 2B, show that FactorMuE models offer a consistent and large improvement in predictive power 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, illustrating the advantages of using the MuE distribution to explore models of different complexity.

Latent space

Continuous-space vector models like probabilistic PCA are widely used in other fields to produce visualizations of complex datasets. We examined the two-dimensional embeddings produced by the FactorMuE model, which combines probabilistic PCA with the MuE emission distribution (Figure 3A). We focused on the TCR dataset, and evaluated the model's capacity to learn richly structured representations in an unsupervised way by cross referenc-

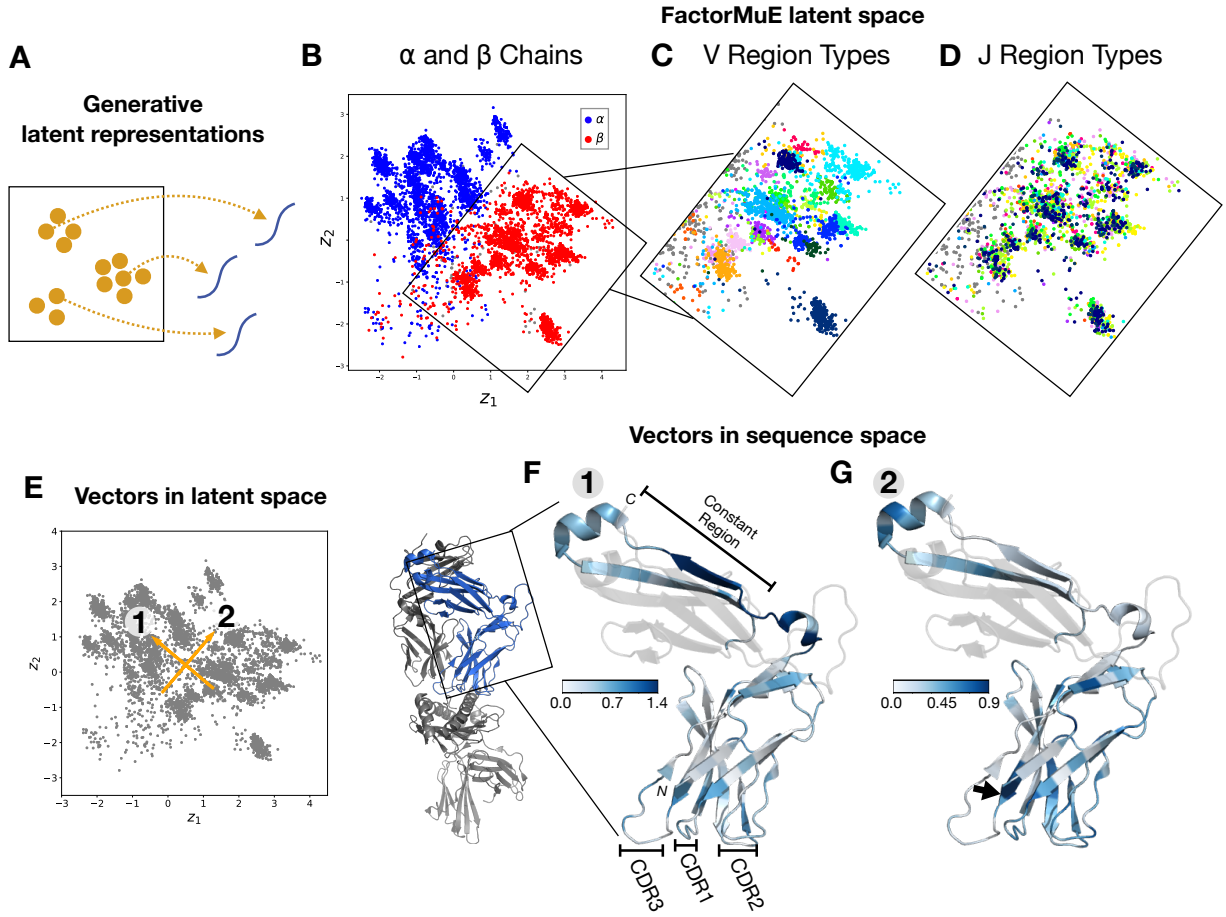


Figure 3: H-MuE model of a single-cell T-cell receptor repertoire. A. Cartoon of a generative model with a latent space sequence representation. B,C,D. The two-dimensional latent space of a FactorMuE model, learned from a TCR dataset taken from a single-cell sequencing experiment. B. Each sequence is colored according to its annotated chain type (grey corresponds to unannotated sequences). C. Each sequence is colored according to its V region type (α chains are excluded from the plot for visual clarity, see Figure S4). D. Each sequence is colored according to its J region type. E. A latent space vector normal to the hyperplane separating α from β chains (vector 1) and an orthogonal vector of the same length (vector 2). F,G. The projection of vectors 1 and 2 back into sequence space, using a reference TCR structure (PDB:2BNR). Residues are colored according to the norm of the shift in amino acid preference from the tail to the head of the latent vector (Equation S78). Transparent residues in the constant region correspond to the portion of the protein that was not sequenced in the experiment. The black arrow in G indicates the start of the CDR3 region, corresponding to the position with the largest shift in preference (Figure S6).

ing the latent representation with supervised annotations of the V, D and J segment types in each sequence. We found that the latent space is divided evenly in two, with one side containing TCRα sequences and one side TCRβ sequences (Figure 3B). Each side contains clusters, which correspond to each type of V segment (Figures 3C and S4B). Meanwhile, J

segments are distributed uniformly across their corresponding α or β half, reflecting their ability to recombine with different V segments (Figures 3D and S4C).

Features

By projecting latent space vectors back into sequence space, with the latent MuE alignment variable fixed, we can visualize the features learned by the FactorMuE model and obtain an overview of the major axes of variation in the human TCR repertoire (Section S9). Note that this approach differs fundamentally from standard analysis techniques which focus on cataloguing the usage frequency of different segments or CDR3s in that it describes what the fine-grained variation adds up to at the population level. 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) primarily determines the sequence of the constant region of the TCR, while the orthogonal vector (vector 2) primarily determines the sequence of the V chain (Figure 3FG). Along vector 2 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 S5). The region of largest variation, however, was the buried C-terminal end of the V segment, corresponding to the start of the CDR3 region, the key specificity-determining region (Figure S6). Interestingly, even along vector 1 we observe variation within the variable region, suggesting that there are systematic and heterogeneous differences between the V segment sequence distribution used in TCRα chains and that used in TCRβ chains. To confirm this observation, we developed another H-MuE model: we extended a linear regression model with a MuE distribution (RegressMuE) (Section S5.2). We then used the RegressMuE model to predict the entire TCR sequence based just on its annotation as a TCRα or TCRβ. Figure S7 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.

2.4 H-MuE models forecast sequence evolution

Prediction

We evaluated the capacity of H-MuE models to forecast future sequence evolution (Figure 4A). Influenza A is responsible for an estimated 500,000 deaths a year and is an ongoing pandemic threat [20]. It is also a model organism for understanding the dynamics of rapidly evolving pathogens, and forecasting its evolution is essential in preparing vaccines and designing therapeutics [21, 22]. Previous forecasting methods have focused on predicting the relative fitness of existing strains in future years, e.g. [21, 23], or the antigenic properties of newly emerged strains, e.g. [24]. We instead predict the full amino acid sequence of the HA1 protein, the primary site of interaction with the immune system [21, 25]. 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) (Section S10) [26]. In contrast to the datasets considered in Section 2.3, indels are considered rare, though not absent, in patient samples, and so this dataset also offers an opportunity to evaluate H-MuE models in a distinct regime from that considered previously.

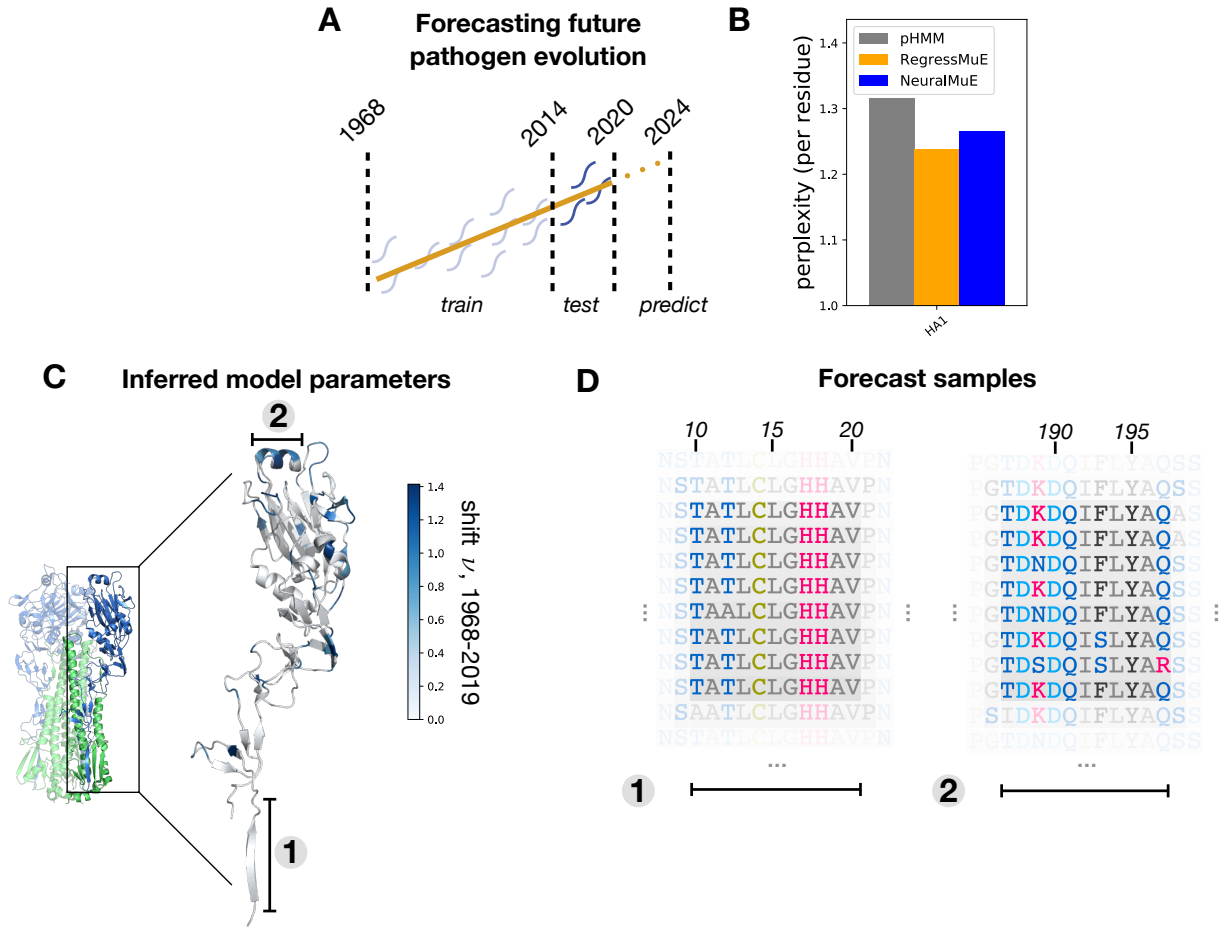


Figure 4: H-MuE model of influenza A(H3N2) evolution. A. Diagram of train-test split for influenza A(H3N2) forecasting. B. Per residue perplexity on the heldout test dataset, for the baseline pHMM and two H-MuE models; lower numbers indicate better performance. C. Magnitude of the shift in amino acid preferences over time inferred by the RegressMuE, ν_l (Equation S80), projected onto an HA1 structure (PDB:4O5N). The full hemagglutinin protein is shown in a smaller size on the left. Region 1 is a portion of the stalk with a small shift over time, while region 2 is the 190 helix, a crucial antigenic region with a particularly large shift over time (Figures S8 and Figure S9). Comparisons with relative solvent accessibility and reproductive fitness measurements are given in Figures S10 and S11. D. Segments of sequences sampled from the posterior predictive distribution for the year 2024. The alignment variable is fixed based on the reference structure (PDB:4O5N), such that segments 1 and 2 correspond to the annotated structural features in C, and the column numbering is standard for influenza A(H3N2) (Section S10).

We considered three different models with increasing complexity. First, the pHMM describes the observed sequences as samples from a population with fixed amino acid frequencies at each site (Section S5.1). The pHMM can capture the observation that there exist key highly variable sites in the HA1 protein, the underlying motivation behind previous prediction methods such as [23]. Next we incorporated sequence collection time as a covariate,

using the RegressMuE (Section S5.2). This model, unlike the pHMM, takes into account the possibility that amino acid frequencies may shift over time. Finally, to capture more complex nonlinear dependencies between sequences and time, we extended a neural network regression model with a MuE distribution (NeuralMuE) (Section S5.4). The pHMM achieves a low per residue perplexity of 1.32 but the RegressMuE improves this to 1.24 (log Bayes factor $> 10^3$) (Figure 4B). This per residue perplexity difference corresponds to a factor of 10^{10} improvement in per sequence perplexity, a substantial reduction in the space of future viral sequences that must be considered. The NeuralMuE has similar predictive performance to the RegressMuE, with a per residue perplexity of 1.26.

Features

Given the success of the RegressMuE in predicting sequences, we investigated in detail what the model can tell us about how HA1 proteins have changed over time. We plotted the magnitude of the shift in amino acid preference from 1968 to 2019 inferred by the model, ν_l, for each residue position l, with the latent MuE alignment variable kept fixed (Equation S80). We found that sites with 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, Mann-Whitney U test $p < 1e-18$) (Figure S9) [25, 27]. 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, Mann-Whitney U test $p < 1e-4$) (Section S10) [28].

Generation

H-MuE 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 consistent with previously observed sequences (Figure 4D) (Section S10).

Latent space

In stark contrast to the TCR dataset, the latent space representation of the influenza HA1 dataset learned by the FactorMuE model shows the data falling roughly along a line (Figure 5A) (Section S10). 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 $\rho = 0.94$) [29]. This observation is consistent with the empirical success of the RegressMuE in sequence prediction, since the RegressMuE model is equivalent to a FactorMuE model with an observed latent representation (Section S5.3). Two clusters of outliers violate the proportionality rule (Figure 5B). The first cluster (marked by ‡) 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

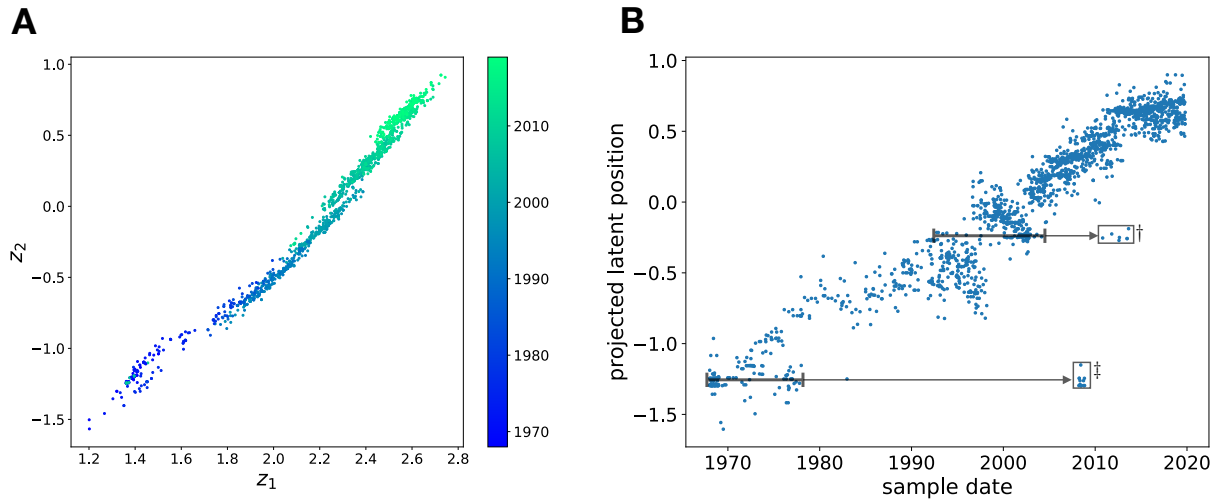


Figure 5: H-MuE model's latent space representation of influenza A(H3N2) evolution. A. The two-dimensional latent space of the FactorMuE model applied to the A(H3N2) influenza dataset (Section S10). Sequences are colored by the time at which the samples were collected. B. Y-axis: orthogonal projection 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 († and ‡), along with the time period at which sequences with similar latent representations were last seen (brackets).

dataset. This observation illustrates how latent representations can be used to clean mis-annotated sequence data. 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 [30, 31]. 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.

3 Discussion

H-MuE models are related to previous and contemporaneous work on hierarchical HMM models, such as methods that combine neural networks with HMMs [32] and Potts models with pHMMs [33]. Our work goes further by (1) offering a unified and comprehensive framework for HMM-based probabilistic alignment methods, (2) using probabilistic alignment models as general-purpose emission distributions (and showing that the conventional categorical emission distribution represents a special case), and (3) providing general and scalable approximate Bayesian inference algorithms. To assist in the creation of new models

and methods, we have made available an implementation[1] of the MuE distribution within the probabilistic programming language Edward2, enabling rapid development and testing of new H-MuE models [34].

The MuE distribution is a widely applicable tool for building generative and predictive probabilistic models of biological sequences. Using the MuE distribution, we can extend arbitrary continuous-space vector models to become H-MuE models of unaligned biological sequences, while accounting for mutational variability and statistical uncertainty. Since H-MuE models avoid the pathologies of MSA-based methods, we can apply H-MuE models in a wide variety of statistical contexts, including causal inference, semi-supervised learning, decision-making and design problems. In this article we have explored common vector models (linear regression, probabilistic PCA, a regression neural network and a neural network latent variable model) and focused on prediction and forecasting problems. We anticipate that a wide variety of other probabilistic models and methods – such as nonlinear time series models, sparse regression models, and independent component analysis models – could have as great an impact on biological sequence statistics as they have had on other fields.

Acknowledgments

We wish to thank Chris Sander, John Ingraham, Elizabeth Wood, 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

[1] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *J. R. Stat. Soc. Series B Stat. Methodol.*, 61(3):611–622, August 1999.

[2] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[3] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, May 1995.

[4] Emma Pierson and Christopher Yau. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.*, 16:241, November 2015.

[5] Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat. Commun.*, 9(1):284, January 2018.

[6] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053–1058, December 2018.

[1] Will be made available during peer review at https://github.com/debbiemarkslab.

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

[8] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *J. Mach. Learn. Res.*, 18(14):1–45, January 2017.

[9] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.*, 18(153):1–43, April 2018.

[10] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *J. Am. Stat. Assoc.*, 112(518):859–877, April 2017.

[11] David Duvenaud and Ryan P Adams. Black-box stochastic variational inference in five lines of python. In *NIPS Workshop on Black-box Learning and Inference*, 2015.

[12] Diederik P Kingma and Max Welling. Auto-Encoding variational bayes. December 2013.

[13] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. January 2014.

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

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

[16] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[17] Thomas A Hopf, John B Ingraham, Frank J Poelwijk, Charlotta P I Schärfe, Michael Springer, Chris Sander, and Debora S Marks. Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.*, 35(2):128–135, February 2017.

[18] Agnes Toth-Petroczy, Perry Palmedo, John Ingraham, Thomas A Hopf, Bonnie Berger, Chris Sander, and Debora S Marks. Structured states of disordered proteins from genomic sequences. *Cell*, 167(1):158–170.e12, September 2016.

[19] Elvira Regina Tamarozzi and Silvana Giuliatti. 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.

[20] A Danielle Iuliano, Katherine M Roguski, Howard H Chang, David J Muscatello, Rakhee Palekar, Stefano Tempia, Cheryl Cohen, Jon Michael Gran, Dena Schanzer, Benjamin J Cowling, Peng Wu, Jan Kyncl, Li Wei Ang, Minah Park, Monika Redlberger-Fritz, Hongjie Yu, Laura Espenhain, Anand Krishnan, Gideon Emukule, Liselotte van Asten, Susana Pereira da Silva, Suchunya Aungkulanon, Udo Buchholz, Marc-Alain Widdowson, Joseph S Bresee, 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.

[21] Marta Luksza and Michael Lässig. A predictive fitness model for influenza. *Nature*, 507(7490):57–61, March 2014.

[22] Nick S Laursen and Ian A Wilson. Broadly neutralizing antibodies against influenza viruses. *Antiviral Res.*, 98(3):476–483, June 2013.

[23] Robin M Bush, Catherine A Bender, Kanta Subbarao, Nancy J Cox, and Walter M Fitch. Predicting the evolution of human influenza A. *Science*, 286(5446):1921–1925, December 1999.

[24] Richard A Neher, Trevor Bedford, Rodney S Daniels, Colin A Russell, and Boris I Shraiman. Prediction, dynamics, and visualization of antigenic phenotypes of seasonal influenza viruses. *Proc. Natl. Acad. Sci. U. S. A.*, 113(12):E1701–9, March 2016.

[25] 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.

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

[27] Enrique T Muñoz and Michael W Deem. Epitope analysis for influenza vaccine design. *Vaccine*, 23(9):1144–1148, January 2005.

[28] Juhye M Lee, Rachel Eguia, Seth J Zost, Saket Choudhary, Patrick C Wilson, Trevor Bedford, Terry Stevens-Ayers, Michael Boeckh, Aeron C Hurt, Seema S Lakdawala, Scott E Hensley, and Jesse D Bloom. Mapping person-to-person variation in viral mutations that escape polyclonal serum targeting influenza hemagglutinin. *Elife*, 8, August 2019.

[29] John Novembre and Matthew Stephens. Interpreting principal component analyses of spatial population genetic variation. *Nat. Genet.*, 40(5):646–649, May 2008.

[30] Michael A Jhung, Scott Epperson, Matthew Biggerstaff, Donna Allen, Amanda Balish, Nathelia Barnes, Amanda Beaudoin, Lashondra Berman, Sally Bidol, Lenee Blanton, David Blythe, Lynnette Brammer, Tiffany D'Mello, Richard Danila, William Davis, Sietske de Fijter, Mary Diorio, Lizette O Durand, Shannon Emery, Brian Fowler, Rebecca Garten, Yoran Grant, Adena Greenbaum, Larisa Gubareva, Fiona Havers, Thomas Haupt, Jennifer House, Sherif Ibrahim, Victoria Jiang, Seema Jain, Daniel

Jernigan, James Kazmierczak, Alexander Klimov, Stephen Lindstrom, Allison Longenberger, Paul Lucas, Ruth Lynfield, Meredith McMorrow, Maria Moll, Craig Morin, Stephen Ostroff, Shannon L Page, Sarah Y Park, Susan Peters, Celia Quinn, Carrie Reed, Shawn Richards, Joni Scheftel, Owen Simwale, Bo Shu, Kenneth Soyemi, Jill Stauffer, Craig Steffens, Su Su, Lauren Torso, Timothy M Uyeki, Sara Vetter, Julie Villanueva, Karen K Wong, Michael Shaw, Joseph S Bresee, Nancy Cox, and Lyn Finelli. Outbreak of variant influenza A(H3N2) virus in the united states. *Clin. Infect. Dis.*, 57(12):1703–1712, December 2013.

[31] Danuta M Skowronski, Naveed Z Janjua, Gaston De Serres, Dale Purych, Vladimir Gilca, David W Scheifele, Marc Dionne, Suzana Sabaiduc, Jennifer L Gardy, Guiyun Li, Nathalie Bastien, Martin Petric, Guy Boivin, and Yan Li. 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.

[32] Anders Krogh and Soren Kamaric Riis. Hidden neural networks. *Neural Comput.*, 11:541–563, 1999.

[33] Grey W Wilburn and Sean R Eddy. Remote homology search with hidden potts models. June 2020.

[34] Dustin Tran, Matthew Hoffman, Dave Moore, Christopher Suter, Srinivas Vasudevan, Alexey Radul, Matthew Johnson, and Rif A Saurous. Simple, distributed, and accelerated probabilistic programming. November 2018.

[35] 10x Genomics. CD8+ T cells isolated from PBMCs of a healthy donor - direct TCR enrichment, August 2018.

[36] Ji-Li Chen, Guillaume Stewart-Jones, Giovanna Bossi, Nikolai M Lissin, Linda Wooldridge, Ed Man Lik Choi, Gerhard Held, P Rod Dunbar, Robert M Esnouf, Malkit Sami, Jonathan M Boulter, Pierre Rizkallah, Christoph Renner, Andrew Sewell, P Anton van der Merwe, Bent K Jakobsen, Gillian Griffiths, E Yvonne Jones, and Vincenzo Cerundolo. Structural and kinetic basis for heightened immunogenicity of T cell vaccines. *J. Exp. Med.*, 201(8):1243–1255, April 2005.

[37] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, December 1983.

[38] Matthew Z Tien, Austin G Meyer, Dariya K Sydykova, Stephanie J Spielman, and Claus O Wilke. Maximum allowed solvent accessibilites of residues in proteins. *PLoS One*, 8(11):e80635, November 2013.

[39] Peter J A Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J L de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, June 2009.

[40] Jian Ye, Ning Ma, Thomas L Madden, and James M Ostell. IgBLAST: an immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Res.*, 41(Web Server issue):W34–40, July 2013.

[41] Paul F Robbins, Yong F Li, Mona El-Gamil, Yangbing Zhao, Jennifer A Wargo, Zhili Zheng, Hui Xu, Richard A Morgan, Steven A Feldman, Laura A Johnson, Alan D Bennett, Steven M Dunn, Tara M Mahon, Bent K Jakobsen, and Steven 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.

[42] Juhye M Lee, John Huddleston, Michael B Doud, Kathryn A Hooper, Nicholas C Wu, Trevor Bedford, and Jesse 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, August 2018.

[43] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3D structure computed from evolutionary sequence variation. *PLoS One*, 6(12):e28766, December 2011.

[44] Andrew Gelman, Xiao-Li Meng, and Hal Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, 6(4):733–760, 1996.

[45] A Philip Dawid. 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.

[46] Miguel A Hernán and James M Robins. *Causal inference: what if*. Chapman & Hill/CRC, 2020.

[47] William H Greene. *Econometrics*. Prentice Hall, 2016.

[48] Saul B Needleman and Christian 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.

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

[50] Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. December 2013.

[51] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. November 2015.

[52] Alexander A Alemi, Ben Poole, Ian Fischer, Joshua V Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken ELBO. November 2017.

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

[54] Elisabeth Gasteiger, Christine Hoogland, Alexandre Gattiker, Severine Duvaud, Marc R Wilkins, Ron D Appel, and Amos Bairoch. Protein identification and analysis tools on the ExPASy server. In John M Walker, editor, *The Proteomics Protocols Handbook*, pages 571–607. Humana Press, Totowa, NJ, 2005.

[55] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U. S. A.*, 89(22):10915–10919, November 1992.

[56] 1000 Genomes Project Consortium, Adam Auton, Lisa D Brooks, Richard M Durbin, Erik P Garrison, Hyun Min Kang, Jan O Korbel, Jonathan L Marchini, Shane McCarthy, Gil A McVean, and Gonçalo R Abecasis. A global reference for human genetic variation. *Nature*, 526(7571):68–74, October 2015.

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

[58] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, March 2015.

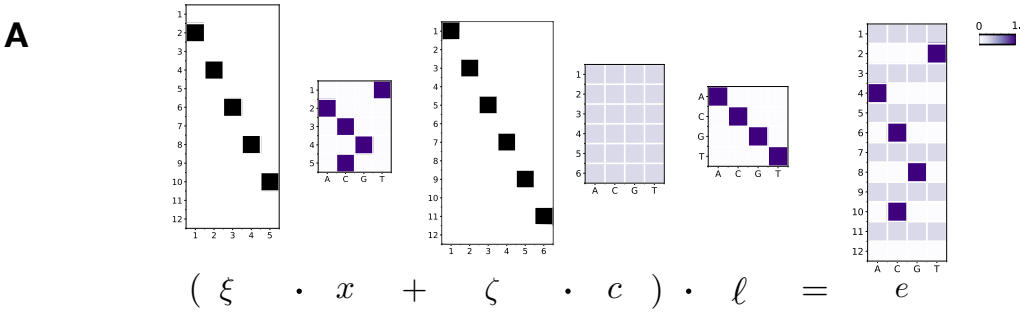
[59] Thomas A Hopf, Anna G Green, Benjamin Schubert, Sophia Mersmann, Charlotta P I Schärfe, John B Ingraham, Agnes Toth-Petroczy, Kelly Brock, Adam J Riesselman, Perry Palmedo, Chan Kang, Robert Sheridan, Eli J Draizen, Christian Dallago, Chris Sander, and Debora S Marks. The EVcouplings python framework for coevolutionary sequence analysis. *Bioinformatics*, 35(9):1582–1584, May 2019.

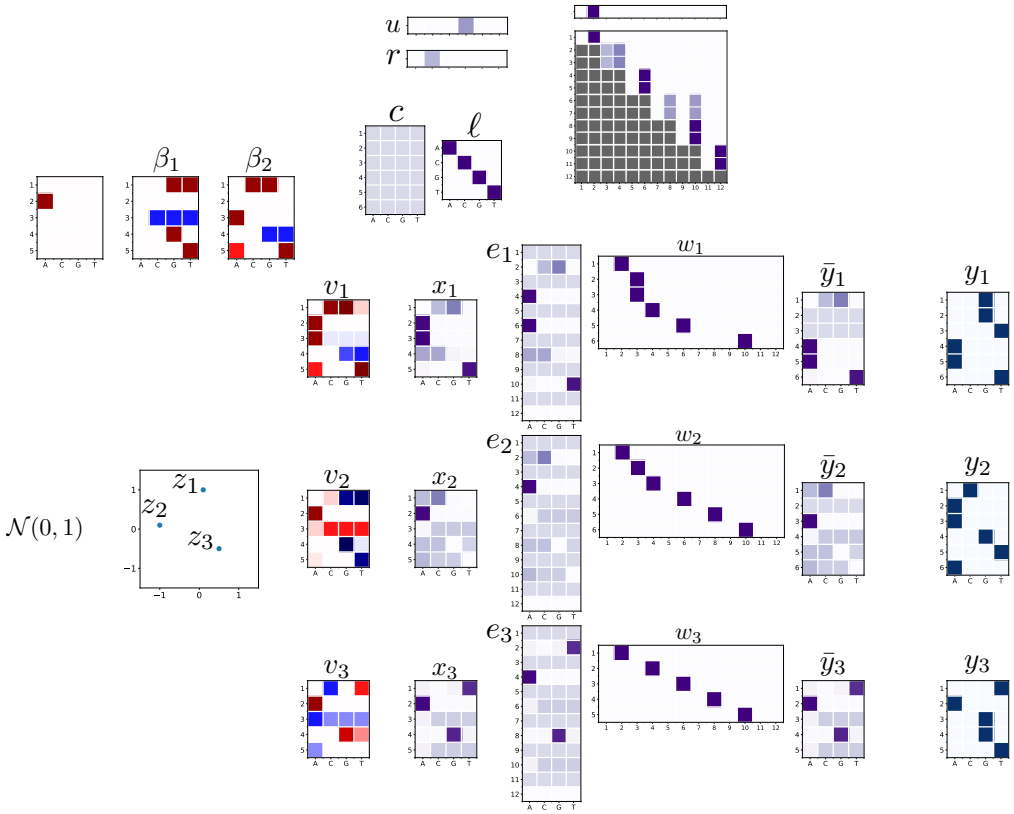
[60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. December 2014.

[61] David F Burke and Derek J Smith. A recommended numbering scheme for influenza A HA subtypes. *PLoS One*, 9(11):e112302, November 2014.

Supplementary material

S1 Supplementary Figures





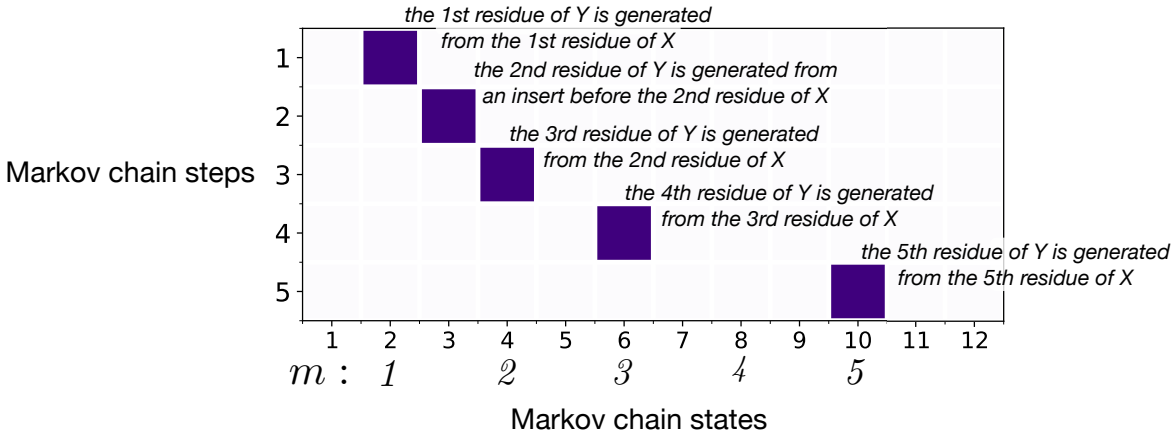


Figure S3: Interpreting the hidden state variable w. This plot shows a one-hot encoding of an example w; dark purple squares are entries set to 1, and the rest of the matrix is 0. In this example there are $M = 5$ ancestral residues, $K = 12$ states and $L = 5$ residues in y. Even-numbered states – that is, match states, with $k\%2 = 0$ – are marked below, with their state number $m = k/2$. The italic text describes how to interpret each row of the w matrix in terms of an alignment. A rigorous mapping from w to an alignment is defined in Section S4.1.

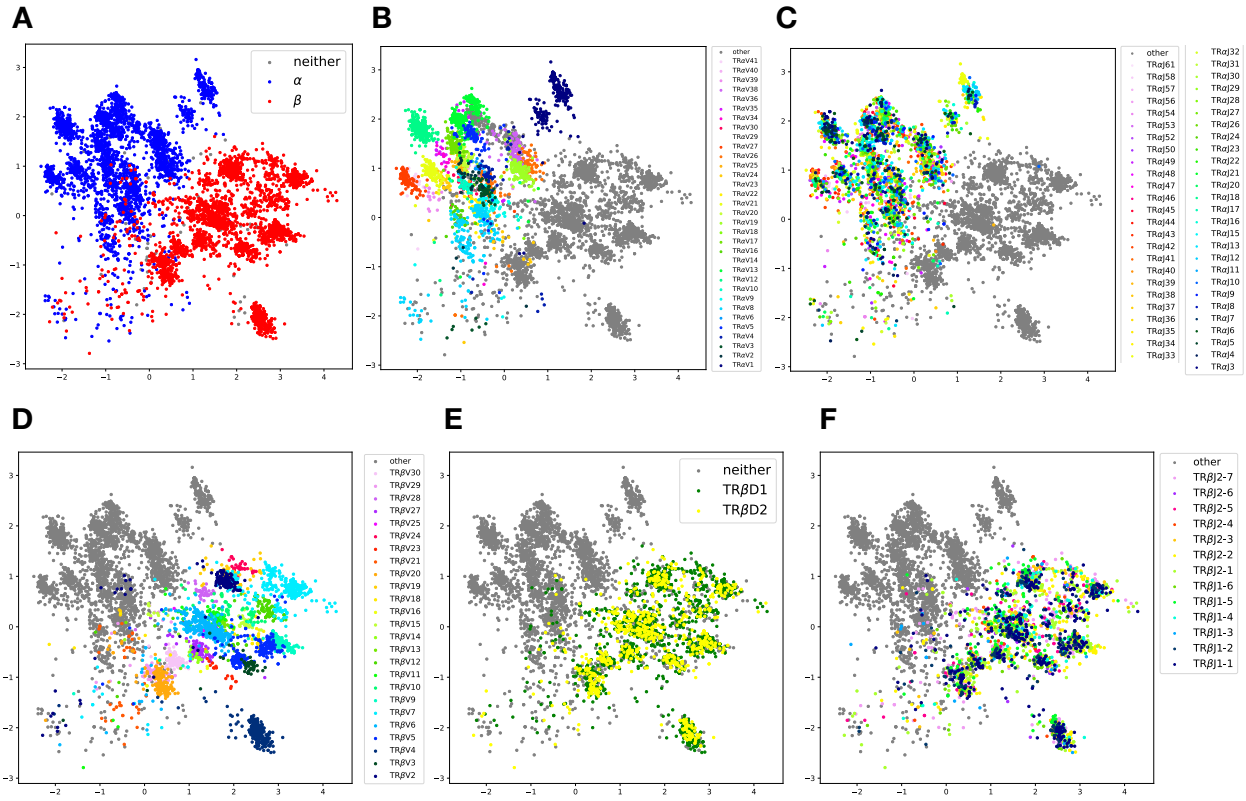


Figure S4: Latent space representation of human T-cell receptor sequences, colored by supervised annotations. Annotations from [35]. A. TCRα versus TCRβ. B. α chain V types. C. α chain J types. D. β chain V types. E. β chain D types. F. β chain J types and subtypes.

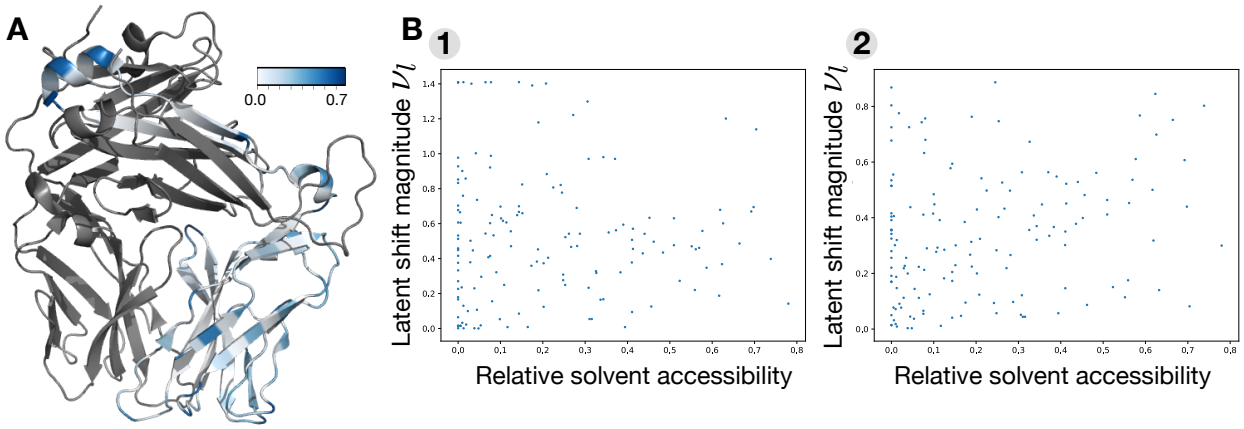


Figure S5: Comparing H-MuE model features to T-cell receptor relative solvent accessibility. A. Relative solvent accessibility of TCRβ from the structure PDB:2BNR [36] (the TCRα chain is shown in grey), computed using DSSP [37] and the maximum values in [38] with the Biopython API [39]. B. Residue relative solvent accessibility versus Factor-MuE shift magnitude ν_l (Equation S78) along vector 1 and vector 2 from Figure 3E. The correlation between the shift along vector 1 and the accessibility is Spearman $\rho = 0.039$, $p = 0.64$.

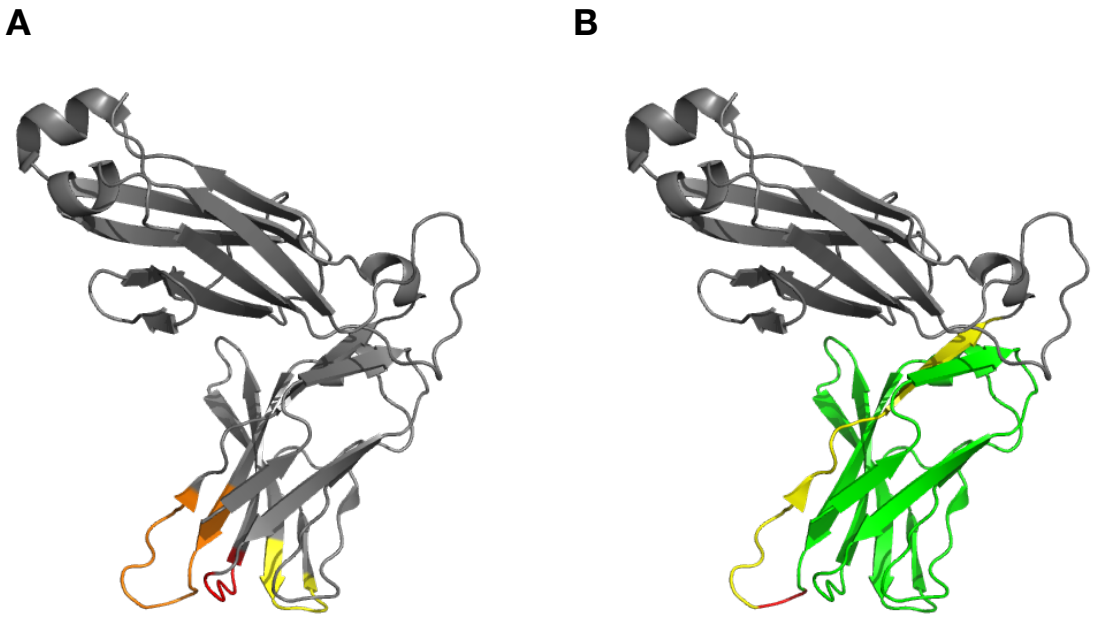


Figure S6: T-cell receptor structural annotations. A. CDR segments of PDB:2BNR chain E [36], based on IgBLAST annotations [40] of the nucleotide sequence of 1G4 TCRβ obtained from [41], and translated from nucleotides into the corresponding positions in the amino acid sequence. CDR1 in red, CDR2 in yellow 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.

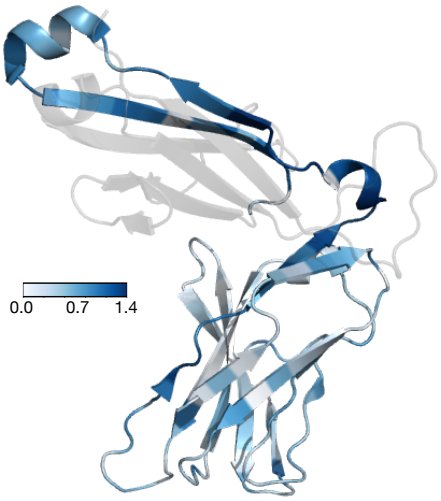


Figure S7: Shift ν_l from chain α to chain β sequences learned by the RegressMuE model. ν_l was computed as in Equation S78, using the chain index in place of the latent variable z.

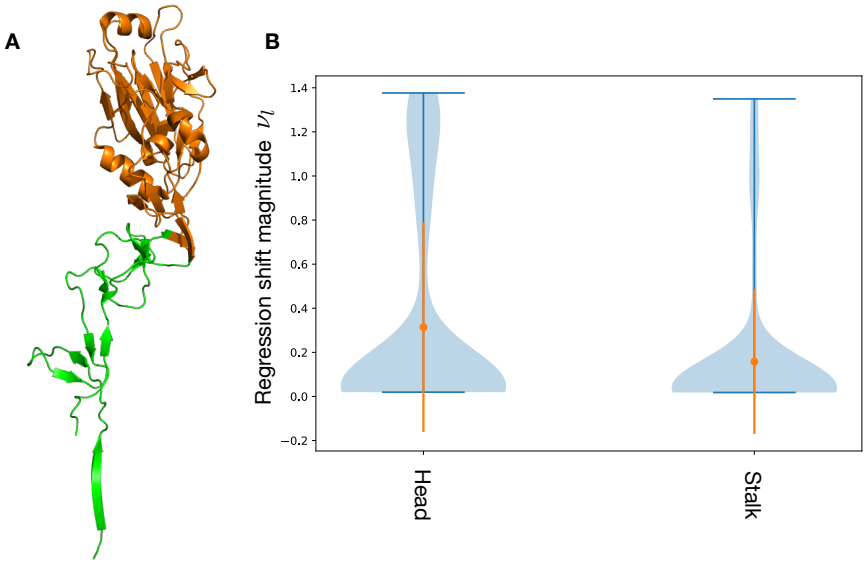


Figure S8: Comparing H-MuE model regression coefficients to HA1 structural domains. A. Head (orange) and stalk (green) domains of the HA1 protein (PDB:4O5N); residues between sites 52 and 277 are defined as the head domain, and all others as stalk, following [42]. B. Violin plots of regression shift ν_l (Equation S80) for residues in the head domain (226 residues) versus the stalk domain (103 residues). Mean and standard deviation are shown in orange.

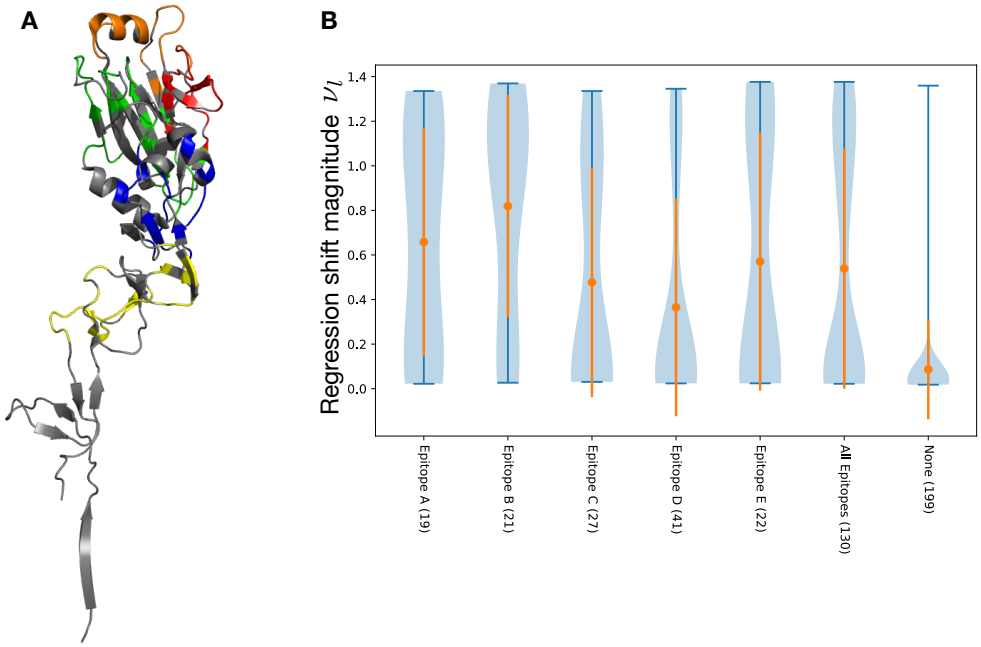


Figure S9: Comparing H-MuE model regression coefficients to HA1 epitope regions. A. Epitope regions A (red), B (orange), C (yellow), D (green), E (blue) [25, 27]. B. Violin plots of regression shift ν_l (Equation S80) for residues in each epitope region, for all epitope regions together, and for residues not in any epitope region; the number of residues in each region is shown in parenthesis. Mean and standard deviation are shown in orange.

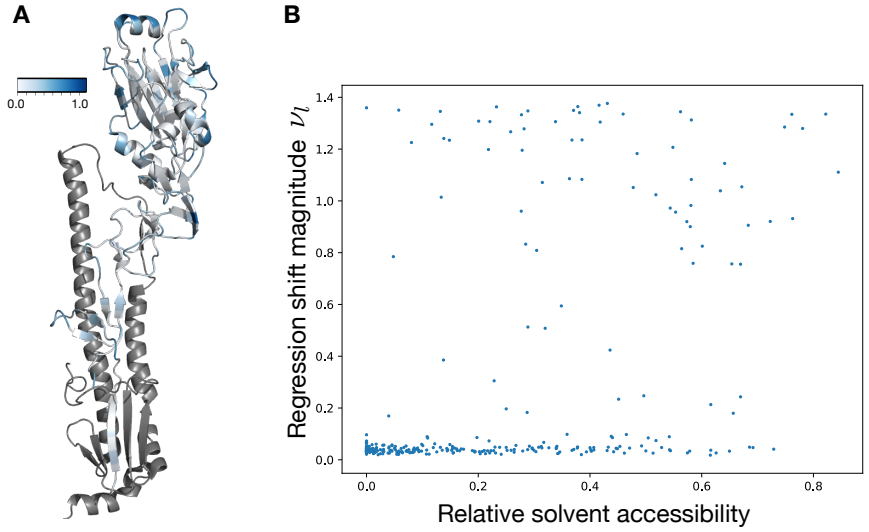


Figure S10: Comparing H-MuE model regression coefficients to HA1 relative solvent accessibility. A. Relative solvent accessibility of the HA1 protein (PDB:4O5N), computed using DSSP [37] and the maximum values in [38] with the Biopython API [39]. HA2 protein shown in grey. B. Relative solvent accessibility versus regression shift magnitude ν_l (Equation S80), residue-by-residue. Spearman $\rho = 0.41$, $p < 1e-13$.

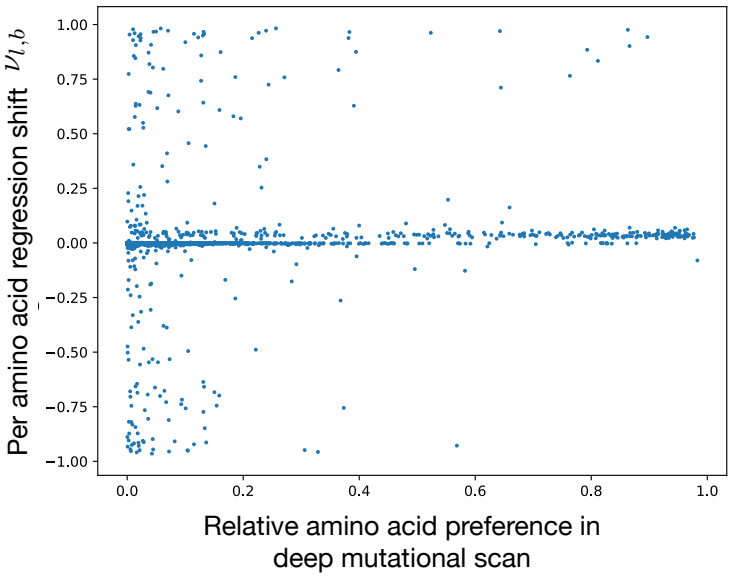


Figure S11:
Comparing H-MuE model regression coefficients to a deep mutational scan of HA. X-axis: regression shift for each amino acid at each position from 1968 to 2019,

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

(terms defined as in Equation S80). Y-axis: relative preference for point mutants with amino acid b at position l in the deep mutational scan performed in Lee et al. [42]. Spearman $\rho = 0.08$, $p < 1e-11$.

S2 Statistical Pathologies in MSA-based Methods

In this section we describe the theoretical problems that arise when using MSA-based methods for predicting biological sequences. The underlying issue is that the multiple sequence alignment of the entire dataset may change as more data is added, since new sequences may contain regions that do not match old sequences, and since new sequences provide evidence for and against similarity among old sequences. The fact that the shape of the data matrix may change as more data is added and the fact that previously observed data may be altered as more data is added make random variables and predictions ill-defined.

More formally, let $Y^{(N)}_{\text{MSA}}$ be the alignment of a dataset of N sequences, with $Y^{(N)}_{\text{MSA},i}$ corresponding to the alignment of the ith sequence. Let $\mathcal{B}$ be the symbol alphabet including the gap symbol, e.g. for DNA $\mathcal{B} = \{A, T, G, C, -\}$. Let $J^{(N)}$ be the width of the alignment. Each row of the alignment is a vector of symbols $Y^{(N)}_{\text{MSA},i} \in \mathcal{B}^{J^{(N)}}$. The standard method of building probabilistic models of pre-aligned sequence data is to assume that aligned sequences $Y^{(N)}_{\text{MSA},1}, ..., Y^{(N)}_{\text{MSA},N}$ are independently and identically distributed (iid) according to some underlying distribution $p(Y|\theta)$ for $Y \in \mathcal{B}^{J^{(N)}}$, such as a Potts model or neural network [14, 17, 43]. Mathematically,

$$Y^{(N)}_{\text{MSA},i} \overset{iid}{\sim} p(Y|\theta). \tag{S1}$$

While models of this form are powerful when used for parameter estimation (that is, inferring

θ), they are ill-defined when used for prediction of unobserved or future sequences.

The posterior predictive distribution $p(Y|Y^{(N)}_{\text{MSA},1}, ..., Y^{(N)}_{\text{MSA},N})$ is defined for $Y \in \mathcal{B}^{J^{(N)}}$. However, when we observe a new sequence and add it to our multiple sequence alignment, we have in general $J^{(N+1)} \neq J^{(N)}$ and consequently $Y^{(N+1)}_{\text{MSA},N+1} \notin \mathcal{B}^{J^{(N)}}$. Thus we find that

$$p(Y^{(N+1)}_{\text{MSA},N+1}|Y^{(N)}_{\text{MSA},1}, ..., Y^{(N)}_{\text{MSA},N}) \tag{S2}$$

is ill-defined. Future datapoints will not necessarily live in the same mathematical space as the model.

Even if we could somehow safely assume that $J^{(N+1)} = J^{(N)}$ we would still run into problems, since the alignment can change with additional data. Mathematically, in general $Y^{(N)}_{\text{MSA},i} \neq Y^{(N+1)}_{\text{MSA},i}$ for $i \in \{1, ..., N\}$. Then,

$$p(Y^{(N+1)}_{\text{MSA},N+1}|Y^{(N)}_{\text{MSA},1}, ..., Y^{(N)}_{\text{MSA},N}) \neq p(Y^{(N+1)}_{\text{MSA},N+1}|Y^{(N+1)}_{\text{MSA},1}, ..., Y^{(N+1)}_{\text{MSA},N}). \tag{S3}$$

In other words, making predictions of unobserved data (the left hand side of the equation), is not the same as holding out and predicting pre-aligned data (the right hand side of the equation).

Ultimately, both of these problems emerge because pre-aligned sequence data automatically violate the iid assumption of the model in Equation S1. The aligned sequence $Y^{(N)}_{\text{MSA},i}$ is not just a function of the unaligned sequence y_i, it is also a function of all the other unaligned sequences $y_{i' \neq i}$ in the dataset. In theory, we could avoid prediction problems by specifying a joint model of the entire dataset, treating the entire MSA as a stochastic object that changes with an increasing number of sequences N. That is, we could define a model $p(Y^{(N)}_{\text{MSA},1}, ..., Y^{(N)}_{\text{MSA},N}|\theta, N)$, where in general $p(Y^{(N)}_{\text{MSA},1}, ..., Y^{(N)}_{\text{MSA},N}|\theta, N) \neq p(Y^{(N+1)}_{\text{MSA},1}, ..., Y^{(N+1)}_{\text{MSA},N}|\theta, N+1)$. While such models are possible in theory, in practice they are challenging to write down, hard to perform inference on, and lack the asymptotic convergence guarantees associated with iid models. H-MuE models offer a principled alternative that preserves the iid assumption while making sequence prediction well-defined.

Note also that prediction is a core concern across all of statistics, not just in forecasting applications. Many model evaluation methods, such as cross validation and posterior predictive checks, depend on the ability of statistical models to predict unobserved data [44]. Prequential statistical approaches depend entirely on prediction [45]. Causal inference methods are founded in prediction [46]. In general, relying on multiple sequence alignments makes application of these powerful ideas ill-defined. H-MuE models make application possible.

S3 MuE Distribution Details

S3.1 Termination

The state variable w in the MuE distribution is drawn from a Markov chain. We consider two possible methods for terminating the chain. The first method is to have an explicit termination state: when the Markov chain reaches the final state $k = K$, it halts without emitting an observation. This method is used in the previously proposed probabilistic alignment models discussed in Section S4. The second method is to sample the length of the chain

independently of the states of the chain. We use this approach in our H-MuE models, since it makes inferring an accurate value for M unnecessary, and we can instead simply choose a large M value compared to the typical length of sequences in the dataset (Section S8).

S3.2 Special Cases

In this section we describe two important special cases of the MuE distribution. First, consider the case where $a_k^{(i)} = \delta_{k,2}$ and $a_{k,k'}^{(t)} = \delta_{k+2,k'}$, where δ is the Kronecker delta. If, in addition, $L = M$ (which will be guaranteed if we are working with an explicit termination state $k = K$), then the MuE distribution simplifies to

$$y \sim \text{Categorical}(x \cdot \ell). \tag{S4}$$

We refer to this as the "no-indel case" of the MuE distribution: the generated y sequences just have substitution mutations relative to x.

If, in addition, we set $D = B$ and $\ell = I_B$, where I_B is the $B \times B$ identity matrix, then $y \sim \text{Categorical}(x)$ under the MuE distribution, and the full H-MuE model becomes

$$\begin{aligned} v_i &\sim p(v|\theta) \\ y_i &\sim \text{Categorical}(x_i = \text{softmax}(v_i)). \end{aligned} \tag{S5}$$

We thus recover the standard multivariate categorical emission distribution, used in e.g. multinomial logit regression models [47] and the state-of-the-art pre-aligned sequence model proposed by Riesselman et al. [14]. We refer to this as the "no-mutation case" of the MuE distribution.

S4 Theory

In this section we describe our theoretical results. First we rigorously establish a mapping between the latent variable w and a pairwise alignment between x and y, and show that the restrictions on the transition matrix $a^{(t)}$ are necessary and sufficient for this mapping to exist with probability one (Section S4.1). We then show that the MuE distribution provides a unified framework for understanding a wide variety of classical biological sequence analysis methods: we can recover, as special case models and estimators, stochastic process models of sequence evolution (the Thorne-Kishino-Felsenstein model, Section S4.2), probabilistic multiple sequence alignment methods (the profile HMM, Section S4.4), probabilistic pairwise sequence alignment methods (the pair HMM, Section S4.3), and non-probabilistic pairwise sequence alignment algorithms (the Needleman-Wunsch algorithm, Section S4.5). We also compare the MuE distribution to a natural language translation model (Section S4.6), which fits the basic form of the MuE distribution but violates the restrictions on $a^{(t)}$.

S4.1 Alignments and States

For convenience, we assume throughout Section S4.1 that the Markov chain terminates at the state $k = K = 2M + 2$, and that it reaches the termination state eventually with probability one. A precisely analogous analysis can be done for the case where there is no termination state and instead L, the length of sequence y, is drawn independently of the Markov chain.

S4.1.1 Definitions

In this section we set up key definitions, illustrated with the example in Equation S6, S7 and S8.

$$\begin{aligned} \tilde{x} &: \texttt{T C T G \$} \\ \tilde{y} &: \texttt{A T G \$} \end{aligned} \tag{S6}$$

$$\begin{aligned} A^x &: \texttt{- T C T - G \$} \\ A^y &: \texttt{A - - T G - \$} \\ \omega^x &: \texttt{0 1 1 1 0 1 1} \\ \omega^y &: \texttt{1 0 0 1 1 0 1} \end{aligned} \tag{S7}$$

$$\begin{aligned} j^y_{1:(L+1)} &:(1,4,5,7) \\ g_{1:(L+1)} &:(1,0,1,0) \\ m_{1:(L+1)} &:(1,3,4,5) \\ k_{1:(L+1)} &:(1,6,7,10) \end{aligned} \tag{S8}$$

Let $\tilde{y} = (\tilde{y}_1, ..., \tilde{y}_L, \$)$ and $\tilde{x} = (\tilde{x}_1, ..., \tilde{x}_M, \$)$ be sequences, with \$ the termination symbol. A pairwise alignment $\mathcal{A}$ of $\tilde{x}$ and $\tilde{y}$ with J columns is defined by the tuple (A^x, A^y) where A^x is a vector of length J consisting of the residues of $\tilde{x}$, in order, interspersed with gap symbols, and similarly for A^y, with the condition that every column of $\mathcal{A}$ contains either a letter from $\tilde{x}$, a letter from $\tilde{y}$, or both, and the final column must contain the termination symbol from both sequences. We can represent the alignment in terms of the index vectors $\omega^x, \omega^y \in \{0,1\}^J$, where $\omega^x_j = 1$ if there is a letter of $\tilde{x}$ at A^x_j and $\omega^x_j = 0$ if there is a gap, and likewise for ω^y and A^y. If

1. $\sum_j \omega^x_j = M + 1$. (The sequence $\tilde{x}$ has M residues plus the termination symbol.)
2. $\sum_j \omega^y_j = L + 1$. (The sequence $\tilde{y}$ has L residues plus the termination symbol.)
3. $\sum_{j=1}^{J} (1 - \omega^x_j)(1 - \omega^y_j) = 0$. (Each column of the alignment has at least one residue; there cannot be two gap symbols aligned.)
4. $\omega^x_J = \omega^y_J = 1$. (The final column contains the terminal symbol of each sequence.)

then the tuple (ω^x, ω^y) uniquely defines an alignment of the sequences $\tilde{x}$ and $\tilde{y}$. Let j^y_l, for $l \in \{1, ..., L+1\}$, index the column of the alignment that the lth residue of $\tilde{y}$ falls in. Mathematically,

$$j^y_l := \sum_{j=1}^{J} j\, \omega^y_j \delta_{\sum_{j'=1}^{j} \omega^y_j, l} \tag{S9}$$

where δ is the Kronecker delta. Let g_l indicate whether the lth residue of $\tilde{y}$ is aligned to a gap in $\tilde{x}$ or not. Mathematically,

$$g_l := 1 - \omega^x_{j^y_l}. \tag{S10}$$

Let m_l be the residue of $\tilde{x}$ aligned to the lth residue of $\tilde{y}$; if the lth residue of $\tilde{y}$ is aligned to a gap in $\tilde{x}$, let m_l be the closest residue in $\tilde{x}$ to the right. Mathematically,

$$m_l := j_l^y - \sum_{l'=1}^{l-1} g_{l'}. \tag{S11}$$

Finally, define

$$k_l := 2m_l - g_l. \tag{S12}$$

S4.1.2 From Alignments to States

Starting from any pairwise alignment $\mathcal{A}$ we can compute k_l for $l \in \{1, ..., L+1\}$. We can use these k_l to define a sequence of states w for the Markov chain in a MuE distribution, by setting $w_{l,k} = \delta_{k_l,k}$.

Alignments are typically intended to represent an evolutionary relationship between sequence $\tilde{x}$ and $\tilde{y}$. If residue l of $\tilde{y}$ is aligned to residue m of $\tilde{x}$, it suggests that they share common descent, and should therefore be similar. In particular, we can expect $\tilde{y}_l$ to either match $\tilde{x}_m$ exactly or be related via a substitution mutation. The states w defined from the alignment $\mathcal{A}$ reflect this idea. If the lth residue of $\tilde{y}$ is aligned to the mth residue of $\tilde{x}$, then $m_l = m$ and $g_l = 1 - \omega^x_{j_l^y} = 0$. The MuE distribution gives

$$y_l \sim \text{Categorical}(\sum_{k,m,d} \delta_{k_l,k}(\delta_{k,2m} x_{m,d} + \delta_{k,2m-1} c_{m,d})\ell_d) = \text{Categorical}(x_{m_l} \cdot \ell). \tag{S13}$$

Thus, y_l is generated from x_{m_l} according to the substitution probabilities ℓ. On the other hand, if the lth residue of $\tilde{y}$ is aligned to a gap, $\tilde{y}_l$ should be independent of $\tilde{x}$, since it comes from an insertion mutation. In this case $g_l = 1 - \omega^x_{j_l^y} = 1$ and we find

$$y_l \sim \text{Categorical}(\sum_{k,m,d} \delta_{k_l,k}(\delta_{k,2m} x_{m,d} + \delta_{k,2m-1} c_{m,d})\ell_d) = \text{Categorical}(c_{m_l} \cdot \ell). \tag{S14}$$

Thus, y_l does not depend on x and is instead determined by the insertion parameter c.

S4.1.3 From States to Alignments

Starting from a sequence of states w drawn from MarkovModel$(a^{(i)}, a^{(t)})$, we can convert uniquely back to an alignment $\mathcal{A}$ between $\tilde{y}$ and $\tilde{x}$. We assume w_{L+1} is in the termination state, ie. $w_{L+1,K} = 1$ (note that this assumption requires that the Markov chain reaches the termination state with probability one). Let $k_l = \arg\max_k w_{l,k}$ be the state of the Markov model at position l. We can now invert Equations S9-S12 to reconstruct the corresponding alignment. We have

$$\begin{aligned} g_l &= k_l \% 2 \\ m_l &= (k_l + g_l)/2 \\ j_l^y &= m_l + \sum_{l'=1}^{l-1} g_{l'}. \end{aligned} \tag{S15}$$

where recall % is the modulo operation. (Note that for j^y, the list of columns of the alignment that the residues of $\tilde{y}$ fall in, to be valid based on its definition, it must be a strictly ascending list of integers; this is discussed further below.) We then obtain, for $l \in \{1, ..., L+1\}$,

$$\begin{aligned} \omega^y_{j^y_l} &= 1 \\ \omega^y_{j \notin j^y} &= 0 \\ \omega^x_{j^y_l} &= 1 - g_l \\ \omega^x_{j \notin j^y} &= 1 \end{aligned} \tag{S16}$$

where we have used the property that $\sum_{j=1}^{J}(1-\omega^x_j)(1-\omega^y_j) = 0$. ω^x and ω^y together uniquely define an alignment $\mathcal{A}$ of $\tilde{x}$ and $\tilde{y}$.

As mentioned above, for j^y to be valid based on its definition, it must be a strictly ascending list of integers. That is, we must have for all $l \in 2, ..., L+1$,

$$0 < j^y_l - j^y_{l-1} = m_l + g_{l-1} - m_{l-1}. \tag{S17}$$

Intuitively, this condition says that alignments cannot "double back": if residue m in $\tilde{x}$ is aligned to residue l in $\tilde{y}$, then a later residue $l' > l$ cannot align to an earlier residue $m' < m$ in y. With some more algebra, we find that for all $l \in 2, ..., L+1$ we must have

$$\begin{aligned} 0 &< 2(m_l + g_{l-1} - m_{l-1}) \\ &= 2\big((k_l + k_l\%2)/2 + k_{l-1}\%2 - (k_{l-1} + k_{l-1}\%2)/2\big) \\ &= k_l + k_l\%2 - k_{l-1} + k_{l-1}\%2. \end{aligned} \tag{S18}$$

For this condition to hold with probability one for any sample from the Markov chain it must be the case that the transition probability $a^{(t)}_{k,k'} = 0$ for $k' + k'\%2 - k + k\%2 \leq 0$, so long as state k is accessible by the Markov model (that is, so long as the Markov chain can eventually reach state k, starting from the initial state). The MuE distribution has this restriction, and therefore every sample from a MuE distribution corresponds to an alignment.

S4.1.4 Multiple Sequence Alignments

When we have multiple independent samples $y_1, ..., y_N$ from a MuE distribution or H-MuE model, the collection of associated state variables $w_1, ..., w_N$ can be interpreted as a multiple sequence alignment of $y_1, ..., y_N$. While there is some ambiguity here in the mapping between states and alignments (unlike in the pairwise alignment case), we present the mapping used with the profile HMM, relying on the re-derivation of the pHMM as a special case of the MuE distribution described in Section S4.4. The core idea is to interpret the positions of x as conserved columns. Let $k_{i,l} = \arg\max w_{i,l,k}$ be the state the Markov chain is in at position l in sequence i, with $m_{i,l}$ and $g_{i,l}$ defined following Equation S15. For each l, if $g_{i,l} = 0$, we place $y_{i,l}$ in the $m_{i,l}$th conserved column of the MSA. Otherwise, if $g_{i,l} = 1$, we place $y_{i,l}$ into the block of insertions immediately before the conserved column $m_{i,l}$. As described in detail in Durbin et al. [16] Chapter 6.5, the choice of how to represent the insertion blocks in the alignment is somewhat arbitrary. The standard approach is to pad the insertion block on the right with gap symbols to reach the length of the longest subsequence in state $k_{i,l}$ across the entire dataset. See Figure S2 for an example and Durbin et al. [16] Chapter 6.5 for further details.

S4.2 Thorne-Kishino-Felsenstein

The Thorne-Kishino-Felsenstein (TKF) model is a continuous-time stochastic process model of sequence evolution [15]. Let x be a one-hot encoding of the initial sequence (excluding the termination symbol). Let $D = B$ and let π be the TKF parameter corresponding to the equilibrium probability of each letter. For all $m \in \{1, ..., M+1\}$ and $b \in \{1, ..., B\}$, assign

$$c_{m,b} := \pi_b. \tag{S19}$$

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

$$\beta(\tau) := \frac{1 - e^{-(\mu-\lambda)\tau}}{\mu - \lambda e^{-(\mu-\lambda)\tau}} \tag{S20}$$

It is convenient to index states k and k' using the corresponding (m, g) values (Equation S15), ie. $m = (k + k\%2)/2$, $g = k\%2$, $m' = (k' + k'\%2)/2$ and $g' = k'\%2$. Then we define the transition matrix

$$a^{(t)}_{k,k'} := \begin{cases} [\mu\beta(\tau)]^{m'-m-1+g} e^{-\mu t}[1 - \lambda\beta(\tau)] & \text{if } m - g < m' < M+1 \text{ and } g' = 0 \\ \lambda\beta(\tau) & \text{if } m - g = m' - 1 \text{ and } g' = 1 \\ [\mu\beta(\tau)]^{m'-m-2+g}[1 - e^{-\mu t} - \mu\beta(\tau)][1 - \lambda\beta(\tau)] & \text{if } m - g < m' - 1 < M+1 \text{ and } g' = 1 \\ [1 - \lambda\beta(\tau)][\mu\beta(\tau)]^{M-m+g} & \text{if } m' = M+1 \text{ and } g' = 0 \\ 0 & \textit{otherwise.} \end{cases} \tag{S21}$$

The initial transition vector follows the same form, and can be written as $a^{(i)}_k := a^{(t)}_{0,k}$. The termination state is the final state $k = K = 2M + 2$. Let $s > 0$ be the TKF substitution rate parameter and define the substitution matrix

$$\ell_{b,b'} := \begin{cases} e^{-s\tau} + \pi_{b'}(1 - e^{-s\tau}) & \text{if } b = b' \\ \pi_{b'}(1 - e^{-s\tau}) & \text{if } b \neq b' \end{cases} \tag{S22}$$

With these definitions, $y \sim \text{MuE}(x, c, a, \ell)$ is the transition distribution of the Thorne-Kishino-Felsenstein model after the sequence x evolves for time τ.

Note that in the limit $\lambda, \mu \to 0$ we recover the no-indel special case of the MuE (Section S3.2). In the limit $\tau \to 0$ we recover the no-mutation case, and $y = x$ with probability one. Figure S12 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 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 Thorne-Kishino-Felsenstein (TKF) alignment scoring system. Our alignment notation in this section follows Thorne et al. [15]. "`X`" represents a residue and "`-`" a gap. (This notation is equivalent to the ω^x, ω^y pairwise alignment notation from Section S4.1, with `X` corresponding to a 1 in ω and `-` corresponding to a 0.) "`.`"

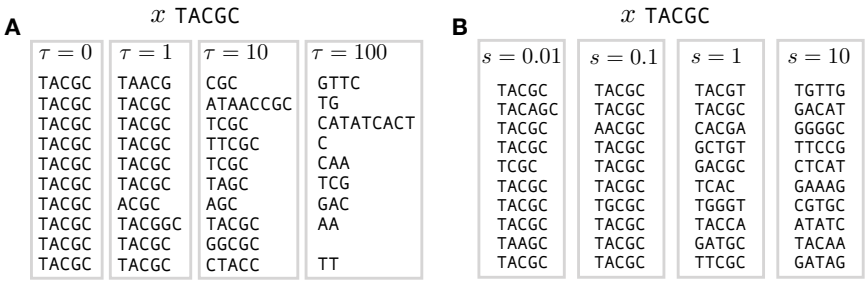


Figure S12: Samples from the Thorne-Kishino-Felsenstein model. Initial sequence TACGC, $\mu = 0.02$, and $\lambda = 0.01$. A. $s = 0.01$ and varying τ. B. $\tau = 1$ and varying s.

represents the "immortal link" in the model, the start of the sequence. We use "$" as a termination symbol. Following the original paper, we define, for $\nu \in \{1, 2, ...\}$,

$$\begin{aligned} p_\nu(\tau) &:= e^{-\mu\tau}[1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \\ p'_0(\tau) &:= \mu\beta(\tau) \\ p'_\nu(\tau) &:= [1 - e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \\ p''_\nu(\tau) &:= [1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \end{aligned} \tag{S23}$$

As before, we refer to states k by tuples (m, g), where $m = (k + k\%2)/2$ and $g = k\%2$. The TKF model assigns probabilities to a pairwise alignment based on the pattern of residues and gaps; we enumerate all possible state transitions in the MuE Markov model 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. Transitioning from $(m, 0)$ to $(M+1, 0)$. This gives a factor $[p_0'(\tau)]^{M-m} = [\mu\beta(\tau)]^{M-m}$.

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

9. Transitioning from $(m, 1)$ to $(M+1, 0)$. 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 $p_1''(\tau)[p_0'(\tau)]^{m-2}p_1'(\tau) = [1 - \lambda\beta(\tau)][\mu\beta(\tau)]^{m-2}e^{-\mu\tau}[1 - \lambda\beta(\tau)]$.

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

13. Initial transition to $(M+1, 0)$. 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

$$(m,g) \to (m',g') : \begin{cases} [\mu\beta(t)]^{m'-m-1+g} e^{-\mu t}[1-\lambda\beta(t)] \\ \qquad \text{if } m-g < m' < M+1 \text{ and } g'=0 \\ \lambda\beta(t) \quad \text{if } m-g = m'-1 \text{ and } g'=1 \\ [\mu\beta(t)]^{m'-m-2+g}[1-e^{-\mu t}-\mu\beta(t)][1-\lambda\beta(t)] \\ \qquad \text{if } m-g < m'-1 < M+1 \text{ and } g'=1 \\ [\mu\beta(t)]^{M-m+g} \\ \qquad \text{if } m'=M+1 \text{ and } g'=0 \\ 0 \qquad \textit{otherwise} \end{cases} \tag{S24}$$

And with each initial transition

$$(0,0) \to (m,g) : \begin{cases} [1-\lambda\beta(t)][\mu\beta(t)]^{m-1} e^{-\mu t}[1-\lambda\beta(t)] & \text{if } 0 < m < M+1 \text{ and } g=0 \\ [1-\lambda\beta(t)]\lambda\beta(t) & \text{if } m=1 \text{ and } g=1 \\ [1-\lambda\beta(t)][\mu\beta(t)]^{m-2}[1-e^{-\mu t}-\mu\beta(t)][1-\lambda\beta(t)] & \text{if } 1 < m < M+1 \text{ and } g=1 \\ [1-\lambda\beta(t)][\mu\beta(t)]^{M} & \text{if } m=M+1 \text{ and } g=0 \\ 0 & \textit{otherwise} \end{cases} \tag{S25}$$

However, these are unnormalized probability factors, not complete probabilities. Note that every alignment will have 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 S21. We can check that this transition matrix normalized. From a state $(m, 0)$, the total outward transition probability is one:

$$\begin{aligned} &\sum_{m'=m+1}^{M} [\mu\beta]^{m'-m-1} e^{-\mu\tau}[1-\lambda\beta] + \lambda\beta + \sum_{m'=m+2}^{M+1} [\mu\beta]^{m'-m-2}[1-e^{-\mu\tau}-\mu\beta][1-\lambda\beta] + [\mu\beta]^{M-m}(1-\lambda\beta) \\ &\quad = \frac{1-(\mu\beta)^{M-m}}{1-\mu\beta}[1-e^{-\mu\tau}-\mu\beta+e^{-\mu\tau}][1-\lambda\beta] + \lambda\beta + [\mu\beta]^{M-m}(1-\lambda\beta) \\ &\quad = 1-(\mu\beta)^{M-m}[1-\lambda\beta] + [\mu\beta]^{M-m}(1-\lambda\beta) \\ &\quad = 1. \end{aligned} \tag{S26}$$

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

$$\begin{aligned}
&\sum_{m'=m}^{M} [\mu\beta]^{m'-m} e^{-\mu\tau}[1-\lambda\beta] + \lambda\beta + \sum_{m'=m+1}^{M+1} [\mu\beta]^{m'-m-1}[1-e^{-\mu\tau}-\mu\beta][1-\lambda\beta] + [\mu\beta]^{M-m+1}(1-\lambda\beta) \\
&\quad = \frac{1-(\mu\beta)^{M-m+1}}{1-\mu\beta}[1-e^{-\mu\tau}-\mu\beta+e^{-\mu\tau}][1-\lambda\beta] + \lambda\beta + [\mu\beta]^{M-m+1}(1-\lambda\beta) \\
&\quad = 1-(\mu\beta)^{M-m+1}[1-\lambda\beta] + [\mu\beta]^{M-m+1}(1-\lambda\beta) \\
&\quad = 1.
\end{aligned} \tag{S27}$$

Conditional on the mth residue of x being aligned to the lth residue of y (i.e. $w_{l,2m} = 1$), 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 (ie. $w_{l,2m-1} = 1$), 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:

$$\begin{aligned}
p_{\text{MuE}}(y_{n,b} = 1 | w, x, c, a, \ell) &= \sum_{b'} c_{m,b'}\ell_{b',b} \\
&= \pi_b e^{-s\tau} + (\pi_b)^2(1-e^{-s\tau}) + \sum_{b''\neq b} \pi_{b''}\pi_b(1-e^{-s\tau}) \\
&= \pi_b e^{-s\tau} + \pi_b(1-e^{-s\tau}) = \pi_b.
\end{aligned} \tag{S28}$$

□

S4.3 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. Figure S13 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 [16]. 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

$$a^{(t)}_{k,k'} := \begin{cases} \frac{1-2\gamma-\kappa}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1=m'\le M \text{ and } g=g'=0 \\ \frac{\gamma\epsilon^{m'-m-2}(1-\epsilon-\kappa)}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1<m'\le M \text{ and } g=g'=0 \\ \frac{\gamma}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1=m'\le M \text{ and } g=0 \text{ and } g'=1 \\ \frac{\gamma\epsilon^{M-m-1}\kappa}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1<m'=M+1 \text{ and } g=g'=0 \\ \frac{\kappa}{\gamma+\kappa} & \text{if } m+1=m'=M+1 \text{ and } g=g'=0 \\ \frac{\gamma}{\gamma+\kappa} & \text{if } m+1=m'=M+1 \text{ and } g=0 \text{ and } g'=1 \\ \frac{1-\epsilon-\kappa}{1-\kappa} & \text{if } m=m'\le M \text{ and } g=1 \text{ and } g'=0 \\ \frac{\epsilon}{1-\kappa} & \text{if } m=m'\le M \text{ and } g=g'=1 \\ \frac{\kappa}{\epsilon+\kappa} & \text{if } m=m'=M+1 \text{ and } g=1 \text{ and } g'=0 \\ \frac{\epsilon}{\epsilon+\kappa} & \text{if } m=m'=M+1 \text{ and } g=g'=1 \\ 0 & \textit{otherwise} \end{cases} \tag{S29}$$

where as before, we refer to states k by tuples (m, g), where $m = (k + k\%2)/2$ and $g = k\%2$. The initial transition vector is defined by $a^{(i)}_k := a^{(t)}_{0,k}$. The termination state is $k = K = 2M + 2$. Define the substitution matrix

$$\ell_{b,b'} := \frac{\psi_{b,b'}}{\pi_b} \tag{S30}$$

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

$$c_m := \ell^{-1} \cdot \pi \tag{S31}$$

where ℓ^{-1} is the inverse of the substitution matrix (assumed to be an invertible matrix). With these definitions, $y \sim \text{MuE}(x, c, a, \ell)$ is equivalent to the conditional distribution of y given x under the pair HMM.

Note that if $\gamma = 0$ then we recover the no-indel case of the MuE distribution (Section S3.2). If, in addition, $\psi = I_B$ then we recover the no-mutation case 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. We use ω^x, ω^y notation to represent alignments, with the symbol "|" placed to the right of the residue we are transitioning *from*.

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

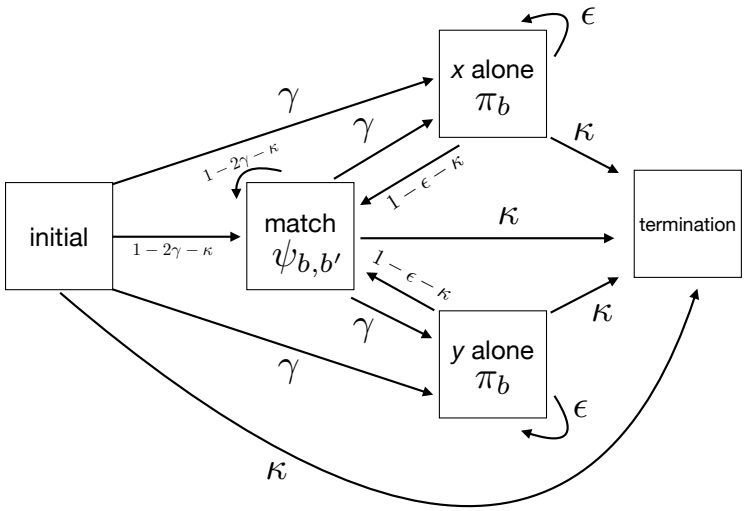


Figure S13: Pair HMM state diagram.

```
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. Transitioning from $(m < M, 0)$ to $(M+1, 0)$, the termination state, has probability $\gamma\epsilon^{M-m-1}\kappa$.

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

5. Transitioning from $(M, 0)$ to $(M+1, 0)$, the termination state, 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. Transitioning from $(M + 1, 1)$ to $(M + 1, 0)$ 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. Transitioning from the initial state to the termination state has probability $\gamma\epsilon^{M-1}\kappa$ when $M > 0$.

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

13. Transitioning from the initial state to $(M+1, 0)$, the termination state, has probability κ when $M = 0$.

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

These transition probabilities were derived without conditioning on the fact that x 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-only 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-only 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-only 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-only 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-only 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-only 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

$$\begin{aligned}
&(1-2\gamma-\kappa)+\gamma\sum_{m'=m+2}^{M}\epsilon^{m'-m-2}(1-\epsilon-\kappa)+\gamma+\gamma\epsilon^{M-m-1}\kappa+\gamma\epsilon^{M-m-1}(1-\kappa)\\
&+(\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})\\
&=1-\gamma+\gamma(1-\epsilon-\kappa)\frac{1-\epsilon^{M-m-1}}{1-\epsilon}+\gamma\epsilon^{M-m-1}+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon}\\
&=1-\gamma+\gamma(1-\epsilon^{M-m-1})+\gamma\epsilon^{M-m-1}\\
&=1
\end{aligned} \tag{S32}$$

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

$$\gamma+\kappa+(1-\gamma-\kappa)=1 \tag{S33}$$

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

$$(1-\epsilon-\kappa)+\epsilon+\kappa=1 \tag{S34}$$

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

$$\kappa+\epsilon+(1-\epsilon-\kappa)=1 \tag{S35}$$

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

$$\begin{aligned}
&(1-2\gamma-\kappa)+\sum_{m=2}^{M}\gamma\epsilon^{m-2}(1-\epsilon-\kappa)+\gamma+\gamma\epsilon^{M-1}\kappa+\gamma\epsilon^{M-1}(1-\kappa)+(\kappa+\gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon})\\
&=1-\gamma+\gamma(1-\epsilon-\kappa)\frac{1-\epsilon^{M-1}}{1-\epsilon}+\gamma\kappa\epsilon^{M-1}+\gamma\epsilon^{M-1}(1-\kappa)+\gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon}\\
&=1-\gamma+\gamma(1-\epsilon^{M-1})-\gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon}+\gamma\epsilon^{M-1}+\gamma\kappa\frac{1-\epsilon^{M-1}}{1-\epsilon}\\
&=1
\end{aligned} \tag{S36}$$

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

$$\gamma+\kappa+(1-\gamma-\kappa)=1 \tag{S37}$$

Consolidating transition probabilities and conditioning on the length of x yields the transition matrix Equation S29.

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,

   $$y_l \sim \text{Categorical}\Big(\sum_{b''} x_{m,b''}\ell_{b''}\Big) = \text{Categorical}\Big(\frac{\psi_b}{\pi_b}\Big). \tag{S38}$$

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,

   $$y_l \sim \text{Categorical}\big((\pi^\top \cdot \ell^{-1} \cdot \ell)^\top\big) = \text{Categorical}(\pi). \tag{S39}$$

   where $\top$ is the transpose symbol.

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.

□

S4.4 Profile HMM

The profile HMM (pHMM) is a widely used model for defining protein sequence families, inferring multiple sequence alignments, and performing database searches [16]. 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

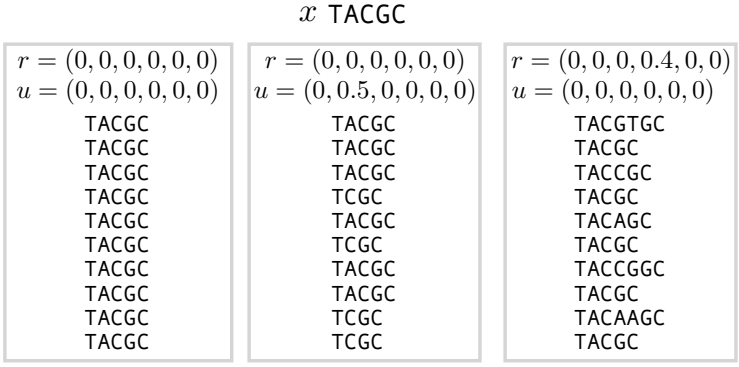


Figure S14: Samples from the profile HMM. The "ancestral" sequence x is set to TACGC, and we set $r_{m,j=0} = r_{m,j=1} = r_{m,j=2}$ and $u_{m,j=0} = u_{m,j=1} = u_{m,j=2}$ for all m.

deletion parameter $u_{m,j} \in [0,1]$ for all $m \in \{1, ..., M+1\}$ and $j \in \{0,1,2\}$, with $u_{M+1,j} = 0$ for $j \in \{0,1,2\}$. Then define the MuE transition matrix

$$a^{(t)}_{k,k'} := \begin{cases} (1 - r_{m+1-g,g})(1 - u_{m+1-g,g}) \\ \qquad \text{if } m+1-g = m' \text{ and } g' = 0 \\ (1 - r_{m+1-g,g}) u_{m+1-g,g} (\prod_{m''=m+2-g}^{m'-1} [(1 - r_{m'',2}) u_{m'',2}]) (1 - r_{m',2})(1 - u_{m',2}) \\ \qquad \text{if } m+1-g < m' \leq M+1 \text{ and } g' = 0 \\ r_{m+1-g,g} \\ \qquad \text{if } m+1-g = m' \text{ and } g' = 1 \\ (1 - r_{m+1-g,g}) u_{m+1-g,g} (\prod_{m''=m+2-g}^{m'-1} [(1 - r_{m'',2}) u_{m'',2}]) r_{m',2} \\ \qquad \text{if } m+1-g < m' \leq M+1 \text{ and } g' = 1 \\ 0 \quad \text{otherwise} \end{cases} \tag{S40}$$

where as before, we refer to states k by tuples (m, g), with $m = (k + k\%2)/2$ and $g = k\%2$. The initial transition vector is defined by $a^{(i)}_k := a^{(t)}_{0,k}$. The state $k = K = 2M + 2$ is the termination state. Let the MuE substitution matrix ℓ be the identity matrix I_B, ie.

$$\ell_{b,b'} := \delta_{b,b'} \tag{S41}$$

for $b, b' \in \{1, ..., B\}$. Then the profile HMM can be written as

$$y_i \sim \text{MuE}(x, c, a, \ell). \tag{S42}$$

Figure S14 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 case of the MuE, since ℓ is the identity matrix (Section S3.2).

Proof

This result follows from the relabeling of the profile HMM Markov state architecture with the (m, g) notation used for the MuE distribution (Figure S15). "Deletion states" in a

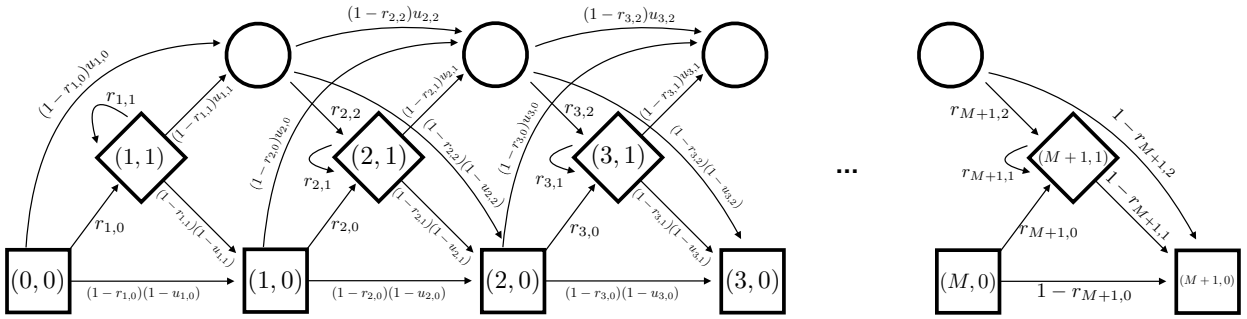


Figure S15: Profile HMM state architecture. The conventional profile HMM state architecture labeled with MuE states, using (m, g) notation; recall that $k = 2m - g$. The state $(0, 0)$ represents the initial state, and the state $(M + 1, 0)$ represents the termination state. Squares indicate match states, diamonds indicate insertion states, and circles indicate "deletion states".

profile HMM 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 S40.

The emission probability of each state in the pHMM is set by its associated emission vector. Without loss of generality, we can write any emission matrix of the pHMM as $e = \xi \cdot x + \zeta \cdot c$. This is equivalent to the MuE emission matrix, since ℓ is set to the identity matrix.

□

S4.5 Needleman-Wunsch

The Needleman-Wunsch (NW) algorithm is a classic non-probabilistic alignment method [48]. Let G be the NW gap penalty (assumed to be negative) and define $u := e^G$. We define the MuE transition matrix

$$a^{(t)}_{k,k'} := \begin{cases} \frac{1-u}{1+u} u^{m'-m-1+g} & \text{if } m - g < m' < M + 1 \text{ and } g' = 0 \\ \frac{1-u}{1+u} u^{m'-m+g} & \text{if } m - g < m' \leq M + 1 \text{ and } g' = 1 \\ \frac{1+u^2}{1+u} u^{M+1-m+g} & \text{if } m' = M + 1 \text{ and } g' = 0 \\ 0 & \textit{otherwise} \end{cases} \tag{S43}$$

where as before, we refer to states k by tuples (m, g), where $m = (k + k\%2)/2$ and $g = k\%2$. The initial transition vector is defined by $a^{(i)}_k := a^{(t)}_{0,k}$. The state $k = K = 2M + 2$ is the termination state. 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 $b, b' \in \{1, ..., B\}$,

$$\ell_{b,b'} := \frac{e^{S_{b,b'}}}{B}. \tag{S44}$$

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

$$c_m := \ell^{-1}(1/B, ..., 1/B)^\top \tag{S45}$$

where ℓ^{-1} is the inverse of the substitution matrix (assumed to be invertible) and $(1/B, ..., 1/B)^\top$ is a length B vector. Let x and y each be one-hot sequence encodings. Now, under the MuE model $y \sim \text{MuE}(x, c, a, \ell)$, 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 $G \to -\infty$ we recover the no-indel case of the MuE distribution. If, in addition, $S_{b,b'} \to -\infty$ for all $b' \neq b$, we recover the no-mutation case 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. Transitioning from $(m, 0)$ to $(M + 1, 0)$ gives a NW score of $(M - m - 1)G$.

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

6. Transitioning from $(m, 1)$ to $(M + 1, 0)$ gives a NW score of $(M - m)G$.

   ```
   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

$$\Delta(l', m, g, m', g') = \begin{cases} (m' - m - 1 + g)G + \sum_{b,b'} x_{m',b} S_{b,b'} y_{l',b'} & \text{if } m - g < m' < M + 1 \text{ and } g' = 0 \\ (m' - m + g)G & \text{if } m - g < m' \leq M + 1 \text{ and } g' = 1 \\ (M - m + g)G & \text{if } m' = M + 1 \text{ and } g' = 0 \\ -\infty & otherwise \end{cases} \tag{S46}$$

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

$$\arg\max_{\vec{m},\vec{g}} \sum_{l=1}^{L+1} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) \tag{S47}$$

where the vectors $\vec{m}$ and $\vec{g}$ are each length $L + 2$ and we set $m_0 = 0, g_0 = 0, m_{L+1} = M + 1, g_{L+1} = 0$. If we find the solution to this objective function, then follow the mapping from the list of Markov chain states $(m_1, g_1), ..., (m_{L+1}, g_{L+1})$ back to an alignment (Section S4.1.3), 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

$$\arg\max_w \log p(y, w|x, c, a, \ell) = \arg\max_w \left[\sum_{l=2}^{L+1} \log p(y_l, w_l|w_{l-1}, x, c, a, \ell) + \log p(y_1, w_1|x, c, a, \ell) \right] \tag{S48}$$

Let $k_l = \arg\max_k w_l$ be the state of the Markov model at the lth residue and let $m_l = (k_l + k_l \% 2)/2$ and $g_l = k_l \% 2$. We then have, under the MuE model,

$$p(y_l, w_l|w_{l-1}) = \begin{cases} \frac{1-u}{1+u} u^{m_l - m_{l-1} - 1 + g_{l-1}} \frac{1}{B} \exp(\sum_{b,b'} x_{m_l,b} S_{b,b'} y_{l,b'}) & \text{if } m_{l-1} - g_{l-1} < m_l < M + 1 \text{ and } g_l = 0 \\ \frac{1-u}{1+u} u^{m_l - m_{l-1} + g_{l-1}} \frac{1}{B} & \text{if } m_{l-1} - g_{l-1} < m_l \leq M + 1 \text{ and } g_l = 1 \\ \frac{1+u^2}{1+u} u^{M+1-m_{l-1}+g_{l-1}} & \text{if } m_l = M + 1 \text{ and } g_l = 0 \\ 0 & otherwise \end{cases} \tag{S49}$$

$$p(y_1, w_1) = \begin{cases} \frac{1-u}{1+u} u^{m_1 - 1} \frac{1}{B} \exp(\sum_{b,b'} x_{m_1,b} S_{b,b'} y_{1,b'}) & \text{if } m_1 < M + 1 \text{ and } g_1 = 0 \\ \frac{1-u}{1+u} u^{m_1} \frac{1}{B} & \text{if } m_1 \leq M + 1 \text{ and } g_1 = 1 \\ \frac{1+u^2}{1+u} u^{M+1} & \text{if } m_1 = M + 1 \text{ and } g_1 = 0 \\ 0 & otherwise \end{cases} \tag{S50}$$

Note that we must have $m_{L+1} = M + 1$ and $g_{L+1} = 0$, since the sequence y is of length L. Now, the maximum *a posteriori* estimator of w can be written as

$$\begin{aligned} \arg\max_w \log p(y, w|x) &= \arg\max_{\vec{m},\vec{g}} \left[L \log(\frac{1-u}{1+u} \frac{1}{B}) + \log(\frac{1+u^2}{1+u}) + \sum_{l=1}^{L+1} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) \right] \\ &= \arg\max_{\vec{m},\vec{g}} \sum_{l=1}^{L+1} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) \end{aligned} \tag{S51}$$

where $m_0 = 0$ and $g_0 = 0$. This objective function is identical to the NW objective function (Equation S47), so the maximum *a posteriori* estimator of w in the MuE distribution corresponds to the Needleman-Wunsch pairwise alignment of x and y.

□

S4.6 Vogel et al. Natural Language Translation

An interesting point of comparison for the MuE distribution is the natural language translation model presented in Vogel et al. [49]. This model takes the same form as a MuE distribution, with x a one-hot encoding of a sentence in one language and y one-hot encoding of a sentence in another language. With states k indexed by tuples (m, g), where $m = (k + k\%2)/2$ and $g = k\%2$, the transition matrix takes the form

$$a^{(t)}_{k,k'} := \begin{cases} \frac{r_{M+m'-m}}{\sum_{m''=1}^{M} r_{M+m''-m}} & \text{if } g = g' = 0 \text{ and } m, m' \leq M \\ 0 & otherwise \end{cases} \tag{S52}$$

where $r \in \mathbb{R}^{2M}_+$ is a vector of non-zero weights. The initial transition vector is defined by $a^{(i)}_k := a^{(t)}_{0,k}$. The length of y is sampled independently of w.

However, the Vogel et al. model is not a MuE distribution: the transition matrix does not satisfy the condition that $a^{(t)}_{k,k'} = 0$ whenever $k' + k'\%2 - k + k\%2 \leq 0$ for all accessible states k, so the Vogel et al. model does not produce valid biological sequence alignments.

S5 Model Details

In this section we provide a detailed description of the models evaluated in the main text. For the MuE distribution in each, we choose a to fit the form of the profile HMM (Section S4.4) with the additional restrictions $r_{m,j=0} = r_{m,j=1} = r_{m,j=2} =: r_m$ and $u_{m,j=0} = u_{m,j=1} = u_{m,j=2} =: u_m$. We set $u_{M+1} = 0$. 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. The transition matrix is

$$a^{(t)}_{k,k'} = \begin{cases} (1 - r_{m+1-g})(1 - u_{m+1-g}) \\ \qquad \text{if } m + 1 - g = m' \text{ and } g' = 0 \\ (1 - r_{m+1-g})u_{m+1-g}(\prod_{m''=m+2-g}^{m'-1}[(1 - r_{m''})u_{m''}])(1 - r_{m'})(1 - u_{m'}) \\ \qquad \text{if } m + 1 - g < m' \leq M + 1 \text{ and } g' = 0 \\ r_{m+1-g,g} \\ \qquad \text{if } m + 1 - g = m' \text{ and } g' = 1 \\ (1 - r_{m+1-g})u_{m+1-g}(\prod_{m''=m+2-g}^{m'-1}[(1 - r_{m''})u_{m''}])r_{m'} \\ \qquad \text{if } m + 1 - g < m' \leq M + 1 \text{ and } g' = 1 \\ 0 \quad \text{otherwise} \end{cases} \tag{S53}$$

where, as in Section S4.4, $m = (k+k\%2)/2$, $g = k\%2$, $m' = (k'+k'\%2)/2$ and $g' = k'\%2$. The initial transition vector follows the same form as the transition matrix, and can be written as $a^{(i)}_k = a^{(t)}_{0,k}$. Rather than assign a termination state we assume the length of the sequence

y_i, ie. L_i, is independent of w (Section S3.1); for convenience, we assign $p(y_l|w_{l,K} = 1) = 0$ for all possible y_l, making the state $k = K$ inaccessible. Since the probability of L_i does not contribute to the per residue perplexity performance metric (Section S7) we do not use an explicit model for L_i.

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

S5.1 Profile HMM

The profile HMM is

$$y_i \sim \text{MuE}(x, c, a(r, u), \ell = I_B) \tag{S54}$$

where $a(r, u)$ depends deterministically on the parameters r and u according to Equation S53, $D = B$ and I_B is the $B \times B$ identity matrix.

S5.2 RegressMuE

The RegressMuE model uses a linear regression model as the H-MuE's continuous-space vector model. Let $h_{i,1}, ..., h_{i,T}$ be covariates associated with sequence y_i. Let $\beta_0^{(x)}, ..., \beta_T^{(x)} \in \mathbb{R}^{M \times D}$ be a set of coefficients associated with x, and let $\beta_0^{(c)}, ..., \beta_T^{(c)} \in \mathbb{R}^{(M+1) \times D}$ be a set of coefficients associated with c. Then the RegressMuE is

$$\begin{aligned} v_i^{(x)} &= \beta_0^{(x)} + \sum_{t=1}^{T} h_{i,t}\beta_t^{(x)} \\ v_i^{(c)} &= \beta_0^{(c)} + \sum_{t=1}^{T} h_{i,t}\beta_t^{(c)} \\ y_i &\sim \text{MuE}(x_i = \text{softmax}(v_i^{(x)}), c_i = \text{softmax}(v_i^{(c)}), a(r, u), \ell). \end{aligned} \tag{S55}$$

Note that in this model, unlike the pHMM, the substitution matrix ℓ is not constrained to the identity. In the no-indel special case of the MuE (Section S3.2), when $r_m = q_m = 0$ for all m and $\ell = I_B$, the RegressMuE reduces to a multi-output multinomial logit regression model.

S5.3 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,

$$\begin{aligned} z_{i,t} &\sim \text{Normal}(0,1) \\ v_i^{(x)} &= \beta_0^{(x)} + \sum_{t=1}^{T} z_{i,t}\beta_t^{(x)} \\ v_i^{(c)} &= \beta_0^{(c)} + \sum_{t=1}^{T} z_{i,t}\beta_t^{(c)} \\ y_i &\sim \text{MuE}(x_i = \text{softmax}(v_i^{(x)}), c_i = \text{softmax}(v_i^{(c)}), a(r,u), \ell) \end{aligned} \tag{S56}$$

S5.4 NeuralMuE

The NeuralMuE model uses a fully connected neural network as the H-MuE's continuous-space vector model. We use a Γ-layer network with 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.

$$\begin{aligned} v_{i,\Gamma+1} &= \beta_{\Gamma+1,0} + \sum_{t=1}^{T_{\Gamma+1}} h_{i,t}\beta_{\Gamma+1,t} \\ v_{i,\Gamma} &= \beta_{\Gamma,0} + \sum_{t=1}^{T_{\Gamma}} \text{relu}(v_{i,\Gamma+1,t})\beta_{\Gamma,t} \\ &\dots \\ v_{i,1}^{(x)} &= \beta_{1,0}^{(x)} + \sum_{t=1}^{T_1} \text{relu}(v_{i,2,t})\beta_{1,t}^{(x)} \\ v_{i,1}^{(c)} &= \beta_{1,0}^{(c)} + \sum_{t=1}^{T_1} \text{relu}(v_{i,2,t})\beta_{1,t}^{(c)} \\ y_i &\sim \text{MuE}(x = \text{softmax}(v_{i,1}^{(x)}), c = \text{softmax}(v_{i,1}^{(c)}), a(r,u), \ell) \end{aligned} \tag{S57}$$

S5.5 LatentNeuralMuE

The LatentNeuralMuE model uses a neural network latent variable model as the H-MuE'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.

$$\begin{aligned}
z_{i,t} &\sim \text{Normal}(0, 1) \\
v_{i,\Gamma+1} &= \beta_{\Gamma+1,0} + \sum_{t=1}^{T_{\Gamma+1}} z_{i,t}\beta_{\Gamma+1,t} \\
v_{i,\Gamma} &= \beta_{\Gamma,0} + \sum_{t=1}^{T_{\Gamma}} \text{relu}(v_{i,\Gamma+1,t})\beta_{\Gamma,t} \\
&\dots \\
v_{i,1}^{(x)} &= \beta_{1,0}^{(x)} + \sum_{t=1}^{T_1} \text{relu}(v_{i,2,t})\beta_{1,t}^{(x)} \\
v_{i,1}^{(c)} &= \beta_{1,0}^{(c)} + \sum_{t=1}^{T_1} \text{relu}(v_{i,2,t})\beta_{1,t}^{(c)} \\
y_i &\sim \text{MuE}(x = \text{softmax}(v_{i,1}^{(x)}), c = \text{softmax}(v_{i,1}^{(c)}), a(r, u), \ell)
\end{aligned} \tag{S58}$$

S5.6 Priors

We place standard normal priors $\text{Normal}(0, 1)$ over each element of each coefficient matrix β in each model. Recall that each row of the matrix ℓ is constrained to the simplex, $\ell \in \Delta_{B-1}$. To enable easy gradient-based optimization and stochastic variational inference [8], we transform an unconstrained parameter $\tilde{\ell} \in \mathbb{R}^{D\times B}$ with a Gaussian prior to the simplex,

$$\begin{aligned}
\tilde{\ell}_{d,b} &\sim \text{Normal}(0, 1) \\
\ell_d &= \text{softmax}(\tilde{\ell}_d).
\end{aligned} \tag{S59}$$

The variables r_m and u_m are constrained to $[0, 1]$. This corresponds to the first dimension of a simplex Δ_1, and so we apply the same approach,

$$\begin{aligned}
\tilde{r}_{m,j} &\sim \text{Normal}(\mu_{r,j}, 1) \\
r_m &= \frac{\exp(\tilde{r}_{m,1})}{\exp(\tilde{r}_{m,2}) + \exp(\tilde{r}_{m,2})}
\end{aligned} \tag{S60}$$

for $j \in \{1, 2\}$. The variable u_m is handled identically, with prior $\tilde{u}_{m,j} \sim \text{Normal}(\mu_{u,j}, 1)$.

S6 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 [10]. To form this approximation, variational inference minimizes the Kullback-Leibler (KL) divergence between the two distributions,

$$\eta_0 = \arg\min_\eta \text{KL}(q_\eta(\theta|y_{1:N})||p(\theta|y_{1:N})) \tag{S61}$$

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

$$\eta_0 = \arg\max_{\eta} \mathbb{E}_{q_\eta(\theta|y_{1:N})}[\log p(y_{1:N}, \theta)] - \mathbb{E}_{q_\eta(\theta|y_{1:N})}[\log q_\eta(\theta|y_{1:N})] = \arg\max_{\eta} \text{ELBO}(\eta) \tag{S62}$$

We employ mean-field variational inference for H-MuE 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 a recognition network (an encoder) [12, 13]. In particular, we set

$$q_{\eta_z}(z_{1:N}|y_{1:N}) = \prod_{i=1}^{N} q_{\eta_z}(z_i|y_i) = \prod_{i=1}^{N} \mathcal{N}(z_i|f^{(\mu)}(y_i, \eta_z), f^{(\sigma)}(y_i, \eta_z)) \tag{S63}$$

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,

$$\begin{aligned}
y_{i,l}^{(q)} &= \mathbb{E}_{y' \sim \text{MuE}(y_i, c^{(q)}, a(r^{(q)}, u^{(q)}), \ell^{(q)})}[y'_l] \\
v_{i,\Gamma^{(q)}+1}^{(q)} &= \beta_{\Gamma^{(q)}+1,0}^{(q)} + \sum_{l=1}^{L^{(q)}} \sum_{b=1}^{B} y_{i,l,b}^{(q)} \beta_{\Gamma^{(q)}+1,l,b}^{(q)} \\
v_{i,\Gamma^{(q)}}^{(q)} &= \beta_{\Gamma^{(q)},0}^{(q)} + \sum_{t=1}^{T_{\Gamma^{(q)}}} \text{relu}(v_{i,\Gamma^{(q)}+1,t}^{(q)}) \beta_{\Gamma^{(q)},t}^{(q)} \\
&\dots \\
f^{(\mu)} &= \beta_{1,0}^{(q,\mu)} + \sum_{t=1}^{T_1} \text{relu}(v_{i,2,t}^{(q)}) \beta_{1,t}^{(q,\mu)} \\
f^{(\sigma)} &= \left| \beta_{1,0}^{(q,\sigma)} + \sum_{t=1}^{T_1} \text{relu}(v_{i,2,t}^{(q)}) \beta_{1,t}^{(q,\sigma)} \right|.
\end{aligned} \tag{S64}$$

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 model (Section S5), but fix $r_1^{(q)} = r_2^{(q)} = ... = r_{M+1}^{(q)}$ and $u_1^{(q)} = u_2^{(q)} = ... = u_M^{(q)}$ and $c_1^{(q)} = c_2^{(q)} = ... = c_{M+1}^{(q)}$. 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 $y_i^{(q)}$ 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 [50]. Let $\phi = (\beta, r, u, \ell)$ be the global parameters of the H-MuE models

and let η_ϕ be the parameters of the associated mean-field variational distribution. Then the gradient of the ELBO is

$$\begin{aligned}\nabla_\eta \mathrm{ELBO}(\eta) &= \sum_{i=1}^{N}\Big(\nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)q_{\eta_z}(z_i|y_i)}[\log p(y_i|z_i,\phi)] + \nabla_\eta \mathbb{E}_{q_{\eta_z}(z_i|y_i)}\Big[\log \frac{p(z_i)}{q_{\eta_z}(z_i|y_i)}\Big]\Big) \\ &\quad + \nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)}\Big[\log \frac{p(\phi)}{q_{\eta_\phi}(\phi)}\Big] \\ &\approx \frac{N}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\Big(\nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)q_{\eta_z}(z_i|y_i)}[\log p(y_i|z_i,\phi)] + \nabla_\eta \mathbb{E}_{q_{\eta_z}(z_i|y_i)}\Big[\log \frac{p(z_i)}{q_{\eta_z}(z_i|y_i)}\Big]\Big) \\ &\quad + \nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)}\Big[\log \frac{p(\phi)}{q_{\eta_\phi}(\phi)}\Big]\end{aligned} \tag{S65}$$

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 [8, 12, 13]. The remaining terms can be computed analytically (see e.g. [12, 13]). Note that this approach relies crucially on the fact that the marginal likelihood of the MuE model, $p_{\mathrm{MuE}}(y|x,c,a,\ell) = \sum_w p_{\mathrm{MuE}}(y|w,x,c,a,\ell)$, 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.

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

$$\begin{aligned}\mathrm{ELBO}_\chi(\eta) &= \sum_{i=1}^{N}\Big(\mathbb{E}_{q_{\eta_\phi}(\phi)q_{\eta_z}(z_i|y_i)}[\log p(y_i|z_i,\phi)] + \chi \mathbb{E}_{q_{\eta_z}(z_i|y_i)}\Big[\log \frac{p(z_i)}{q_{\eta_z}(z_i|y_i)}\Big]\Big) \\ &\quad + \mathbb{E}_{q_{\eta_\phi}(\phi)}\Big[\log \frac{p(\phi)}{q_{\eta_\phi}(\phi)}\Big].\end{aligned} \tag{S66}$$

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 [51]. 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 $\chi_0 < 1$; this annealing schedule was only used for producing data visualizations, rather than prediction of held out data (Section S10) [52].

S7 Perplexity

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

$$\exp\Big(-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\log p(y_i|L_i)\Big). \tag{S67}$$

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

use

$$\hat{\Omega} := \exp\Big(-\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\frac{1}{L_i}\mathbb{E}_{q(\phi|y_{\mathcal{D}})}[\log p(y_i|L_i,\phi)]\Big) \tag{S68}$$

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 [10],

$$\hat{\Omega} \approx \exp\left(-\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\frac{1}{L_i}\big(\mathbb{E}_{q(\phi|\mathcal{D})q(z_i|y_i)}[\log p(y_i|L_i,z_i,\phi)] + \mathbb{E}_{q(z_i|y_i)}\Big[\log\frac{p(z_i)}{q(z_i|y_i)}\Big]\big)\right). \tag{S69}$$

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,

$$\log \mathrm{BF}_{1,2} := \sum_{i\in\mathcal{T}}\mathbb{E}_{q_2(\phi|\mathcal{D})}[\log p_2(y_i|L_i,\phi)] - \sum_{i\in\mathcal{T}}\mathbb{E}_{q_1(\phi|\mathcal{D})}[\log p_1(y_i|L_i,\phi)] \tag{S70}$$

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 S69. 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. In the interest of making this scale interpretable, we computed the expected per-residue perplexity for a variety of different protein sequence models, covering different regimes. In particular, for each model $p(y)$, we examined the expected perplexity in the large data limit, assuming that the model is true,

$$\Omega := \exp\Big(-\mathbb{E}_{p(y)}\Big[\frac{1}{L}\log p(y|L)\Big]\Big). \tag{S71}$$

The expected perplexity is the exponentiated entropy of the model distribution, and so 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

$$\Omega = \exp(-\mathbb{E}[\log(1/20)]) = 20. \tag{S72}$$

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

$$\Omega = \exp\Big(-\mathbb{E}_{y\sim\mathrm{Categorical}(f)}[\log(f^\top\cdot y)]\Big) = \exp(-\sum_{b=1}^{20} f_b\log f_b) \approx 17.92 \tag{S73}$$

where y is a one-hot encoding of an amino acid and $\top$ is the vector transpose symbol [53, 54].

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 [55]. 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:

$$\begin{aligned}\log \ell_{b,b'} &= \log p(y_{b'} = 1 | x_b = 1) = \log\left(\frac{f_{b,b'}}{f_b}\right) = \log f_{b'} + \log\left(\frac{f_{b,b'}}{f_b f_{b'}}\right) \\ &= \log f_{b'} + \frac{\log(2)}{2}\text{BLOSUM62}_{b,b'}\end{aligned} \tag{S74}$$

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 $\ell_b \in \Delta_{B-1}$ (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 distribution of each amino acid within the ancestral sequence follows the UniprotKB overall frequency across evolution. The expected per residue perplexity is

$$\Omega = \exp(-\mathbb{E}_{x\sim\text{Categorical}(f)}\left[\mathbb{E}_{y\sim\text{Categorical}(x^\top\cdot\ell)}[\log(x^\top \cdot \ell \cdot y)]\right]) \approx 11.00. \tag{S75}$$

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 [56]. 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

$$\ell_{b,b'} = \begin{cases} \frac{q_{\text{codon}}}{19} & \text{if } b \neq b' \\ 1 - q_{\text{codon}} & \text{if } b = b'. \end{cases} \tag{S76}$$

If we further 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

$$\Omega = \exp\left(\mathbb{E}_{y\sim\text{Categorical}(x^\top\cdot\ell)}[\log(x^\top \cdot \ell \cdot y)]\right) \approx 1.024. \tag{S77}$$

S8 Density Estimation

Evolutionarily related sequences were collected using jackhmmer (v3.1) from the UniRef100 dataset (date 6/2019) [57, 58]. 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 [17] and ran the jackhmmer search using the API from the EVcouplings package [59]. We included the full envelope of the profile HMM hit (ie. including residues classified as insertions and deleting gap symbols) 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 H-MuE 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 (Section S9).

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 $L^{(q)} = M$. 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 [60] and a batch 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 (GPUs), and we used gradient accumulation to reduce memory usage.

S9 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 [35]. These were sequenced using single cell sequencing of peripheral blood mononuclear cells obtained from an individual healthy donor. Internal stop codons were deleted from the sequence. We used the provided CellRanger annotations of chain features.

To obtain a latent space representation (Figure 3BCD), we trained the FactorH-MuE

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 S8.

We computed the magnitude of the shift in sequence space along a vector with tail z_0 and head z_1 as

$$\nu_l = \left[\sum_{b=1}^{B} \left(\mathbb{E}[y_{l,b} | \hat{w}_{\text{ref}}, z_1, y_{\mathcal{D}}] - \mathbb{E}[y_{l,b} | \hat{w}_{\text{ref}}, z_0, y_{\mathcal{D}}] \right)^2 \right]^{1/2} \tag{S78}$$

where $y_{\mathcal{D}}$ is the training dataset. The expectation is estimated using the variational approximation to the posterior of the FactorMuE (using 10 Monte Carlo samples). $\hat{w}_{\text{ref}}$ is the maximum *a posteriori* estimator of w given the sequence y_{ref} of the reference protein structure PDB:2BNR ($\hat{w}_{\text{ref}}$ is estimated using a single sample from the variational approximation to the posterior and the Viterbi algorithm). Fixing a particular w value enables easy visual comparison between "aligned" sequences position-by-position.

To understand how the full TCR sequence differs between α and β chains we used the RegressMuE, with the covariate vector h_i 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 S78, with the covariate h in place of z.

S10 Influenza Analysis

We downloaded publicly available influenza A(H3N2) HA sequences from GISAID [26]. 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 and imputed them uniformly at random. Following Lee et al. [42], 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 5 prior to subsampling (GISAID identifiers `EPI_ISL_24813`, `EPI_ISL_24814`, ..., `EPI_ISL_24867`). Our main 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 [61].

We used $M = 360$ in the MuE distribution. We set the prior on indels to $\mu_r = \mu_u = (1000, 1)$ since there is expected to be few indels in this dataset. We trained each model for 7 epochs, which was sufficient for convergence. Hyperparameters and training schedule were otherwise set as in Section S8. To produce the latent embedding in Figure 5, however, we annealed the ELBO weighting χ only up to $\chi_0 = 0.001$ after 7 epochs, forcing the FactorMuE model into the autoencoding limit (Section S6) [52].

To generate sequences and 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. To generate future sequences, we sampled from

$$y_i \sim p_{\text{RegressMuE}}(y | \hat{w}_{\text{ref}}, t = 2024, y_{\mathcal{D}}) \tag{S79}$$

using the variational approximation to the posterior; here $y_{\mathcal{D}}$ is the training dataset and $\hat{w}_{\text{ref}}$ is the maximum *a posteriori* estimator of w for the sequence y_{ref} of the reference protein

structure PDB:4O5N ($\hat{w}_{\text{ref}}$ is estimated using a single sample from the variational approximation to the posterior and the Viterbi algorithm). We computed the magnitude of the shift in sequence space from time t_0 to time t_1 in the RegressMuE as

$$\nu_l = \left[\sum_{b=1}^{B} \left(\mathbb{E}[y_{l,b} | \hat{w}_{\text{ref}}, t = 2019, y_{\mathcal{D}}] - \mathbb{E}[y_{l,b} | \hat{w}_{\text{ref}}, t = 1968, y_{\mathcal{D}}] \right)^2 \right]^{1/2} \tag{S80}$$

The expectation is estimated using the variational approximation to the posterior with 10 Monte Carlo samples. 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. [28].