Generative Statistical Methods for Biological Sequences

A dissertation presented
by
Eli N. Weinstein
to
The Committee on Higher Degrees in Biophysics

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Biophysics

Harvard University
Cambridge, Massachusetts
April 2022

Thesis advisor: Professor Debora S. Marks Eli N. Weinstein

Generative Statistical Methods for Biological Sequences

Abstract

Measuring and making sequences is central to modern biology and biomedicine. From evolutionary biology to immunology to therapeutics and beyond, scientists collect massive datasets of DNA, RNA and protein sequences, and create new sequences in the laboratory through large-scale DNA synthesis or genome editing. This dissertation is about the problem of learning from measurements of complex sequence data and predicting unobserved or future sequences that can be made in the laboratory. The dissertation describes new generative statistical methods for biological sequences, working within the framework of Bayesian statistics and probabilistic machine learning, and establishes theoretical guarantees on these methods using frequentist analysis. Part I proposes new tools for building biological sequence models, critiquing biological sequence models, and designing experiments to synthesize samples from biological sequence models. Part II deals with the use of misspecified models in biological sequence analysis and beyond, developing a new understanding of how such "wrong" models can be used effectively for estimation and discovery. Overall, the dissertation contributes principles and methods for reliable and accurate prediction, analysis and design of biological sequences across biology and biomedicine.

Contents

To my parents and my wife.

Acknowledgments

I would first like to thank my advisor, Debora Marks, for her vision, creativity, encouragement and support. I have also spent a substantial amount of time working with, and learning from, Jeffrey Miller over the course of my PhD. I am deeply grateful for his guidance.

I am lucky to have worked closely with Alan Amin for the past two years, and two of the chapters in this thesis consist of work that is as much his as it is mine (Chapters 2 and 4). Thanks also to my additional coauthors Jonny Frazer, Will Grathwohl, Daniel Kassler and Jean Disset, who have contributed insights and assistance (a detailed accounting of their contributions can be found at the start of the chapters to which they contributed, Chapters 3 and 4).

Fritz Obermeyer and Eli Bingham helped make the mutational emission distribution part of the Pyro probabilistic programming language (Chapter 1). Elizabeth Wood provided key advice and suggestions on T cell receptor sequence data and immunotherapy applications (Introduction, Chapters 1 and 3), as well as copy-editing throughout. Rob Patro provided crucial advice on large scale kmer counting (Chapter 2). Winnie Wang made the clean illustrations for the theory section in Chapter 2 (Appendix B). Steven Weber provided insight into combinatorial DNA libraries (Chapter 3). Jonathan Huggins, Pierre Jacob and Andre Nguyen provided advice on the data selection problem and on Stein discrepancies (Chapter 5).

The dissertation is typeset using the Dissertate LaTeX package, due to Jordan Suchow.

I have benefited greatly from the environment of the Marks lab and I am grateful to members both past and present for scientific discussion, including, besides aforementioned coauthors, Sam Berry, Kelly Brock, Hailey Cambra, Hattie Chung, Christian Dallago, Mafalda Dias, David Ding, Anna Green, Tessa Green, Sarah Gurev, Thomas Hopf, Chan Kang, Aaron Kollasch, Rohan Maddamsetti, Rose Orenbuch, Steffan Paul, Adam Riesselman, Joshua Rollins, Nathan Rollins, Benjamin Schubert, Ada Shaw, June Shin, Han Spinner, Amy Tam, Nikki Thadani, and Noor Youssef. John Ingraham in particular taught me a great deal about probabilistic machine learning for biological sequences, and his advice was especially important for the work in Chapters 1 and 4. Many thanks also to Chris Sander for his support, encouragement and discussion throughout.

My dissertation advisory committee – Tamara Broderick, George Church and Ed Boyden – has critiqued my work over the course of my PhD, and I am grateful for their time and encouragement.

The Harvard Biophysics Graduate Program has made my graduate education possible, and provided a wonderful community; I thank Michele Eva, Jim Hogle, Martha Bulyk and Venkatesh Murthy. Thanks also to Deniz Aksel for so many scientific discussions.

Funding for my graduate education came from the John and Fannie Hertz Foundation, and I benefited further from the remarkable community of fellows.

I am grateful beyond words for my parents, my sister, and my wife. And for Abe.

0

Introduction

Measuring and making sequences is central to modern biology and biomedicine. The past decades have seen twin revolutions in technology for reading and writing DNA, with dramatic decreases in cost and increases in scale. High-throughput sequencing technology has led to the creation of massive sequence datasets, including measurements of genomes from organisms across the tree of life; of human genomes from around the world; of bacterial genomes from in and on the human body; of

viral genomes from decades of evolution; and of much more. Meanwhile, high-throughput synthesis technology has led to the routine creation of vast numbers of precisely defined sequences in the laboratory, which can be tested in parallel using modern assays. Efficient genome editing technology has enabled precise modification of existing sequences inside the cells of humans and other organisms. This dissertation is about statistical methods for learning from sequence data, and forming predictions of new sequences that can be made *in vitro* or *in vivo* using synthesis or genome editing. It is about understanding measurements of sequences in enough detail to be able to make new sequences.

The statistical methods presented fall under the framework of probabilistic machine learning, and are specifically Bayesian (with a few exceptions). They revolve centrally around the construction of generative probabilistic models of sequences, building on a fundamental recipe known as "Box's loop"[26] (Fig. 1):

1. Hypothesize a model of the data, in the form of a probability distribution over measured sequences $X_1, \ldots, X_N$ and hidden variables θ, i.e. $p(X_{1:N}, \theta)$.

2. Infer the values of the hidden variables, i.e. compute the posterior $p(\theta \mid X_{1:N})$. Examining the posterior gives us insight into the data through the lens of our hypothesized model.

3. Criticize the model to determine whether or not it accurately captures the data. (If it fails, return to step 1.)

4. Generate new sequences from the posterior predictive distribution of the model that can be made in the laboratory, i.e. $X'_1, X'_2, \ldots \sim p(x \mid X_{1:N})$.

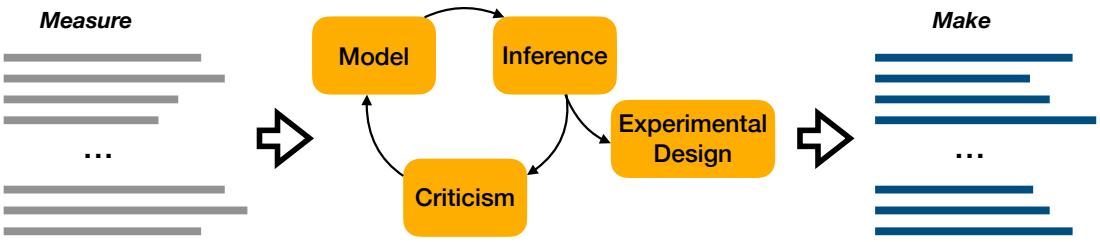


Figure 1: Box's loop [26] provides a framework for going from measurements of sequences to experimental designs for new sequences that can be made in the laboratory.

We can thus use generative probabilistic models to analyze complex sequence data and predict new sequences that can be synthesized experimentally.

While generative methods for biological sequence statistics have existed for decades, they are far from the dominant approach to biological sequence analysis. Instead, existing methods are more commonly either (a) entirely non-probabilistic, with no formally defined estimand, instead simply processing the data through a series of heuristics or (b) focused on predicting some property from sequences, i.e. interested only in conditional distributions $p(y \mid x)$ where y is a covariate of interest (e.g. a phenotype). Generative Bayesian methods allow us to, among other things:

1. Predict and forecast unobserved or future sequences on the basis of past sequence data.
2. Handle uncertainty in inferences drawn from sequence data.
3. Replace heuristic data analysis methods with rigorous and formal statistical methods, whose properties can be analyzed theoretically.
4. Test predictions experimentally, through the construction of novel sequences.

The next section illustrates in more detail why these properties are desirable.

0.1 Applications of generative biological sequence statistics

Our primary goal is to enable new kinds of scientific studies of biological sequences, that are difficult or impossible to perform using current methodology. To illustrate, consider three case studies drawn from virology, immunology and evolutionary biology.

0.1.1 Forecasting pathogen evolution

Rapidly evolving pathogens such as influenza or SARS-CoV-2 are difficult to diagnose, immunize against, and treat, in part because their genome sequences change quickly over time [163,150]. We would therefore like to forecast future genome sequences, to prepare diagnostics, vaccines, and drugs preemptively. In particular, one approach is to (1) assemble a dataset of past sequences collected from different patients at different times, i.e. $(X_1, t_1), \ldots, (X_N, t_N)$, (2) construct $p(x \mid t_{future})$, a prediction of the viral sequences that will be observed at some future time t_{future}, and then (3) make samples $X_1, X_2, \ldots \sim p(x \mid t_{future})$ in the laboratory, so that drugs or diagnostics can be tested against these future sequences. It is especially important that we can form reasonable predictions even with relatively little past sequence data, in order to deal with emerging pathogens. It is also important that we can accurately handle uncertainty, since failure to consider the possibility of a variant that later emerges can have serious real-world consequences: failed diagnostics, failed vaccines, and failed drugs.

Chapter 1 develops new methods for generative sequence regression, i.e. new methods of constructing distributions $p(x \mid t)$ that more accurately account for uncertainty as compared to

commonly-used heuristics. We demonstrate these methods by constructing the first generative forecast of pathogen sequence evolution, focusing on the influenza hemagglutinin protein, the key site of interaction between influenza and the human immune system. Chapter 3 develops an experimental design strategy to efficiently construct large numbers of samples $X'_1, X'_2, \ldots \sim p(x \mid t_{future})$ from generative sequence models in the laboratory. Our overall approach makes possible, for instance, large scale testing of antibody drugs or patient sera against likely future viral antigens, and is generalizable to describe other pathogens or to account for other covariates besides time.

0.1.2 Designing personalized immunotherapies

Cell therapies are a successful and rapidly developing class of therapeutics for cancer and other diseases [131]. TCR T cell therapies use a T cell receptor (TCR) to direct an engineered T cell to kill target cells, e.g. cancer cells. Creating such therapies requires synthesizing TCRs that not only bind specific antigens, but also (1) do not bind any self-antigens in the patient, since this would cause off-target effects and (2) look like the natural TCR sequences found in the patient, to avoid immune rejection of the cell therapy [131]. One possible approach to creating such patient-specific TCR T cell therapies is to (1) measure the repertoire of natural TCR sequences present in a patient or a closely related donor, i.e. record a dataset of sequences $X_1, \ldots, X_N$, (2) estimate the underlying, patient-specific distribution of sequences $p(x)$, and (3) synthesize samples $X'_1, X'_2, \ldots \sim p(x)$ and deliver them into cells to create candidate therapies that can be screened for activity against the tumor. It is crucial in this application that the synthesized sequences accurately match the distribution of patient sequences: we want a high diversity of sequences, to ensure a binder exists, but we also need the

sequences to look like patient TCRs.

Chapters 1 and 2 develop new methods for sequence density estimation, i.e. new methods of estimating distributions $p(x)$ describing sequence samples $X_1, \ldots, X_N$. These are applied to TCR sequencing data, providing a detailed model and map of individual patients' immune systems. Chapter 3 develops new experimental design methods to construct large-scale libraries of approximate samples from models, $X'_1, X'_2, \ldots \sim p(x)$. We apply this method to TCR models, and establish that the resulting libraries can accurately match patients' TCR repertoires using statistical tests developed in Chapter 2. Detailed simulations suggest that our generative statistical methods can potentially yield many orders-of-magnitude more patient-specific binders for TCR T cell therapy as compared to previous techniques.

0.1.3 The past and future of life

Understanding the long-term future evolution of life on Earth is a fundamental biological question. Although difficult to address in general, the question is more tractable when we focus not on entire genomes but on individual proteins whose structure and function has been well-conserved across billions of years of evolution. Here, a common model of long-term evolution describes sequences diffusing over a fixed fitness landscape; under this hypothesis, sequences in the far future can be described as samples from the stationary distribution of the diffusion [230,110]. We would like to estimate the stationary distribution to understand past and future evolution. In particular, given a dataset of present-day genome sequences from across the tree of life, i.e. $X_1, \ldots, X_N$, we would like to estimate the stationary distribution $p^{\infty}(x)$ and then assay samples from the stationary distribution

$X'_1, X'_2, \ldots \sim p^{\infty}(x)$ in the laboratory to determine their properties. A crucial challenge is to remove biases that come from recent phylogenetic history, in order to form a reliable estimate of the underlying landscape that constrains molecular evolution and determines long-run outcomes.

Chapter 4 analyzes a model of long-term molecular evolution, taking into account fitness landscapes and phylogenetic history. It establishes fundamental limits on what we can learn from present-day observational sequence data, and demonstrates how phylogenetic bias can be reduced across a wide range of example protein families.

0.1.4 Conclusions

The above examples are only case studies, representing some specific applications of the methods developed in this dissertation that we have so far explored. Similar questions, however, can be asked in many other subfields of biology: we may be interested in how organisms adapt to climate change, and so want to predict sequences based on temperature or other environmental variables; we may be interested in developing novel therapeutics based on individuals' gut microbiomes, and so want to construct large sequence libraries based on metagenomic data; or we may be interested in the future evolution of a tumor, and want to predict oncogene sequences that could emerge. The methods developed in this dissertation are grounded in underlying statistical and biophysical theory, and can thus be widely applied to address these questions and many more.

0.2 Statistical foundations

In this section, we review some of the key statistical questions that this dissertation addresses, taking a frequentist perspective on Bayesian methodology. Our presentation is general, and holds for any type of data; in Section 0.3 we introduce crucial concepts that arise when working with biological sequence data specifically. Note that our presentation throughout this introduction is heuristic, and we gloss over edge cases and measure-theoretic definitions in an effort to clarify the essential ideas.

0.2.1 Estimating and testing distributions

We assume that sequence data is drawn from some true *data generating distribution* $p_0(x)$ as independently and identically distributed samples,

$$X_1, X_2, \ldots \overset{i.i.d.}{\sim} p_0(x), \tag{1}$$

where each $X_i \in \mathcal{X}$ is a sequence from e.g. a particular individual, species, cell, etc. *Density estimation* is the problem of estimating $p_0(x)$ given a dataset of samples, $\mathcal{D} = \{X_1, \ldots, X_N\}$.

Say we have some other distribution $p_1(x)$ over sequences, for instance from a model. *Goodness-of-fit-tests* asks whether or not $p_1(x) = p_0(x)$, given samples from $p_0(x)$, and given $p_1(x)$. *Two-sample tests* ask the same question, given only samples from $p_1(x)$ rather than the density $p_1(x)$ itself.

We often have covariates Y_i with each sequence X_i, such as the time or place the sequence was

collected, or a property that the sequence possesses, such as whether it binds something or catalyzes a certain chemical reaction. In this case we assume that $(X_1, Y_1), (X_2, Y_2), \ldots \sim_{i.i.d.} p_0(x, y)$. *Regression* is the the problem of estimating the conditional distribution $p_0(x \mid y)$ given a dataset $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$.

This dissertation introduces new density estimation methods, regression methods, goodness-of-fit tests and two-sample tests for biological sequences. Such methods are fundamental tools throughout statistics, and can be used to solve more complex problems. For instance, if we are interested in understanding the causal impact of sequence changes on protein function, in the presence of confounders, we may want to use a propensity score method, which would require a method for sequence regression [120]. Although this dissertation does not go into depth on such downstream uses, they represent an important area of future application.

0.2.2 Models

Models consist of sets of probability distributions, with elements indexed by a parameter, i.e. $\mathcal{M} = \{p_\theta(x) : \theta \in \Theta\}$. In *parametric* models, the dimension of Θ is finite; in *nonparametric* models, the dimension of Θ is infinite. The goal of inference is to find an element of $\mathcal{M}$ that is close to p_0, given a dataset of samples from $\mathcal{D}$. This tells us a latent parameter value and distribution that can explain the observed data. We say a model is *well-specified* if $p_0 \in \mathcal{M}$, so the model can exactly match the data generating distribution. We say a model is *misspecified* if $p_0 \notin \mathcal{M}$, in which case we can only hope to find an element of $\mathcal{M}$ that is close to p_0 according to some distance metric or divergence.

0.2.3 Bayesian inference

Bayesian inference is a method for estimating parameters from data. It proceeds by positing a prior distribution $\pi(\theta)$ over parameters of a model $\mathcal{M}$ and then applying Bayes' rule to construct a posterior distribution over parameters, given the data. For instance, with a dataset of samples $X_1, \ldots, X_N$ from $p_0(x)$, we have the posterior

$$\Pi(\theta|X_1, \ldots, X_N) = \frac{\pi(\theta) \prod_{i=1}^N p_\theta(X_i)}{\int_\Theta \pi(\theta) \prod_{i=1}^N p_\theta(X_i) d\theta}. \tag{2}$$

Bayesian inference is useful because it quantifies uncertainty in possible values of θ, given finite data, providing a distribution over possible parameter values. With infinite data, the posterior will (in general) converge to a delta function at θ_0, where $p_{\theta_0} = p_0$ if the model $\mathcal{M}$ is well-specified. If the model is misspecified, we have $p_{\theta_0} = \text{argmin}_{p_\theta \in \mathcal{M}} \text{KL}(p_0 \| p_\theta)$, where

$$\text{KL}(p_0 \| p_\theta) = \int_{\mathcal{X}} p_0(x) \log[p_0(x)/p_\theta(x)] dx$$

is the *Kullback-Leibler (KL)* divergence. In the misspecified case θ_0 is sometimes referred to as the "pseudo-true" parameter rather than the "true" parameter. Analogous results hold in the regression setting, where we have samples $(X_1, Y_1), (X_2, Y_2), \ldots \sim_{i.i.d.} p_0(x, y)$.

Computing the posterior and sampling likely parameters from the posterior is often challenging. In this case, we will typically use variational inference to approximate the posterior. Variational inference proceeds by first positing a variational family $\mathcal{V} = \{q_\phi(\theta) : \phi \in \Phi\}$ where $q_\phi(\theta) \in$

$\mathcal{P}(\Theta)$. Elements q_ϕ of the variational family $\mathcal{V}$ are chosen to be easy to sample from; for example, they could be Gaussian distributions. Variational inference then proceeds by finding an element of $\mathcal{V}$ that approximates the posterior (Eqn. 2) by minimizing a KL divergence,

$$q_{\phi^*} = \text{argmin}_{q_\phi \in \mathcal{V}} \, \text{KL} \left(q_\phi(\theta) \| \Pi(\theta | X_1, \ldots, X_N) \right). \tag{3}$$

The resulting distribution $q_{\phi^*}(\theta)$ is a tractable approximation to the posterior.

0.3 Sequence space and distributions

So far we have considered statistical questions and methods in the abstract, for any type of data. In this section we focus on the specifics of biological sequence data, describing key spaces and metrics.

0.3.1 Sequence space

A fundamental consideration in statistics is the mathematical space in which the data lies, i.e. $\mathcal{X}$ where $X_1, X_2, \ldots \in \mathcal{X}$. A general definition of the space $\mathcal{X}$ of biological sequences is the set of *finite length* strings of letters drawn from a fixed alphabet. For DNA the alphabet would be the four nucleotides, and for proteins it would be the twenty amino acids. Allowing $\mathcal{X}$ to contain all finite length strings allows us to model the vast majority of genetic elements, including genes, mRNA, proteins, promoters, chromosomes, etc., though note that it does not cover multi-chromosome genomes or other "sequences" that in fact consist of multiple DNA or polypeptide molecules.

Crucially, we will avoid the overly restrictive assumption that $\mathcal{X}$ is the space of *fixed length* strings.

While widespread in the field in the field of biological sequence analysis, such an assumption typically rests on heuristic data preprocessing methods – such as multiple sequence alignment for proteins, or variant calling for genomes – that manipulate variable length sequence data to force it into the space of fixed length strings. These methods are often problematic in that they either make untenable assumptions about future data (e.g. that it has probability zero of being longer than previously observed data), destroy information (e.g. ignore structural variation in genomes) or both.

We will also avoid the assumption that $\mathcal{X}$ is the space of *infinite length* strings. Although this assumption is common in the analysis of phylogenetic sequence models, it is a poor description of reality, particularly when working with protein-length rather than genome-length sequences. Moreover, we are interested in predicting unobserved sequences based on previously observed sequences, rather than predicting the end of a sequence given its start; the relevant asymptotic limit is therefore the limit of large numbers of sequences, not very long sequences.

0.3.2 Sequence distributions

Another important consideration is the set of distributions over sequence space $\mathcal{X}$ that a dataset of sequences might be drawn from, i.e. the set $\mathcal{P}(\mathcal{X})$, where we assume $p_0 \in \mathcal{P}(\mathcal{X})$. We will in general try to avoid the common but overly restrictive assumption that our parametric models $\mathcal{M}$ are well-specified, i.e. that $\mathcal{P}(\mathcal{X}) \subseteq \mathcal{M}$. However, allowing $\mathcal{P}(\mathcal{X})$ to be all possible distributions over finite strings is often too weak an assumption to be tractable theoretically or practically, primarily because of the difficulties of working with extreme variation in sequence length. We therefore introduce plausible assumptions that control the distribution over sequence length.

For many biological sequence datasets, it is appropriate to assume that while there may be variation in sequence length, this variation is not heavy-tailed. Functional constraints, for instance, often restrict variation in sequence length as molecules evolve, e.g. a protein must maintain its three-dimensional shape to perform a particular function and so cannot easily mutate to become extremely long (Sec. 0.4.2). In Chapter 2, we introduce and study *sub-exponential sequence distributions*, which consist of any distribution $p(x)$ for which, for some $t > 0$, we have $\mathbb{E}_{X \sim p(x)}[\exp(t|X|)] < \infty$, where $|X|$ is the length of X.

More restrictive assumptions on $\mathcal{P}(\mathcal{X})$ can be relevant when modeling DNA synthesis. In particular, many synthesis technologies cannot produce arbitrary length sequences, and so in studying these technologies is appropriate to assume that the distribution over sequences that they produce is *bounded*, i.e. the probability of synthesizing a sequence with length above some maximum value is zero. We study bounded sequence distributions in Chapter 3.

0.3.3 Metrics on sequence distributions

To evaluate methods for estimating or approximating sequence distributions, it is important to consider what it might mean for two distributions to be similar or different. In generative biological sequence statistics, we often want to approximate a distribution $p(x)$ closely enough that sequences sampled from our approximation $\tilde{p}(x)$ have the same properties as those sampled from $p(x)$, as measured by some downstream experimental assay. This leads to a natural class of distance metrics on sequence distributions. In particular, let $f(x)$ be a function describing a sequence property; for instance, $f(x)$ could be a quantitative measure of the binding strength of an antibody sequence x,

or a binary indicator for whether or not the antibody x has binding strength above some threshold. We can quantify the distance between $p(x)$ and $\tilde{p}(x)$ using an *integral probability metric (IPM)*, defined as

$$\text{IPM}_{\mathcal{F}}(\tilde{p}, p) = \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{X \sim p}[f(X)] - \mathbb{E}_{X \sim \tilde{p}}[f(X)] \right|, \tag{4}$$

where $\mathcal{F}$ is a set of functions. IPMs measure the worst-case difference in an average property of sequences sampled from each distribution, over all possible assay functions $f \in \mathcal{F}$. A small IPM value guarantees that when we synthesize a library of sequences from our approximation $\tilde{p}$, they will have similar properties to the target distribution p, even when we do not know the assay function f.

IPMs depend crucially on the choice of function class $\mathcal{F}$, which in turn depends on what we can safely assume about the downstream assay function f. In practice, virtually all high-throughput biological assays have limited dynamical range, i.e. there is some lowest possible and highest possible value that they can measure. A natural $\mathcal{F}$ for biological sequence distributions is thus the set of bounded functions, i.e. $\mathcal{F} = \{f : \|f\|_\infty \leq u\}$ where $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)|$ and u is an upper bound. In this case, $\text{IPM}_{\mathcal{F}}$ is (up to a constant factor u) the *total variation (TV)* distance[2.42]. Since the TV distance can be difficult to work with practically, we will try to make sure the TV distance is small by controlling the KL divergence (Chapters 1, 3 and 4), which upper bounds the TV distance, or the Hellinger distance (Chapter 2), which upper and lower bounds the TV distance.

It is useful to draw a contrast here with other areas of statistics, and especially to applications where the data are continuous and we are not interested in physically making and measuring samples from a distribution. In such applications a common goal is to approximate summary statistics of a

target distribution $p(x)$, such as its mean and variance. In this case, TV distance is not so useful: a small TV distance does not ensure a small difference in means $|\mathbb{E}_{X\sim p}[X] - \mathbb{E}_{X\sim\tilde{p}}[X]|$, since for $f(x) = x$ with $\mathcal{X}$ unbounded we have $\|f\|_\infty = \infty$. Instead, the *Wasserstein* distance is often used [115]. By the Kantorovich-Rubinstein duality, the Wasserstein distance is equivalent (up to a constant factor u) to an IPM with $\mathcal{F}$ the set of functions with bounded Lipschitz semi-norm, i.e. $\mathcal{F} = \{f : \|f\|_L \leq u\}$ where $\|f\|_L = \sup_{x,x'\in\mathcal{X}} |f(x) - f(x')|/d(x,x')$, with $d(\cdot,\cdot)$ a distance metric [242]. Small Wasserstein distances imply accurate mean approximations: for $f(x) = x$ and $d(x,x') = |x - x'|$ we have $\|f\|_L = 1 < \infty$. However, bounding the Wasserstein distance is not especially useful in biological sequence statistics. There are many examples of proteins where a single mutation abolishes function, i.e. a very small change in sequence leads to a very large change in a key property (see Ding et al. [56] for an example). Thus in practice for a real assay function f we can expect $\|f\|_L$ to often be large, even as large as $\|f\|_\infty$. In biological sequence statistics, therefore, we do not expect the Wasserstein distance to offer much tighter bounds on the difference in assay output $|\mathbb{E}_{X\sim p}[f(X)] - \mathbb{E}_{X\sim\tilde{p}}[f(X)]|$ as compared to the TV distance.

0.3.4 Entropy of sequence distributions

A useful way of measuring and comparing sequence distributions is in terms of the diversity of sequences that they generate. We will focus on a particular version of the distribution entropy, the *per-residue perplexity (PRP)*,

$$\exp\left(-\mathbb{E}_{X\sim p(x)}\left[\frac{1}{|X|}\log(p(X))\right]\right), \tag{5}$$

where recall $|X|$ is the length of the sequence X. PRP is useful because it is directly comparable between distributions that produce sequences of different lengths, and because it is interpretable as the "effective" number of amino acid or nucleotides that the distribution generates at each position of the sequence, on average across positions.

PRP is an absolute scale on which we can place sequence distributions. Here we give a brief tour of the PRP scale for proteins. The minimum PRP is 1, corresponding to a $p(x)$ that is just a delta function at a single sequence (no diversity). Meanwhile, the PRP for a uniform distribution over all 20 amino acids is 20. We can perform back-of-the-envelope calculations to get a rough sense of the PRP of distributions studied in different areas of biological sequence statistics (calculations in Sec. A.5).

1. **All of life:** A distribution $p(x)$ that generates amino acids sequentially, based on the frequency at which they are observed across all of life, will have a PRP of 17.92.

2. **Evolutionary protein families:** Simple models of evolutionary protein families, i.e. similar or homologous proteins from across life, often rely on the BLOSUM substitution matrices; with the BLOSUM62 matrix, we expect a PRP of 11.00.

3. **Human population:** Based on the number of single nucleotide polymorphisms observed in individual humans relative to the reference genome, we can estimate the PRP of the distribution over human genomes as 1.02.

These calculations are based on simple models of real distributions; a more accurate model of the distribution of sequences across life, for instance, will no doubt have lower PRP than 17.92. Nonethe-

less, they are useful as a rough guess for what PRPs can be expected in different estimation and modeling problems. This dissertation focuses primarily on distributions with higher PRP than the human population but lower PRP than all of life, with typical estimates ranging between 1.5 and 8.

0.3.5 Conclusions

Biological sequence data is not like other kinds of data, and occupies an unusual and challenging position in statistics. Biological sequences cannot be handled with the theory of vector data (e.g. $\mathcal{X} = \{1, \ldots, B\}^M$), since they are variable in length. Nor can sequences be handled using the theory of time series data (e.g. $\mathcal{X} = \{1, \ldots, B\} \times \{1, \ldots, B\} \times \ldots$), since they are finite. Partially as a consequence of length variation, it is difficult to define sensible distance metrics $d(x, x')$ over sequence space $\mathcal{X}$. Moreover, while in other fields a top priority is estimating expectations of known functions – i.e. key distributional summary statistics such as mean and variance – in biological sequence statistics we care more about expectations of unknown functions, which are unlikely to be smooth with respect to any sequence distance we might define *a priori*.

0.4 Biophysical foundations

To build effective statistical methods for biological sequence data we must consider the underlying biophysics of sequence evolution. In this section we describe a broad framework for stochastic models of sequence evolution, accounting for three key phenomena: mutation, fitness and phylogeny. Many existing models can be thought of as special cases, modifications, or extensions of this framework. In particular, although the focus is on describing how species change over evolutionary time,

the same ideas can be applied with modification to a variety of other biological phenomena, such as the development of immune receptor repertoires or tumors within a single organism, or experimental evolution systems.

0.4.1 Mutation

We start by examining models of how DNA mutates over time, as organisms reproduce. Such models typically take the form of a Markov process, with a transition probability function $P^\tau(x, X_0)$ describing the probability that an initial sequence X_0 mutates into another sequence x after time τ (where time may be either discretized or continuous).

The most widely important and well-studied class of mutations are *substitutions*, in which a letter at a particular position in a sequence is replaced with another letter, e.g. because of errors during DNA replication. A standard model says the probability of observing a substitution in a descendent sequence X, at a particular position j, depends only on the letter in that same position in the ancestor, i.e. X_{0j}, and the length of time or number of generations τ that has elapsed since the ancestor. This can be summarized with a transition probability distribution $X \sim P^\tau(x, X_0)$, using a substitution matrix S,

$$X_j \sim \text{Categorical}(X_{0j} \cdot S^\tau) \text{ for all } j \in \{1, \ldots, |X|\}, \tag{6}$$

where j indexes the position in a sequence, S^τ is the matrix S raised to the power τ, and we represent X_0 with a one-hot encoding, i.e. if the bth letter of the alphabet is at position j of X, then

$X_{0jb} = 1$ and $X_{jb'} = 0$ for $b' \neq b$. This independent model is simple and widely used, particularly in phylogenetics, though note that it ignores more complex statistical dependencies that may affect substitution probability, such as the wider sequence context.

A second important class of mutations are *insertions and deletions (indels)*, in which letters are be added or removed from a sequence at a particular position, creating a longer or shorter sequence. While this process is straightforward to simulate – i.e. it is easy to write down implicit models [180] where insertions and deletions are randomly added over time to an ancestral sequence – it is non-trivial to construct probabilistic models with explicit analytic likelihoods over future sequences, $P^\tau(x, X_0)$ [107]. This problem has been extensively studied in biological sequence analysis; a generalized solution for fixed τ appears in Chapter 1. A closely related problem is that of *alignment*: inferring, given two or more sequences, which positions in each sequence are evolutionarily related via substitution mutations ("conserved sites") rather than indels.

Although indels and substitutions are in general the most extensively studied classes of mutations in biological sequence analysis, there are of course other types of mutations, many of which have been understudied primarily because of limitations in sequencing technology rather than biological importance; these include, in particular, mechanisms of large-scale cutting and joining of DNA, such as recombination, structural variation, etc.

0.4.2 Fitness

We next consider the effects of natural selection. To analyze how selection alters the evolution of sequences, we move from the level of individual organisms to that of populations. We again work

with a transition distribution on sequences $P^\tau(x, X_0)$, but now, rather than describing how individual sequences change over time when an organism reproduces, it describes how a population of genomes changes. In particular, let $P^\tau(x, X_0)$ be the probability that a population where genome X_0 is fixed – i.e. the most recent common ancestor of every organism in the population had sequence X_0 – transitions such that x is fixed, after time τ.

In general $P^\tau(x, X_0)$ will depend on both sequence mutation and selection. A *fitness landscape* $F(\cdot) : \mathcal{X} \to \mathbb{R}_+$ describes the relationship between sequence and selection. The (absolute) fitness $\exp[F(x)]$ of a genome sequence x is the number of offspring that organisms with that sequence produce on average. We will consider a simple set of population genetics assumptions, with haploid organisms reproducing according to a Wright process, and the mutation rate assumed to be small relative to the population size. In these conditions, Sella & Hirsh [230] show that the transition operator for a single timestep can be approximated as

$$P^1(x, X_0) = \begin{cases} N\mu(x, X_0)\frac{1-\exp(2[F(X_0)-F(x)])}{1-\exp(2N[F(X_0)-F(x)])} & \text{if } x \neq X_0, \\ 1 - \sum_{x' \neq X_0} P^1(x', X_0) & \text{otherwise,} \end{cases} \tag{7}$$

where N is the population size and $\mu(x, X_0)$ is the probability of mutating from X_0 to x, according to e.g. a substitution or indel model (Sec. 0.4.1). Eqn. 7 combines the effects of mutation (via μ) with the effects of selection (via F) to produce a modified transition distribution describing sequence evolution over time in a population of organisms.

A key consequence of these evolutionary dynamics is their asymptotic behavior in the long-time

limit. Sella & Hirsh [230] show under general assumptions that the stationary distribution takes the form of a Boltzmann distribution with log fitness playing the role of energy,

$$P^{\tau}(x, X_0) \xrightarrow{\tau \to \infty} p^{\infty}(x) = \frac{1}{\mathcal{Z}} \exp(\beta F(x)), \tag{8}$$

where the inverse temperature $\beta = 2(N-1)$ depends on population size and $\mathcal{Z}$ is the normalization constant. Thus, given enough time, we expect the population to have a random fixed genotype $X \sim p^{\infty}(x)$, with the log probability of the sequence $\log p^{\infty}(x)$ proportional to the log fitness of the sequence $F(x)$. Estimating the stationary distribution p^{∞} from data is a key problem because it provides information about the underlying fitness landscape F.

The structure of the fitness landscape function F is of particular importance. We will primarily study the fitness landscapes of individual biomolecules. This can be justified using an additivity assumption, namely that $F(x) = F^{(m)}(x^{(m)}) + F^{(e)}(x^{(e)})$ where $F^{(m)}(x^{(m)})$ is the contribution of the molecule of interest $x^{(m)}$ (e.g. a particular protein encoded within the genome) to the overall fitness $F(x)$, and $F^{(e)}(x^{(e)})$ is the contribution of everything else in the genome. Under this assumption, the stationary distribution $p^{\infty}(x)$ factors as $p^{\infty}(x) = p^{\infty}(x^{(m)})p^{\infty}(x^{(e)})$, so that the molecule of interest $x^{(m)}$ is independent of the rest of the genome, and can be studied in isolation. A further assumption is that the fitness is additive within the molecule of interest, i.e. $F(x) = \sum_{j=1}^{|x|} F_j(x_j)$ (notationally, for the rest of the dissertation we will be focused on individual molecules, so we drop the superscript $^{(m)}$). For instance, if x is a protein, its biological activity may depend critically on the right kind of amino acid being in each position. In this case the stationary

distribution $p^{\infty}(x)$ is independent across positions, and is thus an instance of a "sitewise independent model". A more flexible assumption on F is that it can also depend on pairs of positions, i.e. $F(x) = \sum_{j=1}^{|x|} F_j(x_j) + \sum_{j=1}^{|x|} \sum_{j'>j}^{|x|} F_{jj'}(x_j, x_{j'})$. For instance, if amino acids at two sites interact in three-dimensional space, the function of the protein can depend on having a correct pair of amino acids at these positions, e.g. one positively and one negatively charged. In this case the stationary distribution $p^{\infty}(x)$ is the celebrated Potts model of proteins [168,110]. More complex fitness functions can arise, however, particularly when there is length variation.

Evolutionary dynamics are complex, and the basic Sella & Hirsh [230] model can break down when mutation rates are high, when there are asymmetries in mutational biases, when the population has substructure, when fitness changes as a function of time, etc..

0.4.3 Phylogeny

So far we have described models of sequence evolution at the level of individual organisms, and at the level of individual populations of organisms, all of the same species. We now turn to models of multiple populations, each corresponding to separate species. This necessitates understanding the effects of phylogeny, the history of species.

A *phylogenetic tree* describes the history of species formally. It consists of a directed and rooted full binary tree $\mathbf{H} = (V, E, T)$ with edges E and nodes V, along with time labels for the nodes, $T : V \to \mathbb{R}_+$ (Fig. 4.1A). Each node v corresponds to a particular species, with the sequence X_v fixed in the species' population. Each species derives from its ancestor as $X_v \sim P^{\Delta t}(x, X_{v_0})$, where X_{v_0} is the sequence of the parent node (the ancestor), v is the child node (the descendent),

and $\Delta t = T(v_0) - T(v_1)$ is the length of the edge between them (Fig. 4.1B). The evolutionary dynamics of individual populations $P^\tau(x, X_0)$ thus give rise to the evolutionary dynamics of multiple species through a branching process. A central challenge in phylogenetics is inferring the latent tree structure given only sequences from present-day species, i.e. the leaves of the phylogenetic tree.

Note that this basic model fails to take into account a number of important biological phenomena, including especially horizontal gene transfer. It also ignores situations where the definition of separate populations/species is not clear cut, as well as situations where the transition operator $P^\tau(x, X_0)$ varies across branches.

0.4.4 Conclusions

We have outlined a framework for models of molecular evolution, building up from the individual organism to the population to the multispecies level, taking into account mutations, fitness and phylogeny. At the broadest level, these models pair a description of sequence dynamics in terms of a Markov transition probability function – which is determined by mutation rates and has a stationary distribution that reflects the fitness landscape – with a description of species' history over time in terms of a binary branching tree. Attempting to take into account the full picture – mutations, fitness and phylogeny – typically leads to models that are both highly complex (requiring many parameters) and incomplete (ignoring the possibility of more complex mutational dynamics, population dynamics, etc.). A fundamental challenge is to instead construct models that simultaneously capture key phenomena of interest while remaining robust to biological complexity and scalable to large datasets. For instance, a standard approach to learning evolutionary histories is to use a transi-

tion operator that describes substitution mutations and ignores fitness; but ignoring fitness effects can distort phylogenetic tree estimates, and ignoring indels typically relies on preprocessing methods that violate the i.i.d. data assumption and get in the way of sequence prediction (Sec. 0.3.1; Chapters 1 and 4). A standard approach to learning fitness landscapes is to ignore phylogenetics and treat the data as coming from the stationary fitness distribution (Eqn. 8); but phylogenetic effects can distort fitness inferences arbitrarily, so these methods are not necessarily robust (Chapter 4).

0.5 Outline of the dissertation

The dissertation consists of five chapters organized into two parts. Here we briefly review the contribution of each chapter, placing them in the larger context of generative biological sequence statistics.

0.5.1 Part I: Box's loop for sequences

Box's loop (Fig. 1) is an idealized virtuous feedback loop of improved scientific understanding, in which probabilistic models are proposed, refined, and then applied. However, Box's loop has been challenging to implement in practice in the context of biological sequences. The purpose of Part I of this dissertation is to help remove existing barriers to Box's loop, developing powerful and scalable tools to build, infer, criticize, and design experiments based on generative biological sequence models.

Chapter 1, based on Weinstein & Marks [284], develops a new tool for building generative sequence models, a structured observation distribution which we call the "mutational emission" (MuE) dis-

tribution. Observation distributions (also called "error", "emission" or "output" distributions) are a ubiquitous tool in statistics and machine learning, and provide a systematic way of working with data in a particular space $\mathcal{X}$, with a particular form of variability. Given a covariate or a latent variable Z, we can model an observed datapoint X as $X \sim \text{Observation}(g(Z))$ where $\text{Observation}(\cdot)$ is the observation distribution and g is a function we can choose (a linear function, a deep neural network, etc.). For instance if X were count data, recording the number of times a rare event occurred, a standard choice of observation distribution would be the Poisson distribution. The MuE is an observation distribution for biological sequence data, where X is in the set of finite length strings $\mathcal{X}$ (Sec. 0.3.1). It explicitly accounts for mutational variability, in particular substitutions and indels (Sec. 0.4.1). Methodologically, MuE observation models are intended as an alternative to a ubiquitous preprocessing procedure – multiple sequence alignment (MSA) – that manipulates the data to have standardized length; MuE observation models can be interpreted similarly to MSA-based models (i.e. in terms of variation at conserved sites and indels), but have the virtue of providing valid predictions over unobserved or future sequences, enabling fully generative sequence modeling (Sec. 0.3.1). We develop fast variational inference strategies for MuE observation models (Sec. 0.2.3), taking advantage of parallel scan algorithms. MuE observation models and inference algorithms are now part of the Pyro probabilistic programming language, allowing them to be easily constructed and used in combination with other models for other kinds of data[23]. We apply MuE observation models to build a generative forecast of pathogen sequence evolution (Sec. 0.1.1), as well as detailed maps of immune receptor repertoires in individual patients (Sec. 0.1.2), and improved descriptions of disordered protein families.

Chapter 2, based on Amin et al. [12], develops a new generative sequence model, the Bayesian embedded autoregressive (BEAR) model. BEAR models can be used both for density estimation and for model criticism, in particular goodness-of-fit and two-sample testing (Sec. 0.2.1). They combine a nonparametric Bayesian Markov model with a structured prior, centered on the predictions of a parametric autoregressive model. We develop fast empirical Bayes inference algorithms for BEAR models, which take advantage of powerful database construction tools for biological sequences. These algorithms are scalable to whole genome and even metagenome datasets – terabytes of data or more – and we find BEAR models exhibit excellent predictive performance in both the small and large data regimes. We prove theoretically that BEAR models are asymptotically consistent nonparametric density estimators: their posterior converges to *any* data generating distribution p_0, in terms of Hellinger distance (Sec. 0.3.3), so long as p_0 is sub-exponential (Sec. 0.3.2). Since we can tractably compute the marginal likelihood of BEAR models, they can also be used for goodness-of-fit testing and two-sample testing, and we prove the asymptotic correctness of these tests as well. We demonstrate BEAR models on whole genome sequencing data from plants (whose genomes have notoriously complex structural variation), metagenomic data from patients with irritable bowel syndrome, and single cell RNA sequencing data from tumors. We apply BEAR two-sample tests to evaluate changes in the microbiome of patients before and after kidney transplantation, and to criticize simulator models of whole genome sequencing data.

Chapter 3, based on Weinstein et al. [282], turns to the problem of designing experiments based on generative sequence models, and in particular, synthesizing samples from models in the laboratory. A standard approach, which we term "Monte Carlo (MC) synthesis", is to draw samples

from the model computationally, and then synthesize these samples individually in the laboratory; this approach is typically limited by synthesis costs. We propose an alternative strategy, "variational synthesis", which relies on stochastic synthesis techniques, or biochemical methods that produce a diverse set of product molecules from single reactions rather than a pure product. Stochastic synthesis methods can produce a massive number of unique DNA molecules in a single test tube, vastly more than could be synthesized individually, but the molecules are randomized. We propose to (1) model the distribution over product molecules with another generative probabilistic model $q_\theta(x)$, with parameters θ corresponding to quantities over which the experimentalist has control, and then (2) choose the optimal θ^* such that $q_{\theta^*} \approx p$. Running the synthesis protocol in the laboratory with the optimized parameters θ^* will then produce a large number of approximate samples from $p(x)$. We model the distribution of product sequences in terms of underlying substitution and recombination mutations (Sec. 0.4.1); the full distribution $q_\theta(x)$ takes the form of a mixture model. We optimize θ by minimizing a KL divergence with the target model distribution, in particular $\theta^* = \text{argmin}_\theta \text{KL}(p \| q_\theta)$. Using integral probability metrics (Sec. 0.3.3), we provide theoretical guarantees that variational synthesis will produce large numbers of hits in downstream assays, as compared to MC synthesis. We further show that some, but not all, stochastic synthesis technologies can approximate arbitrary target distributions $p(x)$ arbitrarily well, where $p(x)$ is assumed to be bounded (Sec. 0.3.2). We then demonstrate a complete Box's loop pipeline: building models with the MuE observation distribution, criticizing models with BEAR two-sample tests, and designing experiments with variational synthesis. We show, using simulated assay functions trained on held-out sequence-to-function data, that using variational synthesis instead of MC synthesis can lead to

$> 400\times$ increase in the number of hits in example fluorescent protein and T cell receptor engineering problems (Sec. 0.1.2).

0.5.2 Part II: Using misspecified models for estimation and discovery

The goal of Box's loop is to produce an accurate probabilistic model of a given dataset. From this perspective, misspecified models are a major problem: they are "wrong", in the sense that they cannot accurately capture the underlying data generating distribution (Sec. 0.2.2). Misspecified models should therefore be avoided if possible, and used with caution if necessary, when modeling complex data. Although this attitude is something of a truism in probabilistic modeling, our efforts to replace heuristic and semi-heuristic biological sequence analysis methods with more rigorous statistical methods led to a number of examples where such conventional wisdom breaks down or is incomplete. We first present an example where misspecified models are a powerful tool for accurate estimation, even with infinite data. We then consider a situation where the scientific goal is not to produce an accurate model of the entire dataset, but rather only a piece of the dataset.

Chapter 4, based on Weinstein et al. [281], considers the problem of estimating fitness landscapes from evolutionary sequence data (Sec. 0.4.2). Existing methods ignore the effects of phylogeny, treating the dataset of sequences as if it consisted of i.i.d. samples from the stationary distribution p^∞ (Sec. 0.4.3). In particular, they proceed by fitting a parametric model $\mathcal{M} = \{p_\theta : \theta \in \Theta\}$ to data, and then using the log probability of a sequence under the inferred model as an estimate of its log fitness (applying Eqn. 8). We show that the effects of phylogeny can distort the data generating distribution p_0 away from the stationary distribution p^∞, and there are fundamental limits

on what we can learn about p^∞ given samples from p_0. Further, we show that when the assumption that there are no phylogenetic effects is violated, using a misspecified model can result in better estimates of p^∞ as compared to a well-specified model: the model distribution at the pseudo-true parameter, p_{θ_0}, may be closer to p^∞ than p_0 (Sec. 0.2.2). Applying the BEAR model to estimate p_0, we develop a hypothesis test to determine whether or not this effect holds in practice. Across over a hundred separate datasets, we show that using misspecified models results in systematically improved fitness estimation. Our results have implications for our ability to engineer new proteins and diagnose genetic disease, as well as for our understanding of the long term past and future of evolution (Sec. 0.1.3).

Finally, Chapter 5, based on Weinstein & Miller [285], turns to a more general statistical problem, motivated by common heuristic methods in biological sequence analysis and other fields of computational biology. Given a complex phenomena, scientists often proceed by developing working models for various special cases and subsets; thus, a natural question is where and when a given working model applies. We formalize this as the "data selection" problem: finding a lower-dimensional statistic (such as a subset of dimensions) that is well fit by a given parametric model of interest. In biological sequence statistics, for instance, there are a variety of heuristic methods for determining sets of subsequences that are well fit by a profile hidden Markov model [69]. Since the data selection problem has not been studied systematically before, we focus on the more standard statistical setting of continuous real vector data, rather than sequence data. We introduce a Bayesian approach to data selection, and study its asymptotic behavior, revealing that it quantifies a simple logic: we should apply our working model to explain as much of the data as it can, and no more. We then develop an

alternative to the fully Bayesian approach, with analogous asymptotic behavior, that is faster to compute. We demonstrate our method on single cell RNA sequencing datasets, determining where and when a simple biophysical model of gene expression actually applies. Our results set the stage for future work developing rigorous Bayesian data selection methods specifically for biological sequences.

1

A Structured Observation Distribution

Generative probabilistic modeling of biological sequences has widespread existing and potential application across biology and biomedicine, from evolutionary biology to epidemiology to protein design. Many standard sequence analysis methods preprocess data using a multiple sequence alignment (MSA) algorithm, one of the most widely used computational methods in all of science[270]. However, as we show in this article, training generative probabilistic models with MSA preprocess-

ing leads to statistical pathologies in the context of sequence prediction and forecasting. To address these problems, we propose a principled drop-in alternative to MSA preprocessing in the form of a structured observation distribution (the "MuE" distribution). We prove theoretically that the MuE distribution comprehensively generalizes popular methods for inferring biological sequence alignments, and provide a precise characterization of how such biological models have differed from natural language latent alignment models. We show empirically that models that use the MuE as an observation distribution outperform comparable methods across a variety of datasets, and apply MuE models to a novel problem for generative probabilistic sequence models: forecasting pathogen evolution.

This chapter presents work with Debora S. Marks, published at the International Conference on Machine Learning (2021)[284]. E.N.W. conceived the research, performed the research and wrote the paper; D.S.M. supervised the research at all stages.

1.1 Introduction

High-throughput sequencing is pervasive across biology and biomedicine, and critical to both past and ongoing discoveries and technological advancements. Analyzing large scale sequence data, making predictions about unobserved or future sequences, and generating new functional sequences, are major and growing challenges with relevance to epidemiology (predicting pathogen evolution), immunology (characterizing antibody repertoires), molecular evolution (mapping substructure within protein families), protein design, and many more subfields of biology and biomedicine. In

principal, generative probabilistic modeling enables (a) modular and uncertainty-aware data analysis, (b) formal mathematical statement of underlying assumptions, and (c) generation of new samples, which in the case of sequences can be synthesized and tested in the laboratory (taking advantage of recent rapid progress in high-throughput synthesis)[147,224]. However, although machine learning and statistics offer an extraordinary array of generative probabilistic models, extending existing methods to apply to biological sequences while accounting for domain-specific prior knowledge is nontrivial.

When analyzing biological sequence data, a standard approach is to preprocess the data before building any models by constructing a multiple sequence alignment (MSA). MSA algorithms are among the most widely used methods in all of science; according to a 2014 analysis, the 10th most cited scientific article of all time is an MSA algorithm, ahead of all other computational data analysis and statistics articles[270,256,255]. Recent major advances in machine learning and statistical methods for protein structure prediction, variant effect prediction for clinical genetics, protein design, epidemiological tracking, and more have continued to rely on MSAs[168,79,224,99]. Although MSAs are a powerful tool for understanding sequence evolution, in Section 1.4.1 of this article we show that employing MSAs as preprocessing introduces statistical pathologies in the context of generative sequence prediction and forecasting.

As a principled, drop-in alternative to MSA preprocessing, this article provides a structured observation distribution for biological sequences, the "mutational emission" ("MuE") distribution. Observation distributions are a common general-purpose technique for extending continuous-space models to other types of data, perhaps most familiar in the context of generalized linear models,

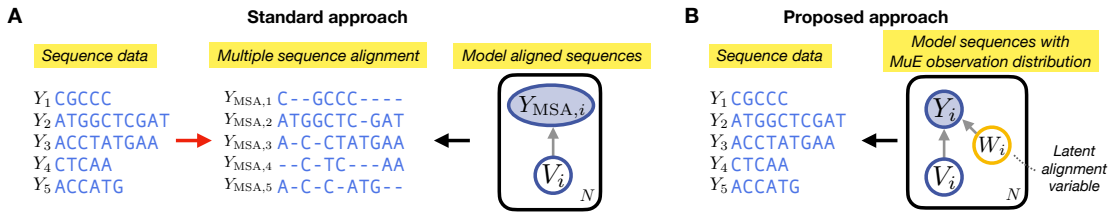


Figure 1.1: (A) A standard approach to building biological sequence models is to preprocess the data by constructing an MSA. (B) We propose modifying the model instead of the data using the MuE distribution.

where they are sometimes also referred to as "error", "emission", or "output" distributions. For instance, to predict count data, one might use a Poisson as an observation distribution, or to predict positive continuous data, one might use a Gamma. Good observation distributions account for both the support of the data and common forms of variability or noise in the data. For biological sequences, we propose using the MuE as an observation distribution. The MuE takes the form of a latent alignment model in which the regressor sequence can also be latent [52].*

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

*We will refer to biological alignments (diagrammatic representations of relatedness between sequences) as "multiple sequence alignments" [67]. We will refer to machine learning alignments (latent variables which indicate which positions in one sequence generate which positions in another sequence) as "latent alignments" [52].

add a MuE.

1.2 Method

1.2.1 Background: MSA Preprocessing

MSA algorithms are applied to families of evolutionarily related biological sequences (proteins, RNA or DNA) in order to infer sites in each sequence that are likely to be related to one another, meaning that they descend from a common ancestor. MSAs can be used as the basis for extrapolation: for instance, knowledge about one region in one sequence can be used to make guesses about related regions in related sequences. MSAs can also be used to understand biological function: for instance, if particular amino acids at particular sites are highly conserved across sequences, it may be evidence that they are crucial to biological function. Generative probabilistic models of MSAs have seen widespread success on these and many other tasks, including predicting the clinical impacts of genetic mutations, inferring three-dimensional protein and RNA structure, and designing new proteins[79,168,280,224]. We next briefly describe how such MSA-based models are built, as well as their advantages and flaws. In Section 1.2.2 we introduce our alternative, MuE observation models, which directly generate sequences rather than MSAs. MuE observation models infer related sites but also simultaneously (1) account for uncertainty in which sites are related, (2) allow rigorous model evaluation and (3) enable prediction and forecasting of sequences.

Let $\{Y_1, \ldots, Y_N\}$ be a dataset of N sequences, which may each be different in length, and let $\mathcal{B}$ denote the alphabet (e.g. $\mathcal{B} = \{A, T, G, C\}$ for DNA). MSA algorithms convert the sequence

dataset into an N by J matrix, an MSA, adding gap symbols "$-$" such that sites in the same matrix column are those inferred to be related (Figure 1.1A). Mathematically, MSA algorithms can be summarized as nonlinear functions f_{MSA} that take in datasets of sequences and return processed datasets, $\{Y_{\text{MSA},1}, \ldots Y_{\text{MSA},N}\} := f_{\text{MSA}}(\{Y_1, \ldots, Y_N\})$; for each $i \in \{1, \ldots, N\}$, we have $Y_{\text{MSA},i} \in (\mathcal{B} \cup \{-\})^J$. Note J itself will depend on the input dataset.

Preprocessing sequence data by constructing an MSA is useful in that it (1) converts the data into a matrix, and (2) adjusts for common sources of variability in biological sequence data, in particular insertion and deletion mutations. MSA preprocessing makes building statistical models of sequences more straightforward. For instance, starting from an arbitrary model p_θ that generates continuous matrices $V_i \in \mathbb{R}^{J \times (B+1)}$, where $B := |\mathcal{B}|$, one general strategy is to employ a softmax linker function and a categorical observation distribution ($\text{softmax}(V_i)_j := \exp(V_{i,j,b}) / \sum_{b'} \exp(V_{i,j,b'})$ for $j \in \{1, \ldots, J\}$). The complete approach is (Figure 1.1A),

$$\begin{aligned} &\text{Preprocess: } \{Y_{\text{MSA},1}, \ldots Y_{\text{MSA},N}\} := f_{\text{MSA}}(\{Y_1, \ldots, Y_N\}), \\ &\text{Model: } V_i \sim p_\theta \\ &Y_{\text{MSA},i} \sim \text{Categorical}(X_i := \text{softmax}(V_i)). \end{aligned} \tag{1.1}$$

By allowing arbitrary p_θ, this method enables, for example, the application of generative image models (such as variational autoencoders) to biological sequence data[215]. However, as we describe in depth in Section 1.4.1, MSA preprocessing introduces substantial problems: each row of the output matrix $Y_{\text{MSA},i}$ depends via f_{MSA} on the entire input dataset $\{Y_1, \ldots, Y_N\}$ and we cannot know

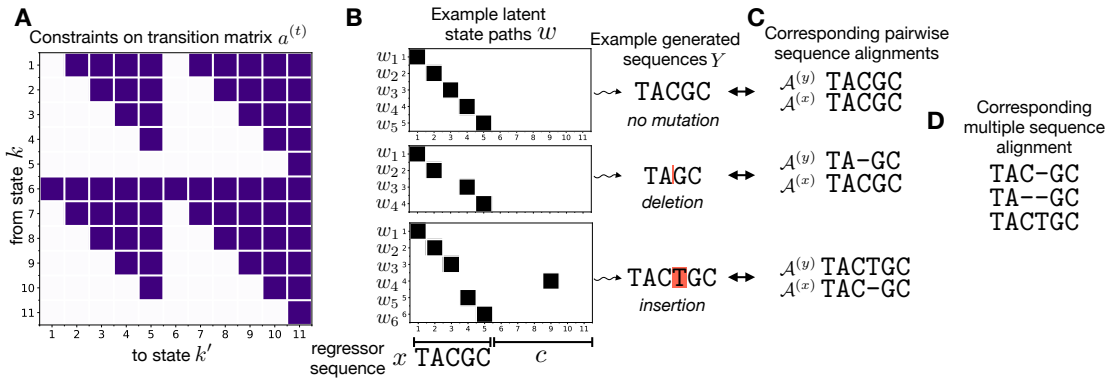


Figure 1.2: (A) Condition 1.2.2 allows only the positions of $a^{(t)}$ in dark purple to be non-zero. (B) Example latent state paths w taken by the Markov model in the MuE, and sequences Y that they can generate, given x is a one-hot encoding of the DNA sequence TACGC. Rows correspond to positions $1, \ldots, L$, columns correspond to latent states $1, \ldots, K$. (C) w defines a pairwise alignment between X and Y via Definition 1.4.3. (D) The collection of w values describe a multiple sequence alignment of the generated sequences Y (Section 1.4.2).

ahead of time how future raw data Y_{N+1} will change preprocessed past data $Y_{\text{MSA},i\leq N}$. This makes likelihood-based model evaluation on newly observed or heldout data ill-defined.

1.2.2 The Mutational Emission Distribution

As a drop-in alternative to MSA preprocessing, we introduce the "mutational emission" ("MuE") distribution. The MuE can be used in place of the Categorical observation distribution in Equation 1.1,

$$\begin{aligned} &\text{Model: } V_i \sim p_\theta \\ &Y_i \sim \text{MuE}(X_i := \text{softmax}(V_i), c, \ell, a^{(0)}, a^{(t)}), \end{aligned} \tag{1.2}$$

where c, ℓ, $a^{(0)}$, and $a^{(t)}$ are parameters of the MuE, and $V_i \in \mathbb{R}^{M \times D}$ where M and D are hyperparameters rather than dimensions of the input data. The MuE avoids the pathologies of MSA preprocessing by directly generating complete, variable-length sequences (Figure 1.1B). We refer generically to models that use a MuE observation distribution, such as Equation 1.2, as "MuE observation" models. (See Figure A.1 for a diagram of MuE observation models and Table A.1 for a notation reference.) In the limiting case where X_i is a one-hot encoding of a sequence (i.e. $X_{i,m,d} \in \{0, 1\}$ and $\sum_d X_{i,m,d} = 1$), the MuE can be interpreted biologically as generating a mutant Y_i of the "ancestral" sequence X_i, with some probability of insertion and deletion mutations (controlled by c, $a^{(0)}$, and $a^{(t)}$) and of substitution mutations (controlled by ℓ) (Section 1.2.3). A latent variable W_i in the MuE determines which positions in the regressor X_i – intuitively, which sites in the "ancestral" sequence – generate which positions in Y_i, and can be interpreted as defining a pairwise alignment between X_i and Y_i. The latent variables $W_1, \ldots, W_N$ define a multiple sequence alignment of the dataset $Y_1, \ldots, Y_N$ (Section 1.4.2). Intuitively, the MuE "adds in", through a generative process, the same mutations that MSA algorithms are intended to "filter out" of the data via preprocessing.

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

Definition 1.2.1 (MuE)**.** $\text{MuE}(x, c, \ell, a^{(0)}, a^{(t)})$ *is an HMM with* $K = 2M + 1$ *latent states. The initial probability of each latent state is given by* $a^{(0)} \in \Delta_K$*, the latent state transition matrix is*

$a^{(t)} \in (\Delta_K)^K$, *and the emission matrix is* $\tilde{x} \in (\Delta_D)^K$. *The matrices have block structure*

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

where $x \in (\Delta_D)^M$, $c \in (\Delta_D)^{M+1}$, $\ell \in (\Delta_B)^D$, $A^{(1,1)} \in \mathbb{R}^{M \times M}$, *and* $A^{(2,2)} \in \mathbb{R}^{(M+1)\times(M+1)}$. *The transition matrix must satisfy Condition 1.2.2.*

Condition 1.2.2 (Biological latent alignments)**.** *Entries of* $A^{(1,1)}$, $A^{(1,2)}$, $A^{(2,1)}$ *and* $A^{(2,2)}$ *below the main diagonal must be zero. Entries of* $A^{(1,1)}$ *and* $A^{(1,2)}$ *on the main diagonal must also be zero.*

Condition 1.2.2, an upper triangular restriction, is illustrated in Figure 1.2A and justified in depth in Section 1.4.2. We use w to denote a latent state path taken by the HMM, while W_i denotes the specific latent state path taken when generating sequence Y_i given X_i following $Y_i \sim \text{MuE}(X_i, c, \ell, a^{(0)}, a^{(t)})$.

1.2.3 Biological Interpretation of the MuE

To describe the biological interpretation of the MuE and its parameters, we consider examples of different latent paths $w = (w_1, \ldots, w_L)$ through state space and the sequences $Y \sim p_{\text{MuE}}(y|x, w)$ that these paths will generate (Figure 1.2B). Assume to start that $D = B$ and $\ell = I_B$, where I_B is the $B \times B$ identity matrix, and consider the limiting case where x is a one-hot encoding of a sequence (in Figure 1.2B, the DNA sequence `TACGC`). We consider three example w values:

1. $w = (1, 2, \ldots, M)$ (no mutation). The generated Y will be an exact copy of x, i.e. $Y = x$ if Y is represented as a one-hot encoding (Figure 1.2B top).

2. $w = (1, \ldots, m-1, m+1, \ldots, M)$ (deletion). The generated Y will be missing the mth letter of x, i.e. $Y = (x_1, \ldots, x_{m-1}, x_{m+1}, \ldots, x_M)$ (Figure 1.2B middle).

3. $w = (1, \ldots, m, M+m+1, m+1, \ldots, M)$ (insertion). The generated Y will have an additional letter inserted after the mth letter of x, with a probability over letters determined by c_{m+1}, i.e. $Y = (x_1, \ldots, x_m, S, x_{m+1}, \ldots, x_M)$ where $S \sim \text{Categorical}(c_{m+1})$ (Figure 1.2B bottom).

Condition 1.2.2 guarantees that the states $k \in \{1, \ldots, M\}$ corresponding to x are each visited at most once and in sequential order. Paths such as $\{1, \ldots, m, m, \ldots, M\}$ (repeat) and $\{1, \ldots, m+1, m, \ldots, M\}$ (backtracking) are not allowed under Condition 1.2.2. More general matrices $\ell \in (\Delta_B)^D$ allow for substitution mutations, with the probability of converting from letter d to letter b given by $\ell_{d,b}$. For example, if $w = (1, \ldots, M)$, then $Y \sim \text{Categorical}(x \cdot \ell)$, that is Y is a mutant of x with substitution probabilities determined by ℓ and no insertion or deletion mutations.

MuE observation models directly generalize models that use MSA preprocessing in the special case where the dataset sequences are all the same length and the MSA algorithm does not add any gap symbols (that is, when $f_{\text{MSA}}(\cdot)$ is the identity). Assume $D = B$, and consider the "no mutation limit" where $\ell = I_B$, $a_1^{(0)} = 1$, and $A_{m,m+1}^{(1,1)} = 1$ for all $m \in \{1, \ldots, M-1\}$. In this case we find, for samples Y of length M, that $Y \sim \text{MuE}(x, c, \ell, a^{(0)}, a^{(t)})$ simplifies to $Y \sim \text{Categorical}(x)$. Thus Equation 1.2 and Equation 1.1 become equivalent. In practice, we typically select priors on

the MuE to favor the no mutation limit, since it serves as a null hypothesis.

1.2.4 Inference

The marginal likelihood of the MuE with the latent state variable of the HMM integrated out, $p_{\text{MuE}}(y|x, c, \ell, a^{(0)}, a^{(t)})$, is analytically tractable via the HMM forward algorithm and differentiable. The standard forward algorithm requires $\mathcal{O}(L)$ sequential matrix multiplications, where L is the length of the sequence (typically a few hundred amino acids in our setting), but it can also be parallelized to achieve $\mathcal{O}(\log L)$ time[227,223]. Using the MuE marginal likelihood allows inference with automatic differentiation variational inference, stochastic gradient MCMC, and related scalable approximate Bayesian inference algorithms (Section A.4.1)[147,286]. We have made available an implementation of the MuE distribution as part of the probabilistic programming language Pyro, making it straightforward to explore different MuE observation models and inference methods (`https://docs.pyro.ai/en/dev/contrib.mue.html`, Section A.4.2)[23].

1.3 Related work

Methods that use MSA preprocessing. MSA preprocessing is widely used as a starting point for biological sequence data analysis, perhaps most commonly in combination with other non-probabilistic analysis methods. One very common class of probabilistic methods that nearly always use MSA preprocessing is phylogenetic models, which are central to evolutionary biology and genomic epidemiology, and widely used in nearly every other area of biology[99,75]. Another is fitness models, including Potts models and variational autoencoder models, which are used to infer the

structure of proteins and RNA, predict the functional effects of clinical variants, design new proteins, etc. [168,110,79,224].

Standard methods that avoid MSA preprocessing. Although MSA preprocessing is problematic from the perspective of probabilistic modeling, the use of probabilistic models to infer multiple sequence alignments – that is, in order to *accomplish* the preprocessing – is standard. Perhaps the most widely used such method is the profile HMM, which, besides being used to infer multiple sequence alignments, is also at the core of modern sequence database search methods and is used to define sequence families, among many other applications [67,130,72]. In Section 1.4.2 we show that the MuE distribution generalizes a variety of popular methods including the profile HMM. While connections between various methods have been described before, the generalization offered by the MuE is both unified and comprehensive, delimiting the extent of the model class [107]. Note also that some of these models can be trained by interpreting an MSA as a point estimate of the latent alignment variable; this is distinct from the more common usage of MSA preprocessing described in Section 1.4.1 and is not subject to the same pathologies. The most closely related method to MuE observation models is the hidden Potts model [287]; we go further by providing a generalized approach to building and inferring similar models.

Natural language processing methods There has been intense recent interest in applying advances from natural language processing to biological sequences [217,235,9]. The MuE is a type of latent alignment model, a key model class in natural language processing; Deng et al. [52] detail the close relationship between latent alignment and popular attention network methods. MuE observation models differ from standard latent alignment models in that (1) rather than regress on an observed

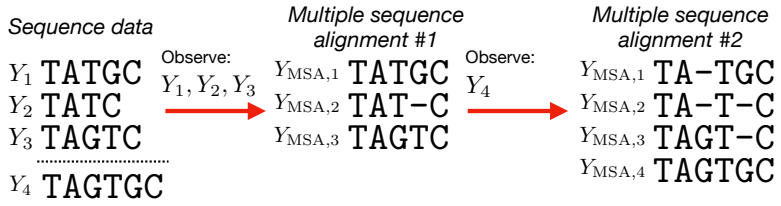


Figure 1.3: The multiple sequence alignment of the initial dataset Y_1, Y_2 and Y_3 can change as more data, Y_4, is added.

sequence (e.g. a sentence in a language to be translated), the model regresses on a latent sequence X_i, and (2) the MuE is structured such that its latent alignment variable is interpretable as a *biological* alignment, not an alignment in the more generic sense used in natural language processing (Sections 1.4.2 and 1.4.3). Note that while the MuE itself is a relatively simple latent alignment model (an HMM), complex neural networks can be used to generate the latent sequence X_i; from a deep learning perspective, the MuE can be thought of as a biologically interpretable final layer.

1.4 Theory

1.4.1 Pathologies in MSA Preprocessing

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

Assumption 1.4.1 (I.i.d. data and model)**.** *Let $p_0(x)$ be a probability distribution defined over a space $\mathcal{X}$, i.e. $p_0(x) \in \mathcal{P}(\mathcal{X})$ where $\mathcal{P}(\mathcal{X})$ is the set of all probability distributions over $\mathcal{X}$. We (1)*

[†]Note that phylogenetic models, although not usually represented as i.i.d., are typically exchangeable and so possess an i.i.d. representation by de Finetti's theorem [283].

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

Now consider models that use MSA preprocessing and take the following form, of which Equation 1.1 is a special case:

$$\text{Preprocess: } \{Y_{\text{MSA},1}, \ldots, Y_{\text{MSA},N}\} := f_{\text{MSA}}(\{Y_1, \ldots, Y_N\}),$$
$$\text{Model: } Y_{\text{MSA},i} \overset{iid}{\sim} p(y_{\text{MSA}}),$$

where $p(y_{\text{MSA}}) \in \mathcal{P}((\mathcal{B} \cup \{-\})^J)$. If we attempt to employ Assumption 1.4.1 to describe the preprocessed data $Y_{\text{MSA},1}, \ldots, Y_{\text{MSA},N}$ we see that it is violated. Part 1 of Assumption 1.4.1 fails because the preprocessed data cannot consist of independent observations: if a datapoint Y_{N+1} is added to the dataset, then past data, i.e. $Y_{\text{MSA},1}, \ldots, Y_{\text{MSA},N}$, can be altered (Figure 1.3). For instance, the new sequence may provide additional evidence to the MSA algorithm that sites in previously observed sequences are related to one another. Part 2 of Assumption 1.4.1 fails because the model is not defined over a space that encompasses future data: if a datapoint Y_{N+1} is added to the dataset, the value of J may change (Figure 1.3). For instance, the new sequence might be longer than any seen before. These failures occur on real sequence datasets, for typical values of N (Figure A.2). Practically, the fact that MSA models violate Assumption 1.4.1 makes rigorous likelihood-based evaluation of their generalization capacity untrustworthy. If we do not know what space future data lives in, or how past data will be altered with future measurements, it is hard to

trust that the average log likelihood of our model on a held out test set is genuinely reflective of future model performance. More technically, the violation of Assumption 1.4.1 causes standard justifications for the use of Bayes factors, heldout likelihood, prequential evaluation, etc. to fail, see e.g. Dawid [49], Vapnik [271], Dawid [50].

Using MSA preprocessing also fails to account for uncertainty in the alignment [292,261]. The goal of an MSA algorithm is to infer related sites among a set of sequences, but the resulting MSA is only a point estimate of this quantity.

1.4.2 Inferring Alignments

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

Definition 1.4.2 (Biological pairwise alignment)**.** *Let X and Y be sequences of length M and L respectively. A pairwise alignment $\mathcal{A}$ of X and Y with J columns is a matrix $[\mathcal{A}^{(x)}, \mathcal{A}^{(y)}]^\top$, where $\mathcal{A}^{(x)} \in (\mathcal{B} \cup \{-\})^J$ is a column vector of length J consisting of the letters of X, in order, and interspersed with gap symbols; similarly for $\mathcal{A}^{(y)}$. The alignment $\mathcal{A}$ must satisfy the condition that for every $j \in \{1, \ldots, J\}$ either $\mathcal{A}^{(x)}_j \in \mathcal{B}$ or $\mathcal{A}^{(y)}_j \in \mathcal{B}$ or both.*

Let j_l be the column of the alignment $\mathcal{A}$ in which the lth letter of Y falls, i.e. $\mathcal{A}^{(y)}_{j_l} = Y_l$ for $l \in \{1, \ldots, L\}$. Let g_l indicate whether the column j_l in $\mathcal{A}$ contains a gap, i.e. $g_l := \mathbb{I}(\mathcal{A}^{(x)}_{j_l} = \text{-})$, where $\mathbb{I}(\cdot)$ is the indicator function which takes value 1 when the expression is true and 0 otherwise. Given X and Y, the sets $\{j_1, \ldots, j_L\}$ and $\{g_1, \ldots, g_L\}$ together uniquely define an alignment $\mathcal{A}$ (Remark A.2.1). We can define a map from the latent state path W to a pairwise alignment $\mathcal{A}$ of X and Y.

Definition 1.4.3 (From latent states to biological alignments)**.** *Given $W \sim p_{\text{MuE}}(w|X, Y)$, let $g_l = \mathbb{I}(W_l > M)$ and $j_l = W_l - M g_l + \sum_{l'=1}^{l-1} g_{l'}$, for $l \in \{1, \ldots, L\}$. Note that this map is invertible.*

Under this definition, when $g_l = 0$, the letter Y_l is generated based on a letter X_{W_l} in the MuE, and Y_l and X_m are placed in the same column of the pairwise alignment $\mathcal{A}$; when $g_l = 1$, however, Y_l does not depend on X at all (it depends on c instead) and $\mathcal{A}^{(x)}_{j_l}$ has the gap symbol (Figure 1.2C).

A zoo of probabilistic and non-probabilistic methods have been proposed for inferring biological sequence alignments from data. Here we show that many of the most widely used methods can be unified as special case examples of the MuE which use Definition 1.4.3 to convert from W to $\mathcal{A}$.[‡]

Proposition 1.4.4 (Unified)**.** *For different choices of parameters c, ℓ, $a^{(0)}$, and $a^{(t)}$, (1) the Thorne-Kishino-Felsenstein model[257], (2) the profile HMM, and (3) the conditional distribution of a sequence Y given a sequence X under the pair HMM[67] are all special cases of the distribution*

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

$\mathrm{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$*, with a state-specific probability of the Markov chain terminating at each step. For another choice of parameters, the maximum a posteriori estimator* $\hat{w} := \mathrm{argmax}_w p_{\mathrm{MuE}}(Y|X, w)$ *corresponds to the Needleman-Wunsch alignment.*

See Section A.2.2 for a proof. In the context of the profile HMM, point estimates of the latent alignment variables $W_1, \ldots, W_N$ associated with each observed sequence $Y_1, \ldots, Y_N$ are used to construct a multiple sequence alignment of the dataset by effectively merging pairwise alignments; sites in each Y_i generated by the same position in X are considered related, and placed in the same column. The same logic and algorithm can be applied to MuE observation models to define an MSA based on $W_1, \ldots, W_N$ (Figure 1.2D; Section A.2.3).

The MuE offers not only a unified but also a comprehensive framework in the sense that HMMs which fail to satisfy Constraint 1.2.2 cannot be interpreted, using Definition 1.4.3, as biological alignments (proof in Section A.2.4):

Proposition 1.4.5 (Comprehensive)**.** *Consider the setup of Definition 1.4.3 and assume each latent state* $k \in \{1, \ldots, K\}$ *of the MuE is Markov accessible under* $a^{(0)}$ *and* $a^{(t)}$ *(meaning that it can be reached with non-zero probability). Condition 1.2.2 is both necessary and sufficient to guarantee that with probability 1,* W *defines a valid pairwise alignment of* X *and* Y *via Definition 1.4.3.*

1.4.3 Comparison to Natural Language Models

Latent alignment models are used in natural language processing, often in combination with hard attention methods for inference[52]. We can compare the MuE directly with a classic latent alignment

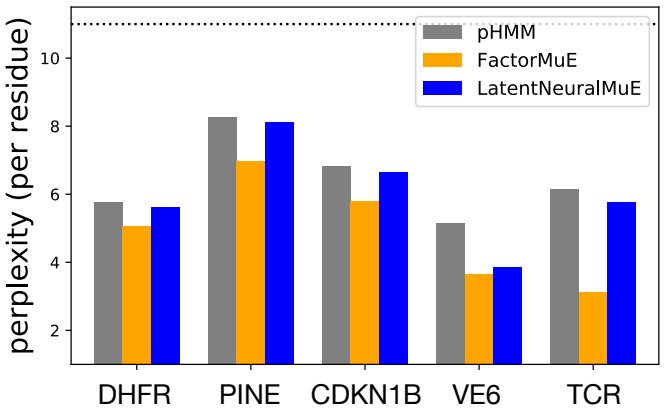


Figure 1.4: Predictive performance on a randomly heldout test set. Dotted line marks theoretically expected performance of the substitution matrix BLOSUM62 as a reference point (Section A.5).

model for statistical translation. The Vogel et al.[276] model takes the form of a MuE model where X and Y are sentences in different languages, except that Condition 1.2.2 is violated (Section A.2.5). As a result latent alignments are allowed to "double back" and rearrange the ordering of words in the regressor sentence X to generate Y.

1.5 Experiments

1.5.1 Predictive Performance

We have seen that models that use MSA preprocessing cannot be rigorously evaluated for their ability to predict sequences. In this section we empirically compare the predictive performance of MuE observation models to a standard model that possesses the same latent alignment structure, the profile HMM (pHMM) (Proposition 1.4.4).

Survey We started by examining five datasets of related protein sequences, ranging in size from

Table 1.1: Heldout perplexity on patient immune repertoire samples (each with 6,000 to 20,000 sequences). MS: multiple sclerosis. HC: healthy control. HC 1 consists of B cell receptors, the rest T cell receptors.

Dataset	HC 1	HC 2	HC 3	MS 1	MS 2	MS 3
pHMM	4.29	3.59	3.56	3.59	3.47	3.54
ICAMuE	**2.87**	**2.33**	**2.34**	**2.45**	**2.19**	**2.26**

1,000 to 10,000 sequences (Section A.6.1). Four were taken from non-redundant sequence databases: sequences similar to dihydrofolate reductase (DHFR), serine recombinase (PINE), cyclin dependent kinase inhibitor 1B (CDKN1B) and the human papillomavirus E6 protein (VE6)[110,261,254]. The fifth dataset consisted of human T cell receptor (TCR) sequences from a healthy donor, obtained using single cell sequencing.

We extended probabilistic PCA and VAE models using the MuE observation distribution; we refer to these models as "FactorMuE" and "LatentNeuralMuE" respectively (model architectures are detailed in Section A.3). We used stochastic variational inference, estimating the ELBO gradient using automatic differentiation, the reparameterization trick, and an inference network, and optimizing with Adam[147,139,213,138]. We evaluated model performance on a randomly held out 10% of sequences, quantified in terms of per residue (that is, per letter) perplexity (Section A.5). The results show that FactorMuE models offer a consistent improvement over the standard pHMM model in every dataset, with an average change in perplexity of -1.50 and log Bayes factor $> 10^3$ across all datasets (Figure 1.4; Section A.6.1). 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.

Patient immune repertoires We next explored further the application of MuE observation models to patient immune repertoire sequencing data, including both B and T cell receptors, taken from patients with autoimmune disease (multiple sclerosis) and healthy controls (Section A.6.2)[207]. Understanding immune receptor repertoires is of crucial biomedical importance, but MSAs are considered highly untrustworthy when applied to this kind of data (see e.g. Figure A.2). We extended another continuous model, an independent component analysis (ICA) model, with a MuE observation distribution ("ICAMuE"; Section A.3.4). On a heldout 20% of data we find substantial improvements in perplexity over the pHMM across all six datasets (Table 1.1).

Disordered proteins Roughly ∼50% of the human proteome contains regions classified as disordered, but common bioinformatic pipelines are often considered highly untrustworthy when applied to disordered proteins because of uncertain MSAs. We examined 56 datasets, each consisting of sequences evolutionarily related to a disordered region of a human protein, that had been discarded in an MSA-based sequence modeling study[261]. The study had sought in part to determine whether epistatic correlation occurred between amino acids at aligned sites (columns of the MSA), but was stymied in these particular datasets by highly uncertain MSAs. In a pHMM, conditional on a latent alignment W_i, the probability of observing a particular amino acid at a particular position in Y_i is independent of all other positions. In MuE observation models such as the FactorMuE, LatentNeuralMuE and ICAMuE, however, p_θ induces correlation between positions in Y_i conditional on W_i[215]. To infer whether there is indeed epistatic correlation in a dataset, therefore, we can perform model selection, comparing a MuE observation model and a pHMM. Note that our approximate Bayesian inference procedure (for both models) integrates over all possible latent

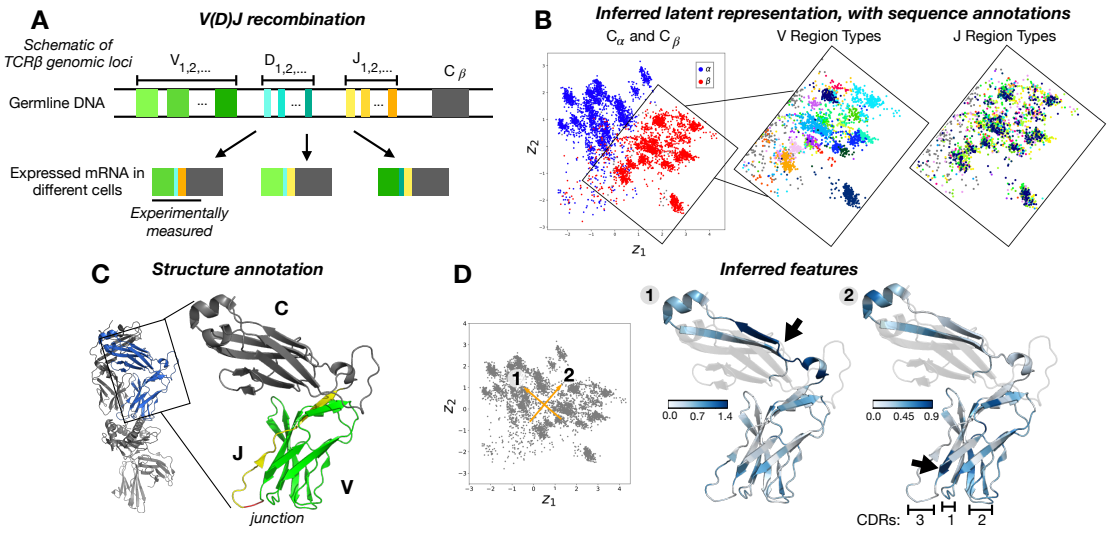


Figure 1.5: (A) Illustration of the TCRβ genomic locus; the TCRα locus is analogous, with C_α in place of C_β and no D segments (based on Abbas et al. [5], Figure 8.7). (B) Inferred latent space representation of the TCR dataset, colored according to supervised annotations. Left: C_α and C_β chains. Middle: V types, $V_2, \ldots V_{30}$ (detailed legend in Figure A.7). Right: J subtypes, $J_{1\text{-}1,\ldots,2\text{-}7}$ (detailed legend in Figure A.7). (C) V (green), J (yellow) and constant C (gray) regions of the TCRβ chain in the reference structure PDB:2BNR, as well as V-J junction nucleotides (red) (Figure A.7). (D) Projections ν of latent space vectors (left, in orange) into sequence space. Transparent areas correspond to the portion of the sequence that is not measured in the experiment. Arrows indicate peaks in ν.

alignments, and that the pHMM is nested inside the MuE observation models in the sense of nested model selection [50]. We found that on 19 datasets an ICAMuE outperformed a pHMM at predicting a heldout 20% of sequences, finding evidence of epistatic correlation despite high alignment uncertainty; among these 19 datasets, the median perplexity decrease was 1.3 (Table A.2, Section A.6.3).

1.5.2 Learning Complex Biology

We examined further what the FactorMuE model had learned from a dataset of TCR sequences. T cell receptors are made up of two separate amino acid chains, α and β, which each develop accord-

ing to a complex process of genome rearrangement termed V(D)J recombination, in which different V, D and J segments in the genome are, with some randomness and additional mutations, joined together with a constant region to produce a complete sequence (Figure 1.5A). We cross-referenced the latent representations of each sequence recorded in the dataset against supervised annotations of its segment types (Section A.7). We found that the latent space is divided evenly in two, with one side containing TCRα sequences and one side TCRβ sequences (Figure 1.5B left). Each side contains clusters, which correspond with the type of V segment found in each TCR sequence (Figure 1.5B middle). The shorter J segments are found uniformly distributed across their corresponding α or β half, reflecting their ability to recombine with different V segments (Figure 1.5B right). See Section A.7 for further results.

We next examined features learned by the FactorMuE model. In MuE observation models, we can separate out variation at conserved positions from variation produced by insertions and deletions by holding the latent alignment variable W_i fixed. In particular, we calculated

$$\nu_l := \left[\sum_{b=1}^{B} \left(\mathbb{E}[Y_{l,b} | \hat{w}_{\text{ref}}, z_1] - \mathbb{E}[Y_{l,b} | \hat{w}_{\text{ref}}, z_0] \right)^2 \right]^{1/2} \tag{1.3}$$

where the expectation is with respect to the variational approximation to the posterior, z_0 and z_1 are the head and tail of a vector in the latent space, $\hat{w}_{\text{ref}}$ is the maximum *a posteriori* estimate of W_{ref} based on a reference sequence Y_{ref}, and $l \in \{1, \ldots, L_{\text{ref}}\}$ where L_{ref} is the length of Y_{ref}. We plotted the vector ν on a TCR crystal structure for the reference sequence, and compared to a supervised annotation of the constant, V, D and J segments of the reference sequence (Figure 1.5CD). Consistent with the annotation of the latent representation, the vector normal to the hyperplane separating

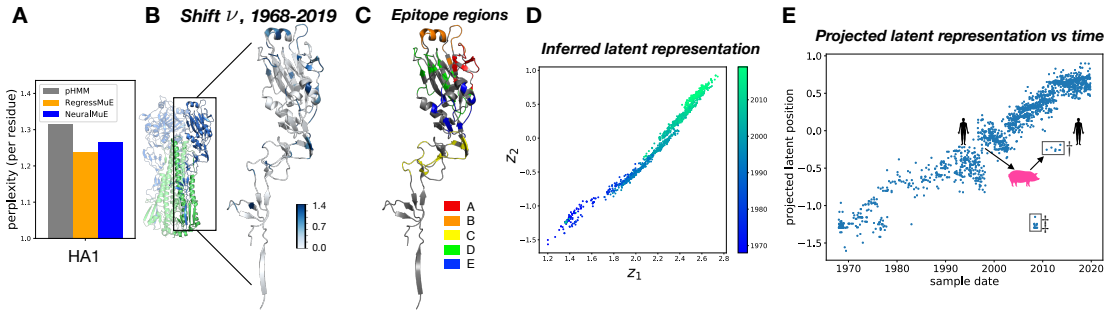


Figure 1.6: (A) Predictive performance measured by heldout per residue perplexity; models are trained on data from 1968-2013, tested on 2014-2020. (B) Magnitude of the shift in amino acid preference over time ν, for the RegressMuE, projected onto a reference HA1 structure (PDB:4O5N). The full hemagglutinin protein is shown on the left. (C) Classical epitope regions of the HA1 protein. (D) Inferred latent representation from a FactorMuE model, with sequences colored by the time at which the sample was collected (Section A.8). (E) Y-axis: orthogonal projection of the latent representation of each sequence onto the least squares fit line relating z_1 and z_2. X-axis: time at which each sample was collected. Two clusters of outliers are marked by † and ‡.

TCRα from TCRβ chains in the latent space (vector 1 in Figure 1.5D) primarily alters the sequence of the constant region, while the orthogonal vector (vector 2 in Figure 1.5D) primarily determines the sequence of the V segment. Along vector 2, the region of largest variation (the largest peak in ν_l) was the buried C-terminal end of the V segment, corresponding to the start of the CDR3 region, the key specificity-determining region of the receptor. Interestingly, even along vector 1 we observe high values of ν_l in the V segment, suggesting that there are systematic and heterogeneous differences between the V segment sequence distribution used in TCRα chains and in TCRβ chains (see Section A.7 for further analysis).

1.5.3 Evolutionary Forecasting

We explored a novel application of generative probabilistic sequence models, evolutionary forecasting, which takes advantage of the capacity of MuE observation models to predict future sequences. Influenza A is responsible for an estimated 500,000 deaths a year and is an ongoing pandemic threat [124]. It is also a model organism for understanding the dynamics of rapidly evolving pathogens, and forecasting its evolution is crucial in preparing vaccines and designing therapeutics [163,150]. Previous forecasting methods have focused on predicting the relative fitness of existing strains in future years [163,31], or the antigenic properties of newly emerged strains [188,103]. We instead predict the full amino acid sequence of the HA1 protein, the primary site of interaction with the immune system [288]. 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 A.8) [236]. Insertions and deletions are considered rare, though not absent, in patient samples, so this dataset also offers an opportunity to evaluate MuE observation models in a distinct regime from that considered previously in Section 1.5.1.

As a benchmark we again used the pHMM, which can capture the observation that there exist key highly variable sites in the HA1 protein, an underlying motivation behind previous prediction methods such as Bush et al. [31]. We then incorporated sequence collection time as a covariate in new MuE observation models, using a linear regression model ("RegressMuE") and a neural network ("NeuralMuE") with MuE observation distributions (Section A.3). The pHMM achieves a per

residue perplexity of 1.32 and the RegressMuE improves this to 1.24 (log Bayes factor $> 10^3$; Figure 1.6A). This per residue perplexity difference corresponds to a factor of $\sim 10^{10}$ improvement in per sequence perplexity. The NeuralMuE has similar per residue perplexity (1.26) to the RegressMuE.

Next we investigated in detail what the model can tell us about how HA1 proteins have changed over time. We computed the magnitude of the shift in amino acid preference from 1968 to 2019 inferred by the model, with the latent MuE alignment variable kept fixed (quantified as ν_l, defined analogously to Equation 1.3 with times t_0 and t_1 replacing latent representations z_0 and z_1) (Figure 1.6B; Section A.8). We found that sites with a large shift are often associated with antigenicity, consistent with the hypothesis that immune evasion is a key driver of influenza evolution. Residues that make up the classical epitope regions A-E of influenza show significantly larger shifts as compared to residues outside these regions (mean ν_l of 0.54 in epitopes A-E versus 0.09 in non-epitope sites, one sided Mann-Whitney U test $p < 10^{-18}$; Figures 1.6C and A.12)[288,184]. The same observation holds for residues identified as key determinants of immune escape in recent high-throughput mutational antigenic profiling experiments (mean ν_l of 0.80 in sites with antigenic selection versus 0.24 elsewhere, one sided Mann-Whitney U test $p < 10^{-4}$; Section A.8)[152].

The latent space representation of the influenza HA1 dataset learned by the FactorMuE model shows the data falling approximately along a line (Figure 1.6D; Section A.8). The position of a sequence along this line is linearly proportional to the time at which the sequence was collected, though this information was not included in the model (correlation coefficient $\rho = 0.94$; Figure 1.6E)[189]. Two clusters of outliers violate the proportionality rule. The first (marked by ‡) origi-

nated from mis-annotated entries in the GISAID database (Section A.8). 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[128,241]. 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.

1.6 Discussion

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

2

A Scalable Nonparametric Model

Generative probabilistic modeling of biological sequences has widespread existing and potential use across biology and biomedicine, particularly given advances in high-throughput sequencing, synthesis and editing. However, we still lack methods with nucleotide resolution that are tractable at the scale of whole genomes and that can achieve high predictive accuracy in theory and practice. In this article we propose a new generative sequence model, the Bayesian embedded autoregressive

(BEAR) model, which uses a parametric autoregressive model to specify a conjugate prior over a nonparametric Bayesian Markov model. We explore, theoretically and empirically, applications of BEAR models to a variety of statistical problems including density estimation, robust parameter estimation, goodness-of-fit tests, and two-sample tests. We prove rigorous asymptotic consistency results including nonparametric posterior concentration rates. We scale inference in BEAR models to datasets containing tens of billions of nucleotides. On genomic, transcriptomic, and metagenomic sequence data we show that BEAR models provide large increases in predictive performance as compared to parametric autoregressive models, among other results. BEAR models offer a flexible and scalable framework, with theoretical guarantees, for building and critiquing generative models at the whole genome scale.

This chapter presents work with Alan N. Amin and Debora S. Marks, published at Neural Information Processing Systems (2021)[12]. E.N.W. conceived and guided the research, contributed to the theoretical and empirical results, and wrote the paper; A.N.A. contributed equally to E.N.W. overall, in particular contributing the bulk of the theoretical and empirical results; D.S.M. supervised the research at all stages.

2.1 Introduction

Measuring and making DNA is central to modern biology and biomedicine. Generative probabilistic modeling offers a framework for learning from sequencing data and forming experimentally testable predictions of unobserved or future sequences that can be synthesized in the labo-

ratory[67,110,224]. Existing approaches to genome modeling typically preprocess the data to build a matrix of genetic variants such as single nucleotide polymorphisms[203,92]. However, most modes of sequence variation are more complex. Structural variation occurs widely within individuals (e.g. in cancer), between individuals (e.g. in domesticated plant populations) and between species (e.g. in the human microbiome), and methods for detecting and classifying structural variants are heuristic and designed only for predefined types of sequence variation such as repeats[277,248,161,45,183]. Ideally, we would be able to directly model genome sequencing data and/or assembled genome sequences. However, building generative models that work with raw nucleotides, not matrices of alleles, raises the extreme statistical challenges of having enough *flexiblility* to account for genomic complexity, *interpretability* to reach scientific conclusions, and *scalability* to train on billions of nucleotides. Given the relevance of genetic analysis to human health, models should also possess strong *theoretical guarantees*.

Autoregressive (AR) models are a natural starting point for generative genome modeling, since they (1) have been successfully applied to biological sequences, as well as many other types of non-biological sequential data, (2) can be designed to have interpretable parameters, and (3) can be scaled to big datasets with very long sequences[235,266]. However, since AR models are parametric models, they will in general suffer from misspecification; as we show empirically in Section 2.6, for genomic datasets misspecification can be a serious practical limitation not only for simple AR models but even for deep neural networks.

As an alternative strategy for building generative probabilistic models at the genome scale, we propose in Section 2.2 the nonparametric "Bayesian embedded autoregressive" (BEAR) model.

BEAR models are Bayesian Markov models, with a prior on the lag and conjugate Dirichlet priors on the transition probabilities. The hyperparameters of the Dirichlet prior are controlled by an "embedded" AR model with parameters θ and an overall concentration hyperparameter h, both of which can be optimized via empirical Bayes. In Section 2.3 we show that BEAR models can capture arbitrary data-generating distributions, and establish asymptotic consistency guarantees and convergence rates for nonparametric density estimation. In Section 2.4, we show that the optimal h provides a diagnostic for whether or not the embedded AR model is misspecified and if so by how much, alerting the practitioner when the parameter estimates θ are untrustworthy. Besides estimation problems, BEAR models can also be used to construct goodness-of-fit tests and two-sample tests, thanks to their analytic marginal likelihoods, and we prove consistency results for these tests in Section 2.5. Finally we apply BEAR models at large scale, to genomic datasets with tens of billions of nucleotides, including whole genome, whole transcriptome, and metagenomic sequencing data; we find that BEAR models can have greatly improved performance over AR models (Section 2.6).

Crucial to our theoretical and empirical analysis is the statistical setting: we assume that the data $X_1, \ldots, X_N$ consists of finite but possibly variable length strings (with small alphabets) drawn i.i.d. from some underlying distribution p^*, and study the behavior of estimators and tests as $N \to \infty$. This setup differs from common theoretical analyses of sequence models outside of biology, which typically consider the limit as the length of an individual sequence goes to infinity[95]. In biology, however, we observe finite sequences recorded from many individual species, organisms, cells, molecules, etc. and want to generalize to unseen sequences, making $N \to \infty$ the appropriate large data limit.

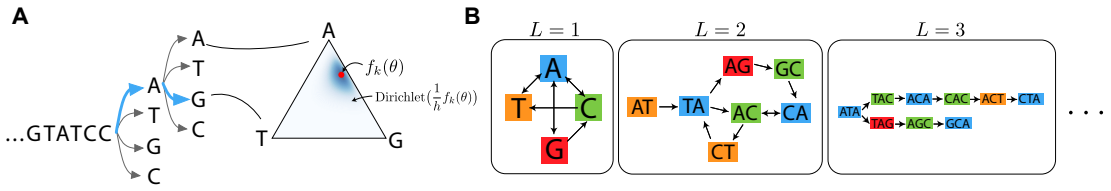


Figure 2.1: Overview of the BEAR model. (A) BEAR models employ a Dirichlet prior on Markov transition probabilities that is centered at the prediction of an AR model. (B) De Bruijn graphs showing BEAR transitions with non-zero probability under an example data-generating distribution. As the lag L increases, the model has higher resolution.

2.2 Bayesian embedded autoregressive models

We first briefly review autoregressive (AR) models as applied to sequences of discrete characters. Let $f(\theta)$ denote an autoregressive function with parameter θ and let L denote the lag of the autoregressive model; then the AR model generates data as

$$X_i | X_{i-L:i-1} \sim \text{Categorical}(f_{X_{i-L:i-1}}(\theta)), \tag{2.1}$$

where i indexes position in the sequence X and $X_{i-L:i-1}$ consists of the previous L letters in the sequence. Since sequence length as well as nucleotide or amino acid content is relevant to biological applications, we use a start symbol $\emptyset$ at the beginning and a stop symbol \$ at the end of each sequence; letters X_i are sampled sequentially starting from the start symbol and continuing until a stop symbol is drawn.

We propose the Bayesian embedded autoregressive (BEAR) model, a Bayesian Markov model

that embeds an AR model into its prior. The BEAR model takes the form,

$$\begin{aligned} L \sim \pi(l), \quad v_k \sim \text{Dirichlet}(\frac{1}{h} f_k(\theta)) \text{ for all } k, \\ X_i | X_{i-L:i-1} \sim \text{Categorical}(v_{X_{i-L:i-1}}), \end{aligned} \tag{2.2}$$

where $\pi(l)$ is a prior on the lag with support up to infinity, $h > 0$ is a concentration hyperparameter, and k is a length L kmer. The BEAR model has three key properties (Fig. 2.1). First, the unrestricted transition parameter v and lag L allow the model to capture exact conditional distributions of p^* to arbitrarily high order: $p^*(X_i|X_{i-1})$ at $L = 1$, then $p^*(X_i|X_{i-2}, X_{i-1})$ at $L = 2$, etc.. This property allows the BEAR model to be used for nonparametric density estimation (Section 2.3). Second, in the limit where $h \to 0$, the BEAR model reduces to the embedded AR model (Eqn. 2.1). The optimal h provides a measurement of the amount of misspecification in the AR model (Section 2.4). Third, the choice of the conjugate Dirichlet prior allows the conditional marginals $p((X_n)_{n=1}^N | L, h, \theta)$ to be computed analytically, and (since L is one-dimensional) the total marginal likelihood $p((X_n)_{n=1}^N | h, \theta)$ to be estimated tractably. This allows BEAR models to be used for hypothesis testing (Section 2.5).

There are a variety of ways of performing inference in BEAR models, but for most applications we will focus on empirical Bayes methods that optimize point estimates of L, h and θ. Let $\#(k, b)$ denote the number of times the length L kmer k is seen followed by the letter or stop symbol b in the dataset $(X_n)_{n=1}^N$. Using a high-performance kmer counter optimized for nucleotide data, KMC, we can compute the count matrix $\#(\cdot, \cdot)$ for all observed kmers k in terabyte-scale datasets, even

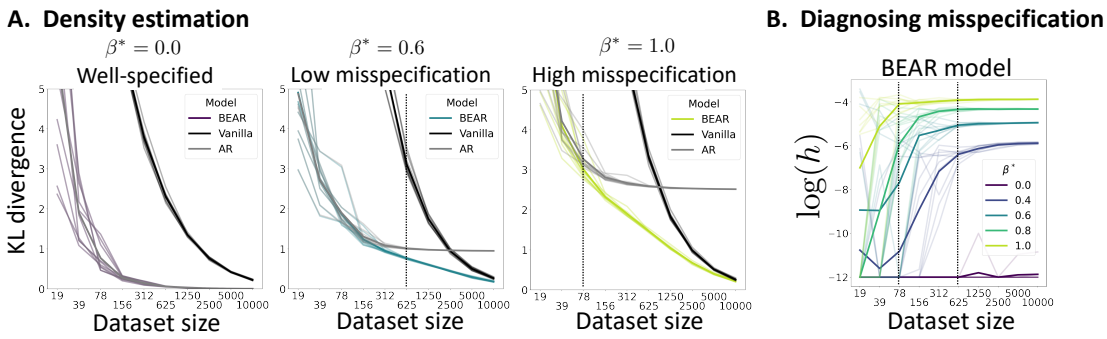


Figure 2.2: BEAR models detect and avoid misspecification without sacrificing small dataset performance. (A) Estimated KL divergence between simulated data-generating distribution p^* and model posterior predictive distribution, as a function of dataset size N. Five independent simulations were run; thin lines correspond to individual simulations, thick lines to the average across simulations. (B) The h misspecification diagnostic as a function of dataset size, for varying β^*. Dataset sizes at which h is close to convergence for $\beta^* = 0.6$ (right) and $\beta^* = 1.0$ (left) are marked with vertical lines.

when the matrix does not fit in main memory (Section B.8.2)[142]. To optimize h and θ, we take advantage of the fact that the log conditional marginal likelihood can be written as a sum over observed kmers,

$$\log p((X_n)_{n=1}^N | L, h, \theta) = \sum_{k:\#k>0} \log \left[\frac{\Gamma(\sum_b \frac{1}{h} f_{kb}(\theta))}{\prod_b \Gamma(\frac{1}{h} f_{kb}(\theta))} \frac{\prod_b \Gamma(\frac{1}{h} f_{kb}(\theta) + \#(k,b))}{\Gamma(\sum_b \frac{1}{h} f_{kb}(\theta) + \#(k,b))} \right]. \quad (2.3)$$

This decomposition lets us construct unbiased stochastic estimates of the gradient with respect to h and θ by subsampling rows of the count matrix (Section B.8.1). Empirical Bayes in the BEAR model therefore costs little extra time as compared to standard stochastic gradient-based optimization of the original AR model. Code is available at https://github.com/debbiemarkslab/BEAR.

2.2.1 Toy example

We next briefly illustrate the properties and advantages of the BEAR model in simulation. We generated samples from an AR model in which $f_k(\theta)$ depends on k linearly as a function of both individual positions and pairwise interactions between positions, with the strength of the pairwise interaction weighted by a parameter β^* (Section B.7.1). We first fit (using maximum likelihood) a linear AR model that lacks pairwise terms and is thus misspecified when $\beta^* > 0$. Since the AR model is misspecified, it does not asymptotically approach the true data-generating distribution p^* (Fig. 2.2A, gray). We next computed the posterior of a vanilla BEAR model without the embedded AR in its prior, instead using the Jeffreys prior $v_k \sim_{iid} \text{Dirichlet}(1/2, \ldots, 1/2)$. The vanilla BEAR model asymptotically approaches the true data generating distribution, since it is a nonparametric model; however, it underperforms the AR model in the low data regime (Fig. 2.2A, black). Finally, we fit a BEAR model with the misspecified linear AR model embedded, using our empirical Bayes procedure. The BEAR model performs just as well as its embedded AR model in the low data regime, just as well as the vanilla model in the high data regime, and better than both at intermediate values (Fig. 2.2A, blue and yellow).

When the AR model is well-specified, the empirical Bayes estimates of the parameters θ under the BEAR model match the maximum likelihood estimates of θ under the AR model nearly exactly (Fig. B.7). When the AR model is misspecified, however, the BEAR model provides a warning: the empirical Bayes estimate of h converges to a non-zero value, rather than zero (Fig. 2.2B). This warning emerges early: h converges well before the vanilla model starts outperforming the misspecified

AR model.

2.2.2 Related work

The key idea behind BEAR models is to nonparametrically perturb a parametric model[177], following a similar strategy to the Polya tree method proposed by Berger & Guglielmi[21]. As in Berger & Guglielmi[21], we use Dirichlet priors centered at the parametric model's predictions, and construct tractable goodness-of-fit tests by exploiting Dirichlet-categorical conjugacy. BEAR models extend these ideas from one-dimensional continuous data to finite-length sequences of discrete characters.

Markov and AR models have a long history and wide range of applications in biological sequence analysis[186,83,211]. Compression methods, in particular, often rely on accurate density estimation and use Markov or AR models to achieve it[63,198,202,237]. We establish theoretical guarantees for density estimation with fully Bayesian Markov models (Section 2.3). AR models used for compression, like other AR models, can be embedded into BEAR models for improved statistical performance and to measure misspecification.

BEAR models are closely linked to non-generative genome analysis methods. Assembly algorithms and variant callers often analyze paths in the de Bruijn graph of a sequence dataset; in the limit $h \to \infty$, samples from the posterior predictive distribution of the BEAR model, conditional on L, correspond to paths through the L-mer de Bruijn graph of the data[44,123]. Comparisons between genomes and other sequences are often made on the basis of kmer counts; our two-sample test provides a generative perspective on this idea[11,62,277].

BEAR models are also connected to ideas in natural language processing, where kmers are re-

ferred to as ngrams. Under the vanilla BEAR model, the mean of the posterior predictive distribution conditional on L corresponds to an ngram additive smoothing model[39]. Comparisons between datasets using their ngram counts are also common in model evaluation metrics such as the BLEU score[192].

2.3 Density estimation

The density estimation problem is that of estimating p^* given data $(X_n)_{n=1}^N$ drawn i.i.d. from p^*. Density estimation is particularly crucial for biological sequence analysis due to its connections to fitness estimation[110,230]. State-of-the-art mutation effect prediction methods and clinical variant interpretation methods rely on density estimates of evolutionary sequence data[215,80]. Density estimation with generative models is particularly useful for protein design, as samples from accurate density estimates are likely to be functional and can be synthesized in the laboratory[224,235]. Despite all these applications, existing density estimation methods for biological sequences lack theoretical guarantees on their accuracy and are often limited in their scale, being restricted to relatively short sequences[284]. Here, we show that the posterior distribution of the BEAR model is consistent and will concentrate on p^* as $N \to \infty$, regardless of what p^* actually is, so long as p^* generates finite length sequences almost surely (a.s.).

We first study the expressiveness of BEAR models. Let $\mathcal{M}_L$ be the set of Markov models p_v with transition probabilities v and lag L that generate finite length strings a.s.. Note that $\mathcal{M}_1 \subset \mathcal{M}_2 \subset \ldots$. Define the union $\mathcal{M} = \cup_{L=1}^{\infty} \mathcal{M}_L$. We can compare $\mathcal{M}$ to the set of distributions over finite

strings S, of which p^* is a member. In Section B.2 we prove that,

Summary of Propositions B.2.1-B.2.4 *Not all possible distributions over S are in $\mathcal{M}$. However, $\mathcal{M}$ is dense on the space of probability distributions over S with the total variation metric.*

The implication of this result is that although BEAR models cannot exactly match arbitrary data-generating distributions, they can approximate p^* arbitrarily well as L increases. This makes asymptotic consistency possible.

We now show that the posterior of the BEAR will in fact asymptotically concentrate on the true p^*, i.e. it is consistent. For tractability, we assume in this section that the prior is fixed (we do not use empirical Bayes). The result relies on the tools for understanding convergence rates of posteriors developed in Ghosal et al.[87]. The most important assumption is that p^* is subexponential, meaning that for some $t > 0$, $E_{p^*} \exp(t|X|) < \infty$ where $|X|$ is the sequence length. Let $\Pi(\cdot|(X_n)_{n=1}^N)$ denote the posterior over sequence distributions. Let $B(p^*, \delta)$ denote a ball of radius δ centered at p^*, using the Hellinger distance.

Summary of Theorem B.6.16 *Given $M > 0$ large enough and $\epsilon \in (0, 1)$ small enough, we have $\Pi(B(p^*, MN^{-\frac{1}{2}\epsilon})|(X_n)_{n=1}^N) \to 1$ in probability.*

A proof is in Section B.6 and simulations in Section B.7.2. This result states that the posterior distribution of the model converges to a delta function at the true distribution p^* regardless of what p^* is. It also provides a rate of convergence: in a parametric model, the uncertainty would shrink as $N^{-\frac{1}{2}}$, but here the rate is slower, $N^{-\frac{1}{2}\epsilon}$, a price paid for the nonparametric model's expressivity[134,100,87]. The proof includes a variety of new theoretical constructions and algorithms that are used to approximate subexponential sequence distributions.

2.4 Robust parameter estimation

To derive a biological understanding of mutational processes, evolutionary history, functional constraints, etc. from sequence data, researchers must estimate model parameters (not just density). However, parameter estimates cannot in general be trusted when models are misspecified [125]. To reach robust scientific conclusions, therefore, parameter estimates should ideally come with a warning about whether or not the model is misspecified and some measurement of the degree of misspecification. Here, we study in BEAR models the asymptotic behavior of empirical Bayes estimates of the AR parameter θ, as well as the hyperparameter h, showing that h diagnoses misspecification in the embedded AR model.

Our analysis builds off the study of empirical Bayes consistency in Petrone et al. [196], which showed that empirical Bayes will, in general, maximize the prior probability of the true data-generating parameter value. Extending this theory to BEAR models is nontrivial, since in BEAR models the standard Laplace approximation to the marginal likelihood can fail. For theoretical tractability, as in many analyses of similar models, we fix L at some arbitrary and large value [106]. Define $p^{*(L)} = \text{argmin}_{p_v \in \mathcal{M}_L} \text{KL}(p^* \| p_v)$ as the closest model in $\mathcal{M}_L$ to p^*, and define v^* such that $p_{v^*} = p^{*(L)}$ (note $p^{*(L)} \to p^*$ as $L \to \infty$). We say that the AR model is misspecified "at resolution L" if f cannot approximate $p^{*(L)}$, i.e. if there does not exist some sequence of parameter values $\tilde{\theta}_N$ such that $p_{f(\tilde{\theta}_N)} \to p^{*(L)}$ as $N \to \infty$; otherwise, the AR model is well-specified at resolution L. Now we can study empirical Bayes estimates of h and θ, denoted h_N and θ_N.

Summary of Propositions B.4.5-B.4.10 *Let* $(h_N)_{N=1}^{\infty}$ *and* $(\theta_N)_{N=1}^{\infty}$ *be sequences maximiz-*

ing the BEAR marginal likelihood $p((X_n)_{n=1}^N | L, h, \theta)$ *for each* N*. If the model is well-specified at resolution* L*, then* $h_N N^{1/4-\epsilon} \to 0$ *for every* $\epsilon > 0$ *and* $p_{f(\theta_N)} \to p^{*(L)}$ *in distribution, with both sequences converging in probability. On the other hand, if the model is misspecified at resolution* L*, then* h_N *is eventually bounded below by some positive (non-zero) number a.s..*

Proofs are in Section B.4 and simulations in Section B.7.1. The implication of this result is that when the AR model is well-specified, h_N converges to zero (at a rate that is a power of the dataset size) and θ_N converges to the parameter value θ^* at which the AR model matches the data (Corollary B.4.6). On the other hand, when the AR model is misspecified, h_N does not converge to zero; heuristically, we find instead that h_N is approximately proportional to a divergence between $p^{*(L)}$ and the AR model,

$$h_N \propto \sum_{k \in \text{acc}_L(p^*)} \left(\text{KL}(f_k(\theta_N) \| v_k^*) + \log(N) \sum_{b \notin \text{supp}_L(p^*)|_k} f_{k,b}(\theta_N) \right), \tag{2.4}$$

where $\text{acc}_L(p^*) = \{k \mid p^*(\#k > 0) > 0\}$ is the set of kmers with non-zero probability and $\text{supp}_L(p^*)|_k = \{b \mid p^*(\#(k,b) > 0) > 0\}$ is the set of transitions from k with non-zero probability. In summary: when fitting a BEAR model by empirical Bayes, you get, along with a parameter estimate θ_N, a value h_N which tells you the amount (from zero to infinity) of misspecification in the AR model. If h_N is close to zero, you can trust the estimate θ_N.

2.5 Hypothesis testing

2.5.1 Goodness-of-fit test

A major outstanding challenge in biological sequence analysis is to build models based on natural sequence data that are accurate enough to generate novel functional sequences [164]. A crucial component of the problem is model evaluation: while relative model performance may be compared on the basis of likelihood, absolute performance – whether or not the model in fact provides an accurate description of the data – is usually addressed solely on the basis of limited numbers of summary statistics, such as average amino acid hydrophobicity or sequence length [235,224]. Given a dataset $(X_n)_{n=1}^N \sim p^*$ i.i.d., a goodness-of-fit test asks whether or not the data distribution p^* matches a model distribution $\tilde{p}$. It takes into account all possible distributions p^* including those that differ from $\tilde{p}$ in a manner that cannot be captured by finitely many summary statistics. We propose a goodness-of-fit test that compares the null hypothesis $\mathcal{H}_0 : p^* = \tilde{p}$ to the alternative $\mathcal{H}_1 : p^* \neq \tilde{p}$ using the Bayes factor $\text{BF} = p((X_n)_{n=1}^N|h,\theta)/\tilde{p}(X_{1:n})$, where $p((X_n)_{n=1}^N|h,\theta) = \sum_L p((X_n)_{n=1}^N|L,h,\theta)\pi(L)$ is the marginal likelihood under the BEAR model. Note that practically, the sum over L is straightforward to approximate by truncation, and that the test can be computed in time linear in the amount of data.

We now prove the consistency of the test. As in comparable theoretical analyses of tests based on Polya trees, for theoretical tractability we truncate the prior, setting $\pi(L) = 0$ for L larger than some arbitrary $\tilde{L}$ but $\pi(L) > 0$ for $L \leq \tilde{L}$ [106]. We treat θ and $h > 0$ as fixed.

Summary of Proposition B.5.1 *If $\tilde{p}$ is at least as close to p^* as $p^{*(L)}$ is, as measured by $\mathrm{KL}(p^* \| \cdot)$, then $BF \to 0$ in probability as $N \to \infty$. On the other hand, if $p^{*(L)}$ is closer than $\tilde{p}$, then $BF \to \infty$ in probability.* A proof is in Section B.5.1 and simulations in Section B.7.3.

An important practical limitation on nonparametric hypothesis testing is low power: since so many alternative distributions must be considered, the null hypothesis can rarely be rejected. However, Proposition B.5.1 holds for the Bayes factor $\mathrm{BF}(L, h, \theta) = p((X_n)_{n=1}^N | L, h, \theta) / \tilde{p}((X_n)_{n=1}^N)$ with any choice of L, $h > 0$, and θ. Thus in practice to increase power we can maximize the value of $\mathrm{BF}(L, h, \theta)$ as a function of L, h, and/or θ (note that this approach is heuristic, since we have not proven the consistency of the maximized Bayes factor). Berger & Guglielmi [21] provide extensive methodological guidance on using analogous tests constructed with Polya trees. Based on their recommendations, we suggest first choosing θ such that $p_{f(\theta)}$ is as close as possible to $\tilde{p}$, then plotting the Bayes factor as a function of h and/or L to identify the maximum value and confirm that any conclusion is robust to changes in h and/or L.

Another challenge in nonparametric hypothesis testing is that it can be difficult to understand how exactly a test reached its conclusion. To identify which sequences provided the most evidence for or against the null hypothesis, we suggest examining the BEAR Bayes factor for each individual sequence conditional on the rest of the dataset, in analogy to the witness function used in kernel-based tests [250,159].

2.5.2 Two-sample test

A two-sample test asks whether or not two datasets $(X_n)_{n=1}^N$ and $(X'_n)_{n=1}^{N'}$ are drawn from the same distribution. Efforts to compare different sequence datasets are widespread in biology: for instance, researchers often wish to determine whether two microbiome samples, taken under different conditions or at different timepoints, are the same up to sampling noise [161]. Two-sample tests can also be used to evaluate generative sequence models that lack tractable likelihoods (for which the goodness-of-fit test proposed above does not apply) such as energy-based models or implicit models like GANs and biophysical simulators [180,96,155]. Assume $(X_n)_{n=1}^N \sim p_1$ and $(X'_n)_{n=1}^{N'} \sim p_2$ i.i.d.. Our BEAR test compares the null hypothesis $\mathcal{H}_0 : p_1 = p_2$ to the alternative $\mathcal{H}_1 : p_1 \neq p_2$ using the Bayes factor

$$\text{BF} = p((X_n)_{n=1}^N | h, \theta) p((X'_n)_{n=1}^{N'} | h, \theta) / p((X_n)_{n=1}^N, (X'_n)_{n=1}^{N'} | h, \theta).$$

As in the goodness-of-fit case, the test can be computed approximately in time linear in the amount of data, and the same advice on increasing power and identifying important sequences holds here too.

We next prove consistency, again truncating the prior at $\tilde{L}$ and fixing h and θ.

Summary of Proposition B.5.3 *If $p_1^{(\tilde{L})} = p_2^{(\tilde{L})}$, then BF $\to 0$ as $N \to \infty$ in probability. Otherwise, if $p_1^{(\tilde{L})} \neq p_2^{(\tilde{L})}$, then BF $\to \infty$ in probability.* A proof is in Section B.5.2 and simulations in Section B.7.3.

Table 2.1: Heldout perplexity. *Whole genome sequencing data*: YSD1: A Salmonella phage. *A. th.*: *Arabidopsis thaliana*, a plant (datasets represent different individuals). *Single cell RNA sequencing data*: PBMC: peripheral blood mononuclear cells, taken from a healthy donor. HL: Hodgkin's lymphoma tumor cells. GBM: glioblastoma tumor cells. *Metagenomic sequencing data*: HC: non-CD and non-UC controls. CD: Crohn's disease. UC: ulcerative colitis. *Full assembled genomes*: Bact.: Bacteria. *Models* Van.: Vanilla (Jeffreys prior). Lin.: Linear. CNN: convolutional neural network. Ref.: reference genome/transcriptome model.

Dataset	AR Lin.	AR CNN	AR Ref.	BEAR Van.	BEAR Lin.	BEAR CNN	BEAR Ref.
YSD1	3.953	3.873	1.266	1.165	**1.144**	**1.144**	1.145
A. th. 1	3.956	3.947	2.686	1.567	1.432	1.432	**1.411**
A. th. 2	3.953	3.949	1.982	1.650	1.463	1.462	**1.441**
A. th. 3	3.998	3.952	2.340	1.834	1.728	**1.727**	1.733
PBMC	3.991	3.974	2.097	1.402	**1.372**	**1.372**	1.374
HL	3.959	3.930	2.141	1.409	**1.378**	**1.378**	1.379
GBM	4.137	4.137	2.366	1.442	**1.406**	**1.406**	**1.406**
HC	3.966	3.946	-	1.652	1.465	**1.464**	-
CD	3.992	3.985	-	1.760	**1.524**	**1.524**	-
UC	3.989	3.986	-	1.644	**1.481**	**1.481**	-
Bact.	3.831	3.794	-	**3.774**	**3.774**	**3.774**	-

2.6 Results

2.6.1 Predicting sequences

We sought to evaluate BEAR models as compared to AR models on the task of predicting real nucleotide (nt) sequences. We considered eleven datasets of four different types: whole genome sequencing read data, single cell RNA sequencing read data (including from patient tumors), metagenomic sequencing read data (including from patient fecal samples) and full bacterial genomes from across the tree of life (Section B.9). Datasets ranged in total size from $\sim 10^7 - 10^{10}$ nt and in individual sequence length from $\sim 10^2 - 10^6$ nt (Table B.1). 25% of data was randomly held out for

testing, in the form of entire sequences (reads, genomes, etc., see Table B.2); our goal was to evaluate BEAR models as density estimators, so we did not use masking (a common holdout strategy in natural language processing). We considered a linear AR model and a deep convolutional neural network (CNN) AR model with $> 10\times$ more parameters, both of which are common models used across a range of applications; we also designed a biologically-structured AR model which makes predictions based on a reference genome and a Jukes-Cantor mutation model (Section B.10.1)[237,202]. We then embedded each AR model to create a corresponding BEAR model. The BEAR models improve over the AR models in nucleotide prediction according to both perplexity (Table 2.1) and accuracy (Table B.3) in all datasets, even when the model lag L is held fixed for comparison (Section B.10.3).

In 10 out of 11 datasets, BEAR models increase nucleotide prediction accuracy from near chance values of $30-35\%$ (in the case of the linear and CNN models) to $78-95\%$, bringing genome-scale models into the realm of potential practical use (Table B.3). The training time for BEAR models is essentially identical to that of AR models, aside from the time required to build the transition count matrix, which need only be done once before training all models (Fig. B.13). Remarkably, the optimal lag L chosen by empirical Bayes is often quite short, less than 20 nt (Table B.4). The improvements offered by BEAR models that use an embedded AR model over the vanilla BEAR model are modest for datasets of this size; however, sequencing experiments are often designed to collect enough data for downstream analyses. We found in an example that, if sequencing coverage was $3\times$ instead of $100\times$, the improvement in prediction accuracy would have been greater than 10 percentage points instead of 0.1 (Section B.10.4; Fig. B.14).

Measuring misspecification When conventional deep neural network methods fail to provide

Table 2.2: Diagnostic h. Abbreviations as in Table 2.1.

Dataset	Lin.	CNN	Ref.
YSD1	5.528	5.461	4.183
A. th. 1	2.765	2.756	2.990
A. th. 2	2.643	2.633	2.326
A. th. 3	3.969	3.964	1.598
PBMC	4.167	4.145	3.762
HL	4.050	4.038	3.581
GBM	4.172	4.154	3.238
HC	4.668	4.651	-
CD	3.096	3.094	-
UC	3.843	3.835	-
Bact.	0.010	0.003	-

strong predictive performance, popular wisdom often ascribes the failure to too much model flexibility or not enough training data, especially in scientific applications. Examining the h misspecification diagnostic in the BEAR models described above, we see that this is not the case here (Table 2.2). The large values of h suggest that where the CNN fails it is not because of too much flexibility but rather too little: the model is not flexible enough to encompass the true data distribution, so it suffers from misspecification. Meanwhile, the reference-based model has only two learned parameters, but is less misspecified than the CNN in all but one dataset. This too runs counter to popular wisdom in machine learning, which often assumes that when principled, low-flexibility scientific models outperform deep neural networks it is thanks to their low variance in the small data regime.

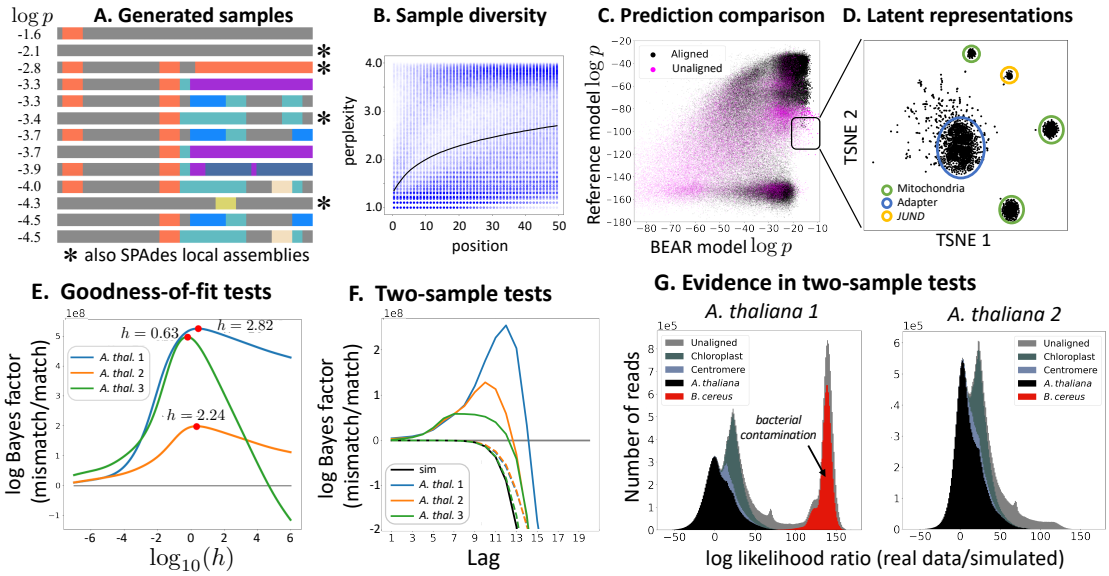


Figure 2.3: Generation, visualization and testing. (A) Sample extrapolations, colored to denote distinct paths through the L-mer de Bruijn graph. (B) Distribution of the perplexity of the next Markov transition under the BEAR model, for each position of the sampled extrapolations, with the per position average shown in black (Section B.11). (C) Log probability of each read in the HL dataset under the BEAR model and a model built from the reference transcriptome. Reads are colored by whether or not they map to the reference. (D) Latent representations of the reads highlighted in C, visualized using tSNE, with clusters annotated as likely coming from mitochondria, the sequencing adapter, or transcripts of the gene *JUND* (Section B.12). (E) Goodness-of-fit test Bayes factor as a function of hyperparameter h. (F) Two-sample test Bayes factor as a function of lag L. Black line compares simulated data to simulated data; dashed lines compare subsampled real data to subsampled real data; solid lines compare real data to simulated data. (G) Log probability of each read under the real data BEAR model minus the log probability under the simulated data BEAR model (Section B.13).

2.6.2 Generating samples

BEAR models are generative and can be used to sample new sequences. We sampled extrapolations from the end of a read sequence recorded in a plant (*A. thaliana*) whole genome sequencing experiment, and compared to an alternative non-probabilistic extrapolation method that is widely used in biology, local assembly (Fig. 2.3A; Section B.11). In this example the assembly algorithm SPAdes returns four possible assemblies, a relatively large number compared to other reads in the dataset (Fig. 2.3A stars)[16]. Samples from the BEAR model include these four possibilities, but also many more, some with higher probability. The distribution over possible nucleotide choices under the BEAR model is much wider than the number of assemblies would suggest: it has a perplexity of 1.4 per position (on average across samples) at the beginning of the extrapolation, and a perplexity of 2.7 at 50 nucleotides (Fig. 2.3B). These observations suggest that SPAdes, which does not provide a measurement of uncertainty, may not be capturing the full range of possible sequences.

2.6.3 Visualizing data

Methods for learning local representations or features of biological sequences can be powerful tools for visualization and semisupervised learning[25]. One approach to extracting such representations is to learn a generative model $q(X_1, \ldots, X_{L+1})$ of kmers, for instance using a variational autoencoder. While such models are not autoregressive, the small size of the DNA alphabet makes it tractable to estimate the conditional $q(X_{L+1}|X_{1:L})$ by Bayes' rule, and this conditional can then be embedded into a BEAR model. We applied this strategy to probabilistic PCA. We visualized in low

dimensions the inferred latent representation for a model trained on a single cell RNA sequencing dataset (HL), and were able to assign annotations to clusters, including those containing unmapped reads (Fig. 2.3CD; Section B.12). The BEAR model however raises the warning that the model is misspecified ($h = 4.836$), suggesting there may be richer latent structure yet to discover.

2.6.4 Testing hypotheses

The question of when and how microbiomes change is widespread, but has in the past relied on summary statistics of sequencing datasets [161]. Schreiber et al. [229] studied changes in patient urine microbiomes before and after kidney transplant, and performed both unbiased metagenomic sequencing and diagnostic quantitative polymerase chain reaction (qPCR) for a specific virus associated with complications (JC polyomavirus). They found evidence of donor-to-recipient viral transmission in 5 cases out of 14. We applied the BEAR two-sample test to patients' metagenomic sequencing data before and after transplantation, using the vanilla Jeffreys prior and integrating over lags, in order to detect changes; the test rejects the null hypothesis in all 5 cases where there was transmission, and accepts the null hypothesis in all but one of the remaining 9 cases (Table B.6; Section B.13.1). These results show, in a small example, that the two-sample test has sufficient power to detect microbiome changes in real data, and can be consistent with more specific tests.

We next applied BEAR hypothesis tests to evaluate generative models. We evaluated the reference-based AR model described above using the BEAR goodness-of-fit test. The test identifies considerable evidence (log Bayes factor $> 10^8$) for misspecification in each *A. thaliana* whole genome sequencing dataset, and this conclusion is robust to a wide range of h values (Fig. 2.3E; Section B.13.2).

Next, we evaluated a detailed simulation model (ART) that is intended to generate likely reads of a given reference genome[111]. The model lacks tractable likelihoods, so we use the BEAR two-sample test. When integrating over all lags, the test accepts the null hypothesis, suggesting that the simulation model is accurate; if we examine the test results for individual lags L to increase power, however, we can see some evidence of differences (Fig. 2.3F; Section B.13.2). Note that as L increases, there is a tradeoff: tests with larger lag can detect more subtle differences between the two distributions, but have less statistical power since they must consider a larger set of possible distributions. Thus the Bayes factor first increases and then decreases with lag, reaching a peak at intermediate values where there is the most evidence of difference. To understand in detail the source of the detected differences between the data and the simulation model, we examined the conditional Bayes factor for individual reads, discovering clusters of reads that are poorly explained by the simulation model (Fig. 2.3G). One group mapped to chloroplasts, an organelle with its own genome that is variable in copy number; reads mapping to centromeres, an area of the plant genome for which the reference genome is considered unreliable, were also poorly explained by the simulation model. In one dataset we found a cluster of outliers that did not map to *A. thaliana* at all, and instead mapped to a common soil bacteria, *Bacillus cereus*, presumably a contaminant in the experiment (Fig. 2.3G, left). These results illustrate how BEAR hypothesis tests can be used not only for testing but also for detailed model criticism.

2.7 Discussion

In this article we proposed the nonparametric BEAR model, studied its theoretical properties, and developed algorithms and implementations for terabyte-scale inference. BEAR models substantially outperform standard AR models on a variety of datasets, and come with extensive theoretical guarantees, including for density estimation, misspecification detection, and hypothesis testing. BEAR models are closely connected to non-probabilistic genome analysis methods, such as de Bruijn graph assembly, but provide an alternative that is uncertainty-aware. Note, however, that BEAR models do not explicitly account for paired-end read information, or other sources of long-distance information; this is an important area for future work. While there has been little previous empirical or theoretical work in the machine learning literature on generative models of full genomic, transcriptomic or metagenomic sequences, we hope BEAR models provide a useful starting point.

3

Variational Synthesis

Generative probabilistic models of biological sequences have widespread existing and potential applications in analyzing, predicting and designing proteins, RNA and genomes. To test the predictions of such a model experimentally, the standard approach is to draw samples, and then synthesize each sample individually in the laboratory. However, often orders of magnitude more sequences can be experimentally assayed than can be affordably synthesized individually. In this article, we pro-

pose instead to use stochastic synthesis methods, such as mixed nucleotides or trimers. We describe a black-box algorithm for optimizing stochastic synthesis protocols to produce approximate samples from any target generative model. We establish theoretical bounds on the method's performance, and validate it in simulation using held-out sequence-to-function predictors trained on real experimental data. We show that using optimized stochastic synthesis protocols in place of individual synthesis can increase the number of hits in protein engineering efforts by orders of magnitude, e.g. from zero to a thousand.

This chapter presents work done in collaboration with Alan N. Amin, Will Grathwohl, Daniel Kassler, Jean Disset and Debora S. Marks, published at the International Conference on Artificial Intelligence and Statistics (2022)[282]. E.N.W. conceived the research, performed the research and wrote the paper. A.N.A. and J.D. contributed code, and A.N.A contributed to the theoretical results. W.G. and D.K. contributed to preliminary experiments. D.S.M. supervised the research at all stages.

3.1 Introduction

Large-scale nucleic acid sequencing and synthesis is integral to modern biology and biomedicine, from biotechnology to epidemiology to neuroscience to agriculture to evolutionary biology and beyond. Generative probabilistic modeling offers a rigorous framework for analyzing large scale sequencing data and forming predictions of new sequences that can be synthesized in the laboratory. Generative models have been used, for instance, to infer underlying structural and functional con-

straints on protein evolution, to predict pathogen sequences that may emerge in the future, and to predict novel enzyme sequences with desired functional properties [168,110,284,224]. In order to assay the properties of predicted sequences and discover novel functional sequences, samples from generative models must be synthesized in the laboratory at scale. Large libraries are particularly important for protein engineering applications, where they are screened for hits with rare properties, e.g. a particular catalytic or binding activity.

Unfortunately, synthesizing large numbers of samples from generative sequence models is challenging. The standard approach, which we refer to as "Monte Carlo (MC) synthesis", is to (1) sample from the model computationally, and then (2) synthesize each sample individually [224,235,164]. In practice, however, MC synthesis is limited by cost: despite recent advances in synthesis technology, gene-length libraries typically do not exceed 10^4 unique sequences [145]. Far larger libraries, on the order of $10^6 - 10^{13}$, can be screened in many high-throughput assays. The set of likely sequences predicted by state-of-the-art generative models is often vastly larger still: a protein model with per-residue perplexity of 2 across sequences of length 100 predicts effectively $2^{100} \approx 10^{30}$ sequences. Thus MC synthesis often will come nowhere near comprehensive exploration of a model's predictions.

In principal, combinatorial and stochastic synthesis methods – such as error prone PCR and mixed nucleotides – offer an alternative approach capable of producing much larger numbers of unique sequences for the same cost. However, the sequences produced by these methods are random, and so it is unclear how to use stochastic synthesis to gain insight into the predictions of a given generative sequence model.

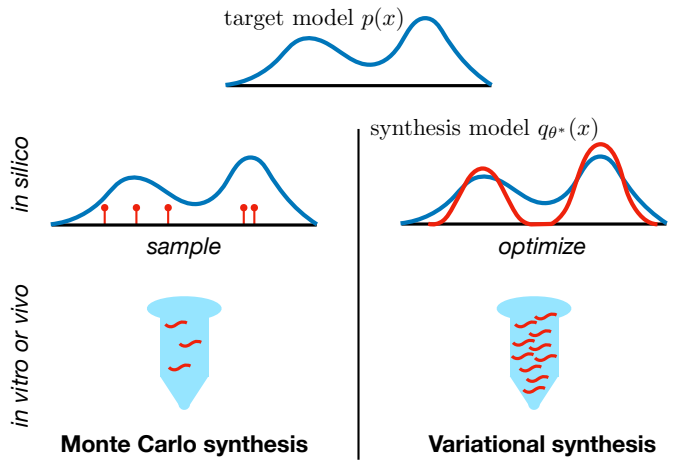


Figure 3.1: The standard synthesis approach for generative sequence models (Monte Carlo synthesis) is to sample sequences *in silico* and synthesize samples individually *in vitro*. The proposed approach (variational synthesis) is to optimize the experimental parameters of a stochastic synthesis protocol *in silico* and then run the protocol *in vitro* or *in vivo* to produce a larger number of samples.

In this article, we describe an experimental design method – "variational synthesis" – that leverages stochastic DNA synthesis to overcome the limitations of MC synthesis. The basic idea is to optimize the parameters of the laboratory synthesis protocol to produce samples from a distribution close to the distribution of the target generative model. Variational synthesis is a rigorous approach to building ultra-large scale libraries based on generative sequence models, and can dramatically accelerate the discovery of novel functional sequences.

3.2 Method

We consider an arbitrary target generative model that describes a probability distribution $p(x)$ over sequences x. We are interested in assaying samples from the model experimentally. The standard method, MC synthesis, is to (1) draw samples $X_1, ..., X_{N_0} \sim p$ i.i.d. computationally and then

(2) synthesize each sequence in the laboratory, deterministically. This approach is limited by the number of sequences N_0 that can be affordably synthesized deterministically, typically on the order of 10^4 or less for gene-length sequences.

As an alternative, we propose "variational synthesis" (Figure 3.1): (1) write down a probabilistic model $q_\theta(x)$ of sequences produced by a stochastic synthesis protocol with experimental parameters θ, (2) minimize a divergence between q_θ and p to find $q_{\theta^*} \approx p$ and (3) run the stochastic synthesis protocol in the laboratory, producing samples $X_1, ..., X_{N_1} \sim q_{\theta^*}$ i.i.d.. This approach is limited by the number of sequences N_1 that can be affordably screened, where in general N_1 can be orders of magnitude larger than N_0, e.g. $10^6 - 10^{11}$. The increase in samples comes at the cost of accuracy, since q_{θ^*} may not exactly match p.

3.2.1 Stochastic Synthesis Models

The first step of variational synthesis is to write down models q_θ of stochastic synthesis protocols. We focus on five key technologies: (1) enzymatic mutagenesis, e.g. error-prone PCR or Orthorep[290,210], (2) mixed nucleotide synthesis, often referred to as "degenerate codon libraries" in the context of proteins[194,174], (3) mixed trimer synthesis[136,135,172], (4) combinatorial variant libraries[264] and (5) combinatorial assembly[89]. We focus on models of protein sequences; models of DNA or RNA are simpler.

We describe stochastic synthesis models q_θ using a four-step generative process (Figure 3.2): (1) sample one of M "templates" from each of K "pools", (2) join the templates together, (3) sample codons independently at each position of the combined templates and (4) translate the DNA se-

quence into protein. For example, consider the protocol of combinatorial assembly plus error prone PCR: we start with a library of oligos, join (assemble) a random sample of oligos into a larger sequence, and then mutagenize the sequence. Abstractly, we refer to the distribution over codons obtained by mutagenizing a particular oligo as a "template". Techniques such as mixed nucleotides can produce alternative distributions over codons, described by different "templates". Mathematically, let $u_{kzj(b_1,b_2,b_3)}$ denote the probability of generating codon (b_1, b_2, b_3) at the jth position of template z in pool k. Let T be the translation matrix, defined as $T_{(b_1,b_2,b_3)d} = 1$ if the codon (b_1, b_2, b_3) codes for the amino acid d and $T_{(b_1,b_2,b_3)d} = 0$ otherwise. (For instance, $T_{(G,T,A)V} = 1$ since the codon GTA codes for the amino acid V.) The complete model (Figure 3.2) is

$$\begin{aligned} Z_i &\sim p_w, \\ C_i &:= \text{concatenate}(u_{1Z_{i1}}, \ldots, u_{KZ_{iK}}), \\ H_i &\sim \text{Categorical}(C_i), \\ X_i &:= H_i \cdot T, \end{aligned} \tag{3.1}$$

where the "concatenate" operation stacks matrices vertically, and the categorical distribution produces one-hot encoded samples based on the probabilities in each row. Here, Z_i is the vector of templates used for sequence i, drawn from an underlying distribution p_w, while C_i is a matrix containing the codon probabilities for each site along sequence i and H_i is a one-hot encoding of the codons in sequence i (Table C.1 provides a complete notation reference).

Different synthesis technologies impose different constraints on p_w, corresponding to different

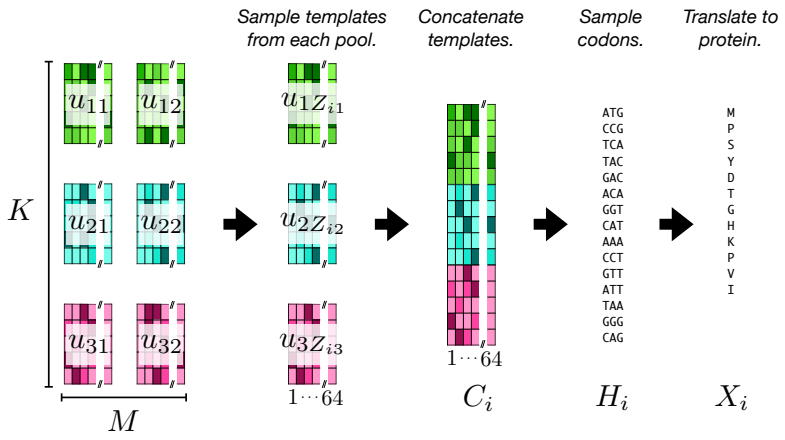


Figure 3.2: Overview of the synthesis model (Equation 3.1). From each of K pools we draw one of M templates, u_{kz}, according to the random vector Z_i. We concatenate the templates to form a matrix of codon probabilities C_i. Then codons are sampled at each position to form H_i, which is finally translated into a protein sequence X_i.

assembly methods, and different constraints on u, corresponding to different codon diversification methods. (The biochemical basis for these different mathematical constraints is described further in Section C.1.) We consider two possible constraints on p_w:

1. Fixed assembly $Z_{i1} \sim \text{Categorical}(w)$ and $Z_{i2} := \ldots := Z_{iK} := Z_{i1}$. Here we assume that there are M templates in each pool, and that the choice of template from the first pool dictates the choice from all the others. The experimentalist can choose the probability vector $w \in \Delta_M$, where Δ_M denotes the $M - 1$ simplex; chemically, w is controlled by the relative concentration of each template. In this case, the synthesis model (Equation 3.1) is a mixture model.

2. Combinatorial assembly: $Z_{ik} \sim \text{Categorical}(w_k)$ for all $k \in \{1, \ldots, K\}$. In this case each template from each pool is drawn independently. The experimentalist can choose the probability vectors $w_k \in \Delta_M$ for each pool.

We describe constraints on the codon probabilities of each template in terms of spaces $\mathcal{U}$, where

the experimentalist can choose any $u_{kzj} \in \mathcal{U}$ for all k, z, j. We use $v \otimes v'$ to denote the outer product of two vectors v and v'. Overloading notation, for two sets of vectors S and S', we use $S \otimes S'$ to denote the set of outer products of their members, that is $S \otimes S' := \{v \otimes v' : v \in S \text{ and } v' \in S'\}$. We consider the following constraints:

1. Arbitrary codon mixtures: $\mathcal{U} = \Delta_{64}$. In this case, the experimentalist can choose any probability distribution over the 64 codons at each position in each template.* Combinatorial variant libraries have this constraint; it is the most flexible of the codon probability constraints we consider.

2. Finite codon mixtures: $\mathcal{U} = \{v_1, \ldots, v_A\}$ where $v_a \in \Delta_{64}$ for all a. In this case, the experimentalist must first fix a library of A different codon mixtures, and then, for each position in each template, choose one of these mixtures v_a to use. Mixed trimer synthesis protocols often have this constraint; in this case, v_a is determined by the relative concentration of each trimer in mixture a.

3. Finite nucleotide mixtures: $\mathcal{U} = \{v_1, \ldots, v_A\} \otimes \{v_1, \ldots, v_A\} \otimes \{v_1, \ldots, v_A\}$ where $v_a \in \Delta_4$ for all a. In this case, the experimentalist must first fix a library of A different *nucleotide* mixtures, and then, for each position in each codon in each template, choose one of these mixtures to use. Mixed nucleotide synthesis protocols often have this constraint; in this case, v_a is determined by the relative concentration of each nucleotide in mixture a

4. Enzymatic mutagenesis: $\mathcal{U} = \{S^\tau e_1, \ldots, S^\tau e_4\} \otimes \{S^\tau e_1, \ldots, S^\tau e_4\} \otimes \{S^\tau e_1, \ldots, S^\tau e_4\}$ where S is a substitution matrix, S^τ is a matrix exponential, and e_j is the length 4 vector of all zeros

*We index the 64 codons either using either tuples $(A, A, A), \ldots, (T, T, T)$ or integers $1, \ldots, 64$, depending on convenience.

except a one at position j. The substitution matrix S is an intrinsic property of the chosen mutagenic enzyme (i.e. the particular error prone polymerase); in general, it has positive non-zero entries, linearly independent columns, and the sum of each column is 1. The number of rounds of mutagenesis $\tau \in \{1, 2, \ldots\}$ can be controlled experimentally.

Once an assembly technology (fixed or combinatorial) and codon diversification technology (arbitrary codon, finite codon, finite nucleotide or enzymatic) are chosen, the parameters θ of the synthesis model q_θ (Equation 3.1) that must be optimized consist of: w (the template probabilities), u (the codon probabilities), v (if we are using finite nucleotide or codon mixtures) and τ (if we are using enzymatic mutagenesis).

3.2.2 Black-Box Optimization

The second step of variational synthesis is to optimize the synthesis protocol, such that $q_{\theta^*} \approx p$. For some target/synthesis pairs – for instance, when the target is a regression model with a MuE output and fixed latent alignment[284], and the synthesis method uses fixed assembly and arbitrary codon mixtures – we can analytically and exactly match q_{θ^*} to p (Supplement C.2.1). In most cases, however, an exact match between the target distribution and the synthesis distribution is impossible, and an analytic minimum intractable. We therefore propose an approximate optimization procedure. The primary desiderata are that it should be (1) black-box, in the sense that it can be applied to arbitrary target distributions p so long as p can be tractably sampled from, (2) scalable to large library sizes, since q_θ may for instance be a mixture model with 1000 or more components and (3) able to handle large numbers of discrete parameters, since $\mathcal{U}$ can be finite.

We propose to minimize the Kullback-Leibler (KL) divergence between the target model and the synthesis model, estimating $\theta^* := \text{argmin}_\theta \, \text{KL}(p \| q_\theta)$ by (1) drawing samples from the target model $X_1, \ldots, X_{\tilde{N}} \sim p$ i.i.d. and (2) maximizing the log likelihood of the samples under q_θ using a stochastic expectation-maximization (EM) algorithm [33]. This approach only relies on samples from p, so can be applied whenever MC synthesis can be applied; in particular, it does not require access to likelihoods of p, allowing p to be an implicit model (e.g. a GAN). EM does not require access to derivatives of $q_\theta(x)$ with respect to θ, and can easily handle categorical parameters. Finally, since the stochastic EM algorithm relies only on minibatches of data, the method is highly scalable. Sections C.2.2 and C.2.3 detail the algorithm and provide advice on training, including the choice of $\tilde{N}$. Code is provided at https://github.com/debbiemarkslab/variational-synthesis.

Often the target p describes a distribution over variable-length sequences. One way to account for this, in the case of protein sequences, is to compute the likelihood of each sequence followed by a stop codon, treating the remainder of the DNA sequence as missing data when fitting q_θ (Supplement C.2.4). Alternatively, a restriction site could be appended, and the remainder of the DNA sequence again treated as missing data; after synthesis, the sequences could be digested to the appropriate length. Our optimization procedure can thus be applied to p that produce variable-length sequences, so long as the length distribution is bounded.

3.3 Related work

Optimal design methods for stochastic synthesis have a long history, but existing techniques are in general non-probabilistic – they do not work with explicit target distributions p or synthesis distributions q_θ – and, practically, cannot be applied to produce samples from an arbitrary generative model p. Methods such as LibDesign [174] and SwiftLib [126] optimize degenerate codon libraries to match the per-position amino acid frequencies in a multiple sequence alignment, while limiting the total size of the library. SwiftLib has for instance been used to design massive libraries of mini-protein sensors and therapeutics [40,140]. OCoM [193] applies similar ideas to handle pairwise correlations. The recent DeCoDe method [234] designs degenerate codon libraries to produce as many members of a set of target sequences as possible, while limiting the total size of the library; it can be interpreted probabilistically as attempting to maximize the overlap in support between a synthesis distribution q_θ and a target distribution p, while regularizing the size of the support of q_θ (Section C.3.1). Meanwhile, SCHEMA and RASPP [278,73] are used to optimize combinatorial assembly protocols based on protein structure, and have been applied to engineer new optogenetic tools [20]; when the target model p is a Potts model that accurately reflects protein structure, variational synthesis will prefer similar solutions (Section C.3.2). Note that these existing non-probabilistic stochastic synthesis design tools are often used to construct libraries of diversified sequences in the context of directed evolution experiments, and we expect variational synthesis to also be applicable in the same context.

Batched stochastic Bayesian optimization [294] is comparable to variational synthesis in that it is a rigorous and probabilistic approach to stochastic synthesis optimization. Unlike variational syn-

thesis, it is focused on optimizing a reward function, rather than drawing samples from a generative sequence model. It is also not black-box, relying on the particular structure of the reward function (a Gaussian process) and focusing on just one stochastic synthesis method.

Stochastic synthesis models related to those proposed in Section 3.2.1 have been used in the past for inference from observational data, rather than experimental design. For instance, Tomezsko et al. [260] use a mixture model of sequences to infer RNA structural diversity from dimethyl sulfate mutational profiling data.

Variational synthesis is inspired by variational inference (VI) [27]. Both minimize a divergence between a simple approximating distribution and a target distribution (a posterior in the case of VI). Both can take advantage of the expressiveness of mixture models to achieve close matches to the target distribution [175,97,162]. Both can be contrasted with older methods for exact sampling from a target distribution (Markov chain Monte Carlo in the case of VI, Monte Carlo synthesis in the case of variational synthesis); both trade accuracy for scale, enabling large numbers of approximate samples to be drawn (computationally in the case of VI, physically in the case of variational synthesis). Both can be black-box, enabling automatic sampling for a large class of target distributions [208,147].

3.4 Theory

3.4.1 Approximation Error

In this section, we analyze the downstream consequences of using variational synthesis in place of MC synthesis. After synthesizing (approximate) samples from p, the sequences will be experi-

mentally characterized using a high-throughput assay, described by a function f, which provides measurements $f(X_1), \ldots, f(X_N)$ of each synthesized sequence. The assay may measure binding strength, enzymatic activity, fluorescence, etc.. f is assumed to be unknown before performing the experiment. We consider two distinct goals. The first goal is to estimate the average value $\mathbb{E}_{X\sim p}[f(X)]$. For instance, we may want to estimate the average drug resistance of future pathogen sequences predicted by p. Second, we may be interested in discovering a large number of sequences with a desired property, i.e. we want to maximize $\sum_{i=1}^{N} f(X_i)$ where $f(x) = 1$ if the sequence has the property and $f(x) = 0$ otherwise. E.g. if we want to engineer a new plastic-degrading protein, we want to find as many sequences as possible with high degradation rates.

Estimating $\mathbb{E}_{X\sim p}[f(X)]$. MC synthesis and variational synthesis lead to two distinct estimators for $I := \mathbb{E}_{X\sim p}[f(X)]$, and in this section we compare their performance theoretically. In particular, the MC synthesis estimator is $\hat{I}^{(a)} := \frac{1}{N_0}\sum_{i=1}^{N_0} f(X_i)$ where $X_1, \ldots, X_{N_0} \sim p$, while the variational synthesis estimator is $\hat{I}^{(b)} := \frac{1}{N_1}\sum_{i=1}^{N_1} f(X_i)$ where $X_1, \ldots, X_{N_1} \sim q_{\theta^*}$. We have no *a priori* knowledge of f, so to compare estimators we evaluate worst-case performance over a family of functions $\mathcal{F}$. In practice, nearly all experimental assays have limited dynamic range; we therefore take $\mathcal{F}$ to be the set of bounded functions, $\mathcal{F} := \{f : \max_{x\in\mathcal{X}} |f(x)| \leq f_{\max}\}$, where $\mathcal{X}$ is the set of protein sequences of length less than or equal to L.

Proposition 3.4.1. *The worst-case mean absolute deviation of the exact synthesis estimator satisfies*

$$\frac{1}{f_{\max}} \sup_{f\in\mathcal{F}} \mathbb{E}[|\hat{I}^{(a)} - I|] \leq \frac{1}{\sqrt{N_0}}. \tag{3.2}$$

The worst-case mean absolute deviation of the stochastic synthesis estimator satisfies

$$\frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \mathbb{E}[|\hat{I}^{(b)} - I|] \leq \frac{1}{\sqrt{N_1}} + \sqrt{\frac{1}{2} \textsc{kl}(p \| q_{\theta^*})}. \tag{3.3}$$

The proof, which can be found in Section C.4.2, uses the integral probability metric representation of total variation along with Pinsker's inequality. This result describes a bias-variance tradeoff: using variational synthesis in place of MC synthesis leads to less variance (since $N_1 > N_0$) but introduces bias if q_{θ^*} does not exactly match p. Our optimization procedure (Section 3.2.2) minimizes bias by minimizing $\textsc{kl}(p \| q_\theta)$.

If we have access to paired sequencing data, for instance if the hits of a high-throughput screen are deep-sequenced, we can remove the bias in the variational synthesis estimator via importance-weighting. We analyze this approach in Section C.4.3.

Maximizing $\sum_{i=1}^N f(X_i)$. How many more hits can we expect to discover when using variational synthesis as opposed to MC synthesis? To address this question, we take $f : \mathcal{X} \mapsto \{0, 1\}$, and compare the total number of hits when using variational synthesis, $N_1 \hat{I}^{(b)}$, to the number of hits when using MC synthesis, $N_0 \hat{I}^{(a)}$.

Corollary 3.4.2. *The expected increase in hits when using variational instead of MC synthesis satisfies*

$$\begin{aligned} &\mathbb{E}[N_1 \hat{I}^{(b)} - N_0 \hat{I}^{(a)}] \geq \\ &\left(I - \sqrt{\frac{1}{2} \textsc{kl}(p \| q_{\theta^*})}\right) N_1 - \sqrt{N_1} - I N_0 - \sqrt{N_0}. \end{aligned} \tag{3.4}$$

See Section C.4.4 for a proof. In general N_1 is much larger than N_0, so the determining factor as to whether variational synthesis outperforms MC synthesis is whether q_{θ^*} is a sufficiently close approximation to p, i.e. whether $\sqrt{\frac{1}{2}\text{KL}(p\|q_{\theta^*})} < I$. If so, the payoff from using variational synthesis can be substantial: to first order, the number of hits increases linearly with the number of sequences N_1. Our optimization procedure maximizes the lower bound on the number of hits by minimizing $\text{KL}(p\|q_\theta)$.

3.4.2 Performance Limits

We have seen that the success of variational synthesis is determined by how closely q_θ can match the target p. In this section, we analyze how closely the stochastic synthesis models described in Section 3.2.1 can match arbitrary target distributions p.

Limits on fixed assembly. We start by showing that synthesis protocols that use fixed assembly, and do not use enzymatic mutagenesis, can match any target distribution p arbitrarily well. We use $q_\theta(x|z)$ as shorthand for $q_\theta(x|Z_{i1} = z)$, the synthesis model distribution conditioned on the choice of template (mixture component). Let $\mathcal{P}(\mathcal{X})$ denote the set of probability distributions over $\mathcal{X}$. Let $\text{supp}(q_\theta(x|z))$ denote the support of the distribution $q_\theta(x|z)$, i.e. the set of all $x \in \mathcal{X}$ such that $q_\theta(x|z) > 0$.

Proposition 3.4.3. *When using either **arbitrary codon mixtures**, **finite codon mixtures** (with $A \geq 21$), or **finite nucleotide mixtures** (with $A \geq 4$): for any $p \in \mathcal{P}(\mathcal{X})$ and $\eta > 0$ there exists some M and θ such that (1) $\text{KL}(p\|q_\theta) < \eta$ and (2) $\text{supp}(q_\theta(x|z)) = \mathcal{X}$ for all $z \in \{1, \ldots, M\}$. When*

*using **enzymatic mutagenesis**: there exists some $p \in \mathcal{P}(\mathcal{X})$ and $\eta > 0$ such that for all M and θ, we have $\text{KL}(p\|q_\theta) > \eta$.*

See Section C.4.5 for a proof. The result says that as long as we are not using enzymatic mutagenesis, the target distribution p can be arbitrarily well approximated without resorting to individual synthesis (that is, without setting $q_\theta(x|z)$ to be a delta function). Fundamentally, the problem with enzymatic mutagenesis is its discreteness: a sequence can be mutated at minimum once, so there is a minimum non-zero codon probability, given by the properties of the enzyme. This sets a limit on the "resolution" of p that can be matched by the synthesis procedure.[†]

Limits on combinatorial assembly. We next show that any synthesis protocols using combinatorial assembly cannot closely match arbitrary targets p even in the limit that the library size M goes to infinity. The result holds for any choice of $\mathcal{U}$.

Proposition 3.4.4. *When using **combinatorial assembly**, so long as $K > 1$, there exists $p \in \mathcal{P}(\mathcal{X})$ and $\eta > 0$ such that for all M and θ, we have $\text{KL}(p\|q_\theta) > \eta$.*

See Section C.4.6 for a proof. The key problem with combinatorial assembly is that it forces templates to be independent of one another; it therefore cannot match probability distributions p which have correlations between regions covered by each template.

[†]In practice, despite the mathematical idealization of our models, all synthesis technologies have a minimum non-zero codon probability, set by engineering constraints. The key question is really how low this number is comparatively.

3.5 Results

3.5.1 Matching Evolutionary Enzyme Models

We next evaluated the ability of variational synthesis to produce approximate samples from target protein models trained on real data. As a first target, we chose a Potts model trained on dihydrofolate reductase (DHFR) sequences from across evolution; DHFR is an enzyme crucial for nucleic acid synthesis. Potts models of protein sequences have been studied extensively, and MC synthesis from Potts models can produce functional sequences[224]. We optimized each of our proposed stochastic synthesis models, setting hyperparameters based on commercially-available technologies (Section C.5.2). We compared our proposed variational synthesis approach to a baseline heuristic library diversification strategy of MC synthesis plus mutagenesis: (1) draw samples from p and then (2) apply five rounds of mutagenesis with ePCR (Section C.5.3). To evaluate how well each synthesis model matched the target distribution we estimated its per residue perplexity (Section C.5.4). However, perplexity only provides a measurement of the relative quality of different synthesis procedures, rather than an absolute measurement of whether they match the data distribution. We therefore applied a Bayesian two-sample test for biological sequences – the BEAR test[12] – to determine whether q_{θ^*} in fact matches p, based on 100,000 samples from each (Section C.5.5).

All variational synthesis methods dramatically outperform the baseline (Figure 3.3A), and some are capable of matching the target p closely, passing the two-sample test (Figure 3.3B). Two key determinants of the performance of the stochastic synthesis model are (1) the expressivity of the codon

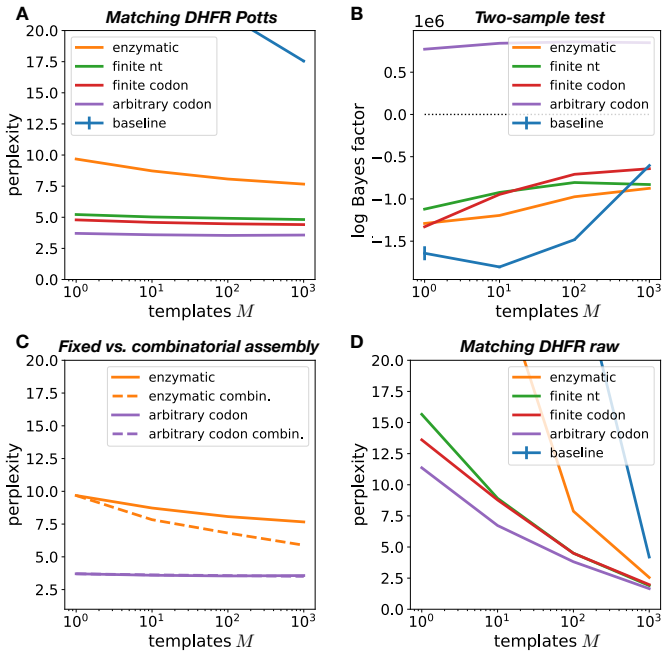


Figure 3.3: Perplexity (A) and two-sample test Bayes factor (B) of different codon diversification methods, with fixed assembly, applied to a target Potts DHFR model. Positive Bayes factors support the hypothesis that the synthesis and target distributions match. (C) Perplexity of combinatorial versus fixed assembly, applied to Potts DHFR model. (D) Perplexity of synthesis models with fixed assembly applied to unaligned DHFR sequences. Error estimates for each plot are described in detail in Section C.5.7.

diversification method – that is, the size of the set of allowed $\mathcal{U}$ – and (2) the number of templates M (Section C.5.2). Performance in terms of perplexity shows an improvement with increasingly large $\mathcal{U}$ and increasing M. Note that due to current technology costs, when using codon mixtures, M must in general be small (e.g. $\leq$ 10) as compared to enzymatic mutagenesis or nucleotide mixtures (where M can be on the order of 1000). Nonetheless, using arbitrary codon mixtures with $M = 1$ templates outperforms the alternative technologies with $M = 1000$ templates.

The advantages of combinatorial assembly over fixed assembly vary depending on the codon diversification technology. Combinatorial assembly improves perplexity when using enzymatic

mutagenesis, but has little effect when using arbitrary codon mixtures (Figure 3.3C and Figure C.3), while introducing error in the covariance matrix of q_{θ^*} (Figure C.4).

We next explored the application of variational synthesis to target distributions over variable-length sequences (the DHFR Potts model was trained on aligned sequences and generates fixed-length sequences). We optimized synthesis models directly on the same evolutionary data used to train the DHFR Potts model (with gaps removed); the target here is the true evolutionary data-generating process, and unknown (Section C.5.1). Enzymatic mutagenesis with large M outperforms arbitrary codon mixtures with small M in this case (Figures 3.3D and C.5). The best synthesis technology can thus depend on the target.

3.5.2 Synthesizing Fluorescent Proteins

Next we sought to determine if variational synthesis can increase the number of discoveries in downstream assays, as compared to MC synthesis. To simulate the results of realistic experimental assays, we used sequence-to-function predictors trained on large-scale experimental studies. We started with green fluorescent protein (GFP), predicting fluorescence using a transformer-based semi-supervised method trained on a GFP deep mutational scan dataset and evolutionary protein data[226,209]. We classified as hits sequences with predicted fluorescence above the functionality threshold specified by Sarkisyan et al. [226] (Section C.5.6). To construct a target p, we trained an unsupervised sequence model – an ICA model with MuE output, proposed in Weinstein & Marks [284] – on evolutionarily related GFP sequences, and then fixed the latent alignment variable of the MuE to generate sequences (Section C.5.1). Using a fixed latent alignment ensures that the fluorescence predictor,

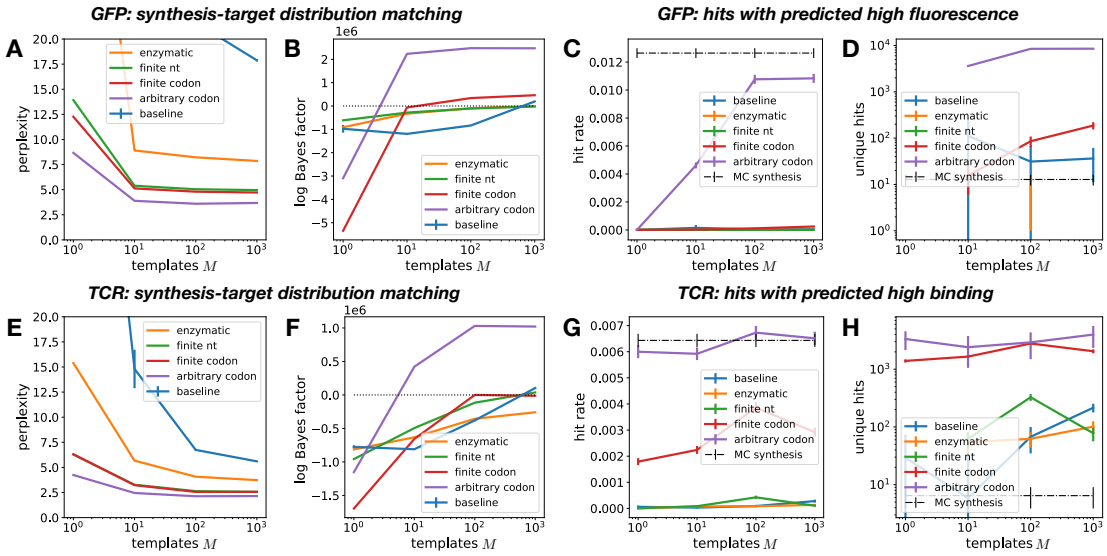


Figure 3.4: Perplexity (A) and two-sample test Bayes factor (B) for different synthesis methods applied to a target GFP model. (C) Hit rate for discovering functional sequences. (D) Expected number of unique hits in a $N_1 = 10^6$ library for variational synthesis, as compared to MC synthesis with a $N_0 = 10^3$ library (Section C.5.6). (E-H) Same as (A-D) for a target TCR model. Error estimates for each plot are described in detail in Section C.5.7.

which was only trained on fixed-length sequences, can be confidently applied. Note that the fluorescence predictor was not used to construct p itself, so we can fairly evaluate variational synthesis in the setting where the experimental results are not known ahead of time. In general, the fluorescence predictions are quite sensitive to the input sequence – a single amino acid change can abolish fluorescence – so generating new fluorescent sequences is nontrivial (Figure C.6). Only 1.3% of sequences sampled from p are hits, with fluorescence above the threshold specified by Sarkisyan et al. [226].

Stochastic synthesis models with arbitrary codon mixtures and fixed assembly have low perplexities, and can pass the two-sample test with large Bayes factors at $M \geq 10$; other methods struggle, including the baseline method (Figure 3.4AB). Samples from arbitrary codon models at $M = 10$ show average fluorescence similar to p (Figure C.8), and the fraction of samples that are hits is only

about half that of MC synthesis, 0.5% (Figure 3.4C). Meanwhile, alternative stochastic synthesis methods show hit rates below 0.05%.

Variational synthesis leads to a decrease in hit rate relative to MC synthesis, but this can be more than compensated for by the increase in the number of synthesized samples. If, for instance, $N_1 = 10^6$ sequences generated via variational synthesis are assayed, as opposed to $N_0 = 10^3$ sequences generated via MC synthesis, an estimated 3600 unique functional sequences will be discovered using variational synthesis as opposed to 10 for MC synthesis (Figure 3.4D; Section C.5.6). Variational synthesis can thus provide orders-of-magnitude increases in the number of hits in protein engineering applications, with the number of hits increasing with larger values of N_1 and/or M.

3.5.3 Synthesizing Antigen-Binding Proteins

Next we sought to evaluate the advantages of variational synthesis over MC synthesis in an application area important for human health. Understanding T cell receptor (TCR) sequences and their binding properties is crucial for understanding the immune response to infection or cancer, and engineering new TCRs with desired binding properties is crucial for immunotherapies[131]. We trained a model of TCR sequences from a healthy donor – an ICA model with MuE output – and fixed the latent alignment variable in the MuE to define p (Section C.5.1). As a held-out sequence-to-function predictor, we used Tcellmatch[77] to predict binding to an influenza epitope (Section C.5.6). The predictor is highly sensitive to the input sequence – a single amino acid change can abolish binding – making this a challenging problem for variational synthesis (Figure C.10). Only 0.6% of samples from the target p are hits.

Synthesis models with arbitrary codon mixtures and fixed assembly achieve low perplexities and can pass the two-sample test with large Bayes factors (Figure 3.4EF). Variational synthesis with this model achieves hit rates similar to MC synthesis (Figure 3.4G). MC synthesis with $N_0 = 10^3$ generates just 6 hits on average across independent libraries; given stochasticity, it is not unlikely to see no hits at all in a given library. Variational synthesis with $N_1 = 10^6$ and $M = 10$ generates an expected 2400 unique hits (Figure 3.4H). Similar results hold for additional epitopes, from other viruses (Section C.5.8). These results suggest that variational synthesis can dramatically accelerate the discovery of new TCRs that bind specific antigens, relying only on unsupervised sequence models and not large-scale supervised sequence-to-function training data.

Close matches between q_{θ^*} and p turn out to be unnecessary for reaching high hit rates in this example. When using arbitrary codon mixtures or finite codon mixtures with $M = 1$, or even using finite nucleotide mixtures with $M = 100$, the two-sample test detects significant differences between q_{θ^*} and p (Figure 3.4F), but nonetheless variational synthesis achieves substantially more hits than MC synthesis (Figure 3.4H).

3.6 Discussion

Variational synthesis trades accuracy for scale, producing large numbers of approximate samples from a target model rather than small numbers of exact samples, as in MC synthesis. When accuracy is high enough – when q_{θ^*} is sufficiently close to p – the payoff can be enormous, as the number of hits increases linearly with the number of assayed sequences N_1. Given that many high-throughput

screens can reach 10^{10} sequences or more, while individual gene synthesis rarely goes beyond $N_0 = 10^4$, using variational synthesis may make the difference between zero hits and a million.

We have shown through detailed simulations that such large payoffs are plausible for real, therapeutically important protein design targets, using commercially available stochastic synthesis technology. Going forward, implementing variational synthesis experimentally is thus a matter of ordering and assaying commercially-made libraries based on q_{θ^*}.

The key limitations of our variational synthesis methods – and opportunities for future work – stem from the challenges of matching synthesis and target distributions. First, our synthesis models (Section 3.2.1) are idealizations based on manufacturers' descriptions of the distribution of sequences their methods produce, but do not take into account possible errors, biases or limitations in the real procedure (Section C.1). Developing more accurate q_θ models, based on e.g. deep sequencing data, may be an important area for future work. Second, our methods for judging whether q_{θ^*} is sufficiently close to p are limited. Empirically, while the BEAR two-sample test appears to be excellent at distinguishing among good and bad fixed assembly models in the examples we studied, it struggles to detect the errors caused by combinatorial assembly, even when they are large enough to abolish function (Figure C.9). Theoretically, tighter bounds than that in Proposition 3.4.1 can be proved with total variation or Wasserstein distance in place of KL, but optimizing these alternative divergences directly is a challenge (Section C.4.2). For sequence-to-function predictors to be more reliable in evaluating variational synthesis methods, they must be robust to covariate shift, since switching from p to q_{θ^*} is, precisely, a covariate shift. Third, while our black-box optimization method allows for arbitrary target distributions p, it may be more effective in many cases to work

with p for which an exactly matching q_{θ^*} can be found analytically (Section C.2.1). Recent progress on mixture models as a competitor to deep generative neural network models make this approach especially promising[214].

Variational synthesis changes the calculus of what makes a successful generative sequence model and what makes a successful synthesis technology. If just 1% of the sequences sampled from an initial model A were functional, and 50% of sequences sampled from a proposed model B were functional, model B would be considered a major advance; however, if we could accurately match a stochastic synthesis protocol to model A and not to model B, then model A could easily lead to orders of magnitude more hits in practice. Meanwhile, the traditional goal of the DNA synthesis community has been large-scale individual synthesis. From a probabilistic perspective, however, it hardly makes sense to focus exclusively on methods to sample from mixtures of point masses. The recent development of methods to synthesize samples from much more flexible mixture models represents a major advance outside the traditional paradigm.

Variational synthesis bridges the gap between generative sequence models and stochastic synthesis technologies, providing a rigorous approach to experimental design. We are optimistic that it will help translate powerful new generative sequence models into laboratory discoveries.

4

Non-identifiability and Misspecification in Models of Fitness

Understanding the consequences of mutation for molecular fitness and function is a fundamental problem in biology. Recently, generative probabilistic models have emerged as a powerful tool for estimating fitness from evolutionary sequence data, with accuracy sufficient to predict both labora-

tory measurements of function and disease risk in humans, and to design novel functional proteins. Existing techniques rest on an assumed relationship between density estimation and fitness estimation, a relationship that we interrogate in this article. We prove that fitness is not identifiable from observational sequence data alone, placing fundamental limits on our ability to disentangle fitness landscapes from phylogenetic history. We show on real datasets that perfect density estimation in the limit of infinite data would, with high confidence, result in poor fitness estimation; the misspecification of current models is a blessing, rather than a curse, when it comes to fitness estimation. Our results challenge the conventional wisdom that bigger models trained on bigger datasets will inevitably lead to better fitness estimation, and suggest novel estimation strategies going forward.

This chapter presents work done in collaboration with Alan N. Amin, Jonathan Frazer and Debora S. Marks, and is currently in submission [281]. E.N.W. conceived the research, derived the theoretical results, contributed to the empirical results and wrote the paper. A.N.A. contributed equally to E.N.W. overall, and in particular contributed to the conception of the research and the theoretical results, and obtained the empirical results. J.F. contributed to the early conceptualization and preliminary experiments. D.S.M. supervised the research at all stages.

4.1 Introduction

The past decades have witnessed a tremendous increase in the scale of genome sequence data available from across life. Recently, methods for estimating molecular fitness using generative sequence models have seen widespread success at translating this evolutionary data into predictions of the

functional consequences of mutation. Such models have been shown to accurately predict the outcomes of experimental assays of protein function [110,215,173], and have been applied to infer 3D structures of RNA and protein [168,280] and to design novel proteins [235,224,165]. The models have also been used to predict whether human mutations are pathogenic, directly informing the diagnosis of genetic disease [80]. In this paper, we investigate how and why generative sequence models fit to evolutionary sequence data are successful at estimating molecular fitness, and how they might be improved and generalized going forward.

Existing approaches to fitness estimation with generative sequence models rest on an assumed relationship between density estimation and fitness estimation. Given a dataset of sequences $X_1, \ldots, X_N$, assumed to be drawn i.i.d. from some underlying distribution p_0, fitness models proceed by (1) fitting a probabilistic model q_θ to $X_{1:N}$ and (2) using the inferred density $\log q_{\hat{\theta}}(x) \approx \log p_0(x)$ as an estimate of the fitness $f(x)$ of a sequence x; this estimate in turn is used to predict other covariates such as whether the mutated sequence is pathogenic [110,215,80]. Innovation in fitness models has come out of a trend of building increasingly flexible models fit to increasing amounts of data: simple models that treat each column of a sequence alignment independently were improved by energy-based models that accounted for epistasis [110], which in turn were improved by deep variational autoencoders [215], which in turn were improved by deep autoregressive alignment-free models [235,165,173]. Naively, one might assume that these improvements have come from obtaining better and better estimates of the data distribution p_0, and improvements will continue with bigger models and bigger datasets. In this article, we argue that this presumption is incorrect.

4.1.1 Technical summary

First, we show that that the true data distribution p_0 may not reflect fitness, and argue instead that we should be focused on estimating another distribution that does, p^∞ (the "stationary distribution", to be defined below). In particular, we demonstrate that phylogenetic effects – i.e. the history of how current sequences evolved over time – can "distort" the observed data, leading to a situation where $p_0 \neq p^\infty$ (Sec. 4.2). Second, we show in this situation that p^∞ and fitness f are non-identifiable: even with infinite data, there always exists some alternative fitness function $\tilde{f}$ that explains the same data just as well as f. This sets fundamental limits on what we can learn about fitness from evolutionary data (Sec. 4.3). Third, although exact estimation of p^∞ is impossible, we show that it is still possible to get closer to p^∞ than p_0, that is, to find a better estimator of fitness than the true data density p_0. This can be done by fitting to data a parametric generative sequence model $\mathcal{M} = \{q_\theta : \theta \in \Theta\}$ that is (approximately) well-specified with respect to p^∞ (i.e. $p^\infty \in \mathcal{M}$) but *misspecified* with respect to the data distribution p_0 (i.e. $p_0 \notin \mathcal{M}$), thus illustrating the potential blessings of misspecification (Sec. 4.4). Fourth, we construct a hypothesis test to determine whether these blessings of misspecification occur on real data, with existing fitness estimation models; here, we rely on a Bayesian nonparametric sequence model to construct a credible set for p_0 (Sec. 4.6). Fifth, we apply our test to over 100 separate sequence datasets and fitness estimation tasks, to conclude that existing fitness estimation models systematically outperform the true data distribution p_0 at estimating fitness (Sec. 4.7). The takeaway is that better fitness estimation (i.e. better p^∞ estimation) will not come from better density estimation (i.e. better p_0 estimation); bigger models

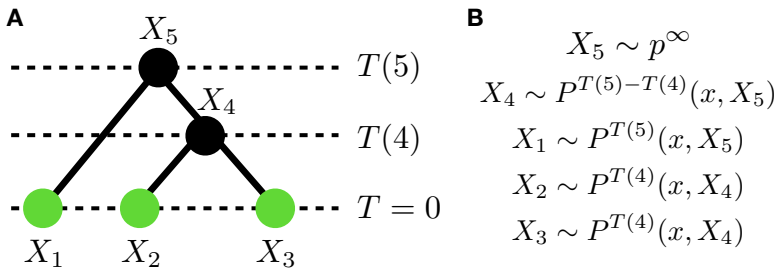


Figure 4.1: Example JFPM for $N = 3$ observed sequences. (A) An example phylogeny **H**. (B) Generative process for sequences at each node of the phylogeny.

and bigger datasets are not enough. Instead, better fitness estimation can come from developing models that describe p^{∞} better but the data density p_0 *worse*.

4.2 Models of fitness and phylogeny

In this section we show how p_0 may not accurately reflect the true fitness landscape, by developing a generative model of sequence evolution that takes into account both fitness and phylogeny. The model is general: it allows for arbitrarily complex epistatic fitness landscapes, and recovers standard generative phylogenetic and fitness models as special cases. Our concerns about the effects of phylogeny on fitness estimation are motivated by the widespread use – and trust – of phylogenetic models for evolutionary sequence data (phylogenetic models are far more widely applied than fitness models)[99,48,74,75]. Although often inferred from the very same datasets, standard fitness models and standard phylogeny models make conflicting assumptions, which our general framework makes explicit.

4.2.1 Joint fitness and phylogeny models

We define "joint fitness and phylogeny models (JFPMs)" using two elements: a description of how individual species (or populations or individuals) change over time, which depends on fitness f, and a description of the species' relationship to one another, a phylogeny $\mathbf{H}$. To describe the dynamics of individual species, let $P^\tau(x, x_0)$ denote the probability of sequence x_0 evolving into sequence x after time τ; in particular, $P^\tau(x, x_0)$ is assumed to be the transition probability of an irreducible continuous-time Markov chain defined over sequence space $\mathcal{X}$. For example, under neutral evolution (i.e. without selection based on fitness), $P^\tau(x, x_0)$ may follow a Jukes-Cantor model[75]. With selection, for simple population genetics models (e.g. Moran or Wright processes), Sella & Hirsh [230] demonstrate under general conditions that for any x_0,

$$P^\tau(x, x_0) \xrightarrow{\tau\to\infty} p^\infty = \frac{1}{\mathcal{Z}} \exp(\beta f(x)) \tag{4.1}$$

where $f(x)$ is the log fitness of the sequence x and $\beta > 0$ is a constant (Appx. D.1). The implication of Eqn. 4.1 is that the stationary distribution of the evolutionary dynamics follows a Boltzmann distribution, with energy proportional to the log fitness of the sequence. Estimating p^∞ is of interest because it provides a direct estimate of log fitness, up to a linear transform, since $f(x) = \beta^{-1}(\log p^\infty(x) + \log \mathcal{Z})$. (N.b. in the remainder of the paper, when we say "estimate fitness" we mean, implicitly, "estimate log fitness up to a linear transform".)

The sequences we observe, however, do not necessarily come from the stationary distribution. In-

stead, they are correlated with one another according to their evolutionary history. This is described by a phylogeny $\mathbf{H} = (V, E, T)$ consisting of a directed and rooted full binary tree with edges E and nodes V, along with time labels for the nodes, $T : V \to \mathbb{R}_+$ (Fig. 4.1A). Each node v is associated with a sequence X_v, drawn as $X_v \sim P^{\Delta t}(x, X_{v_0})$, where X_{v_0} is the sequence of the parent node, v is the child node, and $\Delta t = T(v_0) - T(v_1)$ is the length of the edge between them (Fig. 4.1B). The root sequence is drawn from p^∞. The observed datapoints $X_1, \ldots, X_N$ correspond to the leaf nodes. In general we will assume all leaves are observed at effectively the same time, the present day $T = 0$.

4.2.2 Simplifying assumptions

Standard probabilistic phylogenetic models ignore fitness and assume

Assumption 4.2.1 (Pure phylogeny models (PMs))**.** *There is no difference in fitness among sequences, i.e.* $f(x) = C$.

Example models that fit this form include most of those used in BEAST[61], MrBayes[114], RaxML[245], etc. Standard probabilistic fitness models, on the other hand, ignore phylogenetic history and assume that the stationary distribution has been reached,

Assumption 4.2.2 (Pure fitness models (FMs))**.** *Let* τ_i *be the distance in time between observed sequence* X_i *and its parent node. Take* $\tau_i \to \infty$ *for all* i*, which implies that*

$$X_i \overset{iid}{\sim} \frac{1}{\mathcal{Z}} \exp(\beta f(x)) \text{ for } i \in \{1, 2, \ldots\}. \tag{4.2}$$

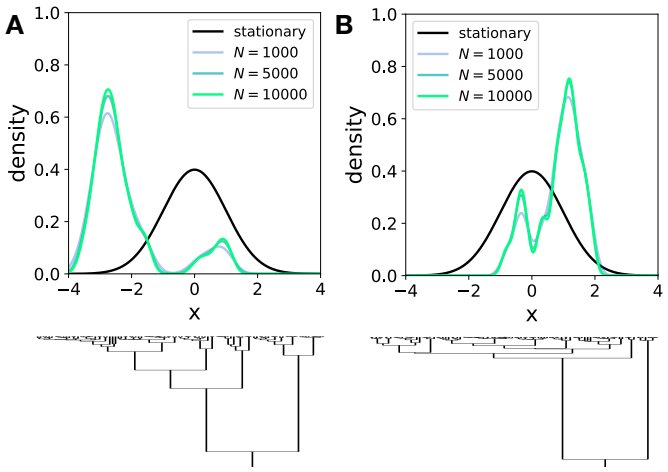


Figure 4.2: Samples from an OUT. (A) *Above:* Stationary distribution p^{∞} and kernel density estimates of the distribution of samples p_0 from an OUT model for increasing N. *Below:* A subset of the phylogeny. (B) Same as (A) for an independent sample of $\mathbf{H}$.

The key implication of this assumption is that density estimation and fitness estimation are linked: the data follows $X_1, \ldots, X_N \sim_{iid} p_0 = p^{\infty}$, and so if we can estimate p_0 we can estimate the fitness. Example models include EVMutation[110], DeepSequence[215], EVE[80], etc. Note although Assumptions 4.2.1 and 4.2.2 do not conflict directly, conclusions made based on them conflict in practice: PMs typically infer finite and different lengths for branches (i.e. $\tau_i < \infty$), while FMs typically infer differences in fitness (i.e. $f(x) \neq C$), even when applied to the same dataset.

4.2.3 1D Example

If Asm. 4.2.2 does *not* hold, then there is no reason for the distribution of observed sequences $X_1, X_2, \ldots$ to follow p^{∞}. We illustrate this with the most widely used example of a JFPM that does not use Assumptions 4.2.1 or 4.2.2: an Ornstein-Uhlenbeck tree (OUT) model[75,32]. In this model, X is continuous, i.e. $X \in \mathbb{R}$, and evolves on a quadratic fitness landscape of the form $f(x) \propto$

$(x-\mu)^2 + C$ according to the dynamics $P^\tau(x, x_0) = \text{Normal}\left(x_0 e^{-\frac{1}{2}\tau} + \mu, \sigma^2(1-e^{-\tau})\right)$. The stationary distribution p^∞ is Normal(μ, σ^2). One can show (Appx. D.2.1) that for any phylogeny **H**,

Proposition 4.2.3 (OUT observations)**.** *The distribution of observed genotypes $X_{1:N}$ is drawn from a multivariate normal distribution with mean $\mu\vec{1}_N$ and covariance Σ where*

$$\Sigma_{ij} := \sigma^2 \exp(-\frac{1}{2} t_{ij}(\mathbf{H}))) \text{ for } i, j \in \{1, \ldots, N\}, \tag{4.3}$$

and $t_{ij}(\mathbf{H})$ is the total time of the shortest path between leaves i and j along the phylogeny **H**.

We drew samples from the OUT with a Kingman coalescent prior on **H** (Bertoin [22], Def. 2.1) and plotted their density (Fig. 4.2A). Even as $N \to \infty$, the distribution of samples does not follow p^∞. Moreover, rerunning the process with a new sample from the prior yields a very different distribution of samples (Fig. 4.2B).

4.3 Non-identifiability

In this section we investigate whether, given infinite sequence data, it is possible to infer fitness f without Asm. 4.2.2, and conversely, whether it is possible to infer phylogeny **H** without Asm. 4.2.1. That is, we are interested in whether fitness and phylogeny are identifiable in JFPMs. We conclude they are not: given infinite data generated with any f and **H**, there exists some alternative $\tilde{f}$ and $\tilde{\mathbf{H}}$, where $\tilde{\mathbf{H}}$ satisfies Asm. 4.2.2, that explains the data equally well.

Naively, this result may be surprising: in FMs, each sequence is drawn independently, i.e. $X_i \perp\!\!\!\perp X_j | \mathbf{H}, f$, while in JFPMs and PMs there is (in general) correlation between sequences, i.e. $X_i \not\perp\!\!\!\perp X_j | \mathbf{H}, f$. One might then hope that examining correlations between sequences would enable us to infer whether Asm. 4.2.2 holds. However, we can show that these correlations are uninformative due to a symmetry in phylogenetic models, exchangeability.

Assumption 4.3.1 (Exchangeability)**.** *Let $m(X_1, X_2, \ldots)$ denote the marginal probability of an infinite set of sequences $X_1, X_2, \ldots$ integrating over all phylogenies, i.e. $m(X_1, X_2, \ldots) = \int p(X_1, X_2, \ldots | \mathbf{H}) p(\mathbf{H}) d\mathbf{H}$. Then, for any permutation π of the integers,*

$$m(X_1, X_2, \ldots) = m(X_{\pi(1)}, X_{\pi(2)}, \ldots). \tag{4.4}$$

Exchangeability says that if we had observed the sequences in a different order, it would not change their probability. In general, models of sequences observed at the same time (i.e. the present day, $T = 0$) satisfy exchangeability; for instance, models with a Kingman coalescent prior are exchangeable[22,61]. Exchangeability implies that fitness and phylogeny are not identifiable. In particular, even if $X_1, X_2, \ldots$ are generated from a JFPM with a finite branch length phylogeny $\mathbf{H}$, we can describe the same data just as well using a model with an infinite branch length phylogeny $\tilde{\mathbf{H}}$ (an FM):

Theorem 4.3.2 (Non-identifiability)**.** *Assume $X_1, X_2, \ldots$ satisfy Assumption 4.3.1. Then with*

probability 1 there exists some function $\tilde{f}$ such that

$$X_i \overset{iid}{\sim} p_0 = \frac{1}{\tilde{\mathcal{Z}}} \exp(\beta \log \tilde{f}(x)) \text{ for } i \in \{1, 2, \ldots\}.$$

Proof. Applying de Finetti's Theorem (Kallenberg [133], Thm. 11.10), there almost surely exists a random measure G such that for $i \in \{1, 2, ...\}$, $X_i \overset{iid}{\sim} G$. Let $p_G(x)$ be the pmf of G (we assume x is a finite discrete sequence; we can also work with continuous genotypes assuming the pdf $p_G(x)$ exists). Set $\tilde{f}(x) = [p_G(x)]^{1/\beta}$. □

This result says that the observed sequences from an exchangeable JFPM, $X_1, X_2, \ldots$, are precisely i.i.d. samples from some p_0. Although in the standard tree representation $X_i \not\perp\!\!\!\perp X_j | \mathbf{H}, f$, there must be some alternative description of the same process where $X_i \perp\!\!\!\perp X_j | \tilde{\mathbf{H}}, \tilde{f}$. Fitness and phylogeny are thus non-identifiable: data generated from a JFPM with fitness f and phylogeny $\mathbf{H}$ can be described just as well using $\tilde{f}$ and $\tilde{\mathbf{H}}$, and vice versa.

The biological intuition behind Thm. 4.3.2 is that if two sequences are similar to each other and distant from a third, they may be similar either because they are closely related (i.e. the distance τ to the most recent common ancestor is small) or because they are in a local maxima of the fitness landscape. Without further assumptions, we cannot tell the difference between these two explanations. The machine learning intuition is that evolution, as described by a JFPM, is in effect a Markov chain Monte Carlo process whose stationary distribution gives the fitness. However, the samples we observe may not be fully independent: each pair of samples was initialized from the same point (the most recent common ancestor), and the burn-in since that point may not be sufficiently long. With-

out independent samples, our estimate of the stationary distribution will be biased.

4.3.1 Fitness inference as hyperparameter inference

While general, Thm. 4.3.2 is not constructive, and does not tell us what the distribution p_0 actually is, or how exactly it differs from p^∞. Thm. 4.3.2 leaves unclear how much we need to know to learn the fitness landscape: could we infer fitness f if we knew the parametric form of p^∞, i.e. if we had some model $\mathcal{M}$ and knew that $p^\infty \in \mathcal{M}$? What if we also knew the underlying phylogeny $\mathbf{H}$? In the long branch limit (Asm. 4.2.2), fitness is identifiable if $\mathbf{H}$ is known; if $\mathcal{M}$ is also known, learning fitness is a matter of inferring model parameters. In the limit where all the branch lengths in the phylogeny are zero, the distribution of observations from a JFPM reduces to $X_1 \sim p_\infty$ and $X_1 = X_2 = X_3 = \ldots$. Here fitness is non-identifiable even if $\mathbf{H}$ and $\mathcal{M}$ are known; learning fitness is a matter of learning from a single sample. In the realistic intermediate branch length case, if $\mathbf{H}$ and $\mathcal{M}$ are known, we will show that learning fitness is essentially a matter of *hyperparameter* rather than *parameter* inference.

We demonstrate this last claim by approximating OUTs as Gaussian process latent variable models (GPLVMs), finding that fitness only appears as a hyperparameter of the GP. The GPLVMs have latent variables $Z_1, Z_2, \ldots$ that lie on the hyperbolic plane $\mathbb{H}$, and use the Gaussian process kernel $k(\cdot,\cdot) = \exp(-d(\cdot,\cdot))$, where $d(\cdot,\cdot)$ is a distance metric over $\mathbb{H}$. Let $\mathcal{W}_1(\cdot,\cdot)$ be the Wasserstein metric for distributions over infinite matrices, i.e. over $\mathbb{R}^{\infty\times\infty}$, using the sup norm on matrices.

Theorem 4.3.3 (GPLVM approximation of OUT)**.** *Assume a prior over phylogenies* $\mathbf{H}$ *that is exchangeable in its leaves and where the minimum time between any pair of nodes is greater than* $\eta > 0$

with probability 1. Define the leaf distance matrix $\nu_{ij} = \log(\frac{1}{2}\, t_{ij}(\mathbf{H}))$. *For any* $\epsilon > 0$, *there exists a.s. a GPLVM of the form,*

$$\begin{aligned} &G \sim \mathcal{G}, \qquad s \sim \text{GaussianProcess}(\mu, \sigma^2 k(\cdot,\cdot)), \\ &Z_i \overset{iid}{\sim} G \ \textit{ for } i \in \{1, 2, \ldots\}, \\ &X_i = s(Z_i), \end{aligned} \tag{4.5}$$

where G *is a random measure over* $\mathbb{H}$, *such that* $\mathcal{W}_1(p(\nu), p(\tilde{\nu})) < \epsilon$, *where* $\tilde{\nu}_{ij} = \log(d(Z_i, Z_j))$.

If $\mathcal{W}_1(p(\nu), p(\tilde{\nu})) = 0$, *the OUT and GPLVM produce identical distributions over* $X_1, X_2, \ldots$ *a.e..*

The proof is in Appx. D.2.2, and uses the embedding of Sarkar [225]. This result says that, by embedding phylogenies $\mathbf{H}$ in a metric space, we can approximate an OUT arbitrarily well with a GPLVM; as the Wasserstein bound gets smaller, the distribution of covariance matrices of the two models get closer. In the GPLVM, the observations are conditionally independent, $X_i \perp\!\!\!\perp X_j | s, G$, in line with Thm. 4.3.2. The phylogeny $\mathbf{H}$ enters the GPLVM only through the latent space embedding $Z_1, Z_2, \ldots$. Learning phylogeny, given the fitness landscape, is thus essentially a matter of inferring latent variables [215,57]. The fitness landscape enters the GPLVM only through the prior on the Gaussian process (i.e. through μ and σ). Inferring fitness given phylogeny is thus essentially a matter of inferring hyperparameters. This is both good and bad news for fitness inference. On the one hand, hyperparameters are often learned in practice, and doing so can yield substantially better predictions, so we should be able learn something about μ and σ given data (Williams &

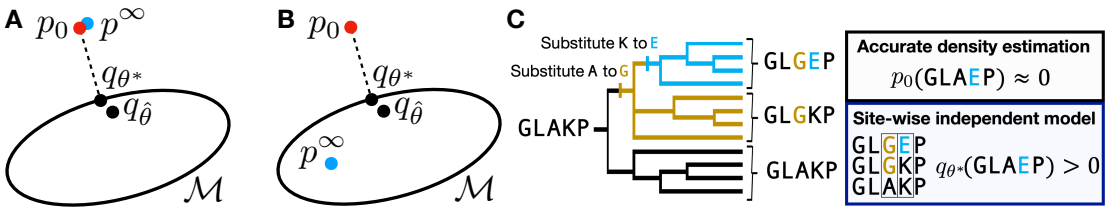


Figure 4.3: Alternative explanations for the success of fitness estimation methods. (A) Setup in which hypothesis 1 would hold true. (B) Setup in which hypothesis 2 would hold true. (C) Biological intuition for the blessings of misspecification (Hypothesis 2).

Rasmussen [289], Chap. 5). On the other hand, hyperparameters are in general (though not always) non-identifiable, and therefore so is fitness [167]. Ho & Ané [104] describe non-identifiability conditions for the OUT in particular. We conclude that even when $\mathbf{H}$ and $\mathcal{M}$ are known, fitness inference in JFPMs is fundamentally challenging.

4.4 Blessings of misspecification

We have demonstrated that phylogenetic effects can produce a data distribution p_0 that is not equal to the stationary distribution p^∞, and exact inference of p^∞ is in general impossible even with infinite data. Nonetheless, the practical success of fitness estimation methods suggest it is possible to at least approximate p^∞ from observational sequence data. Recall that existing methods proceed by fitting a probabilistic model $q_\theta \in \mathcal{M} = \{q_\theta : \theta \in \Theta\}$ to data $X_{1:N}$, typically via maximum likelihood estimation or approximate Bayesian inference, and then using the predicted log density $\log q_{\hat{\theta}}(x)$ as an estimate of the fitness of a sequence x. Why does this approach provide empirically successful estimates of p^∞? In this section we consider two hypotheses, either of which may hold true in theory. In Secs. 4.6-4.7 we develop and apply tests to evaluate them on real data.

Hypothesis #1 (informal). *Fitness estimation methods succeed by finding $q_{\hat{\theta}} \approx p_0$, since for all practical purposes on real data, $p_0 = p^\infty$.*

This hypothesis would make sense if Asm. 4.2.2 held, i.e. branch lengths were long enough in real datasets for $P^{\tau_i}(x, x_0)$ to be close to its stationary distribution. Under this explanation, better density estimators have been, and will continue to be, better fitness estimators. We should focus on developing models $\mathcal{M}$ that are well-specified with respect to the data, i.e. $p_0 \in \mathcal{M}$ (Fig. 4.3A).

Hypothesis #2 (informal). *Fitness estimation methods succeed by using models $\mathcal{M}$ that are misspecified with respect to p_0, i.e. $p_0 \notin \mathcal{M}$. The inferred model $q_{\hat{\theta}}$ is then closer to p^∞ than p_0 itself.*

To show this hypothesis is plausible, we prove that it is guaranteed to hold under general conditions. We study the projection of p_0 onto $\mathcal{M}$ via the Kullback-Leibler (KL) divergence, $q_{\theta^*} = \operatorname{argmin}_{q_\theta \in \mathcal{M}} \mathrm{KL}(p_0 \| q_\theta)$. The KL projection is relevant because maximum likelihood estimation minimizes the approximate KL divergence between the data and the model, and the posterior in Bayesian inference asymptotically concentrates around the maximum likelihood estimator [176]. We thus expect the fit model $q_{\hat{\theta}}$ to be close to q_{θ^*}, and get closer with N. Assume that $\mathcal{M}$ is "log-convex", meaning that for any $\theta, \theta' \in \Theta$ and $0 < r < 1$, there exists some θ'' such that $q_{\theta''}(x) = q_\theta(x)^r q_{\theta'}(x)^{1-r} / \sum_x q_\theta(x)^r q_{\theta'}(x)^{1-r}$; examples of log-convex models include the Potts model, as well as all other exponential family models.

Theorem 4.4.1 (Blessings of misspecification)**.** *Assume that the model $\mathcal{M}$ is log-convex and well-specified with respect to the stationary distribution, i.e. $p^\infty \in \mathcal{M}$. Assume q_{θ^*} exists and is unique.*

Then, if the model is misspecified with respect to the data distribution, i.e. $p_0 \notin \mathcal{M}$*, we have*

$$\textsc{kl}(q_{\theta^*} \| p^\infty) < \textsc{kl}(p_0 \| q_{\theta^*}) + \textsc{kl}(q_{\theta^*} \| p^\infty) \leq \textsc{kl}(p_0 \| p^\infty). \tag{4.6}$$

But if the model is well-specified, i.e. $p_0 \in \mathcal{M}$*, we have*

$$\textsc{kl}(q_{\theta^*} \| p^\infty) = \textsc{kl}(p_0 \| p^\infty). \tag{4.7}$$

Proof. For part 1, apply Thm. 1 from Csiszar & Matus [46]. For part 2, note that $q_{\theta*} = p_0$ when $p_0 \in \mathcal{M}$. □

In words, the model projection $q_{\theta*}$ is closer to p^∞ than p_0 so long as as the model $\mathcal{M}$ is misspecified with respect to p_0 (Fig. 4.3B). To understand the biological intuition behind this result, consider a situation where two neutral mutations with no effect on fitness occur successively at different sites (Fig. 4.3C). Due to phylogenetic correlation, there is no observed sequence x^* in which the second mutation is present but not the first, so an accurate density estimator will find $p_0(x^*) \approx 0$. However, if we can guess correctly that the fitness landscape is independent across sites, then fitting a site-wise independent model $\mathcal{M}$ will imply the mutation is allowed, $q_{\theta^*}(x^*) > 0$, correctly inferring $p^\infty(x^*) > 0$.

Under Hypothesis 2, progress in the field of fitness estimation has *not* come from building better density estimators (Hypothesis 1), but rather from an iterative process of (1) hypothesizing, based partly on biophysical knowledge, models that are (approximately) well-specified with respect to p^∞

but not too flexible, such that $p_0 \notin \mathcal{M}$ and then (2) comparing their density estimates against experimental fitness measurements. We will show that on real data, Hypothesis 1 can often be rejected in favor of Hypothesis 2.

4.5 Related work

Efforts to account for the effects of phylogeny in fitness estimation have a long history [149]. Practical generative sequence models that explicitly account for both epistatic fitness landscapes and phylogeny have long been sought, but stymied primarily by computational challenges [122,220]. In their place, a variety of non-generative (and often heuristic) methods for correcting for phylogeny have been proposed, including data reweighting schemes [168,220], data segmentation schemes [43], post-inference parameter adjustments [65], covariance matrix denoising methods [205], simulation based statistical testing [216], and more. In this article, we show that deconvolving fitness and phylogeny is not just computationally hard, but also in general statistically impossible: fitness and phylogeny are non-identifiable. We further show that use of a misspecified parametric model can on its own (without further corrections) partially adjust for phylogenetic effects.

Our results also intersect with the literature on robust statistics: we can think of the observed data distribution p_0 as a "distorted" version of the true distribution of interest p^∞. However, in typical robust inference frameworks (e.g. Huber's epsilon contamination model), the observed distribution differs from the true distribution by the addition of outliers [113,246]. In our setup, on the other hand, inliers are deleted, as phylogenetic correlations can mean the effective support of p_0 is

smaller than that of p^∞ (Fig. 4.2).

4.6 Diagnostic method

In this section, we develop diagnostic methods to discriminate between Hypothesis 1 and Hypothesis 2 (Sec. 4.4) based on observational sequence data and experimental fitness measurements, and validate these diagnostics in simulation. Recall that under Hypothesis 2, the estimate $q_{\hat{\theta}}$ from a parametric fitness model is a better estimate of fitness than the true data density p_0, while under Hypothesis 1, p_0 is better. Discriminating these two hypotheses on real data is nontrivial because we do not have access to p_0. Ideally, then, a diagnostic test would evaluate the probability that the true density p_0 outperforms $q_{\hat{\theta}}$ at predicting fitness, taking into account uncertainty in what p_0 could actually be, given the data. To accomplish this, we compute a posterior over p_0 using a Bayesian nonparametric sequence model. In particular, we apply the Bayesian embedded autoregressive (BEAR) model, which can be scaled to terabytes of data and satisfies posterior consistency (Amin et al. [12], Thm. 35):

Theorem 4.6.1 (Summary of BEAR posterior consistency)**.** *Assume p_0 is subexponential, i.e. for some $t > 0$, $\mathbb{E}_{X\sim p_0}[\exp(t|X|)] < \infty$, where $|X|$ is the length of sequence X. Assume the conditions on the prior detailed in Amin et al. [12]. If $X_1, X_2, \ldots \sim p_0$ i.i.d, then for $M > 0$ sufficiently large and $\epsilon \in (0, 1/2)$ sufficiently small,*

$$\Pi_{\text{BEAR}}(B(p_0, MN^{-\epsilon})|X_{1:N}) \xrightarrow{N\to\infty} 1$$

in probability, where $B(p, r)$ is a Hellinger ball of radius r centered at p, and $\Pi_{\text{BEAR}}(\cdot|X_{1:N})$ is the

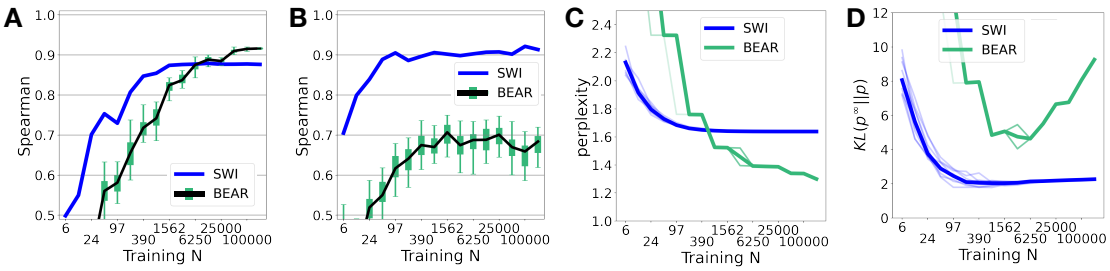


Figure 4.4: The BEAR diagnostic applied to simulated data. (A) Scenario 1. Spearman correlation between the maximum likelihood SWI model and the true fitness $\mathcal{S}_f(q_{\hat{\theta}})$, compared to the BEAR posterior distribution over $\mathcal{S}_f(p)$. Quantiles and 95% credible interval are shown with the green box and whisker plot. Points above (below) the whiskers correspond to SWI models that significantly outperform (underperform) the true data distribution. (B) Same as A, for Scenario 2. (C) Perplexity on heldout data of the BEAR and the SWI models in Scenario 2. Thick line corresponds to the average over 10 individual simulations (thin lines). (D) Same as C, comparing the KL divergence to p^∞.

BEAR posterior.

Crucially, this result implies that the BEAR posterior will converge to effectively any value of p_0, no matter what p_0 is (unlike a parametric model's posterior). Moreover, BEAR quantifies uncertainty in its estimates, giving the range of possible values of p_0 that are consistent with the evidence.

We construct our diagnostic test by comparing the fitness estimation performance of $q_{\hat{\theta}}$ to the range of possible performances of p_0 estimated by BEAR. Let $\mathcal{S}_f(p)$ be a scalar score evaluating how accurately a density p predicts fitness f. In practice, $\mathcal{S}_f$ will be based on experimental and clinical measurements of quantities directly related to fitness.

Diagnostic test (Test Hypothesis 1 vs. Hypothesis 2.) *Hypothesis 1* $\mathcal{H}_1 : \mathcal{S}_f(q_{\hat{\theta}}) < \mathcal{S}_f(p_0)$. *Hypothesis 2* $\mathcal{H}_2 : \mathcal{S}_f(q_{\hat{\theta}}) > \mathcal{S}_f(p_0)$. *Accept Hypothesis 2 at significance level* $\alpha > 0$ *if*

$$\Pi_{\text{BEAR}}(\mathcal{S}_f(q_{\hat{\theta}}) > \mathcal{S}_f(p)|X_{1:N}) > 1 - \alpha. \tag{4.8}$$

Accept Hypothesis 1 at significance level α if

$$\Pi_{\text{BEAR}}(\mathcal{S}_f(q_{\hat{\theta}}) < \mathcal{S}_f(p)|X_{1:N}) > 1 - \alpha. \tag{4.9}$$

So long as $\mathcal{S}_f(p)$ is a well-behaved function of p (in particular, so long as $\mathcal{S}_f$ is continuous in a neighborhood of p_0 with respect to the topology of convergence in total variation), Thm. 4.6.1 implies that this diagnostic test will be asymptotically consistent, in the sense that it converges to the correct hypothesis in probability.

4.6.1 Simulations

We next evaluate the performance of our diagnostic test on simulated data. We considered two scenarios, the first in which Hypothesis 1 holds, and the second in which Hypothesis 2 holds. In both, we let $\mathcal{M}$ be a site-wise independent (SWI) model, in which each position of the sequence is drawn independently, i.e. $X_l \sim \text{Categorical}(v_l)$ for $l \in \{1, \ldots, |X|\}$. The parameter v_l is in the simplex Δ_B, where $B+1$ is the alphabet size. (Further details in Appx. D.3.) In Scenario 1, the true data are generated according to a Potts model and $p_0 = p^\infty$. In this scenario, the SWI model is misspecified, and misspecification is *bad*: using a more flexible model will produce an asymptotically more accurate estimate of p^∞. We find that our diagnostic test asymptotically correctly accepts Hypothesis 1, in line with Thm. 4.6.1 (Figs. 4.4A and D.3A). In Scenario 2, the true data are generated according to a JFPM with finite branch lengths, and $p^\infty \in \mathcal{M}$ while $p_0 \notin \mathcal{M}$. The mutational dynamics P^τ follow the Sella & Hirsh [230] process. The phylogeny $\mathcal{H}$ is drawn from a Kingman coalescent. In

this scenario, the SWI model is again misspecified, but misspecification is *good*: while the nonparametric BEAR model can achieve better density estimates than the SWI model (Fig. 4.4C), the SWI model outperforms BEAR at fitness estimation (Figs. 4.4D and D.4). We find that our diagnostic test correctly accepts Hypothesis 2 (Figs. 4.4B and D.3B).

A possible point of concern is that the test is poorly calibrated from a frequentist perspective, and in the low N regime accepts Hypothesis 2 in Scenario 1 more than $100\alpha\%$ of the time when the data is resampled from p_0 (Fig. D.5A). This behavior is common in nonparametric Bayesian tests, and not necessarily a problem: the test is still valid from a purely Bayesian perspective. Nevertheless, on real data we will check that we are close to the large N regime by (1) checking that the BEAR posterior predictive is at least as close to p_0 as $q_{\hat{\theta}}$ is (as measured by perplexity on held out data; Figs. 4.4C and D.5B) and (2) examining the plot of the BEAR posterior over $\mathcal{S}_f(p)$ as a function of N (as in Fig. 4.4AB), to check that it has converged.

4.7 Empirical results

We now evaluate whether existing fitness estimation methods outperform the true data density p_0, i.e. whether we can reject Hypothesis 1 in favor of Hypothesis 2 on real data.

4.7.1 Tasks

We consider two key prediction tasks where fitness models are applied in practice. The first task is to predict whether variants of a protein are functional, according to an experimental assay of protein function; the metric $\mathcal{S}_f(\cdot)$ is the Spearman correlation between $p(x)$ and the assay result[110]. There

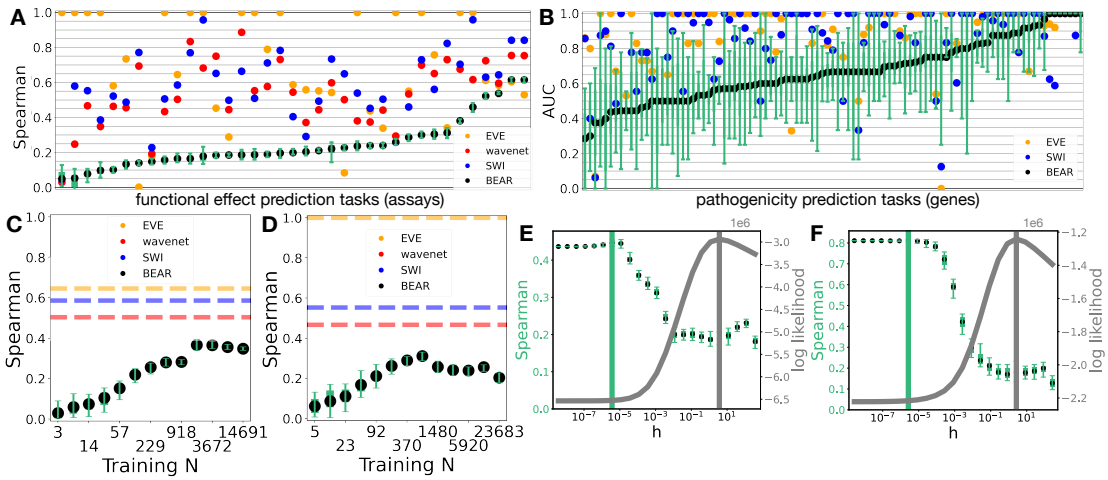


Figure 4.5: Fitness estimation models systematically outperform the data distribution. (A) Results for the first prediction task, predicting functional measurements in experimental assays. Quantiles and 95% credible interval of the BEAR posterior are shown with the green box and whisker plot. Points above (below) the whiskers correspond to fitness estimation models that significantly outperform (underperform) the true data distribution. (B) Results for the second prediction task, predicting variant pathogenicity in human genes. (C) Convergence of the BEAR posterior with datapoints N, for an example assay (β-lactamase). (D) Same as C, for another example assay (TIM barrel). (E) BEAR posterior Spearman (black and green) versus BEAR log likelihood (gray), interpolating between parametric and nonparametric regimes (low and high h), for an example assay (another β-lactamase assay). Peak Spearman indicated with vertical green line, peak log likelihood with gray. (F) Same as E, for another example assay (GAL4 DNA-binding domain).

are typically ∼1000s of measurements per assay. The second task is to predict whether a variant of a protein observed in humans causes disease, according to clinical annotations; the metric $\mathcal{S}_f(\cdot)$ is the area under the ROC curve when $p(x)$ is used to predict whether or not a variant is pathogenic[80]. There are typically only a handful of labels for each gene. For the first task, we considered 37 different assays across 32 different protein families, and for the second task, 97 genes across 87 protein families; for each protein family, we assembled datasets of evolutionarily related sequences, following previous work. Note that across the 37 assays and 97 genes, the data used for $\mathcal{S}_f$ comes from different experiments and different clinical evidence, often collected by different laboratories or doctors. As a consequence, our overall conclusions should be robust to the choice of $\mathcal{S}_f$.

4.7.2 Models

We considered three existing fitness estimation models: a site-wise independent model (SWI), a Bayesian variational autoencoder (EVE[80], which is similar to DeepSequence[215]), and a deep autoregressive model (Wavenet)[235]. Note that SWI and EVE, unlike Wavenet, require aligned sequences as training data. Details in Appx. D.4.

4.7.3 Results

Applied to the first prediction task, our diagnostic test accepts Hypothesis 2 at significance level $\alpha = 0.025$ in 35/37 assays (95%) for SWI, 33/37 assays (89%) for EVE, and 36/37 assays (97%) for Wavenet (Fig. 4.5A). Applied to the second prediction task, our diagnostic test accepts Hypothesis 2 at significance level $\alpha = 0.025$ in 31/97 genes (32%) for SWI and 46/97 genes (47%) for EVE

(Fig. 4.5B). Thus, fitness estimation models are capable of outperforming the true data distribution p_0. We found evidence for Hypothesis 1 in only a handful of examples: on the first task, Hypothesis 1 was accepted at significance level $\alpha = 0.025$ in 0/37 assays for SWI, 3/37 assays (8%) for EVE, and 0/37 assays for Wavenet, while on the second task, Hypothesis 1 was accepted for 5/97 genes (5%) for SWI and 4/97 genes (4%) for EVE. We confirmed that the diagnostic test was in the large N regime: BEAR outperformed Wavenet at density estimation, providing better predictive performance on 27/37 assays (73%) and similar performance on the remaining 10 assays (Fig. D.6).* Example plots of the BEAR posterior's convergence with N on the first prediction task showed convergence to values of $\mathcal{S}_f$ well below that for parametric fitness estimation models (Figs. 4.5C and D.7-D.8). Overall, we conclude that there is strong evidence that existing fitness estimation methods reliably outperform the true data distribution p_0 across a range of datasets and tasks.

To study the tradeoffs between density estimation and fitness estimation in more depth, we smoothly and nonparametrically relaxed a parametric autoregressive (AR) model (Appx. D.4.4). We embedded the AR model (a convolutional neural network) into a BEAR model, and fit the BEAR model with empirical Bayes. We found evidence that the AR model was misspecified on every dataset, following the methodology of Amin et al. [12]: the optimal h selected by empirical Bayes was on the order of $1 - 10$ in each dataset. Now, in the limit as the hyperparameter $h \to 0$, the BEAR model collapses to its embedded AR model; so by scanning h from low to high values we can interpolate between the parametric and nonparametric regime. We find a smooth tradeoff between $\mathcal{S}_f(p)$ and the likelihood of the data under the BEAR model, with higher h corresponding to better

*Note that we cannot do this comparison for SWI or EVE since they are alignment-based [28,84].

density estimation but worse fitness estimation (Fig. 4.5EF and D.9). This relationship held across many datasets: the diagnostic test, evaluated against the AR model (the $h \to 0$ limit), accepts Hypothesis 2 in 28/37 assays (76%), but Hypothesis 1 in only 6/37 (16%) (Fig. D.10). These results confirm that making a model well-specified (relaxing from a parametric to a nonparametric model) can bring improved density estimation at the cost of worse fitness estimation.

4.8 Discussion

In this article, we have argued that better density estimation does not necessarily lead to better fitness estimation. Our results changes the outlook for the future of fitness estimation: the common narrative that progress is inevitable through ever bigger models trained on ever bigger datasets appears to be false. Instead, progress will likely demand more fundamental methodological advances.

One future direction is to improve the current strategy of fitting misspecified models. For instance, it may be worthwhile to explore models that are *less* flexible than existing models and *worse* at density estimation, since they can increase the gap between $\mathrm{KL}(q_{\theta^*} \| p^\infty)$ and $\mathrm{KL}(p_0 \| p^\infty)$ (Thm. 4.4.1). Another option is to improve the geometry of the model: while exponential family models are guaranteed to be log-convex (and thus can satisfy Thm. 4.4.1), we have no such guarantee for variational autoencoders or other neural network methods. Finally, uncertainty quantification is crucial for applications such as those in clinical genetics, but challenging in misspecified models[253,177,117]. Another future direction is to construct scalable JFPM models and carefully handle non-identifiability. Recent progress on amortized variational inference for phylogenetic models is promising[275]. Non-

identifiability is more challenging, and may require new assumptions and/or new methods of sensitivity analysis to infer the full set of fitness landscapes consistent with the data.

Finally, although this article has focused on technological applications of fitness models in solving prediction problems, fitness models also have implications for our fundamental understanding of evolution. Pure phylogeny models and pure fitness models present very different pictures of the past history of life: in PMs, similarities and differences among genetic sequences are determined primarily by history and ancestry (Asm. 4.2.1), while in FMs they are primarily determined by functional constraints (Asm. 4.2.2). PMs and FMs also present very different implications for the future of life: in PMs, the diversity of sequences seen in nature will likely expand dramatically going forward, while in FMs, the landscape of functional sequences has already been well-explored. Our results emphasize that where and to what extent each model offers an accurate picture of reality remains an open question.

5

Bayesian Data Selection

Insights into complex, high-dimensional data can be obtained by discovering features of the data that match or do not match a model of interest. To formalize this task, we introduce the "data selection" problem: finding a lower-dimensional statistic—such as a subset of variables—that is well fit by a given parametric model of interest. A fully Bayesian approach to data selection would be to parametrically model the value of the statistic, nonparametrically model the remaining "back-

ground" components of the data, and perform standard Bayesian model selection for the choice of statistic. However, fitting a nonparametric model to high-dimensional data tends to be highly inefficient, statistically and computationally. We propose a novel score for performing both data selection and model selection, the "Stein volume criterion", that takes the form of a generalized marginal likelihood with a kernelized Stein discrepancy in place of the Kullback–Leibler divergence. The Stein volume criterion does not require one to fit or even specify a nonparametric background model, making it straightforward to compute — in many cases it is as simple as fitting the parametric model of interest with an alternative objective function. We prove that the Stein volume criterion is consistent for both data selection and model selection, and we establish consistency and asymptotic normality (Bernstein–von Mises) of the corresponding generalized posterior on parameters. We validate our method in simulation and apply it to the analysis of single-cell RNA sequencing datasets using probabilistic principal components analysis and a spin glass model of gene regulation.

This chapter presents work with Jeffrey W. Miller, and is currently in submission[285]. E.N.W. and J.W.M. conceived the idea; E.N.W. performed the research, under the supervision of J.W.M.; E.N.W. and J.W.M. wrote the paper.

5.1 Introduction

Scientists often seek to understand complex phenomena by developing working models for various special cases and subsets. Thus, when faced with a large complex dataset, a natural question to ask is where and when a given working model applies. We formalize this question statistically by saying

that given a high-dimensional dataset, we want to identify a lower-dimensional statistic—such as a subset of variables—that follows a parametric model of interest (the working model). We refer to this problem as "data selection", in counterpoint to model selection, since it requires selecting the aspect of the data to which a given model applies.

For example, early studies of single-cell RNA expression showed that the expression of individual genes was often bistable, which suggests that the system of cellular gene expression might be described with the theory of interacting bistable systems, or spin glasses, with each gene a separate spin and each cell a separate observation. While it seems implausible that such a model would hold in full generality, it is quite possible that there are subsets of genes for which the spin glass model is a reasonable approximation to reality. Finding such subsets of genes is a data selection problem. In general, a good data selection method would enable one to (a) discover interesting phenomena in complex datasets, (b) identify precisely where naive application of the working model to the full dataset goes wrong, and (c) evaluate the robustness of inferences made with the working model.

Perhaps the most natural Bayesian approach to data selection is to employ a semi-parametric joint model, using the parametric model of interest for the low-dimensional statistic (the "foreground") and using a flexible nonparametric model to explain all other aspects of the data (the "background"). Then, to infer where the foreground model applies, one would perform standard Bayesian model selection across different choices of the foreground statistic. However, this is computationally challenging due to the need to integrate over the nonparametric model for each choice of foreground statistic, making this approach quite difficult in practice. A natural frequentist approach to data selection would be to perform a goodness-of-fit test for each choice of foreground statistic. How-

ever, this still requires specifying an alternative hypothesis, even if the alternative is nonparametric, and ensuring comparability between alternatives used for different choices of foreground statistics is nontrivial. Moreover, developing goodness-of-fit tests for composite hypotheses or hierarchical models is often difficult in practice.

In this article, we propose a new score—for both data selection and model selection—that is similar to the marginal likelihood of a semi-parametric model but does not require one to specify a background model, let alone integrate over it. The basic idea is to employ a generalized marginal likelihood where we replace the foreground model likelihood by an exponentiated divergence with nice properties, and replace the background model's marginal likelihood with a simple volume correction factor. For the choice of divergence, we use a kernelized Stein discrepancy (KSD) since it enables us to provide statistical guarantees and is easy to estimate compared to other divergences — for instance, the Kullback–Leibler divergence involves a problematic entropy term that cannot simply be dropped. The background model volume correction arises roughly as follows: if the background model is well-specified, then asymptotically, its divergence from the empirical distribution converges to zero and all that remains of the background model's contribution is the volume of its effective parameter space. Consequently, it is not necessary to specify the background model, only its effective dimension. To facilitate computation further, we develop a Laplace approximation for the foreground model's contribution to our proposed score.

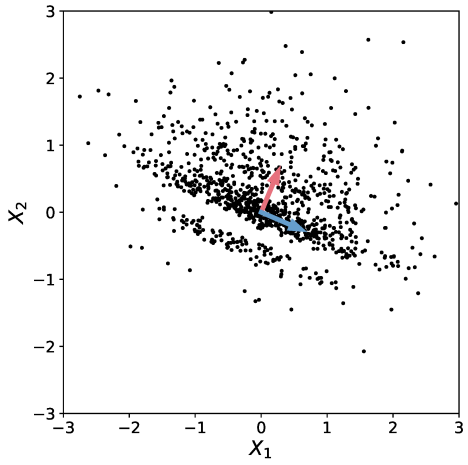
This article makes a number of novel contributions. We introduce the data selection problem in broad generality, and provide a thorough asymptotic analysis. We propose a novel model/data selection score, which we refer to as the *Stein volume criterion*, that takes the form of a generalized

marginal likelihood using a KSD. We provide new theoretical results for this generalized marginal likelihood and its associated posterior, complementing and building upon recent work on the frequentist properties of minimum KSD estimators[17]. Finally, we provide first-of-a-kind empirical data selection analyses with two models that are frequently used in single-cell RNA sequencing analysis.

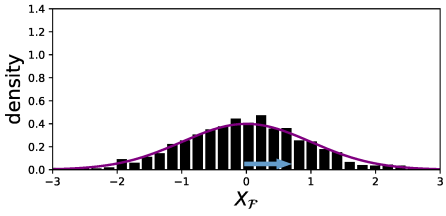
The article is organized as follows. In Section 5.2, we introduce the data selection problem and our proposed method. In Section 5.3 we study the asymptotic properties of Bayesian data selection methods and compare to model selection. Section 5.4 provides a review of related work and Section 5.5 illustrates the method on a toy example. In Section 5.6, we prove (a) consistency results for both data selection and model selection, (b) a Laplace approximation for the proposed score, and (c) a Bernstein–von Mises theorem for the corresponding generalized posterior. In Section 5.7, we apply our method to probabilistic principal components analysis (pPCA), assess its performance in simulations, and demonstrate it on single-cell RNA sequencing (scRNAseq) data. In Section 5.8, we apply our method to a spin glass model of gene expression, also demonstrated on an scRNAseq dataset. Section 5.9 concludes with a brief discussion.

5.2 Method

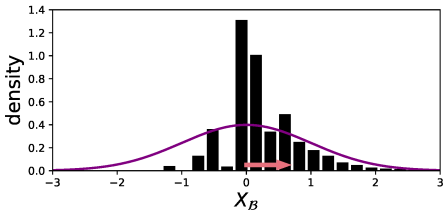
Suppose the data $X^{(1)}, \ldots, X^{(N)} \in \mathcal{X}$ are independent and identically distributed (i.i.d.), where $\mathcal{X} \subseteq \mathbb{R}^d$. Suppose the true data-generating distribution P_0 has density $p_0(x)$ with respect to Lebesgue measure, and let $\{q(x|\theta) : \theta \in \Theta\}$ be a parametric model of interest, where $\Theta \subseteq \mathbb{R}^m$.



(a) An example for which a bivariate normal model is partially misspecified. Basis vectors for $\mathcal{X}_\mathcal{F}$ (foreground) and $\mathcal{X}_\mathcal{B}$ (background) are blue and red, respectively.



(b) A univariate normal model is well-specified for the data projection onto $\mathcal{X}_\mathcal{F}$.



(c) A univariate normal model is misspecified for the data projection onto $\mathcal{X}_\mathcal{B}$.

Figure 5.1: A simple example illustrating the data selection problem.

We are interested in evaluating this model when applied to a projection of the data onto a subspace, $\mathcal{X}_\mathcal{F} \subseteq \mathcal{X}$ (the "foreground" space). Specifically, let $X_\mathcal{F} := V^\top X$ be a linear projection of $X \in \mathcal{X}$ onto $\mathcal{X}_\mathcal{F}$, where V is a matrix with orthonormal columns. Let $q(x_\mathcal{F}|\theta)$ denote the distribution of $X_\mathcal{F}$ when $X \sim q(x|\theta)$, and likewise, let $p_0(x_\mathcal{F})$ be the distribution of $X_\mathcal{F}$ when $X \sim p_0(x)$. Even when the complete model $q(x|\theta)$ is misspecified with respect to $p_0(x)$, it may be that $q(x_\mathcal{F}|\theta)$ is well-specified with respect to $p_0(x_\mathcal{F})$; see Figure 5.1 for a toy example. In such cases, the parametric model is only partially misspecified — specifically, it is misspecified on the "background" space $\mathcal{X}_\mathcal{B}$, defined as the orthogonal complement of $\mathcal{X}_\mathcal{F}$. Our goal is to find subspaces $\mathcal{X}_\mathcal{F}$ for which $q(x_\mathcal{F}|\theta)$ is correctly specified.

A natural Bayesian solution would be to replace the background component of the assumed

model, $q(x_{\mathcal{B}}|x_{\mathcal{F}}, \theta)$, with a more flexible component $\tilde{q}(x_{\mathcal{B}}|x_{\mathcal{F}}, \phi_{\mathcal{B}})$ that is guaranteed to be well-specified with respect to $p_0(x_{\mathcal{B}}|x_{\mathcal{F}})$, such as a nonparametric model. The resulting joint model, which we refer to as the "augmented model", is then

$$\begin{aligned} \theta \sim \pi(\theta), \quad X_{\mathcal{F}}^{(i)} \mid \theta \overset{\text{iid}}{\sim} q(x_{\mathcal{F}} \mid \theta), \\ \phi_{\mathcal{B}} \sim \pi_{\mathcal{B}}(\phi_{\mathcal{B}}), \quad X_{\mathcal{B}}^{(i)} \mid X_{\mathcal{F}}^{(i)}, \phi_{\mathcal{B}} \sim \tilde{q}(x_{\mathcal{B}} \mid X_{\mathcal{F}}^{(i)}, \phi_{\mathcal{B}}). \end{aligned} \tag{5.1}$$

The standard Bayesian approach would be to put a prior on the choice of foreground space $\mathcal{X}_{\mathcal{F}}$, and compute the posterior over the choice of $\mathcal{X}_{\mathcal{F}}$. Computing this posterior boils down to computing the Bayes factor $\tilde{q}(X^{(1:N)}|\mathcal{F})/\tilde{q}(X^{(1:N)}|\mathcal{F}')$ for any given pair of foregrounds $\mathcal{F}$ and $\mathcal{F}'$, where $\tilde{q}(X^{(1:N)}|\mathcal{F})$ denotes the marginal likelihood of $\mathcal{F}$ under the augmented model, that is, $\tilde{q}(X^{(1:N)}|\mathcal{F}) = \int\int q(X_{\mathcal{F}}^{(1:N)}|\theta)\, \tilde{q}(X_{\mathcal{B}}^{(1:N)}|X_{\mathcal{F}}^{(1:N)}, \phi_{\mathcal{B}})\pi(\theta)\pi_{\mathcal{B}}(\phi_{\mathcal{B}})d\theta d\phi_{\mathcal{B}}$.

However, in general, it is difficult to find a background model that (a) is guaranteed to be well-specified with respect to $p_0(x_{\mathcal{B}}|x_{\mathcal{F}})$ and (b) can be integrated over in a computationally tractable way to obtain the posterior on the choice of $\mathcal{F}$. Our proposed method, which we introduce next, sidesteps these difficulties while still exhibiting similar guarantees.

5.2.1 Proposed score for data selection and model selection

In this section, we propose a model/data selection score that is simpler to compute than the marginal likelihood of the augmented model and has similar theoretical guarantees. This score takes the form of a generalized marginal likelihood with a normalized kernelized Stein discrepancy (NKSD) estimate

taking the place of the log likelihood. Specifically, our proposed model/data selection score, termed the "Stein volume criterion" (SVC), is

$$\mathcal{K} := \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} \int \exp\Big(- \frac{N}{T} \widehat{\text{NKSD}}(p_0(x_{\mathcal{F}}) \| q(x_{\mathcal{F}}|\theta))\Big) \pi(\theta) d\theta \tag{5.2}$$

where the "temperature" $T > 0$ is a hyperparameter and $m_{\mathcal{B}}$ is the effective dimension of the background model parameter space. $\widehat{\text{NKSD}}(\cdot\|\cdot)$ is an empirical estimate of the NKSD; see Equations 5.4 and 5.5. The integral in Equation 5.2 can be approximated using techniques discussed in Section 5.2.3. The hyperparameter T can be calibrated by comparing the coverage of the standard Bayesian posterior to the coverage of the NKSD generalized posterior (Section E.1.1). The $(2\pi/N)^{m_{\mathcal{B}}/2}$ factor penalizes higher-complexity background models. In general, we allow $m_{\mathcal{B}}$ to grow with N, particularly when the background model is nonparametric. Crucially, the likelihood of the background model does not appear in our proposed score, sidestepping the need to fit or even specify the background model — indeed, the only place that the background model enters into the SVC is through $m_{\mathcal{B}}$.

Thus, rather than specify a background model and then derive $m_{\mathcal{B}}$, one can simply specify an appropriate value of $m_{\mathcal{B}}$. Reasonable choices of $m_{\mathcal{B}}$ can be derived by considering the asymptotic behavior of a Pitman-Yor process mixture model, a common nonparametric model that is a natural choice for a background model. A Pitman-Yor process mixture model with discount parameter $\alpha \in (0, 1)$, concentration parameter $\theta > -\alpha$, and D-dimensional component parameters will

asymptotically have expected effective dimension

$$m_{\mathcal{B}} \sim D \frac{\Gamma(\theta+1)}{\alpha\Gamma(\theta+\alpha)} N^{\alpha} \tag{5.3}$$

under the prior, where $a_N \sim b_N$ means that $a_N/b_N \to 1$ as $N \to \infty$ and $\Gamma(\cdot)$ is the gamma function (Pitman [199], §3.3). As a default, we recommend setting $m_{\mathcal{B}} = c_{\mathcal{B}}\, r_{\mathcal{B}} \sqrt{N}$, where $r_{\mathcal{B}}$ is the dimension of $\mathcal{X}_{\mathcal{B}}$ and $c_{\mathcal{B}}$ is a constant chosen to match Equation 5.3 with $\alpha = 1/2$. The $\sqrt{N}$ scaling is particularly nice in terms of asymptotic guarantees; see Section 5.3.2.

The SVC uses a novel, normalized version of the KSD between densities $p(x)$ and $q(x)$:

$$\text{NKSD}(p(x)\|q(x)) := \frac{\mathbb{E}_{X,Y\sim p}\left[(s_q(X) - s_p(X))^{\top}(s_q(Y) - s_p(Y))k(X,Y)\right]}{\mathbb{E}_{X,Y\sim p}[k(X,Y)]} \tag{5.4}$$

where $k(x,y) \in \mathbb{R}$ is an integrally strictly positive definite kernel, $s_q(x) := \nabla_x \log q(x)$, and $s_p(x) := \nabla_x \log p(x)$; see Section 5.6.1 for details. The numerator corresponds to the standard KSD [158]. The denominator, which is strictly positive and independent of $q(x)$, is a normalization factor that we have introduced to make the divergence comparable across spaces of different dimension. See Section E.1.2 for kernel recommendations. Extending the technique of Liu et al. [158], we propose to estimate the normalized KSD using U-statistics:

$$\widehat{\text{NKSD}}(p(x)\|q(x)) = \frac{\sum_{i\neq j} u(X^{(i)}, X^{(j)})}{\sum_{i\neq j} k(X^{(i)}, X^{(j)})} \tag{5.5}$$

where $X^{(i)} \sim p(x)$ i.i.d., the sums are over all $i, j \in \{1, \ldots, N\}$ such that $i \neq j$, and

$$u(x, y) := s_q(x)^\top s_q(y) k(x, y) + s_q(x)^\top \nabla_y k(x, y) + s_q(y)^\top \nabla_x k(x, y) + \text{trace}(\nabla_x \nabla_y^\top k(x, y)).$$

Importantly, Equation 5.5 does not require knowledge of $s_p(x)$, which is unknown in practice.

5.2.2 Comparison with the standard marginal likelihood

It is instructive to compare our proposed model/data selection score, the Stein volume criterion, to the standard marginal likelihood $\tilde{q}(X^{(1:N)}|\mathcal{F})$. In particular, we show that the SVC approximates a generalized version of the marginal likelihood. To see this, first define $H := -\int p_0(x) \log p_0(x) dx$, the entropy of the complete data distribution, and note that if were H somehow known, then the Kullback-Leibler (KL) divergence between the augmented model and the data distribution could be approximated as

$$\widehat{\text{KL}}(p_0(x) \| q(x_\mathcal{F}|\theta)\, \tilde{q}(x_\mathcal{B}|x_\mathcal{F}, \phi_\mathcal{B})) := -\frac{1}{N} \sum_{i=1}^{N} \log q(X_\mathcal{F}^{(i)}|\theta)\, \tilde{q}(X_\mathcal{B}^{(i)}|X_\mathcal{F}^{(i)}, \phi_\mathcal{B}) - H.$$

Since multiplying the marginal likelihoods by a fixed constant does not affect the Bayes factors, the following expression could be used instead of the marginal likelihood $\tilde{q}(X^{(1:N)}|\mathcal{F})$ to decide among foreground subspaces:

$$\frac{\tilde{q}(X^{(1:N)}|\mathcal{F})}{\exp(-NH)} = \int\int \exp\Big(- N\, \widehat{\text{KL}}(p_0(x) \| q(x_\mathcal{F}|\theta)\, \tilde{q}(x_\mathcal{B}|x_\mathcal{F}, \phi_\mathcal{B}))\Big) \pi(\theta) \pi_\mathcal{B}(\phi_\mathcal{B}) d\theta d\phi_\mathcal{B}. \quad (5.6)$$

Now, consider a generalized marginal likelihood where the NKSD replaces the KL:

$$\tilde{\mathcal{K}} := \int\int \exp\Big(- N \frac{1}{T} \widehat{\text{NKSD}}(p_0(x) \| q(x_{\mathcal{F}}|\theta)\, \tilde{q}(x_{\mathcal{B}}|x_{\mathcal{F}}, \phi_{\mathcal{B}}))\Big) \pi(\theta)\pi_{\mathcal{B}}(\phi_{\mathcal{B}}) d\theta d\phi_{\mathcal{B}}. \tag{5.7}$$

We refer to $\tilde{\mathcal{K}}$ as the "NKSD marginal likelihood" of the augmented model. Intuitively, we expect it to behave similarly to the standard marginal likelihood, except that it quantifies the divergence between the model and data distributions using the NKSD instead of the KL.

However, a key advantage of the NKSD marginal likelihood is that it admits a simple approximation via the SVC when the background model is well-specified, unlike the standard marginal likelihood. For instance, if the foreground and background are independent, that is, $p_0(x) = p_0(x_{\mathcal{F}})p_0(x_{\mathcal{B}})$ and $\tilde{q}(x_{\mathcal{B}}|x_{\mathcal{F}}, \phi_{\mathcal{B}}) = \tilde{q}(x_{\mathcal{B}}|\phi_{\mathcal{B}})$, then the theory in Section 5.6 can be extended to the full augmented model to show that

$$\frac{\log \tilde{\mathcal{K}}}{\log \mathcal{K}} \xrightarrow[N\to\infty]{P_0} 1, \tag{5.8}$$

where $\mathcal{K}$ is the SVC (Equation 5.2). Thus, the SVC approximates the NKSD marginal likelihood of the augmented model, suggesting that the SVC may be a convenient alternative to the standard marginal likelihood. Formally, Section 5.3 shows that the SVC exhibits consistency properties similar to the standard marginal likelihood, even when $p_0(x) \neq p_0(x_{\mathcal{F}})p_0(x_{\mathcal{B}})$.

5.2.3 Computation

Next, we discuss methods for computing the SVC including exact solutions, Laplace/BIC approximation, variational approximation, and comparing many possible choices of $\mathcal{F}$.

Exact solution for exponential families

When the foreground model is an exponential family, the SVC can be computed analytically. Specifically, in Section E.1.3, we show if $q(x_{\mathcal{F}}|\theta) = \lambda(x_{\mathcal{F}}) \exp(\theta^\top t(x_{\mathcal{F}}) - \kappa(\theta))$, then

$$\widehat{\text{NKSD}}(p_0(x_{\mathcal{F}}) \| q(x_{\mathcal{F}}|\theta)) = \theta^\top A\, \theta + B^\top \theta + C \tag{5.9}$$

where A, B, and C depend on the data $X^{(1:N)}$ but not on θ. Therefore, we can place a multivariate Gaussian prior on θ and compute the SVC in closed form; see Section E.1.3.

Laplace and BIC approximations

The Laplace approximation is a widely-used technique for computing marginal likelihoods. In Theorem 5.6.9, we establish regularity conditions under which a Laplace approximation to the SVC is justified by being asymptotically correct. The resulting approximation is

$$\mathcal{K} \approx \frac{\exp\left(-\frac{N}{T}\widehat{\text{NKSD}}(p_0(x_{\mathcal{F}}) \| q(x_{\mathcal{F}}|\theta_N))\right) \pi(\theta_N)}{|\det \frac{1}{T}\nabla^2_\theta \widehat{\text{NKSD}}(p_0(x_{\mathcal{F}}) \| q(x_{\mathcal{F}}|\theta_N))|^{1/2}} \left(\frac{2\pi}{N}\right)^{(m_{\mathcal{F}}+m_{\mathcal{B}})/2} \tag{5.10}$$

where $\theta_N := \arg\min_\theta \widehat{\text{NKSD}}(p_0(x_\mathcal{F}) \| q(x_\mathcal{F}|\theta))$ is the point at which the estimated NKSD is minimized, the "minimum Stein discrepancy estimator" as defined by Barp et al. [17].

We can also make a rougher approximation, analogous to the Bayesian information criterion (BIC), which does not require one to compute second derivatives of $\widehat{\text{NKSD}}$:

$$\mathcal{K} \approx \exp\Big(- \frac{N}{T} \widehat{\text{NKSD}}(p_0(x_\mathcal{F}) \| q(x_\mathcal{F}|\theta_N)) \Big) \left(\frac{2\pi}{N} \right)^{(m_\mathcal{F} + m_\mathcal{B})/2}. \tag{5.11}$$

This approximation is easy to compute, given a minimum Stein discrepancy estimator θ_N. Like the SVC, it satisfies all of our consistency desiderata (Section E.2). However, we expect it to perform worse than the SVC when there is not yet enough data for the NKSD posterior to be highly concentrated, that is, when a range of θ values can plausibly explain the data.

Comparing many foregrounds using approximate optima

Often, we would like to evaluate many possible subspaces $\mathcal{X}_\mathcal{F}$ when performing data selection. Even when using the Laplace or BIC approximation to the SVC, this can get computationally prohibitive since we need to re-optimize to find θ_N for every $\mathcal{F}$ under consideration. Here, we propose a way to reduce this cost by making a fast linear approximation. Define $\ell_j(\theta) := \widehat{\text{NKSD}}(p_0(x_{\mathcal{F}_j}) \| q(x_{\mathcal{F}_j}|\theta))$ for $j \in \{1, 2\}$. For $w \in [0, 1]$, we can linearly interpolate

$$\theta_N(w) := \arg\min_\theta \ell_1(\theta) + w(\ell_2(\theta) - \ell_1(\theta)). \tag{5.12}$$

Now, $\theta_N(0)$ and $\theta_N(1)$ are the minimum Stein discrepancy estimators for $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. Given $\theta_N(0)$, we can approximate $\theta_N(1)$ by applying the implicit function theorem and a first-order Taylor expansion (Section E.1.4):

$$\theta_N(1) \approx \theta_N(0) - \nabla^2_\theta \ell_1(\theta_N(0))^{-1} \nabla_\theta \ell_2(\theta_N(0)). \tag{5.13}$$

Note that the derivatives of ℓ_j are often easy to compute with automatic differentiation [19]. Note also that when we are comparing one foreground subspace, such as $\mathcal{X}_{\mathcal{F}_1} = \mathcal{X}$, to many other foreground subspaces $\mathcal{X}_{\mathcal{F}_2}$, the inverse Hessian $\nabla^2_\theta \ell_1(\theta_N(0))^{-1}$ only needs to be computed once. Thus, Equation 5.13 provides a fast method for computing Laplace or BIC approximations to the SVC for a large number of candidate foregrounds $\mathcal{F}$.

Variational approximation

Variational inference is a method for approximating both the posterior distribution and the marginal likelihood of a probabilistic model. Since the SVC takes the form of a generalized marginal likelihood, we can derive a variational approximation to the SVC. Let $r_\zeta(\theta)$ be an approximating distri-

bution parameterized by ζ. By Jensen's inequality, we have

$$\begin{aligned}
&\log \int \exp\Big(-\frac{N}{T}\widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\Big)\pi(\theta)d\theta \\
&\quad = \log \int \frac{\exp\left(-\frac{N}{T}\widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\right)\pi(\theta)}{r_\zeta(\theta)} r_\zeta(\theta)d\theta \\
&\quad \geq \mathbb{E}_{r_\zeta}\left[\log\left(\frac{\exp\left(-\frac{N}{T}\widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\right)\pi(\theta)}{r_\zeta(\theta)}\right)\right] \\
&\quad = -\frac{N}{T}\mathbb{E}_{r_\zeta}\left[\widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\right] + \mathbb{E}_{r_\zeta}[\log\pi(\theta)] - \mathbb{E}_{r_\zeta}[\log r_\zeta(\theta)].
\end{aligned} \tag{5.14}$$

Maximizing this lower bound with respect to the variational parameters ζ provides an approximation to the SVC, or more precisely, to $\log \mathcal{K} - (m_{\mathcal{B}}/2)\log(2\pi/N)$. Note that this variational approximation falls within the framework of generalized variational inference proposed by Knoblauch et al. [141].

5.3 Data selection and model selection consistency

This section presents our consistency results when comparing two different foreground subspaces (data selection) or two different foreground models (model selection). The theory supporting these results is in Sections 5.6 and E.2. We consider four distinct properties that a procedure would ideally exhibit: data selection consistency, nested data selection consistency, model selection consistency, and nested model selection consistency; see Section 5.6.4 for precise definitions. We consider six possible model/data selection scores, and we establish which scores satisfy which properties; see Table 5.1. The SVC and the full marginal likelihood are the only two of the six scores that satisfy all

Table 5.1: Consistency properties satisfied by various model/data selection scores. Only the Stein volume criterion $\mathcal{K}$ and the full marginal likelihood $\tilde{q}(X^{(1:N)}|\mathcal{F})$ satisfy all four desiderata. (d.s. = data selection, m.s. = model selection, marg = marginal, lik = likelihood.)

	Consistency property			
Score	d.s.	nested d.s.	m.s.	nested m.s.
$\tilde{q}(X^{(1:N)}\|\mathcal{F})$ full marginal likelihood	✓	✓	✓	✓
$\mathcal{K}^{(a)}$ foreground marg lik, background volume	✗	✗	✓	✓
$\mathcal{K}^{(b)}$ foreground marg NKSD	✓	✗	✓	✓
$\mathcal{K}^{(c)}$ foreground marg KL, background volume	✓	✗	✓	✓
$\mathcal{K}^{(d)}$ foreground NKSD, background volume	✓	✓	✓	✗
$\mathcal{K}$ foreground marg NKSD, background volume	✓	✓	✓	✓

four consistency properties.

The intuition behind Bayesian model selection is often explained in terms of Occam's razor: a theory should be as simple as possible but no simpler. Data selection and nested data selection encapsulate a complementary intuition: a theory should explain as much of the data as possible but no more. In other words, when choosing between foreground spaces, a consistent data selection score will asymptotically prefer the highest-dimensional space on which the model is correctly specified.

As in standard model selection, a practical concern in data selection is robustness. For instance, if the foreground model is even slightly misspecified on $\mathcal{X}_{\mathcal{F}_2}$, then the empty foreground $\mathcal{X}_{\mathcal{F}_1} = \varnothing$ will be asymptotically preferred over $\mathcal{X}_{\mathcal{F}_2}$. Since the SVC takes the form of a generalized marginal likelihood, techniques for improving robustness with the standard marginal likelihood—such as coarsened posteriors, power posteriors, and BayesBag—could potentially be extended to address this issue [177,118]. We leave exploration of such approaches to future work.

5.3.1 Data selection consistency

First, consider comparisons between different choices of foreground, $\mathcal{F}_1$ and $\mathcal{F}_2$. When the model is correctly specified over $\mathcal{F}_1$ but not $\mathcal{F}_2$, we refer to asymptotic concentration on $\mathcal{F}_1$ as "data selection consistency" (and vice versa if $\mathcal{F}_2$ is correct but not $\mathcal{F}_1$). For the standard marginal likelihood of the augmented model, we have (see Section E.2.2)

$$\frac{1}{N}\log\frac{\tilde{q}(X^{(1:N)}|\mathcal{F}_1)}{\tilde{q}(X^{(1:N)}|\mathcal{F}_2)} \xrightarrow[N\to\infty]{P_0} \mathrm{KL}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta^{\mathrm{KL}}_{2,*})) - \mathrm{KL}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta^{\mathrm{KL}}_{1,*})) \quad (5.15)$$

where $\theta^{\mathrm{KL}}_{j,*} := \arg\min \mathrm{KL}(p_0(x_{\mathcal{F}_j})\|q(x_{\mathcal{F}_j}|\theta))$ for $j \in \{1,2\}$, that is, $\theta^{\mathrm{KL}}_{j,*}$ is the parameter value that minimizes the KL divergence between the projected data distribution $p_0(x_{\mathcal{F}_j})$ and the projected model $q(x_{\mathcal{F}_j}|\theta)$. Thus, $\tilde{q}(X^{(1:N)}|\mathcal{F}_j)$ asymptotically concentrates on the $\mathcal{F}_j$ on which the projected model can most closely match the data distribution in terms of KL.

In Theorem 5.6.17, we show that under mild regularity conditions, the Stein volume criterion behaves precisely the same way but with the NKSD in place of the KL:

$$\frac{1}{N}\log\frac{\mathcal{K}_1}{\mathcal{K}_2} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\mathrm{NKSD}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta^{\mathrm{NKSD}}_{2,*})) - \frac{1}{T}\mathrm{NKSD}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta^{\mathrm{NKSD}}_{1,*})) \quad (5.16)$$

where $\theta^{\mathrm{NKSD}}_{j,*} := \arg\min \mathrm{NKSD}(p_0(x_{\mathcal{F}_j})\|q(x_{\mathcal{F}_j}|\theta))$ for $j \in \{1,2\}$. Therefore, $\tilde{q}(X^{(1:N)}|\mathcal{F})$ and $\mathcal{K}$ both yield data selection consistency. It is important here that the SVC uses a true divergence, rather

than a divergence up to a data-dependent constant. If we instead used

$$\mathcal{K}^{(\mathrm{a})} := \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} q(X_{\mathcal{F}}^{(1:N)}), \tag{5.17}$$

which employs the foreground marginal likelihood $q(X_{\mathcal{F}}^{(1:N)}) = \int q(X_{\mathcal{F}}^{(1:N)}|\theta)\pi(\theta)d\theta$ and a background volume correction, we would get qualitatively different behavior (Section E.2.2):

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{(\mathrm{a})}}{\mathcal{K}_2^{(\mathrm{a})}} \xrightarrow[N\to\infty]{P_0} \mathrm{KL}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta_{2,*}^{\mathrm{KL}})) - \mathrm{KL}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta_{1,*}^{\mathrm{KL}})) + H_{\mathcal{F}_2} - H_{\mathcal{F}_1} \tag{5.18}$$

where $H_{\mathcal{F}_j} := -\int p_0(x_{\mathcal{F}_j})\log p_0(x_{\mathcal{F}_j})dx_{\mathcal{F}_j}$ is the entropy of $p_0(x_{\mathcal{F}_j})$ for $j \in \{1,2\}$. In short, the naive score $\mathcal{K}^{(\mathrm{a})}$ is a bad choice: it decides between data subspaces based not just on how well the parametric foreground model performs, but also on the entropy of the data distribution in each space. As a result, $\mathcal{K}^{(\mathrm{a})}$ does not exhibit data selection consistency.

5.3.2 Nested data selection consistency

When $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$, we refer to the problem of deciding between subspaces $\mathcal{F}_1$ and $\mathcal{F}_2$ as nested data selection, in counterpoint to nested model selection, where one model is a subset of another[279]. If the model $q(x|\theta)$ is well-specified over $\mathcal{X}_{\mathcal{F}_1}$, then it is guaranteed to be well-specified over any lower-dimensional sub-subspace $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$; in this case, we refer to asymptotic concentration on $\mathcal{F}_1$ as "nested data selection consistency". In this situation, $\mathrm{KL}(p_0(x_{\mathcal{F}_j})\|q(x_{\mathcal{F}_j}|\theta_{j,*}^{\mathrm{KL}}))$ and $\mathrm{NKSD}(p_0(x_{\mathcal{F}_j}), q(x_{\mathcal{F}_j}|\theta_{j,*}^{\mathrm{NKSD}}))$ are both zero for $j \in \{1,2\}$, making it necessary to look at higher-

order terms in Equations 5.15 and 5.16. In Section E.2.3, we show that if $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$, $q(x|\theta)$ is well-specified over $\mathcal{X}_{\mathcal{F}_1}$, the background models are well-specified, and their dimensions $m_{\mathcal{B}_1}$ and $m_{\mathcal{B}_2}$ are constant with respect to N, then

$$\frac{1}{\log N} \log \frac{\tilde{q}(X^{(1:N)}|\mathcal{F}_1)}{\tilde{q}(X^{(1:N)}|\mathcal{F}_2)} \xrightarrow[N\to\infty]{P_0} \frac{1}{2}(m_{\mathcal{F}_2} + m_{\mathcal{B}_2} - m_{\mathcal{F}_1} - m_{\mathcal{B}_1}) \tag{5.19}$$

where $m_{\mathcal{F}_j}$ is the effective dimension of the parameter space of $q(x_{\mathcal{F}_j}|\theta)$. In Theorem 5.6.17, we show that under mild regularity conditions, the SVC behaves the same way:

$$\frac{1}{\log N} \log \frac{\mathcal{K}_1}{\mathcal{K}_2} \xrightarrow[N\to\infty]{P_0} \frac{1}{2}(m_{\mathcal{F}_2} + m_{\mathcal{B}_2} - m_{\mathcal{F}_1} - m_{\mathcal{B}_1}). \tag{5.20}$$

Thus, so long as $m_{\mathcal{F}_2} + m_{\mathcal{B}_2} > m_{\mathcal{F}_1} + m_{\mathcal{B}_1}$ whenever $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$, the marginal likelihood and the SVC asymptotically concentrate on the larger foreground $\mathcal{F}_1$; hence, they both exhibit nested data selection consistency. This is a natural assumption since the background model is generally more flexible—on a per dimension basis—than the foreground model.

The volume correction $(2\pi/N)^{m_{\mathcal{B}}/2}$ in the definition of the SVC is important for nested data selection consistency (Equation 5.20). An alternative score without that correction,

$$\mathcal{K}^{(\mathrm{b})} := \int \exp\Big(- \frac{N}{T} \widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}}) \| q(x_{\mathcal{F}}|\theta)) \Big) \pi(\theta) d\theta, \tag{5.21}$$

exhibits data selection consistency (Equation 5.16 holds for $\mathcal{K}^{(\mathrm{b})}$), but not nested data selection consistency; see Sections E.2.2 and E.2.3. More subtly, the asymptotics of the SVC in the case of

nested data selection also depend on the variance of U-statistics. To illustrate, consider a score that is similar to the SVC but uses $\widehat{\text{KL}}$ instead of $\widehat{\text{NKSD}}$:

$$\mathcal{K}^{(\text{c})} := \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} \int \exp\Big(- N\widehat{\text{KL}}(p_0(x_{\mathcal{F}}) \| q(x_{\mathcal{F}}|\theta))\Big)\pi(\theta)d\theta \tag{5.22}$$

where $\widehat{\text{KL}}(p_0(x_{\mathcal{F}}) \| q(x_{\mathcal{F}}|\theta)) := -\frac{1}{N}\sum_{i=1}^{N} \log q(X_{\mathcal{F}}^{(i)}|\theta) - H_{\mathcal{F}}$ and $H_{\mathcal{F}}$ is required to be known. The score $\mathcal{K}^{(\text{c})}$ exhibits data selection consistency, but not nested data selection consistency. The reason is that the error in estimating the KL is of order $1/\sqrt{N}$ by the central limit theorem, and this source of error dominates the $\log N$ term contributed by the volume correction; see Section E.2.3. Meanwhile, the error in estimating the NKSD is of order $1/N$ when the model is well-specified, due to the rapid convergence rate of the U-statistic estimator. Thus, in the SVC, this source of error is dominated by the volume correction; see Theorem 5.6.12.

The nested data selection results we have described so far assume $m_{\mathcal{B}}$ does not depend on N, or at least $m_{\mathcal{B}_2} - m_{\mathcal{B}_1}$ does not depend on N (Theorem 5.6.17). However, in Section 5.2.1, we suggest setting $m_{\mathcal{B}} = c_{\mathcal{B}}\, r_{\mathcal{B}}\sqrt{N}$ where $c_{\mathcal{B}}$ is a constant and $r_{\mathcal{B}}$ is the dimension of $\mathcal{X}_{\mathcal{B}}$. With this choice, the asymptotics of the SVC for nested data selection become (Theorem 5.6.17)

$$\frac{1}{\sqrt{N}\log N} \log \frac{\mathcal{K}_1}{\mathcal{K}_2} \xrightarrow[N\to\infty]{P_0} \frac{1}{2} c_{\mathcal{B}}\,(r_{\mathcal{B}_2} - r_{\mathcal{B}_1}). \tag{5.23}$$

Since $r_{\mathcal{B}_1} < r_{\mathcal{B}_2}$ when $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$, the SVC concentrates on the larger foreground $\mathcal{F}_1$, yielding nested data selection consistency. Going beyond the well-specified case, Theorem 5.6.17 shows that

Equation 5.23 holds when $\text{NKSD}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta^{\text{NKSD}}_{1,*})) = \text{NKSD}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta^{\text{NKSD}}_{2,*})) \neq 0$, that is, when the models are misspecified by the same amount as measured by the NKSD. Equation 5.23 holds regardless of whether $m_{\mathcal{F}_1}$ is equal to $m_{\mathcal{F}_2}$.

5.3.3 Model selection and nested model selection consistency

Consider comparing different foreground models $q_1(x_{\mathcal{F}}|\theta_1)$ and $q_2(x_{\mathcal{F}}|\theta_2)$ over the same subspace $\mathcal{X}_{\mathcal{F}}$, while using the same background model. We say that a score exhibits "model selection consistency" if it concentrates on the correct model, when one of the models is correctly specified and the other is not. When the two models are nested and both are correct, a score exhibits "nested model selection consistency" if it concentrates on the simpler model.

Like the standard marginal likelihood, the SVC exhibits both types of model selection consistency. The standard marginal likelihood satisfies (Section E.2.4)

$$\frac{1}{N}\log\frac{\tilde{q}_1(X^{(1:N)}|\mathcal{F})}{\tilde{q}_2(X^{(1:N)}|\mathcal{F})} \xrightarrow[N\to\infty]{P_0} \text{KL}(p_0(x_{\mathcal{F}})\|q_2(x_{\mathcal{F}}|\theta^{\text{KL}}_{2,*})) - \text{KL}(p_0(x_{\mathcal{F}})\|q_1(x_{\mathcal{F}}|\theta^{\text{KL}}_{1,*})) \tag{5.24}$$

where $\theta^{\text{KL}}_{j,*} := \arg\min \text{KL}(p_0(x_{\mathcal{F}})\|q_j(x_{\mathcal{F}}|\theta_j))$ for $j \in \{1,2\}$. Analogously, by Theorem 5.6.17,

$$\frac{1}{N}\log\frac{\mathcal{K}_1}{\mathcal{K}_2} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\text{NKSD}(p_0(x_{\mathcal{F}})\|q_2(x_{\mathcal{F}}|\theta^{\text{NKSD}}_{2,*})) - \frac{1}{T}\text{NKSD}(p_0(x_{\mathcal{F}})\|q_1(x_{\mathcal{F}}|\theta^{\text{NKSD}}_{1,*})) \tag{5.25}$$

where $\theta^{\text{NKSD}}_{j,*} := \arg\min \text{NKSD}(p_0(x_{\mathcal{F}})\|q_j(x_{\mathcal{F}}|\theta_j))$ for $j \in \{1,2\}$. Thus, for both scores, concentration occurs on the model that comes closer to the data distribution in terms of the corresponding

divergence (KL or NKSD).

For nested model selection, suppose both foreground models are well-specified and $m_{\mathcal{B}_1} = m_{\mathcal{B}_2}$. Letting $m_{\mathcal{F},j}$ be the parameter dimension of $q_j(x_{\mathcal{F}}|\theta_j)$, we have (Section E.2.5)

$$\frac{1}{\log N} \log \frac{\tilde{q}_1(X^{(1:N)}|\mathcal{F})}{\tilde{q}_2(X^{(1:N)}|\mathcal{F})} \xrightarrow[N\to\infty]{P_0} \frac{1}{2}(m_{\mathcal{F},2} - m_{\mathcal{F},1}). \tag{5.26}$$

In Theorem 5.6.17, we show that the SVC behaves identically:

$$\frac{1}{\log N} \log \frac{\mathcal{K}_1}{\mathcal{K}_2} \xrightarrow[N\to\infty]{P_0} \frac{1}{2}(m_{\mathcal{F},2} - m_{\mathcal{F},1}). \tag{5.27}$$

Here, a key role is played by the volume of the foreground parameter space, which quantifies the foreground model complexity. The SVC accounts for this by integrating over foreground parameter space. Meanwhile, a naive alternative that ignores the foreground volume,

$$\mathcal{K}^{(\mathrm{d})} := \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} \exp\Big(- \frac{N}{T} \min_{\theta} \widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}}) \| q(x_{\mathcal{F}}|\theta))\Big), \tag{5.28}$$

exhibits model selection consistency (Equation 5.25 holds for $\mathcal{K}^{(\mathrm{d})}$) but not nested model selection consistency (Section E.2.5). The Laplace and BIC approximations to the SVC (Equations 5.10 and 5.11) explicitly correct for the foreground parameter volume without integrating.

5.4 Related work

Projection pursuit methods are closely related to data selection in that they attempt to identify "interesting" subspaces of the data. However, projection pursuit uses certain pre-specified objective functions to optimize over projections, whereas our method allows one to specify a model of interest[112].

Another related line of research is on Bayesian goodness-of-fit (GOF) tests, which compute the posterior probability that the data comes from a given parametric model versus a flexible alternative such as a nonparametric model. Our setup differs in that it aims to compare among different semi-parametric models. Nonetheless, in an effort to address the GOF problem, a number of authors have developed nonparametric models with tractable marginals[273,21], and using these models as the background component in an augmented model could in theory solve data selection problems. In practice, however, such models can only be applied to one-dimensional or few-dimensional data spaces. In Section 5.7, we show that naively extending the method of Berger & Guglielmi[21] to the multi-dimensional setting has fundamental limitations.

There is a sizeable frequentist literature on GOF testing using discrepancies[96,18,98]. Our proposed method builds directly on the KSD-based GOF test proposed by Liu et al.[158] and Chwialkowski et al.[41]. However, using these methods to draw comparisons between different foreground subspaces is non-trivial, since the set of alternative models considered by the GOF test, though non-parametric, will be different over data spaces with different dimensionality. Moreover, the Bayesian aspect of the SVC makes it more straightforward to integrate prior information and employ hierar-

chical models.

In composite likelihood methods, instead of the standard likelihood, one uses the product of the conditional likelihoods of selected statistics [156,272]. Composite likelihoods have seen widespread use, often for robustness or computational purposes. However, in composite likelihood methods, the choice of statistics is fixed before performing inference. In contrast, in data selection the choice of statistics is a central quantity to be inferred.

Relatedly, our work connects with the literature on robust Bayesian methods. Doksum & Lo [58] propose conditioning on the value of an insufficient statistic, rather than the complete dataset, when performing inference; also see Lewis et al. [154]. However, making an appropriate choice of statistic requires one to know which aspects of the model are correct; in contrast, our procedure infers the choice of statistic. The NKSD posterior also falls within the general class of Gibbs posteriors, which have been studied in the context of robustness, randomized estimators, and generalized belief updating [299,298,129,24,127,177].

Our theoretical results also contribute to the emerging literature on Stein discrepancies [13]. Barp et al. [17] recently proposed minimum kernelized Stein discrepancy estimators and established their consistency and asymptotic normality. In Section 5.6, we establish a Bayesian counterpart to these results, showing that the NKSD posterior is asymptotically normal (in the sense of Bernstein–von Mises) and admits a Laplace approximation. To prove this result, we rely on the recent work of Miller [176] on the asymptotics of generalized posteriors. Since Barp et al. [17] show that the kernelized Stein discrepancy is related to the Hyvärinen divergence in that both are Stein discrepancies, our work bears an interesting relationship to that of Shao et al. [233], who use a Bayesian version of the

Hyvärinen divergence to perform model selection with improper priors. They derive a consistency result analogous to Equation 5.16, however, their model selection score takes the form of a prequential score, not a Gibbs marginal likelihood as in the SVC, and cannot be used for data selection.

In independent recent work, Matsubara et al. [169] propose a Gibbs posterior based on the KSD and derive a Bernstein-von Mises theorem similar to Theorem 5.6.9 using the results of Miller [176]. Their method is not motivated by the Bayesian data selection problem but rather by (1) inference for energy-based models with intractable normalizing constants and (2) robustness to ϵ-contamination. Their Bernstein-von Mises theorem differs from ours in that it applies to a V-statistic estimator of the KSD rather than a U-statistic estimator of the NKSD.

Our linear approximation to the minimum Stein discrepancy estimator (Section 5.2.3) is directly inspired by the Swiss Army infinitesimal jackknife of Giordano et al. [91], which similarly computes the linear response of an extremum estimator with respect to perturbations of the dataset.

5.5 Toy example

The purpose of this toy example is to illustrate the behavior of the Stein volume criterion, and compare it to some of the defective alternatives listed in Table 5.1, in a simple setting where all computations can be done analytically (Section E.1.3). In all of the following experiments, we simulated data from a bivariate normal distribution: $X^{(1)}, \ldots, X^{(N)}$ i.i.d. $\sim \mathcal{N}((0,0)^\top, \Sigma_0)$.

To set up the Stein volume criterion, we set $T = 5$ and we choose a radial basis function kernel, $k(x,y) = \exp(-\frac{1}{2}\|x-y\|_2^2)$, which factors across dimensions. We considered both dataset

size-independent values of $m_{\mathcal{B}}$ (in particular, $m_{\mathcal{B}} = 5\, r_{\mathcal{B}}$) and dataset size-dependent values of $m_{\mathcal{B}}$ (in particular, Equation 5.3 with $\alpha = 0.5, \theta = 1$, and $D = 0.2$, where fractional values of D correspond to shared parameters across components in the Pitman-Yor mixture model), obtaining very similar results in each case (shown in Figures 5.2 and E.1, respectively). These choices of $m_{\mathcal{B}}$ ensure that, except for at very small N, the background model has more parameters per data dimension than each of the foreground models considered below, which have just one. In particular, $m_{\mathcal{B}} > 1\, r_{\mathcal{B}}$ for all N (in the size-independent case) and for $N \geq 5$ (in the size-dependent case).

Data selection consistency

First, we set Σ_0 to be a diagonal matrix with entries $(1, 1/2)$, that is, $\Sigma_0 = \text{diag}(1, 1/2)$, and for $x \in \mathbb{R}^2$, we consider the model

$$\begin{aligned} q(x|\theta) &= \mathcal{N}(x \mid \theta, I) \\ \pi(\theta) &= \mathcal{N}(\theta \mid (0,0)^\top, 10I) \end{aligned} \tag{5.29}$$

where I denotes the identity matrix. This parametric model is misspecified, owing to the incorrect choice of covariance matrix. We consider two choices of foreground subspace: the first dimension (defined by the projection matrix $V_{\mathcal{F}_1} = (1,0)^\top$) or the second dimension (projection matrix $V_{\mathcal{F}_2} = (0,1)^\top$). The model is only well-specified for $\mathcal{F}_1$ (not $\mathcal{F}_2$), so a successful data selection procedure would asymptotically select $\mathcal{F}_1$.

In Figure 5.2a, we see that the SVC correctly concentrates on $\mathcal{F}_1$ as the number of datapoints N increases, with the log SVC ratio growing linearly in N, as predicted by Equation 5.16. Meanwhile,

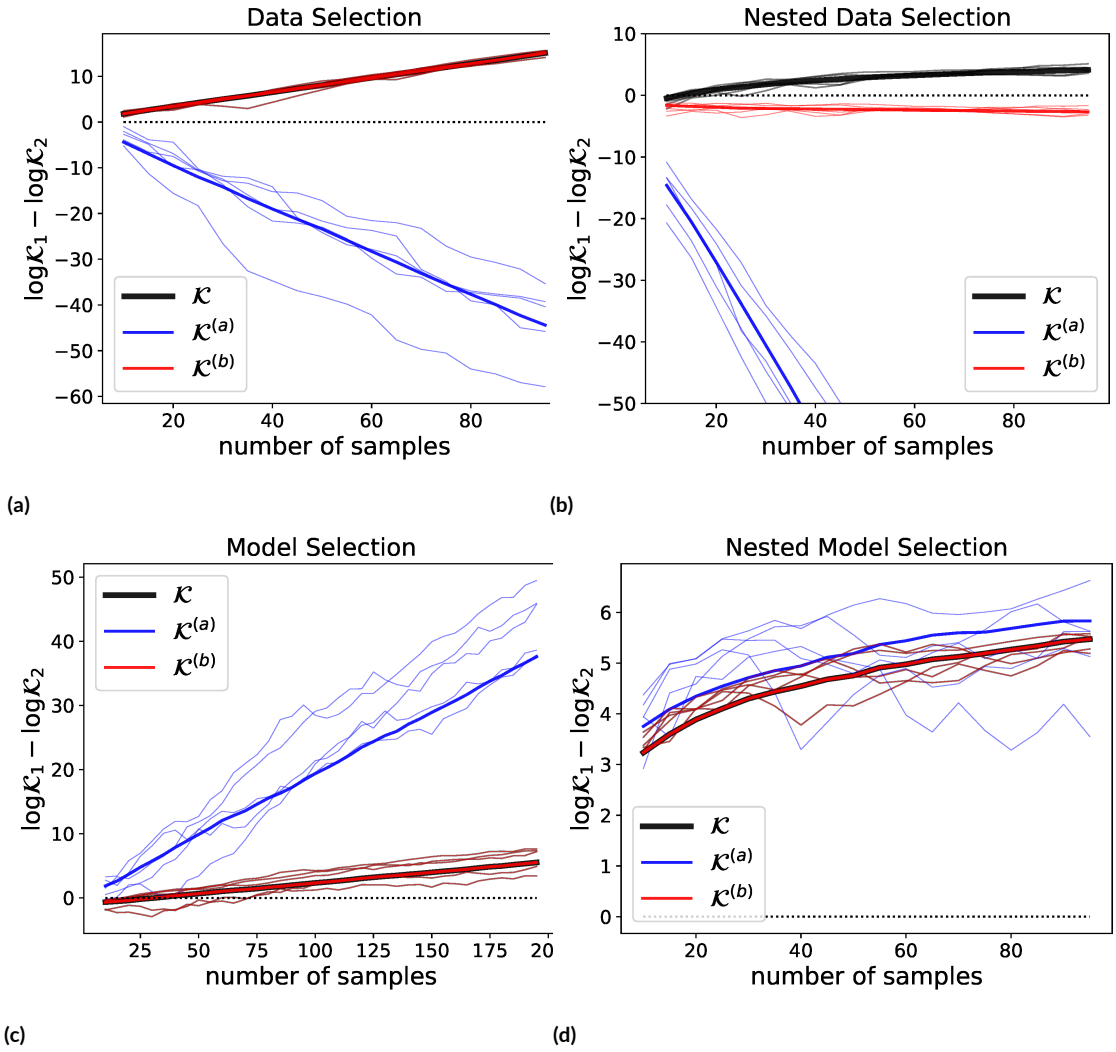


Figure 5.2: Behavior of the Stein volume criterion $\mathcal{K}$, the foreground marginal likelihood with a background volume correction $\mathcal{K}^{(\mathrm{a})}$, and the foreground marginal nksd $\mathcal{K}^{(\mathrm{b})}$ on toy examples. Here, we set $m_{\mathcal{B}} = 5\, r_{\mathcal{B}}$. The plots show the results for 5 randomly generated datasets (thin lines) and the average over 100 random datasets (bold lines).

the naive alternative score $\mathcal{K}^{(\mathrm{a})}$ (Equation 5.17) fails since it depends on the foreground entropies, while $\mathcal{K}^{(\mathrm{b})}$ (Equation 5.21) succeeds since the volume correction is negligible in this case; see Section 5.3.1 and Table 5.1.

Nested data selection consistency

Next, we examine the nested data selection case. We use the same model (Equation 5.29), but we set $\Sigma_0 = I$ so that the model is well-specified even without being projected. We compare the complete data space ($\mathcal{X}_{\mathcal{F}_1} = \mathcal{X}$, projection matrix $V_{\mathcal{F}_1} = I$) to the first dimension alone (projection matrix $V_{\mathcal{F}_1} = (1, 0)^\top$). Nested data selection consistency demands that the higher-dimensional data space $\mathcal{X}_{\mathcal{F}_1}$ be preferred asymptotically, since the model is well-specified for both $\mathcal{X}_{\mathcal{F}_1}$ and $\mathcal{X}_{\mathcal{F}_2}$. Figure 5.2b shows that this is indeed the case for the Stein volume criterion, with the log SVC ratio growing at a $\log N$ rate when $m_{\mathcal{B}}$ is independent of N, as predicted by Equation 5.20. When $m_{\mathcal{B}}$ depends on N via the Pitman-Yor expression, the log SVC ratio grows at a $N^\alpha \log N$ rate (Figure E.1b). Meanwhile, Figure 5.2b shows that $\mathcal{K}^{(\mathrm{a})}$ and $\mathcal{K}^{(\mathrm{b})}$ both fail to exhibit nested data selection consistency, in accordance with our theory (Section 5.3.2 and Table 5.1).

Model selection consistency (nested and non-nested)

Finally, we examine model selection and nested model selection consistency. We again set $\Sigma_0 = I$. We first compare the (well-specified) model $q(x|\theta) = \mathcal{N}(x \mid \theta, I)$ to the (misspecified) model $q(x|\theta) = \mathcal{N}(x \mid \theta, 2I)$, using the prior $\pi(\theta) = \mathcal{N}(\theta \mid (0, 0)^\top, 10I)$ for both models. As shown in Figure 5.2c, the SVC correctly concentrates on the first model, with the log SVC ratio growing linearly in N, as predicted by Equation 5.25. The same asymptotic behavior is exhibited by $\mathcal{K}^{(\mathrm{a})}$, which is equivalent to the standard Bayesian marginal likelihood in this setting (Section 5.3.3). Finally, to check nested model selection consistency, we compare two well-specified nested models:

$q(x) = \mathcal{N}(x \mid (0,0)^\top, I)$ and $q(x|\theta) = \mathcal{N}(x \mid \theta, I)$. Figure 5.2d shows that the SVC correctly selects the simpler model (that is, the model with smaller parameter dimension) and the log SVC ratio grows as $\log N$ (Equation 5.27). This, too, matches the behavior of the standard Bayesian marginal likelihood, seen in the plot of $\mathcal{K}^{(a)}$.

5.6 Theory

5.6.1 Properties of the NKSD

Suppose $X^{(1)}, \ldots, X^{(N)}$ are i.i.d. samples from a probability measure P on $\mathcal{X} \subseteq \mathbb{R}^d$ having density $p(x)$ with respect to the Lebesgue measure. Let $L^1(P)$ denote the set of measurable functions f such that $\int \|f(x)\| p(x) dx < \infty$ where $\|\cdot\|$ is the Euclidean norm. We impose the following regularity conditions to use the NKSD to compare P with another probability measure Q having density $q(x)$ with respect to the Lebesgue measure; these are similar to conditions used for the standard KSD in previous work [158,17].

Condition 5.6.1 (Restrictions on p and q)**.** *Assume* $s_p(x) := \nabla_x \log p(x)$ *and* $s_q(x) := \nabla_x \log q(x)$ *exist and are continuous for all* $x \in \mathcal{X}$*, and assume* $\mathcal{X}$ *is connected and open. Further, assume* $s_p, s_q \in L^1(P)$.

We refer to s_p as the Stein score function of p. Note that existence of $s_p(x)$ implies $p(x) > 0$. Now, consider a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The kernel k is said to be *integrally strictly positive definite* if for any $g : \mathcal{X} \to \mathbb{R}$ such that $0 < \int_{\mathcal{X}} |g(x)| dx < \infty$, we have $\int_{\mathcal{X}} \int_{\mathcal{X}} g(x) k(x,y) g(y) dx dy > 0$. The kernel k is said to *belong to the Stein class of* P if $\int_{\mathcal{X}} \nabla_x (k(x,y)p(x)) dx = 0$ for all $y \in \mathcal{X}$.

Condition 5.6.2 (Restrictions on k)**.** *Assume the kernel k is symmetric, bounded, integrally strictly positive definite, and belongs to the Stein class of P.*

The following result shows that the NKSD can be written in a way that does not involve s_p; this is particularly useful for estimating the NKSD when P is unknown.

Proposition 5.6.3. *If Conditions 5.6.1 and 5.6.2 hold, then the NKSD is finite and*

$$\textit{NKSD}(p(x)\|q(x)) := \frac{\mathbb{E}_{X,Y\sim p}[u(X,Y)]}{\mathbb{E}_{X,Y\sim p}[k(X,Y)]} \tag{5.30}$$

where

$$u(x,y) = s_q(x)^\top s_q(y)k(x,y) + s_q(x)^\top \nabla_y k(x,y) + s_q(y)^\top \nabla_x k(x,y) + \text{trace}(\nabla_x \nabla_y^\top k(x,y)). \tag{5.31}$$

The proof is in Section E.3.1. Next, we show the NKSD satisfies the properties of a divergence.

Proposition 5.6.4. *If Conditions 5.6.1 and 5.6.2 hold, then*

$$\textit{NKSD}(p(x)\|q(x)) \geq 0, \tag{5.32}$$

with equality if and only if $p(x) = q(x)$ almost everywhere.

The proof is in Section E.3.1. Unlike the standard KSD, but like the KL divergence, the NKSD exhibits subsystem independence[34,35,212]: if two distributions P and Q have the same independence

structure, then the total NKSD separates into a sum of individual NKSD terms. This is formalized in Proposition 5.6.6.

Condition 5.6.5 (Shared independence structure)**.** *Let $x = (x_1^\top, x_2^\top)^\top$ be a decomposition of a vector $x \in \mathbb{R}^d$ into two subvectors, x_1 and x_2. Assume $p(x)$ and $q(x)$ factor as $p(x) = p(x_1)p(x_2)$ and $q(x) = q(x_1)q(x_2)$, and that the kernel k factors as $k(x, y) = k_1(x_1, y_1)k_2(x_2, y_2)$ where k_1 and k_2 both satisfy Condition 5.6.2.*

Proposition 5.6.6 (Subsystem independence)**.** *If Conditions 5.6.1, 5.6.2, and 5.6.5 hold, then*

$$\textit{NKSD}(p(x)\|q(x)) = \textit{NKSD}(p(x_1)\|q(x_1)) + \textit{NKSD}(p(x_2)\|q(x_2)) \tag{5.33}$$

where the first term on the right-hand side uses kernel k_1 and the second term uses k_2.

See Section E.3.1 for the proof. Subsystem independence is powerful since it separates the problem of evaluating the foreground model from that of evaluating the background model. A modified version applies to the estimator $\widehat{\text{NKSD}}(p\|q)$ (Equation 5.5); see Proposition E.3.1.

5.6.2 Bernstein–von Mises theorem for the NKSD posterior

In this section, we establish asymptotic properties of the SVC and, more broadly, of its corresponding generalized posterior, which we refer to as the NKSD posterior, defined as

$$\pi_N(\theta) \propto \exp\Big(- \frac{N}{T} \widehat{\text{NKSD}}(p_0(x_\mathcal{F})\|q(x_\mathcal{F}|\theta))\Big)\pi(\theta). \tag{5.34}$$

In particular, in Theorem 5.6.9, we show that the NKSD posterior concentrates and is asymptotically normal, and we establish that the Laplace approximation to the SVC (Equation 5.10) is asymptotically correct. These results form a Bayesian counterpart to those of Barp et al. [17], who establish the consistency and asymptotic normality of minimum KSD estimators. Thus, in both the frequentist and Bayesian contexts, we can replace the average log likelihood with the negative KSD and obtain similar key properties. Our results in this section do not depend on whether or not we are working with a foreground subspace, so we suppress the $x_{\mathcal{F}}$ notation.

Let $\Theta \subseteq \mathbb{R}^m$, and let $\{Q_\theta : \theta \in \Theta\}$ be a family of probability measures on $\mathcal{X} \subseteq \mathbb{R}^d$ having densities $q_\theta(x)$ with respect to Lebesgue measure. For notational convenience, we sometimes write $q(x|\theta)$ instead of $q_\theta(x)$. Suppose the data $X^{(1)}, \ldots, X^{(N)}$ are i.i.d. samples from some probability measure P_0 on $\mathcal{X}$ having density $p_0(x)$ with respect to Lebesgue measure. To ensure the NKSD satisfies the properties of a divergence for all q_θ, and that convergence of $\widehat{\text{NKSD}}$ is uniform on compact subsets of Θ (Proposition E.3.2), we require the following.

Condition 5.6.7. *Assume Conditions 5.6.1 and 5.6.2 hold for p_0, k, and q_θ for all $\theta \in \Theta$. Further, assume that the kernel k has continuous and bounded partial derivatives up to and including second order, and $k(x, y) > 0$ for all $x, y \in \mathcal{X}$.*

Now we can set up the generalized posterior. First define

$$f_N(\theta) := \frac{1}{T}\widehat{\text{NKSD}}(p_0(x)\|q(x|\theta)) = \frac{1}{T}\frac{\sum_{i\neq j} u_\theta(X^{(i)}, X^{(j)})}{\sum_{i\neq j} k(X^{(i)}, X^{(j)})}, \tag{5.35}$$

where $u_\theta(x, y)$ is the $u(x, y)$ function from Equation 5.5 with q_θ in place of q. For the case of $N =$

1, we define $f_1(\theta) = 0$ by convention. Note that $-Nf_N(\theta)$ plays the role of the log likelihood. Also define

$$\begin{aligned} f(\theta) &:= \frac{1}{T}\textsc{nksd}(p_0(x)\|q(x|\theta)), \\ z_N &:= \int_\Theta \exp(-Nf_N(\theta))\pi(\theta)d\theta, \\ \pi_N(\theta) &:= \frac{1}{z_N}\exp(-Nf_N(\theta))\pi(\theta), \end{aligned} \tag{5.36}$$

where $\pi(\theta)$ is a prior density on Θ. Note that $\pi_N(\theta)d\theta$ is the NKSD posterior and z_N is the corresponding generalized marginal likelihood employed in the SVC. Denote the gradient and Hessian of f by $f'(\theta) = \nabla_\theta f(\theta)$ and $f''(\theta) = \nabla^2_\theta f(\theta)$, respectively. To ensure that the NKSD posterior is well defined and has an isolated maximum, we assume the following condition.

Condition 5.6.8. *Suppose $\Theta \subseteq \mathbb{R}^m$ is a convex set and (a) Θ is compact or (b) Θ is open and f_N is convex on Θ with probability 1 for all N. Assume $z_N < \infty$ a.s. for all N. Assume f has a unique minimizer $\theta_* \in \Theta$, $f''(\theta_*)$ is invertible, π is continuous at θ_*, and $\pi(\theta_*) > 0$.*

By Proposition 5.6.4, f has a unique minimizer whenever $\{Q_\theta : \theta \in \Theta\}$ is well-specified and identifiable, that is, when $Q_\theta = P_0$ for some θ and $\theta \mapsto Q_\theta$ is injective.

In Theorem 5.6.9 below, we establish the following results: (1) the minimum $\widehat{\textsc{nksd}}$ converges to the minimum NKSD; (2) π_N concentrates around the minimizer of the NKSD; (3) the Laplace approximation to z_N is asymptotically correct; and (4) π_N is asymptotically normal in the sense of Bernstein–von Mises. The primary regularity conditions we need for this theorem are restraints

on the derivatives of s_{q_θ} with respect to θ (Condition 5.6.10). Our proof of Theorem 5.6.9 relies on the theory of generalized posteriors developed by Miller [176]. We use $\|\cdot\|$ for the Euclidean–Frobenius norms: for vectors $A \in \mathbb{R}^D$, $\|A\| = (\sum_i A_i^2)^{1/2}$; for matrices $A \in \mathbb{R}^{D\times D}$, $\|A\| = (\sum_{i,j} A_{i,j}^2)^{1/2}$; for tensors $A \in \mathbb{R}^{D\times D\times D}$, $\|A\| = (\sum_{i,j,k} A_{i,j,k}^2)^{1/2}$; and so on.

Theorem 5.6.9. *If Conditions 5.6.7, 5.6.8, and 5.6.10 hold, then there is a sequence $\theta_N \to \theta_*$ a.s. such that:*

1. *$f_N(\theta_N) \to f(\theta_*)$, $f_N'(\theta_N) = 0$ for all N sufficiently large, and $f_N''(\theta_N) \to f''(\theta_*)$ a.s.,*
2. *letting $B_\epsilon(\theta_*) := \{\theta \in \mathbb{R}^m : \|\theta - \theta_*\| < \epsilon\}$, we have*

$$\int_{B_\epsilon(\theta_*)} \pi_N(\theta) d\theta \xrightarrow[N\to\infty]{\text{a.s.}} 1 \text{ for all } \epsilon > 0, \tag{5.37}$$

3.

$$z_N \sim \frac{\exp(-N f_N(\theta_N))\pi(\theta_*)}{|\det f''(\theta_*)|^{1/2}} \left(\frac{2\pi}{N}\right)^{m/2} \tag{5.38}$$

   almost surely, where $a_N \sim b_N$ means that $a_N/b_N \to 1$ as $N \to \infty$, and

4. *letting h_N denote the density of $\sqrt{N}(\theta - \theta_N)$ when θ is sampled from π_N, we have that h_N converges to $\mathcal{N}(0, f''(\theta_*)^{-1})$ in total variation, that is,*

$$\int_{\mathbb{R}^m} \left| h_N(\tilde{\theta}) - \mathcal{N}(\tilde{\theta} \mid 0, f''(\theta_*)^{-1}) \right| d\tilde{\theta} \xrightarrow[N\to\infty]{\text{a.s.}} 0. \tag{5.39}$$

The proof is in Section E.3.2. We write $\nabla_\theta^2 s_{q_\theta}$ to denote the tensor in $\mathbb{R}^{d\times m\times m}$ in which entry

(i, j, k) is $\partial^2 s_{q_\theta}(x)_i / \partial\theta_j \partial\theta_k$. Likewise, $\nabla^3_\theta s_{q_\theta}$ denotes the tensor in $\mathbb{R}^{d\times m\times m\times m}$ in which entry (i, j, k, ℓ) is $\partial^3 s_{q_\theta}(x)_i / \partial\theta_j \partial\theta_k \partial\theta_\ell$. We write $\mathbb{N}$ to denote the set of natural numbers.

Condition 5.6.10 (Stein score regularity)**.** *Assume $s_{q_\theta}(x)$ has continuous third-order partial derivatives with respect to the entries of θ on Θ. Suppose that for any compact, convex subset $C \subseteq \Theta$, there exist continuous functions $g_{0,C}, g_{1,C} \in L^1(P_0)$ such that for all $\theta \in C$, $x \in \mathcal{X}$,*

$$\begin{aligned} \|s_{q_\theta}(x)\| &\le g_{0,C}(x), \\ \|\nabla_\theta s_{q_\theta}(x)\| &\le g_{1,C}(x). \end{aligned} \tag{5.40}$$

Further, assume there is an open, convex, bounded set $E \subseteq \Theta$ such that $\theta_* \in E$, $\bar{E} \subseteq \Theta$, and the sets

$$\Big\{ \frac{1}{N} \sum_{i=1}^{N} \|\nabla^2_\theta s_{q_\theta}(X^{(i)})\| : N \in \mathbb{N}, \theta \in E \Big\}, \tag{5.41}$$

$$\Big\{ \frac{1}{N} \sum_{i=1}^{N} \|\nabla^3_\theta s_{q_\theta}(X^{(i)})\| : N \in \mathbb{N}, \theta \in E \Big\} \tag{5.42}$$

are bounded with probability 1.

Next, Theorem 5.6.11 shows that in the special case where $q_\theta(x)$ is an exponential family, many of the conditions of Theorem 5.6.9 are automatically satisfied.

Theorem 5.6.11. *Suppose $\{Q_\theta : \theta \in \Theta\}$ is an exponential family with densities of the form $q_\theta(x) = \lambda(x) \exp(\theta^\top t(x) - \kappa(\theta))$ for $x \in \mathcal{X} \subseteq \mathbb{R}^d$. Assume $\Theta = \{\theta \in \mathbb{R}^m : |\kappa(\theta)| < \infty\}$,*

and assume Θ is convex, open, and nonempty. Assume $\log \lambda(x)$ and $t(x)$ are continuously differentiable on $\mathcal{X}$, $\|\nabla_x \log \lambda(x)\|$ and $\|\nabla_x t(x)\|$ are in $L^1(P_0)$, and the rows of the Jacobian matrix $\nabla_x t(x) \in \mathbb{R}^{m \times d}$ are linearly independent with positive probability under P_0. Suppose Condition 5.6.7 holds, f has a unique minimizer $\theta_* \in \Theta$, the prior π is continuous at θ_*, and $\pi(\theta_*) > 0$. Then the assumptions of Theorem 5.6.9 are satisfied for all N sufficiently large.

The proof is in Section E.3.2.

5.6.3 Asymptotics of the Stein volume criterion

The Laplace approximation to the SVC uses the estimate $\widehat{\text{NKSD}}$ and its minimizer θ_N, rather than the true NKSD and its minimizer θ_*. To establish the consistency properties of the SVC, we need to understand the relationship between the two. To do so, we adapt a standard approach to performing such an analysis of the marginal likelihood, for instance, as in Theorem 1 of Dawid [50].

Theorem 5.6.12. *Assume the conditions of Theorem 5.6.9 hold, and assume $s_{q_{\theta_*}}$ and $\nabla_\theta|_{\theta=\theta_*} s_{q_\theta}$ are in $L^2(P_0)$. Then as $N \to \infty$,*

$$f_N(\theta_N) - f_N(\theta_*) = O_{P_0}(N^{-1}). \tag{5.43}$$

Further, if $\text{NKSD}(p_0(x)\|q(x|\theta_*)) > 0$ then

$$f_N(\theta_*) - f(\theta_*) = O_{P_0}(N^{-1/2}), \tag{5.44}$$

whereas if $\text{NKSD}(p_0(x)\|q(x|\theta_*)) = 0$ *then*

$$f_N(\theta_*) - f(\theta_*) = O_{P_0}(N^{-1}). \tag{5.45}$$

The proof is in Section E.3.3. Remarkably, Equation 5.45 shows that $f_N(\theta_*)$ converges to $f(\theta_*)$ more rapidly when the model is well-specified, specifically, at a $1/N$ rate instead of $1/\sqrt{N}$. This is unusual and is crucial for our results in Section 5.6.4. The standard log likelihood does not exhibit this rapid convergence; see Section E.2.1. This property of the NKSD derives from similar properties exhibited by the standard KSD (Theorem 4.1 in Liu et al. [158]). Combined with Theorem 5.6.9 (part 3), Theorem 5.6.12 implies that when the model is misspecified, the leading order term of $\log z_N$ is $-Nf(\theta_*)$, whereas when the model is well-specified, the leading order term is $-\frac{1}{2}\, m \log N$.

5.6.4 Data and model selection consistency of the SVC

In this section, we establish the asymptotic consistency of the Stein volume criterion (SVC) when used for data selection, nested data selection, model selection, and nested model selection; see Theorem 5.6.17. This provides rigorous justification for the claims in Section 5.3. These results are all in the context of pairwise comparisons between two models or two model projections, M_1 and M_2. Before proving the results, we formally define the consistency properties discussed in Section 5.3. Each property is defined in terms of a pairwise score $\rho(M_1, M_2)$, such as $\rho(M_1, M_2) = \log(\mathcal{K}_1/\mathcal{K}_2)$. For simplicity, we assume $\rho(M_1, M_2) = -\rho(M_2, M_1)$; this is satisfied for all of the cases we consider. Let $\dim(\cdot)$ denote the dimension of a real space.

Definition 5.6.13 (Data selection consistency)**.** *Consider foreground model projections* $M_j := \{q(x_{\mathcal{F}_j}|\theta) : \theta \in \Theta\}$ *for* $j \in \{1, 2\}$*. We say that* ρ *satisfies "data selection consistency" if* $\rho(M_1, M_2) \to \infty$ *as* $N \to \infty$ *when* M_1 *is well-specified with respect to* $p_0(x_{\mathcal{F}_1})$ *and* M_2 *is misspecified with respect to* $p_0(x_{\mathcal{F}_2})$*.*

Definition 5.6.14 (Nested data selection consistency)**.** *Consider foreground model projections* $M_j := \{q(x_{\mathcal{F}_j}|\theta) : \theta \in \Theta\}$ *for* $j \in \{1, 2\}$*. We say that* ρ *satisfies "nested data selection consistency" if* $\rho(M_1, M_2) \to \infty$ *as* $N \to \infty$ *when* M_1 *is well-specified with respect to* $p_0(x_{\mathcal{F}_1})$*,* $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$*, and* $\dim(\mathcal{X}_{\mathcal{F}_2}) < \dim(\mathcal{X}_{\mathcal{F}_1})$*.*

Definition 5.6.15 (Model selection consistency)**.** *Consider foreground models* $M_j := \{q_j(x_{\mathcal{F}}|\theta_j) : \theta_j \in \Theta_j\}$ *for* $j \in \{1, 2\}$*. We say that* ρ *satisfies "model selection consistency" if* $\rho(M_1, M_2) \to \infty$ *as* $N \to \infty$ *when* M_1 *is well-specified with respect to* $p_0(x_{\mathcal{F}})$ *and* M_2 *is misspecified.*

Definition 5.6.16 (Nested model selection consistency)**.** *Consider foreground models* $M_j := \{q_j(x_{\mathcal{F}}|\theta_j) : \theta_j \in \Theta_j\}$ *for* $j \in \{1, 2\}$*. We say that* ρ *satisfies "nested model selection consistency" if* $\rho(M_1, M_2) \to \infty$ *as* $N \to \infty$ *when* M_1 *is well-specified with respect to* $p_0(x_{\mathcal{F}})$*,* $M_1 \subset M_2$*, and* $\dim(\Theta_1) < \dim(\Theta_2)$*.*

In each case, ρ may diverge almost surely ("strong consistency") or in probability ("weak consistency"). Note that in Definitions 5.6.13–5.6.14, the difference between M_1 and M_2 is the choice of foreground data space $\mathcal{F}$, whereas in Definitions 5.6.15–5.6.16, M_1 and M_2 are over the same foreground space but employ different model spaces.

In Theorem 5.6.17, we show that the SVC has the asymptotic properties outlined in Section 5.3. In combination with the subsystem independence properties of the NKSD (Propositions 5.6.6 and E.3.1), Theorem 5.6.17 also leads to the conclusion that the SVC approximates the NKSD marginal likelihood of the augmented model (Equation 5.8). Our proof is similar in spirit to previous results for model selection with the standard marginal likelihood, notably those of Hong & Preston [108] and Huggins & Miller [118], but relies on the special properties of the NKSD marginal likelihood in Theorem 5.6.12.

Theorem 5.6.17. *For $j \in \{1, 2\}$, assume the conditions of Theorem 5.6.12 hold for model M_j defined on $\mathcal{X}_{\mathcal{F}_j}$, with density $q_j(x_{\mathcal{F}_j}|\theta_j)$ for $\theta_j \in \Theta_j \subseteq \mathbb{R}^{m_{\mathcal{F}_j,j}}$. Let $\mathcal{K}_{j,N}$ be the Stein volume criterion for M_j, with background model penalty $m_{\mathcal{B}_j} = m_{\mathcal{B}_j}(N)$, and let $\theta_{j,*} := \arg\min_{\theta_j} \textsc{nksd}(p_0(x_{\mathcal{F}_j})\|q_j(x_{\mathcal{F}_j}|\theta_j))$. Then:*

1. *If $m_{\mathcal{B}_j} = o(N/\log N)$ for $j \in \{1, 2\}$, then*

$$\frac{1}{N}\log\frac{\mathcal{K}_{1,N}}{\mathcal{K}_{2,N}} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\textsc{nksd}(p_0(x_{\mathcal{F}_2})\|q_2(x_{\mathcal{F}_2}|\theta_{2,*})) - \frac{1}{T}\textsc{nksd}(p_0(x_{\mathcal{F}_1})\|q_1(x_{\mathcal{F}_1}|\theta_{1,*})).$$

2. *If $\textsc{nksd}(p_0(x_{\mathcal{F}_1})\|q_1(x_{\mathcal{F}_1}|\theta_{1,*})) = \textsc{nksd}(p_0(x_{\mathcal{F}_2})\|q_2(x_{\mathcal{F}_2}|\theta_{2,*})) = 0$ and $m_{\mathcal{B}_2} - m_{\mathcal{B}_1}$ does not depend on N, then*

$$\frac{1}{\log N}\log\frac{\mathcal{K}_{1,N}}{\mathcal{K}_{2,N}} \xrightarrow[N\to\infty]{P_0} \frac{1}{2}(m_{\mathcal{F}_2,2} + m_{\mathcal{B}_2} - m_{\mathcal{F}_1,1} - m_{\mathcal{B}_1}).$$

3. *If $\textsc{nksd}(p_0(x_{\mathcal{F}_1})\|q_1(x_{\mathcal{F}_1}|\theta_{1,*})) = \textsc{nksd}(p_0(x_{\mathcal{F}_2})\|q_2(x_{\mathcal{F}_2}|\theta_{2,*}))$, $m_{\mathcal{B}_1} = c_{\mathcal{B}_1}\sqrt{N}$, and*

$m_{\mathcal{B}_2} = c_{\mathcal{B}_2}\sqrt{N}$, *where* $c_{\mathcal{B}_1}$ *and* $c_{\mathcal{B}_2}$ *are positive and constant in* N, *then*

$$\frac{1}{\sqrt{N}\log N}\log\frac{\mathcal{K}_{1,N}}{\mathcal{K}_{2,N}} \xrightarrow[N\to\infty]{P_0} \frac{1}{2}(c_{\mathcal{B}_2} - c_{\mathcal{B}_1}).$$

The proof is in Section E.3.4. In particular, assuming the conditions of Theorem 5.6.12, we obtain the following consistency results in terms of convergence in probability. Let $D_j := \textsc{nksd}(p_0(x_{\mathcal{F}_j})\|q_j(x_{\mathcal{F}_j}|\theta_{j,*}))$ for $j \in \{1,2\}$.

- If $m_{\mathcal{B}_j} = o(N/\log N)$ then the SVC exhibits data selection consistency and model selection consistency. This holds by Theorem 5.6.17 (part 1) since $D_2 > D_1 = 0$.
- If $m_{\mathcal{B}_1} = m_{\mathcal{B}_2}$ then the SVC exhibits nested model selection consistency. This holds by Theorem 5.6.17 (part 2) since $D_1 = D_2 = 0$, $m_{\mathcal{B}_2} - m_{\mathcal{B}_1} = 0$, and $m_{\mathcal{F}_2,2} > m_{\mathcal{F}_1,1}$.
- Consider a nested data selection problem with $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$. If (A) $m_{\mathcal{B}_2} - m_{\mathcal{B}_1}$ does not depend on N and $m_{\mathcal{F}_2,2} + m_{\mathcal{B}_2} > m_{\mathcal{F}_1,1} + m_{\mathcal{B}_1}$ or (B) $m_{\mathcal{B}_j} = c_{\mathcal{B}_j}\sqrt{N}$ and $c_{\mathcal{B}_2} > c_{\mathcal{B}_1} > 0$, then the SVC exhibits nested data selection consistency. Cases A and B hold by Theorem 5.6.17 (parts 2 and 3, respectively) since $D_1 = D_2 = 0$.

5.7 Application: Probabilistic PCA

Probabilistic principal components analysis (pPCA) is a commonly used tool for modeling and visualization. The basic idea is to model the data as linear combinations of k latent factors plus Gaussian noise. The inferred weights on the factors are frequently used to provide low-dimensional sum-

maries of the data, while the factors themselves describe major axes of variation in the data. In practice, pPCA is often applied in settings where it is likely to be misspecified – for instance, the weights are often clearly non-Gaussian. In this section, we show how data selection can be used to uncover sources of misspecification and to analyze how this misspecification affects downstream inferences.

The generative model used in pPCA is

$$\begin{aligned} Z^{(i)} &\sim \mathcal{N}(0, I_k), \\ X^{(i)} | Z^{(i)} &\sim \mathcal{N}(HZ^{(i)}, vI_d), \end{aligned} \tag{5.46}$$

independently for $i = 1, \ldots, N$, where I_k is the k-dimensional identity matrix, $Z^{(i)} \in \mathbb{R}^k$ is the weight vector for datapoint i, $H \in \mathbb{R}^{d \times k}$ is the unknown matrix of latent factors, and $v > 0$ is the variance of the noise. To form a Laplace approximation for the Stein volume criterion, we follow the approach developed by Minka [179] for the standard marginal likelihood. Specifically, we parameterize H as

$$H = U(L - vI_k)^{1/2} \tag{5.47}$$

where U is a $d \times k$ matrix with orthonormal columns (that is, it lies on the Stiefel manifold) and L is a $k \times k$ diagonal matrix. We use the priors suggested by Minka [179],

$$\begin{aligned} U &\sim \text{Uniform}(\mathcal{U}), \\ L_{ii} &\sim \text{InverseGamma}(\alpha/2,\ \alpha/2), \\ v &\sim \text{InverseGamma}\big((\alpha/2+1)(d-k) - 1,\ (\alpha/2)(d-k)\big), \end{aligned} \tag{5.48}$$

where $\mathcal{U}$ is the set of $d \times k$ matrices with orthonormal columns and L_{ii} is the ith diagonal entry of L. We set $\alpha = 0.1$ in the following experiments, and we use pymanopt[262] to optimize U over the Stiefel manifold (Section E.4).

5.7.1 Simulations

In simulations, we evaluate the ability of the SVC to detect partial misspecification. We set $d = 6$, draw the first four dimensions from a pPCA model with $k = 2$ and

$$H = \begin{pmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \\ -1 & -1 \end{pmatrix}, \tag{5.49}$$

and generate dimensions 5 and 6 in such a way that pPCA is misspecified. We consider two misspecified scenarios: scenario A (Figure 5.3a) is that

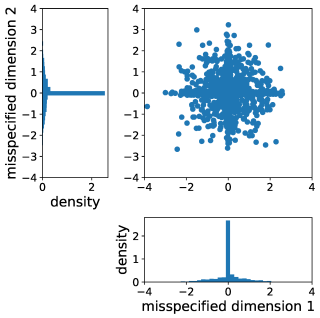
$$\begin{aligned} W^{(i)} &\sim \text{Bernoulli}(0.5), \\ X_{5:6}^{(i)} \mid W^{(i)} &\sim \mathcal{N}\left(0, \Sigma_{W^{(i)}}\right), \end{aligned} \tag{5.50}$$

where $\Sigma_{W^{(i)}} = (0.05)^{W^{(i)}} I_2$. Scenario B (Figure 5.3d) is the same but with

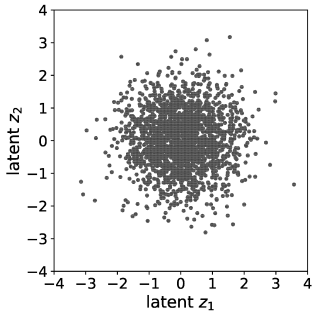
$$\Sigma_{W^{(i)}} = \begin{pmatrix} 1 & (-1)^{W^{(i)}} 0.99 \\ (-1)^{W^{(i)}} 0.99 & 1 \end{pmatrix}. \tag{5.51}$$

Scenario B is more challenging because the marginals of the misspecified dimensions are still Gaussian, and thus, misspecification only comes from the dependence between X_5 and X_6. As illustrated in Figures 5.3b and 5.3e, both kinds of misspecification are very hard to see in the lower-dimensional latent representation of the data.

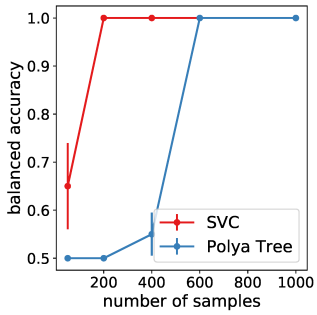
Our method can be used to both (i) detect misspecified subsets of dimensions, and (ii) conversely, find a maximal subset of dimensions for which the pPCA model provides a reasonable fit to the data. We set $T = 0.05$ in the SVC, based on the calibration procedure in Section E.1.1 (Section E.4.3). We use the Pitman-Yor mixture model expression for the background model dimension (Equation 5.3), with $\alpha = 0.5, \theta = 1$, and $D = 0.2$. This value of D ensures that the number of background model parameters per data dimension is greater than the number of foreground model parameters per data dimension except for at very small N, since there are two foreground parameters for each additional data dimension in the pPCA model, and $m_{\mathcal{B}} > 2\, r_{\mathcal{B}}$ for $N \geq 20$. We performed leave-one-out data selection, comparing the foreground space $\mathcal{X}_{\mathcal{F}_0} = \mathcal{X}$ to foreground spaces $\mathcal{X}_{\mathcal{F}_j}$ for $j \in \{1, \ldots, d\}$, which exclude the jth dimension of the data. We computed the log SVC ratio $\log(\mathcal{K}_j / \mathcal{K}_0) = \log \mathcal{K}_j - \log \mathcal{K}_0$ using the BIC approximation to the SVC (Section 5.2.3) and the approximate optima technique (Section 5.2.3). We quantify the performance



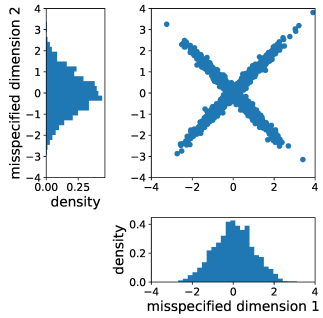
(a) Scenario A, misspecified dimensions.



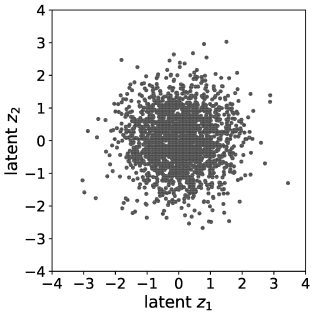
(b) Scenario A, pPCA latent space.



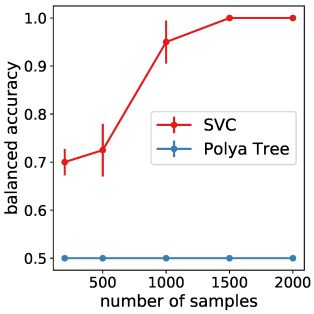
(c) Scenario A, accuracy in detecting misspecified dimensions.



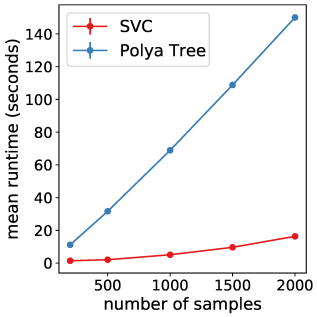
(d) Scenario B, misspecified dimensions.



(e) Scenario B, pPCA latent space.



(f) Scenario B, accuracy in detecting misspecified dimensions.



(g) Mean runtime over 5 repeats.

Figure 5.3: Data selection in the probabilistic PCA model.

of the method in detecting misspecified dimensions in terms of the balanced accuracy, defined as $(TN/N + TP/P)/2$ where TN is the number of true negatives (dimension by dimension), N is

the number of negatives, TP is the number of true positives, and P is the number of positives. Experiments were repeated independently five times. Figures 5.3c and 5.3f show that as the sample size increases, the SVC correctly infers that dimensions 1 through 4 should be included and dimensions 5 and 6 should be excluded.

5.7.2 Comparison with a nonparametric background model

To benchmark our method, we compare with an alternative approach that uses an explicit augmented model. The Pólya tree is a nonparametric model with a closed-form marginal likelihood that is tractable for one-dimensional data [151]. We define a flexible background model by sampling each dimension j of the background space independently as

$$X_j \sim \text{PolyaTree}(F, \tilde{F}, \eta), \tag{5.52}$$

with the Pólya tree constructed as by Berger & Guglielmi [21] (Section E.4.4). We set $F = \mathcal{N}(0, 10)$, $\tilde{F} = \mathcal{N}(0, 10)$, and $\eta = 1000$ so that the model is weighted only very weakly towards the base distribution.

We performed data selection using the marginal likelihood of the Pólya tree augmented model, computing the marginal of the pPCA foreground model using the approximation of Minka [179]. The accuracy results for data selection are in Figures 5.3c and 5.3f. On scenario A (Equation 5.50), the Pólya tree augmented model requires significantly more data to detect which dimensions are misspecified. On scenario B (Equation 5.51) the Pólya tree augmented model fails entirely, preferring

the full data space $\mathcal{X}_{\mathcal{F}_0} = \mathcal{X}$ which includes all dimensions (Figure 5.3f). The reason is that the background model is misspecified due to the assumption of independent dimensions, and thus, the asymptotic data selection results (Equations 5.15 and 5.19) do not hold. This could be resolved by using a richer background model that allows for dependence between dimensions, however, computing the marginal likelihood under such a model would be computationally challenging. Even with the independence assumption, the Pólya tree approach is already substantially slower than the SVC (Figure 5.3g).

5.7.3 Application to pPCA for single-cell RNA sequencing

Single-cell RNA sequencing (scRNAseq) has emerged as a powerful technology for high-throughput characterization of individual cells. It provides a snapshot of the transcriptional state of each cell by measuring the number of RNA transcripts from each gene. PCA is widely used to study scRNAseq datasets, both as a method for visualizing different cell types in the dataset and as a pre-processing technique, where the latent embedding is used for downstream tasks like clustering and lineage reconstruction[206,269]. We applied data selection to answer two practical questions in the application of probabilistic PCA to scRNAseq data: (1) Where is the pPCA model misspecified? (2) How does partial misspecification of the pPCA model affect downstream inferences?

Model criticism

Our first goal was to verify that the SVC provides reasonable inferences of partial model misspecification in practice. We examined two different scRNAseq datasets, focusing for illustration on a dataset from human peripheral blood mononuclear cells taken from a healthy donor, and preprocessed the data following standard procedures in the field (Section E.4.5). We subsampled each dataset to 200 genes (selected randomly from among the 2000 most highly expressed) and 2000 cells (selected randomly) for computational tractability, then mean-subtracted and standardized the variance of each gene, again following standard practice in the field. The number of latent components k was set to 3, based on the procedure of Minka [178]. We performed leave-one-out data selection, comparing the foreground space $\mathcal{X}_{\mathcal{F}_0} := \mathcal{X}$ to foreground spaces $\mathcal{X}_{\mathcal{F}_j}$ that exclude the jth gene. We computed the log SVC ratio $\log \mathcal{K}_j - \log \mathcal{K}_0$ using the BIC approximation to the SVC (Section 5.2.3) and the approximate optima technique (Section 5.2.3). We used the same setting of T and of $m_{\mathcal{B}}$ as was used in simulation, resulting in a background model complexity of $m_{\mathcal{B}} = 20\, r_{\mathcal{B}}$ for datasets of this size. Based on the SVC criterion, 162 out of 200 genes should be excluded from the foreground pPCA model, suggesting widespread partial misspecification. Figure 5.4 compares the histogram of individual genes to their estimated density under the pPCA model inferred for $\mathcal{X}_{\mathcal{F}_0} = \mathcal{X}$. Those genes most favored to be excluded (namely, UBE2V2 and IRF8) show extreme violations of normality, in stark contrast to those genes most favored to be included (MT-CO1 and RPL6).

Next, we compared the results of our data selection approach to a more conventional strategy for

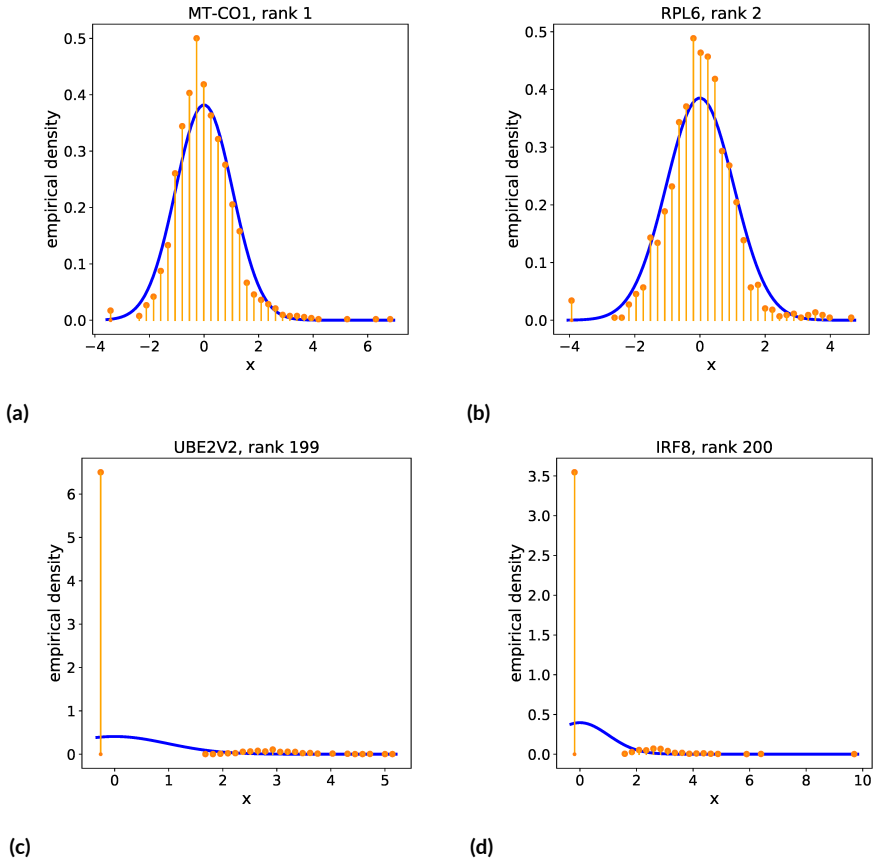


Figure 5.4: (a,b) Histograms of example genes (after pre-processing) selected to be included in the foreground space based on the log SVC ratio, $\log \mathcal{K}_j - \log \mathcal{K}_0$. The estimated density under the pPCA model is shown in blue. (c,d) Histograms of example genes selected to be excluded. Higher ranks (in each title) correspond to larger log SVC ratios.

model criticism. Criticism of partially misspecified models can be challenging in practice because misspecification of the model over some dimensions of the data can lead to substantial model-data mismatch in dimensions for which the model is indeed well-specified [125]. The standard approach to model criticism—first fit a model, then identify aspects of the data that the model poorly explains—can therefore be misleading if our aim is to determine how the model might be improved (e.g., in the context of "Box's loop", Blei [26]). In particular, standard approaches such as posterior predictive

checks will be expected to overstate problems with components of the model that are well-specified and understate problems with components of the model that are misspecified. Bayesian data selection circumvents this issue by evaluating augmented models, which replace potentially misspecified components of the model by well-specified components. To illustrate the difference between these approaches in practice, we compared the SVC to a closely analogous measurement of error for the full foreground model (inferred from $\mathcal{X}_{\mathcal{F}_0} = \mathcal{X}$),

$$\log \mathcal{E}_j - \log \mathcal{E}_0 := -\frac{N}{T}\widehat{\text{NKSD}}(p_0(x_{\mathcal{F}_j}) \| q(x_{\mathcal{F}_j}|\theta_{0,N})) + \frac{N}{T}\widehat{\text{NKSD}}(p_0(x) \| q(x|\theta_{0,N})) \tag{5.53}$$

where $\theta_{0,N} := \arg\min \widehat{\text{NKSD}}(p_0(x) \| q(x|\theta))$ is the minimum NKSD estimator for the foreground model when including all dimensions. This model criticism score evaluates the amount of model-data mismatch contributed by the subspace $\mathcal{X}_{\mathcal{B}_j}$ when modeling all data dimensions with the foreground model. For comparison, the BIC approximation to the log SVC ratio is

$$\begin{aligned} \log \mathcal{K}_j - \log \mathcal{K}_0 \approx -\frac{N}{T}\widehat{\text{NKSD}}(p_0(x_{\mathcal{F}_j}) \| q(x_{\mathcal{F}_j}|\theta_{j,N}) + \frac{N}{T}\widehat{\text{NKSD}}(p_0(x) \| q(x|\theta_{0,N})) \\ + \frac{m_{\mathcal{B}_j} + m_{\mathcal{F}_j} - m_{\mathcal{F}_0}}{2} \log\left(\frac{2\pi}{N}\right) \end{aligned} \tag{5.54}$$

where $\theta_{j,N} := \arg\min \widehat{\text{NKSD}}(p_0(x_{\mathcal{F}_j}) \| q(x_{\mathcal{F}_j}|\theta))$ is the minimum NKSD estimator for the projected foreground model applied to the restricted dataset, which we approximate as $\theta_{0,N}$ plus the implicit function correction derived in Section 5.2.3. Figure 5.5 illustrates the differences between the conventional criticism approach ($\log \mathcal{E}_j - \log \mathcal{E}_0$) and the log SVC ratio on an scRNAseq dataset. To enable direct comparison of the two methods, we focus on the lower order terms of Equation 5.54,

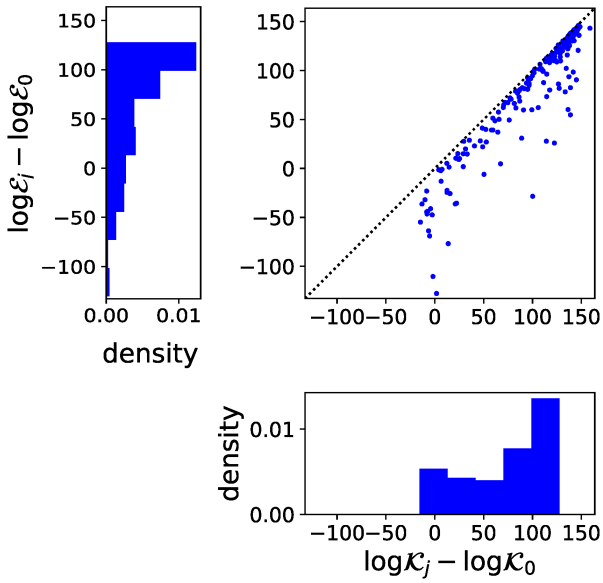


Figure 5.5: Scatterplot comparison and projected marginals of the leave-one-out log SVC ratio, $\log \mathcal{K}_j - \log \mathcal{K}_0$ (with $m_{\mathcal{B}_j} = m_{\mathcal{F}_0} - m_{\mathcal{F}_j}$), and the conventional full model criticism score, $\log \mathcal{E}_j - \log \mathcal{E}_0$, for each gene.

that is, we set $m_{\mathcal{B}_j} = m_{\mathcal{F}_0} - m_{\mathcal{F}_j}$. We see that the amount of error contributed by $\mathcal{X}_{\mathcal{B}_j}$, as judged by the SVC, is often substantially higher than the amount indicated by the conventional criticism approach, implying that the conventional criticism approach understates the problems caused by individual genes and, conversely, overstates the problems with the rest of the model.

Using the SVC instead of a standard criticism approach can also help clarify trends in where the proposed model fails. A prominent concern in scRNAseq data analysis is the common occurrence of cells that show exactly zero expression of a certain gene [197,102]. We found a Spearman correlation of $\rho = 0.89$ between the conventional criticism $\log \mathcal{E}_j - \log \mathcal{E}_0$ for a gene j and the fraction of cells with zero expression of that gene j, suggesting that this is an important source of model-data mismatch in this scRNAseq dataset, but not necessarily the only source (Figure 5.6a). However, the log SVC ratio yields a Spearman correlation of $\rho = 0.98$, suggesting instead that the amount of model-

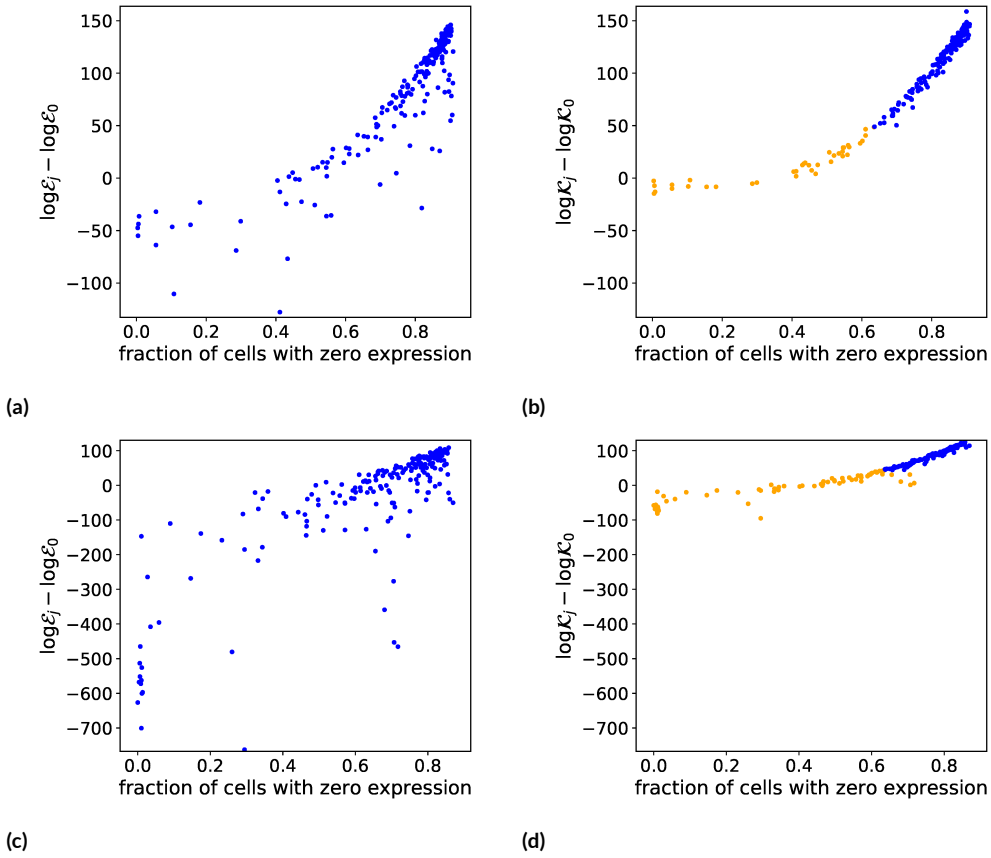


Figure 5.6: (a) Comparison of the conventional criticism score, for each gene j, and the fraction of cells that show zero expression of that gene j in the raw data. Spearman $\rho = 0.89, p < 0.01$. (b) Same as (a) but with the log SVC ratio. Spearman $\rho = 0.98, p < 0.01$. In orange are genes that would be included when using a background model with $c_{\mathcal{B}} = 20$ and in blue are genes that would be excluded. (c) Same as (a) for a dataset taken from a MALT lymphoma (Section E.4.5). Spearman $\rho = 0.81, p < 0.01$. (d) Same as (b) for the MALT lymphoma dataset. Spearman $\rho = 0.99$, $p < 0.01$.

data mismatch can be entirely explained by the fraction of cells with zero expression (Figure 5.6b).

These observations are repeatable across different scRNAseq datasets (Figure 5.6c, 5.6d).

Evaluating robustness

Data selection can also be used to evaluate the robustness of the foreground model to partial model misspecification. This is particularly relevant for pPCA on scRNAseq data, since the inferred latent embeddings of each cell are often used for downstream tasks such as clustering, lineage reconstruction, and so on. Misspecification may produce spurious conclusions, or alternatively, misspecification may be due to structure in the data that is scientifically interesting. To understand how partial misspecification of the pPCA model affects the latent representation of cells (and thus, downstream inferences), we performed data selection with a sequence of background model complexities $c_{\mathcal{B}}$, where $m_{\mathcal{B}} = c_{\mathcal{B}}\, r_{\mathcal{B}}$ (Figure 5.7a). We inferred the pPCA parameters based only on genes that the SVC selects to include in the foreground subspace. Figures 5.7e-5.7b visualize how the latent representation changes as $c_{\mathcal{B}}$ grows and fewer genes are selected. We can observe the representation morphing into a standard normal distribution, as we would expect in the case where the pPCA model is well-specified. However, the relative spatial organization of cells in the latent space remains fairly stable, suggesting that this aspect of the latent embedding is robust to partial misspecification. We can conclude that, at least in this example, misspecification strongly contributes to the non-Gaussian shape of the latent representation of the dataset, but not to the distinction between subpopulations.

5.8 Application: Glass model of gene regulation

A central goal in the study of gene expression is to discover how individual genes regulate one another other's expression. Early studies of single cell gene expression noted the prevalence of genes

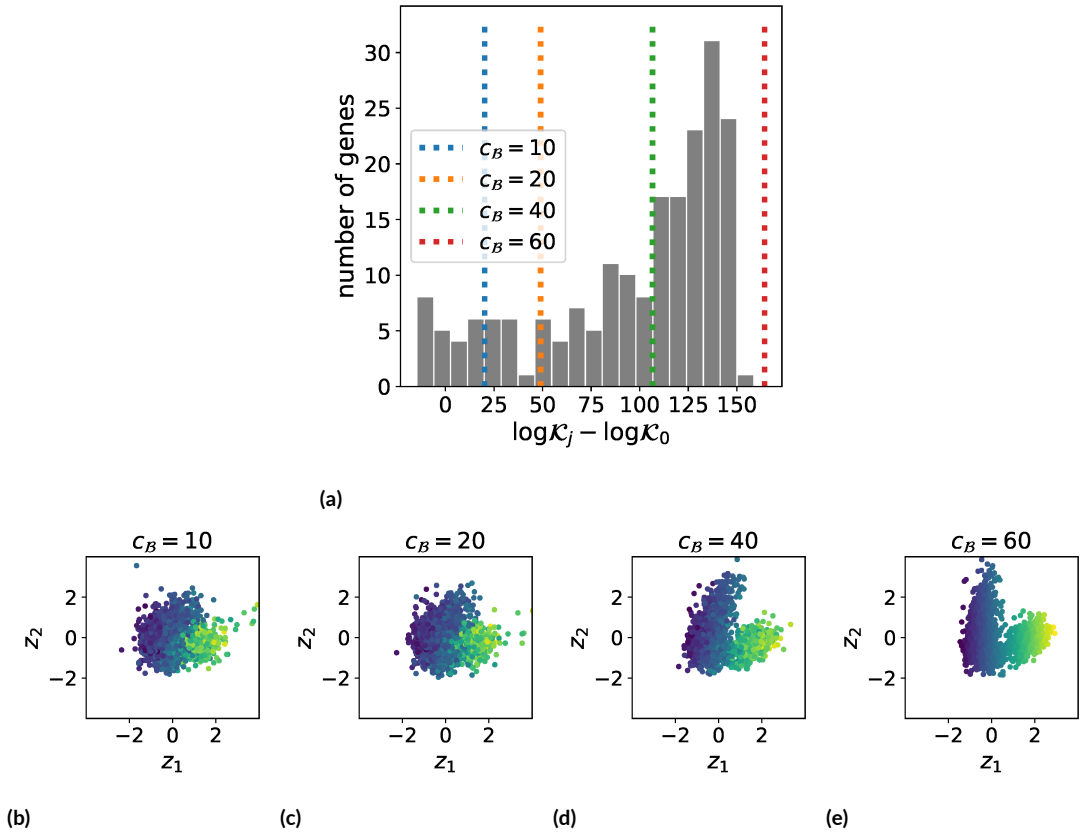


Figure 5.7: (a) Histogram of log SVC ratios $\log \mathcal{K}_j - \log \mathcal{K}_0$ for all 200 genes in the dataset (with $m_{\mathcal{B}_j} = m_{\mathcal{F}_0} - m_{\mathcal{F}_j}$). Dotted lines show the value of the volume correction term in the SVC for different choices of background model complexity $c_\mathcal{B}$; for each choice, genes with $\log \mathcal{K}_j - \log \mathcal{K}_0$ values above the dotted line would be excluded from the foreground subspace based on the SVC. (b) Posterior mean of the first two latent variables (z_1 and z_2), with the pPCA model applied to the genes selected with a background model complexity of $c_\mathcal{B} = 10$ (keeping 23 genes in the foreground). (c-e) Same as (b), but with $c_\mathcal{B} = 20$ (keeping 38 genes), $c_\mathcal{B} = 40$ (keeping 87 genes) and $c_\mathcal{B} = 60$ (keeping all 200 genes). In (a)-(d), the points are colored using the z_1 value when $c_\mathcal{B} = 60$.

that were bistable in their expression level[232,240]. This suggests a simple physical analogy: if individual gene expression is a two-state system, we might study gene regulation with the theory of interacting two-state systems, namely spin glasses. We can consider for instance a standard model of this type in which each cell i is described by a vector of spins $z_i = (z_{i1}, \ldots, z_{id})^\top$ drawn from an Ising model, specifying whether each gene $j \in \{1, \ldots, d\}$ is "on" or "off". In reality, gene expres-

sion lies on a continuum, so we use a continuous relaxation of the Ising model and parameterize each spin using a logistic function, setting $z_{ij1}(x_{ij}, \mu, \tau) = 1/(1 + \exp(-\tau(x_{ij} - \mu)))$ and $z_{ij2}(x_{ij}, \mu, \tau) = 1 - z_{ij1}(x_{ij}, \mu, \tau)$. Here, x_{ij} is the observed expression level of gene j in cell i, the unknown parameter μ controls the threshold for whether the expression of a gene is "on" (such that $z_{ij} \approx (1, 0)^\top$) or "off" (such that $z_{ij} \approx (0, 1)^\top$), and the unknown parameter $\tau > 0$ controls the sharpness of the threshold. The complete model is then given by

$$\begin{aligned} X^{(i)} &\sim p(x_i | H, J, \mu, \tau) \\ &:= \frac{1}{\mathcal{Z}_{H,J,\mu,\tau}} \exp\Big(\sum_j H_j^\top z_{ij}(x_{ij}, \tau, \mu) + \sum_{j'>j} z_{ij}^\top(x_{ij}, \tau, \mu) J_{jj'} z_{ij'}(x_{ij'}, \tau, \mu) \Big) \end{aligned}$$

where $\mathcal{Z}_{H,J,\mu,\tau}$ is the unknown normalizing constant of the model, and the vectors $H_j \in \mathbb{R}^2$ and matrices $J_{jj'} \in \mathbb{R}^{2\times 2}$ are unknown parameters. This model is motivated by experimental observations and is closely related to RNAseq analysis methods that have been successfully applied in the past[82,81,55,37,15,68,157,119,181,170]. However, from a biological perspective we can expect that serious problems may occur when applying the model naively to an scRNAseq dataset. Genes need not exhibit bistable expression: it is straightforward in theory to write down models of gene regulation that do not have just one or two steady states—gene expression may fall on a continuum, or oscillate, or have three stable states—and many alternative patterns have been well-documented empirically[10]. Interactions between genes may also be more complex than the model assumes, involving for instance three-way dependencies between genes. All of these biological concerns can potentially produce severe violations of the proposed two-state glass model's assumptions. Data selection provides

a method for discovering where the proposed model applies.

Applying standard Bayesian inference to the glass model is intractable, since the normalizing constant is unknown (it is an energy-based model). However, the normalizing constant does not affect the SVC, so we can still perform data selection. We used a variational approximation to the SVC (Section 5.2.3). We placed a Gaussian prior on H and a Laplace prior on each entry of J to encourage sparsity in the pairwise gene interactions; we also used Gaussian priors for μ and τ after applying an appropriate transform to remove constraints (Section E.5.1). Following the logic of stochastic variational inference, we optimized the variational approximation using minibatches of the data and a reparameterization gradient estimator [105,139,147]. We also simultaneously stochastically optimized the set of genes included in the foreground subspace, using the Leave-One-Out REINFORCE estimator [143,54]. We implemented the model and inference strategy within the probabilistic programming language Pyro by defining a new distribution with log probability given by the negative NKSD [23]. Pyro provides automated, GPU-accelerated stochastic variational inference, requiring less than an hour for inference on datasets with thousands of cells.

We examined three scRNAseq datasets, taken from (i) peripheral blood monocytes (PBMCs) from a healthy donor (2,428 cells), (ii) a MALT lymphoma (7,570 cells), and (iii) mouse neurons (10,658 cells) (Section E.5.2). We preprocessed the data following standard protocols and focused on 200 high expression, high variability genes in each dataset, based on the metric of Gigante et al. [90]. We set $T = 0.05$ as in Section 5.7, and used the Pitman-Yor expression for $m_{\mathcal{B}}$ (Equation 5.3) with $\alpha = 0.5, \theta = 1$ and $D = 100$. This value of D ensures that the number of background model parameters per data dimension is larger than the number of foreground model parameters per data

dimension except for at very small N; in particular, there are 798 foreground model parameter dimensions associated with each data dimension (from the 199 interactions $J_{jj'}$ that each gene has with each other gene, plus the contribution of H_j), and $m_{\mathcal{B}} > 798\, r_{\mathcal{B}}$ for $N \geq 13$. Our data selection procedure selects 65 genes (32.5%) in the PBMC dataset, 0 genes in the neuron dataset, and 187 genes (93.5%) in the MALT dataset; note that for a lower value of $m_{\mathcal{B}}$, in particular using $D = 10$, no genes are selected in the MALT dataset. These results suggest substantial partial misspecification in the PBMC and neuron datasets, and more moderate partial misspecification in the MALT dataset.

We investigated the biological information captured by the foreground model on the MALT dataset. In particular, we looked at the approximate NKSD posterior for the selected 187 genes, and compared it to the approximate NKSD posterior for the model when applied to all 200 genes. (Note that, since the glass model lacks a tractable normalizing constant, we cannot compare standard Bayesian posteriors.) Figure 5.8 shows, for a subset of selected genes, the posterior mean of the interaction energy $\Delta E_{jj'} := J_{jj'21} + J_{jj'12} - J_{jj'22} - J_{jj'11}$, that is, the total difference in energy between two genes being in the same state versus in opposite states. We focused on strong interactions with $|\Delta E_{jj'}| > 1$, corresponding to just 5% of all possible gene-gene interactions (Figure E.3).

One foreground gene with especially large loading onto the top principal component of the ΔE matrix is CD37 (Figure 5.8). In B-cell lymphomas, of which MALT lymphoma is an example, CD37 loss is known to be associated with decreased patient survival[293]. Further, previous studies have observed that CD37 loss leads to high NF-κB pathway activation[293]. Consistent with this observation, the estimated interaction energies in our model suggest that decreasing CD37 will

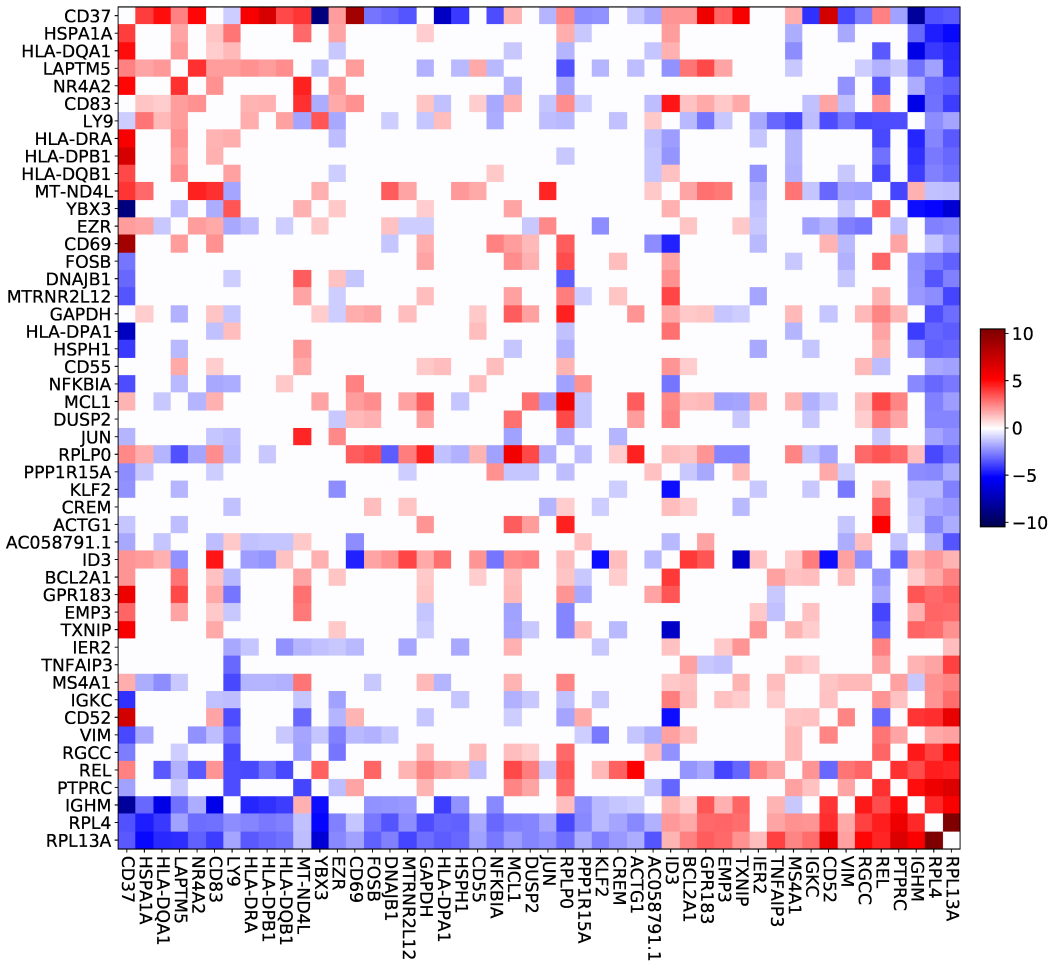


Figure 5.8: Posterior mean interaction energies $\Delta E_{jj'} := J_{jj'21} + J_{jj'12} - J_{jj'22} - J_{jj'11}$ for a subset of the selected genes. For visualization purposes, weak interactions ($|\Delta E_{jj'}| \leq 1$) are set to zero, and genes with less than 10 total strong connections are not shown. Genes are sorted based on their (signed) projection onto the top principal component of the ΔE matrix.

lead to higher expression of REL, an NF-κB transcription factor ($\Delta E_{\text{CD37,REL}} = 2.5$), decreased expression of NKFBIA, an NF-κB inhibitor ($\Delta E_{\text{CD37,NKFBIA}} = -3.6$), and higher expression of BCL2A1, a downstream target of the NF-κB pathway ($\Delta E_{\text{CD37,BCL2A1}} = 2.1$). Separately, a knockout study of Cd37 in B-cell lymphoma in mice does not show IgM expression [51], consis-

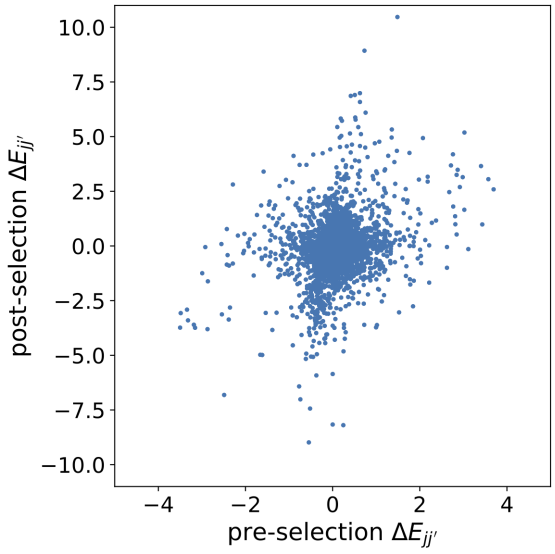


Figure 5.9: Comparison of posterior mean interaction energies $\Delta E_{jj'}$ for a model applied to all 200 genes (pre-data selection) to those learned from a model applied to the selected foreground subspace (post-data selection). Each point corresponds to a pairwise interaction between two of the selected 187 genes.

tent with our model ($\Delta E_{\text{CD37,IGHM}} = -8.2$). The same study does show MHC-II expression, and our model predicts the same result, for HLA-DQ in particular ($\Delta E_{\text{CD37,HLA-DQA1}} = 5.0$, $\Delta E_{\text{CD37,HLA-DQB1}} = 3.7$). These results suggest that the data selection procedure can successfully find systems of interacting genes that can plausibly be modeled as a spin glass, and which, in this case, are relevant for cancer.

To investigate whether data selection provided a benefit in this analysis, we compare with the results obtained by applying the foreground model to the full dataset of all 200 genes. All but one of the interactions listed above have $|\Delta E| < 1$ in the full foreground model, and three have opposite signs ($\Delta E_{\text{CD37,NFKBIA}} = +0.7$, $\Delta E_{\text{CD37,IGHM}} = +0.0$, $\Delta E_{\text{CD37,HLA-DQB1}} = -0.6$); see Figure E.4. Across all 187 selected genes, we find only a moderate correlation between the interaction

energies estimated when using the full foreground model compared with the data selection-based model (Spearman's rho $= 0.30$, $p < 0.01$; Figure 5.9). These results show that using data selection can lead to substantially different, and arguably more biologically plausible, downstream conclusions as compared to naive application of the foreground model to the full dataset.

As a simple alternative, one might wonder whether genes that are poorly fit by the model could be identified simply by looking their posterior uncertainty under the full foreground model. This simple approach does not work well, however, since it is possible for parameters to have low uncertainty even when the model poorly describes the data. Indeed, we found that examining uncertainty in the glass model does not lead to the same conclusions as performing data selection: the genes excluded by our data selection procedure are not the ones with the highest uncertainty in their interactions (as measured by the mean posterior standard deviation of $\Delta E_{jj'}$ under the NKSD posterior), though they do have above average uncertainty (Figure E.5a). Instead, the genes excluded by our data selection procedure are the ones with the highest fraction of cells with zero expression, violating the assumptions of the foreground model (Figure E.5b). These results show how data selection provides a sound, computationally tractable approach to criticizing and evaluating complex Bayesian models.

5.9 Discussion

Statistical modeling is often described as an iterative process, where we design models, infer hidden parameters, critique model performance, and then use what we have learned from the critique to

design new models and repeat the process[85]. This process has been called "Box's loop"[26]. From one perspective, data selection offers a new criticism approach. It goes beyond posterior predictive checks and related methods by changing the model itself, replacing potentially misspecified components with a flexible background model. This has important practical consequences: since misspecification can distort estimates of model parameters in unpredictable ways, predictive checks are likely to indicate mismatch between the model and the data across the entire space $\mathcal{X}$ even when the proposed parametric model is only partially misspecified. Our method, by contrast, reveals precisely those subspaces of $\mathcal{X}$ where model-data mismatch occurs.

From another perspective, data selection is outside the design-infer-critique loop. An underlying assumption of Box's loop is that scientists want to model the entire dataset. As datasets get larger, and measurements get more extensive, this desire has led to more and more complex (and often difficult to interpret) models. In experimental science, however, scientists have often followed the opposite trajectory: faced with a complicated natural phenomenon, they attempt to isolate a simpler example of the phenomenon for close study. Data selection offers one approach to formalizing this intuitive idea in the context of statistical analysis: we can propose a simple parametric model and then isolate a piece of the whole dataset—a subspace $\mathcal{X}_{\mathcal{F}}$—to which this model applies. When working with large, complicated datasets, this provides a method of searching for simpler phenomena that are hypothesized to exist.

6

Conclusion

Measuring and making sequences is central to modern biology, biotechnology and biomedicine. This dissertation has presented generative statistical methods for biological sequences, which enable inference from complex sequence data, rigorous accounting of uncertainty, and prediction of unobserved or future sequences that can be made in the laboratory. Our focus has been on addressing fundamental statistical problems: regression (Chapter 1), latent variable modeling (Chapter 1),

density estimation (Chapter 2), goodness-of-fit testing (Chapter 2), two-sample testing (Chapter 2), sampling (Chapter 3) and robust estimation (Chapter 4). Our new methods directly generalize and replace widely successful heuristic methods for biological sequence analysis: MuE observation models generalize alignment preprocessing methods (Chapter 1), BEAR tests generalize kmer spectra comparison methods (Chapter 2), and variational synthesis generalizes error prone PCR protocols (Chapter 3). In some cases our attempts to find more rigorous versions of existing methods led to novel statistical questions that had not been previously studied for any type of data. In particular, our attempts to generalize profile hidden Markov model search algorithms led to Chapter 5, which formalizes and studies the broader problem of data selection. Overall, the principles and methods developed in this dissertation contribute to an emerging toolbox of generative statistical methods for biological sequences. We next outline two key directions for future work that can make use of and expand this toolbox.

6.1 Latent and hierarchical structure

For many scientific questions, it is important to incorporate latent and hierarchical structure into generative sequence models. Consider for example the problem of forecasting pathogen evolution. Epidemiological models of the spread of infection over time and space have been well-studied [36]. Models of viral population dynamics, which account for inter-strain competition under selective pressure from the immune system, have also been studied, along with phylogenetic methods for predicting evasion of the immune system [146,163,188]. Fitness models, based on evolutionary multi-species

sequence data, have been shown to predict viral protein function[215,235]. Any or all of these models may be informative in forecasting pathogen sequence evolution, including not only future strains of existing human pathogens but also novel zoonotic spillovers. Generative Bayesian modeling offers a rigorous framework for combining information from different types of data (e.g. other species' genomes, infection counts, etc.) and accounting for complex dynamics (e.g. the time course of infection, immunological responses, etc.), via hierarchical parameter sharing and latent structure. Indeed, we can extend arbitrary continuous vector space dynamics models to sequence space using the MuE observation distribution (Chapter 1), while probabilistic programming languages such as Pyro[23] enable building and inferring complex hierarchical models. BEAR goodness-of-fit and two-sample tests (Chapter 2) allow these models to be criticized and checked. Thus, it seems possible to combine our piecemeal understanding of pathogen evolution into larger generative sequence models that can better forecast future sequences, and use these models to generate large libraries of likely future sequences with which we can test candidate drugs and diagnostics prospectively. Similar opportunites abound in other areas of biological sequence statistics, for instance in forecasting changes in the immune system and in the microbiome.

6.2 Causal inference

Another important area for future work is causal inference[195]. Consider, for example, questions at the intersection of microbiome, diet and human health, such as occur in the context of inflammatory bowel diseases[161,160,228]. We might, for instance, be interested in measuring the causal im-

pact of changes in diet on the metagenome, at nucleotide resolution. To estimate this causal effect we would need a regression model which describes the conditional distribution of the outcome (in this case, sequences) given treatment (in this case, diet) and any confounders. If we are focused on a particular protein or RNA molecule, we can use a MuE regression model (Chapter 1); if we are interested in the entire metagenome, and the treatment and confounder are discrete and low-dimensional, we can adapt BEAR models for regression (Chapter 2).

Generative sequence models are not just useful, however, when we are interested in effects *on* sequences; they can also be useful in understanding the effects *of* sequences. Consider the problem of estimating the impact of changes in the metagenome on disease, with diet a confounder that can affect both the microbiome and disease. One way of adjusting for confounding is by using propensity scores, which require a generative regression model for the treatment, i.e. sequences [120]. Or we might be interested in causal inference problems where the metagenome is itself a confounder, in which case to perform a backdoor adjustment we would need a density estimator for sequences; we can apply the BEAR model (Chapter 2). Note also that misspecification can bias causal estimates, so effective nonparametric tests are especially important in causal inference; we can apply BEAR tests (Chapter 2). Thus, our generative statistical methods open up new strategies for observational causal inference with biological sequence data, beyond the naive "no confounder" assumption, with possible applications in microbiology, immunology, evolutionary biology, agriculture and beyond.

6.3 Broader implications

Increasing capacity to learn from complex biological sequence data can potentially have indirect impacts on society, beyond direct technological applications in developing new therapies, diagnostics, enzymes, etc.. In particular, it may substantially increase the value of biological sequence data, as has been repeatedly seen in other areas of machine learning[300]. This includes sequence data not just from humans and their pathogens, but from across all of life. For example, fitness estimation methods, such as those discussed in Chapter 4, directly translate genome data from highly diverse organisms into technologies for diagnosing disease and engineering proteins[80,224]. This makes biodiversity important not just in terms of the moral and ecological value of preserving unique species, but also in terms of medical and economic value. The same is true of other emerging applications of biological sequence statistics, such as genome mining[6,59]. Speculatively, the growth of companies and organizations that depend on biological sequence data could potentially produce new economic and political interests invested in biodiversity, as well as new incentives for privatization of biological sequence data.

6.4 Conclusions

Generative statistical methods offer a powerful, rigorous and flexible strategy for learning from sequence data and forming predictions of new sequences that can be constructed in the laboratory. Their potential for widespread impact will only grow as technologies for sequencing and synthesis advance. The goal of this dissertation has been to help build stronger foundations for biological se-

quence statistics, firmly rooted in the underlying statistical and biophysical theory. However, much work remains to be done to realize the full potential of generative statistical methods for biological sequences.

A

Supplementary Material for Chapter 1

A.1 Overview diagram and notation

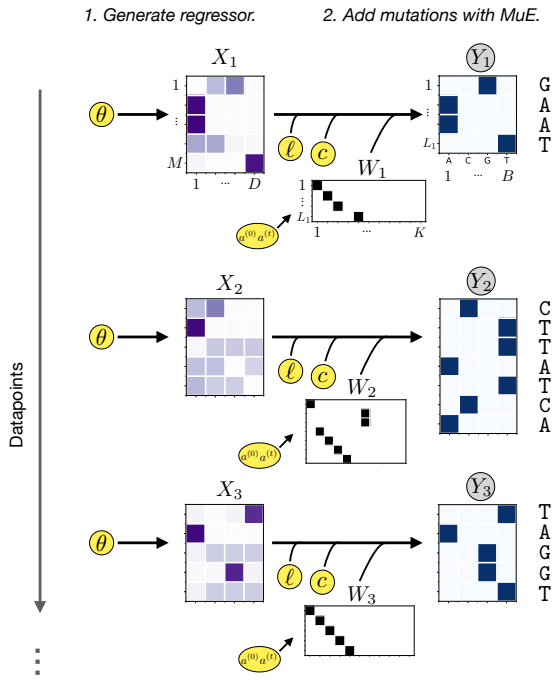


Figure A.1: MuE observation model. Overview of the generative process in MuE observation models. First, the latent regressor sequence X_i is sampled. Then, the MuE distribution adds mutations to generate Y_i. A latent variable W_i controls the pattern of insertions and deletions. Global parameters that must be inferred are highlighted in yellow.

Table A.1: Notation for MuE observation models A summary of the notation used in the main text, for convenient reference. Space refers to the space the variable lives in, i.e. $N \in \mathbb{N}$, the set of positive non-zero integers.

Variable	Space	Generation	Description
N	$\mathbb{N}$	Observed	Number of observed sequences.
$\mathcal{B}$	finite set	Hyperparameter	Alphabet (e.g. $\{A, T, G, C\}$ for DNA).
B	$\mathbb{N}$	$B := \lvert\mathcal{B}\rvert$	Alphabet size.
M	$\mathbb{N}$	Hyperparameter	Length of latent regressor sequence. (Typically set to be somewhat larger than $\max_i L_i$.)
D	$\mathbb{N}$	Hyperparameter	Size of latent regressor sequence's alphabet. (Typically set to be somewhat larger than B.)
V_i	$\mathbb{R}^{M \times D}$	$V_i \sim p_\theta$	Output of the initial continuous-space generative model.
X_i	$(\Delta_D)^M$	$X_i := \mathrm{softmax}(V_i)$	Latent regressor sequence, intuitively the "precursor" or "ancestor" to Y_i.
$a^{(0)}$	Δ_K	Parameter	Controls the probability of insertion and deletion mutations occurring in X_i.
$a^{(t)}$	$(\Delta_K)^K$	Parameter	Controls the probability of insertion and deletion mutations occurring in X_i. Must satisfy Condition 2.2.
W_i	$\{1, \ldots, K\}^{L_i}$	$W_i \sim \mathrm{MarkovModel}(a^{(0)}, a^{(t)})$	The hidden Markov model state variable, which defines a latent alignment between X_i and Y_i. (W_i is marginalized out during inference.)
c	$(\Delta_D)^{M+1}$	Parameter	Controls the probability of the insertion sequence letters (but not the presence or absence of the insertion).
ℓ	$(\Delta_B)^D$	Parameter	Substitution matrix.
Y_i	$\mathcal{B}^{L_i}$	$Y_i \sim \mathrm{MuE}(X_i, c, a^{(0)}, a^{(t)})$	Observed sequence, intuitively generated by mutating X_i with substitutions, insertions and deletions.
L_i	$\mathbb{N}$	$L_i := \lvert Y_i\rvert$	Length of observed sequence Y_i.

A.2 Theory

A.2.1 Illustrating MSA pathologies

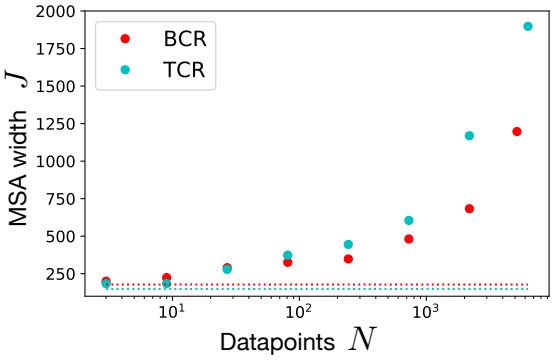


Figure A.2: MSA width can diverge with dataset size. MSA width J as a function of sequences N in the dataset. BCR is a B cell receptor dataset, TCR a T cell receptor dataset.

To illustrate the problems described in Section 4.1 of the main text, we examined a B cell receptor dataset and a T cell receptor dataset (the 10x Genomics datasets described in Section A.6). Sequences were subsampled and aligned using MUSCLE[70], a standard MSA algorithm. Figure A.2 shows the growth in MSA width J as a function of the subsampled dataset size.

A.2.2 Proof of Proposition 4.4

To prove the result, we will examine each existing model individually; exact specifications and assumptions for each model are provided in their corresponding section. The probability of the Markov chain terminating given that it is at a state k is denoted $t_k^{(t)}$, and the probability of the Markov chain terminating initially (that is, of the Markov chain taking zero steps) is denoted $t^{(0)}$.

Algorithm 1 Pairwise alignment construction

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

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

It is useful for understanding the following results to have in mind a particular example to illus-

trate the definitions in the main text.

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

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

Remark A.2.1. *Given sequences X and Y of length M and L respectively, $(j_1, \ldots, j_L)$ and $(g_1, \ldots, g_L)$ uniquely define a pairwise alignment $\mathcal{A}$.*

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

Thorne-Kishino-Felsenstein

The Thorne-Kishino-Felsenstein (TKF) model is a continuous-time stochastic process model of sequence evolution that satisfies detailed balance[257].

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

$$c_{m,b} := \pi_b. \tag{A.1}$$

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

$$\beta(\tau) := \frac{1 - e^{-(\mu-\lambda)\tau}}{\mu - \lambda e^{-(\mu-\lambda)\tau}}. \tag{A.2}$$

Define the transition matrix and termination probability as

$$a^{(t)}_{k,k'} := \begin{cases} [\mu\beta(\tau)]^{m'-m-1+g} e^{-\mu\tau}[1-\lambda\beta(\tau)] & \text{if } m-g < m' < M+1 \\ & \text{and } g' = 0 \\ \lambda\beta(\tau) & \text{if } m-g = m'-1 \text{ and } g' = 1 \\ [\mu\beta(\tau)]^{m'-m-2+g}[1-e^{-\mu\tau}-\mu\beta(\tau)][1-\lambda\beta(\tau)] & \text{if } m-g < m'-1 \text{ and } g' = 1 \\ 0 & otherwise. \end{cases} \tag{A.3}$$

$$t^{(t)}_k := [1-\lambda\beta(\tau)][\mu\beta(\tau)]^{M-m+g} \tag{A.4}$$

The initial transition vector follows the same form, and can be written as $a^{(0)}_k := a^{(t)}_{0,k}$, and the initial termination probability can be written $t^{(0)} := t^{(t)}_0$ (i.e. they match Equations A.3 and A.4 with $(m, g) = (0, 0)$ plugged in). Let $s > 0$ be the TKF substitution rate parameter and define the substitution matrix

$$\ell_{b,b'} := \begin{cases} e^{-s\tau} + \pi_{b'}(1-e^{-s\tau}) & \text{if } b = b' \\ \pi_{b'}(1-e^{-s\tau}) & \text{if } b \neq b' \end{cases} \tag{A.5}$$

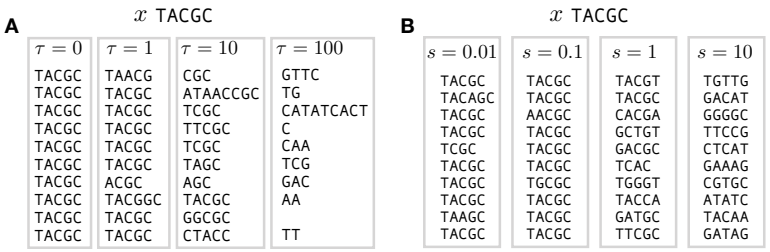


Figure A.3: Samples from the Thorne-Kishino-Felsenstein model. Initial sequence TACGC, $\mu = 0.02$, and $\lambda = 0.01$. A. $s = 0.01$ and varying τ. B. $\tau = 1$ and varying s.

With these definitions, $Y \sim \text{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$ is the distribution of the Thorne-Kishino-Felsenstein model after the sequence X evolves for time τ. Note that the limit $\tau \to 0$ is the no-mutation limit. Figure A.3 illustrates samples from the TKF model with changing parameters.

Proof We will show that the joint probability of W and Y under the MuE distribution is identical to the joint probability of the corresponding alignment pairwise alignment and Y under the TKF model. To start, we systematically enumerate state transitions in the MuE model and compute the corresponding probability factor under the TKF alignment scoring system. Our alignment notation in this section follows the original paper. "x" represents a residue and "-" a gap. "." represents the "immortal link" in the model, the start of the sequence. We use "$" as a termination symbol. Following the original paper, we define, for $\nu \in \{1, 2, \ldots\}$,

$$\begin{aligned} p_\nu(\tau) &:= e^{-\mu\tau}[1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \\ p'_0(\tau) &:= \mu\beta(\tau) \\ p'_\nu(\tau) &:= [1 - e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \\ p''_\nu(\tau) &:= [1 - \lambda\beta(\tau)][\lambda\beta(\tau)]^{\nu-1} \end{aligned} \tag{A.6}$$

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

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

   ```
   x | x ... x x
   x | - ... - x
   ```

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

   ```
   - | x ... x x
   x | - ... - x
   ```

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

   ```
   x - ... - | -
   x x ... x | x
   ```

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

```
x - ... - | -

- x ... x | x
```

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

```
x | -

x | x
```

6. Transitioning from $(m, 0)$ to $(m' > m + 1, 1)$. This gives a factor $[p'_0(\tau)]^{m'-m-2}p'_1(\tau) = [\mu\beta(\tau)]^{m'-m-2}[1 - e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)]$.

```
x | x ... x -

x | - ... - x
```

7. Transitioning from $(m, 1)$ to $(m' > m, 1)$. This gives a factor $[p'_0(\tau)]^{m'-m-1}p'_1(\tau) = [\mu\beta(\tau)]^{m'-m-1}[1 - e^{-\mu\tau} - \mu\beta(\tau)][1 - \lambda\beta(\tau)]$.

```
- | x ... x -

x | - ... - x
```

8. Terminating after $(m, 0)$. This gives a factor $[p'_0(\tau)]^{M-m} = [\mu\beta(\tau)]^{M-m}$.

```
x | x ... x $

x | - ... - $
```

9. Terminating after $(m, 1)$. This gives a factor $[p'_0(\tau)]^{M+1-m} = [\mu\beta(\tau)]^{M+1-m}$.

```
- | x ... x $

x | - ... - $
```

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

```
. | -

. | x
```

11. Initial transition to $(m, 0)$. This gives a factor

$p_1''(\tau)[p_0'(\tau)]^{m-1}p_1(\tau) = [1 - \lambda\beta(\tau)][\mu\beta(\tau)]^{m-1}e^{-\mu\tau}[1 - \lambda\beta(\tau)]$.

```
. | x ... x x

. | - ... - x
```

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

```
. | x ... x -

. | - ... - x
```

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

```
. | x ... x $

. | - ... - $
```

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

$$(m,g) \to (m',g') : \begin{cases} [\mu\beta(t)]^{m'-m-1+g} e^{-\mu t}[1-\lambda\beta(t)] \\ \qquad \text{if } m-g < m' < M+1 \text{ and } g' = 0 \\ \lambda\beta(t) \text{ if } m-g = m'-1 \text{ and } g' = 1 \\ [\mu\beta(t)]^{m'-m-2+g}[1-e^{-\mu t}-\mu\beta(t)][1-\lambda\beta(t)] \\ \qquad \text{if } m-g < m'-1 \text{ and } g' = 1 \\ 0 \qquad otherwise \end{cases} \tag{A.7}$$

$$(m,g) \to \textit{termination} : [\mu\beta(t)]^{M-m+g}$$

And with each initial transition

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

$$\textit{initial} \rightarrow \textit{termination}: [1-\lambda\beta(t)][\mu\beta(t)]^{M} \tag{A.8}$$

However, these are unnormalized probability factors, not complete probabilities. Note that every alignment will include a factor $[1-\lambda\beta(t)]$, which in the original TKF description is associated with the initial transition. However, if we instead rearrange this factor and assign it to the final transition we obtain the transition matrix given in Equation A.3. We can check that this transition matrix

normalized. From a state $(m, 0)$, the total outward transition probability is one:

$$\begin{aligned}
&\sum_{m'=m+1}^{M} [\mu\beta]^{m'-m-1} e^{-\mu\tau}[1-\lambda\beta] + \lambda\beta + \sum_{m'=m+2}^{M+1} [\mu\beta]^{m'-m-2}[1-e^{-\mu\tau}-\mu\beta][1-\lambda\beta] \\
&\quad + [\mu\beta]^{M-m}(1-\lambda\beta) \\
&\quad = \frac{1-(\mu\beta)^{M-m}}{1-\mu\beta}[1-e^{-\mu\tau}-\mu\beta+e^{-\mu\tau}][1-\lambda\beta] + \lambda\beta + [\mu\beta]^{M-m}(1-\lambda\beta) \\
&\quad = 1-(\mu\beta)^{M-m}[1-\lambda\beta] + [\mu\beta]^{M-m}(1-\lambda\beta) \\
&\quad = 1.
\end{aligned} \tag{A.9}$$

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

$$\begin{aligned}
&\sum_{m'=m}^{M} [\mu\beta]^{m'-m} e^{-\mu\tau}[1-\lambda\beta] + \lambda\beta + \sum_{m'=m+1}^{M+1} [\mu\beta]^{m'-m-1}[1-e^{-\mu\tau}-\mu\beta][1-\lambda\beta] \\
&\quad + [\mu\beta]^{M+1-m}(1-\lambda\beta) \\
&\quad = \frac{1-(\mu\beta)^{M+1-m}}{1-\mu\beta}[1-e^{-\mu\tau}-\mu\beta+e^{-\mu\tau}][1-\lambda\beta] + \lambda\beta + [\mu\beta]^{M+1-m}(1-\lambda\beta) \\
&\quad = 1-(\mu\beta)^{M+1-m}[1-\lambda\beta] + [\mu\beta]^{M+1-m}(1-\lambda\beta) \\
&\quad = 1.
\end{aligned} \tag{A.10}$$

Conditional on the mth residue of X being aligned to the lth residue of Y (i.e. $w_l = m$), the TKF model specifies that the probability of y_l given x_m is $\sum_{b,b'} x_{m,b}\ell_{b,b'}y_{l,b'}$, which is identical to the probability under the MuE model. In the case where the lth residue of y is aligned to a gap (i.e.

$g_l = 1$), the TKF model says the probability of choosing the specific base b is π_b, the equilibrium probability of the base. We can check that the MuE provides the same factor:

$$\begin{aligned} p_{\text{MuE}}(y_{l,b} = 1 | w, x, c, \ell) &= \sum_{b'} c_{m,b'} \ell_{b',b} \\ &= \pi_b e^{-s\tau} + (\pi_b)^2 (1 - e^{-s\tau}) + \sum_{b'' \neq b} \pi_{b''} \pi_b (1 - e^{-s\tau}) \\ &= \pi_b e^{-s\tau} + \pi_b (1 - e^{-s\tau}) = \pi_b. \end{aligned} \tag{A.11}$$

□

Pair HMM

The pair HMM model generates pairwise alignments by switching between three states: (1) a state emitting residues in both X and Y (a match state), (2) a state emitting a residue in X and a gap in the alignment of Y, and (3) a state emitting a gap in the alignment of X and a residue in Y (Durbin et al.[67], Chapter 4.1).

Statement Figure A.4 shows a standard pair HMM diagram and state probabilities, with γ the probability of transitioning to a gap state, ϵ the probability of staying in a gap state, and κ the probability of the Markov chain terminating. We assume $1 - 2\gamma - \kappa \geq 0$ and $1 - \epsilon - \kappa \geq 0$. When in a match state, the pair HMM emits letters b and b' in the x and y sequences with probability $\psi_{b,b'}$; otherwise, in gap states, the probability of letter b in the non-gapped sequence is π_b.

Define the MuE transition matrix and termination probability vector as

$$a_{k,k'}^{(t)} := \begin{cases} \frac{1-2\gamma-\kappa}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1=m'\leq M \text{ and } g=g'=0 \\ \frac{\gamma\epsilon^{m'-m-2}(1-\epsilon-\kappa)}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1<m'\leq M \text{ and } g=g'=0 \\ \frac{\gamma}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m+1=m'\leq M \text{ and } g=0 \text{ and } g'=1 \\ \frac{\gamma}{\gamma+\kappa} & \text{if } m+1=m'=M+1 \text{ and } g=0 \text{ and } g'=1 \\ \frac{1-\epsilon-\kappa}{1-\kappa} & \text{if } m=m'\leq M \text{ and } g=1 \text{ and } g'=0 \\ \frac{\epsilon}{1-\kappa} & \text{if } m=m'\leq M \text{ and } g=g'=1 \\ \frac{\epsilon}{\epsilon+\kappa} & \text{if } m=m'=M+1 \text{ and } g=g'=1 \\ 0 & otherwise \end{cases} \tag{A.12}$$

$$t_k^{(t)} := \begin{cases} \frac{\gamma\epsilon^{M-m-1}\kappa}{1-(\gamma\epsilon^{M-m-1}(1-\kappa)+\kappa+\gamma\kappa\frac{1-\epsilon^{M-m-1}}{1-\epsilon})} & \text{if } m<M \text{ and } g=0 \\ \frac{\kappa}{\gamma+\kappa} & \text{if } m=M \text{ and } g=0 \\ \frac{\kappa}{\epsilon+\kappa} & \text{if } m=M+1 \text{ and } g=1 \\ 0 & otherwise \end{cases} \tag{A.13}$$

The initial transition vector is defined by $a_k^{(0)} := a_{0,k}^{(t)}$ and initial termination probability is $t^{(0)} := t_0^{(t)}$. Define the substitution matrix

$$\ell_{b,b'} := \frac{\psi_{b,b'}}{\pi_b} \tag{A.14}$$

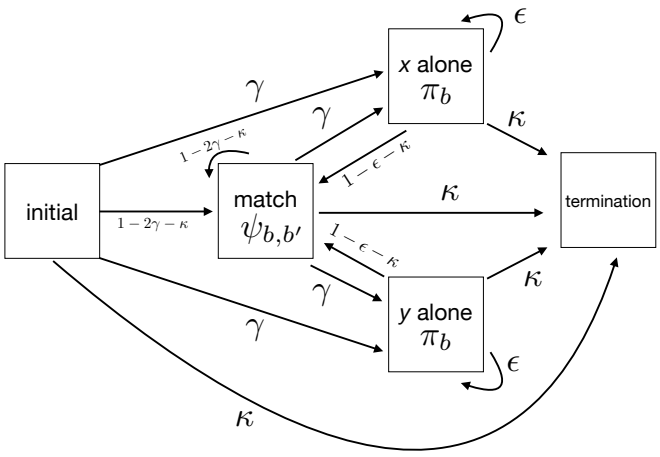


Figure A.4: Pair HMM state diagram.

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

$$c_m := (\ell^{-1})^\top \cdot \pi \tag{A.15}$$

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

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

Proof We will show that the joint probability of W and Y under the MuE model is identical to the joint probability of the corresponding alignment and Y under the pair HMM, conditional on X. We start by enumerating all possible transitions between states of the MuE Markov chain

and computing their probability under the pair HMM model without conditioning on X. Define $\omega_j^x := \mathbb{I}(\mathcal{A}_j^{(x)} \in \mathcal{B})$ and ω^y likewise. We use ω^x, ω^y notation to represent possible alignments, with the symbol "|" placed to the right of the residue we are transitioning *from*.

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

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

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

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

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

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

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

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

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

```
x: 1 | $

y: 1 | $
```

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

```
x: 0 | 1

y: 1 | 1
```

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

```
x: 0 | 0

y: 1 | 1
```

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

```
x: 0 | $

y: 1 | $
```

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

```
x: | 1

y: | 1
```

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

```
x: | 1 ... 1 1

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

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

```
x: | 0

y: | 1
```

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

```
x: | 1 ... 1 $

y: | 0 ... 0 $
```

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

```
x: | $

y: | $
```

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

1. Starting from a state $(m < M, 0)$, the probability of the pair HMM generating an invalid alignment that is too long (rather than transitioning to a valid MuE state) is $\gamma\epsilon^{M-m-1}(1 -$

$\epsilon - \kappa) + \gamma\epsilon^{M-m} = \gamma\epsilon^{M-m-1}(1-\kappa)$. The first term is from alignments that use a match state instead of terminating.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned}
&(1 - 2\gamma - \kappa) + \gamma \sum_{m'=m+2}^{M} \epsilon^{m'-m-2}(1 - \epsilon - \kappa) + \gamma + \gamma\epsilon^{M-m-1}\kappa + \gamma\epsilon^{M-m-1}(1 - \kappa) \\
&\quad + (\kappa + \gamma\kappa\frac{1 - \epsilon^{M-m-1}}{1 - \epsilon}) \\
&= 1 - \gamma + \gamma(1 - \epsilon - \kappa)\frac{1 - \epsilon^{M-m-1}}{1 - \epsilon} + \gamma\epsilon^{M-m-1} + \gamma\kappa\frac{1 - \epsilon^{M-m-1}}{1 - \epsilon} \\
&= 1 - \gamma + \gamma(1 - \epsilon^{M-m-1}) + \gamma\epsilon^{M-m-1} \\
&= 1
\end{aligned} \tag{A.16}$$

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

$$\gamma + \kappa + (1 - \gamma - \kappa) = 1 \tag{A.17}$$

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

$$(1 - \epsilon - \kappa) + \epsilon + \kappa = 1 \tag{A.18}$$

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

$$\kappa + \epsilon + (1 - \epsilon - \kappa) = 1 \tag{A.19}$$

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

$$\begin{aligned}
&(1 - 2\gamma - \kappa) + \sum_{m=2}^{M} \gamma\epsilon^{m-2}(1 - \epsilon - \kappa) + \gamma + \gamma\epsilon^{M-1}\kappa + \gamma\epsilon^{M-1}(1 - \kappa) \\
&\quad + (\kappa + \gamma\kappa\frac{1 - \epsilon^{M-1}}{1 - \epsilon}) \\
&\quad = 1 - \gamma + \gamma(1 - \epsilon - \kappa)\frac{1 - \epsilon^{M-1}}{1 - \epsilon} + \gamma\kappa\epsilon^{M-1} + \gamma\epsilon^{M-1}(1 - \kappa) + \gamma\kappa\frac{1 - \epsilon^{M-1}}{1 - \epsilon} \\
&\quad = 1 - \gamma + \gamma(1 - \epsilon^{M-1}) - \gamma\kappa\frac{1 - \epsilon^{M-1}}{1 - \epsilon} + \gamma\epsilon^{M-1} + \gamma\kappa\frac{1 - \epsilon^{M-1}}{1 - \epsilon} \\
&\quad = 1
\end{aligned} \tag{A.20}$$

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

$$\gamma + \kappa + (1 - \gamma - \kappa) = 1 \tag{A.21}$$

Consolidating transition probabilities and conditioning on the length of X yields the transition matrix Equation A.12.

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,

$$Y_l \sim \text{Categorical}(\sum_{b''} X_{m,b''} \ell_{b''}) = \text{Categorical}(\frac{\psi_b}{\pi_b}). \tag{A.22}$$

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,

$$Y_l \sim \text{Categorical}((\pi^\top \cdot \ell^{-1} \cdot \ell)^\top) = \text{Categorical}(\pi). \tag{A.23}$$

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.

□

Profile HMM

The profile HMM (pHMM) is a widely used model for defining protein sequence families, inferring multiple sequence alignments, and performing database searches[67].

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

$$a_{k,k'}^{(t)} := \begin{cases} (1-r_{m+1-g,g})(1-u_{m+1-g,g}) \\ \quad \text{if } m+1-g = m' \text{ and } g' = 0 \\ (1-r_{m+1-g,g})u_{m+1-g,g}(\prod_{m''=m+2-g}^{m'-1}[(1-r_{m'',2})u_{m'',2}])(1-r_{m',2})(1-u_{m',2}) \\ \quad \text{if } m+1-g < m' \text{ and } g' = 0 \\ r_{m+1-g,g} \\ \quad \text{if } m+1-g = m' \text{ and } g' = 1 \\ (1-r_{m+1-g,g})u_{m+1-g,g}(\prod_{m''=m+2-g}^{m'-1}[(1-r_{m'',2})u_{m'',2}])r_{m',2} \\ \quad \text{if } m+1-g < m' \text{ and } g' = 1 \\ 0 \quad \text{otherwise} \end{cases} \tag{A.24}$$

$$t_k^{(t)} := \begin{cases} 1 - r_{M+1,g} \\ \quad \text{if } m - g = M \\ (1 - r_{m+1-g,g}) u_{m+1-g,g} (\prod_{m''=m+2-g}^{M} [(1 - r_{m'',2}) u_{m'',2}]) (1 - r_{M+1,2}) \\ \quad \text{if } m - g < M \end{cases} \tag{A.25}$$

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

$$\ell_{b,b'} := \delta_{b,b'} \tag{A.26}$$

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

With these definitions the profile HMM can be written as $Y \sim \text{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$. Figure A.5 illustrates samples from the pHMM. Intuitively, r controls insertion probabilities and u controls deletion probabilities; when $r_{m,j} = 0$ and $u_{m,j} = 0$ for all m and j, we recover the no-mutation limit of the MuE.

Proof This result follows from the relabeling of the profile HMM Markov state architecture with the (m, g) notation (Figure A.6). So-called "delete states" in profile HMMs do not generate observations Y_l. To compute the probability of transitioning between two observable states (m, g) and (m', g'), we compute the probability of (1) direct paths between the two states and (2) all possi-

x TACGC

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

Figure A.5: Samples from the profile HMM. The regressor sequence $X_{1,\ldots,M}$ is set to TACGC, and we set $r_{m,j=0} = r_{m,j=1} = r_{m,j=2}$ and $u_{m,j=0} = u_{m,j=1} = u_{m,j=2}$ for all m.

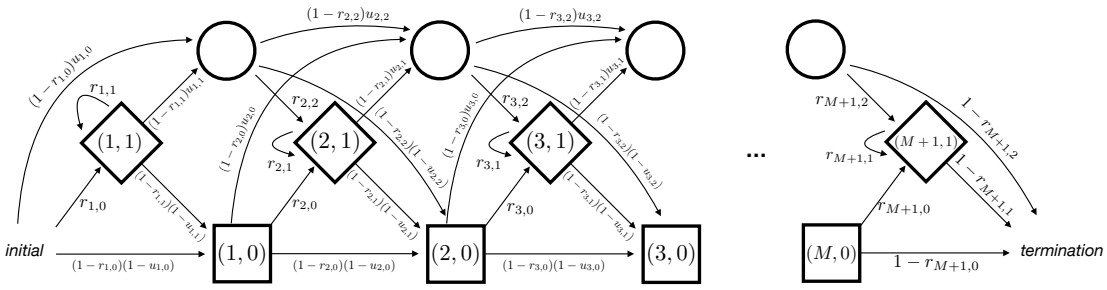


Figure A.6: Profile HMM state architecture. The conventional profile HMM state architecture labeled with MuE states, using (m, g) notation. Squares indicate "match states", diamonds indicate "insert states", and circles indicate "delete states".

ble paths between the two states that go only through deletion states. This yields Equation A.24.

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

□

Needleman-Wunsch

The Needleman-Wunsch (NW) algorithm is a classic non-probabilistic alignment method [187].

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

We define the MuE transition matrix and termination probabilities

$$a^{(t)}_{k,k'} := \begin{cases} \frac{1-u}{1+u} u^{m'-m-1+g} & \text{if } m-g < m' < M+1 \text{ and } g' = 0 \\ \frac{1-u}{1+u} u^{m'-m+g} & \text{if } m-g < m' \leq M+1 \text{ and } g' = 1 \\ 0 & otherwise \end{cases} \tag{A.27}$$

$$t^{(t)}_k := \frac{1+u^2}{1+u} u^{M-m+g} \tag{A.28}$$

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

$$\ell_{b,b'} := \frac{e^{S_{b,b'}}}{B}. \tag{A.29}$$

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

$$c_m := (\ell^{-1})^\top \cdot (1/B, ..., 1/B)^\top \tag{A.30}$$

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

Under the MuE model $Y \sim \text{MuE}(X, c, \ell, a^{(0)}, a^{(t)})$, the maximum *a posteriori* estimator of the alignment variable w given X and Y corresponds to the Needleman-Wunsch pairwise alignment

between X and Y. Note that in the limit $G \to -\infty$ and $S_{b,b'} \to -\infty$ for all $b' \neq b$, we recover the no-mutation limit of the MuE distribution.

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

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

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

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

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

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

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

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

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

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

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

```
x: 1 | 1 ... 1 $

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

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

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

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

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

$$\Delta(l', m, g, m', g') := \begin{cases} (m' - m - 1 + g)G & \\ + \sum_{b,b'} x_{m',b} S_{b,b'} y_{l',b'} & \text{if } m - g < m' < M \text{ and } g' = 0 \\ (m' - m + g)G & \text{if } m - g < m' \leq M \text{ and } g' = 1 \\ -\infty & otherwise \end{cases} \tag{A.31}$$

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

$$\arg\max_{\vec{m},\vec{g}} \sum_{l=1}^{L} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) + (M - m_L + g_L)G \tag{A.32}$$

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

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

$$\arg\max_w \log p(y, w|x, c, a, \ell) = \arg\max_w \left[\log p(\textit{term.}|w_L) + \sum_{l=2}^{L} \log p(y_l, w_l|w_{l-1}) + \log p(y_1, w_1) \right] \tag{A.33}$$

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

$$p(y_l, w_l|w_{l-1}) = \begin{cases} \frac{1-u}{1+u} u^{m_l - m_{l-1} - 1 + g_{l-1}} \frac{1}{B} \exp\left(\sum_{b,b'} x_{m_l,b} S_{b,b'} y_{l,b'} \right) & \\ \qquad \text{if } m_{l-1} - g_{l-1} < m_l < M+1 \text{ and } g_l = 0 & \\ \frac{1-u}{1+u} u^{m_l - m_{l-1} + g_{l-1}} \frac{1}{B} & \\ \qquad \text{if } m_{l-1} - g_{l-1} < m_l \leq M+1 \text{ and } g_l = 1 & \\ 0 \qquad otherwise & \end{cases} \tag{A.34}$$

$$p(\textit{term.}|w_L) = \frac{1+u^2}{1+u} u^{M - m_L + g_L} \tag{A.35}$$

$$p(y_1, w_1) = \begin{cases} \frac{1-u}{1+u} u^{m_1 - 1} \frac{1}{B} \exp(\sum_{b,b'} x_{m_1,b} S_{b,b'} y_{1,b'}) & \text{if } m_1 < M+1 \text{ and } g_1 = 0 \\ \frac{1-u}{1+u} u^{m_1} \frac{1}{B} & \text{if } m_1 \leq M+1 \text{ and } g_1 = 1 \\ 0 & otherwise \end{cases} \tag{A.36}$$

$$p(\textit{term.}|\textit{init.}) = \frac{1+u^2}{1+u} u^M \tag{A.37}$$

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

$$\begin{aligned} \arg\max_w \log p(y, w|x) &= \arg\max_{\vec{m},\vec{g}} \Big[L \log(\frac{1-u}{1+u}\frac{1}{B}) + \log(\frac{1+u^2}{1+u}) + \sum_{l=1}^{L} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) \\ &\qquad + (M - m_L + g_L)G \Big] \\ &= \arg\max_{\vec{m},\vec{g}} \Big[\sum_{l=1}^{L} \Delta(l, m_{l-1}, g_{l-1}, m_l, g_l) + (M - m_L + g_L)G \Big] \end{aligned} \tag{A.38}$$

where again $m_0 = 0$ and $g_0 = 0$. This objective function is identical to the NW objective function (Equation A.32), so the maximum *a posteriori* estimator of w in the MuE distribution corresponds to the Needleman-Wunsch pairwise alignment of X and Y.

We can confirm that the transition probabilities of the MuE distribution are normalized by con-

sidering transitions from state (m, g):

$$\begin{aligned}
&\frac{1-u}{1+u}\sum_{m'=m-g+1}^{M} u^{m'-m-1+g} + \frac{1-u}{1+u}\sum_{m'=m-g+1}^{M+1} u^{m'-m+g} + \frac{1+u^2}{1+u}u^{M-m+g} \\
&\quad = \frac{1-u}{1+u}\Big[\sum_{m''=0}^{M-m-1+g} u^{m''} + u\sum_{m''=0}^{M-m+g} u^{m''}\Big] + \frac{1+u^2}{1+u}u^{M-m+g} \\
&\quad = \frac{1}{1+u}[1 - u^{M-m+g} + u - u^{M-m+g+2}] + \frac{1+u^2}{1+u}u^{M-m+g} \\
&\quad = 1 - \frac{1+u^2}{1+u}u^{M-m+g} + \frac{1+u^2}{1+u}u^{M-m+g} \\
&\quad = 1.
\end{aligned} \tag{A.39}$$

□

A.2.3 Inferring multiple sequence alignments

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

Definition A.2.2 (Multiple sequence alignment)**.** *Let $Y_1, \ldots, Y_N$ be sequences with lengths $L_1, \ldots, L_N$. A multiple sequence alignment $Y_{\text{MSA}} \in (\mathcal{B} \cup \{-\})^J$ has rows $Y_{\text{MSA},1}, \ldots, Y_{\text{MSA},N}$ each consisting of the letters of Y_i, in order, interspersed with gap symbols. The alignment Y_{MSA} must satisfy the condition that for every $j \in \{1, \ldots, J\}$, there exists some $i \in \{1, \ldots, N\}$ such that $Y_{\text{MSA},i,j} \in \mathcal{B}$.*

Consider models of the form of Equation 2, and let W_i be the latent alignment variable associated with sequence Y_i, i.e. $W_{i,1}, \ldots, W_{i,L_i}$ is the path through the latent state space that generated Y_i with length L_i. Algorithm 2 constructs a multiple sequence alignment of the dataset $Y_1, \ldots, Y_N$

Algorithm 2 Multiple sequence alignment construction

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

given $W_1, \ldots, W_N$, placing $Y_{i,l}$ that are generated from the same state $(m, 0)$ (corresponding to a particular position in the "ancestral" sequence X_i) in the same column. Note in the case of multiple sequence alignments, as opposed to pairwise alignments, there is no longer a unique alignment given W, since X is not observed. The Algorithm 2 construction is chosen to match a standard construction used for the profile HMM (see Durbin et al.[67], Chapter 6.5), using the fact that the profile HMM is a special case of Equation 2 with $p_\theta(v) = \delta_{v_0}(v)$, where $\delta_{v_0}(v)$ is the Dirac delta function at v_0. In MuE observation models we can apply the same algorithm as for pHMMs, placing $Y_{i,l}$ that are generated from the same state $(m, 0)$ in the same column.

A.2.4 Proof of Proposition 4.5

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

$$m_{l+1} > m_l - g_l, \tag{A.40}$$

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

A.2.5 Vogel et al. natural language translation

The Vogel et al.[276] translation model takes the same general form as a MuE distribution, with X a sentence in one language and Y a sentence in another language (encoded as sequences of words). In

particular, with states k indexed by tuples (m, g), the transition matrix takes the form

$$a^{(t)}_{k,k'} := \begin{cases} \frac{r_{M+m'-m}}{\sum_{m''=1}^{M} r_{M+m''-m}} & \text{if } g = g' = 0 \text{ and } m, m' \leq M \\ 0 & otherwise \end{cases} \tag{A.41}$$

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

A.3 Models

In this section we provide a detailed description of the models evaluated in the main text. We parameterized the transition matrix $a^{(t)}$ in terms of r and u following Equation A.24 (the profile HMM parameterization). We also considered a simplified variation on Equation A.24 where we enforce the constraint $u_{m,0} = u_{m,1} = u_{m,2}$ and likewise $r_{m,0} = r_{m,1} = r_{m,2}$ for all m. We enforced (in both cases) the constraint $u_{M,j} = 0$ for $j \in \{0, 1, 2\}$ (termination has probability zero); rather than assign a termination state we assume the length of the sequence Y_i, that is L_i, is independent of W_i. Since the probability of L_i does not contribute to the per residue perplexity performance metric (Section A.5) we do not use an explicit model for L_i. The initial transition vector followed the same form as the transition matrix, i.e. $a^{(0)}_k = a^{(t)}_{0,k}$.

Note that in our experiments we go slightly beyond the vanilla MuE observation model presented in the main text (Equation 2), and allow the insertion sequence c to also depend on p_θ.

A.3.1 Profile HMM

The profile HMM is

$$Y_i \sim \text{MuE}(x, c, \ell = I_B, a^{(0)}(r, u), a^{(t)}(r, u)) \tag{A.42}$$

where $a^{(0)}(r, u)$ and $a^{(t)}(r, u)$ depend deterministically on the parameters r and u according to Equation A.24, $D = B$, and I_B is the $B \times B$ identity matrix.

A.3.2 RegressMuE

The RegressMuE model uses a linear regression model as the MuE observation's continuous-space vector model. Let $H_{i,1}, ..., H_{i,T}$ be covariates associated with sequence Y_i. Let $\beta_0^{(x)}, ..., \beta_T^{(x)} \in \mathbb{R}^{M \times D}$ 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

$$\begin{aligned} V_i^{(x)} &= \beta_0^{(x)} + \sum_{t=1}^{T} H_{i,t}\beta_t^{(x)} \\ V_i^{(c)} &= \beta_0^{(c)} + \sum_{t=1}^{T} H_{i,t}\beta_t^{(c)} \\ Y_i &\sim \text{MuE}(X_i = \text{softmax}(V_i^{(x)}), C_i = \text{softmax}(V_i^{(c)}), \ell, a^{(0)}(r, u), a^{(t)}(r, u)). \end{aligned} \tag{A.43}$$

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

A.3.3 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,

$$\begin{aligned} Z_{i,t} &\sim \text{Normal}(0, 1) \\ V_i^{(x)} &= \beta_0^{(x)} + \sum_{t=1}^{T} Z_{i,t}\beta_t^{(x)} \\ V_i^{(c)} &= \beta_0^{(c)} + \sum_{t=1}^{T} Z_{i,t}\beta_t^{(c)} \\ Y_i &\sim \text{MuE}(X_i = \text{softmax}(V_i^{(x)}), C_i = \text{softmax}(V_i^{(c)}), \ell, a^{(0)}(r,u), a^{(t)}(r,u)) \end{aligned} \tag{A.44}$$

A.3.4 ICAMuE

The ICAMuE model the same as the FactorMuE model, except that it uses a Laplace prior instead of a Normal prior on the local latent variable (Murphy [185], Chapter 12.6).

$$\begin{aligned} Z_{i,t} &\sim \text{Laplace}(0, 1) \\ V_i^{(x)} &= \beta_0^{(x)} + \sum_{t=1}^{T} Z_{i,t}\beta_t^{(x)} \\ V_i^{(c)} &= \beta_0^{(c)} + \sum_{t=1}^{T} Z_{i,t}\beta_t^{(c)} \\ Y_i &\sim \text{MuE}(X_i = \text{softmax}(V_i^{(x)}), C_i = \text{softmax}(V_i^{(c)}), \ell, a^{(0)}(r,u), a^{(t)}(r,u)) \end{aligned} \tag{A.45}$$

A.3.5 NeuralMuE

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

$$
\begin{aligned}
V_{i,\Gamma+1} &= \beta_{\Gamma+1,0} + \sum_{t=1}^{T_{\Gamma+1}} H_{i,t}\beta_{\Gamma+1,t} \\
V_{i,\Gamma} &= \beta_{\Gamma,0} + \sum_{t=1}^{T_{\Gamma}} \text{relu}(V_{i,\Gamma+1,t})\beta_{\Gamma,t} \\
&\dots \\
V_{i,1}^{(x)} &= \beta_{1,0}^{(x)} + \sum_{t=1}^{T_1} \text{relu}(V_{i,2,t})\beta_{1,t}^{(x)} \\
V_{i,1}^{(c)} &= \beta_{1,0}^{(c)} + \sum_{t=1}^{T_1} \text{relu}(V_{i,2,t})\beta_{1,t}^{(c)} \\
Y_i &\sim \text{MuE}(X_i = \text{softmax}(V_{i,1}^{(x)}), C_i = \text{softmax}(V_{i,1}^{(c)}), \ell, a^{(0)}(r,u), a^{(t)}(r,u))
\end{aligned}
\tag{A.46}
$$

A.3.6 LatentNeuralMuE

The LatentNeuralMuE model uses a neural network latent variable model as the MuE observation's continuous-space vector model. It is the latent covariate version of the NeuralMuE, where instead

of observing H we draw a latent variable Z from a standard normal prior.

$$
\begin{aligned}
Z_{i,t} &\sim \text{Normal}(0, 1) \\
V_{i,\Gamma+1} &= \beta_{\Gamma+1,0} + \sum_{t=1}^{T_{\Gamma+1}} Z_{i,t}\beta_{\Gamma+1,t} \\
V_{i,\Gamma} &= \beta_{\Gamma,0} + \sum_{t=1}^{T_{\Gamma}} \text{relu}(V_{i,\Gamma+1,t})\beta_{\Gamma,t} \\
&\ldots \\
V_{i,1}^{(x)} &= \beta_{1,0}^{(x)} + \sum_{t=1}^{T_1} \text{relu}(V_{i,2,t})\beta_{1,t}^{(x)} \\
V_{i,1}^{(c)} &= \beta_{1,0}^{(c)} + \sum_{t=1}^{T_1} \text{relu}(V_{i,2,t})\beta_{1,t}^{(c)} \\
Y_i &\sim \text{MuE}(X_i = \text{softmax}(V_{i,1}^{(x)}), C_i = \text{softmax}(V_{i,1}^{(c)}), \ell, a^{(0)}(r, u), a^{(t)}(r, u))
\end{aligned}
\tag{A.47}
$$

A.3.7 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, $\ell_d \in \Delta_B$. To enable easy gradient-based optimization and stochastic variational inference[147], we transform an unconstrained parameter $\tilde{\ell} \in \mathbb{R}^{D \times B}$ with a Gaussian prior to the simplex,

$$
\begin{aligned}
\tilde{\ell}_{d,b} &\sim \text{Normal}(0, 1) \\
\ell_d &= \text{softmax}(\tilde{\ell}_d).
\end{aligned}
\tag{A.48}
$$

The variables $r_{m,j}$ and $u_{m,j}$ are constrained to $[0, 1]$ for all m and j. This corresponds to the first dimension of a simplex Δ_2, and so we apply the same approach,

$$\begin{aligned} \tilde{r}_{m,j,\vartheta} &\sim \text{Normal}(\mu_{\vartheta}^{(r)}, 1) \text{ for } \vartheta \in \{1, 2\} \\ r_{m,j} &= \frac{\exp(\tilde{r}_{m,j,2})}{\exp(\tilde{r}_{m,j,1}) + \exp(\tilde{r}_{m,j,2})} \end{aligned} \tag{A.49}$$

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

In the case of the ICAMuE model we found that training improved with an annealing strategy: we multiplied each coefficient matrix β by a scalar inverse-temperature parameter ξ, drawn according to $\tilde{\xi} \sim \text{Normal}(100, 1)$ and $\xi = \text{softplus}(\tilde{\xi})$ where $\text{softplus} = \log(1 + \exp(\cdot))$; the variational approximation to ξ (see below) was initialized such that $q(\tilde{\xi})$ had mean 0. Note that this annealing approach does not change the expressivity of the model, only the prior and training dynamics. Details can be found in the supplementary code (see Section A.4.2).

A.4 Inference

A.4.1 Stochastic variational inference

Variational inference approximates the posterior distribution $p(\theta|Y_{1:N})$ of a given probabilistic model using a tractable family of distributions $q_\eta(\theta|Y_{1:N})$ parameterized by η[27]. To form this approximation, variational inference minimizes the Kullback-Leibler (KL) divergence between the two

distributions,

$$\eta_0 := \arg\min_{\eta} \textsc{kl}(q_\eta(\theta|Y_{1:N})||p(\theta|Y_{1:N})) \tag{A.50}$$

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

$$\eta_0 = \arg\max_{\eta} \mathbb{E}_{q_\eta(\theta|Y_{1:N})}[\log p(Y_{1:N}, \theta)] - \mathbb{E}_{q_\eta(\theta|Y_{1:N})}[\log q_\eta(\theta|Y_{1:N})] = \arg\max_{\eta} \textsc{elbo}(\eta) \tag{A.51}$$

We employ mean-field variational inference for MuE observation models. We use a diagonal Gaussian distribution, with unknown mean and standard deviation, for the variational distribution over the global parameters $\tilde{r}$, $\tilde{u}$, $\tilde{\ell}$, $\tilde{\xi}$ and β. For the local variable z in the FactorMuE and LatentNeuralMuE, we amortize inference using an inference network (also known as an encoder network)[139,213]. In particular, we set

$$q_{\eta_z}(z_{1:N}|Y_{1:N}) = \prod_{i=1}^{N} q_{\eta_z}(z_i|Y_i) = \prod_{i=1}^{N} \mathcal{N}(z_i|f^{(\mu)}(Y_i;\eta_z), f^{(\sigma)}(Y_i;\eta_z)) \tag{A.52}$$

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,

$$
\begin{aligned}
y_{i,l}^{(q)} &= \mathbb{E}_{Y' \sim \text{MuE}(Y_i, c^{(q)}, \ell^{(q)}, a^{(0)}(r^{(q)}, u^{(q)}), a^{(t)}(r^{(q)}, u^{(q)}))}[Y'_l] \\
v_{i,\Gamma^{(q)}+1}^{(q)} &= \beta_{\Gamma^{(q)}+1,0}^{(q)} + \sum_{l=1}^{L^{(q)}} \sum_{b=1}^{B} y_{i,l,b}^{(q)} \beta_{\Gamma^{(q)}+1,l,b}^{(q)} \\
v_{i,\Gamma^{(q)}}^{(q)} &= \beta_{\Gamma^{(q)},0}^{(q)} + \sum_{t=1}^{T_{\Gamma^{(q)}}} \text{relu}(v_{i,\Gamma^{(q)}+1,t}^{(q)}) \beta_{\Gamma^{(q)},t}^{(q)} \\
&\dots \\
f^{(\mu)} &= \beta_{1,0}^{(q,\mu)} + \sum_{t=1}^{T_1} \text{relu}(v_{i,2,t}^{(q)}) \beta_{1,t}^{(q,\mu)} \\
f^{(\sigma)} &= |\beta_{1,0}^{(q,\sigma)} + \sum_{t=1}^{T_1} \text{relu}(v_{i,2,t}^{(q)}) \beta_{1,t}^{(q,\sigma)}|.
\end{aligned}
\tag{A.53}
$$

where we have introduced the variational parameters $(\beta^{(q)}, c^{(q)}, r^{(q)}, u^{(q)}, \ell^{(q)}) =: \eta_z$. The first layer of the encoder employs the MuE distribution and computes the expected value of mutants of Y_i, at positions $l \in \{1, ..., L^{(q)}\}$; this expected value is a differentiable function of the MuE parameters, and can be tractably computed using the forward algorithm. We use the same parameterization of the MuE distribution as in the models (Section A.3), but fix $r_{1,0}^{(q)} = r_{1,1}^{(q)} = r_{1,2}^{(q)} = r_{2,0}^{(q)} = ... = r_{M,2}^{(q)}$ and $u_{1,0}^{(q)} = u_{1,1}^{(q)} = u_{1,2}^{(q)} = u_{2,0}^{(q)} = ... = u_{M-1,2}^{(q)}$ and $c_1^{(q)} = c_2^{(q)} = ... = c_M^{(q)}$. Intuitively, the MuE encoding serves to "smear out" the one-hot encoded sequence Y_i according to learnable insertion, deletion 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 varia-

tional inference, approximating the gradient at each update step using a minibatch of data[208]. Let $\phi := (\beta, r, u, \ell)$ be the global parameters of the MuE observation models proposed in Section A.3 and let η_ϕ be the parameters of the associated mean-field variational distribution. Then the gradient of the ELBO is

$$\begin{aligned}\nabla_\eta \text{ELBO}(\eta) &= \sum_{i=1}^{N}\Big(\nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)q_{\eta_z}(z_i|Y_i)}[\log p(Y_i|Z_i,\phi)] + \nabla_\eta \mathbb{E}_{q_{\eta_z}(z_i|Y_i)}\Big[\log \frac{p(Z_i)}{q_{\eta_z}(Z_i|Y_i)}\Big]\Big) \\ &\quad + \nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)}\Big[\log \frac{p(\phi)}{q_{\eta_\phi}(\phi)}\Big] \\ &\approx \frac{N}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\Big(\nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)q_{\eta_z}(z_i|Y_i)}[\log p(Y_i|Z_i,\phi)] + \nabla_\eta \mathbb{E}_{q_{\eta_z}(z_i|Y_i)}\Big[\log \frac{p(Z_i)}{q_{\eta_z}(Z_i|Y_i)}\Big]\Big) \\ &\quad + \nabla_\eta \mathbb{E}_{q_{\eta_\phi}(\phi)}\Big[\log \frac{p(\phi)}{q_{\eta_\phi}(\phi)}\Big]\end{aligned} \tag{A.54}$$

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[147,139,213]. The remaining terms can be computed analytically (see e.g. Kingma & Welling[139], Rezende et al.[213]). Note that this approach relies crucially on the fact that the marginal likelihood of the MuE model, $p_{\text{MuE}}(y|x, c, \ell, a^{(0)}, a^{(t)}) = \sum_w p_{\text{MuE}}(y|w, x, c, \ell, a^{(0)}, a^{(t)})$, is a differentiable function of x, c, a and ℓ. We integrate over all possible values of the Markov chain state variable w using the forward algorithm.

It is useful in some circumstances to reweight the variational objective to reduce the amount of

regularization placed on the local latent variable. In particular, for $\chi \in [0, 1]$, we reweight the ELBO as

$$\begin{aligned} \text{ELBO}_\chi(\eta) = \sum_{i=1}^{N} \Big(\mathbb{E}_{q_{\eta_\phi}(\phi) q_{\eta_z}(z_i|Y_i)} [\log p(Y_i|Z_i, \phi)] + \chi \mathbb{E}_{q_{\eta_z}(z_i|Y_i)} \Big[\log \frac{p(Z_i)}{q_{\eta_z}(Z_i|Y_i)} \Big] \Big) \\ + \mathbb{E}_{q_{\eta_\phi}(\phi)} \Big[\log \frac{p(\phi)}{q_{\eta_\phi}(\phi)} \Big]. \end{aligned} \tag{A.55}$$

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 [29]. To avoid posterior collapse and produce informative latent representations, we found it useful in certain cases to anneal χ only up to a low value $\chi_0 << 1$ in which case we are approximating the maximum likelihood estimator of z; this annealing schedule was only used for producing data visualizations, rather than prediction of held out data (Section A.8)[7].

A.4.2 Probabilistic programming

We implemented a MuE distribution in both Pyro [23] and Edward2 [263], probabilistic programming languages that are GPU-enabled and can use a variety of different inference procedures including both stochastic variational inference and MCMC methods. Probabilistic programming systems make it easy to try out different priors and different continuous-space matrix models p_θ; they also make it easy to build joint models of sequences and other types of data.

Documentation for the Pyro implementation can be found at `https://docs.pyro.ai/en/dev/contrib.mue.html`. Example Pyro models can be found at `https://github.com/pyro-ppl/`

pyro/tree/dev/examples/contrib/mue. The Edward2 implementation, along with a brief tutorial, is available at https://github.com/debbiemarkslab/MuE.

A.5 Evaluation

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

$$\Omega := \exp\Big(-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\log p(Y_i|L_i)\Big). \tag{A.56}$$

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

$$\hat{\Omega} := \exp\Big(-\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\frac{1}{L_i}\mathbb{E}_{q(\phi|Y_{\mathcal{D}})}[\log p(Y_i|L_i,\phi)]\Big) \tag{A.57}$$

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[27],

$$\hat{\Omega} \approx \exp\Bigg(-\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\frac{1}{L_i}\Big(\mathbb{E}_{q(\phi|Y_{\mathcal{D}})q(z_i|Y_i)}[\log p(Y_i|L_i,Z_i,\phi)] + \mathbb{E}_{q(z_i|Y_i)}\Big[\log\frac{p(Z_i)}{q(Z_i|Y_i)}\Big]\Big)\Bigg). \tag{A.58}$$

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,

$$\log \text{BF}_{1,2} := \sum_{i \in \mathcal{T}} \mathbb{E}_{q_2(\phi|Y_{\mathcal{D}})}[\log p_2(Y_i|L_i, \phi)] - \sum_{i \in \mathcal{T}} \mathbb{E}_{q_1(\phi|Y_{\mathcal{D}})}[\log p_1(Y_i|L_i, \phi)] \tag{A.59}$$

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 A.58. The Bayes factor provides a measurement of the total evidence in favor of one model versus another.

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

$$\Omega_0 := \exp\big(-\mathbb{E}_{p(y)}\big[\frac{1}{L}\log p(Y|L)\big]\big). \tag{A.60}$$

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

Naive

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

$$\Omega_0 = \exp(-\mathbb{E}[\log(1/20)]) = 20. \tag{A.61}$$

Amino acid frequencies

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

$$\Omega_0 = \exp\left(- \mathbb{E}_{Y \sim \text{Categorical}(f)}[\log(f^\top \cdot Y)]\right) = \exp(-\sum_{b=1}^{20} f_b \log f_b) \approx 17.92 \tag{A.62}$$

where Y is a one-hot encoding[265,84].

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[101]. 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:

$$\begin{aligned}\log \ell_{b,b'} &= \log p(y_{b'} = 1 | x_b = 1) = \log\left(\frac{f_{b,b'}}{f_b}\right) = \log f_{b'} + \log\left(\frac{f_{b,b'}}{f_b f_{b'}}\right) \\ &= \log f_{b'} + \frac{\log(2)}{2}\text{BLOSUM62}_{b,b'}\end{aligned} \tag{A.63}$$

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

$$\Omega_0 = \exp(-\mathbb{E}_{X\sim\text{Categorical}(f)}\left[\mathbb{E}_{Y\sim\text{Categorical}(X\cdot\ell)}[\log(X^\top \cdot \ell \cdot Y)]\right]) \approx 11.00. \tag{A.64}$$

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 [1]. 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

$$\ell_{b,b'} = \begin{cases} \frac{q_{\text{codon}}}{19} & \text{if } b \neq b' \\ 1 - q_{\text{codon}} & \text{if } b = b'. \end{cases} \tag{A.65}$$

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

$$\Omega_0 = \exp\left(\mathbb{E}_{Y \sim \text{Categorical}(x^\top \cdot \ell)}[\log(x^\top \cdot \ell \cdot Y)]\right) \approx 1.024. \tag{A.66}$$

A.6 Predictive performance

A.6.1 Survey

Dihydrofolate reductase (DHFR) is a widely conserved enzyme, serine recombinase (PINE) is used as a tool for genomic engineering, cyclin dependent kinase inhibitor 1B (CDKN1B/p27) is a cell cycle inhibitor, and the human papillomavirus E6 protein (VE6) is an oncogenic viral protein [110,261,254]. Evolutionarily related sequences for each were collected using jackhmmer (v3.1) from the UniRef100 dataset (date 6/2019) [130,69,251]. We used seed sequences with Uniprot identifiers DYR_HUMAN (DHFR dataset), PINE_ECOLI (PINE dataset), CDN1B_HUMAN (CDKN1B dataset), and VE6_HPV16 (VE6 dataset). Note that CDKN1B and VE6 have regions classified as disordered. We set a bitscore threshold of 0.5 bits/residue as in Hopf et al. [110] and ran the jackhmmer search using the API from the EVcouplings package [109]. We included the full envelope of the

profile HMM hit in the final dataset. The CDN1B dataset had 1,055 sequences and the VE6 dataset 1,609 sequences. We found 32,510 and 79,354 hits respectively for the DHFR and PINE datasets, which we randomly subsampled to 10,000 sequences to create the final datasets. Note that the jackhmmer search algorithm uses a profile HMM to find distant homologs, and thus may bias the dataset to look more like samples from a pHMM; we therefore expect the performance gains from using other MuE observation models, as compared to the pHMM, on these datasets to be smaller (more conservative) than the performance gains that might be achieved on alternative datasets assembled using different search methods. The TCR dataset was not assembled using jackhmmer. Instead, we downloaded a public dataset from 10x Genomics of 6,327 TCR sequences found in CD8+ cytotoxic T-cells https://support.10xgenomics.com/single-cell-vdj/datasets/2.2.0/vdj_v1_hs_cd8_t (download file dated July 28, 2018). These were sequenced using single cell sequencing of peripheral blood mononuclear cells obtained from an individual healthy donor. Internal stop codons were removed from the sequence.

We set the latent alphabet size $D = 25$. In each experiment, we set M to be 10% longer than the longest sequence in the dataset. We used $T = 5$ latent space dimensions in the FactorMuE and layer sizes $T_2 = 5, T_1 = 10$ in the LatentNeuralMuE (we found a substantial dropoff in performance when increasing network width or depth). In the recognition network, we set $L^{(q)} = M - 1$. We also used $\Gamma^{(q)} = 0$ (no relu nonlinearities) in the FactorMuE recognition network and $\Gamma^{(q)} = 1$, $T_1 = 10$ in the LatentNeuralMuE recognition network. For the MuE, we used the constraint $u_{m,0} = u_{m,1} = u_{m,2}$ and likewise $r_{m,0} = r_{m,1} = r_{m,2}$ for all m. For the prior on the MuE insertion and deletion parameters we used $\mu^{(r)} = \mu^{(u)} = (100, 1)$ to disfavor indels.

In these particular experiments, models were implemented in PyTorch, with variational inference implemented by hand and without the parallelized forward algorithm (experiments in Sections A.6.2 and A.6.3 were performed second with the Pyro implementation). We optimized the variational approximation using Adam [138] and a minibatch size of 5. The mean of the variational distribution was initialized at the prior mean, while the variance was initialized to a small random value (the absolute value of a sample from a normal distribution with standard deviation 0.01). We used one Monte Carlo sample to estimate the ELBO gradient at each step. For each model and dataset, we evaluated two different learning rates, 0.1 and 0.01, and three different random restarts, selecting among training runs the parameter values that reached the highest ELBO on the training set for making predictions. For models with local latent variables (the FactorMuE and LatentNeuralMuE), we annealed the ELBO reweighting factor χ from 0 to 1 linearly over the first 2 epochs. We trained for 4 epochs total on the DHFR and PINE datasets, and 7 epochs total on the smaller CDKN1B, VE6 and TCR datasets, which was sufficient for convergence in each model. We estimated the heldout perplexity using one independent Monte Carlo sample per batch. Computations were performed on graphics processing units (NVIDIA Tesla M40, K80 and V100 GPUs), with double precision, and we used gradient accumulation to reduce memory usage. Single training runs ranged from ∼30 min. for smaller datasets (CDKN1B and VE6) to ∼2.5 hours for larger datasets (DHFR, PINE and TCR).

A.6.2 Patient immune repertoires

We considered six datasets. "HC 1" consisted of 5,179 BCR sequences from a healthy donor, obtained with single cell sequencing of peripheral blood mononuclear cells, available from 10x Genomics https://support.10xgenomics.com/single-cell-vdj/datasets/3.0.0/vdj_v1_hs_pbmc2_b (download file dated November 15, 2018). The rest of the datasets all were taken from a study of T cell receptors in patients with and without multiple sclerosis during pregnancy[207]. Sequences were translated to amino acids based on the provided nucleotide sequence annotations. The dataset "HC 2" is from a healthy patient, third trimester, CD8+ cells. "HC 3" is from a healthy patient, third trimester, CD4+ cells. "MS 1" is from a patient with MS, before pregnancy, CD8+ cells. "MS 2" is from a patient with MS, second trimester, CD8+ cells. "MS 3" is from a patient with MS, third trimester, CD4+ cells. Each of the datasets from Ramien et al. [207] was uniformly subsampled to 20,000 sequences. Across all datasets, internal stop codons were modeled along with the 20 amino acids (i.e. $B = 21$).

We again set the latent alphabet size to $D = 25$. We set $M = 200$, longer than most sequences in each dataset. We used $T = 5$ latent dimensions in the ICAMuE. In the recognition network we used $\Gamma^{(q)} = 0$ and set $r^{(q)}$, $u^{(q)}$ and $\ell^{(q)}$ to the no-mutation limit (avoiding the need for the forward algorithm, to speed up inference at some cost in flexibility). We did *not* use either the constraint $u_{m,0} = u_{m,1} = u_{m,2}$ or the constraint $r_{m,0} = r_{m,1} = r_{m,2}$ in these experiments. For the prior on the MuE insertion and deletion parameters we used $\mu^{(r)} = \mu^{(u)} = (10, 0)$, for both the ICAMuE and pHMM models. We used the $\tilde{\xi} \sim \text{Normal}(100, 1)$ prior as described in

Section A.3.7 for the ICAMuE model.

Models were implemented in Pyro. We used Pyro's stochastic variational inference method (in particular, `JitTrace_ELBO`, the jit-compiled ELBO), and the parallelized forward algorithm[227]. Optimization was performed with Adam, with a learning rate of 0.01, and a minibatch size of 5. Initialization was performed the same as previously, with the exception that $q(\tilde{\xi})$ was initialized to have mean zero. Pyro's low-variance ELBO gradient estimators enabled more reliable inference, and so we only used one initialization in each experiment (rather than three). For the HC 1 dataset we trained for 10 epochs, annealing χ for the first 4; for the remaining (larger) datasets, we trained for two epochs, annealing for 1. This was sufficient for convergence. We used the same GPU hardware as previously, but did not use gradient accumulation. Training took $\sim$20 min. on the larger datasets (the Pyro implementation offers considerable speedup advantages, thanks in part to the parallelized filtering algorithm).

A.6.3 Disordered proteins

Toth-Petroczy et al.[261] collected datasets of evolutionarily related sequences using jackhmmer on the Uniref and Uniprot databases, starting from regions of human proteins classified as disordered. They developed a (heuristic) alignment uncertainty score to determine whether the MSA provided by jackhmmer was trustworthy enough to apply a Potts model and reach conclusions about epistatic interactions between positions in the MSA. They did not proceed with the Potts model analysis on datasets with a sufficiently high uncertainty score; we examined these datasets in particular (`https://marks.hms.harvard.edu/disorder/proteome`). We focused on moderately sized datasets:

those with more than 3,000 but less than 25,000 sequences, with the disordered segment less than 160 amino acids long. As before, we included the full envelope of the profile HMM hit in the final dataset.

We used the same hyperparameters and training procedure as in Section A.6.2, but set the number of epochs to be the minimum number such that at least 50,000 optimization steps were taken, and the number of epochs of χ annealing to half this number (rounded up).

Detailed results Perplexity on a randomly held out 20% of sequences are shown in Table A.2. In 55 out of the 56 datasets, the relative performance of the pHMM and ICAMuE on the training data accurately reflected their relative performance on the test set, i.e. when the pHMM outperformed the ICAMuE model on the training set it also did so on the test set and vice versa. The ICAMuE seems to offer particular advantages when the pHMM itself has low perplexity: among datasets with pHMM perplexity below 8, we find the ICAMuE performs better in more than half (16 out of 31), while among datasets with pHMM perplexity below 5, the ICAMuE performs better in 5 out of 6.

Table A.2: Heldout perplexity on disordered protein datasets. "Disordered segment" is the region of the protein classified as disordered that was used as a seed in jackhmmer. "Size" is the total number of sequences in the dataset. Rows sorted by pHMM perplexity.

Gene name	Uniprot id	Disordered segment	Size (sequences)	pHMM	ICAMuE
AKAP6	Q13023	293-431	6349	2.88	**1.98**
NSD1	Q96L73	2463-2590	6517	2.93	**2.64**
NFAT5	O94916	633-769	10283	2.94	**1.98**

Continued on next page

Table A.2 – continued from previous page

Gene name	Uniprot id	Disordered segment	Size (sequences)	pHMM	ICAMuE
CIC	Q96RK0	48-207	7511	**3.10**	3.79
S26A8	Q96RN1	847-970	9466	4.14	**2.67**
TADBP	Q13148	261-373	12873	4.78	**2.97**
TET2	Q6N021	1475-1587	22017	**5.11**	5.98
K2022	Q5QGS0	589-707	3719	**5.14**	5.99
YAF2	Q8IY57	53-180	16005	**5.42**	5.98
HDAC5	Q9UQL6	479-631	14275	**5.44**	5.85
MUC19	Q7Z5P9	5890-6021	13491	5.59	**4.84**
RBM27	Q9P2N5	91-247	11685	**5.80**	6.28
DEN1A	Q8TEH3	453-567	6070	**5.84**	6.11
K1683	Q9H0B3	383-502	10098	5.94	**4.39**
FNBP1	Q96RU3	280-432	23781	5.96	**3.82**
TOX	O94900	135-269	9881	**6.02**	6.48
SRPK3	Q9UPE1	238-348	9345	6.06	**5.73**
CAC1G	O43497	470-626	16502	**6.16**	7.07
NGAP	Q9UJF2	803-953	6356	6.39	**4.45**
PS1C1	Q9UIG5	1-126	3434	6.62	**6.13**

Continued on next page

Table A.2 – continued from previous page

Gene name	Uniprot id	Disordered segment	Size (sequences)	pHMM	ICAMuE
GPKOW	Q92917	31-157	5888	6.67	**4.90**
GOG8B	A8MQT2	1-131	3674	**7.17**	8.04
CPXM1	Q96SM3	30-137	3538	**7.28**	11.51
ESX1	Q8N693	34-147	10234	**7.34**	10.44
PPIL4	Q8WUA2	337-492	5897	7.38	**6.06**
TAOK3	Q9H2K8	316-433	8661	**7.38**	8.32
CAAP1	Q9H8G2	197-335	19715	7.54	**5.00**
CCD66	A2RUB6	681-830	9586	**7.66**	9.02
GCC2	Q8IWJ2	1416-1552	7593	7.74	**5.71**
ASXL3	Q9C0F0	107-236	5108	**7.75**	8.42
ARHGF	O94989	273-413	4290	7.97	**7.66**
YJ013	Q6ZQT7	1-158	22994	8.07	**4.32**
PHLB2	Q86SQ0	842-976	22091	**8.34**	10.61
CC168	Q8NDH2	86-232	12240	**8.40**	9.73
41	P11171	690-805	7429	**8.70**	10.05
CEBPA	P49715	161-314	20149	**8.81**	10.18
CP250	Q9BV73	2213-2346	17867	**9.08**	12.34

Continued on next page

Table A.2 – continued from previous page

Gene name	Uniprot id	Disordered segment	Size (sequences)	pHMM	ICAMuE
CHD6	Q8TD26	2312-2457	8843	**9.19**	10.62
ANKH1	Q8IWZ3	2000-2149	15540	**9.22**	10.59
CPLX4	Q7Z7G2	18-128	20000	**9.42**	11.16
WAC	Q9BTA9	198-353	3385	9.58	**9.15**
BAHC1	Q9P281	1357-1482	9092	**9.76**	11.13
GOG6B	A6NDN3	473-580	7947	**9.78**	11.50
NOB1	Q9ULX3	110-221	4659	**9.86**	11.93
DGKH	Q86XP1	581-705	5903	**9.86**	11.41
CASZ1	Q86V15	1589-1735	7943	**9.92**	11.38
POTED	Q86YR6	367-502	15076	**9.95**	11.48
POTEC	B2RU33	367-502	15076	**10.02**	11.43
POTEH	Q6S545	405-545	8777	**10.32**	11.90
ZKSC2	Q63HK3	586-738	18126	**10.32**	14.57
PTRF	Q6NZI2	175-297	18730	**10.35**	14.07
PERQ1	O75420	289-441	10443	**10.44**	12.35
U17L8	P0C7I0	383-530	2759	10.57	**9.99**
LRCH2	Q5VUJ6	491-642	6832	**10.66**	12.60

Continued on next page

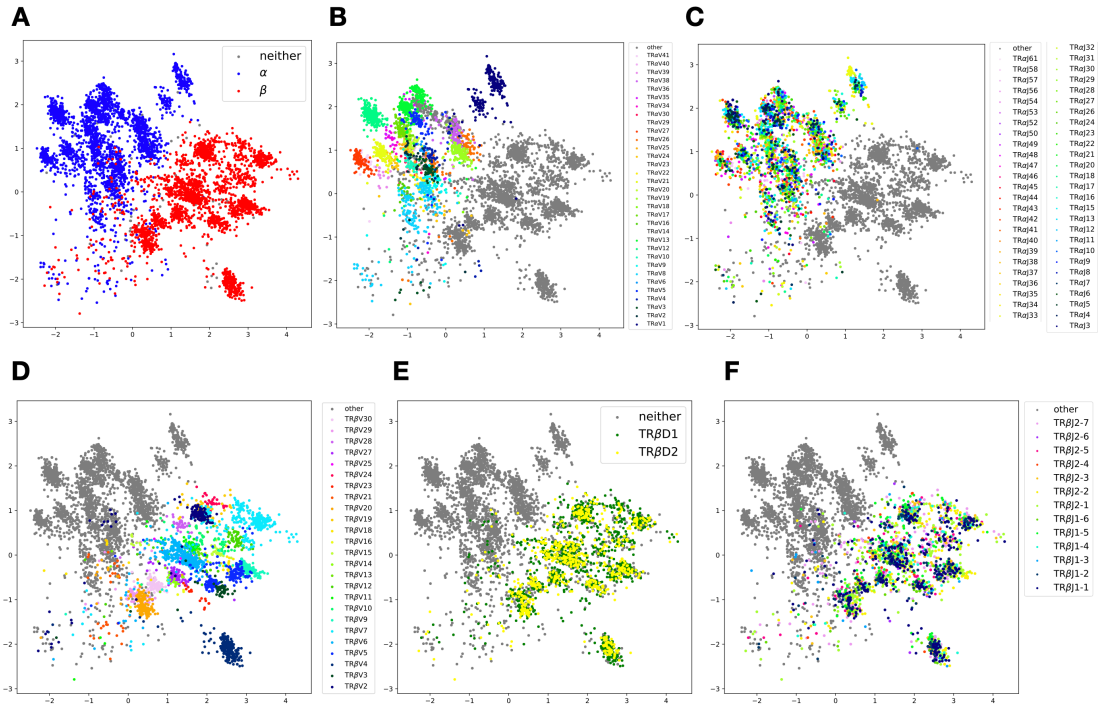


Figure A.7: Latent space representation of human T-cell receptor sequences, colored by supervised annotations. Annotations were provided with the 10x Genomics dataset. (A) C_α versus C_β. (B) α chain V types. (C) α chain J types. (D) β chain V types. (E) β chain D types. (F) β chain J types and subtypes.

Table A.2 – continued from previous page

Gene name	Uniprot id	Disordered segment	Size (sequences)	pHMM	ICAMuE
EMIL2	Q9BXX0	121-259	5666	**11.16**	14.16
LMO7	Q8WWI1	763-901	7893	**11.18**	12.68

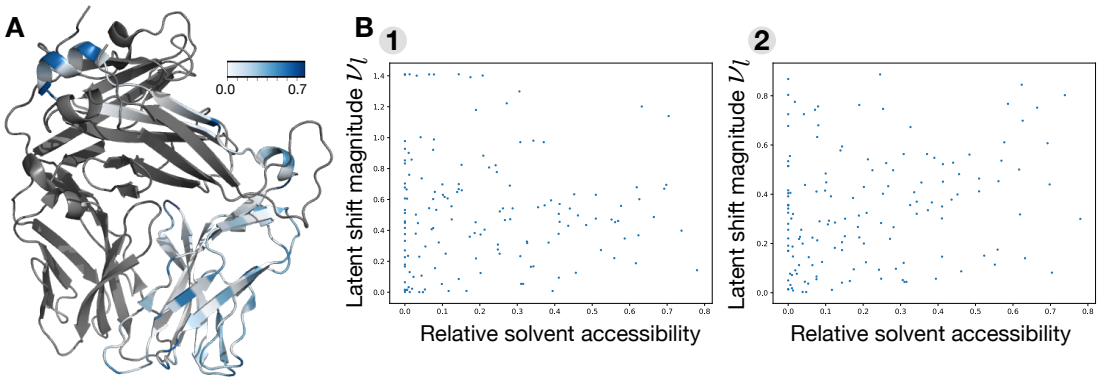


Figure A.8: Comparing MuE observation model features to T-cell receptor relative solvent accessibility. (A) Relative solvent accessibility of TCRβ from the structure PDB:2BNR [38] (the TCRα chain is shown in gray), computed using DSSP [132] and the maximum values in Tien et al. [258] with the Biopython API [42]. (B) Residue relative solvent accessibility versus FactorMuE shift magnitude ν_l along vector 1 and vector 2 from Figure 5D. The correlation between the shift along vector 1 and the accessibility is Spearman $\rho = 0.039, p = 0.64$.

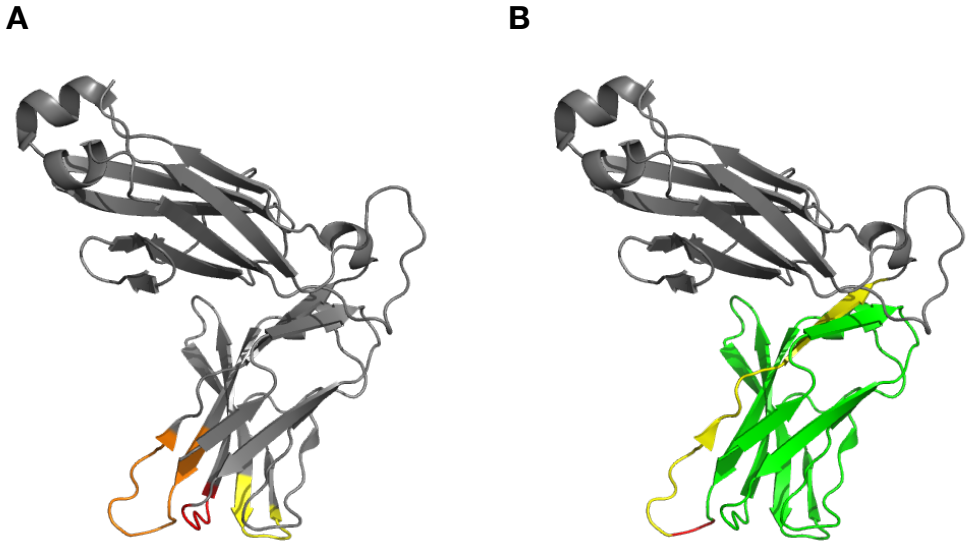


Figure A.9: T-cell receptor structural annotations. (A) CDR segments of PDB:2BNR chain E [38], based on IgBLAST annotations [295] of the nucleotide sequence of 1G4 TCRβ obtained from Robbins et al. [218], and translated from nucleotides into the corresponding positions in the amino acid sequence. CDR1 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.

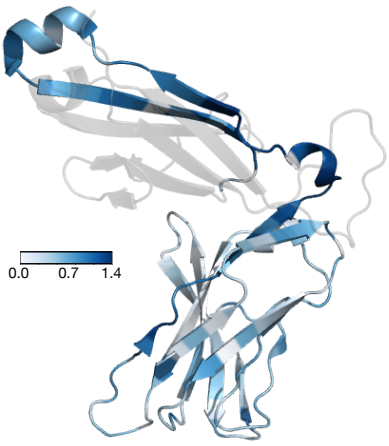


Figure A.10: Shift ν from chain α to chain β sequences learned by the RegressMuE model. ν_l was computed as in Equation 3, using the chain annotation in place of the latent variable z.

A.7 T-Cell receptor analysis

A.7.1 Details

We used the 10x Genomics single-cell TCR sequencing dataset described in Section A.6.1, along with the CellRanger annotations of chain features provided along with the dataset. Annotations of the reference structure PDB:2BNR are based on IgBLAST annotations[295] of the nucleotide sequence of 1G4 TCRβ obtained from Robbins et al.[218], and translated from nucleotides into the corresponding positions in the amino acid sequence (Figure A.9).

To obtain a latent space representation (Figure 5B), we trained the FactorMuE observation model with $T = 2$ latent dimensions, and chose among training runs based on a randomly held out test set (5% of the data). Hyperparameters were otherwise set as in Section A.6.1. The shift ν is estimated using the variational approximation to the posterior of the FactorMuE (using 10 Monte

Carlo samples). $\hat{w}_{\text{ref}}$ is estimated using a single sample from the variational approximation to the posterior and the Viterbi algorithm.

A.7.2 Further results

Along feature vector 2 (Figure 5D) we found weak positive correlation between the magnitude of variation and the relative surface accessibility of each site (Spearman correlation $\rho = 0.20$, $p < 0.02$; Figure A.8). Along feature vector 1 (Figure 5D) we observed high values of ν_l in the V segment, suggesting that there are systematic and heterogeneous differences between the V segment sequence distribution used in TCRα chains and in TCRβ chains. To confirm the observation, we used the RegressMuE model to predict the entire TCR sequence based just on its annotation as TCRα or TCRβ. In particular, as covariate vector H_i we used a one-hot encoding of the chain type annotated by CellRanger; sequences without an annotation were labeled as $(0, 0)$. We computed the regression shift ν_l in the same way as Equation 3, with the covariate H in place of z. Figure A.10 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.

A.8 Influenza analysis

A.8.1 Details

We downloaded publicly available influenza A(H3N2) HA sequences from GISAID[236]. We selected only sequences longer than 500 amino acids and with no ambiguous amino acids. Some se-

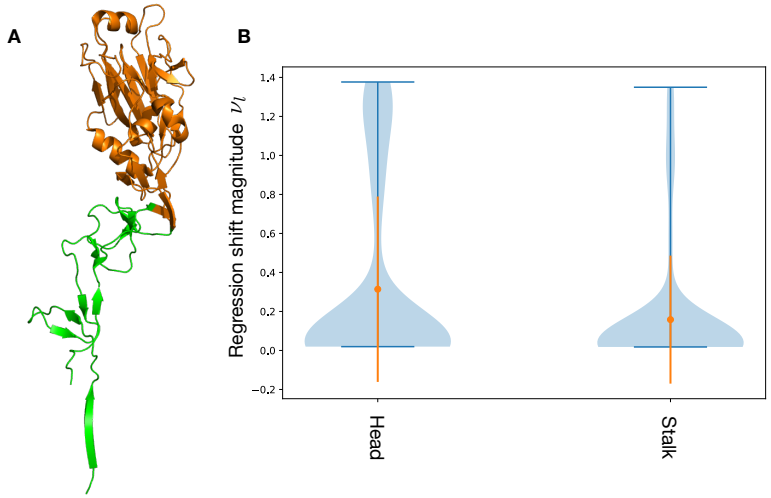


Figure A.11: Comparing RegressMuE model coefficients to HA1 structural domains. (A) Head (orange) and stalk (green) domains of the HA1 protein (PDB:4O5N); residues between sites 52 and 277 are defined as the head domain, and all others as stalk, following Lee et al. [153]. (B) Violin plots of regression shift ν_l (Equation A.67) for residues in the head domain (226 residues) versus the stalk domain (103 residues). Mean and standard deviation are shown in orange.

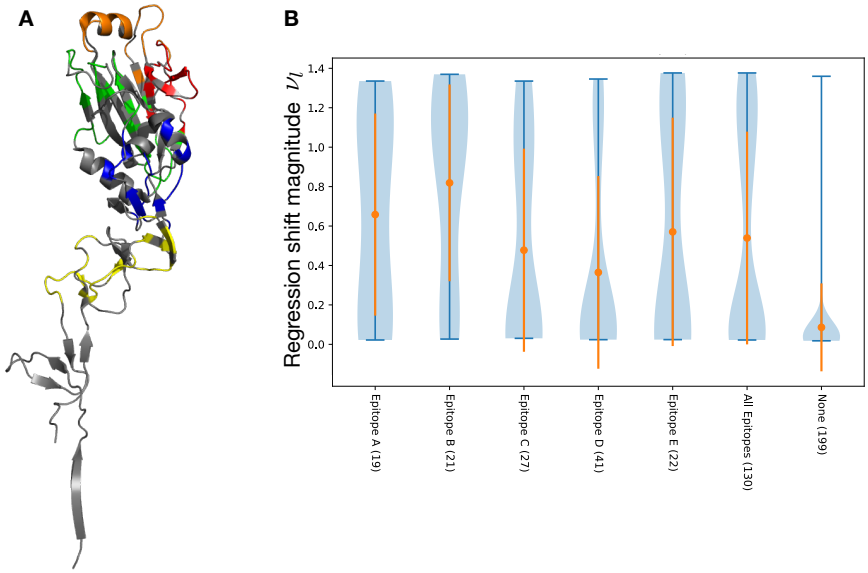


Figure A.12: Comparing MuE observation model regression coefficients to HA1 epitope regions. (A) Epitope regions A (red), B (orange), C (yellow), D (green), E (blue) [288,184]. (B) Violin plots of regression shift ν_l (Equation A.67) for residues in each epitope region, for all epitope regions together, 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.

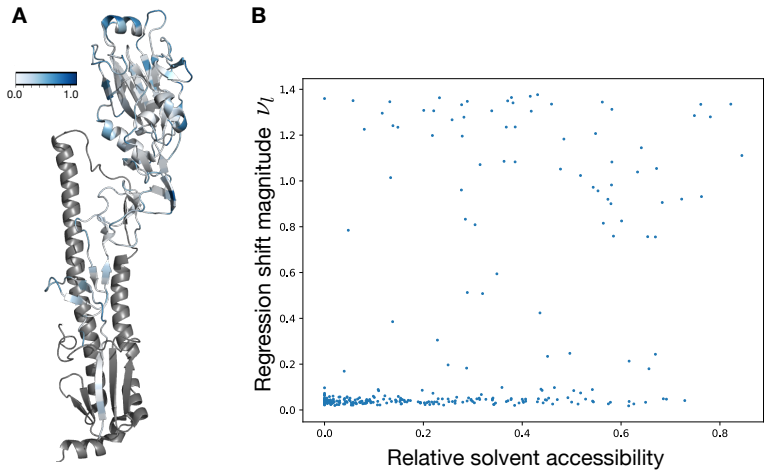


Figure A.13: Comparing MuE observation model regression coefficients to HA1 relative solvent accessibility. (A) Relative solvent accessibility of the HA1 protein (PDB:4O5N), computed using DSSP [132] and the maximum values in Tien et al. [258] with the Biopython API [42]. HA2 protein shown in dark gray. (B) Relative solvent accessibility versus regression shift magnitude ν_l (Equation A.67), residue-by-residue. Spearman $\rho = 0.41, p < 10^{-13}$.

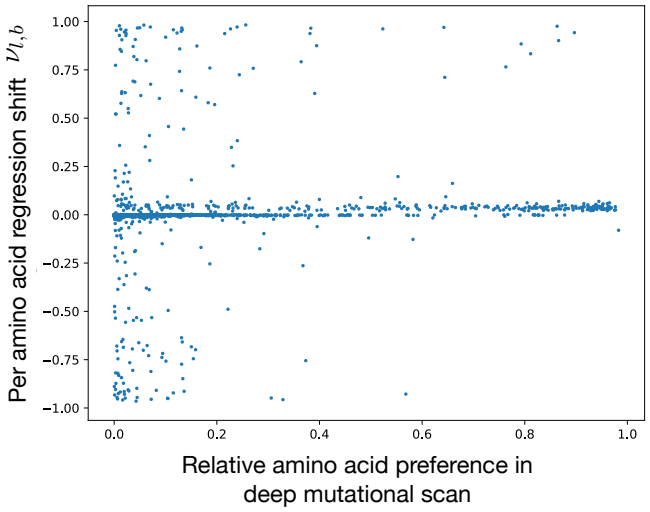


Figure A.14:
Comparing MuE observation model regression coefficients to a deep mutational scan of HA. X-axis: regression shift for each amino acid at each position from 1968 to 2019,

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

(terms defined as in Equation A.67). Y-axis: relative preference for point mutants with amino acid b at position l in the deep mutational scan performed in Lee et al. [153]. Spearman $\rho = 0.08, p < 10^{-11}$.

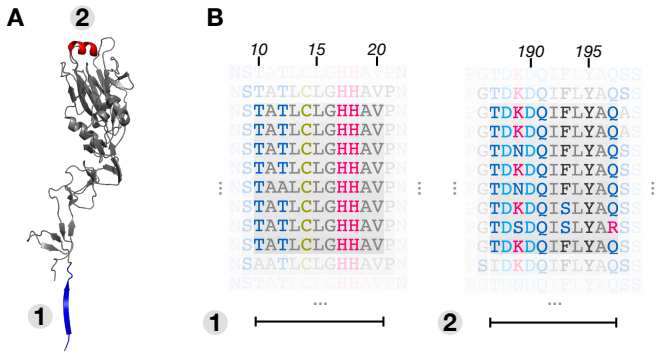


Figure A.15: Generating forecasted samples. (A) Two locations in the reference structure PDB:4O5N, indicated in blue and red, corresponding to low and high ν_l values (Figure 6B). (B) Segments of sequences sampled from the posterior predictive distribution for the year 2024. The alignment variable w_{ref} is fixed based on the reference (PDB:4O5N), such that segments 1 and 2 correspond to the annotated structural features in A, and the column numbering is standard for influenza A(H3N2) [30].

quences 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. [153], we randomly subsampled six sequences per month, from 1968 to October 2019, to form the dataset. In the forecasting experiments we removed the mis-annotated data identified in the 2008 outlier cluster marked by ‡ in Figure 6E prior to subsampling (GISAID identifiers `EPI_ISL_24813`, `EPI_ISL_24814`, ..., `EPI_ISL_24867`). Accession numbers for the complete dataset can be found in the Supplementary Table 1 file in the published paper [284]; our results were stable upon resampling. We extracted only the first 350 amino acids of each HA sequence, covering HA1 in the reference A(H3N2) numbering [30].

We used $M = 361$ in the MuE distribution. We set the prior on indels to $\mu^{(r)} = \mu^{(u)} = (1000, 1)$. We trained each model for 7 epochs, which was sufficient for convergence. Hyperparameters and training schedule were otherwise set as in Section A.6.1. To produce the latent embedding

in Figure 6D, however, we annealed the ELBO weighting χ only up to $\chi_0 = 0.001$ after 7 epochs, providing only very weak prior regularization such that the embedding corresponds to approximately the maximum likelihood estimator of z (and we avoid posterior collapse).

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

$$\nu_l = \left[\sum_{b=1}^{B} \left(\mathbb{E}[Y_{l,b} | \hat{w}_{\text{ref}}, t = 2019] - \mathbb{E}[Y_{l,b} | \hat{w}_{\text{ref}}, t = 1968] \right)^2 \right]^{1/2} \tag{A.67}$$

using as reference the HA1 sequence from PDB:4O5N. The expectation is estimated using the variational approximation to the posterior with 10 Monte Carlo samples. $\hat{w}_{\text{ref}}$ is estimated using a single sample from the variational approximation to the posterior and the Viterbi algorithm. In evaluating the association between the shift vector ν_l and epitope regions of HA1, we specifically compared to the 16 sites with clear antigenic selection in at least one human sera identified in Lee et al. [152].

A.8.2 Further results

In addition to the classic epitope regions, we also compared the regression shift ν to the structural domains of the HA1 protein (Figure A.11), relative solvent accessibility (Figure A.13), and relative amino acid preference in a deep mutational scan evaluating fitness effects of mutations (Figure A.14).

The cluster marked ‡ in Figure 6E appears around 2008 but the latent representation of these

sequences is close to that of sequences from the late 1960s or 1970s; this cluster comes from an experiment performed in 2008 on 1968 sequences, rather than contemporary patient samples as in the rest of the GISAID dataset.

MuE observation models can be used to generate samples of future sequences, enabling experimental tests of immune response and antibody titer on sequences that are likely to emerge in the future. We generated samples for the year 2024 from the RegressMuE, and confirmed that they are similar to previously observed sequences, as would be expected (Figure A.15). In particular, we sampled from

$$\begin{aligned} \phi &\sim q(\phi|Y_{\mathcal{D}}) \\ Y_i &\sim p_{\text{RegressMuE}}(y|\hat{w}_{\text{ref}}, \phi, t = 2024) \end{aligned} \tag{A.68}$$

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

B

Supplementary Material for Chapter 2

Sections B.1-B.6 are our theoretical results. Section B.7 describes our simulation experiments. Section B.8 details how we implemented scalable inference for BEAR models. Sections B.9-B.13 provide details on our empirical results based on real data. The Datasets.xlsx file in the supplementary material of the publication contains information on all the datasets, including links or accession numbers for public databases. Code and documentation are available at https://github.com/

debbiemarkslab/BEAR.

B.1 Theory introduction

BEAR models can be used to address a variety of different estimation and testing problems, and the theoretical questions that arise in each case are related but distinct. One crucial, high-level distinction is between the "finite-lag case" (where we assume the model lag L is finite) and the "infinite-lag case" (where we allow the model lag L to approach infinity). In addressing nonparametric density estimation, it is crucial to consider the infinite lag case, since it is likely in practice that the true distribution can only be matched in the infinite L limit. On the other hand, when it comes to diagnosing misspecification or constructing hypothesis tests, the finite lag case is more acceptable since it is likely in practice that any differences between the model and the data, or between two datasets, will be reflected in finite marginals of the data distribution. The finite lag case is complicated by the fact that it is likely that many kmer-to-base transitions have extremely low probability in practice; even on massive datasets, we observe many transitions with no counts whatsoever. To deal with this case, we develop theoretical tools to accommodate the possibility that some transitions truly have probability zero under the data generating distribution.

An essential and innovative aspect of our formalism is the focus on "subexponential" sequence distributions that obey an exponential moment bound on their length. Our choice to consider sequence distributions that have no upper bound on the lengths of sequences they produce separates our theory from the theory of distributions on finite sets. On the other hand, moment bound as-

sumptions separate our theory from the theory of distributions on countable sets.

The theory will be organized as follows. Section B.2 describes basic theoretical properties finite-lag Markov sequence models, including their expressiveness and subexponentiality. Subexponential sequence models will be introduced in general here. Section B.3 demonstrates consistency of inference with a fixed lag and in model selection between lags. A connection is established between effective model dimensions and topologies of de Bruijn graphs. Section B.4 describes the behavior of the model when inference proceeds by empirical Bayes. The parameter h is established as a descriptor of misspecification. Section B.5 describes theoretical guarantees on the behavior of goodness-of-fit and two-sample tests. Finally, section B.6 demonstrates consistency in the infinite lag case. Later sections depend on definitions and results established in previous sections with the exception that section B.6 may be read immediately after reading the definitions at the top of section B.3.

B.1.1 Notation

We consider an alphabet $\mathcal{B}$ with more than one letter. Define $\tilde{\mathcal{B}} = \mathcal{B} \cup \{\$\}$ where $\$$ is interpreted as the stop symbol, i.e. $\$$ may only appear as the last letter of a sequence. Also define the set of strings of the alphabet $\mathcal{B}$ of length L that start with any number (including 0) of repeated $\emptyset$ symbols, $\mathcal{B}^o_L$. For a sequence X of letters in $\mathcal{B}$, possibly terminated by $\$$, we define $|X|$ as its length, including the stop symbol $\$$ but not any start symbols $\emptyset$. For two strings X, X' define $\#X'(X)$ the number of occurrences of X' as a substring in X and, if X is not terminated by $\$$, define (X, X') as the concatenated string. We also define the substring from index i to j (inclusive) of X as $X_{i:j}$.

Define the set S of all finite sequences terminated by a stop symbol and give it the discrete topol-

ogy. Note that S is countable. Say p is a distribution of S. We will use $\mathbb{E}_p$, or $\mathbb{E}$ if there is an unambiguous data-generating distribution, to denote taking an expectation; for example, $\mathbb{E}_p \# X'$ is the expected number of occurrences of the substring X' in sequences drawn from p. For a sequence Y possibly not terminated by a stop symbol, we define $p(Y \dots) = p(\{X \in S \mid X_i = Y_i \, \forall i \leq |Y|\})$. We also define subexponential moment bounds, an assumption we will make great use of:

Definition B.1.1 (Subexponential sequence distributions)**.** *We say a distribution p on S is subexponential if for a $t > 0$, $\mathbb{E}_p \exp(t|X|) < \infty$.*

For a random variable Z on a probability space with probability P, and a measurable set A in the sample space, we define

$$\mathbb{E}[Z; A] = \mathbb{E}[Z \mathbb{1}_A] = \mathbb{E}[Z|A]P(A)$$

where $\mathbb{1}_A$ is the random variable with $\mathbb{1}_A = 1$ on A and $\mathbb{1}_A = 0$ outside of A. As well, for two real sequences $(a_n)_{n \in \mathbb{N}}, (b_n)_{n \in \mathbb{N}}$, both possibly undefined for small n, we write $a_n \lesssim b_n$ to mean that there is a positive constant C such that eventually $a_n \leq C b_n$. We write $a_n \sim b_n$ when $a_n \lesssim b_n$ and $a_n \gtrsim b_n$. We define $a \wedge b$ as the minimum of a and b, and $a \vee b$ as the maximum.

B.2 Finite-lag Markov models

In this section we define finite-lag Markov models, and then study the expressiveness of the model class. After defining finite-lag Markov models, this section will concern itself with the expressiveness of the model class. We first show that while there are sequence distributions over S that are not

finite-lag Markov models, the set of finite-lag Markov models is nevertheless dense in the space of distributions over S. We then show that finite-lag Markov models are subexponential.

The class of finite-lag Markov models is defined to be

$$\text{Parameters: lag } L, \text{ transition probabilities } \{v_{k,b}\}_{k\in\mathcal{B}_L^o, b\in\tilde{\mathcal{B}}}$$

$$X_i = \emptyset \text{ for } i \leq 0$$

$$X_{i+1} \sim \text{Categorical}(\{v_{X_{i-L+1:i},b}\}_{b\in\tilde{\mathcal{B}}})$$

stopping generation when a \$ symbol is drawn and with parameters picked so that $|X| < \infty$ a.s.. These models are equivalent to Markov processes on the set $\mathcal{B}_L^o \cup \{(X, \$) \mid X \in \mathcal{B}_{L-1}^o\}$. The requirement that generated sequences be finite length a.s. is equivalent to the requirement that for every $k \in \mathcal{B}_L^o$ that is Markov-accessible, there is another $k' \in \mathcal{B}_L^o$ that is Markov-accessible from k such that $v_{k',\$} > 0$. Call p_v a probability distribution generated this way with parameters L, v. Call the set of such probability distributions with lag L $\mathcal{M}_L$. Define the set of all finite lag Markov models $\mathcal{M} := \cup_{L=1}^{\infty}\mathcal{M}_L$ and note $\mathcal{M}_1 \subset \mathcal{M}_2 \subset \ldots$. Defining $\Delta_{\tilde{\mathcal{B}}}$ as the $|\tilde{\mathcal{B}}| - 1$-dimensional simplex with coordinates indexed by $\tilde{\mathcal{B}}$, $\mathcal{M}_L$ is parametrized by transition probabilities in $\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$. This parametrization is not defined everywhere on the boundary and is not injective as if an L-mer k is not Markov-accessible by a distribution p_v, the vector of probabilities v_k does not affect p_v's distribution. This parametrization is continuous in the sense of the topology described by proposition B.2.2.

We first give some examples of simple sequence distributions that are not finite-lag Markov.

Proposition B.2.1. *Not all possible distributions over S are in $\mathcal{M}$.*

Proof. Let $A \in \mathcal{B}$ and p^* be a distribution over finite sequences that puts probability a_i on the sequence $A \times i := A \dots A$ of length i with $\sum_{i=0}^{\infty} a_i = 1$. Assume $p^* \in \mathcal{M}_L$ with transition probabilities $\{v_{k,b}\}_{k \in \mathcal{B}_L^o, b \in \tilde{\mathcal{B}}}$.

For $i \leq L$, define $v_i := v_{(\emptyset \times (L-i), A \times i)}$, i.e. the vector of transition probabilities from the L-mer that is $L - i$ $\emptyset$ symbols followed by i A symbols. For $i \geq L$ call $v_i := v_L$.

Notice that for any i, the \$-component of the vector v_i is $p^*(|X| = i \,||X| \geq i) = \frac{a_i}{S_i}$ where $S_i := \sum_{j=i}^{\infty} a_j$. Thus the A-component is $1 - \frac{a_i}{S_i} = \frac{S_{i+1}}{S_i}$. By the definition of the sequence $(v_i)_{i=1}^{\infty}$, it is constant for $i \geq L$. Call $\alpha := S_{L+1}/S_L = v_{L,A} = v_{i,A} = S_{i+1}/S_i$ for all $i \geq L$. Thus for all $i > L$, $a_i = S_i v_{i,\$} = \alpha^{i-L} S_L v_{L,\$} = \alpha^{i-L} a_L$. Thus the sequence a_i eventually decays exponentially and, as examples, it is impossible that $a_i \sim 1/i!$ or $a_i \sim 1/i^2$. □

Next we show that $\mathcal{M}$ is dense in the set of probability distributions on S. To speak of density, we review the topology and types of convergence on the set of distributions of S in this next proposition.

Proposition B.2.2. *The topology of convergence in total variation, convergence in distribution, and pointwise convergence of the probability of each $X \in S$ are identical.*

Proof. Pointwise convergence of the probability of each $X \in S$ implies convergence in total variation by Scheffé's lemma. It is also known that the topology induced by the total variation metric is stronger than the topology of convergence in distribution. Finally, since for each $X \in S$, the set

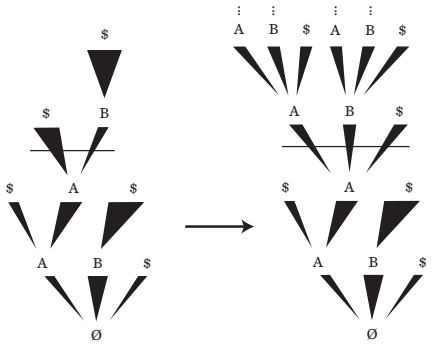


Figure B.1: Example application of this construction to the distribution on the left. Transition probabilities for kmers smaller than $L = 2$ are those defined by the original distribution, while those for larger kmers are all $1/3$. The thickness of each line denotes the probability of the transition.

$\{X\}$ is open and closed, so that the Portmanteau lemma shows that convergence in distribution implies pointwise convergence. □

Lemma B.2.3. *Say p is a distribution on S. There is a lag L Markov model, p_L, such that for all $X \in S$, if $|X| \leq L$, $p_L(X) = p(X)$, and if $|X| > L$, $p_L(X) = p(X_{1:L})|\tilde{\mathcal{B}}|^{-(|X|-L)}$.*

Proof. For all $k \in \mathcal{B}_L^o, b \in \tilde{\mathcal{B}}$, if there is a start symbol $\emptyset$ in k, define $v_{k,b} = \frac{p((k,b)\ldots)}{p(k\ldots)}$, otherwise, define $v_{k,b} = |\tilde{\mathcal{B}}|^{-1}$. It is clear p_v satisfies the properties of p_L (Fig B.1).

□

Corollary B.2.4. *$\mathcal{M}$ is dense in the set of distributions of S.*

Proof. Define p to be a distribution on S with finite support, $\{X_n\}_{n=1}^N$. Pick an $L > |X_n|$ for all n, so that with the definition of lemma B.2.3, $p_L = p$ and thus $p \in \mathcal{M}_L$. Now note that any distribution on S can be approximated at finitely many points in S arbitrarily well by distributions with finite support. The result follows from proposition B.2.2. □

Proposition B.2.5. *Finite-lag Markov models are subexponential.*

Proof. Say $p \in \mathcal{M}_L$ for some L, with transition probabilities v. Every $k \in \mathcal{B}_L^o$ that is Markov-accessible by p has a $k' \in \mathcal{B}_L^o$ that is Markov-accessible from k in less than s_k transitions such that $v_{k',\$} > 0$. Thus, $\inf_i p(|X| \leq i + s_k \,|\, |X| > i, X_{i-L+1:i} = k) > 0$. Define $s = \max_{k \text{ accessible}} s_k$, $q = \inf_i p(|X| \leq i + s \,|\, |X| > i) > 0$. Now note, for any positive integer m, $p(|X| > ms) = \prod_{i=1}^m p(|X| > is \,|\, |X| > (i-1)s) \leq (1-q)^m$. For a random variable $Z \sim \text{Geom}(q)$,

$$\begin{aligned}
\mathbb{E}_p \exp(t|X|) &= \int_0^\infty dy p\left(|X| > s\left(\frac{1}{st}\log(y)\right)\right) \\
&\leq \int_0^\infty dy p\left(|X| > s\left(\lfloor\frac{1}{st}\log(y)\rfloor\right)\right) \\
&\leq \int_0^\infty dy P\left(Z > \left(\lfloor\frac{1}{st}\log(y)\rfloor\right)\right) \\
&\leq \int_0^\infty dy P\left(Z > \left(\frac{1}{st}\log(y) - 1\right)\right) \\
&= \mathbb{E}\exp\left(ts(Z+1)\right)
\end{aligned} \tag{B.1}$$

The integral is finite for some $t > 0$ as geometric random variables are sub-exponential. □

B.3 Consistency in the finite L case

In this section we consider fitting to data BEAR models with fixed hyperparameters h and θ (that is, standard Bayesian Markov models). We first study the asymptotic behavior of the posterior over v, the transition probability parameter, conditional on a particular lag L. We prove a Wald-type consistency result, showing that the posterior concentrates on a neighborhood of the true data-generating

parameter value v^*, if such a value exists; when p^* is not in the model class $\mathcal{M}_L$, the posterior over v concentrates at the point v^* corresponding to the distribution $p_{v^*} \in \mathcal{M}_L$ closest in KL divergence to p^*. We next study the asymptotic behavior of the posterior over the lag L, building on the theory of nested model selection since L is a discrete variable. We show that the posterior concentrates at the true data-generating value L^* when such a lag exists (i.e. when there is some L^* such that $p^* \in \mathcal{M}_{L^*}$), and otherwise diverges. At a high level, neither of these results are surprising, and they would be expected to hold in general for well-behaved Bayesian models. The details of the model's asymptotic behavior, however, turn out to be somewhat unusual; as we shall see, the fact that some transitions from a particular kmer k to a base b have probability zero under the data-generating distribution p^* can complicate the normal story of Bayesian asymptotics.

To describe the possible kmer-base transitions, we define, for a distribution on S, p, and a lag L, the set of accessible kmers $\text{acc}_L(p) = \{k \in \mathcal{B}_L^o \mid p(\#k > 0) > 0\}$ and transitions $\text{supp}_L(p) = \{(k, b) \mid k \in \mathcal{B}_L^o, b \in \tilde{\mathcal{B}}, p(\#(k, b) > 0) > 0\}$. Define also, for any particular a $k \in \mathcal{B}_L^o$, the set of allowed transitions $\text{supp}_L(p)|_k := \{b \in \tilde{\mathcal{B}} \mid (k, b) \in \text{supp}_L(p)\}$. Define the restriction of the parameter space $\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$ to the support of p^*, $\tilde{\Delta}_L(p^*) = \prod_{k \in \text{acc}_L(p^*)} \Delta_{\text{supp}_L(p^*)|_k}$. If $v \in \Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$, we will often use the abbreviation $\text{supp}(v) = \text{supp}_L(p_v)$ for convenience.

Say p^* is a distribution on S and L is a lag. Define the transition probabilities v^*, corresponding to the closest model in $\mathcal{M}_L$ to p^* (as measured by KL), as

$$v^* = \underset{v \in \Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}}{\arg\min}\, \text{KL}(p^* || p_v) = \underset{v}{\arg\max}\, \mathbb{E} \log p_v(X) = \underset{v}{\arg\max} \sum_{k,b} \mathbb{E}\left[\#(k, b)\right] \log v_{k,b}.$$

Unlike for many other statistical models studied in other contexts, here we can easily compute the closest model to the data-generating distribution: using Lagrange multipliers, one may see that for all $k \in \text{acc}_L(p^*)$, $v^*_{k,b} = \mathbb{E}\left[\#(k,b)\right] / \mathbb{E}\left[\#k\right]$. We then define $p^{*(L)} = p_{v^*}$ as the best approximation to p^* in $\mathcal{M}_L$. Note $\text{supp}(v^*) = \text{supp}_L(p^{*(L)}) = \text{supp}_L(p^*)$.

We now ask whether Bayesian inference on $\mathcal{M}_L$ is consistent, i.e., whether the posterior converges to a point mass at $p^{*(L)}$, even in the case where $\text{supp}_L(p^*)$ is not all of $\mathcal{B}_L^o \times \tilde{B}$. The result is a classic Wald-type argument, adapted from theorem 2.3 of Miller [176] and theorem 1.3.4 in Ghosh & Ramamoorthi [88]. The primary difficulty in the proof is that these previous theorems assume the true parameter value lies on the interior of the parameter space and rely on uniform convergence of the mean log likelihood in a neighborhood around the true value. In our case, we can have $v^*_{k,b} = 0$, so that the true parameter value lies on the boundary of its space $\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$ and the likelihood function diverges at this boundary point.

Theorem B.3.1. *Say p^* is a distribution on S with $\mathbb{E}|X| < \infty$. Say Π is a prior on $\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$ that assigns probability 0 to the set of v with $p_v \notin \mathcal{M}_L$. Say $X_1, X_2, \cdots \sim p^*$ iid. Call $V = \{v \in \Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o} \mid p_v = p^{*(L)}\}$ and assume that it is not disjoint from the support of Π. Then for all open sets U containing V,*

$$\Pi(U|X_1, \ldots, X_N) \to 1$$

a.s.. As a probability distribution on the space of measures on S, $\Pi|X_1, \ldots, X_N \to \delta_{p^{*(L)}}$.

Proof. Define v^* as the transition probabilities of $p^{*(L)}$. Define $l_N(v) = -\frac{1}{N}\sum_{n=1}^{N} \log(p_v(X_n))$,

which is continuous in v and $\nu^* = \min_{(k,b)\in\text{supp}(v^*)} v^*_{k,b}$. Note that

$$\mathbb{E}\log p^{*(L)}(X) = \mathbb{E}\sum_{i=1}^{|X|}\log v^*_{X_{i-L:i-1},X_i} \geq \mathbb{E}|X|\log\nu^*.$$

First we show that the likelihood of the data is eventually small in a neighborhood of the boundary. Pick an $\eta_1 > 0$. Say $(k,b) \in \text{supp}(v^*) = \text{supp}_L(p^*)$ and define $q_{k,b} = p^*(\#(k,b) > 0)$ which is positive. Pick a positive

$$\nu_{k,b} < \exp\left(-q^{-1}_{k,b}(\eta_1 - \mathbb{E}|X|\log\nu^*)\right) \wedge v^*_{k,b}.$$

$$\begin{aligned}\mathbb{E}\sup_{v \text{ s.t. } v_{k,b}<\nu_{k,b}} l_1(v^*) - l_1(v) =& \mathbb{E}\left[\sup_{v \text{ s.t. } v_{k,b}<\nu_{k,b}} \log p_v(X)\right] - \mathbb{E}\left[\log p_{v^*}(X)\right] \\ \leq& q_{k,b}\log\nu_{k,b} + (-\log\nu^*)\mathbb{E}|X| < -\eta_1.\end{aligned} \tag{B.2}$$

Thus defining $U_1 = \{v \in \Delta^{\mathcal{B}^o_L}_{\tilde{\mathcal{B}}} \mid \text{there exists } (k,b) \in \text{supp}(v^*) \text{ s.t. } v_{k,b} < \nu_{k,b}\}$, a.s., for large enough N, $l_N(v^*) - l_N(v) < -\eta_1$ for all $v \in U_1$ by the SLLN.

Call the complement of U_1 K. K is compact and for all $v \in K$, $\text{supp}(v^*) \subseteq \text{supp}(v)$. Note that V is compact and in the interior of K. Pick a positive ν_K which has, for every $(k,b) \in \text{supp}(v^*)$, $\nu_K < \nu_{k,b}$. Then

$$\mathbb{E}\sup_{v\in K}|l_1(v^*) - l_1(v)| \leq |\log(\nu_K \wedge \nu^*)|\mathbb{E}|X| < \infty.$$

Then by theorem 1.3.3 in Ghosh & Ramamoorthi [88], a.s., $l_N(v^*) - l_N(v)$ converges uniformly to $\text{KL}(p^*||p^{*(L)}) - \text{KL}(p^*||p_v) \leq 0$ on K (note, for the application of theorem 1.3.3 in Ghosh & Ramamoorthi [88], this quantity is well defined even if p_v is not a distribution over finite strings).

Now pick an open neighborhood U of V. By the continuity of $v \mapsto \text{KL}(p^*||p_v)$, since $K \setminus U$ is compact, $\inf_{v \in K \setminus U} \text{KL}(p^*||p_v) > \text{KL}(p^*||p^{*(L)})$ otherwise there would be a $v \in V \setminus K$. Thus we can pick a positive $\text{KL}(p^*||p^{*(L)}) + \eta_2 < \inf_{v \in K \setminus U} \text{KL}(p^*||p_v)$. Since $v \mapsto \text{KL}(p_{v^*}||p_v)$ is continuous and K is a neighborhood of V, there is an open $U_2 \subset K \cap U$ containing V such that one can pick an η_3 with $\sup_{v \in U_2} \text{KL}(p^*||p_v) - \text{KL}(p^*||p^{*(L)}) < \eta_3 < \eta_1 \wedge \eta_2$. Then a.s. eventually, $l_N(v^*) - l_N(v) < -\eta_2$ for all $v \in K \setminus U$ and $l_N(v^*) - l_N(v) > -\eta_3$ for all $v \in U_2$. Thus, a.s. for large enough N,

$$
\begin{aligned}
&\Pi(U|X_1, \ldots, X_n) \\
&= \frac{\int_U d\Pi e^{N(l_N(v^*) - l_N(v))}}{\int_U d\Pi e^{N(l_N(v^*) - l_N(v))} + \int_{K \setminus U} d\Pi e^{N(l_N(v^*) - l_N(v))} + \int_{U_1 \setminus U} d\Pi e^{N(l_N(v^*) - l_N(v))}} \\
&\geq \left(1 + \frac{\int_{K \setminus U} d\Pi e^{N(l_N(v^*) - l_N(v))}}{\int_{U_2} d\Pi e^{N(l_N(v^*) - l_N(v))}} + \frac{\int_{U_1} d\Pi e^{N(l_N(v^*) - l_N(v))}}{\int_{U_2} d\Pi e^{N(l_N(v^*) - l_N(v))}}\right)^{-1} \\
&\geq \left(1 + \frac{\Pi(K \setminus U) e^{-N\eta_2}}{\Pi(U_2) e^{-N\eta_3}} + \frac{\Pi(U_1) e^{-N\eta_1}}{\Pi(U_2) e^{-N\eta_3}}\right)^{-1} \\
&\to 1.
\end{aligned}
\tag{B.3}
$$

Finally, as a probability distribution on the space of measures on S, $\Pi|X_1, \ldots, X_n \to \delta_{p^{*(L)}}$. This follows from the fact that the prior and thus posterior probability of $p_v \notin \mathcal{M}_L$ is 0 and so one may push forward the measure from $\Delta_{|\tilde{\mathcal{B}}|}^{\mathcal{B}_L^o}$ to the space of probability measures on S. The im-

age of V is a point p_{v^*}. Since this mapping is continuous, it preserves the weak convergence of the measure, in this case to a point mass. □

Next we will study the posterior distribution of the BEAR model over the lag L, showing under general assumptions that the posterior concentrates on the true data-generating value L^* (when such a value exists). Our analysis builds off of standard asymptotic analyses of nested Bayesian model selection, since models with different lags are nested, i.e. $\mathcal{M}_L \subset \mathcal{M}_{L'}$ when $L' > L$. Typically, when a simpler model (e.g. $\mathcal{M}_L$) is nested inside a more complex model (e.g. $\mathcal{M}_{L'}$), and the data-generating distribution p^* is in the simpler model, the log Bayes factor comparing the two models will asymptotically prefer the simpler model and scale as $\frac{1}{2}(\text{dim}' - \text{dim}) \log N$ where dim$'$ is the dimension of the parameter space in the more complex model and dim is the dimension in the simpler model[50]. This $O(\log N)$ term, which is independent of the prior, can be thought of as originating from the Laplace approximation to the marginal likelihood; it is the basis of such widely used model-selection techniques as the Bayesian information criterion.

Somewhat surprisingly, the fact that some transitions may have probability zero ($v^*_{k,b} = 0$) changes the asymptotic behavior of the log Bayes factor, in particular by altering the dimension factor dim$'$ − dim. In effect, dimensions of the parameter space corresponding to kmers that occur with probability zero under p^* do not contribute to the dimension count, while dimensions for which $v^*_{k,b} = 0$ do not count as full dimensions; this leads to the notion of an "effective model dimension", defined as $\text{dim}^{\text{eff}}_L(p^*) := |\text{supp}_L(p^*)| - |\text{acc}_L(p^*)| + \sum_{k \in \text{acc}_L(p^*)} \sum_{b \notin \text{supp}_L(p^*)|_k} \alpha_{k,b}$ where $\alpha_{k,b}$ is the concentration of the Dirichlet prior. This effective dimension depends the data-

generating distribution p^* and on the prior hyperparameters, not just on L. Note that the unusual asymptotic behavior of BEAR models does not just come from their Markov structure; even in the everyday example of a Dirichlet-Categorical model, if some outcomes of the Categorical distribution have probability exactly zero under the true data-generating distribution, the standard Laplace approximation does not hold, and the Dirichlet prior contributes $O(\log N)$ terms to the log marginal likelihood[222].

Theorem B.3.2. *Say p^* is a distribution on S with $\mathbb{E}|X|^2 < \infty$ and say $X_1, X_2, \cdots \sim p^*$ iid. Given L, consider a* $\text{Dirichlet}(\alpha_{k,b})_{b\in\tilde{\mathcal{B}}}$ *prior on the simplex in $\Delta^{\mathcal{B}_L^o}_{\tilde{\mathcal{B}}}$ corresponding to the L-mer k. For all L, assume $\alpha_{k,b} > 0$ for $(k,b) \in \text{supp}_L(p^*)$ (otherwise $p((X_n)_{n=1}^N|\mathcal{M}_L)$ is eventually 0 a.s.).*

Define $\kappa\text{L}(p^*||\mathcal{M}_L) := \inf_{p\in\mathcal{M}_L} \kappa\text{L}(p^*||p)$. Given $L_1 \neq L_2$, if[*] $\kappa\text{L}(p^*||\mathcal{M}_{L_2}) > \kappa\text{L}(p^*||\mathcal{M}_{L_1})$,

$$\log \frac{p((X_n)_{n=1}^N|\mathcal{M}_{L_1})}{p((X_n)_{n=1}^N|\mathcal{M}_{L_2})} = N\left(\kappa\text{L}(p^*||\mathcal{M}_{L_2}) - \kappa\text{L}(p^*||\mathcal{M}_{L_1})\right) + O_p(\sqrt{N}). \tag{B.4}$$

Otherwise, if $p^* \in \mathcal{M}_{L_1}, \mathcal{M}_{L_2}$ and, defining, for a lag L, $dim_L^{\text{eff}}(p^*) := |\text{supp}_L(p^*)| - |\text{acc}_L(p^*)| + \sum_{k\in\text{acc}_L(p^*)} \sum_{b\notin\text{supp}_L(p^*)|_k} \alpha_{k,b}$,

$$\log \frac{p((X_n)_{n=1}^N|\mathcal{M}_{L_1})}{p((X_n)_{n=1}^N|\mathcal{M}_{L_2})} = \frac{1}{2}\left(dim_{L_2}^{\text{eff}}(p^*) - dim_{L_1}^{\text{eff}}(p^*)\right)\log N + O_p(1). \tag{B.5}$$

[*] We do not need to assume $\mathbb{E}\log p > -\infty$ as we may define in this case $\text{KL}(p^*||\mathcal{M}_{L_2}) - \text{KL}(p^*||\mathcal{M}_{L_1}) = -\mathbb{E}\log p^{*(L_2)}(X) + \mathbb{E}\log p^{*(L_1)}(X)$ which we will see is bounded by the moment bound assumption $\mathbb{E}|X|^2 < \infty$.

Proof. For a lag L, note $\dim(\tilde{\Delta}_L(p^*)) = \text{supp}_L(p^*) - \text{acc}_L(p^*)$. Put a Dirichlet$(\alpha_{k,b})_{b\in\text{supp}_L(p^*)|_k}$ prior on each $\Delta_{\text{supp}_L(p^*)|_k}$. Call $\tilde{\mathcal{M}}_L$ the set of probability distributions described by $\tilde{\Delta}_L(p^*)$. We will show that $\text{KL}(p^*||p_\cdot)$ is maximized in the interior of $\tilde{\Delta}_L(p^*)$ so that the asymptotics of the marginal likelihood $\left(p(X|\tilde{\mathcal{M}}_L)\right)$ are well understood. In $\Delta_{|\tilde{\mathcal{S}}|}^{\mathcal{B}_L^o}$ however, there are dimensions that correspond to k-mer - base transitions that are impossible under p^*. Using the symmetry of the Dirichlet prior, we can de-couple the asymptotics of these excess dimensions from the asymptotics of the much more "natural" space of $\tilde{\Delta}_L(p^*)$:

$$\begin{aligned}\log\left(p((X_n)_{n=1}^N|\mathcal{M}_L)\right) &= \sum_{k\in\text{acc}_L(p^*)}\left(\log\frac{\Gamma(\sum_b\alpha_{k,b})}{\Gamma(\sum_b\alpha_{k,b}+\#k)} - \sum_{\text{supp}_L(p^*)|_k}\log\frac{\Gamma(\alpha_{k,b})}{\Gamma(\alpha_{k,b}+\#(k,b))}\right)\\ &= \log\left(p((X_n)_{n=1}^N|\tilde{\mathcal{M}}_L)\right)\\ &\quad+\sum_{k\in\text{acc}_L(p^*)}\left(\log\frac{\Gamma(\sum_b\alpha_{k,b})}{\Gamma(\sum_b\alpha_{k,b}+\#k)} - \log\frac{\Gamma(\sum_b'\alpha_{k,b})}{\Gamma(\sum_b'\alpha_{k,b}+\#k)}\right)\end{aligned} \tag{B.6}$$

where $\sum_b'$ is a sum over the $b\in\text{supp}_L(p^*)|_k$, and where $\#k$ in this case is $\sum_{n=1}^N\#k(X_n)$ and $\#(k,b)$ is similar. We will deal with each of these terms in turn.

To analyze the first of these terms, we first check regularity conditions. For $v\in\tilde{\Delta}_L(p^*)$ and strings $X_1,\dots X_N$, define

$$l_N(v) = -\frac{1}{N}\log\prod_{n=1}^N p_v(X_n) = -\frac{1}{N}\sum_{(k,b)\in\text{supp}_L(p^*)}\#(k,b)\log v_{k,b}$$

$$l(v) = -\mathbb{E}\log p_v(X) = -\sum_{(k,b)\in\text{supp}_L(p^*)}\mathbb{E}[\#(k,b)]\log v_{k,b}.$$

Call v_n the minimizer of l_N and v^* the minimizer of l. Note v^* is also the minimizer of $v \mapsto \text{KL}(p^*||p_v)$ for $v \in \tilde{\Delta}_L(p^*)$ and has $v^*_{k,b} = \mathbb{E}\#(k,b)/\mathbb{E}\#k$. In particular $p_{v^*} = p^{*(L)}$ so that $\text{KL}(p^*||\mathcal{M}_L) = \text{KL}(p^*||\tilde{\mathcal{M}}_L)$. One may check that l_N is C^∞, and, by seeing that it is a sum of convex functions, convex. Calling D^m the m-th derivative operator (D^0 the identity), $\|\cdot\|$ some norm on $\mathbb{R}^{\dim(\tilde{\Delta}_L(p^*))^m}$, and E some set whose closure is in the interior of $\tilde{\Delta}_L(p^*)$

$$\mathbb{E}\sup_{v\in E}\|D^m l_N(v)\| \leq \sum_{(k,b)\in \text{supp}_L(p^*)} \mathbb{E}\left[\#(k,b)\right]\sup_{v\in E}\|D^m \log v_{k,b}\| < \infty$$

since E is relatively compact. Thus, by theorem 1.3.3 of Ghosh & Ramamoorthi [88], $D^m l_N \to \mathbb{E}D^m l_1 = D^m l$ locally uniformly where the last equality is by Leibniz's rule due to the local boundedness of all derivatives. In particular, $D^3 l_N$ are uniformly bounded across N on a neighborhood of v^* and, sending $E \nearrow \tilde{\Delta}_L(p^*)$, and noting l_N is a.s. eventually $-\infty$ on the boundary of $\tilde{\Delta}_L(p^*)$, we see $l_N \to l$ pointwise a.s..

As in the analysis of Dawid [50], we write

$$\log p((X_n)_{n=1}^N|\tilde{\mathcal{M}}_L) = \log\frac{p((X_n)_{n=1}^N|\tilde{\mathcal{M}}_L)}{p_{v_N}(X_n)_{n=1}^N} + \log\frac{p_{v_N}(X_n)_{n=1}^N}{p^{*(L)}(X_n)_{n=1}^N} + \log p^{*(L)}(X_n)_{n=1}^N.$$

The above paragraph demonstrates that we satisfy conditions (2) of theorem 3.2 of Miller [176] and thus we can write

$$\log\frac{p((X_n)_{n=1}^N|\tilde{\mathcal{M}}_L)}{p_{v_N}(X_n)_{n=1}^N} = -\frac{1}{2}\dim\left(\tilde{\Delta}_L(p^*)\right)\log N + O(1)$$

and say that $v_N \to v^*$. Now, using the mean value theorem,

$$\log \frac{p_{v_N}(X_n)_{n=1}^N}{p^{*(L)}(X_n)_{n=1}^N} = -n(l_N(v_N) - l_N(v^*)) = -(\sqrt{N}(v^* - v_N))^T D^2 l_N(v')(\sqrt{N}(v^* - v_N))$$

for some v'_N on the ray connecting v^* and v_N. Call $Z_N = \sqrt{D^2 l_N(v'_N)}(\sqrt{N}(v^* - v_N))$. By local uniform convergence, since $v_N \to v^*$, $D^2 l_N(v'_N) \to D^2 l(v^*)$. Satisfying the conditions on a neighborhood of v^*, since $v_N \to v^*$, by theorem 5.41 in van der Vaart [268], $\sqrt{N}(v^* - v_N)$ converges in distribution to $N(0, D^2 l(v^*)^{-1})$. Thus, by Slutsky's theorem, Z_n converges to $N(0, I)$, and by the continuous mapping theorem $\log \frac{p_{v_n}(X_n)_{n=1}^N}{p_{v_0}(X_n)_{n=1}^N} = Z_n^T Z_n$ converges in distribution to $\chi^2_{\dim(\tilde{\Delta}_L(p^*))}$; thus this term is $O_P(1)$. Recall from the remark in the last paragraph that $\mathrm{KL}(p^*||\tilde{\mathcal{M}}_L) = \mathrm{KL}(p^*||\mathcal{M}_L)$ for all L; note in particular $p^* \in \tilde{\mathcal{M}}_L$ if and only if $p^* \in \mathcal{M}_L$. Then finally, by the analysis of Dawid [50], since $\mathbb{E}[\log p^{*(L)}(X_n)_{n=1}^N]^2 \leq (\log(\min_{k,b} v_{k,b}^{-1}))^2 \mathbb{E}|X|^2 < \infty$, $\log p^{*(L)}(X_n)_{n=1}^N = \log p^*(X_n)_{n=1}^N$ if $p^* \in \tilde{\mathcal{M}}_L$ and

$$\log p^{*(L)}(X_n)_{n=1}^N = N\left[-\mathrm{KL}(p^*||\mathcal{M}_L) + \mathbb{E}\log p(X)\right] + O_P(\sqrt{N})$$

otherwise.

By our analysis above we can say that given $L_1 \neq L_2$, if $\mathrm{KL}(p^*||\mathcal{M}_{L_2}) > \mathrm{KL}(p^*||\mathcal{M}_{L_1})$,

$$\log \frac{p((X_n)_{n=1}^N|\tilde{\mathcal{M}}_{L_1})}{p((X_n)_{n=1}^N|\tilde{\mathcal{M}}_{L_2})} = N\left(\mathrm{KL}(p^*||\mathcal{M}_{L_2}) - \mathrm{KL}(p^*||\mathcal{M}_{L_1})\right) + O_p(\sqrt{N}). \tag{B.7}$$

Otherwise, if $p^* \in \mathcal{M}_{L_1}, \mathcal{M}_{L_2}$,

$$\log \frac{p((X_n)_{n=1}^N | \tilde{\mathcal{M}}_{L_1})}{p((X_n)_{n=1}^N | \tilde{\mathcal{M}}_{L_2})} = \frac{1}{2} \left(\dim(\tilde{\Delta}_{L_2}(p^*)) - \dim(\tilde{\Delta}_{L_1}(p^*)) \right) \log N + O_p(1). \tag{B.8}$$

Moving to the second term, for a $k \in \mathrm{supp}(v^*)$, by Stirling's formula,

$$\begin{aligned}
&\left(\log \frac{\Gamma(\sum_b \alpha_{k,b})}{\Gamma(\sum_b \alpha_{k,b} + \#k)} + \log \frac{\Gamma(\sum_b' \alpha_{k,b} + \#k)}{\Gamma(\sum_b' \alpha_{k,b})} \right) \\
&= \left(\sum_b \alpha_{k,b} - \frac{1}{2} \right) \log \left(\sum_b \alpha_{k,b} \right) - \sum_b \alpha_{k,b} \\
&\quad - \left(\#k + \sum_b \alpha_{k,b} - \frac{1}{2} \right) \log \left(\#k + \sum_b \alpha_{k,b} \right) + \left(\#k + \sum_b \alpha_{k,b} \right) \\
&\quad - \left(\sum_b{}' \alpha_{k,b} - \frac{1}{2} \right) \log \left(\sum_b{}' \alpha_{k,b} \right) + \sum_b{}' \alpha_{k,b} \\
&\quad + \left(\#k + \sum_b{}' \alpha_{k,b} - \frac{1}{2} \right) \log \left(\#k + \sum_b{}' \alpha_{k,b} \right) - \left(\#k + \sum_b{}' \alpha_{k,b} \right) \\
&\quad + O(1) \\
&= \left(\#k + \sum_b{}' \alpha_{k,b} - \frac{1}{2} \right) \log \left(\frac{\sum_b' \alpha_{k,b} + \#k}{\sum_b \alpha_{k,b} + \#k} \right) \\
&\quad - \left(\sum_b \alpha_{k,b} - \sum_b{}' \alpha_{k,b} \right) \log \left(\sum_b \alpha_{k,b} + \#k \right) + O(1) \\
&= \left(\#k + \sum_b{}' \alpha_{k,b} - \frac{1}{2} \right) O\left(\frac{1}{\#k} \right) \\
&\quad - \left(\sum_b \alpha_{k,b} - \sum_b{}' \alpha_{k,b} \right) \log \#k + O(1) \\
&= - \left(\sum_{b \notin \mathrm{supp}(p^*)|_k} \alpha_{k,b} \right) \log \#k + O(1)
\end{aligned} \tag{B.9}$$

Now note $\log \#k = \log N + \log\left(\frac{1}{N}\#k\right) = \log N + O(1)$ by the strong law of large numbers. Putting this together with B.7, B.8, B.6, and B.9 gives the result. □

So far, we've studied pairwise comparisons between models with different lags; we now study the posterior over lags. We start with the case where there is no true data-generating lag, i.e. $p^* \notin \mathcal{M}$. In this case, we can apply theorem B.3.2 to show that the posterior over lags diverges to infinity.

Corollary B.3.3. *Let $\pi(L)$ denote a prior over lags, with $\pi(L) > 0$ for all L. Choose for each lag a Dirichlet prior on the simplex $\Delta^{\mathcal{B}_L^o}_{\tilde{\mathcal{B}}}$ that satisfies the conditions of Theorem B.3.2. If p^* is subexponential but $p^* \notin \mathcal{M}$, the posterior diverges in the sense that for any choice of lag $\tilde{L}$, we have $\Pi(L > \tilde{L}|(X_n)_{n=1}^N) \to 1$ a.s..*

Proof. It is shown in the proof of theorem B.6.1 that as $L \to \infty$, we have $\textsc{kl}(p^*\|\mathcal{M}_L) \to 0$. Say $\tilde{L}$ is a lag, so, since $p^* \notin \mathcal{M}_{\tilde{L}}$, there exists some $\tilde{L}' > \tilde{L}$ such that $\textsc{kl}(p^*\|\mathcal{M}_{\tilde{L}'}) < \textsc{kl}(p^*\|\mathcal{M}_{\tilde{L}}) \leq \textsc{kl}(p^*\|\mathcal{M}_L)$ for all $L \leq \tilde{L}$. Note we have

$$\Pi(L \leq \tilde{L}|(X_n)_{n=1}^N) \leq \frac{\sum_{L\leq\tilde{L}} p((X_n)_{n=1}^N|\mathcal{M}_{L'})}{\sum_{L\leq\tilde{L}} p((X_n)_{n=1}^N|\mathcal{M}_{L'}) + p((X_n)_{n=1}^N|\mathcal{M}_{\tilde{L}'})}.$$

There are only finitely many L' less than or equal to $\tilde{L}$, so we can apply theorem B.3.2 and the conclusion follows. □

We now consider the case where $p^* \in \mathcal{M}$. Pick L^* to be the minimum lag such that $p^* \in \mathcal{M}_{L^*}$. We will need to assume, for theoretical tractability, that the prior over lags has finite support. Then we can establish sufficient conditions for the posterior to concentrate on the true value L^*.

Lemma B.3.4. *Let $\pi(L)$ be a prior over lags with $\pi(L) > 0$ for all L less than some $\tilde{L} \geq L^*$, and with $\pi(L) = 0$ for all $L > \tilde{L}$. Then $\Pi(L^*|(X_n)_{n=1}^N) \to 1$ in probability if $(\dim_L^{\text{eff}}(p^*))_{L \geq L^*}$ is non-decreasing and $\dim_{L^*+1}^{\text{eff}}(p^*) > \dim_{L^*}^{\text{eff}}(p^*)$.*

Proof. Apply theorem B.3.2. □

If transition probabilities $v_{k,b}^*$ were always non-zero, the effective dimension of the model would simply be the dimension of the parameter space $\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$, and thus the dimension would always increase with increasing lag, making lag selection consistent. Allowing for $v_{k,b}^* = 0$ makes the situation more complicated, since in fact the effective dimension may not increase with increasing lag. If this is indeed the case, the posterior will no longer be guaranteed to determine the true L^* from data, even asymptotically. In order to describe how the effective dimension in fact scales with the lag, we will introduce the notion of a distribution's de Bruijn graph: for a distribution p on S, the L-mer de Bruijn graph is the directed graph with nodes $\text{acc}_L(p)$ and a directed edge connecting L-mers $k \to k'$ if $k' = (k_{2:L}, b)$ for a $b \in \text{supp}_L(p)|_k$. (De Bruijn graphs are a common data analysis tool in biological sequence analysis, where they are typically constructed from an empirical distribution over observed sequences; here, we are in effect studying the asymptotic de Bruijn graph, i.e. the de Bruijn graph that we would have if an infinite amount of data were observed.) Call a de Bruijn graph a tree if every node has at most one parent (since sequences must start and end with start and stop symbols, there cannot be a loop where each kmer has just one parent). The next two results show that we can only consistently infer the true lag if the the L^*-mer de Bruijn graph of p^* is not a tree.

Proposition B.3.5. *Say $p^* \in \mathcal{M}_{L^*}$ and for each L, consider a $\text{Dirichlet}(\alpha_{k,b})_{b \in \tilde{\mathcal{B}}}$ prior on the*

simplex in $\Delta^{\mathcal{B}_L^o}_{\tilde{\mathcal{B}}}$ corresponding to the L-mer k. Say for $L \geq L^*$, for all L-mers k and bases b, $\alpha_{k,b} = \alpha_{k_{L-L^*+1:L},b}$ (i.e. the prior concentration depends only on the last L^* letters of the L-mer). There exists a $\tilde{L}$ (possibly infinity) such that for all $L \geq L^*$, the L-mer de Bruijn graph is a tree if and only if $L > \tilde{L}$. Then $(\dim_L^{\text{eff}}(p^*))_{L \geq L^*}$ is a non-decreasing sequence, strictly increasing until $\tilde{L}$, and constant past $\tilde{L}$.

Proof. Call v^* the transition coefficients of p^*. Say $L > L^*$, $k \in \text{acc}_L(p^*)$. Call $k' \in \text{acc}_{L^*}(p^*)$ the last L^* letters of k. If for some $b \in \tilde{\mathcal{B}}$, $p^*(\#(k,b) > 0) > 0$ then clearly $p^*(\#(k',b) > 0)$ thus $\text{supp}_L(p^*)|_k \subseteq \text{supp}_{L^*}(p^*)|_{k'}$. On the other hand, say $b \in \text{supp}_{L^*}(p^*)|_{k'} = \text{supp}(v^*)|_{k'}$ and Y is a string, not terminated with \$, and with its last L characters equal to k and $p^*(Y \dots)$. $p^*((Y,b)\dots \mid Y \dots) = v^*_{k',b} > 0$ so, $p^*(\#(k,b) > 0) > 0$. Thus $\text{supp}_L(p^*)|_k = \text{supp}_{L^*}(p^*)|_{k'}$.

Now write

$$\dim_L^{\text{eff}}(p^*) = \sum_{k \in \text{acc}_L(p^*)} \sum_{b \in \text{supp}_L(p^*)|_k} \left[\mathbb{1}_{b \in \text{supp}_L(p^*)|k} + \mathbb{1}_{b \notin \text{supp}_L(p^*)|k} \alpha_{k,b}\right] - 1$$

where, for a statement A, $\mathbb{1}_A = 1$ if A is true and $\mathbb{1}_A = 0$ if A is false. Thus, since in this case $\text{supp}_L(p^*)|_k = \text{supp}_{L^*}(p^*)|_{k'}$, and by the assumption on the prior coefficients,

$$\begin{aligned} \dim_L^{\text{eff}}(p^*) = &\sum_{k' \in \text{acc}_{L^*}(p^*)} \left|\{k \in \text{acc}_{L^*}(p^*) \mid k_{L-L^*+1:L} = k'\}\right| \\ &\times \left(\sum_{b \in \text{supp}_L(p^*)|'_k} \left[\mathbb{1}_{b \in \text{supp}_L(p^*)|k'} + \mathbb{1}_{b \notin \text{supp}_L(p^*)|k'} \alpha_{k',b}\right] - 1 \right). \end{aligned} \tag{B.10}$$

Since for each $k' \in \text{acc}_{L^*}(p^*)$ there is a $k \in \text{acc}_L(p^*)$ that has its last L^* letters equal to k',

$\dim_L^{\text{eff}}(p^*) \geq \dim_{L^*}^{\text{eff}}(p^*)$. Since $p^* \in \mathcal{M}_L$ for all $L \geq L^*$ the argument may be repeated for all pairs $L_1 > L_2 \geq L^*$ to conclude $(\dim_L^{\text{eff}}(p^*))_{L \geq L^*}$ is non-decreasing.

Note if for $L' > L$, $\dim_{L'}^{\text{eff}}(p^*) = \dim_L^{\text{eff}}(p^*)$ then for all $k' \in \text{acc}_L(p^*)$ there is a unique $k \in \text{acc}_{L'}(p^*)$ with its last L letters equal to k. Thus if $X_1, X_2 \in S$ with $p^*(X_1), p^*(X_2) > 0$ and X_1, X_2 end in the same last L letters (not including \$), then X_1, X_2 end in the same last L' letters. Looking at positions $|X_j| - L' : |X_j| - L' + L - 1$, one can also conclude that X_1, X_2 end in the same last $L' + (L' - L)$ letters. Continuing, one may conclude $X_1 = X_2$. It can be seen that this is equivalent to the L-mer de Bruijn of p^* being a tree. On the other hand it is not difficult to see that if the L-mer de Bruijn of p^* is a tree then $\dim_{L'}^{\text{eff}}(p^*) = \dim_L^{\text{eff}}(p^*)$ for all $L' > L$. □

Corollary B.3.6. *Say $p^* \in \mathcal{M}$ and L^* is the minimum lag such that $p^* \in \mathcal{M}_{L^*}$. Let $\pi(L)$ be a prior over lags with $\pi(L) > 0$ for all L less than some $\tilde{L} \geq L^*$, and with $\pi(L) = 0$ for all $L > \tilde{L}$. For each L, consider a* $\text{Dirichlet}(\alpha_{k,b})_{b \in \tilde{\mathcal{B}}}$ *prior on the simplex in $\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$ corresponding to the L-mer k. Assume that for $L \geq L^*$, for all L-mers k and bases b, $\alpha_{k,b} = \alpha_{k_{L-L^*+1:L},b}$. Then lag selection is consistent if and only if the L^*-mer de Bruijn graph of p^* is not a tree.*

Remark B.3.7. *If $p^*(X) > 0$ for infinitely many $X \in S$, as is the case if the transition coefficients of p^* are all positive or there is a cycle in the L^*-mer de Bruijn graph of p^*, then no L-mer de Bruijn graph of p^* is a tree as sequences with $p(X) > 0$ cannot be identified by their last L letters. As another example, pick a particular sequence $X \in S$ and say X' is one letter away from X. For a $0 < q < 1$, define $p = q\delta_X + (1-q)\delta_{X'}$. Pick L^* the smallest lag such that $p^* \in \mathcal{M}_{L^*}$. Then the L^*-mer de Bruijn graph splits into two paths at the position where X and X' differ. These paths may rejoin*

after L^* nodes. Thus the L^*-mer de Bruijn graph is a tree if and only if the position at which X and X' differ is less than L^* letters away from the end symbol \$.

B.4 Misspecification detection

In this section, we turn from studying the parameter v and lag L in the BEAR model to studying the hyperparameters h and θ. Intuitively, we expect the empirical Bayes estimate of h to behave as a diagnostic of misspecification, since h controls the extent to which the prior predictive distribution of the BEAR model is concentrated at the embedded AR model. Here we make this idea rigorous by examining the asymptotic behavior of the empirical Bayes estimates of h and θ.

We first briefly introduce the setup and some notation. We will assume p^* is subexponential. We will work with fixed lag L, though the results can be straightforwardly extended to the case of a prior over a finite number of lags. The function $f : \Theta \mapsto \Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$ defines an autoregressive model, with parameter space Θ some set. For any $h > 0, \theta \in \Theta$, define a prior $\pi(\cdot|h,\theta)$ on $\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o}$ consisting of independent Dirichlet$(\frac{1}{h} f_{k,b}(\theta))_{b\in\tilde{\mathcal{B}}}$ priors on each simplex corresponding to $k \in \mathcal{B}_L^o$. Define $m((X_n)_{n=1}^N|h,\theta)$ to be the marginal likelihood of the data $(X_n)_{n=1}^N$ under the prior $\pi(\cdot|h,\theta)$, that is $m((X_n)_{n=1}^N|h,\theta) = \int p_v((X_n)_{n=1}^N)\pi(v|h,\theta)$. For our purposes we may assume $f_{k,b}(\theta) > 0$ for all $(k,b) \in \text{supp}_L(p^*)$; if this is not the case for some θ then the marginal likelihood at θ, for any choice of h, is a.s. eventually 0. We will study maximum marginal likelihood/empirical Bayes estimates $(h_N,\theta_N) = \text{argmax}_{h,\theta} m((X_n)_{n=1}^N|h,\theta)$.

Our starting point is the analysis of empirical Bayes presented in Petrone et al. [196]. Here is the

(very heuristic) intuition behind their result: the Laplace approximation to the marginal likelihood is proportional to the probability of the true data-generating parameter under the prior, so asymptotically we expect $m((X_n)_{n=1}^N|h,\theta) \propto \pi(v^*|h,\theta)$. Then, roughly speaking, the empirical Bayes estimate will be $(h_N,\theta_N) \approx \text{argmax}_{h,\theta}\pi(v^*|h,\theta)$; in other words, the empirical Bayes estimate should asymptotically maximize the probability of the true parameter parameter value under the prior. Petrone et al. [196] give conditions under which this is indeed true, but BEAR models fail to meet them. There are two major problems: (1) in the limit as $h \to 0$, the prior converges to a point mass, making the Laplace approximation invalid (the "degenerate" case mentioned by Petrone et al. [196]) and (2) when some transitions have probability zero, $v^*_{k,b} = 0$, the standard Laplace approximation does not hold regardless of the value of h. Our analysis in this section adjusts for both these issues, and also provides more detailed insight such as convergence rates and intuitive approximations for the optimal h.

In analyzing extremum estimators, such as the maximum marginal likelihood estimator used in empirical Bayes, uniform convergence results are particularly powerful. Ideally, we might try to establish a Laplace-like approximation to the marginal likelihood that holds uniformly for all h and θ, but this is unavailable because of the degeneracy at $h = 0$. Our strategy will be to first demonstrate a uniform Laplace approximation over all h,θ with some caveats: (1) we ignore transitions that are not possible under p^* and analyze their contribution to the likelihood later; (2) if $h \to 0$ we assume it does not decrease too fast; and (3) we assume similar control over the prior density at the "true" transition probabilities v^*. In proposition B.4.3 we prove that (3) must indeed hold for when h_N,θ_N are the maximizers of the marginal likelihood.

For any $v \in \tilde{\Delta}_L(p^*)$, define the negative average log likelihood $l_N(v) = -\frac{1}{N}\log p_v(X_n)_{n=1}^N$, and let $v_N \in \tilde{\Delta}_L(p^*)$ be the (a.s. eventually unique) maximizer of l_N. Define a prior $\tilde{\pi}(\cdot|h,\theta)$ on $\tilde{\Delta}_L(p^*)$ consisting of independent Dirichlet$(\frac{1}{h}f_{k,b}(\theta))_{b\in \mathrm{supp}_L(p^*)|_k}$ priors on each simplex corresponding to $k \in \mathrm{acc}_L(p^*)$ (for a scalar α, Dirichlet(α) is just defined as the point mass on the 0-dimensional simplex $\{1\}$). Let $\tilde{m}((X_n)_{n=1}^N|h,\theta)$ denote the marginal likelihood under the prior $\tilde{\pi}(\cdot|h,\theta)$ and define

$$\log r_N(h,\theta) = \sum_{k\in \mathrm{acc}_L(p^*)} \left(\log \frac{\Gamma(\sum_b \frac{1}{h} f_{k,b}(\theta))}{\Gamma(\sum_b \frac{1}{h} f_{k,b}(\theta) + \#k)} - \log \frac{\Gamma(\sum_b' \frac{1}{h} f_{k,b}(\theta))}{\Gamma(\sum_b' \frac{1}{h} f_{k,b}(\theta) + \#k)} \right)$$

where $\sum_b'$ is a sum over the $b \in \mathrm{supp}_L(p^*)|_k$. So, as shown in theorem B.3.2, $\log m((X_n)_{n=1}^N|h,\theta) = \log \tilde{m}((X_n)_{n=1}^N|h,\theta) + \log r_N(h,\theta)$. define $B(v,\eta)$ to be the ball of radius η around v in some norm; finally, define $B_{\mathrm{KL}}(\eta) = \{v \in \tilde{\Delta}_L(p^*) \mid \mathbb{E}\log \frac{p^{*(L)}(X)}{p_v(X)} < \eta\}$ and, for convenience $B(\eta) = B(v^*,\eta)$, for any $\eta > 0$.

Theorem B.4.1. *With probability* 1*, for any sequence* $(h_N)_N$ *and* $(\theta_N)_N$*, possibly dependent on the data, if* $h_N N^{1/4-\epsilon} \to \infty$ *for an* $1/4 > \epsilon > 0$ *and* $\liminf(\log \tilde{\pi}(v^*|h_N,\theta_N))/\sqrt{N} \neq -\infty$*, then*

$$\left| \log \tilde{m}((X_n)_{n=1}^N|h_N,\theta_N) - \left(-N l_N(v_N) - \frac{1}{2}\dim \tilde{\Delta}_L(p^*)\log N + \log \tilde{\pi}(v^*|h_N,\theta_N) + C_{v^*} \right) \right| \to 0$$

for a fixed C_{v^*} *dependent only on* v^**.*

Proof. First note, calling $e_{k,b}$ the indicator vector at position k,b for some $k \in \mathrm{acc}_L(p^*), b, b' \in$

$\text{supp}_L(p^*)|_k$, the directional derivatives with respect to v

$$D_{e_{k,b}-e_{k,b'}} \log \tilde{\pi}(v|h,\theta) = \frac{\frac{1}{h} f_{k,b}(\theta) - 1}{v_{k,b}} - \frac{\frac{1}{h} f_{k,b'}(\theta) - 1}{v_{k,b'}}$$

are bounded by J/h, for some $J > 0$ in a neighborhood of v^* for all θ.

For an $\eta > 0$, define the KL ball

$$\hat{B}_{\text{KL}}(\eta) = \{v \in \tilde{\Delta}_L(p^*) \mid v_{k,b} \geq v^*_{k,b}(1 - \eta/\mathbb{E}|X|) \; \forall k, b\}.$$

Note if $v \in \hat{B}_{\text{KL}}(\eta)$, then the KL divergence is bounded,

$$\mathbb{E} \log \frac{p^{*(L)}(X)}{p_v(X)} \leq (\mathbb{E}|X|) \sup_{k,b} \log \frac{v^*_{k,b}}{v_{k,b}} \leq \eta$$

so $v \in B_{\text{KL}}(\eta)$. Note

$$(w_{k,b})_{(k,b)\in\text{supp}_L(p^*)} \mapsto \left(\frac{\eta}{\mathbb{E}|X|} w_{k,b} + v^*_{k,b}\left(1 - \frac{\eta}{\mathbb{E}|X|}\right)\right)_{(k,b)\in\text{supp}_L(p^*)}$$

is a diffeomorphism from $\tilde{\Delta}_L(p^*)$ to $\hat{B}_{\text{KL}}$ so by the change of variables theorem the volume of $\hat{B}_{\text{KL}}$ is $(\eta/\mathbb{E}|X|)^{\dim \tilde{\Delta}_L(p^*)}$ (which comes from the factor multiplying $w_{k,b}$) times the volume of $\tilde{\Delta}_L(p^*)$. Finally note that by an application of the triangle inequality,

$\hat{B}_{\text{KL}}(\eta) \subset B(2\eta \text{diam}(\tilde{\Delta}_L(p^*))/\mathbb{E}|X|)$.

Define the information matrix at $\tilde{v}^*, \mathcal{I} = \mathbb{E}[D^2 l_1(\tilde{v}^*)]$, and an $\epsilon' > 0$ less than the smallest

eigenvalue of $\mathcal{I}$ ($\mathcal{I}$ is positive definite by the strict convexity of l_0 described in theorem B.3.2). Also pick an $\epsilon'' < \frac{1}{8}\epsilon'$ such that $\hat{B}_{\text{KL}}(\epsilon''\eta^2) \subset B(\eta)$ for all small η. Now define a sequence $\eta_N =$

$N^{-(1/4-\epsilon)}$ noting $\eta_N/h_N \to 0$. Let $|\mathcal{I}|$ denote the determinant of the information matrix.

$$
\begin{aligned}
&\Bigg| \log \tilde{m}((X_n)_{n=1}^N | h_N, \theta_N) \\
&\quad - \left(-N l_N(v_N) - \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log(2\pi N) - \frac{1}{2} \log |\mathcal{I}| + \log \tilde{\pi}(v^* | h_N, \theta_N) \right) \Bigg| \\
&\leq \left| \log \left(\int_{\tilde{\Delta}_L(p^*)} e^{-N l_N(v)} \tilde{\pi}(v | h_N, \theta_N) \right) - \log \left(\int_{B(\eta_N)} e^{-N l_N(v)} \tilde{\pi}(v | h_N, \theta_N) \right) \right| \\
&\quad + \left| \log \left(\int_{B(\eta_N)} e^{-N l_N(v)} \tilde{\pi}(v | h_N, \theta_N) \right) - \log \left(\int_{B(\eta_N)} e^{-N l_N(v)} \tilde{\pi}(v^* | h_N, \theta_N) \right) \right| \\
&\quad + \left| \log \left(\int_{B(\eta_N)} e^{-N l_N(v)} \tilde{\pi}(v^* | h_N, \theta_N) \right) - \log \left(\int_{B(v_N, \eta_N)} e^{-N l_N(v)} \tilde{\pi}(v^* | h_N, \theta_N) \right) \right| \\
&\quad + \Bigg| \log \left(\int_{B(v_N, \eta_N)} e^{-N l_N(v)} \tilde{\pi}(v^* | h_N, \theta_N) \right) \\
&\qquad - \left(-N l_N(v_N) - \frac{1}{2} \dim \hat{\Delta}_L(p^*) \log(2\pi N) - \frac{1}{2} \log |\mathcal{I}| + \log \tilde{\pi}(v^* | h_N, \theta_N) \right) \Bigg| \\
&\leq \log \left(1 + \frac{\int_{\tilde{\Delta}_L(p^*) \setminus B(\eta_N)} e^{N l_N(v_N) - N l_N(v)} \tilde{\pi}(v | h_N, \theta_N)}{\int_{B(\eta_N)} e^{N l_N(v_N) - N l_N(v)} \tilde{\pi}(v | h_N, \theta_N)} \right) \\
&\quad + \left| \log \left(\left(\int_{B(\eta_N)} e^{-N l_N(v)} \frac{\tilde{\pi}(v | h_N, \theta_N)}{\tilde{\pi}(v^* | h_N, \theta_N)} \right) / \left(\int_{B(\eta_N)} e^{-N l_N(v)} \right) \right) \right| \\
&\quad + \log \left(\left(\int_{B(v_N, \eta_N + \|v_N - v^*\|)} e^{-N l_N(v)} \right) / \left(\int_{B(v_N, \eta_N - \|v_N - v^*\|)} e^{-N l_N(v)} \right) \right) \\
&\quad + \left| \log \left((2\pi)^{-\frac{1}{2} \dim \hat{\Delta}_l(p^*)} |\mathcal{I}|^{-1/2} \int_{\|y\| < \eta_N \sqrt{N}} e^{N(l_N(v_N) - l_N(v_N + y/\sqrt{N}))} \right) \right| \\
&\leq \exp \left(N \sup_{\|v^* - v\| > \eta_N} (l_N(v^*) - l_N(v)) \right) / \left(\int_{\hat{B}_{\mathrm{KL}}(\epsilon'' \eta_N^2)} e^{N l_N(v^*) - N l_N(v)} \tilde{\pi}(v | h_N, \theta_N) \right) \\
&\quad + \sup_{v \in B(\eta_N)} |\log \tilde{\pi}(v | h_N, \theta_N) - \log \tilde{\pi}(v^* | h_N, \theta_N)| \\
&\quad + \left(\int_{B(v_N, \eta_N + \|v_N - v^*\|) \setminus B(v_N, \eta_N - \|v_N - v^*\|)} e^{-N l_N(v)} \right) / \left(\int_{B(v_N, \eta_N - \|v_N - v^*\|)} e^{-N l_N(v)} \right) \\
&\quad + \left| \log \left((2\pi)^{-\frac{1}{2} \dim \hat{\Delta}_l(p^*)} |\mathcal{I}|^{-1/2} \int_{\|y\| < \eta_N \sqrt{N}} e^{N(l_N(v_N) - l_N(v_N + y/\sqrt{N}))} \right) \right| .
\end{aligned}
\tag{B.11}
$$

The third line in this inequality follows since $B(v_N, \eta_N - \|v_N - v^*\|) \subseteq B(v_N, \eta_N) \cap B(\eta_N)$ and $B(v_N, \eta_N) \cup B(\eta_N) \subseteq B(v_N, \eta_N + \|v_N - v^*\|)$. First note that the second term is bounded by $J\eta_N/h_N$ and thus vanishes a.s.. We will show the rest of these terms also vanish a.s..

To analyze the last term, we will use a simplified proof of a Laplace approximation. First note, given the regularity conditions established in the proof of theorem B.3.2, a.s. $v_N \to v^*$, and $D^2 l_N \to D^2 E l_N$ locally uniformly. Thus, for each y, since $\eta_N\sqrt{N} \to \infty$, and and $\eta_N \to 0$ (so that if $\|y\| < \eta_N\sqrt{N}$ then $y/\sqrt{N} \leq \eta_N \to 0$), a.s.

$$\mathbb{1}_{\|y\|<\eta_N\sqrt{N}} e^{N(l_N(v_N) - l_N(v_N + y/\sqrt{N}))} = \mathbb{1}_{\|y\|<\eta_N\sqrt{N}} e^{-\frac{1}{2} y^T D^2 l_N(v'_N) y} \to e^{-\frac{1}{2} y^T \mathcal{I} y},$$

where v'_N is on a ray connecting v_N to $v_N + y/\sqrt{N}$. As well, eventually,

$$\mathbb{1}_{\|y\|<\eta_N\sqrt{N}} e^{N(l_N(v_N) - l_N(v_N + y/\sqrt{N}))} = \mathbb{1}_{\|y\|<\eta_N\sqrt{N}} e^{-\frac{1}{2} y^T D^2 l_N(v'_N) y} \leq e^{-\frac{1}{4} y^T \mathcal{I} y}.$$

The right hand side is integrable and takes the form of a Gaussian pdf. Thus, integrating the Gaussian pdf, the last term of equation B.11 goes to 0 a.s. by the dominated convergence theorem.

To analyze the third term of equation B.11, recall from the proof of B.3.2 that l_N is convex, so, the value of $-N l_N$ is less on the annulus $B(v_N, \eta_N + \|v_N - v^*\|) \setminus B(v_N, \eta_N - \|v_N - v^*\|)$ than on $B(v_N, \eta_N - \|v_N - v^*\|)$. Thus, to demonstrate that this term vanishes, it suffices to show that $\|v_N - v^*\|/\eta_N \to 0$ a.s.. Recall from the proof of B.3.2 that we showed that a.s. $v_N \to v^*$ and $D^2 l_N$ converges to $\mathbb{E} D^2 l_1$ uniformly in a neighborhood of v^*. Thus, eventually, recalling the

definition of ϵ' as less than the minimal eigenvalue of $\mathcal{I}$, and defining $t \mapsto v_t$ as a linear path from v_N to v^*,

$$\|Dl_N(v^*)\| = \|Dl_N(v^*) - Dl_N(v_N)\| = \left\| \left(\int_0^1 dt D^2 l_N(v_t) \right) (v^* - v_N) \right\| \geq \frac{1}{2}\epsilon' \|v^* - v_N\|.$$

On the other hand, defining $e_{k,b}$ as above, $|D_{e_{k,b}-e_{k,b'}} l_1(v^*)| \leq |X|/\inf_{k,b} v^*_{k,b}$ and so, $D_{e_{k,b}-e_{k,b'}} l_1(v^*)$ is subexponential. Recalling $\mathbb{E}Dl_1(v^*) = D\mathbb{E}l_1(v^*) = 0$, using Bernstein's inequality (theorem 2.8.1 in Vershynin [274]),

$$p^*(|D_{e_{k,b}-e_{k,b'}} l_N(v^*)| > \eta_N^2) \leq C \exp(-C' N \eta_N^4) \leq C \exp(-C' N^{4\epsilon}).$$

Since $\sum_{N=1}^{\infty} C \exp(-C' N^{4\epsilon}) \lesssim \int_0^{\infty} dx \exp\left(-C' x^{4\epsilon}\right) < \infty$, by the Borel-Cantelli lemma, a.s. eventually, $\|Dl_N(v^*)\| \leq C\eta_N^2$ for some $C > 0$. Finally, since $\eta_N \to 0$, we have $\|v_N - v^*\|/\eta_N \to 0$ a.s..

To analyze the first term of equation B.11 first note that for small enough η_N, recalling that $\mathbb{E}l_N$ is convex with maximum at v^*, and by the definition of ϵ', we can Taylor expand around v^* and find

$$\sup_{\|v^* - v\| > \eta_N} (\mathbb{E}l_N(v^*) - \mathbb{E}l_N(v)) = \sup_{\|v^* - v\| = \eta_N} (\mathbb{E}l_N(v^*) - \mathbb{E}l_N(v)) \leq -1/2\epsilon' \eta_N^2.$$

We will also show below that a.s. eventually, for all v away from the boundary (i.e. outside a fixed neighborhood of the boundary), $|l_N(v) - \mathbb{E}l_N(v)| < \frac{1}{16}\epsilon'\eta_N^2$. For now, assume that this is the case. So, a.s. eventually, $\sup_{\|v^* - v\| > \eta_N} (l_N(v^*) - l_N(v)) < -3/8\epsilon'\eta_N^2$, by the triangle inequality.

Having bounded the numerator, we now turn to the denominator. Note that by equi-continuity, since $J\eta_N/h_N$ is eventually less than $\log 2$, $\tilde{\pi}(v|h_N,\theta_N) \geq \frac{1}{2}\tilde{\pi}(v^*|h_N,\theta_N)$ for all $v \in B(\eta_N)$. As well, again, by a triangle inequality, a.s. eventually, for all $v \in B_{\text{KL}}(\epsilon''\eta_N^2)$, $l_N(v^*) - l_N(v) \geq -\epsilon''\eta_N^2 - \frac{1}{8}\epsilon'\eta_N^2 \geq -\frac{1}{4}\epsilon'\eta_N^2$. Recall that the volume of $\hat{B}_{\text{KL}}(\epsilon''\eta_N^2)$ is equal to $C(C'\eta_N^2)^{\dim \hat{\Delta}_L(p^*)}$ for some $C, C' > 0$. Then the first term of equation B.11 is bounded above by

$$2C \exp\left(-\frac{1}{8}\epsilon' N\eta_N^2 + 2 \dim \tilde{\Delta}_L(p^*) \log\left(\eta_N^{-1}\right) - \log \tilde{\pi}(v^*|h_N,\theta_N)\right)$$

for some $C > 0$. This expression goes to 0 as $\log \tilde{\pi}(v^*|h_N,\theta_N)/\sqrt{N}$ is bounded below and thus $\liminf \log \tilde{\pi}(v^*|h_N,\theta_N)/N^{1/2+2\epsilon} = 0$.

We now show that a.s. eventually, for all v away from the boundary, $|l_N(v) - \mathbb{E}l_N(v)| < \frac{1}{16}\epsilon'\eta_N^2$. First write

$$D_{e_{k,b}-e_{k,b'}} l_N(v) = \frac{1}{N}\#(k,b)v_{k,b}^{-1} - \frac{1}{N}\#(k,b')v_{k,b'}^{-1}$$

which is almost surely eventually bounded by the strong law of large numbers for all v away from the boundary of $\tilde{\Delta}_L(p^*)$. The derivatives of $\mathbb{E}l_N$ with respect to v are similarly bounded away from the boundary; say the derivatives of both functions are eventually bounded by J'. Also note that the random variables $|l_1(v)(X)| \leq C''|X|$ are uniformly sub-exponential for all v away from the boundary. The covering number of $\tilde{\Delta}_L(p^*)$ by balls of radius $\frac{1}{64}J'^{-1}\epsilon'\eta_N^2$ is $\lesssim \eta_N^{-2\dim \hat{\Delta}_L(p^*)}$. Say $(v_i)_i$ are centers of the balls of such a covering. By uniform sub-exponentiality and Bernstein's inequality (theorem 2.8.1 in Vershynin [274]), for small enough η_N, $P(|l_N(v_i) - \mathbb{E}l_N(v_i)| >$

$\frac{1}{32}\epsilon'\eta_N^2) \lesssim \exp(-CN\eta_N^4) = \exp(-CN^{4\epsilon})$ for some $C > 0$. Now, for some $C, C' > 0$,

$$\begin{aligned}
&\sum_{N=0}^{\infty} P(\text{there is a } v_i \text{ such that } |l_N(v_i) - \mathbb{E}l_N(v_i)| > \frac{1}{32}\epsilon'\eta_N^2) \\
&\qquad \leq \sum_{N=0}^{\infty} \sum_i P(|l_N(v_i) - \mathbb{E}l_N(v_i)| > \frac{1}{32}\epsilon'\eta_N^2) \\
&\qquad \lesssim \sum_{N=0}^{\infty} \exp\left(-CN^{4\epsilon} - 2\dim \hat{\Delta}_L(p^*)\log\eta_N\right) \qquad \text{(B.12)} \\
&\qquad \lesssim \sum_{N=0}^{\infty} \exp\left(-C'N^{4\epsilon}\right) \\
&\qquad \lesssim \int_0^{\infty} dx \exp\left(-C'x^{4\epsilon}\right) < \infty.
\end{aligned}$$

By the Borel-Cantelli lemma, $|l_N(v_i) - \mathbb{E}l_N(v_i)| \leq \frac{1}{32}\epsilon'\eta_N^2$ for all i a.s. eventually. Thus, eventually, by the triangle inequality and the a.s. eventual boundedness of the derivatives of l_N and $\mathbb{E}l_N$, $|l_N(v) - \mathbb{E}l_N(v)| \leq \frac{1}{16}\epsilon'\eta_N^2$ for all v away from the boundary a.s. eventually. □

We now focus on the behavior of not just any sequence of h_N, θ_N, but rather specifically on h_N, θ_N which maximize the marginal likelihood.† The next two results both use a proof by contradiction strategy that relies on the following logic.

Remark B.4.2. *Fix h, θ. We showed in theorem B.3.2 that $\log r_N(h,\theta) = O(\log N)$ a.s. and we can conclude from theorem B.4.1 that $\log \tilde{m}((X_n)_{n=1}^N \mid h, \theta) = -Nl_N(v_N) - O(\log(N))$. Thus, $m((X_n)_{n=1}^N \mid h, \theta) = -Nl_N(v_N) - O(\log(N))$. On the other hand, for any h', θ', $\log r_N(h', \theta') \leq 0$ and $\log \tilde{m}((X_n)_{n=1}^N \mid h', \theta') \leq -Nl_N(v_N)$. Thus for the maximizers of*

†It is not crucial that maximizers of the marginal likelihood exist for any of the result below: the results below hold assuming only that h_N, θ_N are approximate maximizers, i.e. $\log m((X_n)_{n=1}^N \mid h_N, \theta_N) = \sup_{h,\theta} \log m((X_n)_{n=1}^N \mid h, \theta) + o(1)$ or in slightly altered form swapping the $o(1)$ for $o_P(1)$.

m, h_N, θ_N, *it is a contradiction if* $\log r_N(h_N, \theta_N) \lesssim -N^\beta$ *or* $\log \tilde{m}((X_n)_{n=1}^N \mid h_N, \theta_N) \leq -Nl_N(v_N) - CN^\beta$ *for any* $\beta > 0$*: say* $\log \tilde{m}(h_N, \theta_N) \leq -Nl_N(v_N) - N^\beta$*. Then, for some* $C > 0$, $-C\log(N) \leq m((X_n)_{n=1}^N | h, \theta) + Nl_N(v_N) \leq m((X_n)_{n=1}^N | h_N, \theta_N) + Nl_N(v_N) \leq \log \tilde{m}((X_n)_{n=1}^N | h_N, \theta_N) + Nl_N(v_N) \leq -CN^\beta$*, a contradiction. On the other hand, say* $\log r_N(h_N, \theta_N) \lesssim -N^\beta$*. Then* $-C\log(N) \leq m((X_n)_{n=1}^N | h, \theta) + Nl_N(v_N) \leq m((X_n)_{n=1}^N | h_N, \theta_N) + Nl_N(v_N) \leq \log r_N(h_N, \theta_N) \leq -C'N^\beta$*, also a contradiction.*

Proposition B.4.3. *Say* $(h_N)_N$ *and* $(\theta_N)_N$ *are sequences maximizing* $\log m((X_n)_{n=1}^N | h_N, \theta_N)$ *for each* N*. Then a.s. there is no subsequence* $(h_{N_j})_j$ *and* $(\theta_{N_j})_j$ *such that for some* $\epsilon > 0$*,* $h_{N_j} N_j^{1/4-\epsilon} \to \infty$ *and for some* $\beta > 0$, $\lim \log \tilde{\pi}(v^* | h_{N_j}, \theta_{N_j}) / N_j^\beta < 0$.

Proof. Assume the opposite. Define $(v_N)_N$ and pick $(\eta_N)_N, \epsilon'$ as in theorem B.4.1 such that a.s. eventually, for all v away from the boundary, $|l_N(v) - \mathbb{E}l_N(v)| < \frac{1}{16}\epsilon'\eta_N^2, \eta_{N_j}/h_{N_j} \to 0$, and $\inf_{\|v^* - v\| > \eta_N} \mathbb{E}l_N(v) \geq \mathbb{E}l_N(v_N) + \frac{1}{2}\epsilon'\eta_N^2$. Then, eventually,

$$\begin{aligned} \int_{B(\eta_{N_j})^C} e^{-N_j l_{N_j}(v)} \tilde{\pi}(v | h_{N_j}, \theta_{N_j}) &\leq \exp\left(-N_j \inf_{\|v^* - v\| > \epsilon} l_{N_j}(v)\right) \\ &\leq \exp\left(-N_j (l_{N_j}(v_{N_j}) + \frac{3}{8}\epsilon'\eta_{N_j}^2)\right) \\ &\leq \exp\left(-N_j l_{N_j}(v_{N_j}) - \frac{3}{8}\epsilon' {N_j}^{1/4}\right). \end{aligned} \tag{B.13}$$

where $B(\eta_{N_j})^C$ denotes the complement of $B(\eta_{N_j})$. On the other hand, by equi-continuity of the

prior density, since η_{N_j}/h_{N_j} becomes small, for some $C > 0$

$$\begin{aligned}\int_{B(\eta_{N_j})} e^{-N_j l_{N_j}(v)} \tilde{\pi}(v|h_{N_j}, \theta_{N_j}) &\lesssim \exp\big(- N_j l_{N_j}(v_{N_j}) + \log \tilde{\pi}(v^*|h_{N_j}, \theta_{N_j}) \\ &\qquad + \dim \tilde{\Delta}_L(p^*) \log(\eta_{N_j})\big) \\ &\leq \exp\Big(-N_j l_{N_j}(v_{N_j}) - C{N_j}^{\beta} + O(\log N_j))\Big)\end{aligned} \tag{B.14}$$

for some $C > 0$. By remark B.4.2, this completes the proof. □

We have so far explored what happens to the marginal likelihood when h_N does not converge quickly to 0, showing that it satisfies a Laplace-like approximation in this case. Next we show that h_N will in fact converge to zero quickly only if the estimated autoregressive model $f(\theta_N)$ converges to the optimal parameter value v^*.

For a sequence $(\theta_N)_N$ define, for $k \in \mathrm{acc}_L(p^*)$, $\sigma_{N,k} = \sum_{b \in \mathrm{supp}_L(p^*)|_k} f_{k,b}(\theta_N)$ and $\lambda_{N,k} = 1 - \sigma_{N,k}$.

Proposition B.4.4. *Say $(h_N)_N$ and $(\theta_N)_N$ are sequences maximizing* $\log m(\{X_n\}_{n=1}^N | h_N, \theta_N)$. *Then a.s.,* $\limsup h_{N_j} N_j^{\beta} < \infty$ *for some $\beta > 0$ along a subsequence $(N_j)_j$ only if $f_{k,b}(\theta_{N_j}) \to v^*_{k,b}$ for all $k, b \in \mathrm{supp}_L(p^*)$.*

Proof. Take a subsequence such that: $h_{N_j} \to 0$; $h_{N_j} N_j^{\beta}$ and $h_{N_j} N_j$ both converge, the latter possibly to ∞; $f_{k,b}(\theta_{N_j})$ converges for all k, b; and $f_{k,b}(\theta_{N_j})/h_{N_j}$ converges, possibly to ∞, for all k, b. Note since $[0, \infty]$ is compact, every subsequence with $\limsup h_{N_j} N_j^{\beta} < \infty$ has a further subsequence with these properties. Thus it will be sufficient to show that $f_{k,b}(\theta_{N_j}) \to v^*_{k,b}$ for all

$k \in \text{acc}_L(p^*), b \in \tilde{\mathcal{B}}$. Now define $\lambda_k = \lim \lambda_{N_j,k}$ and σ_k similarly for all $k \in \text{acc}_L(p^*)$.

The proof will proceed in two parts. First we will show that if $\lambda_k \neq 0$ for some $k \in \text{acc}(p^*)$, then $\log r_{N_j}(h_{N_j}, \theta_{N_j}) \lesssim -N_j^{\beta'}$ for some $\beta' > 0$. This is a contradiction by remark B.4.2 so that $\lambda_k = 0$ and $\sigma_k = 1$ for all k. Then we will show that if $f_{k,b}(\theta_{N_j}) \not\to v^*_{k,b}$ for any $k, b \in \text{supp}_L(p^*)$, eventually $\sup_{v \in B(\eta)} \log \tilde{\pi}(v|h_{N_j}, \theta_{N_j}) \lesssim -N_j^{\beta''} \left(\|f(\theta_{N_j}) - v^*\| - \eta\right)^2$. for some $\beta'' > 0$ for small η. Assume this is the case for now. By similar logic to that in equation B.13 of proposition B.4.3, for small fixed η, it can be seen that for some $\beta''', C, C' > 0$,

$$\log \int_{B(\eta)^C} e^{-N_j l_{N_j}(v)} \tilde{\pi}(v|h_{N_j}, \theta_{N_j}) \leq -N_j l_{N_j}(v_{N_j}) - CN^{\beta'''}.$$

As well,

$$\begin{aligned}\log \int_{B(\eta)} e^{-N_j l_{N_j}(v)} \tilde{\pi}(v|h_{N_j}, \theta_{N_j}) &\leq -N_j l_{N_j}(v_{N_j}) + \sup_{\|v^* - v\| < \eta} \log \tilde{\pi}(v \mid h_{N_j}, \theta_{N_j}) \\ &\leq -N_j l_{N_j}(v_{N_j}) - C' {N_j}^{\beta''}.\end{aligned}$$

using the fact that $\log \tilde{\pi}(v|h_{N_j}, \theta_{N_j}) \lesssim -{N_j}^{\beta''}$. This is also a contradiction by remark B.4.2 and the statement of the theorem follows.

Part one: Assume that for some k', $\lambda_{k'} > 0$. Performing the Stirling approximation on the terms of $\log r_{N_j}$ depends on the behavior of $\sigma_{N_j,k}/h_{N_j}$. Based on the properties of the subsequence we chose, this quantity converges. If it converges to a number greater than or equal to 1 we can perform the usual Stirling approximation with $O(1)$ error. On the other hand, if $\sigma_{N_j,k}/h_{N_j}$ has limit less

than 1, using the properties of the Gamma function we write

$$\begin{aligned}\log\Gamma\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right) &= -\log\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right) + \log\Gamma\left(1+\frac{\sigma_{N_j,k}}{h_{N_j}}\right) \\ &= \left(\frac{\sigma_{N_j,k}}{h_{N_j}} - 1\right)\log\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right) - \frac{\sigma_{N_j,k}}{h_{N_j}} + O(1)\end{aligned} \tag{B.15}$$

where additional $O(1)$ terms were added explicitly in the second line so that the approximation is similar in form to the usual Stirling approximation with the exception of a 1 in the first term instead of $1/2$. Define $\gamma_k = 1/2$ if the limit of $\frac{\sigma_{N_j,k}}{h_{N_j}}$ is greater than or equal to 1 and 1 otherwise. Finally

recall that $h_{N_j} \to 0$ and write

$$
\begin{aligned}
\log r_{N_j}(h_{N_j}, \theta_{N_j}) &= \sum_{k \in \mathrm{acc}_L(p^*)} \left[\log \frac{\Gamma\left(\frac{1}{h_{N_j}}\right)}{\Gamma\left(\frac{1}{h_{N_j}} + \#k\right)} - \log \frac{\Gamma\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right)}{\Gamma\left(\frac{\sigma_{N_j,k}}{h_{N_j}} + \#k\right)} \right] \\
&= \sum_{k \in \mathrm{acc}_L(p^*)} \Bigg[\left(\frac{1}{h_{N_j}} - \frac{1}{2}\right) \log\left(\frac{1}{h_{N_j}}\right) \\
&\qquad - \left(\frac{1}{h_{N_j}} + \#k - \frac{1}{2}\right) \log\left(\frac{1}{h_{N_j}} + \#k\right) \\
&\qquad - \left(\frac{\sigma_{N_j,k}}{h_{N_j}} - \gamma_k\right) \log\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right) \\
&\qquad + \left(\frac{\sigma_{N_j,k}}{h_{N_j}} + \#k - \frac{1}{2}\right) \log\left(\frac{\sigma_{N_j,k}}{h_{N_j}} + \#k\right) \Bigg] + O(1) \\
&= \sum_{k \in \mathrm{acc}_L(p^*)} \Bigg[- \frac{\lambda_{N_j,k}}{h_{N_j}} \log(1 + h_{N_j} \#k) \\
&\qquad - \left(\frac{\sigma_{N_j,k}}{h_{N_j}} - \frac{1}{2}\right) \log\left(\sigma_{N_j,k}\right) \\
&\qquad + \left(\frac{\sigma_{N_j,k}}{h_{N_j}} + \#k - \frac{1}{2}\right) \log\left(\frac{\sigma_{N_j,k} + \#k h_{N_j}}{1 + \#k h_{N_j}}\right) \Bigg] \\
&\quad + \sum_{k \in \mathrm{acc}_L(p^*)} (\gamma_k - 1/2) \log\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right) + O(1) \\
&= \sum_{k \in \mathrm{acc}_L(p^*)} \frac{1}{h_{N_j}} \Bigg[- \lambda_{N_j,k} \log(1 + \#k h_{N_j}) - \sigma_{N_j,k} \log(\sigma_{N_j,k}) \\
&\qquad + \left(\sigma_{N_j,k} + \#k h_{N_j}\right) \log\left(\frac{\sigma_{N_j,k} + \#k h_{N_j}}{1 + \#k h_{N_j}}\right) \Bigg] \\
&\quad + \sum_{\sigma_{N_j,k}/h_{N_j} \to 0} (\gamma_k - 1/2) \log\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right) + O(1) \\
&\leq \sum_{k \in \mathrm{acc}_L(p^*)} \frac{1}{h_{N_j}} \Bigg[- \lambda_{N_j,k} \log(1 + \#k h_{N_j}) - \sigma_{N_j,k} \log(\sigma_{N_j,k}) \\
&\qquad + \left(\sigma_{N_j,k} + \#k h_{N_j}\right) \log\left(\frac{\sigma_{N_j,k} + \#k h_{N_j}}{1 + \#k h_{N_j}}\right) \Bigg] + O(1)
\end{aligned}
\tag{B.16}
$$

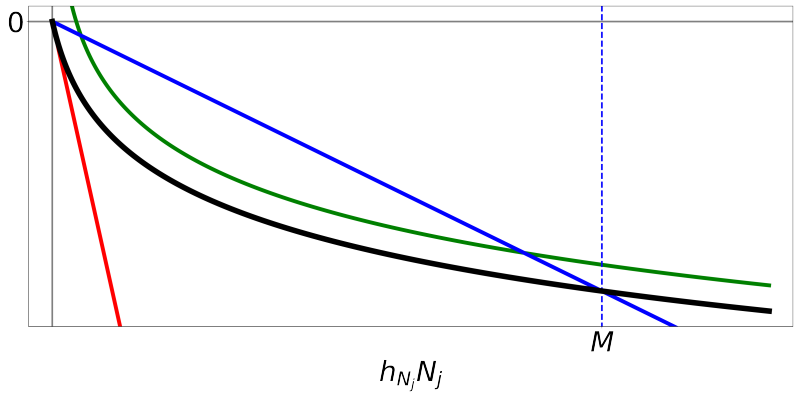


Figure B.2: Graph of the function evaluated at $h_{N_j}N_j$ in black when $\sigma_{N_j,k} < 1$. The red line shows the tangent at 0 with slope $\log(\sigma_{N_j,k}) < 0$. The blue line shows that in this case, where $\sigma_{N_j,k} < 1$, the function may be dominated by some line for all values less than M. The green line shows that as $h_{N_j}N_j \to \infty$, the function is $-\lambda_{N_j,k}\log(h_{N_j}N_j) + O(1)$.

The function

$$x \mapsto -\lambda_{N_j,k}\log(1+x) - \sigma_{N_j,k}\log(\sigma_{N_j,k}) + \left(\sigma_{N_j,k}+x\right)\log\left(\frac{\sigma_{N_j,k}+x}{1+x}\right)$$

has intercept 0, and derivative $\log\left(\frac{\sigma_{N_j,k}+x}{1+x}\right)$, and is thus convex since the derivative is increasing (Fig B.2).

As $x \to \infty$, the function is $-\lambda_{N_j,k}\log x + O(1)$ while the function has tangent $x \mapsto x\log\sigma_{N_j,k}$ at $x = 0$. In our case, we evaluate at $x = h_{N_j}N_j$, which, based on the chosen subsequence, is either bounded or goes to infinity. First assume $h_{N_j}N_j$ is bounded, say by M, and recall that we assumed $\lambda_{k'} > 0$ for some k', so $\sigma_{k'} < 1$. Then, because the function is decreasing and eventually has negative derivative at 0, we can eventually bound it on $[0, M]$ by a line with negative

slope and intercept 0 (Fig B.2), so eventually, for some $C, C' > 0$,

$$\log r_{N_l}(h_{N_j}, \theta_{n_l}) \leq -C\frac{1}{h_{N_j}} N_j h_{N_j} + C' \lesssim -N_j.$$

Otherwise $h_{N_j} N_j \to \infty$ so, by the above remark about the limits of the function as $x \to \infty$,

$$\log r_{N_l}(h_{N_j}, \theta_{n_l}) \leq -\frac{1}{2h_{N_j}} \log(h_{N_j} N_j) \sum_{k \in \text{acc}_L(p^*)} \lambda_{N_j,k} + C$$

for some $C > 0$ eventually. Recalling that $h_{N_j} N_j^\beta$ is eventually bounded above, and by assumption $\log(h_{N_j} N_j) \to \infty$,

$$\log r_{N_l}(h_{N_j}, \theta_{N_l}) \lesssim -N_j^\beta \frac{\log(h_{N_j} N_j)}{h_{N_j} N_j^\beta} \max_k \lambda_k \lesssim -N_j^\beta \max_k \lambda_k.$$

This completes part one of the proof.

Part two: Assume $\| f_{k,b}(\theta_{N_j}) - \tilde{v}_k \| \not\to 0$. We will perform the same technique to allow a Stirling approximation of the prior: define $\gamma_{k,b} = 1/2$ if the limit of $f_{k,b}(\theta_{N_j})/h_{N_j}$ is greater than or equal to 1 and 1 otherwise. Then, for all $v \in \tilde{\Delta}_L(p^*)$ away from the boundary, recalling that we showed

in part 1 $\sigma_{N_j,k} \to 1$ for all k, if $\frac{f_k(\theta_{N_j})}{\sigma_{N_j,k}} \neq v_k$ for some k,

$$
\begin{aligned}
\log \tilde{\pi}(v|h_{N_j}, \theta_{N_j}) =& \sum_k \log \Gamma\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right) \\
&- \sum_{b \in \mathrm{supp}_L(p^*)|_k} \left[\log \Gamma\left(\frac{1}{h_{N_j}} f_{k,b}(\theta_{N_j})\right) - \frac{1}{h_{N_j}} f_{k,b}(\theta_{N_j}) \log v_{k,b}\right] + O(1) \\
=& \sum_k \left(\frac{\sigma_{N_j,k}}{h_{N_j}} - 1/2\right) \log\left(\frac{\sigma_{N_j,k}}{h_{N_j}}\right) \\
&- \sum_{b \in \mathrm{supp}_L(p^*)|_k} \Bigg[\left(\frac{1}{h_{N_j}} f_{k,b}(\theta_{N_j}) - \gamma_{k,b}\right) \log\left(\frac{1}{h_{N_j}} f_{k,b}(\theta_{N_j})\right) \\
&\quad - \frac{1}{h_{N_j}} f_{k,b}(\theta_{N_j}) \log v_{k,b}\Bigg] + O(1) \\
=& \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log\left(\frac{1}{h_{N_j}}\right) - \frac{1}{h_{N_j}} \sum_k \sigma_{N_j,k} \mathrm{KL}\left(\frac{f_k(\theta_{N_j})}{\sigma_{N_j,k}} \middle\| v_k\right) \\
&+ \sum_{k,b \text{ s.t. } \gamma_{k,b}=1} \left(\gamma_{k,b} - \frac{1}{2}\right) \log\left(\frac{1}{h_{N_j}} f_{k,b}(\theta_{N_j})\right) + O(1) \\
\lesssim& - \frac{1}{h_{N_j}} \sum_k \mathrm{KL}\left(\frac{f_k(\theta_{N_j})}{\sigma_{N_j,k}} \middle\| v_k\right).
\end{aligned}
\tag{B.17}
$$

Now note $h_{N_j} \lesssim N^{-\beta}$ and for any norm $|\cdot|$, by Pinsker's inequality,

$$
\sum_k \mathrm{KL}\left(\frac{f_k(\theta_{N_j})}{\sigma_{N_j,k}} \middle\| v_k\right) \gtrsim \sum_k \left\| \frac{f_k(\theta_{N_j})}{\sigma_{N_j,k}} - v_k \right\|^2.
$$

One may check that $\left(\sum_k \|\cdot\|^2\right)^{1/2}$ is also a norm and $\sigma_{N_j,k} \to 1$ for all k, so

$$
\sum_k \mathrm{KL}\left(\frac{f_k(\theta_{N_j})}{\sigma_{N_j,k}} \middle\| v_k\right) \gtrsim \left\| f(\theta_{N_j}) - v \right\|^2 + o(1)
$$

for any norm $\|\cdot\|$. Now note if $\eta < \|f(\theta_{N_j}) - v^*\|$,

$$\sup_{v \in B(\eta)} \log \tilde{\pi}(v|h_{N_j}, \theta_{N_j}) \lesssim -N_j^{\beta} \left(\|f(\theta_{N_j}) - v^*\| - \eta\right)^2.$$

This concludes part two. □

We now have the tools to determine the behavior of h_N and $f(\theta_N)$ in the well and misspecified cases.

B.4.1 The well-specified case

We now examine the asymptotic behavior of empirical Bayes inference for the BEAR model in the well-specified case, or, more precisely, when the model is well-specified "at resolution L", in the sense that there are $\tilde{\theta}_N$ such that for all $k, b \in \text{supp}_L(p^*)$, $f_{k,b}(\tilde{\theta}_N) \to v^*_{k,b}$ (we say the model is misspecified at resolution L otherwise). We first show that the misspecification diagnostic is guaranteed to converge to zero ($h_N \to 0$), correctly indicating that the model is well-specified, and that the embedded AR model converges to the true transition probabilities ($f(\theta_N) \to v^*$). We also give a bound on the rate for the convergence of h_N, a power of the dataset size. We then establish additional weak conditions under which θ_N also converges to the true value θ^*.

Proposition B.4.5. *Say the model is well-specified and $(h_N)_N$ and $(\theta_N)_N$ are sequences maximizing $\log m(\{X_n\}_{n=1}^N | h_N, \theta_N)$. Then $h_N N^{1/4-\epsilon} \to 0$ for every $\epsilon > 0$ and $f_{k,b}(\theta_N) \to v^*_{k,b}$ for all $k, b \in \text{supp}_L(p^*)$ with both sequences converging in probability.*

Proof. If U is a neighborhood of v^* and $\beta > 0$, proposition B.4.4 shows that

$$p^*(h_N < N^{-\beta}, f(\theta_N) \notin U) \to 0$$

(otherwise $p^*(h_N < N^{-\beta}, f(\theta_N) \notin U \text{ for infinitely many } N) > 0$). We show below that $p^*(h_N \geq N^{-1/4+\epsilon}) \to 0$ for any $\epsilon > 0$ and it will thus follow that we also get $f(\theta) \to v^*$ in probability.

Proposition B.4.3 shows that

$$p^*(h_N \geq N^{-1/4+\epsilon}, \log \tilde{\pi}(v^*|h_N, \theta_N) < -\sqrt{N}) \to 0$$

as $h_N \leq N^{-1/4+\epsilon}$ if and only if $h_N N^{1/4-\epsilon/2} \geq N^{\epsilon/2}$. Thus it is sufficient to show that

$$p^*(h_N \geq N^{-1/4+\epsilon}, \log \tilde{\pi}(v^*|h_N, \theta_N) \geq -\sqrt{N}) \to 0.$$

On this set, we may apply theorem B.4.1, but we will need to control $\log \tilde{\pi}(v^*|h, \theta)$.

For any h, θ, defining $\gamma_{k,b} = 1$ if $\frac{1}{h} f_{k,b}(\theta) < 1$ and $1/2$ otherwise, and $\hat{\gamma}_k = 1$ if $\frac{\sigma_k}{h} < 1$ (where

recall $\sigma_k = \sum_{b \in \text{supp}_L(p^*)|_k} f_{k,b}(\theta))$ and $1/2$ otherwise, by the same derivation as equation B.17,

$$\begin{aligned}\log \tilde{\pi}(v^*|h,\theta) =& \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log\left(\frac{1}{h}\right) - \frac{1}{h}\sum_k \sigma_k \textsc{kl}\left(\frac{f_k(\theta)}{\sigma_{N_j,k}}\middle\| v_k^*\right) \\ &+ \sum_{k,b \text{ s.t. } \gamma_{k,b}=1} \left(\gamma_{k,b} - \frac{1}{2}\right)\log\left(\frac{1}{h} f_{k,b}(\theta)\right) \\ &- \sum_{k, \text{ s.t. } \hat{\gamma}_k=1} \left(\hat{\gamma}_k - \frac{1}{2}\right)\log\left(\frac{\sigma_k}{h}\right) + O(1)\end{aligned} \tag{B.18}$$

where $O(1)$ is uniform over h or θ. Since $\hat{\gamma}_k = 1$ only if $\gamma_{k,b} = 1$ for all $b \in \text{supp}_L(p^*)|_k$, by the concavity of the log function, the sum of these last two terms is negative. Thus,

$$\log \tilde{\pi}(v^*|h,\theta) \leq \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log\left(\frac{1}{h}\right) + C \tag{B.19}$$

for all h, θ for some $C > 0$.

Now we derive a lower bound for $\tilde{m}((X_n)_{n=1}^N|h_N,\theta_N)$. Pick $\tilde{\theta}_j$ such that for all $k, b \in \text{supp}_L(p^*)$, $f_{k,b}(\tilde{\theta}_j) \to v^*_{k,b}$. Thus, $\tilde{\pi}(\cdot|h,\tilde{\theta}_j) \to \prod_{k \in \text{acc}_L(p^*)} \text{Dirichlet}(\frac{1}{h}v^*_{k,b})_{b \in \text{supp}_L(p^*)|_k}$ for any $h > 0$ in distribution. And as $h \to 0$, we also have $\prod_{k \in \text{acc}_L(p^*)} \text{Dirichlet}(\frac{1}{h}v^*_{k,b})_{b \in \text{supp}_L(p^*)|_k} \to \delta_{v^*}$. So, pick a sequence $\tilde{\theta}'_j, \tilde{h}_j$ such that $\tilde{\pi}(\cdot|\tilde{h}_j,\tilde{\theta}'_j) \to \delta_{v^*}$ in distribution.[‡] Then $\log m((X_n)_{n=1}^N|\tilde{h}_j,\tilde{\theta}'_j) \to -Nl_N(v^*)$. Thus, $\log m((X_n)_{n=1}^N|h_N,\theta_N) \geq -Nl_N(v^*)$. Also recall that from the proof of theorem B.3.2 that,

[‡]Since $\tilde{\Delta}_L(p^*)$ is compact, the set of polynomials with rational coefficients, $(g_i)_{i=1}^\infty$ is dense in the space of continuous functions under the infinite norm. Pick $\tilde{h}_j$ to have $|g_i(v^*) - \int g_i d\prod_{k \in \text{acc}_L(p^*)} \text{Dirichlet}(\frac{1}{h}v^*_{k,b})_{b \in \text{supp}_L(p^*)|_k}| < 1/j$ for all $i \leq j$ and then $\tilde{\theta}'_j$ to have $|\int g_i d\prod_{k \in \text{acc}_L(p^*)} \text{Dirichlet}(\frac{1}{h}v^*_{k,b})_{b \in \text{supp}_L(p^*)|_k} - \int g_i d\tilde{\pi}(\cdot|h_j,\tilde{\theta}'_j)| < 1/j$ for all $i \leq j$.

defining $Z_N = Nl_N(v_n) - Nl_N(v^*)$, Z_N converges in distribution (to a chi-squared distribution). Since $\log r_N \leq 0$ we can write

$$\log \tilde{m}((X_n)_{n=1}^N | h_N, \theta_N) \geq -Nl_N(v_N) + Z_N. \tag{B.20}$$

Now, when both $h_N \geq N^{-1/4+\epsilon}$, $\log \tilde{\pi}(v^* | h_N, \theta_N) \geq -\sqrt{N}$, applying theorem B.4.1, we've shown that with probability going to 1, for some fixed $C > 0$,

$$\log \tilde{m}((X_n)_{n=1}^N | h_N, \theta_N) \leq -Nl_N(v_N) - \frac{1}{2} \dim \hat{\Delta}_L(p^*) \log N + \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log \left(\frac{1}{h}\right) + C.$$

Thus, as $h_N \geq N^{-1/4+\epsilon}$,

$$\begin{aligned} -\frac{1}{4} \dim \hat{\Delta}_L(p^*) \log N + C \geq & -\frac{1}{2} \dim \hat{\Delta}_L(p^*) \log N + (1/4 - \epsilon) \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log N + C \\ & \geq \log \tilde{m}((X_n)_{n=1}^N | h_N, \theta_N) + Nl_N(v_N) \\ & \geq Z_N. \end{aligned} \tag{B.21}$$

Since Z_N converges in distribution, this occurs with vanishing probability. □

We have thus far discussed the asymptotic behavior of h_N and $f(\theta_N)$. To draw conclusions about θ_N itself, we need to place some assumptions on the autoregressive function f. Here we provide an example of such assumptions, drawn from the theory of M-estimators, which say in essence that f must have an isolated peak at θ^*. These assumptions are enough to guarantee that the empirical Bayes estimate of the AR model parameter θ converges to the true value θ^*.

Corollary B.4.6. *Say $\theta^* \in \Theta$ and d is a metric on Θ such that $f_{k,b}(\theta^*) = v^*_{k,b}$ for all $k, b \in \text{supp}_L(p^*)$ and for all $\delta > 0$,*

$$0 < \inf_{d(\theta,\theta^*)>\delta} \|f(\theta) - v^*\|.$$

Then $\theta_N \to \theta^*$ in probability.

Proof. Since by proposition B.4.5 we have $\|f(\theta_N) - v^*\| = o_P(1)$, we may apply theorem 5.7 of van der Vaart [268] to get the result. □

Taking a step back, a perhaps surprising aspect of these results is the weak conditions on f. Were we, instead of trying to diagnose misspecification in the AR model, simply trying to analyze uncertainty in the AR model's parameter estimate, we might proceed by putting a prior on θ and performing Bayesian inference for the AR model. In this case, to guarantee asymptotic normality and well-calibrated frequentist coverage, we would in general need strong conditions on f, such as bounded third derivatives [176]. Intuitively, the task of diagnosing misspecification might seem to be harder than describing parameter uncertainty, but our conditions on f in this section and the next are in fact much weaker, involving no restrictions on the derivatives of f whatsoever.

B.4.2 The misspecified case

We now consider the case where the AR model is misspecified at resolution L. In this case, we can rewrite the marginal likelihood of the BEAR model (using propositions B.4.3 and B.4.4 to apply

theorem B.4.1) as

$$\log m((X_n)_{n=1}^N | h_N, \theta_N) = -N l_N(v_N) - \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log N + C_{v^*} - \mathcal{L}_N(h_N, \theta_N) + o(1)$$

where we define $\mathcal{L}_N(h_N, \theta_N) = -\log \tilde{\pi}(v^* | h, \theta) - r_N(h, \theta)$.[§] This expression for the marginal likelihood takes the form of a modified Laplace approximation where, instead of the original prior π evaluated at the true parameter value, we have the prior over the support of the data, $\tilde{\pi}(v^* | h, \theta)$, as well as the additional term r_N, which is $O(\log N)$ rather than $O(1)$ and depends on the concentration of the prior outside the support of the data. Instead of the standard empirical Bayes behavior described by Petrone et al. [196], wherein the prior probability of the true parameters is maximized, we instead heuristically expect that the objective function $\mathcal{L}_N(h, \theta)$ is minimized. The following result makes this intuition formal, showing that h_N and θ_N indeed behavior similarly to the minimizers of $\mathcal{L}_N$.

Corollary B.4.7. *If the model is misspecified at resolution L, a.s. $\mathcal{L}_N(h_N, \theta_N) = \sup_{h,\theta} \mathcal{L}_N(h, \theta) - o(1)$.*

Proof. Say $\hat{h}_N, \hat{\theta}_N$ are sequences such that $\mathcal{L}_N(\hat{h}_N, \hat{\theta}_N) = \sup_{h,\theta} \mathcal{L}_N(h, \theta) - o(1)$. For fixed h, θ, we have $\mathcal{L}_N(h, \theta) = O(\log N)$. Thus, for any $\beta > 0$ we clearly have $\liminf(\log \tilde{\pi}(v^* \mid \hat{h}_N, \hat{\theta}_N))/N^\beta \geq 0$ and since we are in the misspecified case, following the logic of proposition B.4.4, equation B.16 may be used to see that we also have $\hat{h}_N N^\beta \to \infty$.

[§] $\mathcal{L}_N$ is stochastic due to r_N, but since $h_N N^\beta \to \infty$ for any $\beta > 0$, using the expansion in equation B.16, one may show that the $\#k$ in r_N can be replaced with $N\mathbb{E}\#k$ incurring only a penalty of $O_P(N^{-1/2+\epsilon})$.

Thus theorem B.4.1 may be applied to $\hat{h}, \hat{\theta}$ and a comparison of the Laplace approximations of $m((X_n)_{n=1}^N | \hat{h}_N, \hat{\theta}_N)$ and $m((X_n)_{n=1}^N | h_N, \theta_N)$ gives the result. □

We next examine in greater detail the behavior of the misspecification diagnostic h_N, along with the AR parameter estimate θ_N. There are two cases to consider. First, if the support of the AR model matches the support of the data-generating distribution (that is, $\text{supp}(f(\theta)) = \text{supp}_L(p^*)$ for all θ), then $r_N = 0$ and $\mathcal{L}_N = -\log \tilde{\pi}(v^* | h, \theta)$; we thus recover the standard empirical Bayes behavior of Petrone et al. [196], with h_N and θ_N asymptotically maximizing the prior probability of the true parameter value. In this case we find that h_N converges to a finite positive value. The second case to consider is when $\text{supp}_L(p^*) \subsetneq \text{supp}(f(\theta))$. Here, we have $r_N \neq 0$, and in particular $r_N(h, \theta) \approx -\frac{1}{h} \log(N) \sum_k \lambda_k(\theta)$. In this case we find that $h_N \to \infty$. Thus, in either case, $h_N \not\to 0$, and so h_N will correctly diagnose misspecification in the AR model.

Corollary B.4.8. *If the model is misspecified at resolution L but $\text{supp}(f(\theta)) = \text{supp}_L(p^*)$ for all θ, h_N is eventually bounded above and below.*

Proof. Recall from proposition B.4.4 that if $h \to 0$, $\log \tilde{\pi}(v^* | h, \theta) \leq -C \frac{1}{h} \inf_\theta \| f(\theta) - v^* \|$ for some $C > 0$. This expression diverges to $-\infty$ as $h \to 0$. We also showed in proposition B.4.5 that $\log \tilde{\pi}(v^* | h, \theta) \leq \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log(1/h) + C$ for some $C > 0$. This expression also diverges as $h \to \infty$. Combining these two observations along with corollary B.4.7 we get the result. □

To say something about θ_N, due to corollary B.4.7, we may use the theory of extremum estimators we can apply theorem 5.7 of van der Vaart [268], replacing limits in probability with a.s. limits to get

Corollary B.4.9. *Say the model is misspecified at resolution L but* $\operatorname{supp}(f(\theta)) = \operatorname{supp}_L(p^*)$ *for all* θ*. Say also that* $\theta^* \in \Theta$*,* $h^* > 0$ *and* d *is a metric on* Θ *such that for every* $\delta > 0$*,*

$$\log \tilde{\pi}(v^*|h^*, \theta^*) > \sup_{|h-h^*| \vee d(\theta,\theta^*) > \delta} \log \tilde{\pi}(v^*|h, \theta).$$

Then $\theta_N \to \theta^*$ *and* $h_N \to h^*$ *a.s..*

Now we consider the case where the support do not match, i.e. $\inf_\theta \max_k \lambda_k(\theta) > 0$, where $\lambda_k(\theta) = \sum_{b \notin \operatorname{supp}_L(p^*)|_k} f_{k,b}(\theta)$.

Proposition B.4.10. *If the model is misspecified at resolution* L*,* $\operatorname{supp}_L(p^*) \subsetneq \operatorname{supp}(f(\theta))$ *for all* θ*, and* $\inf_\theta \max_k \lambda_k(\theta) > 0$*, then* $h_N \to \infty$*.*

Proof. We first show h_N is a.s. bounded below. Since $h_N N^\beta \to \infty$ for all $\beta > 0$, if $h_{N_j} \to 0$ for some subsequence, we showed in proposition B.4.4 that a.s.
$\log r_{N_j}(h_{N_j}, \theta_{N_j}) \leq -C\frac{\log(h_{N_j} N_j)}{2h_{N_j}} \inf_\theta \max_k \lambda_{k,\theta} + C' \leq -C''\frac{\log(N_j)}{2h_{N_j}} \inf_\theta \max_k \lambda_{k,\theta} + C'$
for some $C, C', C'' > 0$. In particular, $\log r_{N_j} \lesssim -O(\log(N))$ but $\log r_{N_j} \not\sim -O(\log(N))$ if $h_{N_j} \to 0$. Thus, since $\log r_N(h, \theta) \geq -C \log(N)$ for fixed h, θ, for some $C > 0$ dependent on h, θ and $\tilde{\pi}$ also diverges as $h \to 0$, the assumption that h_N maximizes the marginal likelihood is contradicted. Thus, $h_N \not\to 0$. In particular, we showed in proposition B.4.5 (equation B.19) that $\log \tilde{\pi}(v^*|h, \theta) \leq \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log(1/h) + C$ for some $C > 0$ so we get that $\log \tilde{\pi}(v^*|h_N, \theta_N)$ is bounded above a.s..

Assume h_N is bounded above; we will show that this leads to a contradiction. Define $\gamma_{N,k} =$

$1/2$ if $\sigma_k(\theta_N)/h_N \geq 1$ and $\gamma_{N,k} = 1$ otherwise. Define $\hat{\gamma}_{N,k}$ similarly for $1/h_N$ alone. We next perform the same trick as in proposition B.4.4, expanding $\Gamma(\frac{1}{h_N})$ in the form of a Stirling approximation, to analyze r_N further. Noting that $\log(h_N N) \to \infty$, we have a.s.,

$$\begin{aligned}
\log r_N(h_N, \theta_N) = &\sum_{k \in \text{acc}_L(p^*)} \frac{1}{h_N} \Bigg[- \lambda_k(\theta_N) \log(1 + \#k h_N) - \sigma_{N,k} \log(\sigma_k(\theta_N)) \\
&\qquad + (\sigma_k(\theta_N) + \#k h_N) \log\left(\frac{\sigma_k(\theta_N) + \#k h_N}{1 + \#k h_N}\right) \Bigg] \\
&+ \sum_{k \in \text{acc}_L(p^*)} (\gamma_{N,k} - 1/2) \log\left(\frac{\sigma_k(\theta_N)}{h_N}\right) \\
&- \sum_{k \in \text{acc}_L(p^*)} (\hat{\gamma}_{N,k} - 1/2) \log\left(\frac{1}{h_N}\right) + O(1) \qquad \text{(B.22)} \\
= &- \frac{\log(h_N N)}{h_N} \sum_{k \in \text{acc}_L(p^*)} \Big[\lambda_k(\theta_N) + o(1) \Big] \\
&+ \sum_{k \in \text{acc}_L(p^*)} (\gamma_{N,k} - 1/2) \log\left(\frac{\sigma_k(\theta_N)}{h_N}\right) \\
&- \sum_{k \in \text{acc}_L(p^*)} (\hat{\gamma}_{N,k} - 1/2) \log\left(\frac{1}{h_N}\right) + O(1).
\end{aligned}$$

Note $\hat{\gamma}_{N,k} = 1$ only if $\gamma_{N,k} = 1$ so that the the sum of these last two terms is negative. So, since h_N is bounded above, $\log r_N(h_N, \theta_N) \leq -C \log(N) \inf_\theta \max_k \lambda_k(\theta)$ for some $C > 0$. Thus, since we also have that $\log \tilde{\pi}(v^* | h_N, \theta_N)$ is bounded above a.s., we get that $\mathcal{L}_N(h_N, \theta_N) \gtrsim \log(N)$ a.s.. On the other hand, with fixed θ, if $\hat{h}_N \to \infty$ (so we still have $\log(h_N N) \to \infty$), then

$$\log r_N(\hat{h}_N, \theta) = -\frac{\log(N)}{\hat{h}_N} \sum_{k \in \text{acc}_L(p^*)} \Big[\lambda_k(\theta) + o(1) \Big] + \frac{1}{2} \sum_{k \in \text{acc}_L(p^*)} \log\left(\sigma_k(\theta)\right) + O(1)$$

which is $-o(\log N)$, where we wrote $\log(\hat{h}_N)/\hat{h}_N = o(1)$. Now pick $\hat{h}_N$ increasing slowly so that $\mathcal{L}_N(\hat{h}_N, \theta) = o(\log(N))$. This is eventually less than $\mathcal{L}_N(h_N, \theta_N)$, a contradiction. Thus, $h_N \to \infty$. □

We can also study the behavior of θ_N in this mismatched supports case, using again the theory extremum estimators. We briefly outline the strategy, omitting details. Further analysis of equations B.18 and B.22 gives an objective, as $h \to \infty$, [¶]

$$\begin{aligned}\mathcal{L}(h, \theta) = &- \frac{\log(N)}{h}\left(\sum_k \lambda_k + o(1)\right) \\ &- \dim \tilde{\Delta}_L(p^*) \log h + (1 + o(1)) \sum_{k,b \in \text{supp}_L(p^*)} \log(f_{k,b}(\theta)) + C + o(1)\end{aligned}$$

for some fixed $C > 0$. Careful analysis of the $o(1)$ terms shows that h approaches $\frac{\log(N)\sum_k \lambda_k}{\dim \tilde{\Delta}_L(p^*)}$. Plugging this value of h in, the objective becomes

$$\mathcal{L}(h, \theta) = - \dim \tilde{\Delta}_L(p^*) \log \sum_k \lambda_k + \sum_{k,b \in \text{supp}_L(p^*)} \log(f_{k,b}(\theta)) + C_N + o(1)$$

for some constant C_N dependent only on N and p^*. One can then see that θ_N is an M-estimator of $\dim \tilde{\Delta}_L(p^*) \log \sum_k \lambda_k + \sum_{k,b \in \text{supp}_L(p^*)} \log(f_{k,b}(\theta))$ and apply a similar analysis as in corollary B.4.9.

So far we have seen that $h_N \not\to 0$ when the AR model is misspecified at resolution L, but exactly what value will h_N take and what can it tell us about the amount of misspecification? Here we ana-

[¶] Note that the KL term in $\tilde{\pi}$ can be dominated by $\sum_{k,b \in \text{supp}_L(p^*)} \log(f_{k,b}(\theta))$.

lyze the objective $\mathcal{L}_N$ heuristically to address these questions. From the expansions in proposition B.4.4, we can write, for reasonable values of h, θ, assuming not too much misspecification,

$$\log \tilde{\pi}(v^*|h,\theta) \approx \frac{1}{2} \dim \tilde{\Delta}_L(p^*) \log\left(\frac{1}{h}\right) - \frac{1}{h} \sum_{k \in \text{acc}_L(p^*)} \text{KL}(f_k(\theta) \| v_k^*)$$

$$\log r_N(h,\theta) \approx -\frac{\log(N)}{h} \sum_{k \in \text{acc}_L(p^*)} \lambda_k(\theta).$$

We see, then, that θ_N and h_N depend on an unconventional but valid divergence between the AR model and $p^{*(L)}$: the sum of the KL divergence between the AR model transition probabilities (from kmers that occur with non-zero probability) and the true transition probabilities, plus a penalty proportional to $\log(N)$ when the support of the AR model does not match the support of p^*. We can thus interpret h_N not only as a diagnostic of misspecification, but also as a measurement of the *amount* of misspecification, and make comparisons between different AR models on the basis of their h_N values.

B.5 Hypothesis testing

In this section we use the results of the above sections to develop goodness-of-fit and two sample tests.

B.5.1 Goodness-of-fit test

Say p^* is a distribution on S with $\mathbb{E}|X|^2 < \infty$ and say $X_1, X_2, \cdots \sim p^*$ iid. Say $\tilde{p}$ is another distribution on S with $\mathbb{E} \log^2 \tilde{p}(X) < \infty$ where the expectation is with respect to p^*. We are

interested in testing whether or not $p^* = \tilde{p}$, so we will consider the Bayes factor

$$\mathrm{BF}_L = \frac{\tilde{p}(X_n)_{n=1}^N}{p((X_n)_{n=1}^N \mid \mathcal{M}_L)}.$$

This test asks whether or not $\tilde{p}$ approximates p^* at least as well as the optimal model in $\mathcal{M}_L$. We can use it in particular to test whether $\tilde{p}$ matches the data-generating distribution p^* at resolution L, that is, whether $\tilde{p}$ matches $p^{*(L)}$.

Proposition B.5.1. *Given L, consider a* Dirichlet$(\alpha_{k,b})_{b\in\mathcal{B}}$ *prior on the simplex in $\Delta_{\mathcal{B}}^{\mathcal{B}_L^o}$ corresponding to the L-mer k. For all L, assume $\alpha_{k,b} > 0$ for $(k,b) \in \mathit{supp}_L(p^*)$ (otherwise $p((X_n)_{n=1}^N|\mathcal{M}_L)$ is eventually 0 a.s.). Then if $\tilde{p} \neq p^{*(L)}$,*

$$\log BF_L = N(\kappa\iota(p^*||p^{*(L)}) - \kappa\iota(p^*||\tilde{p})) + O_P(\sqrt{N}),$$

which goes to ∞ in probability if $\kappa\iota(p^*||p^{*(L)}) > \kappa\iota(p^*||\tilde{p})$ and to $-\infty$ in probability if $\kappa\iota(p^*||p^{*(L)}) < \kappa\iota(p^*||\tilde{p})$. If $\tilde{p} = p^{*(L)}$

$$\log BF_L = \frac{1}{2}\dim_L^{eff}(p^*)\log N + O_P(1),$$

which goes to ∞ in probability.

Proof. Note that as shown in the proof of theorem B.3.2, $\mathrm{KL}(p^*||\mathcal{M}_L) = \mathrm{KL}(p^*||p^{*(L)})$, and

$$\log p((X_n)_{n=1}^N|\mathcal{M}_L) = \log p^{*(L)}((X_n)_{n=1}^N) - \frac{1}{2}\dim_L^{eff}(p^*)\log N + O_P(1). \tag{B.23}$$

As well, $\tilde{p}(X_n)_{n=1}^N = N\mathbb{E}\log\tilde{p}(X) + O_P(\sqrt{N})$ and a similar expression can be written for $p^{*(L)}$. These two facts prove the result. □

Remark B.5.2. *One may also consider a Bayes factor that integrates over many L:*

$$BF = \frac{\tilde{p}(X_n)_{n=1}^N}{\sum_{L=1}^{\tilde{L}}\pi(L)p((X_n)_{n=1}^N \mid \mathcal{M}_L)} = \left(\sum_{L=1}^{\tilde{L}}\pi(L)BF_L^{-1}\right)^{-1}$$

for a prior π with $\pi(\tilde{L}) > 0$. By proposition B.5.1, this Bayes factor goes to 0 if $\mathrm{KL}(p^*\|p^{*(\tilde{L})}) < \mathrm{KL}(p^*\|\tilde{p})$ and goes to ∞ if $\mathrm{KL}(p^*\|p^{*(\tilde{L})}) > \mathrm{KL}(p^*\|\tilde{p})$ or $\tilde{p} = p^{*(\tilde{L})}$ (this later condition is implied by $\tilde{p} = p^{*(L)}$ for some $L \leq \tilde{L}$ and $\mathrm{KL}(p^*\|p^{*(L)}) = \mathrm{KL}(p^*\|\tilde{p})$). Thus this Bayes factor has the same asymptotics as $BF_{\tilde{L}}$.

B.5.2 Two-sample test

To set up the two-sample testing problem, consider two distributions p_1 and p_2 on S such that $\mathbb{E}_{p_j}|X|^2 < \infty$ for $j \in \{1, 2\}$. We will assume that the two groups of datapoints are sampled together according to a mixture model with observed labels. That is, let $j_1, j_2, \ldots$ be observed Bernoulli iid random variables indicating the group, with $j_n = 1$ with probability β and $j_n = 2$ with probability $1 - \beta$ for a $0 < \beta < 1$. Then, let $X_n \sim p_{j_n}$ independently. The pooled dataset

thus follows the generative process $X_1, X_2, \cdots \sim p^* = \beta p_1 + (1-\beta)p_2$ iid. We are interested in whether or not $p_1 \neq p_2$. To make this question theoretically tractable, we will fix the lag L, and attempt only to discern whether $p_1^{(L)} \neq p_2^{(L)}$ where $p_j^{(L)}$ is the best approximation to p_j in $\mathcal{M}_L$ (as defined in section B.3). In other words, we attempt to distinguish between p_1 and p_2 only up to a "resolution", in analogy to Holmes et al. [106]. We thus consider the Bayes factor

$$\begin{aligned} \mathrm{BF}_L &= \frac{p((X_n)_{n=1}^N | (j_n)_{n=1}^N, p_1 = p_2 \text{ and } p_1, p_2 \in \mathcal{M}_L)}{p((X_n)_{n=1}^N | (j_n)_{n=1}^N, p_1 \neq p_2 \text{ and } p_1, p_2 \in \mathcal{M}_L)} \\ &= \frac{p((X_n)_{n=1}^N | \mathcal{M}_L)}{p((X_n)_{n \leq N, j_n = 1} | \mathcal{M}_L) p((X_n)_{n \leq N, j_n = 2} | \mathcal{M}_L)}. \end{aligned} \tag{B.24}$$

In the subsequent remark, we also extend the theory to Bayes factors that integrate over all L up to some fixed maximum.

Consider independent Dirichlet$(\alpha_{k,b})_{b \in \tilde{\mathcal{B}}}$ priors on the simplexes in $\Delta^{\mathcal{B}_L^o}_{|\tilde{\mathcal{B}}|}$ corresponding to the L-mers k. Assume $\alpha_{k,b} > 0$ for $(k,b) \in \mathrm{supp}_L(p^*) = \mathrm{supp}_L(p_1) \cup \mathrm{supp}_L(p_2)$.

Proposition B.5.3. *If* $p_1^{(L)} \neq p_2^{(L)}$,

$$\begin{aligned} \log BF_L &= N \left[\beta \mathbb{E}_{p_1} \log \frac{p^{*(L)}(X)}{p_1^{(L)}(X)} + (1-\beta) \mathbb{E}_{p_2} \log \frac{p^{*(L)}(X)}{p_2^{(L)}(X)} \right] + O_P(\sqrt{N}) \\ &\to -\infty \text{ as } N \to \infty. \end{aligned} \tag{B.25}$$

Otherwise $p_1^{(L)} = p_2^{(L)}$ *and*

$$\begin{aligned} \log BF_L &= \frac{1}{2} \dim_L^{eff}(p^*) \log N + O_P(1) \\ &\to \infty \text{ as } N \to \infty. \end{aligned} \tag{B.26}$$

Proof. First note that as shown in the proof of theorem B.3.2, noting $|\{n|j_n = j\}|/N = O_P(1)$,

$$\begin{aligned}\log p((X_n)_{n=1}^N|\mathcal{M}_L) &= \log p^{*(L)}((X_n)_{n=1}^N) - \frac{1}{2}\dim_L^{eff}(p^*)\log N + O_P(1)\\ \log p((X_n)_{n\leq N, j_n=j}|\mathcal{M}_L) &= \log p_j^{(L)}((X_n)_{n\leq N, j_n=j}) - \frac{1}{2}\dim_L^{eff}(p_j)\log N + O_P(1)\end{aligned} \tag{B.27}$$

for $j \in \{1,2\}$. As well, $\log p^{*(L)}((X_n)_{n=1}^N) = N\mathbb{E}\log p^{*(L)}(X) + O_P(\sqrt{N})$ by our assumption on the moments $\mathbb{E}_{p_j}|X|^2 < \infty$ and similar expressions exist for p_1 and p_2. Finally note that

$$\begin{aligned}\arg\min_{v\in\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}^o_L}} \textsc{kl}(p^*||p_v) &= \arg\max \mathbb{E}_{p^*}\log p_v(X)\\ &= \arg\max \beta\mathbb{E}_{p_1}\log p_v(X) + (1-\beta)\mathbb{E}_{p_2}\log p_v(X).\end{aligned} \tag{B.28}$$

Thus, if $p_1^{(L)} = p_2^{(L)}$ then $p_1^{(L)} = p_2^{(L)} = p^{*(L)}$.

First assume $p_1^{(L)} \neq p_2^{(L)}$. So, we have

$$\begin{aligned}\log \mathrm{BF}_L &= N\mathbb{E}_{p^*}\log p^{*(L)} - \beta N\mathbb{E}_{p_1}\log p_1^{(L)}(X) - (1-\beta)N\mathbb{E}_{p_2}\log p_2^{(L)}(X) + O_P(\sqrt{N})\\ &= N\left[\beta\mathbb{E}_{p_1}\log\frac{p^{*(L)}}{p_1^{(L)}} + (1-\beta)\mathbb{E}_{p_2}\log\frac{p^{*(L)}}{p_2^{(L)}}\right] + O_P(\sqrt{N}).\end{aligned} \tag{B.29}$$

Note $\mathbb{E}_{p_1}\log\frac{p^{*(L)}}{p_1^{(L)}} = \textsc{kl}(p_1||p_1^{(L)}) - \textsc{kl}(p_1||p^{*(L)}) \leq 0$ by the definition of $p_1^{(L)}$. Since $p_1^{(L)} \neq p_2^{(L)}$, at least one of $\mathbb{E}_{p_1}\log\frac{p^{*(L)}}{p_1^{(L)}}$, $\mathbb{E}_{p_2}\log\frac{p^{*(L)}}{p_2^{(L)}}$ must be negative and so $\log \mathrm{BF}_L \to -\infty$.

Now say $p_0^{(L)} = p^{*(L)} = p_1^{(L)}$. In this case,

$$\log \mathrm{BF}_L = \frac{1}{2}\dim_L^{eff}(p^*)\log N + O_P(1).$$

Clearly $\log \mathrm{BF}_L \to \infty$.

□

Remark B.5.4. *One may also consider a Bayes factor that integrates over many lags:*

$$BF = \frac{\sum_{L=1}^{\tilde{L}} \pi(L) p((X_n)_{n=1}^N | \mathcal{M}_L)}{\left(\sum_{L=1}^{\tilde{L}} \pi(L) p((X_n)_{n \le N, j_n = 1} | \mathcal{M}_L)\right)\left(\sum_{L=1}^{\tilde{L}} \pi(L) p((X_n)_{n \le N, j_n = 2} | \mathcal{M}_L)\right)}.$$

By theorem B.3.2, for all three sums, eventually either (a) assuming the condition for consistency in corollary B.3.6 the term corresponding to the smallest L such that $p^* \in \mathcal{M}_L$ will dominate, if $p^* \in \mathcal{M}_{\tilde{L}}$, or (b) the term corresponding to $\tilde{L}$ will dominate, if $p^* \notin \mathcal{M}_{\tilde{L}}$. Thus, by analysis similar to that of proposition B.5.3, in any case, we have equation B.25 with L replaced by $\tilde{L}$, so that the Bayes factor goes to 0 if $p_1^{(\tilde{L})} \neq p_2^{(\tilde{L})}$. If, on the other hand, we have $p_1^{(\tilde{L})} = p_2^{(\tilde{L})}$, then there are two cases: $p_1 = p_2 \in \mathcal{M}_{L^*}$ for some $L^* \le \tilde{L}$ (and L^* is picked to be the smallest such lag), or $p_1, p_2 \notin \mathcal{M}_{\tilde{L}}$. In the first case, $p^* \in \mathcal{M}_{L^*}$ so the asymptotics of BF are identical to that of BF_{L^*} and we can refer to proposition B.5.3 to see that the Bayes factor goes to ∞. In the second case, we may still have $p^* \in \mathcal{M}_{L^*}$ for some minimal $L^* \le \tilde{L}$; if p^* is not a Markov model with lag $\le \tilde{L}$, call $L^* = \tilde{L}$. In this case, by the analysis of proposition B.5.3,

$$\log BF = \left(\dim_{\tilde{L}}^{\mathit{eff}}(p^*) - \frac{1}{2}\dim_{L^*}^{\mathit{eff}}(p^*)\right)\log N + O_P(1) \to \infty.$$

Thus the asymptotics of this integrated Bayes factor are identical to that of $BF_{\tilde{L}}$.

B.6 Consistency in the infinite L case

So far we have only studied consistency in the finite lag L case, that is, our results only show that we can approximate p^* up to some finite resolution L (corresponding to the largest available lag). In this section, we develop frequentist and Bayesian consistency results for the fully nonparametric model, that is, we allow for priors with support over all lags L up to infinity, and show that we can approximate p^* itself even if $p^* \notin \mathcal{M}$. The Bayesian consistency result is our main result, and the most practically useful, but the frequentist result is a natural first step toward the Bayesian result, and an opportunity to develop novel constructions (such as the projection algorithm in section B.6.2) useful in proving the Bayesian result.

B.6.1 Frequentist consistency

We first show that maximum likelihood estimation is consistent, using the method of sieves described in Geman & Hwang[86]. The idea is to increase the size of the model class with the amount of data N slowly enough to avoid over-fitting. We define the model class considered for N data points first with the lag L, but also by restricting transition probabilities to be bounded below by a ν: In particular, when there are N datapoints, the model class we consider, or the N-th "sieve", is $\mathcal{S}_N = \{v \in \Delta_{\vec{\mathcal{B}}}^{\mathcal{B}^o_{L_N}} \mid \forall k, b, v_{k,b} \geq \nu_N\}$ where $(\nu_N)_{N=1}^{\infty}$, $(L_N)_{N=1}^{\infty}$ are sequences with $L_N \to \infty$, $\nu_N \to 0$.

Theorem B.6.1. *Say $X_1, X_2, \cdots \sim p^*$ iid where p^* is a subexponential distribution on S. Say p_{v_N} is a maximum likelihood distribution with $v_N \in \mathcal{S}_N$ given $(X_n)_{n=1}^N$. $p_{v_N} \to p^*$ and*

$\text{KL}(p^*||p_{v_N}) \to 0$ *a.s. if for some* $\epsilon > 0$,

$$\frac{|supp_{L_N}(p^*)|(\log\left(\nu_N^{-1}\right))^{1+\epsilon}}{N} \to 0. \tag{B.30}$$

Proof. The proof follows that of theorem 3 of Geman & Hwang [86].

First note that $\mathcal{S}_N$ is compact and the likelihood function is continuous so a maximum likelihood v_N always exists. This satisfies condition C1 of theorem 2 of Geman & Hwang [86].

Next, to satisfy condition C2 (b) of theorem 2 of Geman & Hwang [86] we show that there are $\tilde{v}_N \in \mathcal{S}_N$ such that $\text{KL}(p^*||p_{\tilde{v}_N}) \to 0$. First, for each L, pick a distribution p^L on S such that for all $|X| \leq L, p^L(X) > 0$ and $\text{KL}(p^*||p^L) \to 0$ as $L \to \infty$ (for example, pick $p^L(|X| > L) = p^*(|X| > L), p^L(\cdot||X| > L) = p^*(\cdot||X| > L)$ and $p^L(\cdot||X| \leq L)$ positive with $\text{KL}(p^*(\cdot||X| \leq L)||p^L(\cdot||X| \leq L)) < 1/L$). p_L^L as defined in proposition B.2.3 is a lag L Markov model with positive transition probabilities. Thus, for large N, its transition probabilities are in $\mathcal{S}_N$.

Now notice,

$$\begin{aligned}
\text{KL}(p^*||p_L^L) =& \mathbb{E}\left[\log\left(\frac{p^*(X)}{p_L^L(X)}\right); |X| \leq L\right] + \mathbb{E}\left[\log\left(\frac{p^*(X)}{p_L^L(X)}\right); |X| > L\right] \\
=& \mathbb{E}\left[\log\left(\frac{p^*(X)}{p^L(X)}\right); |X| \leq L\right] \\
&+ \mathbb{E}\left[\log\left(\frac{p^*(X)}{p^L(X_{1:L}\ldots)|\tilde{\mathcal{B}}|^{-(|X|-L)}}\right); |X| > L\right] \\
\leq& \mathbb{E}\left[\log\left(\frac{p^*(X)}{p^L(X)}\right); |X| \leq L\right] \\
&+ \mathbb{E}\left[\log\left(\frac{p^*(X)}{p^L(X)|\tilde{\mathcal{B}}|^{-(|X|-L)}}\right); |X| > L\right] \\
=& \text{KL}(p^*||p^L) + \left(\log|\tilde{\mathcal{B}}|\right)\mathbb{E}\left[|X| - L; |X| > L\right] \\
\to& 0 \text{ as } L \to \infty \text{ as } \mathbb{E}|X| < \infty.
\end{aligned} \tag{B.31}$$

Now we can pick $\tilde{v}_N \in \mathcal{S}_n$ such that $\text{KL}(p^*||p_{\tilde{v}_N}) \to 0$.

That $\text{KL}(p^*||p_N) \to 0$ implies $p_N \to p$ for distributions p_N on S follows from Pinsker's inequality. This satisfies condition C2 (a) of theorem 2 of Geman & Hwang [86]. However, note that the proof of theorem 2 of Geman & Hwang [86] also shows that if v_N is an MLE in S_N and the conditions of the theorem hold, then $\text{KL}(p^*||p_{v_N}) \to 0$ a.s..

Finally, we define a partition of each S_N that satisfies conditions i-iii of theorem 2 of Geman & Hwang [86] to get the result. Pick a sequence $\rho_N \to 0$ with $\log\left(\nu_N^{-1}\right) > (\log(1+\rho_N))^{-1}$ eventually. Call $\mathbb{N}$ the set of positive integers and for a $\zeta \in \mathbb{N}^{\text{supp}_{L_N}(p^*)}$, define

$$\hat{\mathcal{O}}_N(\zeta) := \{v \in \mathcal{S}_N | \, \forall (k,b) \in \text{supp}_{L_N}(p^*),\ (1+\rho_n)^{\zeta_{k,b}}\nu_N > v_{k,b} \geq (1+\rho_N)^{\zeta_{k,b}-1}\nu_N\}$$

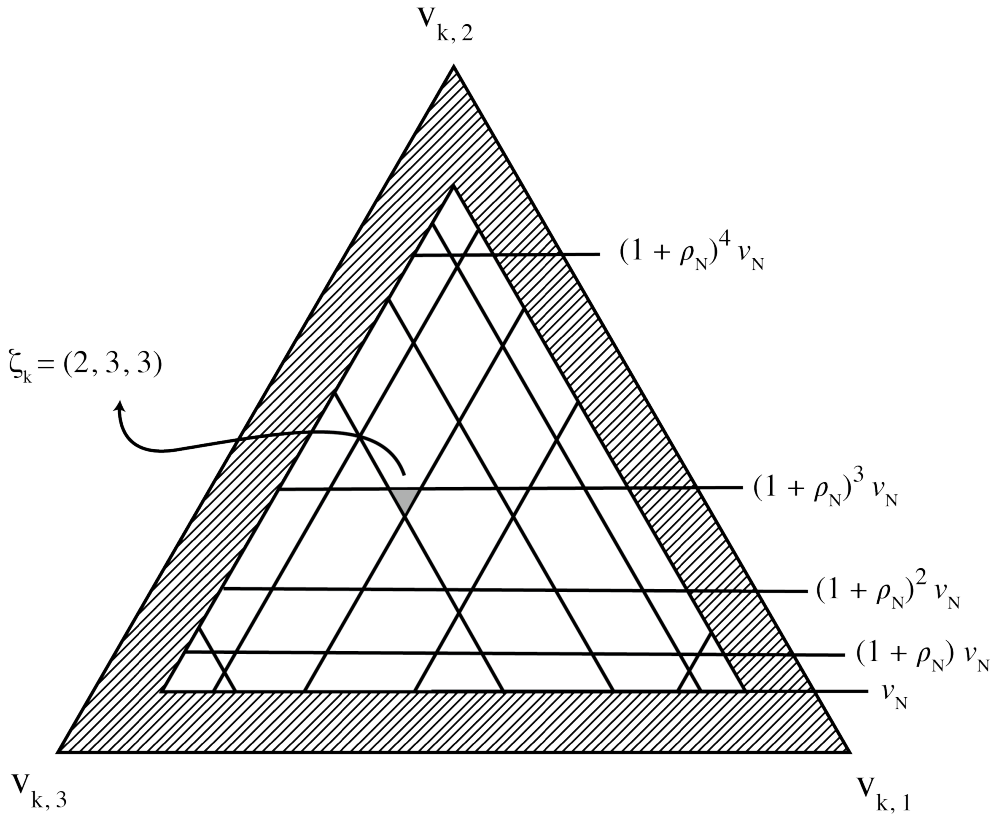


Figure B.3: Sieves $\mathcal{S}_N$ are broken up into subsets $\hat{\mathcal{O}}_N(\zeta)$, each a Cartesian product of subsets of $\Delta_{\tilde{\mathcal{B}}}$, and these subsets in turn are indexed by ζ_k for each k. Here we illustrate one such subset of $\Delta_{\tilde{\mathcal{B}}}$, when $|\tilde{\mathcal{B}}| = 3$ and $\text{supp}_{L_N}(p^*)|_k = \tilde{\mathcal{B}}$. The region included in $\hat{\mathcal{O}}_N(\zeta)$ when $\zeta_k = (2, 3, 3)$ is shown in solid gray, while all other possible subsets for different values of ζ_k are shown in white. The region adjacent to the border of the simplex (hatched lines) corresponds to those transition vectors that have components less than ν_N and are therefore not part of the sieve $\mathcal{S}_N$.

so that $\cup_{\zeta \in \mathbb{N}^{\text{supp}_{L_N}(p^*)}} \hat{\mathcal{O}}_N(\zeta) = \mathcal{S}_N$ (Fig. B.3).

Call $\gamma_N = \left(\frac{\log\left(\nu_N^{-1}\right)}{\log(1+\rho_N)} + 1 \right)$ and note $(1+\rho_N)^{\gamma_N - 1}\nu_N = 1$. Thus the number of choices of ζ that give non-empty sets, call this $\#\hat{\mathcal{O}}_N$, is bounded above by $\gamma_N^{|\text{supp}_{L_N}(p^*)|}$. Now notice eventually

$$\gamma_N = \frac{\log\left(\nu_N^{-1}\right)}{\log(1+\rho_N)} + 1 \geq \left(\log\left(\nu_N^{-1}\right)\right)^2 + 1 \geq \left(\log\left(\nu_N^{-1}\right)\right)^4$$

so that $\#\hat{\mathcal{O}}_N \leq \exp\left(4\left(\log\log\left(\nu_n^{-1}\right)\right)|\text{supp}_{L_n}(p^*)|\right)$.

Say $\eta > 0$ and, picking a $\zeta \in \mathbb{N}^{\text{supp}_{L_N}(p^*)}$, define

$$\mathcal{O}_N(\zeta) = \{v \in \hat{\mathcal{O}}_N(\zeta) | \ \text{KL}(p^*||p_{\tilde{v}_N}) - \text{KL}(p^*||p_v) = \mathbb{E} \log \left(\frac{p_v(X)}{p_{\tilde{v}_N(X)}} \right) \leq -\eta \}$$

$$\phi_\zeta(t) = \mathbb{E} \exp \left(t \log \left(\frac{\sup_{v \in \mathcal{O}_N(\zeta)} p_v(X)}{p_{\tilde{v}_N}(X)} \right) \right).$$

Note $\phi_\zeta(t) \leq \mathbb{E} \exp(t|X|(\log\left(\nu_N^{-1}\right)))$ which is finite for small enough t by assumption. ϕ_ζ and the bound $\mathbb{E} \exp(t|X|(\log\left(\nu_n^{*-1}\right)))$ are partition functions for exponential families so, since they are finite for small t, they are C^∞ with derivatives obtained by exchanging differentiation and integration for small t by theorem 4.5 of van der Vaart [268]. In particular, for $t < C_{p^*}/(\log\left(\nu_N^{-1}\right))$ for some C_{p^*} that depends on p^*, defining another constant that depends on p^*, $C'_{p^*} < \infty$,

$$\begin{aligned} \phi''_\zeta(t) =& \mathbb{E} \left[\left(\log \left(\frac{\sup_{v \in \mathcal{O}_N(\zeta)} p_v(X)}{p_{\tilde{v}_N}(X)} \right) \right)^2 \exp \left(t \log \left(\frac{\sup_{v \in \mathcal{O}_N(\zeta)} p_v(X)}{p_{\tilde{v}_N}(X)} \right) \right) \right] \\ \leq& (\log\left(\nu_N^{-1}\right))^2 \mathbb{E} \left[|X|^2 \exp\left(t|X|(\log\left(\nu_N^{-1}\right)) \right) \right] \\ \leq& C'_{p^*} (\log\left(\nu_N^{-1}\right))^2. \end{aligned} \tag{B.32}$$

As well, for any $v_1, v_2 \in \hat{\mathcal{O}}_N(\zeta)$, for all $(k,b) \in \text{supp}_{L_N}(p^*)$, $|\log(v_{1,k,b}/v_{2,k,b})| < \log(1 + \rho_N) < \rho_N$. Thus, for all $v \in \mathcal{O}_N(\zeta)$, since if $p^*(X) > 0$ then all L_N-mer-base transitions in X are in $\text{supp}_{L_N}(p^*)$, $\mathbb{E} \log \left(\frac{\sup_{v \in \mathcal{O}_N(\zeta)} p_v(X)}{p_v(X)} \right) < \rho_N \mathbb{E}|X|$. So, defining $C''_{p^*} = \mathbb{E}|X|$,

$$\phi'_\zeta(0) = \mathbb{E} \log \left(\frac{\sup_{v \in \mathcal{O}_N(\zeta)} p_v(X)}{p_v(X)} \right) + \mathbb{E} \log \left(\frac{p_v(X)}{p_{\tilde{v}_N(X)}} \right) < \rho_N C''_{p^*} - \eta. \tag{B.33}$$

Putting things together we get, for small t,

$$\phi_\zeta(t) \leq 1 + t(\rho_N C''_{p^*} - \eta) + \frac{1}{2}t^2 C'_{p^*}(\log\left(\nu_N^{-1}\right))^2. \tag{B.34}$$

Picking $t = 2(\log\left(\nu_N^{-1}\right))^{-(1+\epsilon)}$ for some $\epsilon > 0$ gives, for large enough N, for any ζ, $\phi_\zeta(t) \leq 1 - \eta/(\log\left(\nu_N^{-1}\right))^{1+\epsilon} \leq \exp(-\eta/(\log\left(\nu_n^{-1}\right))^{1+\epsilon})$. Finally note that

$$\frac{(\log\left(\nu_N^{-1}\right))^{1+\epsilon}}{N^{1-\epsilon'}} = \left(\frac{(\log\left(\nu_N^{-1}\right))^{(1+\epsilon)/(1-\epsilon')}}{N}\right)^{1-\epsilon'} \to 0$$

by equation B.30 if ϵ, ϵ' are small enough. Now write, for large N' and positive constants

ϵ'', C, C', C'',

$$
\begin{aligned}
&\sum_{N>N'}^{\infty} \left(\#\hat{\mathcal{O}}_N\right) \left(\sup_{\zeta} \inf_{t>0} \phi_\zeta(t)\right)^N \leq \\
&\quad \sum_N \exp\left(4\left(\log\log\left(\nu_N^{-1}\right)\right)|\mathrm{supp}_{L_N}(p^*)| - \frac{N\eta}{\left(\log\left(\nu_N^{-1}\right)\right)^{1+\epsilon}}\right) \\
&\quad \leq \sum_N \exp\left(-\frac{N\eta}{\left(\log\left(\nu_N^{-1}\right)\right)^{1+\epsilon}}\left(1 - C\frac{|\mathrm{supp}_{L_N}(p^*)|\left(\log\left(\nu_N^{-1}\right)\right)^{1+\epsilon''}}{N}\right)\right) \\
&\quad \leq \sum_N \exp\left(-N^{\epsilon'}C'\right) \\
&\quad \leq \int_0^\infty dx \exp\left(-C''x^{\epsilon'}\right) \\
&\quad = \epsilon'^{-1}C''^{-1/\epsilon'} \int_0^\infty dx x^{1/\epsilon'-1}\exp(-x) \\
&\quad = \epsilon'^{-1}C''^{-1/\epsilon'}\Gamma(1/\epsilon') \\
&\quad < \infty
\end{aligned}
\tag{B.35}
$$

using the assumptions of the theorem and replacing ϵ by ϵ'' to absorb $\log\log\left(v_n^{-1}\right)$ (note one can make $\epsilon, \epsilon', \epsilon''$ as close to 1 as desired). This shows that all conditions of theorem 2 of Geman & Hwang[86] are satisfied. □

Remark B.6.2. *To pick viable* $(L_N)_N, (\nu_N)_N$*, note* $|supp_{L_N}(p^*)| \leq |\tilde{\mathcal{B}}||\mathcal{B}^o_{L_N}|$*, so, since*

$$
|\mathcal{B}^o_N| = \sum_{l=0}^{L_n} |\mathcal{B}|^l = \frac{|\mathcal{B}|^{L_N+1}-1}{|\mathcal{B}|-1} \leq |\mathcal{B}|^{L_N+1},
$$

we have $|supp_{L_N}(p)| \lesssim |\mathcal{B}|^{L_N}$*. Thus, as an example, for* $c_1, c_2 > 0$ *such that* $1 > c_1 + c_2$*,*

$L_N = \lceil c_1 \log N / \log |\mathcal{B}| \rceil$ and $\nu_N = e^{-Nc_2}$ satisfy condition B.3 o. We can see that without any a priori knowledge of $|supp_{L_N}(p^*)|$ we are forced to pick a very slow growing sequence $(L_N)_N$, and thus it is likely that are model class is too conservative for p^* whose support have cardinality far from the upper bound. By adapting L_N to the content of the data in addition to its cardinality, the Bayesian approach described in section B.6.3 does not suffer from this conceptual issue.

B.6.2 The projection algorithm

Fix L and ν for this section and define $\mathcal{S} = \{v \in \Delta^{\mathcal{B}_L^o}_{\tilde{\mathcal{B}}} \mid v_{k,b} \geq \nu \, \forall \, k, b\}$. Given data $X_1, \ldots, X_N$, any maximum likelihood estimate (MLE) in $\mathcal{M}_L$, v, has, for every L-mer k that is seen in the data, $v_{k,b} = \#(k,b) / \left(\sum_{b' \in \tilde{\mathcal{B}}} \#(k,b')\right)$ where $\#(k,b)$ is the number of times k is seen in the data immediately preceding b. If $\bar{v}$ is a MLE in $\mathcal{S}$, it will be shown that for each L-mer k that is seen in the data, $(\bar{v}_{k,b})_{b \in \tilde{\mathcal{B}}}$ is equal to a "projection" of $(v_{k,b})_{b \in \tilde{\mathcal{B}}}$ onto the smaller simplex $\{v_k \in \Delta_{|\tilde{\mathcal{B}}|} \mid v_{k,b} \geq \nu \, \forall b\}$. This projection is defined in algorithm 4, and the rest of this section will be devoted to its properties, including continuity, bounds, and proof of the above statement in proposition B.6.8. Some of these bounds will be used to prove the consistency of nonparametric Bayesian inference in section B.6.3. For ease of exposition, we will first present a conceptually simpler version of the projection algorithm, algorithm 3.

Algorithm 3 PROJECTION ALGORITHM I

Input : Non-negative numbers $(u_b)_{b\in\tilde{\mathcal{B}}}$, with $\sum_{b\in\tilde{\mathcal{B}}} u_b > 0$, and a positive number $\nu \leq 1/|\tilde{\mathcal{B}}|$.

Output: $(\bar{u}_b)_{b\in\tilde{\mathcal{B}}}$ such that $\sum_{b\in\tilde{\mathcal{B}}} \bar{u}_b = 1$ and $\bar{u}_b \geq \nu$ for all b.

$\bar{u}_b^{(0)} \leftarrow u_b/(\sum_{b'\in\tilde{\mathcal{B}}} u_{b'})$

$B^{(0)} \leftarrow |\{b \mid \bar{u}_b^{(0)} \leq \nu\}|$

$i \leftarrow 1$

while there exists a b with $\bar{u}_b^{(i-1)} < \nu$ **do**

  for $b \in \tilde{\mathcal{B}}$ **do**

    if $\bar{u}_b^{(i-1)} \leq \nu$ **then**

      $\bar{u}_b^{(i-1)} \leftarrow \nu$

    else

      $\bar{u}_b^{(i-1)} \leftarrow \left(1 - B^{(i-1)}\nu\right) u_b/(\sum_{b' \mid \bar{u}_{b'}^{(i-1)} > \nu} u_{b'})$.

  $B^{(i)} \leftarrow |\{b' \mid \bar{u}_{b'}^{(i)} \leq \nu\}|$

  $i \leftarrow i + 1$

for $b \in \tilde{\mathcal{B}}$ **do**

  $\bar{u}_b \leftarrow \bar{u}_b^{(i-1)}$

Proposition B.6.3. *Say algorithm 3 is applied to non-negative numbers $(u_b)_{b\in\tilde{\mathcal{B}}}$ with $\sum_b u_b > 0$. Define $(\bar{u}_b)_b$, $((\bar{u}_b^{(i)})_b)_i$ and $(B^{(i)})_i$ as in the algorithm. Say the algorithm terminates at step I.*

1) For all i, $\sum_{b=1}^{|\tilde{\mathcal{B}}|} \bar{u}_b^{(i)} = 1$.

2) If $(u_b)_b$ are scaled by a positive constant, the output $(\bar{u}_b)_b$ remains the same.

3) Say $(\bar{v}_b^{(i)})_b$ is the i-th iteration of algorithm 3 with input $(\bar{u}_b^{(j)})_b$. $(\bar{v}_b^{(i)})_b = (\bar{u}_b^{(j+i)})_b$.

*4) $I < |\tilde{\mathcal{B}}|$. The algorithm remains unchanged if the while loop were replaced by "**for** $i = 1, \ldots, |\tilde{\mathcal{B}}| - 1$ **do**".*

5) $\bar{u}_b \geq (1 - (|\tilde{\mathcal{B}}| - 1)\nu)\bar{u}_b^{(0)}$.

Proof. Results 1 and 2 are clear. For 3, note that if both $\bar{u}_b^{(j)}$ and $\bar{u}_{b'}^{(j)}$ are greater than ν, then $\bar{u}_b^{(j)}/\bar{u}_{b'}^{(j)} = u_b/u_{b'}$. Thus, if $\bar{u}_b^{(j)} > \nu$,

$$\bar{u}_b^{(j+1)} = \left(1 - B^{(j)}\nu\right)\bar{u}_b^{(j)} \Big/ \left(\sum_{b' \mid \bar{u}_{b'}^{(j)} > \nu} \bar{u}_{b'}^{(j)}\right) = \bar{v}_b^{(1)}.$$

Similar logic may be used to show $(\bar{v}_b^{(2)})_b = (\bar{u}_b^{(j+2)})_b$ and so on.

To see 4, notice that for every $i \leq I$, at least one b has $\bar{u}_b^{(i)} = \nu$ while $\bar{u}_b^{(i-1)} < \nu$. Thus, $(\hat{B}^{(i)})_{i=0}^{I} := (|\{b' \mid \bar{u}_{b'}^{(i)} \leq \nu\}|)_{i=0}^{I}$ is a strictly increasing sequence. $\hat{B}^{(i)} \leq |\tilde{\mathcal{B}}|$ as $\nu \leq 1/|\tilde{\mathcal{B}}|$. If $\hat{B}^{(I)} = |\tilde{\mathcal{B}}|$ then $\nu = 1/|\tilde{\mathcal{B}}|$ and, by property 1, $\hat{B}^{(i)} \neq |\tilde{\mathcal{B}}| - 1$ for every i. In any case, the sequence $(\hat{B}^{(i)})_{i=0}^{I}$ may take on at most $|\tilde{\mathcal{B}}|$ values (including 0) and thus $I < |\tilde{\mathcal{B}}|$. The second statement of 4 follows from the fact that for all b, $\bar{u}_b \geq \nu$ and thus would remain unaltered by the procedure in the while statement.

Finally, for 5, first say $\bar{u}_b > \nu$ and note that $B^{(I-1)} < |\tilde{\mathcal{B}}|$ (otherwise the algorithm is terminated

or property 1 is violated).

$$\bar{u}_b = (1 - B^{(I-1)}\nu)\frac{u_b}{\left(\sum_{b' \mid \bar{u}_{b'}^{(I-1)} > \nu} u_{b'}\right)} \geq (1 - B^{(I-1)}\nu)\frac{u_b}{(\sum_{b'} u_{b'})} \geq (1 - (|\tilde{\mathcal{B}}| - 1)\nu)\bar{u}_b^{(0)}.$$

Now say $\bar{u}_b = \nu$. Call i' the first step such that $\bar{u}_b^{(i')} = \nu$. If $i' = 0$ or $i' = 1$ then $\bar{u}_b = \nu \geq (1 - (|\tilde{\mathcal{B}}| - 1)\nu)\nu \geq (1 - (|\tilde{\mathcal{B}}| - 1)\nu)\bar{u}_b^{(0)}$. Finally, if $i' > 1$ then

$$\bar{u}_b = \nu \geq \bar{u}_b^{(i'-1)} = \left(1 - B^{(i'-2)}\nu\right) u_b / \left(\sum_{b' \mid \bar{u}_{b'}^{(i'-2)} > \nu} u_{b'}\right) \geq \left(1 - (|\tilde{\mathcal{B}}| - 1)\nu\right) \bar{u}_b^{(0)}.$$

Thus in all cases $\bar{u}_b \geq (1 - (|\tilde{\mathcal{B}}| - 1)\nu)\bar{u}_b^{(0)}$. □

We now turn to the main projection algorithm.

Algorithm 4 PROJECTION ALGORITHM II

Input : Non-negative numbers $(u_b)_{b=1}^{|\tilde{\mathcal{B}}|}$, with $\sum_{b=1}^{|\tilde{\mathcal{B}}|} u_b > 0$, and a positive number $\nu \leq 1/|\tilde{\mathcal{B}}|$.

Output: $(\bar{u}_b)_{b=1}^{|\tilde{\mathcal{B}}|}$ such that $\sum_{b=1}^{|\tilde{\mathcal{B}}|} \bar{u}_b = 1$ and $\bar{u}_b \geq \nu$ for all b.

$\bar{u}_b^{(0)} \leftarrow u_b / (\sum_{b'=1}^{|\tilde{\mathcal{B}}|} u_{b'})$

$\mathcal{C}^{(0)} \leftarrow \emptyset$

$i \leftarrow 1$

while there is a $b \notin \mathcal{C}^{(i-1)}$ with $\bar{u}_b^{(i-1)} \leq \nu$ **do**

    Pick $b^{(i-1)} \in \{b \mid \bar{u}_b^{(i-1)} \leq \nu\} \setminus \mathcal{C}^{(i-1)}$

    $\mathcal{C}^{(i)} \leftarrow \mathcal{C}^{(i-1)} \cup \{b^{(i-1)}\}$

    for $b = 1, \ldots, |\tilde{\mathcal{B}}|$ **do**

        if $b \in \mathcal{C}^{(i)}$ **then**

            $\bar{u}_b^{(i)} \leftarrow \nu$

        else

            $\bar{u}_b^{(i)} \leftarrow (1 - i\nu)\, u_b / \left(\sum_{b' \notin \mathcal{C}^{(i)}} u_{b'}\right)$

    $i \leftarrow i + 1$

for $b = 1, \ldots, |\tilde{\mathcal{B}}|$ **do**

    $\bar{u}_b \leftarrow \bar{u}_b^{(i-1)}$

An example run of algorithm 4 is visualized in figure B.4 (top row). Clearly this algorithm re-

turns $\bar{u}_b = \nu$ if $\nu = 1/|\tilde{\mathcal{B}}|$ and all the following results are trivial. Thus below we will assume $\nu < 1/|\tilde{\mathcal{B}}|$.

Remark B.6.4. *We will first consider an alternative representation of the algorithm.*

Given a $\mathcal{C} \subset \tilde{\mathcal{B}}$, call

$$u^{\mathcal{C}} = \nu \frac{\sum_{b \notin \mathcal{C}} u_b}{1 - |\mathcal{C}|\nu}$$

and if $u^{\mathcal{C}} > 0$, define

$$\bar{u}_b^{\mathcal{C}} := (1 - |\mathcal{C}|\nu) u_b / \left(\sum_{b' \notin \mathcal{C}} u_{b'} \right) = \nu u_b / u^{\mathcal{C}}$$

for $b \notin \mathcal{C}$ and $\bar{u}_b^{\mathcal{C}} = \nu$ for $b \in \mathcal{C}$; so one gets $\bar{u}_b^{(\tilde{i})} = \bar{u}_b^{\mathcal{C}^{(\tilde{i})}}$ at each iteration $\tilde{i}$. If $b \notin \mathcal{C}$, $\bar{u}_b^{\mathcal{C}} \leq \nu$ if and only if $u_b \leq u^{\mathcal{C}}$.

Say $b \notin \mathcal{C}$ and call $\mathcal{C}' := \{b\} \cup \mathcal{C}$.

$$u^{\mathcal{C}'} - u^{\mathcal{C}} = \nu \frac{\nu \left(\sum_{b' \notin \mathcal{C}} u_{b'} \right) - u_b \left(1 - |\mathcal{C}|\nu\right)}{\left(1 - |\mathcal{C}'|\nu\right) \left(1 - |\mathcal{C}|\nu\right)} = \frac{\nu}{1 - |\mathcal{C}'|\nu} \left(u^{\mathcal{C}} - u_b \right).$$

Thus $u^{\mathcal{C}'} \geq u^{\mathcal{C}}$ if and only if $u_b \leq u^{\mathcal{C}}$ with equality if and only if $u_b = u^{\mathcal{C}}$.

We can see that at iteration i the next $b^{(i-1)}$ is chosen from $\{b \mid \bar{u}_b^{(i-1)} \leq \nu\} \setminus \mathcal{C}^{(i-1)} = \{b \,|u_b \leq u^{\mathcal{C}^{(i-1)}}\} \setminus \mathcal{C}^{(i-1)}$, i.e. from those b with u_b below the threshold $u^{\mathcal{C}^{(i-1)}}$. Thus, $u^{\mathcal{C}^{(0)}} \leq u^{\mathcal{C}^{(1)}} \leq \ldots$. This is reflected in figure B.4 (bottom row).

By induction (or from inspection of figure B.4), one may show that all the elements b of $\mathcal{C}^{(i)}$ must have u_b below the threshold $u^{\mathcal{C}^{(i-1)}}$ and the algorithm is complete only when all b with u_b below the threshold $u^{\mathcal{C}^{(i)}}$ are inside $\mathcal{C}^{(i)}$. In other words, for $i < I$ (where I is the final iteration) we have $\mathcal{C}^{(i)} \subsetneq$

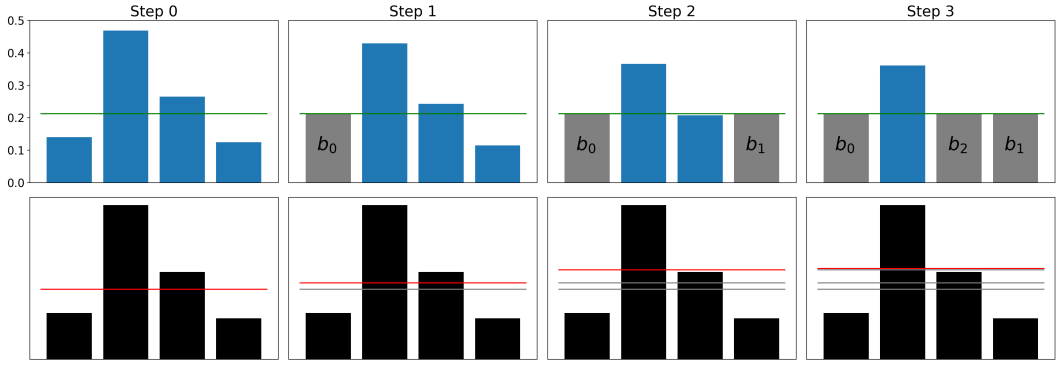


Figure B.4: Example application of algorithm 4. $(\bar{u}_b^{(i)})_b$ at the end of each step of the algorithm is shown on the top row with ν in green and those elements in $\mathcal{C}^{(i)}$ in grey. $(u_b)_{b=1}^{|\tilde{\mathcal{B}}|}$ is shown as black bars in the plots in the bottom row with $u^{\mathcal{C}^{(i)}}$ shown as a red line. $u^{\mathcal{C}^{(j)}}$ for previous steps $j < i$ are also shown on the bottom row as grey lines. The scale of the inputs $(u_b)_{b=1}^{|\tilde{\mathcal{B}}|}$ is of no consequence for the algorithm.

$\{b \mid u_b \leq u^{\mathcal{C}^{(i)}}\}$, *and* $\mathcal{C}^{(I)} = \{b \mid u_b \leq u^{\mathcal{C}^{(I)}}\}$.

The important points from the above remark are summarized as:

Lemma B.6.5. *1) Given a* $\mathcal{C} \subset \tilde{\mathcal{B}}$, *say* $b \notin \mathcal{C}$ *and call* $\mathcal{C}' := \{b\} \cup \mathcal{C}$. $u^{\mathcal{C}'} \geq u^{\mathcal{C}}$ *if and only if* $u_b \leq u^{\mathcal{C}}$ *with equality if and only if* $u_b = u^{\mathcal{C}}$.

2) $u^{\mathcal{C}^{(0)}} \leq u^{\mathcal{C}^{(1)}} \leq \dots$.

3) If the algorithm ends on step I, $\mathcal{C}^{(i)} \subseteq \{b \mid u_b \leq u^{\mathcal{C}^{(i-1)}}\}$ *for all* $i \leq I$, $\mathcal{C}^{(i)} \subsetneq \{b \mid u_b \leq u^{\mathcal{C}^{(i)}}\}$ *for* $i < I$, *and* $\mathcal{C}^{(I)} = \{b \mid u_b \leq u^{\mathcal{C}^{(I)}}\}$.

Proposition B.6.6. *Say algorithm 4 is applied to non-negative numbers* $(u_b)_{b \in \tilde{\mathcal{B}}}$ *with* $\sum_b u_b > 0$. *Define* $(\bar{u}_b)_b$, $((\bar{u}_b^{(i)})_b)_i$ *and* $(\mathcal{C}^{(i)})_i$ *as in the algorithm. Say the algorithm terminates at step* I.

1) The output of the algorithm is the same regardless of the choice of $(b_0, b_1, \dots)$.

2) The output of the algorithm is the same as that of algorithm 3.

3) we can replace lines 4 and 5 of algorithm 4 with

4: ***while*** *there is a* $b \notin \mathcal{C}^{(i-1)}$ *with* $\bar{u}_b^{(i-1)} < \nu$ ***do***

5: ***Pick*** $b^{(i-1)} \in \{b \mid \bar{u}_b^{(i-1)} < \nu\} \setminus \mathcal{C}^{(i-1)}$

and receive the same output. With this adjustment, $I < |\tilde{\mathcal{B}}|$.

4) Say $b \notin \mathcal{C}^{(i)}$. $\bar{u}_b^{(i-1)} - \bar{u}_b^{(i)} \leq |\tilde{\mathcal{B}}|(\nu - \bar{u}_{b^{(i-1)}}^{(i-1)})$ *so that* $\bar{u}_b^{(i-1)}$ *is close to* $\bar{u}_b^{(i)}$ *if* $\bar{u}_{b^{(i-1)}}^{(i-1)}$ *is close to* ν.

Proof. 1) Say the choices $(b^{(0)}, \ldots, b^{(I)})$ were made when running the algorithm. Consider a different sequence of choices $(b'^{(0)}, \ldots, b'^{(I')})$ to produce $\mathcal{C}'^{(I')}$. Note that by lemma B.6.5, $\mathcal{C} := \mathcal{C}^{(I)} = \{b \,|u_b \leq u^{\mathcal{C}^{(I)}}\}$ and $\mathcal{C}' := \mathcal{C}'^{(I')} = \{b \mid u_b \leq u^{\mathcal{C}'^{(I')}}\}$. Without loss of generality assume $\mathcal{C} \supsetneq \mathcal{C}'$ so $u^{\mathcal{C}} > u^{\mathcal{C}'}$. We will show that this leads to a contradiction. Pick the smallest $i \leq I$ such that $u^{\mathcal{C}^{(i)}} > u^{\mathcal{C}'}$. Then $u^{\mathcal{C}^{(i-1)}} \leq u^{\mathcal{C}'}$, so by lemma B.6.5, $\mathcal{C}^{(i)} \subseteq \mathcal{C}'$.

Pick an enumeration $(\tilde{b}_1, \ldots, \tilde{b}_J) = (\mathcal{C}' \setminus \mathcal{C}^{(i)})$. $u_{\tilde{b}_1} \leq u^{\mathcal{C}'} \leq u^{\mathcal{C}^{(i)}}$ so $u^{\mathcal{C}^{(i)} \cup \{\tilde{b}_1\}} \geq u^{\mathcal{C}^{(i)}}$. By induction, one may show that $u^{\mathcal{C}'} = u^{\mathcal{C}^{(i)} \cup (\mathcal{C}' \setminus \mathcal{C}^{(i)})} \geq u^{\mathcal{C}^{(i)} \cup \{\tilde{b}_1, \ldots, \tilde{b}_{J-1}\}} \geq \cdots \geq u^{\mathcal{C}^{(i)} \cup \{\tilde{b}_1\}} \geq u^{\mathcal{C}^{(i)}}$. This contradicts the choice of i above. Thus, $\mathcal{C} = \mathcal{C}'$ and $I = |\mathcal{C}| = |\mathcal{C}'| = I'$. Moreover, since the final output $(\bar{u}_b)_{b=1}^{|\tilde{\mathcal{B}}|}$ of the algorithm can be defined purely in terms of the final set $\mathcal{C}^{(i)}$, the output must be identical among runs of the algorithm.

2) Consider choosing $(b^{(0)}, \ldots, b^{(i)})$ as such: first pick $\{b^{(0)}, \ldots, b^{(i_1-1)}\} = \{b \mid \bar{u}_b^{(0)} \leq \nu\}$, which we know can be done since by lemma B.6.5, $\bar{u}_b^{(0)} \leq \nu$ if and only if $u_b \leq u^{\mathcal{C}^{(i_1)}}$ and $u^{\mathcal{C}^{(i_1)}} \leq u^{\mathcal{C}^{(i_1+1)}} \leq \ldots$. This is equivalent to one step of the while loop of algorithm 3. Then choose $\{b^{(i_1)}, \ldots, b^{(i_2-1)}\} = \{b \mid \bar{u}_b^{(i_1)} \leq \nu\} \setminus \mathcal{C}^{(i_1)}$, which we can do by similar logic. This is

equivalent to the second step of the while loop of algorithm 3. Continuing the construction in the same way, by conclusion (1) above, we get that the outputs of algorithms 3 and 4 are identical.

3) Note, by lemma B.6.5, picking a $b^{(i-1)}$ with $\bar{u}^{(i-1)}_{b^{(i-1)}} = \nu$ gives $u^{\mathcal{C}^{(i)}} = u^{\mathcal{C}^{(i-1)}}$ and $\bar{u}^{(i)}_b = \bar{u}^{(i-1)}_b$ for all b. Say $(b_0, \dots, b_i)$, $i < I$ are selected in the algorithm such that $\bar{u}^{(j)}_{b_j} < \nu$ for each $j \leq i$ and all $b \in \{b \mid \bar{u}^{(i)}_b \leq \nu\} \setminus \mathcal{C}^{(i)}$ have $\bar{u}^{(i)}_b = \nu$, then $(\bar{u}^{(i+1)}_{b'})_{b'} = (\bar{u}^{(i)}_{b'})_{b'}$ and all $b \in \{b \mid \bar{u}^{(i+1)}_b \leq \nu\} \setminus \mathcal{C}^{(i+1)}$ have $\bar{u}^{(i)}_b = \nu$. Continuing by induction demonstrates property (3). That $I < |\tilde{\mathcal{B}}|$ follows by the same logic as conclusion (4) in proposition B.6.3 on algorithm 3.

4) Say $b \notin \mathcal{C}^{(i)}$,

$$\begin{aligned}
\bar{u}^{(i-1)}_b - \bar{u}^{(i)}_b =& \nu u_b \left(1/u^{\mathcal{C}^{(i-1)}} - 1/u^{\mathcal{C}^{(i)}}\right) \\
=& \frac{u_b}{\sum_{b' \notin \mathcal{C}^{(i)}} u_{b'}} \frac{\nu(\sum_{b' \notin \mathcal{C}^{(i-1)}} u_{b'}) - u_{b^{(i-1)}}(1-(i-1)\nu)}{\sum_{b' \notin \mathcal{C}^{(i-1)}} u_{b'}} \\
=& \bar{u}^{(i)}_b (1-i\nu)^{-1}(\nu - \bar{u}^{(i-1)}_{b^{(i-1)}}) \\
\leq& |\tilde{\mathcal{B}}|(\nu - \bar{u}^{(i-1)}_{b^{(i-1)}})
\end{aligned} \tag{B.36}$$

with the last inequality since $i \leq |\tilde{\mathcal{B}}| - 1$ and $\nu \leq 1/|\tilde{\mathcal{B}}|$.

□

Next we show that the projection defined by algorithm B.6.6 is continuous.

Lemma B.6.7. *Say $0 < \nu \leq 1/|\tilde{\mathcal{B}}|$ and $((u_{j,b})_{b=1}^{|\tilde{\mathcal{B}}|})_{j=1}^{\infty}$ is a sequence of sets of non-negative numbers, each with at least one positive element, with $u_{j,b} \to u_b$ for each b as $j \to \infty$, where $(u_b)_{b=1}^{|\tilde{\mathcal{B}}|}$ is set of non-negative numbers with at least one positive element. Apply algorithm 3 or 4 to each set*

$((u_{j,b})_{b=1}^{|\tilde{\mathcal{B}}|})_{j=1}^{\infty}$ *to get* $((\bar{u}_{j,b})_{b=1}^{|\tilde{\mathcal{B}}|})_{j=1}^{\infty}$ *and to* $(u_b)_{b=1}^{|\tilde{\mathcal{B}}|}$ *to get* $(\bar{u}_b)_{b=1}^{|\tilde{\mathcal{B}}|}$. *Then* $\bar{u}_{j,b} \to \bar{u}_b$ *for all* b.

Proof. Define $\bar{u}_{j,b}^{(i)}$ as in the steps of algorithm 4, with $b^{(0)}, b^{(1)}, \ldots$ to be defined below. Say $\bar{u}_{b^{(0)}}^{(0)} < \nu$. Eventually, $\bar{u}_{j,b^{(0)}}^{(0)} < \nu$ and thus it becomes possible to pick $b^{(0)}$ in the first step of the algorithm for all large enough j. Then, we get $\bar{u}_{j,b^{(0)}}^{(1)} = \nu = \bar{u}_{b^{(0)}}^{(1)}$. For $b \neq b^{(0)}$, $\bar{u}_{j,b}^{(1)} = \nu u_{j,b}/u_j^{\mathcal{C}^{(1)}}$ as defined as part of lemma B.6.5. $u^{\mathcal{C}^{(1)}}$ is a continuous function of $(u_b)_{b=1}^{|\tilde{\mathcal{B}}|}$ so that $\bar{u}_{j,b}^{(1)} \to \bar{u}_b^{(1)}$ for all b. Using the same logic, for large enough j, we may pick an $b^{(1)}$ with $\bar{u}_{b^{(1)}}^{(1)} < \nu$ and see $\bar{u}_{j,b}^{(2)} \to \bar{u}_b^{(2)}$ for all b. We may continue as such until the algorithm terminates for $(u_b)_{b=1}^{|\tilde{\mathcal{B}}|}$ by property (3) in proposition B.6.6. Thus, for some i, we have that $\bar{u}_{j,b}^{(i)} \to \bar{u}_b$ for all b.

Note each $(u_{j,b}^{(i)})_{b=1}^{|\tilde{\mathcal{B}}|}$ may require another $|\tilde{\mathcal{B}}| - i - 1$ steps for the algorithm to complete. For large enough j, we have the implication $\bar{u}_b > \nu \implies \bar{u}_{j,b}^{(i)} > \nu$ for all b so that if for a b, $\bar{u}_{j,b}^{(i)} < \nu$, then $\bar{u}_{j,b}^{(i)} \to \bar{u}_b = \nu$. Applying property (4) in proposition B.6.6 to each of the remaining steps of the algorithm applied to $(u_{j,b})_{b=1}^{|\tilde{\mathcal{B}}|}$ for high enough j, considering $\bar{u}_{j,b}^{(i)} \to \bar{u}_b$ for all b, we can see that $\bar{u}_{j,b} \to \bar{u}_b$ for all b. □

Finally, we can show that the projection algorithms 3 and 4 indeed return the MLE on the sieve $\mathcal{S}$, given observed kmer transition counts.

Proposition B.6.8. *Given data* $X_1, \ldots, X_N$, *a lag* L, *and a positive number* $\nu < 1/|\tilde{\mathcal{B}}|$, *say* $\bar{v}$ *is an MLE in* $\mathcal{S} := \{v \in \Delta_{|\tilde{\mathcal{B}}|}^{\mathcal{B}_L^o} | \, \forall k, b, v_{k,b} \geq \nu\}$. *For every* L*-mer* k *that has been seen in the data,* $(\bar{v}_{k,b})_{b \in \tilde{\mathcal{B}}}$ *is equal to the output of algorithm 3 or 4 applied to* $(\#(k,b))_{b \in \tilde{\mathcal{B}}}$ *where* $\#(k,b)$ *is the number of times* k *is seen in the data immediately preceding* b.

Proof. The likelihood of the data under a $p_v \in \mathcal{M}_L$ is

$$\sum_k \sum_b \#(k,b) \log(v_{k,b}).$$

Thus, the MLE in $\mathcal{S}$ can be found by finding, for each k with $\#k > 0$,

$$\text{argmax}_{v_k \in \Delta^{(0)}} \sum_b \#(k,b) \log(v_{k,b})$$

where $\Delta^{(0)} := \{v_k \in \Delta_{\tilde{\mathcal{B}}} \mid \text{for all } b, v_{k,b} \geq \nu\}$.

Say k has been seen in the data, so the MLE on $\Delta_{\tilde{\mathcal{B}}}$, $v_k^{(0)}$, is unique and satisfies $v_{k,b}^{(0)} \propto \#(k,b)$. Call $\hat{v}_k$ an MLE on $\Delta^{(0)}$. Say $v_k^{(0)} \notin \Delta^{(0)}$ so that for some b, $v_{k,b}^{(0)} < \nu_n$. By the uniqueness of the MLE, the likelihood of the data under $v_k^{(0)}$ must be strictly greater than under $\hat{v}_k$. Connecting $\hat{v}_k$ and $v_k^{(0)}$ by a line, considering the concavity of the log likelihood function, the likelihood must be decreasing from $v_k^{(0)}$ to $\hat{v}_k$. As the likelihood function is analytic and not constant on the line, it must be strictly decreasing. Thus the line cannot intersect $\Delta^{(0)}$ except at $\hat{v}_k$. For every b, $\lambda \hat{v}_{k,b} + (1-\lambda) v_{k,b}^{(0)} \geq \nu$ for all $\lambda \in [0,1]$ if $v_{k,b}^{(0)} \geq \nu$; for all $\lambda \in [c,1]$ for a $c < 1$ if $v_{k,b}^{(0)} < \nu_n$ and $\hat{v}_{k,b} > \nu$; and only for $\lambda = 1$ if $v_{k,b}^{(0)} < \nu_n$ and $\hat{v}_{k,b} = \nu$. Therefore, for some $b^{(0)}$ such that $v_{k,b^{(0)}}^{(0)} < \nu$ we have $\hat{v}_{k,b^{(0)}} = \nu$.

Call $v_k^{(1)}$ the MLE on $\{v_k \in \Delta_{\tilde{\mathcal{B}}} \mid v_{k,b^{(0)}} = \nu\}$. Using Lagrange multipliers again, one may see that

$$v_{k,b}^{(1)} = (1-\nu) \frac{(k,b)}{\sum_{b \neq b^{(1)}} \#(k,b)}$$

for $b \neq b^{(0)}$. Note that $v_k^{(1)}$ is the result of one step of applying algorithm 4 to $v_k^{(0)}$ using $b^{(0)}$. Call $\Delta^{(1)} := \{v_k \in \Delta_{\tilde{\mathcal{B}}} \mid \text{for all } b, v_{k,b} \geq \nu \text{ and } v_{k,b^{(1)}} = \nu\}$ so $\hat{v}_k \in \Delta^{(1)}$. One may perform the same analysis as above to see that if for some b, $v_{k,b}^{(1)} < \nu$, then there is a $b^{(1)}$ such that $v_{k,b^{(1)}}^{(1)} < \nu$ and $\hat{v}_{k,b^{(1)}} = \nu$.

We may then construct $v_k^{(2)}, v_k^{(3)}, \ldots$ by applying algorithm 4, picking $b^{(i)}$. Defining $\Delta^{(i)}$ in analogy to $\Delta^{(0)}$ and $\Delta^{(1)}$, the algorithm stops at step i when $v_k^{(i)} \in \Delta^{(i)}$ and $\hat{v}_k = v_k^{(i)} = \bar{v}_k$. That $\bar{v}_k$ is unique follows from property (2) in remark B.6.6. □

B.6.3 Bayesian consistency

In this section we take a Bayesian approach to inferring a subexponential p^* from data $X_1, X_2, \cdots \sim p^*$ iid. We put a prior on L, with support over all $L > 0$, to construct a nonparametric Bayesian model and then study the consistency and concentration rate of its posterior. Recall that the Bernstein von-Mises theorem states that given some regularity conditions, for a Bayesian parametric model, the posterior concentrates in a neighborhood centered at the data-generating distribution, with radius proportional to $1/\sqrt{N}$. For nonparametric models in general, and (as we shall see) the BEAR model in particular, the concentration rate of the posterior can be strictly slower than $\sqrt{N}$ [87,221].

In order to guarantee consistency and derive a concentration rate, we will, instead of placing a prior directly on L, place a prior on sieves constructed similarly to those in section B.6.1. In particu-

lar, define for all $L, \nu' > 0$ and $\nu > 0$ the sieve

$$\mathcal{S}(\nu', \nu, L) = \{v \in \Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}_L^o} \mid \forall k,\ v_{k,\$} \geq \nu \text{ and } v_{k,b} \geq \nu' \, \forall b \in \mathcal{B}\}$$

where ν is a lower bound on the stop transition probability and ν' is a lower bound on all other transitions. In particular, we will define a prior over the sieves that depends on how well a distribution from each sieve can match p^*. Define the sieve approximation mismatch

$$\xi(\nu', \nu, L) = \min_{v \in \mathcal{S}(\nu', \nu, L)} \mathbb{E} \log^2 \left(\frac{p^*(X)}{p_v(X)} \right).$$

In the next section, we will show that we can guarantee ξ is sufficiently small by using the fact that p^* is subexponential. Here, we define the prior.

We may now define our prior:

Condition B.6.9. *Assume, for monotonic sequences $(\nu_m)_m, (L_m)_m$, and a distribution on the natural numbers π,*

$$\log \left(\nu_m^{-1} \right) \sim m^{c_1}$$

$$|\mathcal{B}|^{L_m} \sim m^{c_2}$$

$$\xi(\nu_m, \nu_m, L_m) \lesssim m^{-c_3}$$

$$\log \pi(m) \sim -m^{\omega}$$

with $c_1, c_2, c_3 > 0$ *and* $1 > c_1 + c_2$. c_3 *must obey the following condition: calling* $\delta = 1 - \frac{1-(c_1+c_2)}{c_3/2}$, $\delta > 0$ *and* $(1-\delta)^{-1}(c_1+c_2) \geq \omega > c_1 + c_2$. *Consider positive numbers* $(\alpha_{k,b})_{L \geq 1, k \in \mathcal{B}^o_L, b \in \tilde{\mathcal{B}}}$ *such that* $\sup \alpha_{k,b} < \infty$ *and* $\inf \alpha_{k,b} > 0$. *Consider a prior* Π *on the disjoint union* $\sqcup_{m=1}^{\infty} S(0, \nu_m, L_m)$ *that factorizes as such:*

$$\Pi(p_v) = \pi(m) \prod_{k \in \mathcal{B}^o_{L_m}} \Pi_k(v_k) \text{ if } p_v \in \mathcal{S}_m.$$

where for a $k \in \mathcal{B}^o_{L_m}$, Π_k *is a restricted and renormalized Dirichlet*$(\alpha_{k,b})_{b \in \tilde{\mathcal{B}}}$ *prior on the simplex in* $\mathcal{S}(0, \nu_m, L_m)$ *corresponding to transition coefficients out of* k.

Note as well the difference between the sieve we approximate p^* with $(S(\nu_m, \nu_m, L_m))$ and the one our prior is defined over $(S(0, \nu_m, L_m))$. It is best to consider the constraints on c_1, c_2, ω with the fact that c_3 is limited in the values it may take on by how well p^* can be approximated by finite lag Markov models. Our main result will be the consistency of the posterior under this prior and the calculation of its concentration rate.

Remark B.6.10. *Using the techniques in section B.6.2, we can see that the maximum a posteriori estimate on each sieve* $\mathcal{S}(0, \nu_m, L_m)$ *has, for every* k *that has been seen in the data,*

$$v_{k,b} \propto \#(k,b) + \alpha_{k,b}$$

if $\frac{\#(k,\$)+\alpha_{k,\$}}{\sum_{b' \neq \$}\left(\#(k,b')+\alpha_{k,b'}\right)} \geq \nu_m$*; otherwise,* $v_{k,\$} = \nu_m$ *but we still have* $v_{k,b} \propto \#(k,b) + \alpha_{k,b}$ *for* $b \in \mathcal{B}$. *One may then compare the densities of the maximum a posteriori estimators in each sieve across*

L to get the maximum a posteriori estimator of the entire posterior.

We now discuss two interpretations of this prior. On the one hand,

$$\Pi = \sum_{L=1}^{\infty} \sum_{m \mid L_m = L} \pi_{\text{lag}}(m) \Pi(\cdot \mid \mathcal{S}(0, \nu_m, L_m))$$

and thus, since $\mathcal{S}(0, \nu_m, L_m) \subset \mathcal{M}_{L_m}$, and the fact that multiple m correspond to the same L_m, the prior can be interpreted as similar to putting a prior on the lag, with the standard Dirichlet priors on each $\mathcal{M}_L$, but with the prior having a "staircase" shape for very small stopping probabilities. On the other hand, we have carefully chosen the values of ν_m and L_m in order to balance the size of $\mathcal{S}_m$ against the amount of information about p^* received from m datapoints. How this works will become clear in the proof of theorem B.6.16.

In section B.6.3 we will show that there exists a c_3 such that $\xi(\nu_m, \nu_m, L_m) \lesssim m^{-c_3}$, i.e., p^* may be efficiently approximated by the sieves. Then we will derive our main result with the concentration rate in section B.6.3. Finally we describe how to use this result in practice on real data in section B.6.3. Throughout we will consider a data generating distribution p^* and all expectations will be with respect to the data generating distribution unless otherwise stated.

Approximating subexponential sequence distributions

In this section we will be interested in finding an asymptotic upper bound for $\xi(\nu_m, \nu_m, L_m)$ of the form m^{-c_3}, thus showing that a prior as in Condition B.6.9 exists (proposition B.6.13). The result relies on the assumption that p^* is subexponential; our main consistency result (theorem

B.6.16) would only require $\mathbb{E}|X|^2 < \infty$ if Condition B.6.9 were somehow otherwise satisfied. In its essence, this section is about constructing approximations to subexponential sequence distributions, with control not only over the expected log ratio of p^* and the approximating distribution p – the KL divergence, $\mathbb{E}\log(p^*(X)/p(X))$ – but also over the variance of this log ratio – i.e. control of $\mathbb{E}\log^2(p^*(X)/p(X))$. We will make use of lemma B.2.3 but need another construction and technical lemma.

Note that if p^* is a distribution on S and $X \in S$,

$$p^*(X) = p^*(X_1 \dots)p^*(X_{1:2} \dots | X_1 \dots) \dots p^*(X_{1:|X|} | X_{1:|X|-1} \dots)$$

where, recall, for a sequence Y, possibly not terminated by \$, $p^*(Y \dots) = p^*(\{X \in S \mid X_i = Y_i \; \forall i \leq |Y|\})$. Thus a probability distribution on S may be described by its infinite-lag transition probabilities $p^*((Y, b) \dots | Y \dots)$ for sequences Y not terminated by \$ and $b \in \tilde{\mathcal{B}}$, ignoring those Y with $p^*(Y \dots) = 0$. Infinite-lag transition probabilities were considered in the construction of p^*_L in proposition B.2.3. Below we will be interested in constructing another distribution from p by projecting, for some L, the transition probabilities at each Y with $|Y| < L$ onto $\{v \in \Delta_{|\tilde{\mathcal{B}}|} \mid v_b \geq \nu^* \; \forall \, b\}$. This first lemma will be used to guarantee the existence of this distribution.

Lemma B.6.11. *Say p^* is a probability distribution on S. Given a lag L and positive numbers $((v_{X,b})_{b \in \tilde{\mathcal{B}}})_{l \in \{0,\dots,L-1\}, X \in \mathcal{B}^l}$ with $\sum_b v_{X,b} = 1$ for all X, there is a p^{*L} such that for all sequences*

Y not terminated by \$,

$$p^{*L}((Y,b)\dots|Y\dots) = \begin{cases} v_{Y,b} & \text{if } |Y| < L \\ p^*(Yb\dots|Y\dots) & \text{if } |Y| \geq L \text{ and } p^*(Y\dots) > 0. \end{cases} \tag{B.37}$$

Proof. For $X \in S, |X| \leq L$ define

$$p^{*L}(X) = \prod_{i=1}^{|X|} v_{X_{1:i-1},X_i}.$$

For $Y \in \mathcal{B}^L$ with $p(Y\dots) = 0$, define

$$p^{*L}((Y,\$)) = \prod_{i=1}^{L} v_{Y_{1:i-1},Y_i}$$

and $p^{*L}(X) = 0$ for $X \in S$ with $X_{1:L} = Y$ and $X_{L+1} \neq \$$. Finally, if $p^*(Y\dots) > 0$ define, for all $X \in S$ with $X_1 \cdots X_L = Y$,

$$p^{*L}(X) = \left(\prod_{i=1}^{L} v_{Y_{1:i-1},Y_i}\right) p^*(X|Y\dots).$$

It is not difficult to check that p^{*L} is well defined and satisfies the requirements in the statement (Fig. B.5).

□

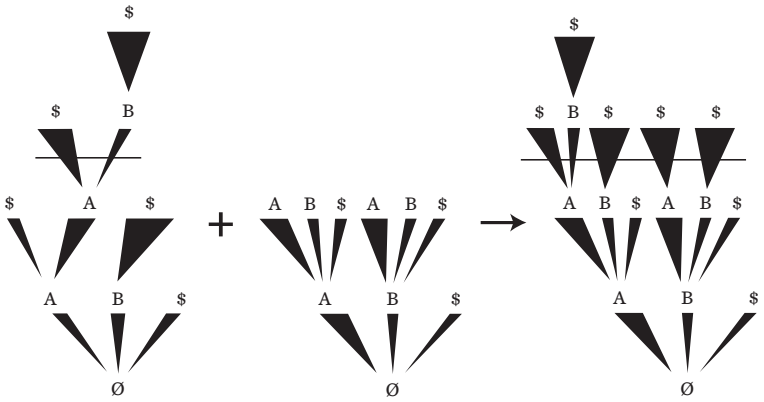


Figure B.5: Example application of this construction to the distribution p^* on the left, with the v represented in the center. Transition probabilities for kmers smaller than $L = 2$ are those defined by v while those after are those of the original distribution. Thickness of lines denote probability of particular transition.

Finally, we write a technical lemma:

Lemma B.6.12. *There exists a positive constant C such that for any p^* and p that are distributions over S,*

$$\mathbb{E}_{p^*} \log^2 \left(\frac{p^*(X)}{p(X)} \right) \leq \mathbb{E}_{p^*} \left[\log^2 \left(\frac{p^*(X)}{p(X)} \right) ; p^*(X) > p(X) \right] + C_{KL}(p^* || p)^{1/2}.$$

Proof. $x \mapsto (\log x)^2$ is differentiable with derivative $2x^{-1} \log x$. The derivative is bounded above

on $[1, \infty)$, say by C. Thus, for all $x \geq 1$, $(\log x)^2 \leq (\log 1)^2 + C(x-1) = C(x-1)$. Now,

$$\begin{aligned}\mathbb{E}_{p^*}\left[\log^2\left(\frac{p(X)}{p^*(X)}\right); p^*(X) \leq p(X)\right] \leq & C\mathbb{E}_{p^*}\left[\left(\frac{p(X)}{p^*(X)} - 1\right); p^*(X) \leq p(X)\right] \\ = & C\left(p(p(X) > p^*(X)) - p^*(p(X) > p^*(X))\right) \\ \leq & C\|p^* - p\|_{\text{TV}} \\ \lesssim & \text{KL}(p^*||p)^{1/2}.\end{aligned} \tag{B.38}$$

□

Proposition B.6.13. *If* $\mathbb{E}\exp(t|X|) < \infty$ *for some* $t > 0$ *then* $\xi(\nu_m, \nu_m, L_m) \lesssim m^{-\frac{c_2}{\log|\tilde{\mathcal{B}}|}t}$.

Proof. To approximate p^* with a distribution in $\mathcal{S}(\nu_m, \nu_m, L_m)$ we will use the construction in lemma B.2.3, however we must make sure that the transition probabilities are not less than ν_m. To do so, for sequences X without \$, with $|X| < L_m$, define $(v_{X,b})_{b\in\tilde{\mathcal{B}}}$ to be the output of the application of algorithm 3 or 4 to $(p^*((X,b)\ldots|X\ldots))_{b\in\tilde{\mathcal{B}}}$ if $p^*(X\ldots) > 0$. For X with $p^*(X\ldots) = 0$, make any choice of $(v_{X,b})_b$ with $v_{X,b} \geq \nu_m$ for all b. Thus, for all X, b, $v_{X,b} \geq \nu_m$. Now, by lemma B.6.11, there is a distribution p^{*L_m} with the same infinite-lag transition probabilities as p^* for $|X| \geq L_m$ and infinite-lag transition probabilities $(v_{X,b})_{b\in\tilde{\mathcal{B}}}$ for $|X| < L_m$. Finally perform the construction in lemma B.2.3 to p^{*L_m} to produce a $p^{*L_m}_{L_m} \in \mathcal{S}(\nu_m, \nu_m, L_m)$.

By lemma B.6.12

$$\mathbb{E}\log^2\left(\frac{p^*(X)}{p^{*L_m}_{L_m}(X)}\right) \lesssim \mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{*L_m}_{L_m}(X)}\right); p^*(X) > p^{*L_m}_{L_m}(X)\right] + \left[\mathbb{E}\log\left(\frac{p^*(X)}{p^{*L_m}_{L_m}(X)}\right)\right]^{1/2}.$$

To achieve our result, we will show the first of these terms is $\lesssim m^{-\frac{c_2}{\log|\tilde{\mathcal{B}}|}t}$ and one may use a similar proof to make the same deduction about the second term.

First we will split the term into two that represent the "distance" from p^* to p^{*L_m} and that from p^{*L_m} to $p^{*L_m}_{L_m}$:

$$
\begin{aligned}
&\mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{L_m}_{L_m}(X)}\right); p^{L_m}_{L_m}(X) < p^*(X)\right] \\
=&\mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{L_m}(X)}\right); |X| \leq L_m, p^{L_m}_{L_m}(X) < p^*(X)\right] \\
&+ \mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{L_m}((X_1,\ldots,X_{L_m})\ldots)|\tilde{\mathcal{B}}|^{-(|X|-L_m)}}\right); |X| > L_m, p^{L_m}_{L_m}(X) < p^*(X)\right] \\
\leq&\mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{L_m}(X)}\right); |X| \leq L_m, p^{L_m}_{L_m}(X) < p^*(X)\right] \\
&+ \mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{L_m}(X)|\tilde{\mathcal{B}}|^{-(|X|-L_m)}}\right); |X| > L_m, p^{L_m}_{L_m}(X) < p^*(X)\right] \\
\leq&\mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{L_m}(X)}\right); |X| \leq L_m, p^{L_m}_{L_m}(X) < p^*(X)\right] \\
&+ 4\mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{L_m}(X)}\right); |X| > L_m, p^{L_m}_{L_m}(X) < p^*(X)\right] \\
&+ 4\log^2\left(|\tilde{\mathcal{B}}|\right)\mathbb{E}\left[(|X|-L_m); |X| > L_m\right] \\
\lesssim&\mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{L_m}(X)}\right)\right] + \mathbb{E}\left[(|X|-L_m)^2; |X| > L_m\right].
\end{aligned}
\tag{B.39}
$$

Now we will show each of these two terms $\lesssim m^{-\frac{c_2}{\log|\tilde{\mathcal{B}}|}t}$ in turn.

We will first consider $\mathbb{E}\left[(|X|-L_m)^2; |X|>L_m\right]$.

$$\begin{aligned} p^*((|X|-L_m)^2>l) =& p^*(e^{t|X|}>e^{t(\sqrt{l}+L_m)}) \\ \leq& e^{-tL_m}\mathbb{E}\left[e^{t|X|}\right]e^{-t\sqrt{l}} \end{aligned} \tag{B.40}$$

by Markov's inequality, so

$$\begin{aligned} \mathbb{E}\left[(|X|-L_m)^2; |X|>L_m\right] &= \int_{L_m}^{\infty} p^*((|X|-L_m)^2>l)dl \\ &\leq e^{-tL_m}\mathbb{E}\left[e^{t|X|}\right]\int_{L_m}^{\infty} e^{-t\sqrt{l}}dl \\ &\leq e^{-tL_m}\mathbb{E}\left[e^{t|X|}\right]2t^{-2}(t\sqrt{L_m}+1)e^{-t\sqrt{L_m}} \\ &= \exp\Bigg(-tL_m - t\sqrt{L_m} - 2\log t \\ &\quad + \log\left(t\sqrt{L_m}+1\right) + \text{const.}\Bigg) \\ &\lesssim \exp\left(-tL_m\right) \\ &\sim m^{-\frac{c_2}{\log|\vec{\mathcal{B}}|}t} \end{aligned} \tag{B.41}$$

as desired.

For the other term in equation B.39, again by lemma B.6.12,

$$\mathbb{E}\log^2\left(\frac{p^*(X)}{p^{*L_m}(X)}\right) \lesssim \mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{*L_m}(X)}\right); p^*(X)>p^{*L_m}(X)\right] + \left[\mathbb{E}\log\left(\frac{p^*(X)}{p^{*L_m}(X)}\right)\right]^{1/2}.$$

In this case, we will show that the first of these terms is $\lesssim e^{-Cm^{c_1}}$ for some positive constant C, and by a similar proof one may show the same for the second. This will complete the proof of part 2.

If $p^*(X) > p^{L_m}(X) \geq 0$, by the definition of p^{*L_m},

$$\frac{p^*(X)}{p^{*L_m}(X)} = \prod_{i=1}^{L_m \vee |X|} \frac{p^*(X_{1:i} \ldots | X_{1:i-1} \ldots)}{v_{X_{1:i-1}, X_i}} \leq \left(1 - (|\tilde{\mathcal{B}}| - 1)\nu_m\right)^{L_m}$$

with the inequality by property (5) in proposition B.6.3. Thus,

$$\mathbb{E}\left[\log^2\left(\frac{p^*(X)}{p^{*L_m}(X)}\right); p^*(X) > p^{*L_m}(X)\right] \lesssim L_m^2 \nu_m^2 \lesssim \log^2(m) e^{-2Cm^{c_1}} \lesssim e^{-C'm^{c_1}}$$

for two positive constants C, C'. □

Consistency and rate

The proof of theorem B.6.16 relies on a consequence of theorem 2.1 of Ghosal et al. [87], which is stated in a simplified form herein as theorem B.6.14. Intuitively, the key challenge in establishing nonparametric consistency is that the size of the space of probability measures $\mathscr{P}$ (infinite dimensional) may overwhelm the evidence provided by the data, leading to a posterior that is too spread out. To establish consistency, theorem 2.1 of Ghosal et al. [87] requires that the prior over probability measures is sufficiently large on a neighborhood of p^* (denoted $\mathfrak{B}_\eta$), and sufficiently small on the complement of an effectively parametric (finite dimensional) subset of $\mathscr{P}$ (denoted $\mathscr{P}_N$).

Theorem B.6.14. *Say $\mathscr{P}$ is a set of probability measures, $p^* \in \mathscr{P}$. $X_1, \ldots, X_N \sim p^*$ iid, d is the Hellinger distance, Π is a distribution on $\mathscr{P}$, $(\eta_N)_{N=1}^\infty$ is a sequence of positive numbers such that*

$\eta_N \to 0$ and $N\eta_N^2 \to \infty$, and $(\mathscr{P}_N)_{N=1}^{\infty}$ are a sequence of subsets of $\mathscr{P}$. Define, for positive η,

$$\mathfrak{B}_\eta = \{p \in \mathscr{P} \mid \mathrm{KL}(p^*||p) < \eta^2, \mathrm{Var}[\log(p^*(X)/p(X))] < \eta^2\}.$$

Then if

i) $\log \mathcal{N}(\eta_N/2, \mathscr{P}_N, d) \lesssim N\eta_N^2$

ii) $\log \Pi(\mathfrak{B}_{\eta_N}) \gtrsim -N\eta_N^2$

iii) For an $\epsilon > 0$, $\Pi(\mathscr{P} \setminus \mathscr{P}_N)\Pi(\mathfrak{B}_{\eta_N})^{-1} e^{(1+\epsilon)N\eta_N^2} \to 0$

Then for large enough M,

$$\Pi(B(p^*, M\eta_n)|X_1, \ldots, X_N) \to 1$$

in probability, where $B(p^*, \delta)$ is a Hellinger ball of radius δ centered at p^*

Proof. For some C,

$$CN\eta_N^2 \geq \log \mathcal{N}(\eta_N/2, \mathcal{S}_N, d) \geq \log \mathcal{D}(\eta_N, \mathcal{S}_N, d).$$

Defining $\eta'_N = \sqrt{C}\eta_N$, condition 2.2 in theorem 2.1 of Ghosal et al. [87] is satisfied for the sequence $(\eta'_N)_{N=1}^{\infty}$. Note condition 2.4 is also satisfied by the above condition ii.

Note by lemma 8.1 in Ghosal et al. [87]

$$\begin{aligned} D_N =& \int \prod_{n=1}^{N} \frac{p(X_n)}{p^*(X_n)} d\Pi(p) \\ \geq& \Pi(\mathfrak{B}_{\eta'_N}) \left(\frac{1}{\Pi(\mathfrak{B}_{\eta'_N})} \int_{\mathfrak{B}_{\eta'_N}} \prod_{n=1}^{N} \frac{p(X_n)}{p^*(X_n)} d\Pi(p) \right) \\ \geq& \Pi(\mathfrak{B}_{\eta'_N}) e^{-(1+\epsilon)N\eta'^2_N} \end{aligned} \tag{B.42}$$

with probability $1 - \left(\epsilon^2 N \eta'^2_N\right)^{-1} \to 1$. Call the set where this occurs A. As in the proof of theorem 2.1 of Ghosal et al. [87], for large enough M, C', we may then use condition i to write

$$\begin{aligned} 1 - \mathbb{E}_{p^*} \left[\Pi(B(p^*, M\eta'_N) | X_1, \ldots, X_N) \right] \leq& 2e^{-C'N\epsilon'^2_N} + (1 - p^*(A)) \\ &+ \mathbb{E}_{p^*} \left[D_N^{-1} \left(\Pi(\mathscr{P} \setminus \mathscr{P}_N) + e^{-C'NM^2\epsilon'^2_N} \right); A \right]. \end{aligned} \tag{B.43}$$

By conditions ii and iii, this last term $\to 0$ for large enough M. Finally, write $M\eta'_N = \left(M\sqrt{C}\right)\eta_N$ to get the result in terms of η_N.

□

To work with sieves without restrictions on transition probabilities to $b \in \mathcal{B}$ we need the following technical lemma.

Lemma B.6.15. *Assume for positive numbers* $(\alpha_{k,b})_{L>0, k\in\mathcal{B}^o_L, b\in\tilde{\mathcal{B}}}$, $\sup_{L>0,k\in\mathcal{B}^o_L,b\in\tilde{\mathcal{B}}} \alpha_{k,b} < \infty$ *and* $\inf_{L>0,k\in\mathcal{B}^o_L,b\in\tilde{\mathcal{B}}} \alpha_{k,b} > 0$. *Consider independent* $\text{Dirichlet}(\alpha_{k,b})_{b\in\tilde{\mathcal{B}}}$ *priors on each simplex of* $\Delta_{\tilde{\mathcal{B}}}^{\mathcal{B}^o_L}$ *indexed by* $k \in \mathcal{B}^o_L$. *Call the joint distribution* Π. *Then, for some* $C, \epsilon > 0$, *for all* $\nu > \nu'$ *small,*

L,

$$\log \frac{\Pi(\mathcal{S}(\nu', \nu, L))}{\Pi(\mathcal{S}(0, \nu, L))} \geq -C|\mathcal{B}|^L \nu'^{\epsilon}$$

.

Proof. Define $\alpha^\wedge \leq \inf_{L,k,b} \alpha_{k,b}$. Let $Z_k \sim \text{Dirichlet}(\alpha_{k,b})_b$ for some k. As a property of the Dirichlet distribution,

$$\left(Z_{k,\$}, \frac{\sum_{b' \in \mathcal{B}} Z_{k,b'}}{\sum_{b' \in \tilde{\mathcal{B}}} Z_{k,b'}}\right) \perp\!\!\!\perp \left(\frac{Z_{k,b'}}{\sum_{b' \in \mathcal{B}} Z_{k,b'}}\right)_{b' \in \mathcal{B}}$$

Call this later variable Y_k, and note $Y_k \sim \text{Dirichlet}(\alpha_{k,b})_{b \in \mathcal{B}}$. Now for any $b \in \mathcal{B}, v < \nu$, since $(Y_{k,b}, \sum_{b' \neq b} Y_{b'}) \sim \text{Beta}(\alpha_{k,b}, \sum_{b' \neq b} \alpha_{k,b'})$,

$$\begin{aligned} P(Y_{k,b} < \nu'/(1-v)) =& \frac{\Gamma(\sum_{b' \in \tilde{\mathcal{B}}} \alpha_{k,b'})}{\Gamma(\alpha_{k,b})\Gamma(\sum_{b' \neq b} \alpha_{k,b'})} \int_0^{\nu'/(1-v)} x_b^{\alpha_{k,b}-1}(1-x_b)^{(\sum_{b' \neq b} \alpha_{k,b'})-1} \\ =& O(1) \int_0^{\nu'/(1-v)} x_b^{\alpha_{k,b}-1} \\ =& O((\nu'/(1-v))^{\alpha_{k,b'}}). \end{aligned} \tag{B.44}$$

Thus, using a union bound, for some C, regardless of the choice of k,

$$P(Y_{k,b} < \nu'/(1-v) \text{ for some } b \in \mathcal{B}) \leq C(\nu'/(1-v))^{\alpha^\wedge}.$$

Thus, for some $C' > 0$, calling $F_{k,\$}$ the density of $Z_{k,b}$, noting $P(Z_{k,\$} > \nu) = O(1)$ for small ν,

$$\begin{aligned} P(Z_{k,b} < \nu' \text{ for some } b \in \mathcal{B} \mid Z_{k,\$} > \nu) \lesssim & \int_{\nu}^{1} P(Y_{k,b} < \nu'/(1-v) \text{ for some } b \in \mathcal{B}) dF_{k,\$}(v) \\ \lesssim & \nu'^{\alpha^\wedge} \int_{\nu}^{1} dv v^{\alpha_{k,\$}-1}(1-v)^{\sum_{b\in\mathcal{B}} \alpha_{k,b}-1-\alpha^\wedge}. \end{aligned} \tag{B.45}$$

The integral is equal to the probability of a $(Beta)(\alpha_{k,\$}, \sum_{b\in\mathcal{B}} \alpha_{k,b} - \alpha^\wedge)$ distribution being greater than ν and is thus $O(1)$. For small enough ν, ν', for some $C' > 0$,

$$\begin{aligned} \log \frac{\Pi(\mathcal{S}(\nu', \nu, L))}{\Pi(\mathcal{S}(0, \nu, L))} &= \prod_{k \in \mathcal{B}_L^o} \log P(Z_{k,b} \geq \nu' \text{ for all } b \in \mathcal{B} \mid Z_{k,\$} > \nu) \\ &\geq \log\left((1 - C\nu'^{\alpha^\wedge})^{|\mathcal{B}_L^o|}\right) \\ &\geq -C'|\mathcal{B}_L^o|\nu'^{\alpha^\wedge} \\ &\geq -C''|\mathcal{B}|^L \nu'^{\alpha^\wedge}. \end{aligned} \tag{B.46}$$

□

We can now prove the main result, establishing posterior consistency and the posterior convergence rate. We show that the prior in condition B.6.9 satisfies the conditions of B.6.14. In particular, we use sieves $\mathcal{S}$ to define the effectively parametric subset $\mathscr{P}_N$ of the infinite dimensional space of probability measures $\mathscr{P}$, and then condition B.6.9 controls the prior probability over the $\mathfrak{B}_{\eta_N}$ and $\mathscr{P}_N$.

Theorem B.6.16. *Assume p^* is sub-exponential and thus we can choose a prior as in condition B.6.9.*

For any large enough M,

$$\Pi(B(p^*, MN^{-\frac{1}{2}(1-(c_1+c_2))})|X_1, \ldots, X_N) \to 1$$

in probability where $B(p^*, \delta)$ is a Hellinger ball of radius δ centered at p^*.

Proof. The proof will proceed by checking the conditions of theorem B.6.14. First define a monotonic sequence $(\nu'_m)_{m=1}^{\infty}$ with $\log \nu_N'^{-1} \sim N^{\omega}$, $\xi_N = \xi(\nu_N, \nu_N, L_N)$, $\mathscr{P}$ the set of distributions on S, and

$$\mathscr{P}_N = \{p_v \mid v \in \cup_{n=1}^{N} \mathcal{S}(\nu'_n, \nu_n, L_n)\} = \{p_v \mid v \in S(\nu'_N, \nu_N, L_N)\}.$$

Throughout we will use $\eta_N = N^{-\frac{1}{2}(1-(c_1+c_2))}$ and so checking the conditions of theorem B.6.14 will demonstrate a posterior concentration rate of $\frac{1}{2}(1-(c_1+c_2))$.

First we will check condition i. Define, for $\zeta \in \mathbb{N}^{\mathcal{B}^o_{L_N} \times \tilde{\mathcal{B}}}$, $\rho_N > 0$,

$$\hat{\mathcal{O}}_N(\zeta) = \{v \in \mathcal{S}(\nu'_N, \nu_N, L_N) | \, \forall (k,b), \; (1+\rho_N)^{\zeta_{k,b}} \nu_N^b > v_{k,b} \geq (1+\rho_N)^{\zeta_{k,b}-1} \nu_N^b\}$$

(where $\nu_N^b = \nu'_N$ if $b \neq \$$ and equal to ν_N otherwise) so that $\cup_\zeta \hat{\mathcal{O}}_N(\zeta) = \mathcal{S}(\nu'_N, \nu_N, L_N)$ (Fig. B.3).

Note that for $v_1, v_2 \in \hat{\mathcal{O}}_N(\zeta)$, $\textsc{kl}(p_{v_1}||p_{v_2}) \leq \log(1+\rho_N)\mathbb{E}_{v_1}|X| \leq \rho_N \nu_N^{-1}$ the last inequality as $p(|X| > L||X| \geq L) \geq \nu_N$ where the last inequality comes from $p(|X| = L||X| \geq L) \geq \nu_N$ and a geometric sum (this is where a distinction between ν_N and ν'_N is necessary). Defining d as the

Hellinger metric,

$$d(p_{v_1}, p_{v_2}) \leq \frac{1}{\sqrt{2}} \|p_{v_1} - p_{v_2}\|_1^{1/2} \leq \textsc{kl}(p_{v_1} || p_{v_2})^{1/4} \leq (\rho_N \nu_N^{-1})^{1/4}$$

so picking $\rho_N = \nu_N(\eta_N/2)^4$, for $v_1, v_2 \in \hat{\mathcal{O}}_N(\zeta)$, $d(p_{v_1}, p_{v_2}) \leq \eta_N/2$. Call $\gamma^b = \left(\frac{\log\left(\left(\nu_N^b\right)^{-1}\right)}{\log(1+\rho_N)} + 1 \right)$ and note $(1+\rho_N)^{\gamma^b - 1} \nu_N^b = 1$. Thus the number of choices of $\zeta \in \mathbb{N}^{\mathcal{B}^o_{L_N} \times \tilde{\mathcal{B}}}$ that give non-empty $\hat{\mathcal{O}}_N(\zeta)$, is bounded above by $\prod_{b \in \tilde{\mathcal{B}}} \left(\gamma^b\right)^{|\mathcal{B}^o_{L_N}|}$. Note also that since $\rho_N \to 0$, $\gamma^b \lesssim \frac{\log\left(\left(\nu_N^b\right)^{-1}\right)}{\rho_N}$.

Now we can establish condition i of theorem B.6.14:

$$\begin{aligned}
\log \mathcal{N}(\eta_N/2, \mathcal{S}_N, d) \leq & \log \#\{\zeta \mid \hat{\mathcal{O}}_N(\zeta) \neq \emptyset\} \\
\leq & |\mathcal{B}^o_{L_N}| \sum_b \log\left(\gamma^b\right) \\
\lesssim & |\mathcal{B}|^{L_N} \sum_b \left(\log\log\left(\left(\nu_N^b\right)^{-1}\right) - \log(\nu_N(\eta_N/2)^4) \right) \\
\lesssim & |\mathcal{B}|^{L_N} \left(\log(\nu_N^{-1}) + \log(N) \right) \\
\lesssim & N^{c_1 + c_2} \\
\lesssim & N \eta_N^2.
\end{aligned} \tag{B.47}$$

Now we will demonstrate condition ii. Define, as in theorem B.6.14,

$$\begin{aligned}
\mathfrak{B}_\eta = & \{p \in \mathcal{M} \mid \textsc{kl}(p^* || p) < \eta^2, \operatorname{Var}[\log(p^*(X)/p(X))] < \eta^2\} \\
\supseteq & \{p \in \mathcal{M} \mid \mathbb{E} \log^2(p^*(X)/p(X)) < \eta^4 \wedge 1\}
\end{aligned} \tag{B.48}$$

since $\mathrm{Var}[\log(p^*(X)/p(X))] \vee \mathrm{KL}(p^*||p)^2 \leq \mathbb{E}\log^2(p^*(X)/p(X))$.

Fix N. First we will delineate a volume in $\mathcal{S}(\nu_m, \nu_m, L_m)$ for any $m > 0$ that is within $\mathfrak{B}_{\eta_N}$. Using the definition of ξ, we can label a $v_m^* \in \mathcal{S}(\nu_m, \nu_m, L_m)$ such that $\mathbb{E}[\log(p^*(X)/p_{v_m^*}(X))^2] \leq 2\xi_m$. Note that if there exists a $v \in \mathcal{S}(\nu_m, \nu_m, L_m)$ such that for some $\rho_m > 0$ and all k, b, $(1+\rho_m) \geq \frac{v_{k,b}}{v^*_{m,k,b}} \geq (1+\rho_m)^{-1}$ then

$$\begin{aligned}\mathbb{E}[\log(p^*(X)/p_v(X))^2] \leq& 8\xi_m + 4\mathbb{E}\log^2(p_{v_m^*}(X)/p_v(X)) \\ \leq& 8\xi_m + 4\log^2(1+\rho_m)\mathbb{E}|X|^2.\end{aligned} \tag{B.49}$$

Now pick, for large enough m,

$$\rho_m = \sqrt{\frac{\eta_N^4 - 8\xi_m}{4\mathbb{E}|X|^2}} \leq \exp\left(\sqrt{\frac{\eta_N^4 - 8\xi_m}{4\mathbb{E}|X|^2}}\right) - 1$$

so that if $(1+\rho_m) \geq \frac{v_{k,b}}{v^*_{m,k,b}} \geq (1+\rho_m)^{-1}$ for all k, b, then $p_v \in \mathfrak{B}_{\eta_m}$.

Fixing k, the probability under a Dirichlet$(\alpha_{k,b})_b$ distribution of $W_{m,k} = \{v_k \mid (1+\rho_m) \geq \frac{v_{k,b}}{v^*_{m,k,b}} \geq (1+\rho_m)^{-1} \, \forall \, b\}$ (depicted in Fig. B.6(A)) is, considering the case where v_m^* is on one of the corners of the simplex $\{v_k \mid v_{k,b} \geq \nu_m\}$, at least

$$V_{m,k} = \left(C_1 \nu_m^{(\sum_b (\alpha_b \wedge 1 - 1))}\right)\left(C_2(\nu_m\rho_m)^{|\tilde{\mathcal{B}}|-1}\right)$$

where the first term is a lower bound on the density and the second on the volume of $W_{m,k}$ and C_1, C_2 are constants depending on $|\tilde{\mathcal{B}}|$. $C_1 > 0$ as $\inf_{k,b} \alpha_{k,b} > 0$. As well, one may check that

the volume is minimized should $v^*_{m,k,b} = \nu_m$ for all but one b; in this case, the volume forms a particular diamond-like shape with side-lengths scaled as $\nu_m \rho_m$ and dimensionality $|\tilde{\mathcal{B}}| - 1$ (Fig. B.6(B)), (if $v^*_{m,k,b} = 1 - (|\tilde{\mathcal{B}}| - 1)\nu_m$, then the condition $v_{k,b} \geq (1+\rho_m)^{-1} v^*_{m,k,b} \gtrsim \rho_m$ does not affect the $W_{m,k}$ for large m as $\nu_m \to 0$) (Fig. B.6).

Now we will lower bound the probability of $\mathfrak{B}_{\eta_N}$ by the probability of the above defined volume for a particular m, m_N. Call $\delta = 1 - \frac{1-(c_1+c_2)}{c_3/2} > 0$ and define

$$m_N = \left\lceil \left(\frac{\eta_N^4}{16C} \right)^{-1/c_3} \right\rceil \lesssim N^{1-\delta}$$

so that $8\xi_{m_N} \leq \frac{1}{2}\eta_N^4$ for all $m \geq m_N$, and $m_N \to \infty$. Now,

$$\begin{aligned}
\log(\Pi(\mathfrak{B}_{\eta_N})) \geq & \log \left(\pi(m_N) \prod_{k \in \mathcal{B}^o_{L_{m_N}}} V_{m_N,k} \right) \\
\gtrsim & \log(\pi(m_N)) + \left(|\mathcal{B}^o_{L_{m_N}}| - \sum_{k,b} \alpha_{k,b} \wedge 1 \right) \log(\nu_{m_N}^{-1}) \\
& - |\mathcal{B}^o_{L_{m_N}}| (|\tilde{\mathcal{B}}| - 1) \log(\rho_{m_N}^{-1}) \\
\gtrsim & \log(\pi(m_N)) - |\mathcal{B}|^{L_{m_N}} \log(\nu_{m_N}^{-1}) - |\mathcal{B}|^{L_{m_N}} \log(\rho_{m_N}^{-1}).
\end{aligned} \tag{B.50}$$

For the first term, due to condition B.6.9, $(c_1 + c_2) > (1-\delta)\omega > (1-\delta)(c_1+c_2)$, so,

$$\log \pi(m_N) \sim -m_N^{\omega} \gtrsim -N^{(1-\delta)\omega} \gtrsim -N^{c_1+c_2}.$$

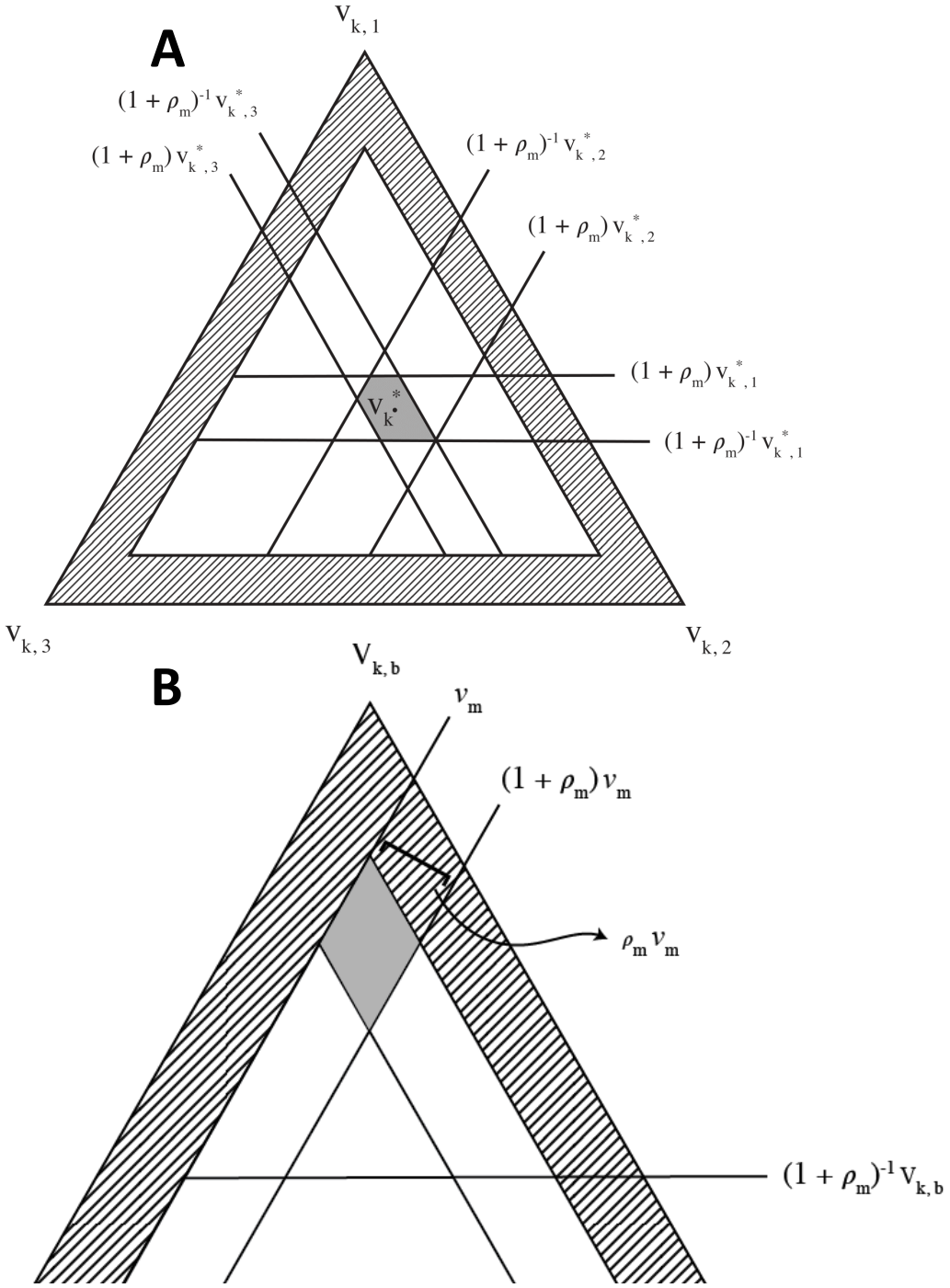


Figure B.6: (A) Example of a set $W_{m,k}$ (solid gray) where $(1+\rho_m) \geq \frac{v_{k,b}}{v^*_{m,k,b}} \geq (1+\rho_m)^{-1} \, \forall \, b$ on $\Delta_{\tilde{\mathcal{B}}}$ for a particular k and m when $|\mathcal{B}| = 3$. (B) Depiction of minimum volume possible. The dashed region represents those transition probabilities that have components less than ν_m.

The second term has

$$|\mathcal{B}|^{L_{m_N}} \log(\nu_{m_N}^{-1}) \lesssim {m_N}^{(c_1+c_2)} \lesssim N^{(1-\delta)(c_1+c_2)}.$$

Finally, for the third, note that since $8\xi_{m_N} \leq \frac{1}{2}\eta_N^4$,

$$\log(\rho_{m_N}^{-1}) \lesssim -\log(\eta_N^4 - 8\xi_{m_N}) \lesssim -\log(\eta_N^4) \lesssim -\log(N).$$

Thus,

$$\log(\Pi(\mathfrak{B}_{\eta_N})) \gtrsim -N^{(1-\delta)(1+\omega)c_2} \gtrsim -N^{(c_1+c_2)} = -N\eta_N^2.$$

Finally, for condition iii, note

$$\Pi(\mathscr{P} \setminus \mathscr{P}_N) = \pi(m > N) + \sum_{m=1}^{N} \pi(m)(1 - \Pi(\mathcal{S}(\nu'_N, \nu_m, L_m) \mid \mathcal{S}(0, \nu_m, L_m))).$$

From lemma B.6.15, we have, for $C, C', \epsilon > 0$, the second term is dominated by

$$\begin{aligned} \sum_{m=1}^{N} \pi(m) \log \frac{\Pi(\mathcal{S}(0, \nu_m, L_m))}{\Pi(\mathcal{S}(\nu'_N, \nu_m, L_m))} &\lesssim \sum_{m=1}^{N} |\mathcal{B}|^{L_m} \nu_N'^{\epsilon} \\ &\lesssim \nu_N'^{\epsilon} L_N |\mathcal{B}|^{L_N} \\ &\lesssim \exp(-2\epsilon C N^{\omega}) \end{aligned} \tag{B.51}$$

for some $C > 0$. On the other hand, since one may check that $\pi(m+1)/\pi(m) < 1/2$ for all L,

we have $\pi(m > N) \leq \pi(N)$. Thus,

$$\log \Pi(\mathscr{P} \setminus \mathscr{P}_N) \lesssim -N^{\omega}.$$

Now we may write, for any $\epsilon > 0$, since $\omega > c_1 + c_2$

$$\log(\log \Pi(\mathscr{P} \setminus \mathscr{P}_N) e^{(1+\epsilon)N\eta_N^2} \Pi(\mathfrak{B}_{\eta_N})^{-1}) \lesssim -N^{\omega} + N^{c_1+c_2} + N^{(1-\delta)\omega} \to -\infty.$$

□

Use in practice

Theorem B.6.16 reveals that the choice of prior controls a kind of bias-variance tradeoff in the model's posterior. In particular, from condition B.6.9 we have

$$c_3 > 2(1 - (c_1 + c_2)) \tag{B.52}$$

Decreasing the prior hyperparameters c_1 and c_2 decreases the width of the posterior distribution (which plays the role of variance). However, reducing c_1 and c_2 forces down c_3 (by the definition of ξ), and this reduces the weight that the prior places on larger sieves that can match the data distribution better (i.e. sieves with lower $\xi(\nu, \nu, L)$ values), consequently increasing the model's bias. When c_1 and c_2 become low enough, the bias becomes overwhelming, equation B.52 is violated, and consistency is no longer guaranteed.

In practice it is often sensible heuristically to set $\nu_m = 0$. In the case, for instance, of short-read sequencing data, there's relatively little correlation between the letters of the read and where it terminates. The probability of stopping is thus often similar across different kmers, even when comparing among kmers of different length. As the posterior concentrates at a roughly constant stopping probability, even a low one, ν_m quickly becomes irrelevant as it decays to zero exponentially. When $\nu_m = 0$, the prior simplifies: it can be written as a distribution over lags $\pi(L)$ times independent Dirichlet priors on each $\mathcal{M}_L$ for $L \in \{1, 2, \ldots\}$. The prior over lags takes the form

$$\log \pi(\{m \mid L_m = L\}) \sim -|\mathcal{B}|^{\frac{\omega}{c_2} L}.$$

Since $\omega > c_2$, we may write $\frac{\omega}{c_2}$ as $1 + c$ for a small $c > 0$.

B.7 Toy models

In this section we describe in depth our simulation experiments.

B.7.1 Finite lag models

This subsection describes experiments conducted to study in practice the finite lag consistency results described in Sections B.3 and B.4, and includes details on the results presented in Section 2.2 and Figure 2.2.

Setup

To simulate data, we used an AR model with parameters $\theta = (A, B)$ defined by the function,

$$f_k(A, B) = \text{softmax}\left((1-\beta^*) \sum_{l=1}^{L} \sum_{b' \in \mathcal{B}^o} A_{b,l,b'} k_{l,b'} + \beta^* \sum_{l,l'=1}^{L} \sum_{b',b'' \in \mathcal{B}^o} B_{b,l,l',b',b''} k_{l,b'} k_{l',b''} \right)_{b \in \tilde{\mathcal{B}}} \tag{B.53}$$

where $\mathcal{B}^o = \mathcal{B} \cup \{\emptyset\}$ and $k_{l,b}$ is 1 if $k_l = b$ and 0 otherwise. The AR model thus takes the form of a multi-output logistic regression, with β^* controlling the contribution of the pairwise interaction terms. In each independent simulation, rows of the matrix A were sampled following,

$$(A_{b,l,b'})_{b \neq \$} \sim (5/L)(\text{Categorical}),\ A_{\$,l,b'} = -1.5/L.$$

for each l, b', where (Categorical) denotes a one-hot encoded sample from a Categorical distribution with uniform probabilities. The matrix B was generated similarly,

$$(B_{b,l,l',b',b''})_{b \neq \$} \sim (5/L^2)(\text{Categorical}),\ B_{\$,l,l',b',b''} = -1.5/L^2.$$

for each l, l', b', b''. Simulations were repeated five times for each β^* value. We set $L = 5$.

We then fit AR and BEAR models that lack the pairwise terms. In particular, we optimized A alone, setting $B = 0$, i.e. $\theta = (A, 0)$. For the AR models, we trained θ using maximum likelihood, and for the BEAR models, we trained the h, θ hyperparameters using empirical Bayes. In both cases, we trained without mini-batching, using 1000 steps of the Adam optimizer with a training rate of

0.05 [138].

To approximate the KL divergence and total variation distance between the models and the data, 2,000 independent sequences were sampled from the data-generating distribution p^* and used to calculate averages of $\log\left(p^*(X)/p(X)\right)$ and $\frac{1}{2}\left|1 - p(X)/p^*(X)\right|$ respectively, where p is either the maximum likelihood estimator (for the AR models) or the posterior predictive (for the BEAR models, estimated using the maximum *a posteriori* value). (Note that the total variation distance is equal to half the L^1 distance since the set of sequences is countable.)

The parameter A is not identifiable, so to compare between the value of A inferred by the models and the true data-generating value, we transformed A to a canonical representation. Define $\tilde{A}_{b,l,b'} = A_{b,l,b'} - A_{\$,l,b'}$ and define the canonical representation

$$A^{\text{can}}_{b,l,b'} = \tilde{A}_{b,l,b'} - \frac{1}{|\mathcal{B}^o|}\left(\sum_{b''}\tilde{A}_{b,l,b''} - \frac{1}{L}\sum_{l',b''}\tilde{A}_{b,l',b''}\right).$$

Proposition B.7.1. *Two linear AR matrices A, A' define the same linear AR model of lag L if and only if $A^{can} = A'^{can}$.*

Proof. Define the vector space

$$V = \{v \in \mathbb{R}^{L\times\mathcal{B}^o} \mid \forall i,j, \sum_{b'} v_{i,b'} = \sum_{b'} v_{j,b'}\}.$$

One hot encodings of sequences of length L are contained in V. As well, it can be seen that V is spanned by the vectors $(e_{i,b} - e_{i,b'})_{1\le i\le L, b\neq b'\in\mathcal{B}^o}$ (where $e_{i,b}$ is the indicator of position i, b) and

the vector consisting of ones in each entry. This basis of V is made up of linear combinations of one hot encodings of sequences of length L and thus the span of one hot encodings of sequences of length L is V. The orthogonal complement of V is spanned by $(e_i - e_1)_{1<i\leq L}$ where e_i is 1 at position j, b if $j = i$ and 0 otherwise. The transformation

$$v \mapsto \left(v_{i,b} - \frac{1}{|\mathcal{B}^o|} \left(\sum_{b''} v_{i,b''} - \frac{1}{L} \sum_{i',b''} v_{i',b''} \right) \right)_{1\leq i\leq L, b\neq b'\in\mathcal{B}^o}$$

preserves V and annihilates the orthogonal complement of V and is thus the orthogonal projection onto V, P_V.

Thanks to the softmax in Equation B.53, two linear AR matrices A and A' define the same linear AR model if there is a constant C such that for all sequences k of length L and $b \in \tilde{\mathcal{B}}$,

$$\sum_{l=1}^{L} \sum_{b'\in\mathcal{B}^o} A_{b,l,b'} k_{l,b'} = \sum_{l=1}^{L} \sum_{b'\in\mathcal{B}^o} A'_{b,l,b'} k_{l,b'} + C.$$

This is equivalent to the condition

$$\sum_{l=1}^{L} \sum_{b'\in\mathcal{B}^o} \tilde{A}_{b,l,b'} k_{l,b'} = \sum_{l=1}^{L} \sum_{b'\in\mathcal{B}^o} \tilde{A}'_{b,l,b'} k_{l,b'}$$

for all k, b and thus to the condition

$$P_V \tilde{A}_b = P_V \tilde{A}'_b$$

for all b. □

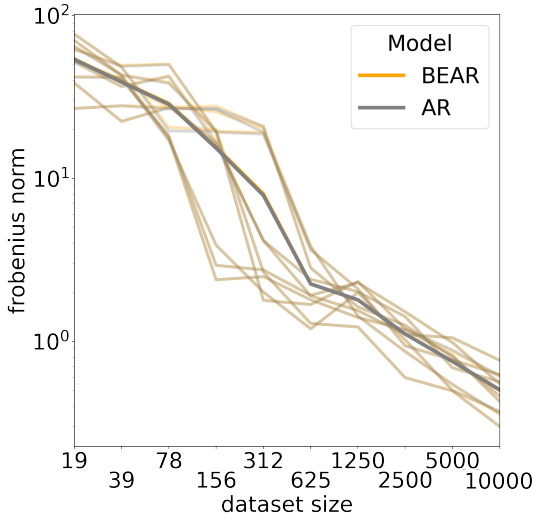


Figure B.7: Frobenius norm between the canonical representation (Section B.7.1) of the AR model parameters θ inferred by fitting an AR model with maximum likelihood and those inferred by fitting the BEAR model with empirical Bayes, in the well-specified ($\beta^* = 0$) case. Thick lines show the average across five independent simulations (small lines). Note that the differences between the two models are indistinguishable relative to the variation across datasets and the variation as dataset size increases.

Results

We first fixed L at the same value as the simulation data, to study the effect of the structured prior in the BEAR model. Figure 2.2A shows the convergence in KL of each model as the dataset size increases, and Figure B.8 the convergence in total variation distance. Figure 2.2B shows the convergence of the hyperparameter h in the BEAR model. In Figure B.7, we compare the parameter A inferred with the AR model to the true data-generating value using the Frobenius norm of the canonical representation of each; likewise for the parameter A inferred with the BEAR model. In this well-specified case, we see that the BEAR model parameter estimate converges just as quickly as the AR model.

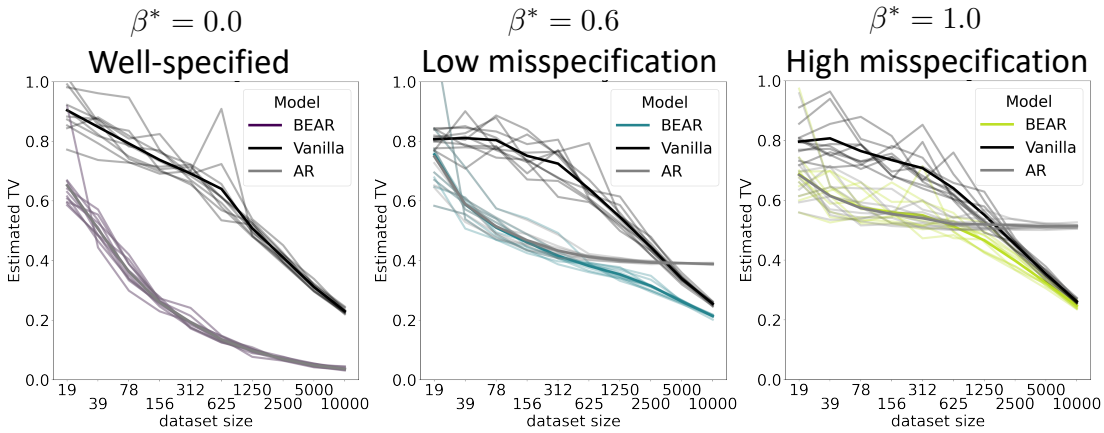


Figure B.8: As in Figure 2.2A, except using the total variation distance in place of the KL norm.

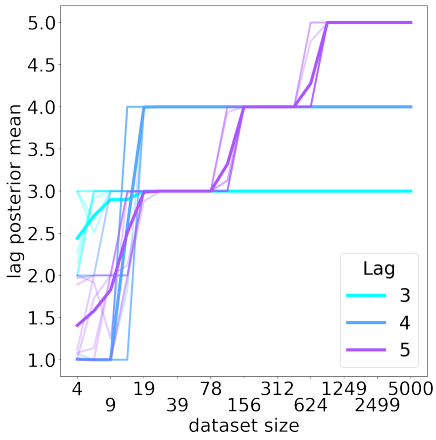


Figure B.9: Mean of the BEAR model posterior over lags, as a function of dataset size. Thick lines show the average across five independent simulations (thin lines).

Next we considered inference of L. We simulated data from models with different L values ($L \in \{3, 4, 5\}$) and $\beta^* = 0$. We computed the expected value of L under the posterior with a uniform prior on lags from 1 to 8. Figure B.9 shows that the inferred lag converges to the true data-generating value.

B.7.2 Infinite lag models

This subsection describes experiments conducted to study the infinite lag (nonparametric) consistency results of Section B.6 in practice.

Setup

To generate from a distribution that was not a finite lag AR model, we chose the first letter in each sequence X uniformly from the alphabet $\mathcal{B}$, then sampled the rest of the sequence following,

$$p(X_i = b | X_1, \ldots, X_{i-1}) \propto \sum_{l=1}^{i-1} l^{-2} \sum_{b' \in \mathcal{B}^o} A_{b,l,b'} X_{i-l,b'}.$$

In each independent simulation, the parameter A was sampled as $A_{b,l,b'} \sim \text{Bernoulli}(0.2)$ for each l, b and $b' \neq \$$, and as $A_{b,l,b'} \sim (0.2)(\text{Bernoulli}(0.2))$ for each l, b and $b' = \$$.

Following Section B.6.3, we set $\nu_m = 0$ and used the prior on lags $\pi(L) \propto \exp(-4^{(1+c)L})$. We used a Jeffreys prior ($\alpha_{k,b} = 1/2$ for all k, b) and took the maximum *a posteriori* value of L and v. We also considered the maximum likelihood estimator of L (i.e. with the prior dropped). To approximate the KL divergence and the total variation distance, we used 30,000 samples; the

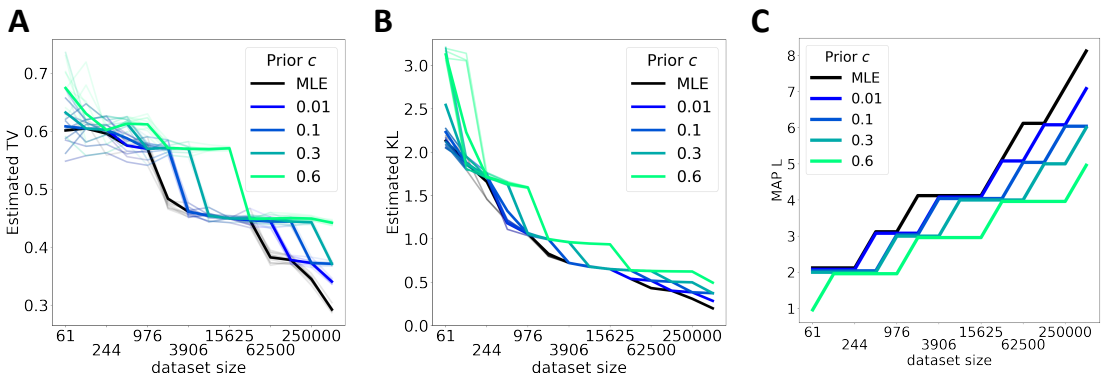


Figure B.10: Convergence in total variation (A) and KL (B) between data-generating distribution and model. Thick lines indicate averages across five individual simulations (thin lines). (C) Maximum *a posteriori* estimator of the lag L in an individual example simulation.

training procedure was otherwise the same as in Section B.7.1.

Results

We examined the convergence of the posterior predictive distribution of the BEAR model for different values of the prior hyperparameter c. In all cases we see convergence to p^* in both total variation and KL (Figure B.10AB). Decreasing c produces a longer-tailed prior, making the maximum *a posteriori* value of L diverge more quickly with dataset size (Figure B.10C). In this example, decreasing c yields faster convergence to p^*. Using the maximum likelihood value of L (equivalent to an improper uniform prior) yields even faster convergence to p^*. As discussed in Section B.6.3, lower c corresponds to larger c_2, and so is expected to yield lower posterior variance but larger bias; in this simulation, the reduction in bias clearly contributes more to accurate density estimation. This may be because the data-generating distribution is close enough to a finite-lag Markov model that the asymptotics of the BEAR model behave similarly to the finite-lag case.

B.7.3 Hypothesis testing

This subsection describes experiments conducted to study the hypothesis testing consistency results of Section B.5 in practice.

Setup

We used the same setup as in Section B.7.1, including the same training and divergence estimation procedures, and sampled datasets from a linear AR model with different values of β^*.

In the goodness-of-fit test, we set $\tilde{p}$ (the model we aimed to test) to a linear AR model with the true, data-generating value of the parameter A but $\beta^* = 0$. We embedded the same linear model, with the same value of A and β^*, in the BEAR model to compute a Bayes factor. Here we set $h = 10^{-3}$, and fixed L at the data-generating value, $L = 5$.

In the two-sample test, instead of comparing to $\tilde{p}$ directly, we compared to samples drawn from $\tilde{p}$. Here we used a Jeffreys prior rather than embed a more complex AR model. We explored both fixing $L = 5$ and using a truncated uniform prior $\pi(L) = 1/8$ for L from 1 to 8 (to evaluate both forms of the consistency results in Section B.5).

Results

We first examined the consistency of the goodness-of-fit test, using the Bayes factor

$$\text{BF} = p((X_n)_{n=1}^N)/\tilde{p}((X_n)_{n=1}^N)$$

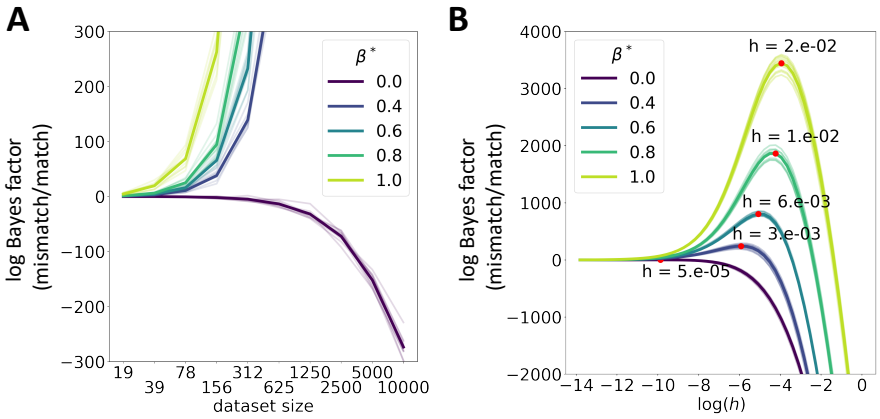


Figure B.11: (A) Log Bayes factor for the BEAR goodness-of-fit test. (B) Log Bayes factor as a function of the hyperparameter h, with peaks identified by red points. In both subfigures, thick lines are averages across five simulations (thin lines).

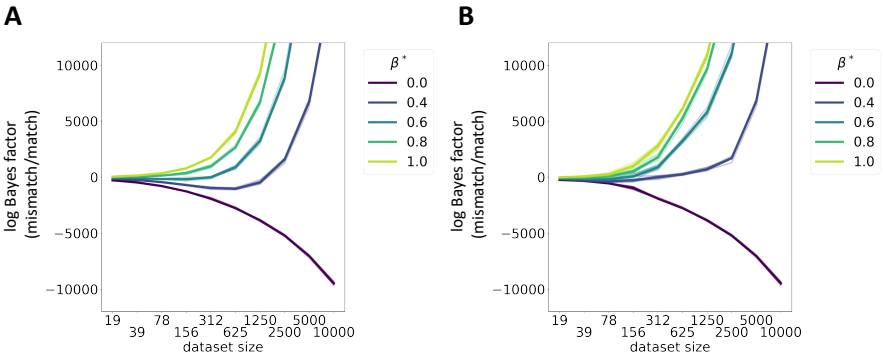


Figure B.12: (A) Log Bayes factor for the BEAR two-sample test, using fixed L. (B) Log Bayes factor for the BEAR two-sample test, marginalizing over a truncated prior on L. In both subfigures, thick lines are averages across five simulations (thin lines). Dataset size is the size of each individual dataset that the two-sample test compares, not their pooled size.

which compares the probability of the data under the BEAR model to the probability under the model of interest $\tilde{p}$. Figure B.11A shows the Bayes factor diverge to $+\infty$ when the data does not match the model ($\beta^* > 0$), but diverge to $-\infty$ when the data does match the model ($\beta^* = 0$). We also explored the Bayes factor as function of h, holding the amount of data fixed at $N = 2500$ (Figure B.11B). In the limit $h \to 0$, the BEAR model reduces to its embedded AR model $\tilde{p}$, and so the Bayes factor converges to 0. On the other hand, in the limit $h \to \infty$, the BEAR model becomes diffuse and the Bayes factor diverges to negative infinity (accepting the null hypothesis). Intermediate values of h in effect "center" the test at the model $\tilde{p}$ we aim to evaluate, increasing its power to detect differences between the data and the model[21].

We next examined the consistency of the two-sample test, using the Bayes factor

$$\text{BF} = p((X_n)_{n=1}^N)p((X'_n)_{n=1}^{N'})/p((X_n)_{n=1}^N, (X'_n)_{n=1}^{N'}),$$

which compares the probability of the two samples being drawn from separate distributions to the probability of their being drawn from the same distribution. Both when using the Bayes factor computed with fixed lag $L = 5$, and when using the Bayes factor computed by marginalizing over a truncated uniform prior on L, we find consistency, with the Bayes factor diverging to $+\infty$ when $\beta^* > 0$ and to $-\infty$ when $\beta^* = 0$ (Figure B.12).

B.8 Scalable inference

In this section we describe how BEAR models were trained at large scale on real data.

B.8.1 Stochastic gradient estimates

Let $\mathcal{S}$ be a set of length L kmers k in $\hat{\mathcal{B}}_L$ chosen uniformly at random (a minibatch). Then, we can form an unbiased stochastic gradient estimate of the marginal likelihood as

$$\nabla_{h,\theta} \log p(X_{1:n}|L,h,\theta) \approx \frac{|\hat{\mathcal{B}}_L|}{|\mathcal{S}|} \sum_{k \in \mathcal{S}} \nabla_{h,\theta} \log \left[\frac{\Gamma(\sum_b \frac{1}{h} f_{kb}(\theta))}{\prod_b \Gamma(\frac{1}{h} f_{kb}(\theta))} \frac{\prod_b \Gamma(\frac{1}{h} f_{kb}(\theta) + \#(k,b))}{\Gamma(\sum_b \frac{1}{h} f_{kb}(\theta) + \#(k,b))} \right].$$

Note also that it is straightforward to parallelize the training algorithm by sending individual minibatches to individual processors at each step, then compiling the results.

B.8.2 Extracting summary statistics

KMC counts kmers in large sequence datasets, outputting a list of kmers k and counts $\#k$ that is typically too large to fit in memory. However, our inference procedure requires full count vectors $\#(k,\cdot)$. We take advantage of the lexicographical ordering of KMC's output to merge kmer counts into count vectors in a (single pass) streaming algorithm. We also take advantage of the lexicographical ordering to construct count vectors $\#(k,\cdot)$ for all lags L given just KMC's output for the largest lag L, thus reducing the number of times KMC needs to be run; this too is done using a single pass streaming algorithm. In order to quickly evaluate models by heldout marginal likelihood, it is convenient to store together the counts $\#(k,\cdot)$ associated with both the training and testing datasets. We accomplish this by merging the KMC output for different datasets as part of the same single pass streaming algorithm. This dataset merging is also useful in training the reference-based models pro-

posed in Section B.10.1, and we merge reference genome counts with sequencing dataset counts in the same way.

B.8.3 Code availability

Code for implementing BEAR models and documentation (including a tutorial for getting started and reproducing basic results) are available at https://github.com/debbiemarkslab/BEAR.

B.9 Datasets

Here we briefly describe each data type and dataset used in evaluating BEAR models, along with some motivation for each. NCBI accession numbers and links for each dataset can be found in the supplementary table, available with the published paper [12]. Dataset sizes are listed in Table B.1. All data is publicly available for research use. Patient data was anonymized by the creators of each dataset, and further details on ethical oversight and patient consent can be found in the cited links and papers.

B.9.1 Whole genome sequencing

Whole genome sequencing is a standard technique for measuring genome sequences. It is often the starting point for running a genome assembly algorithm or variant caller, which aims to infer (non-probabilistically) the underlying genome from the read data. Directly modeling sequencing reads can be interesting, however, since (a) there are typically portions of the genome that are difficult to reliably assemble, such as centromeres and telomeres, (b) there may not be enough data to reli-

Table B.1: Dataset sizes In nucleotides (nt). Dataset abbreviations as in Table 2.1.

Dataset	Total nt	Max. sequence length (nt)
YSD1	151,691,700	150
A. th. 1	3,238,613,507	100
A. th. 2	2,485,960,312	100
A. th. 3	6,831,756,793	100
PBMC	34,935,800,234	91
HL	24,185,778,348	91
GBM	21,506,001,361	65
HC	2,283,930,547	202
CD	1,052,405,190	202
UC	956,179,237	202
Bact.	1,388,421,381	6,358,077

ably detect variants via standard variant callers or assembly, and (c) although the experiment may be directed towards a particular organism's genome other DNA may still be present.

- **YSD1** This is a bacteriophage found in the waterways of the United Kingdom which infects *Salmonella*. It was chosen as an example of a relatively small genome sequencing experiment (phage genomes are short). The sequencing experiment was reported in Dunstan et al. [66].

- **A. th.** Arabidopsis thaliana is a small flowering plant, used as a model organism in plant research. Structural variants are extremely complicated in plants, making traditional variant-calling methods challenging, and kmer-based analysis approaches are of considerable ongoing interest in the literature (see e.g. Voichek & Weigel [277]). The datasets are from the 1001 Genomes Consortium, https://1001genomes.org/ [2].

B.9.2 Single cell RNA sequencing

Single cell RNA sequencing is an increasingly ubiquitous technique for characterizing the transcriptional state of cells. It is used to discover new cell types, track development and disease, as a readout in cellular engineering efforts, and more. Most analysis techniques coarse-grain the data by just counting transcripts or isoforms. Statistical modeling of reads at the nucleotide level may lead to new insight into the joint distribution of sequences and their expression levels, accounting for such phenomena as somatic variation and RNA editing. Single cell RNA sequencing is increasingly used as a method for understanding tumors and their microenvironment; cancer involves both genome mutations as well as transcriptional changes.

- **PBMC** Samples of peripheral blood mononuclear cells are easy to collect from humans, making this a standard type of single cell RNA sequencing dataset. These cells were taken from a healthy donor. The dataset is from 10x Genomics, using its v3 technology.

- **HL** These cells come from a human dissociated lymph node tumor, from a 19-year-old male Hodgkin's lymphoma patient. The dataset is from 10x Genomics, using its v3 technology.

- **GBM** These cells were taken from a patient with glioblastoma, the most common primary brain cancer in adults, and include both tumor and peripheral cells. The dataset was reported in [47] and uses a distinct technology from 10x Genomics methods.

B.9.3 Metagenomics

Metagenomics is an increasingly ubiquitous technique for characterizing microbiomes, including human and environmental microbiomes. Analysis often proceeds by local assembly, annotation of genes or taxa, etc. Statistical modeling of reads at the nucleotide level avoids this coarse graining and can enable detection and analysis of changes in the microbiome outside known genomic elements.

All three of the metagenomics datasets analyzed in the prediction experiments are from [161], a study of inflammatory bowel disease (IBD) as part of the Integrative Human Microbiome Project, and were taken from stool samples. IBD affects more than 3.5 million people worldwide.

- **HC** This dataset was collected from a control patient without IBD.
- **CD** This dataset was collected from a patient with Crohn's disease, a form of IBD involving relapsing and remitting inflammation of the gastrointestinal tract.
- **UC** This dataset was collected from a patient with ulcerative colitis, a form of IBD involving relapsing and remitting inflammation of the colon.

We also examined metagenomics datasets from a study of kidney transplants [229]. Viral transmission from donor to recipient has been associated with complications and increases the risk of allograft failure. Schreiber et al. [229] performed metagenomic sequencing on patient urine samples before and after transplant to assess viral transmission. Further description of this dataset can be found in Section B.13.

B.9.4 Full assembled genomes

Comparisons between distant species are challenging due to complex and large scale genomic changes over evolutionary time. However, generative probabilistic models of protein sequences separated by billions of years of evolution has yielded direct insight into their functional constraints, as well as improved understanding of the large scale evolution of life on earth [110,215]. As a first step towards extending these ideas to whole genomes, we analyzed diverse bacterial genomes from across the tree of life.

- **Bact.** We examined reference bacterial genomes available in RefSeq [190]. Genomes were selected to be taxonomically diverse, representing different genera and families from across the kingdom of Bacteria; the NCBI accessions are listed in supplementary table in the publication.

B.10 Prediction experiments details

Here we provide details on the results reported in the **Predicting sequences** and **Measuring misspecification** subsections of the results (Section 2.6).

B.10.1 Model architectures

- **Linear** The linear model is the same as that used in the toy experiments,

$$f_k(A) = \text{softmax}\left(\sum_{l=1}^{L}\sum_{b'\in\mathcal{B}^o} A_{b,l,b'}k_{l,b'}\right)_{b\in\tilde{\mathcal{B}}}. \tag{B.54}$$

- **CNN** We use a four layer convolutional neural network with the architecture: input $\mapsto$ convolution $\mapsto$ elu $\mapsto$ elu $\mapsto$ softmax $\mapsto$ output, where the convolution is one-dimensional and the elu layers are exponential linear units. Layer normalization was used before each of the elu nonlinearities [14]. Exact details on the model architecture can be found in the supplementary code (Section B.8.3, function `make_ar_func_cnn` in `ar_funcs.py`).

- **Reference-based** Biologists often make use of a reference genome – a canonical example sequence that is intended to be representative of a species – in analyzing genome sequencing data; reference transcriptomes are used similarly in RNA sequencing analysis, etc.. Reads are aligned to the reference in order to infer the portion of the underlying genome or transcriptome that the read originated from. We built on this basic idea to design an AR model that uses a reference sequence to make predictions. In particular, let $\#_{\text{ref}}(k, b)$ denote the number of times the length $L + 1$ kmer (k, b) occurs in the reference sequence(s). One way to form a prediction is by normalizing these counts for each lag, i.e. $f_{k,b} = \#_{\text{ref}}(k, b) / \sum_{b'} \#_{\text{ref}}(k, b')$. We go a step further by (1) accounting for possible mutational or sequencing noise using a Jukes-Cantor mutation model, and (2) accounting for short reads by learning the stop symbol probability. Our complete model is

$$f_{k,b}(\nu, \tau) = (1 - \nu) \left[e^{-\tau} \frac{\#_{\text{ref}}(k, b)}{\sum_{b' \neq \$} \#_{\text{ref}}(k, b')} + (1 - e^{-\tau}) \frac{1}{|\mathcal{B}|} \right] + \nu \mathbb{I}(b = \$) \tag{B.55}$$

where $\tau \in [0, \infty)$ is the (scalar) Jukes-Cantor time parameter, $\nu \in [0, 1]$, and $\mathbb{I}(\cdot)$ is the indicator function that takes value 1 when the expression is true and 0 otherwise.

The reference sequences for each dataset are listed in the supplementary table available with the publication. In analyzing human single cell RNAseq data we pooled multiple reference transcriptomes. We included the reverse complement of each sequence as well as the original sequence when constructing the reference kmer transition counts.

B.10.2 Training

The maximum marginal likelihood lag L was chosen for the vanilla BEAR model (with prior concentration parameter $\alpha_{k,b} = 0.5$ for all k, b). We found in general that the posterior under a uniform prior on lags was strongly peaked at a particular lag (Figure B.15). All other models (both BEAR and AR) were run with this same lag (that is, we did not integrate over all lags in the BEAR model). Using a fixed lag L as a comparison point provides a controlled study of the effects of switching from an AR model of transition probabilities to the BEAR model's AR-structured prior, and choosing L based on the vanilla BEAR model ensures that the comparison to the vanilla BEAR model is conservative.

The kmer count summary statistics were shuffled once before training (in chunks, due to the large size dataset size), and visited in the same order across epochs. Training was initialized only once; preliminary experiments suggested that training was robust to changes in the random seed. Gradient updates were computed in parallel across two GPUs, at double precision. The minibatch size was 250,000. Gradients were accumulated across minibatches to reduce variance (that is, the gradients from multiple minibatches were added together), and optimization was performed using Adam [138]. Models were trained to convergence. Detailed training hyperparameters are displayed in

Table B.2. The CNN models used 30 filters of width 8, except in the case of YSD1 where the filter width was reduced to 5 (for both BEAR and AR models); other neural network architecture hyperparameters are given in the supplementary code (function `make_ar_func_cnn` in `ar_funcs.py`). Experiments were run on an internal cluster (Tesla K80, Tesla M40 and Tesla V100 GPUs).

B.10.3 Evaluation

Accuracy was evaluated based on the maximum likelihood prediction (in the case of AR models) and the maximum *a posteriori* prediction (in the case of BEAR models). Ties in prediction probabilities were resolved uniformly at random.

The perplexity was calculated based on the heldout test dataset as

$$\exp\left[-\frac{\log p((X_n)_{n=1}^{N_{\text{test}}})}{\sum_{n=1}^{N_{\text{test}}}|X_n|}\right] \tag{B.56}$$

where $p((X_n)_{n=1}^{N_{\text{test}}})$ is the probability of the heldout data conditional on the maximum likelihood parameter value (in the case of AR models) or the marginal probability of the heldout data under the posterior predictive distribution (in the case of BEAR models).

B.10.4 Further performance results

The maximum marginal likelihood lag L (under the vanilla BEAR model) for each dataset is reported in B.4. Interestingly, the optimal lags are intermediate between the large kmer lengths (e.g. more than 30) often used for non-probabilistic assembly algorithms (e.g. [238]) and the small kmer

Table B.2: Training parameters Train-test splits and Adam optimization parameters. Dataset abbreviations as in Table 2.1. Accum. steps stands for accumulation steps, the number of steps gradients were accumulated over. Paired end reads were treated as separate and split into train and test sets independently.

Dataset	Train/test split	Epochs	Learning rate	Accum. steps
YSD1	3:1 on reads	500	0.01	10
A. th. 1	3:1 on reads	15	0.02	100
A. th. 2	3:1 on reads	15	0.02	100
A. th. 3	3:1 on reads	3	0.02	100
PBMC	3:1 on reads	3	0.02	100
HL	3:1 on reads	5	0.02	100
GBM	55:23 on cells	4	0.02	100
HC	3:1 on reads	10	0.02	100
CD	3:1 on reads	10	0.02	100
UC	3:1 on reads	10	0.02	100
Bact.	500:166 on genomes	2000	0.01	1

lengths (e.g. less than 10) often used as features in clustering or classification algorithms (e.g. [11]). The marginal likelihood was in general strongly peaked at a particular value (Figure B.15). Increasing the lag generally led to slightly better performance in terms of both perplexity and accuracy for the non-vanilla BEAR models and the AR models, but (unsurprisingly) worse performance for the vanilla BEAR model; the increases in AR model performance were far from enough to make up the difference with BEAR models (Table B.5).

Plots of training loss versus wall clock time for an AR model and the corresponding BEAR model (with the same fixed lag L) are shown in Figure B.13; the loss for each is normalized by the minimum and maximum values to be comparable (the BEAR model substantially outperforms the AR model). The BEAR model converges at least as fast as the AR model.

Table B.3: Predictive accuracy. *Whole genome sequencing data* YSD1: A Salmonella phage. *A. th.*: *Arabidopsis thaliana*, a plant (datasets represent different individuals). *Single cell RNA sequencing data* PBMC: peripheral blood mononuclear cells, taken from a healthy donor. HL: Hodgkin's lymphoma tumor cells. GBM: glioblastoma tumor cells. *Metagenomic sequencing data* HC: healthy (non-CD and non-UC) controls. CD: Crohn's disease. UC: ulcerative colitis. *Full assembled genomes* Bact.: Bacteria. *Models* Van: Vanilla (constant). Lin: Linear. CNN: convolutional neural network. Ref: reference genome/transcriptome model (only applicable to datasets with a reference).

Dataset	AR Lin.	AR CNN	AR Ref.	BEAR Van.	BEAR Lin.	BEAR CNN	BEAR Ref.
YSD1	33.73%	35.86%	90.8%	94.69%	**94.75%**	**94.75%**	94.71%
A. th. 1	35.47%	35.59%	53.81%	86.03%	86.32%	86.34%	**86.50%**
A. th. 2	35.32%	35.61%	70.41%	85.36%	85.71%	**85.77%**	85.66%
A. th. 3	34.94%	35.41%	60.94%	76.46%	78.51%	**78.52%**	77.13%
PBMC	34.36%	34.76%	67.39%	87.83%	**88.16%**	**88.16%**	87.99%
HL	34.67%	35.59%	67.17%	87.68%	**87.96%**	**87.96%**	87.82%
GBM	30.71%	30.9%	61.3%	78.99%	80.44%	80.42%	**81.43%**
HC	32.98%	33.54%	–	83.86%	85.03%	**85.06%**	–
CD	32.13%	32.32%	–	81.72%	83.30%	**83.32%**	–
UC	32.27%	32.23%	–	82.71%	84.26%	**84.27%**	–
Bact.	33.89%	34.78%	-	35.27%	**35.28%**	**35.28%**	-

To evaluate performance as a function of dataset size, we subsampled reads uniformly at random without replacement from the YSD1 dataset, and retrained the models on the smaller datasets (Figure B.14). The original dataset had $\sim 1000\times$ coverage of the bacteriophage genome, meaning that on average 1000 reads were observed overlapping each position in the genome. Note that the vanilla BEAR model performance falls off substantially relative to the BEAR model below $\sim 3\times$ coverage (in the case of the reference model) (Figure B.14BD)

Table B.4: Maximum marginal likelihood lag L. Maximum marginal likelihood lag L for the vanilla BEAR model. Dataset abbreviations as in Table 2.1.

Dataset	L
YSD1	13
A. th. 1	17
A. th. 2	17
A. th. 3	18
PBMC	18
HL	17
GBM	17
HC	16
CD	16
UC	16
Bact.	9

Table B.5: Performance with increasing lag L. The symbol † indicates the maximum marginal likelihood lag L for the vanilla BEAR model. Dataset abbreviations as in Table 2.1.

Perplexity

Dataset	Lag	AR Lin.	AR CNN	AR Ref.	BEAR Van.	BEAR Lin.	BEAR CNN	BEAR Ref.
YSD1	13†	3.953	3.873	1.266	1.165	**1.144**	**1.144**	1.145
YSD1	20	3.937	3.855	1.352	1.177	**1.138**	**1.138**	**1.138**
Bact.	9†	3.831	3.794	-	**3.774**	**3.774**	**3.774**	-
Bact.	12	3.807	3.772	-	3.776	3.741	**3.738**	-

Accuracy

Dataset	Lag	AR Lin.	AR CNN	AR Ref.	BEAR Van.	BEAR Lin.	BEAR CNN	BEAR Ref.
YSD1	13†	33.73%	35.86%	90.8%	94.69%	**94.75%**	**94.75%**	94.71%
YSD1	20	34.19%	36.3%	87.21%	94.88%	94.97%	**94.98%**	94.91%
Bact.	9†	33.89%	34.78%	-	35.27%	**35.28%**	**35.28%**	–
Bact.	12	34.42%	35.13%	–	35.54%	35.86%	**35.93%**	–

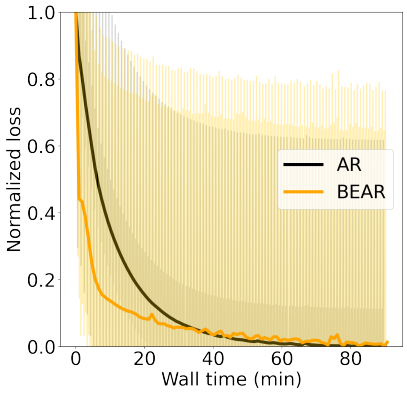


Figure B.13: Relative loss (normalized to be between 0 and 1 based on minimum and maximum values) as a function of wall time for a CNN AR model versus the corresponding BEAR model on the YSD1 dataset ($L = 20$).

B.11 Generation details

Here we provide details on the results reported in the **Generating samples** subsection of the results (Section 2.6).

The CNN BEAR model was trained on the full (combined train/test data) *Arabidopsis thaliana* 1 dataset, with $L = 17$, using identical training parameters as in the performance experiments (Table B.2). 50 bases were generated on the end of reads using the maximum *a posteriori* value of v, and conditional on a stop symbol not occurring, i.e. following the distribution

$$p_{\text{extr}}(X_i = b | k = (X_{i-L}, ..., X_{i-1})) = \frac{f_{k,b}(\theta)/h + \#(k,b)}{\sum_{b' \neq \$} f_{k,b'}(\theta)/h + \#(k,b')} \tag{B.57}$$

for $b \neq \$$ and $p(X_i = \$|k) = 0$, where recall $\#(k,b)$ is the number of times b is seen succeeding k in the data, and θ and h are the learned hyperparameters. The values of $\#(k,b)$ are retrieved from

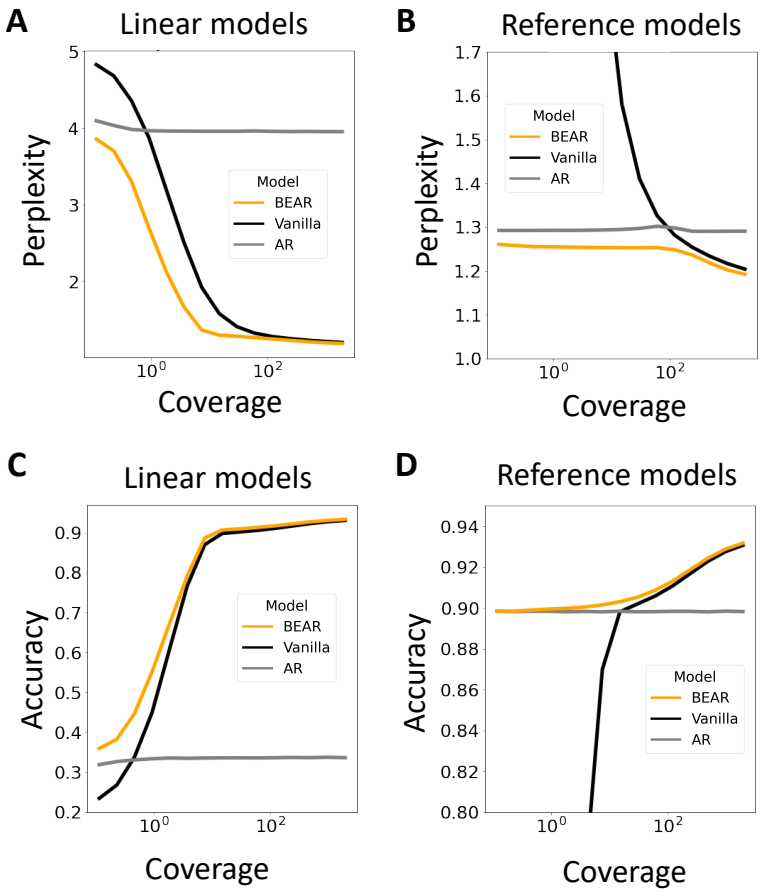


Figure B.14: Perplexity (AB) and accuracy (CD) of AR and BEAR models as a function of total dataset size, measured in terms of coverage (coverage is the expected number of reads from each position in the genome; it is linearly proportional to the total number of reads). Subfigures A and C show results for the the linear AR model (and its BEAR embedding), and B and D for the reference-based AR model (and its BEAR embedding). The lag was held fixed in all cases.

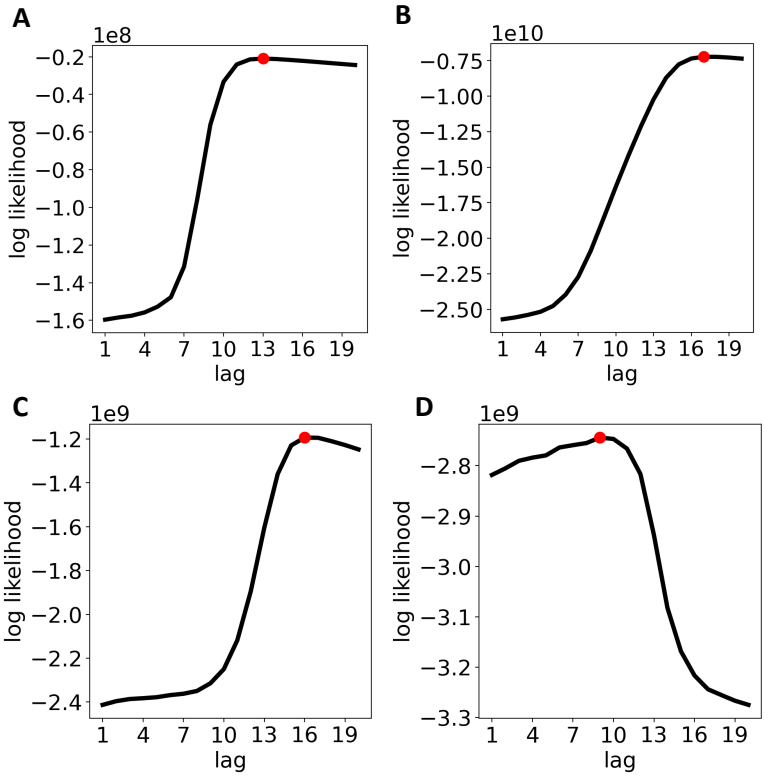


Figure B.15: Marginal log likelihood under the vanilla BEAR model as a function of lag L for the bacteriophage YSD1 (A), glioblastoma GBM (B), control metagenomic HC (C) and bacteria Bact. (D) datasets. Note the large scale (upper left) of each plot.

the dataset efficiently using the Jellyfish kmer indexing package [166]. 50 extrapolations each of length 50 were sampled without replacement using the stochastic beam search method proposed by Kool et al. [144].

We performed local assembly using SPAdes, starting from the last 17 bases of the read, and recorded the portion of each scaffold returned by SPAdes that extended in the direction of extrapolation. We used the `----careful` flag in SPAdes, following Voichek & Weigel [277].

The colors in Figure 2.3A correspond to unique paths through the 17-mer de Bruijn graph. Figure 2.3B plots the per nucleotide perplexity of the sampled extrapolations, i.e.

$$\exp\left(-\sum_{b} p_{\text{extr}}(b|k=(X_{n,i-L},...,X_{n,i-1}))\log p_{\text{extr}}(b|k=(X_{n,i-L},...,X_{n,i-1}))\right)$$

where n indexes the sampled extrapolation and i the position in the sample.

B.12 Visualization details

Here we provide details on the results reported in the **Visualizing data** subsection of the results (Section 2.6).

B.12.1 Latent representation model

As a local latent representation model, we used a categorical probabilistic principal component analysis (pPCA) model, with automatic relevance determination [259,147]. We trained on kmers (k_t, b_t) of

length $L+1=18$ and used $D=20$ latent dimensions. The complete model was,

$$\begin{aligned}
\kappa_d &\sim \text{Exponential for } d \in \{1,\ldots,D\} \\
W_d &\sim \text{Normal}(0_{L+1,|\mathcal{B}|}, 1/\kappa_d) \text{ for } d \in \{1,\ldots,D\} \\
W_0 &\sim \text{Normal}(0_{L+1,|\mathcal{B}|}, 1) \\
z_t &\sim \text{Normal}(0_D, 1) \\
(k_t, b_t) &\sim \text{Categorical}\left(\text{softmax}(W \cdot z_t + W_0)\right)
\end{aligned} \tag{B.58}$$

where $t \in \{1,\ldots,T\}$ runs over all length $L+1$ kmers in the dataset, $0_{L+1,|\mathcal{B}|}$ is an $L+1 \times |\mathcal{B}|$ matrix of zeros, and 0_D is a length D vector of zeros. Here the local variable z_t provides a representation associated with the kmer (k_t, b_t), the global parameter W controls the factors of variation, and κ determines the relevance of each factor through the variance of the prior on W. We trained this latent representation model, and embedded it into a BEAR model, in three stages.

Stage 1 First, we performed stochastic variational inference to learn the parameters of the model [139,213,147]. In particular, we used normally distributed mean field posterior approximations $q(W), q(z|k,b)$, and a deterministic approximation to κ, and optimized the evidence lower bound (ELBO)

$$\begin{aligned}
\mathbb{E}_{W\sim q(W)}\Bigg[\sum_{k,b} \#(k,b)\left(\mathbb{E}_{z\sim q(z|k,b)} \log p\left(k,b|W,z\right) - \textsc{kl}(q(z|k,b)||p(z))\right) \\
+ \log p\left(W|\kappa\right) - \textsc{kl}(q(W)||p(W))\Bigg] + \log p(\kappa)
\end{aligned} \tag{B.59}$$

where $\#(k, b)$ denotes the number of kmers (k, b) seen in the data and the sum runs over all $k \in \mathcal{B}_L^o, b \in \tilde{\mathcal{B}}$. For the local latent variable z, we use a guide (recognition network) $q(z|k, b) = \text{Normal}(\mu(k, b), \sigma(k, b))$ where $\mu(k, b)$ and $\sigma(k, b)$ are each small CNNs. Gradients with respect to the variational approximation parameters were taken using automatic differentiation and the reparameterization trick (elliptical standardization), with one sample for the Monte Carlo approximation at each step.

Stage 2 Once the pPCA model was trained, we approximated its conditional distribution. In particular, we obtained a variational approximation to $p(z|k, (k_t, b_t)_{t=1}^T)$, namely $q(z|k)$, by optimizing the evidence lower bound

$$\mathbb{E}_{W \sim q(W)} \left[\sum_k \#k \left(\mathbb{E}_{z \sim q(z|k)} \log p\left(k|W, z\right) - \text{KL}(q(z|k)||p(z)) \right) \right]. \tag{B.60}$$

Note that $q(W)$ was held fixed, at the value learned in stage 1. $q(z|k)$ was parameterized analogously to $q(z|k, b)$. Now we can approximate the conditional distribution of the pPCA model as

$$p(b|k) \approx \mathbb{E}_{W \sim q(W)} \mathbb{E}_{z \sim q(z|k)} p(b|W, z).$$

This defines an AR model.

Stage 3 Finally, we embedded the conditional pPCA AR model into a BEAR model and optimized h via empirical Bayes (note that here we are not using empirical Bayes to train the BEAR model's embedded AR parameters θ, but instead embedding a pretrained AR model). Since the

variational distribution $q(W)$ was highly concentrated at a single point, we used a computationally convenient approximation to the marginal likelihood of the BEAR model, moving the expectation over the global parameters outside the log marginal likelihood:

$$\mathbb{E}_{W\sim q(W)}\left[\sum_k \log \text{DirichletCategorical}\left(\#(k,\cdot)|\frac{1}{h}\mathbb{E}_{z\sim q(z|k)}p(b|W,z)\right)\right]$$

where DirichletCategorical $(\#(k,\cdot)|\alpha_k)$ denotes the probability of the count vector $\#(k,\cdot)$ under a Dirichlet-Categorical distribution with concentration vector α_k.

Training protocol and hyperparameters The entire variational inference and embedding procedure was implemented using the Edward2 [263] probabilistic programming language with a TensorFlow [4] back-end. We applied the method to the Hodgkin's lymphoma single cell RNAseq described in section B.9, using the same train/test split as for the performance results in Section B.10. Optimization was performed with Adam with a batch size of $125,000$. Gradients were accumulated over 200 steps. The three stages of training described above were repeated iteratively four times until each converged. In each iteration, the first two stages were trained for 5 epochs, and we used a decaying learning rate across iterations $\{0.02, 0.02, 0.01, 0.005\}$; the third stage was trained for 100 batches with a constant learning rate of 0.1 across all iterations.

Inference results At the end of training, the conditional pPCA AR model had a perplexity of 4.28 on heldout data, while the BEAR model had a perplexity of 1.39.

B.12.2 Visualization and annotation

We next sought to understand in greater depth what the BEAR model had learned in the lymphoma dataset.

Reference model We first aimed to understand how the model's predictions differed from predictions based on the reference transcriptome. On the full dataset (combined train/test) we compared the log probability of each read under the pPCA BEAR model to the log probability of each read under a vanilla BEAR model trained on the reference transcriptome (Figure 2.3C; see the supplementary table available with the publication for details on the reference transcriptome). We found a substantial disparity between the two model's predictions, with a number of reads having high probability under the BEAR model but low probability according to the reference model.

Alignments Single cell RNAseq analysis often begins by aligning reads to the reference transcriptome; reads that do not align are typically discarded from further analysis. We performed alignments on the read dataset with hisat2[137] using parameters `--reorder --no-hd --n-ceil L,0,0.001 --no-sq -k 1 -p 4` and with the default hisat2 *Homo sapiens* GRCh38 genome index with transcripts and SNPs, available at https://genome-idx.s3.amazonaws.com/hisat/grch38_snptran.tar.gz. Whether or not each read was successfully aligned is indicated in Figure 2.3C. We observe that many of the reads with low probability under both the pPCA BEAR model and the reference model are unaligned. We also observed a cluster with a large number of unaligned reads, with high probability under the pPCA BEAR model and relatively low probability under the reference model. We focused on a subset of this cluster with particularly high probabilities under the

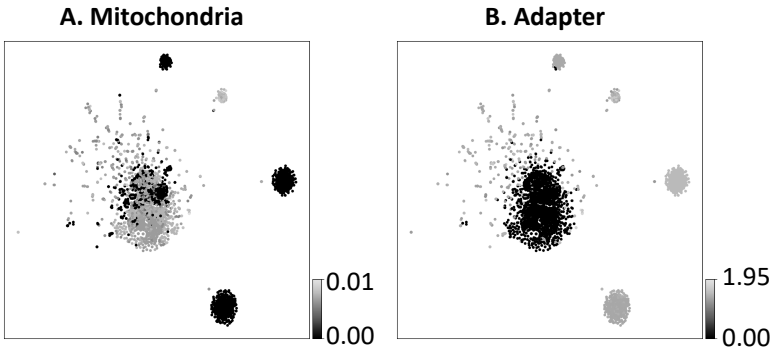


Figure B.16: tSNE visualization of a cluster of single cell RNAseq reads colored by (A) latent embedding distance to the mitochondrial reference genome and (B) latent embedding distance from the sequencing adapter.

pPCA BEAR model for follow-up visualization (black box in Figure 2.3C).

Visualization The pPCA model provides a latent embedding of kmers in a $D = 20$ dimensional continuous space. We sought to visualize the representation of each sequence's kmers in a low dimensional space. To compare two sequences X, X', we defined a measure of dissimilarity,

$$\inf_{i,i'} \textsc{kl}(q(z|X_{i-L:i})||q(z|X'_{i'-L:i'})) + \textsc{kl}(q(z|X'_{i'-L:i'})||q(z|X_{i-L:i})).$$

where $i > L$ and $i' > L$ index positions in X and X' respectively. This dissimilarity measure was used to define a distance matrix over reads in the Hodgkin's lymphoma dataset, which was passed to tSNE[267] to obtain a low-dimensional visualization (Figure 2.3D).

Annotation Observing the clusters in Figure 2.3D, we sought to determine where the reads in each cluster likely originated from, and, by implication, what the reference transcriptome model had trouble explaining in the data. We started by using NCBI's BLAST tool[28] to search for likely sources, and found hits against the mitochondrial genome and the transcript of the gene *JUND*,

part of the AP-1 early response transcription factor. We found that the mitochondrial reads are from a nonreference haplotype, which explains why the reference model gave them low probability. The low likelihood of the *JUND* reads under the reference was due to a TG repeat region in the 3' UTR; similar repeats are present in many variations in different transcripts, thus the particular kmer-base transitions in this case become less likely. We also observed that many reads were chimeric, consisting of fusions of sequences from various parts of the transcriptome with some portion of the sequence CTGTCTCTTATACACATCTCTGAACGGGCTGGCAAGGCAGACCG. The prefix CTGTCTCTTATACACATCT is a standard Illumina Nextera adapter sequence https://support-docs.illumina.com/SHARE/AdapterSeq/illumina-adapter-sequences.pdf, and the remainder of the sequence is presumably part of the primer. The adapter is an experimental artifact (presumably left in the read data due to inaccurate read trimming and quality control), and so is not part of the reference human transcriptome.

We used the same dissimilarity measure as above to compare reads to the mitochondria reference genome and to the adapter sequence CTGTCTCTTATACACATCTCTGAACGGGCTGGCAAGGCAGACCG (Figure B.16). (The distance to each of these sequences was taken to be the minimum of the distance to the forward and reverse complements.) Figure B.16, along with the BLAST results for *JUND*, were the basis for the annotations in Figure 2.3D.

B.13 Hypothesis tests details

Here we provide details on the results reported in the **Testing hypotheses** subsection of the results (Section 2.6).

B.13.1 Kidney transplant metagenomics

The Schreiber et al. [229] data is available for public download, as detailed in the supplementary table available with the publication. The read data was pre-sorted into viral and non-viral reads, but we pooled each of these to reconstruct the full sequencing experiment. We compared the day zero timepoint, i.e. before transplant, to the 4-6 week timepoint, i.e. after transplant, for each patient for which samples from both were available (note this did not include all patients in the study). We used the BEAR two-sample test, with the Jeffreys prior on v, and a truncated uniform prior over lags $1 \leq L \leq 20$. We cross referenced our two-sample test results with whether Schreiber et al. [229] determined there to be likely JC polyomavirus (JCPyV) transmission.

The results are shown in Table B.6, and suggest that JCPyV transmission is associated with an overall shift in the patient microbiome at the sequence level. Patients indicated with an asterisk were diagnosed as having JCPyV before receiving the transplant, and thus the determination of whether the transplant transmitted JCPyV is less certain; for patient wdk036, phylogenetic analysis suggested that the transplant did transmit JCPyV, while for jns976 phylogenetic analysis suggested that it did not. Although the two-sample test results show close correlation with whether or not there was transmission, we caveat them by noting that for very small lags the Bayes factor rejects the null

Table B.6: BEAR two-sample test results, performed on patient metagenome samples from before and after kidney transplant. Bayes factors that reject the null hypothesis are colored red, for easy comparison with whether or not JC polyomavirus (JCPyV) transmission was detected. Asterisks * indicate patients that were already infected with JCPyV before the transplant occurred.

Patient id	JCPyV transmission	log Bayes factor
ume111	True	110407
vpi912	False	234361
iwv346	False	-955252
pqg516	False	-504784
tvy653	True	70223
bgk952	False	-357457
wdk036*	True	3152401
jns976*	False	-199006
aag951	True	242877
qfv506	False	-155391
qnx429	True	369129
poo581	False	-290382
xph346	False	-254856
mek642	False	-348120

hypothesis for all patients; the question of the most "biologically relevant" prior on the lag L is an open question.

B.13.2 *A. thaliana* hypothesis tests

Goodness-of-fit test We trained reference-based AR models (described in Section B.10.1) via maximum likelihood on each *A. thaliana* sequencing dataset (the full dataset, with train/test subsets combined). We used $L = 17$ in the AR model for all three datasets (corresponding the vanilla BEAR maximum marginal likelihood lag for two datasets, see Table B.4). We embedded each trained AR model into a BEAR model to construct a goodness-of-fit test (i.e. we used the learned $f(\theta)$).

We fixed $L = 17$ in the BEAR model (i.e. a deterministic prior over L) to determine if there was misspecification at the same resolution as the AR model. Figure 2.3E plots the Bayes factor as a function of h.

Two-sample tests We simulated sequencing reads based on the *A. thaliana* reference genome using the ART Illumina [111] simulator with parameters `-ss HS20 -p -l 100 -m 200 -s 10 -f 30`. We simulated roughly the same number of reads as was in each real dataset. We examined the Bayes factor $\mathrm{BF}(L) = p((X_n)_{n=1}^N|L)p((X'_n)_{n=1}^{N'}|L)/p((X_n)_{n=1}^N,(X'_n)_{n=1}^{N'}|L)$, computed using vanilla BEAR models for each term (Figure 2.3F). As control experiments, we cut each dataset (and the simulated data) in half, and compared each of these halves to each other using the same two-sample test; as shown by the dotted lines in Figure 2.3F, the two-sample test correctly accepts the null hypothesis in these cases.

Individual log likelihood ratio To understand in detail the differences between the real and simulated data, we computed the conditional individual Bayes factor $\log p(X_n|(X_n)_{n=1}^N) - \log p(X_n|(X'_n)_{n=1}^{N'})$ where $(X_n)_{n=1}^N$ is the real data and $(X'_n)_{n=1}^{N'}$ the simulated data. We approximated the log likelihood using the maximum *a posteriori* value of the transition parameter v under the vanilla BEAR model, and fixed $L = 17$. Computing this likelihood efficiently for each read requires retrieving counts $\#(k,\cdot)$ for each kmer k in the read, which we accomplished using the Jellyfish kmer indexing package [166]. Histograms of the log likelihood ratio of each read X_n in two of the *A. thaliana* datasets are shown in Figure 2.3G (gray).

Annotation Observing the distinct peaks in Figure 2.3G, we sought to determine where the reads in each originated from. We discovered that many reads in the outlier peak from *A. thaliana* 1

matched *Bacillus cereus*, using NCBI's BLAST tool[28]. To annotate the clusters further, we aligned the reads to reference sequences for centromeres, chloroplasts, and *B. cereus*, as well as (if the read did not align to one of these) the reference *A. thaliana* genome (reference sequences are listed in the supplementary table available with the publication). Alignments were performed using `hisat2` on paired end read data using parameters `--reorder --no-hd --n-ceil L,0,0.001 --no-sq -k 1 -p 4` to facilitate subsequent analysis and remove reads with ambiguous bases. The alignment to the centromere included the parameter `--mp 1,1` to allow lower quality alignments. Histograms of the set of reads that align to each reference are shown (stacked on top of one another, not overlayed) in Figure 2.3G.

C

Supplementary Material for Chapter 3

Table C.1: Synthesis model notation.

General notation	**Description**
$\mathbb{Z}_+$	The set of positive non-zero integers.
$\mathbb{R}_+$	The set of positive non-zero reals.
Δ_M	The $M-1$ probability simplex.
$(\Delta_M)^D$	The set of matrices with D rows and each row in Δ_M.
e_j	The length 4 vector of all zeros except a 1 at position j.
Hyperparameter	**Description**
$M \in \mathbb{Z}_+$	Number of templates.
$K \in \mathbb{Z}_+$	Number of pools.
$L_k \in \mathbb{Z}_+$	Length (in codons) of templates in pool $k \in \{1, \ldots, K\}$.
$L = \sum_k L_k$	Total length (in codons) of generated sequences.
$A \in \mathbb{Z}_+$	Number of codon or nucleotide mixtures.
$S \in \mathbb{R}_+^4$	Substitution matrix. $S_{b',b}$ is the probability of mutating b to b'.
	$S^\top \in (\Delta_4)^4$ where $S^\top$ is the transpose of S.
	The columns of S are linearly independent.
$T \in \{0,1\}^{64 \times 21}$	Translation matrix, mapping from codons
	to the twenty amino acids plus the stop codon.
	$T_{(b_1,b_2,b_3)d} = 1$ if (b_1, b_2, b_3) codes for d,
	and $T_{(b_1,b_2,b_3)d} = 0$ otherwise.
	We assume the standard (universal) codon table.
	Note $\sum_{b_1,b_2,b_3} T_{(b_1,b_2,b_3)d} \geq 1$ for all $d \in \{1, \ldots, 21\}$.
Codon diversification model	**Description**
$\mathcal{U} = \Delta_{64}$	*Arbitrary codon mixtures.*
$\mathcal{U} = \{v_1, \ldots, v_A\}$	*Finite codon mixtures.*
$\mathcal{U} = \{v_1, \ldots, v_A\} \otimes \{v_1, \ldots, v_A\} \otimes \{v_1, \ldots, v_A\}$	*Finite nucleotide mixtures.*
	Nb. in this model, the probability of a codon (b_1, b_2, b_3) is
	the product of mixture probabilities $v_{a_1 b_1} v_{a_2 b_2} v_{a_3 b_3}$
	where $a_1, a_2, a_3 \in \{1, \ldots, A\}$
$\mathcal{U} = \{S^\tau e_1, \ldots, S^\tau e_4\} \otimes \{S^\tau e_1, \ldots, S^\tau e_4\}$	*Enzymatic mutagenesis.*
$\otimes \{S^\tau e_1, \ldots, S^\tau e_4\}$	Nb. in this model, the probability of a codon (b_1, b_2, b_3) is
	the product of mixture probabilities $S^\tau_{b_1 a_1} S^\tau_{b_2 a_2} S^\tau_{b_3 a_3}$
	where $a_1, a_2, a_3 \in \{1, \ldots, 4\}$
Assembly model	**Description**
$Z_{i1} \sim \text{Categorical}(w)$	*Fixed assembly*
$Z_{i2} := \ldots := Z_{iK} := Z_{i1}$	
$Z_{ik} \sim \text{Categorical}(w_k)$ for all $k \in \{1, \ldots, K\}$	*Combinatorial assembly*

Continued on next page...

Continued from previous page...

Parameter	Description
w	Template probabilities. $w \in \Delta_M$ if using fixed assembly. $w \in (\Delta_M)^K$ if using combinatorial assembly.
v	Nucleotide or codon mixture probabilities. $v \in (\Delta_{64})^A$ if using finite codon mixtures, $v \in (\Delta_4)^A$ if using finite nucleotide mixtures.
τ	Number of rounds of mutagenesis. $\tau \in \mathbb{Z}_+$.
u	Template defining codon probabilities. $u_{kzj} \in \mathcal{U}$ for all $k \in \{1, \ldots, K\}$, $z \in \{1, \ldots, M\}$ and $j \in \{1, \ldots, L_k\}$
Latent variable	**Description**
$Z_i \in \{1, \ldots, M\}^K$	Templates to generate sequence i. Z_{ik} is the template drawn from pool k.
$C_i \in (\Delta_{64})^L$	Codon probabilities to generate sequence i. $C_{ij(b_1,b_2,b_3)}$ is the probability of generating codon (b_1, b_2, b_3) at position j.
$H_i \in \{0,1\}^{L \times 64}$	Codons of generated sequence i. $H_{ij(b_1,b_2,b_3)} = 1$ if the codon (b_1, b_2, b_3) is at position j, and $H_{ij(b_1,b_2,b_3)} = 0$ otherwise.
Observed variable	**Description**
$X_i \in \{0,1\}^{L \times 21}$	The ith generated protein sequence, one-hot encoded and including the stop codon. $X_{ijd} = 1$ if the amino acid d is at the jth position, and $X_{ijd} = 0$ otherwise.

C.1 Model details and limitations

In this section we explain further the synthesis models proposed in Section 3.2.1, as well some of the limitations of our mathematical idealization.

Physically, for the finite codon or nucleotide mixture models, codon diversification happens during chemical synthesis of oligos (DNA segments). DNA in each well (or isolated reaction volume) is synthesized position by position, with mixtures of nucleotides or codons (trinucleotides) added in defined ratios one at a time, such that a large number of different molecules is eventually constructed. Twist Bioscience's combinatorial variant libraries, which can achieve arbitrary codon mix-

tures, rely on proprietary technology; however, it produces analogous results[264]. For all of these technologies, what we refer to as a "template" corresponds physically to a very large number of molecules in an individual well, with independent nucleotide or codon probabilities at each site. We assume that the number of molecules is effectively infinite in comparison to N_1, such that we do not need to account for sampling noise at this stage. We ignore the possibility of skipped positions, where nucleotides or codons randomly fail to add to the growing oligos, a type of error that is sometimes of particular concern for trimer-based synthesis. We enforce the constraint that the number of mixtures A is finite and small, since to the best of our knowledge commercially available technologies have this requirement, but it is not necessarily a fundamental technological constraint[194].

Physically, for the enzymatic mutagenesis model, template oligos are synthesized deterministically, such that there is a large population of identical molecules in each well. Codon diversification occurs only after assembly (i.e. after oligos from different wells are combined) and may take place either *in vitro* or *in vivo*. We assume that there is an error correction mechanism after each round of mutagenesis, such that each strand of each DNA molecule has effectively gone through the same number of rounds of mutagenesis; in some ePCR protocols error correction is not used, and so alternative models may be more appropriate[182,204]. We also assume that the mutation probability depends only on individual nucleotides, and not their sequence context, although empirically dependencies on sequence context (especially the adjacent two nucleotides) can be found[8]. Finally, we require that each template undergoes the same number of rounds of mutagenesis τ, with the same enzyme and thus the same S. For small M, it can be experimentally tractable in many cases to use different τ, and even different S, for each template, in which case the model should be adjusted to

make τ and S depend on the template.

Physically, assembly requires joining oligos together using e.g. Gibson assembly[89]. For the fixed assembly model, the oligos corresponding to the kth template in each pool must be joined in an isolated reaction, for all $k \in \{1, \ldots, K\}$; in combinatorial assembly, the sets of oligos corresponding to each template in each pool are first mixed, and then oligos from these combined pools are joined. Assembly requires short overhangs, sequences that closely match one another, at the ends of each oligo that are to be joined. Our synthesis model ignores any restrictions that come from overhangs needing to match, as well as variation in assembly probability that depend on overhang mismatch. Our model also assumes full control over the relative concentration of templates, w. While this is tractable for low M, it may be more challenging for large M, particularly if technologies like Dropsynth are used for fixed assembly[200].

C.2 Optimization details

C.2.1 Exact Solutions

As an example of a target sequence model which we can exactly match, consider a RegressMuE[284], which has been used for forecasting the evolution of influenza. Let $\mathbf{B}$ be a covariate vector (e.g. a future time), let $\mathbf{A}$ be the regression coefficients, and let $\mathbf{W}$ be the latent alignment. The predictive distribution $p(x|\mathbf{B}, \mathbf{W}, \mathbf{A})$ can be written as Categorical($\mathbf{U}$), where $\mathbf{U}$ is a matrix of independent amino acid probabilities over L positions. We can exactly match this distribution with a synthesis model using $M = 1$ templates, fixed assembly and arbitrary codon mixtures.

We can also approximate the posterior predictive distribution. Let $p(\mathbf{A}|\mathcal{D})$ be the posterior distribution over regression parameters given the training data. The posterior predictive distribution can be approximated as $\sum_{m=1}^{M} \frac{1}{M} p(x|\mathbf{B}, \mathbf{W}, \mathbf{A}_m)$ where $\mathbf{A}_1, \ldots, \mathbf{A}_M \sim p(\mathbf{A}|\mathcal{D})$ are posterior samples. This distribution can be exactly matched by a stochastic synthesis model using fixed assembly with $w = (\frac{1}{M}, \ldots, \frac{1}{M})$ and arbitrary codon mixtures.

C.2.2 Stochastic EM

We used the online EM algorithm proposed by Cappé & Moulines [33], modified to update using minibatches instead of individual datapoints. Here we derive the algorithm for the stochastic synthesis model (Equation 3.1). Without loss of generality, we focus on combinatorial assembly models; the fixed assembly case can be obtained by setting $K = 1$. The local variable of the synthesis model is Z_i, which we represent here as a one-hot encoding, i.e. $Z_i \in \{0, 1\}^{K \times M}$. At iteration t of the optimization algorithm, given the current parameter estimate $\theta^{(t)} = (w^{(t)}, u^{(t)}, v^{(t)}, \tau^{(t)})$, the conditional expectation of Z_i can be written as

$$r_{ikm} := \mathbb{E}_{q_{\theta^{(t)}}}[Z_{ikm}|X_i] = \frac{w_{k,m} \exp\left(\sum_{j=1}^{L_k} \log(u_{kmj} \cdot T) \cdot X_{i(j+\bar{L}_k)}\right)}{\sum_{m'=1}^{M} w_{k,m'} \exp\left(\sum_{j=1}^{L_k} \log(u_{km'j} \cdot T) \cdot X_{i(j+\bar{L}_k)}\right)}, \qquad \text{(C.1)}$$

where $\bar{L}_k = \sum_{k'<k} L_k$. Now we can compute the conditional expectation of the mean log likelihood as

$$\begin{aligned} Q_{\theta^{(t)}}(X_1, \ldots, X_N; \theta) &:= \frac{1}{N} \mathbb{E}_{q_{\theta^{(t)}}}[\log q_\theta(X_1, \ldots, X_N, Z_1, \ldots, Z_N)] \\ &= \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M} \left[\sum_{j=1}^{L_k} \log(u_{kmj} \cdot T) \cdot X_{i(j+\bar{L}_k)} r_{ikm} + \log w_{km} r_{ikm} \right]. \end{aligned} \tag{C.2}$$

In standard EM, we would optimize this function with respect to θ. However, this requires summing over the whole dataset at each step. To derive the stochastic EM algorithm, we rewrite $Q_{\theta^{(t)}}$ in terms of summary statistics of the data that can be estimated from minibatches. In particular, let $\mathcal{S} \subseteq \{1, \ldots, N\}$ be a subset of the data, and define the summary statistics

$$\begin{aligned} \bar{s}^{(1)}(X_{\mathcal{S}}; \theta^{(t)})_{kmj} &:= \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} X_{ij} r_{ikm}, \\ \bar{s}^{(2)}(X_{\mathcal{S}}; \theta^{(t)})_{km} &:= \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} r_{ikm}. \end{aligned} \tag{C.3}$$

Now we can estimate $Q_{\theta^{(t)}}$ as

$$\hat{Q}(\bar{s}; \theta) := \sum_{k=1}^{K} \sum_{m=1}^{M} \left[\sum_{j=1}^{L_k} \log(u_{kmj} \cdot T) \cdot \bar{s}^{(1)}_{km(j+\bar{L}_k)} + \log w_{km} \bar{s}^{(2)}_{km} \right]. \tag{C.4}$$

The complete algorithm alternates between estimating summary statistics from minibatches of data $\mathcal{S}^{(t)}$ drawn at each step and maximizing the estimated expected log likelihood $\hat{Q}_{\theta^{(t)}}$,

$$\begin{aligned}\hat{s}^{(t+1)} &= \hat{s}^{(t)} + \gamma^{(t+1)}(\bar{s}(X_{\mathcal{S}^{(t)}}; \theta^{(t)}) - \hat{s}^{(t)}) \\ \theta^{(t+1)} &= \underset{\theta}{\text{argmax}}\, \hat{Q}(\hat{s}^{(t+1)}; \theta)\end{aligned} \tag{C.5}$$

where $\gamma^{(t)}$ is the step size. As suggested by Cappé & Moulines [33], we set $\gamma^{(t)} = t^{-0.6}$. We also use Polyak-Ruppert averaging, as suggested by Cappé & Moulines [33], taking the mean of the summary statistics $\hat{s}^{(t)}$ for the last half of training, i.e. $\hat{s}^* = \frac{2}{t_{\max}} \sum_{t=(t_{\max}/2+1)}^{t_{\max}} \hat{s}^{(t)}$, and producing the final parameter estimate $\hat{\theta}^* = \text{argmax}_\theta \hat{Q}(\hat{s}^*; \theta)$.

The maximization step $\theta^{(t+1)} = \text{argmax}\, \hat{Q}(\hat{s}^{(t)}; \theta)$ can vary depending on the codon diversification technology used. For all technologies, we have

$$w^{(t+1)} = \hat{s}_{km}^{(t+1)(2)}. \tag{C.6}$$

For arbitrary codon mixtures and finite codon mixtures, we can without loss of generality pick one codon for each amino acid and the stop symbol, and work with template probabilities $\tilde{u}$ directly over amino acids, i.e. where $\tilde{u}_{k,m,j,d}$ is the probability of amino acid d at position j of template m in pool k. Then, for arbitrary codon mixtures,

$$\tilde{u}_{kmjd}^{(t+1)} = \frac{\bar{s}_{km(j+\bar{L}_k)d}^{(t+1)(1)}}{\sum_{d'=1}^{21} \bar{s}_{km(j+\bar{L}_k)d'}^{(t+1)(1)}}. \tag{C.7}$$

For finite codon mixtures, let $\tilde{\chi}_{kmj}$ be a one-hot encoding of the codon mixture used at position j of template m in pool k, such that $\tilde{\chi}_{kmj} \in \{0,1\}^A$. We work directly with mixtures defined over amino acids, with $\tilde{v}_{ad}$ the probability of amino acid d in mixture a. Thus $\tilde{u}_{kmj} = \tilde{\chi}_{kmj} \cdot \tilde{v}$. Then we can use the coordinate-wise update

$$\begin{aligned}
\tilde{\chi}_{kmj}^{(t+1)} &= \text{argmax}_a \sum_{d=1}^{21} \log(\tilde{v}_{ad}) \hat{s}_{km(j+\bar{L}_k)d}^{(t+1)(1)} \\
\tilde{v}_{ad}^{(t+1)} &= \frac{\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{j=1}^{L_k} \hat{s}_{km(j+\bar{L}_k)d}^{(t+1)(1)} \tilde{\chi}_{kmja}^{(t+1)}}{\sum_{d'=1}^{21}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{j=1}^{L_k} \hat{s}_{km(j+\bar{L}_k)d'}^{(t+1)(1)} \tilde{\chi}_{kmja}^{(t+1)}}
\end{aligned} \tag{C.8}$$

For finite nucleotide mixtures, we use χ_{kmj1} to denote a one-hot encoding of the mixture used at the first position of the codon at position j in template m in pool k, i.e. $\chi_{kmj1} \in \{0,1\}^A$, and likewise for χ_{kmj2} and χ_{kmj3}. We update χ by optimizing over all three positions of each codon jointly, enumerating all combinations of a_1, a_2 and a_3,

$$\chi_{kmj}^{(t+1)} = \underset{(a_1,a_2,a_3)}{\text{argmax}} \sum_{d=1}^{21} \log\Big(\sum_{b_1,b_2,b_3} v_{a_1 b_1} v_{a_2 b_2} v_{a_3 b_3} T_{(b_1,b_2,b_3)d} \Big) \bar{s}_{km(j+\bar{L}_k)d}^{(t+1)(1)}. \tag{C.9}$$

Once χ has been updated, we update v. This is harder, as there is no closed form solution. We directly optimize $\hat{Q}$ with respect to v by taking gradients and applying 5 steps of the Adam optimizer[138] with a learning rate of 0.01 (that is, we take 5 steps of Adam for every 1 EM update). For enzymatic mutagenesis, we can also apply Equation C.9 to update χ, replacing v with S^τ. To update τ, we directly enumerate all values of $\hat{Q}$ for $\tau \in \{1, \ldots, \tau_{\text{max}}\}$ and choose the maximum.

Code implementing the stochastic EM algorithm for all of the proposed stochastic synthesis

models is available in the Supplementary Material.

C.2.3 Choosing $\tilde{N}$

Recall that our proposed black-box optimization procedure is to draw $X_1, \ldots, X_{\tilde{N}} \sim p$ computationally and then maximize the synthesis model parameters,

$$\hat{\theta}_{\tilde{N}} := \text{argmax}_\theta \sum_{i=1}^{\tilde{N}} \log q_\theta(X_i). \tag{C.10}$$

In this section, we argue that $\tilde{N}$ should be chosen to be either equal to N_1, or, if N_1 is too large to be tractable computationally, $\tilde{N}$ should be as large as is tractable. In particular, we *do not* suggest choosing $\tilde{N}$ to be larger than N_1, nor do we suggest regularizing θ as one would in a standard inference problem. The reason is that "overfitting" the synthesis model to the samples $X_1, \ldots, X_{\tilde{N}}$ can help rather than hurt.

To be more precise, consider the extreme case where q_θ can exactly match the empirical distribution of $X_1, \ldots, X_{N_1} \sim p$ but cannot exactly match p itself. For example, this situation can occur when using fixed assembly and $M = N_1$, allowing each mixture component be a point mass. If we use $\tilde{N} = N_1$, we find

$$q_{\hat{\theta}_{\tilde{N}}}(x) = \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} \delta_{X_i}(x) \tag{C.11}$$

where $\delta_{x'}(x)$ is the Kronecker delta function at x'. In this case, variational synthesis is equivalent to large-scale MC synthesis, and will produce N_1 samples from p.* On the other hand, if we let

*Technically, variational synthesis in this case produces a size N_1 bootstrap of N_1 samples from p, rather

$\tilde{N} \to \infty$, we have $\hat{\theta}_{\tilde{N}} \to \theta^*$. In this case, variational synthesis will produce N_1 samples from $q_{\theta^*} \neq p$. Thus, it can be preferable to use $\tilde{N} = N_1$ as compared to $\tilde{N} > N_1$, since using $\tilde{N} = N_1$ leads to synthesis of N_1 exact samples from p instead of N_1 samples from $q_{\theta^*} \neq p$.

In practice, of course, q_θ will rarely be able to exactly match the empirical distribution of samples from p. Nonetheless, we expect using $\tilde{N} \approx N_1$ to be useful, as in this case we avoid trying to match $q_{\hat{\theta}_{\tilde{N}}}$ to components of p that are too rare to occur in practice, and instead regularize $q_{\hat{\theta}_{\tilde{N}}}$ towards the empirical distribution of samples from p.

C.2.4 Variable length protein sequences

To handle variable length protein sequences, we treat everything past the stop codon as missing data which does not contribute to the likelihood. That is, for a sequence X_i with a stop codon at position $\mathbf{j}$, we have $q_\theta(X_i) = q_\theta(X_{i,1:\mathbf{j}})$.

C.3 Related work details

C.3.1 DeCoDe

DeCoDe can be applied to datasets of fixed-length (or aligned) sequences, $X'_1, \ldots, X_{N'}$, which are assumed to be unique (i.e. $X'_i \neq X'_{i'}$ if $i \neq i'$). Consider the empirical distribution $p(x) = \sum_{i=1}^{N'} \delta_{X'_i}(x)$ where $\delta_{x'}(x)$ is the Kronecker delta. Take q_θ to be a stochastic synthesis model using finite nucleotide mixtures and fixed assembly, with $\theta = (w, u, v)$. Let $\text{supp}(p)$ denote the support

than directly producing N_1 samples from p. Although bootstrapping introduces some additional sampling noise, we expect it is unlikely in practice to make using $q_{\hat{\theta}_{\tilde{N}}}$ worse than using q_{θ^*}, since the bootstrap directly approximates p. Section C.4.1 discusses this subtlety further.

of p, i.e. the set of all length L sequences with non-zero probability. Let $\zeta \in \mathbb{Z}_+$ denote the maximum allowed support of q_θ. Then, we can rewrite the DeCoDe objective (Section 2.2.2 in Shimko et al. [234]) in terms of the size of the intersection of supports of p and q_θ,

$$\theta^* = \underset{\theta:\text{supp}(q_\theta)\leq\zeta}{\text{argmax}} |\text{supp}(p) \cap \text{supp}(q_\theta)|. \tag{C.12}$$

Note that the size of the intersection of supports does not correspond to a valid divergence between p and q_θ.

C.3.2 SCHEMA

RASPP[73] is an algorithm for designing site-directed recombination or combinatorial assembly libraries based on a crystal structure and a dataset of homologous proteins from the same family. It chooses a set of template lengths $L_1, \ldots, L_K$, where $L_{\text{min}} \leq L_k \leq L_{\text{max}}$ for $k \in \{1, \ldots, K\}$, in order to minimize the SCHEMA score, roughly the number of structural contacts between positions of the protein generated by different template pools. In this section we give a heuristic argument connecting RASPP to variational synthesis, in the special case where RASPP finds a solution with no structural contacts across regions covered by each pool.

Consider a target model p that consists of a Potts model learned from the same protein family as the dataset of homologous proteins. In general, the Potts model will infer energetic interactions only between positions of the alignment that are in structural contact[168]. Let $\tilde{L}_k$ denote the region generated by template k, i.e. $\tilde{L}_1 = \{1, \ldots, L_1\}$, $\tilde{L}_2 = \{L_1 + 1, \ldots, L_1 + L_2\}$, etc. and let $p(x_{\tilde{L}_k})$

denote the marginal of p over these positions. For the set of $L_1, \ldots, L_k$ chosen by RASPP, we have no structural contacts across regions, and so no energetic interactions under the Potts model, and thus $p(x_{\tilde{L}_k}, x_{\tilde{L}_{k'}}) = p(x_{\tilde{L}_k})p(x_{\tilde{L}_{k'}})$ for $k \neq k'$. In other words, there is no correlation between segments under the Potts model p. When using stochastic synthesis with combinatorial assembly, there is also no correlation between segments under q_θ. If we try to minimize the KL divergence between a q_θ with combinatorial assembly and the Potts model p, and optimize the template lengths $L_1, \ldots, L_K$, we can expect in general to find a similar solution to RASPP, where both the SCHEMA score and the correlation between templates under p is zero.

C.4 Theory details

Note that the proofs in this section rely on the definitions in Table C.1.

C.4.1 The MC synthesis estimator

In our theoretical analysis we do not treat MC synthesis as variational synthesis with point mass (deterministic) mixture components. In particular, we analyze the estimator

$$\begin{aligned} X_1, \ldots, X_{N_0} &\sim p, \\ \hat{I}^{(a)} &:= \frac{1}{N_0} \sum_{i=1}^{N_0} f(X_i), \end{aligned} \tag{C.13}$$

which comes from measuring each synthesized sequence individually, and *not* the alternative estimator

$$\begin{aligned} X'_1, \ldots, X'_{N_0} &\sim p, \\ X_1, \ldots, X_{N_1} &\sim \frac{1}{N_0} \sum_{i=1}^{N_0} \delta_{X'_i}(x), \\ \hat{I}'^{(a)} &:= \frac{1}{N_1} \sum_{i=1}^{N_1} f(X_i). \end{aligned} \tag{C.14}$$

which would come from pooling the synthesized sequences and then measuring a random sample of size N_1 (here $\delta_{x'}(x)$ is the Kronecker delta function at x'). Note this alternative estimator $\hat{I}'^{(a)}$ takes the form of a bootstrap estimator of size N_1, taken from an initial sample of size N_0 from p, and thus in general introduces additional sampling noise as compared to $\hat{I}^{(a)}$. There are three reasons for focusing our analysis on $\hat{I}^{(a)}$ instead of $\hat{I}'^{(a)}$. First, since N_0 is low, in practice it is often tractable for experimentalists to measure the N_0 sequences individually (e.g. in 96 well plates), rather than pooling them, making the estimate $\hat{I}^{(a)}$ possible. Second, in the limit where N_1 is much greater than N_0, the estimators converge, making $\hat{I}^{(a)}$ a reasonable approximation for pooled experiments in practice. Third, we want our analysis to be conservative in measuring the benefits of variational synthesis vis-à-vis the alternative, MC synthesis, so we use the better estimator $\hat{I}^{(a)}$.

C.4.2 Proof of Proposition 3.4.1

Proof. Using Jensen's inequality,

$$\begin{aligned}\frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \mathbb{E}[|\hat{I}^{(a)} - I|] &\leq \frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \sqrt{\frac{1}{N_0^2} \mathbb{E}_p \left[\left(\sum_{i=1}^{N_0} (f(X_i) - \mathbb{E}_p[f(X)]) \right)^2 \right]} \\ &\leq \frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \sqrt{\frac{\mathbb{V}_p[f(X)]}{N_0}} \leq \frac{1}{\sqrt{N_0}}\end{aligned} \tag{C.15}$$

where $\mathbb{V}_p[f(x)]$ is the variance with respect to p.

We can decompose the error in the $\hat{I}^{(b)}$ estimate into variance and bias terms, and then apply a similar analysis.

$$\begin{aligned}\frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \mathbb{E}[|\hat{I}^{(b)} - I|] &\leq \frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \mathbb{E}[|\hat{I}^{(b)} - \mathbb{E}_{q_{\theta^*}}[f(X)]|] + \frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} |\mathbb{E}_{q_{\theta^*}}[f(X)] - \mathbb{E}_p[f(X)]| \\ &\leq \frac{1}{\sqrt{N_1}} + \textsc{tv}(p, q_{\theta^*}).\end{aligned} \tag{C.16}$$

where we have used the integral probability metric representation of the total variation metric $\textsc{tv}(\cdot, \cdot)$[242]. The result follows from application of Pinsker's inequality. □

We can see from the proof that the bound in Equation 3.3 could be tighter if we use total variation in place of KL. It could also be tighter if we restrict the family of functions $\mathcal{F}$ further. In particular, consider the metric space defined over the set of fixed length discrete sequences $\mathcal{X}$ with the Hamming distance $\|x - x'\|_H := \sum_{j=1}^{L} \sum_{d=1}^{21} \frac{1}{2} |x_{jd} - x'_{jd}|$ (where x is a one hot en-

coding of a length L nucleotide sequence). Then, we can introduce the function family $\mathcal{F}_W := \{f : \|f\|_L \leq D_{\max}\}$, that is, the set of functions with bounded Lipschitz constant $\|f\|_L := \max_{x,x' \in \mathcal{X}, x \neq x'} |f(x) - f(x')|/\|x - x'\|_H$. Biologically, the Lipschitz constant is interpretable as the sensitivity of a sequence's biological function to single mutations. In particular, if a point mutation can dramatically change the assayed property of the sequence, then the Lipschitz constant will be large; otherwise it will be small. If we assume the Lipschitz constant of the experimental assay is bounded by some constant $D_{\max}$, we can find an alternative error bound on the stochastic synthesis estimator:

$$\frac{1}{f_{\max}} \sup_{f \in \mathcal{F} \cap \mathcal{F}_W} \mathbb{E}[|\hat{I}^{(b)} - I|] \leq \frac{1}{\sqrt{N_1}} + \frac{D_{\max}}{f_{\max}} \mathrm{W}(p, q_{\theta^*}). \tag{C.17}$$

where $\mathrm{W}(p, q) := \inf_{\gamma \in \Gamma(p,q)} \int \|x - x'\|_H \gamma(x, x')$ is the first Wasserstein distance, with $\Gamma(p, q)$ the set of couplings of p and q. This result follows from Equation C.16 by applying the Kantorovich-Rubinstein duality theorem (e.g. Dudley[64], Theorem 11.8.2), using the fact that the metric space of finite sequences with the Hadamard distance is a finite discrete space and separable. We see from Equation C.17 that the error bound on variational synthesis can be lower than that in Equation C.16, so long as $D_{\max}$ is sufficiently small. In other words, we can get away with using synthesis models that do not match p closely if the assay is not very sensitive to small changes in sequence.

C.4.3 Importance sampling estimates

In some cases we can get access to paired sequence and function data, and in particular the dataset $\mathcal{D}_0 := \{(f(X_i), X_i) : f(X_i) \neq 0\}$. For instance, if we deep sequence the hits of a screen, with

$f : \mathcal{X} \mapsto \{0, 1\}$, we will have $\mathcal{D}_0 := \{(1, X_i) : f(X_i) = 1\}$. We can then construct an importance-sampling estimate of $I = \mathbb{E}_p[f(X)]$ using samples $X_1, \ldots, X_{N_1} \sim q_{\theta^*}$,

$$\hat{I}^{(c)} := \frac{1}{N_1} \sum_{i=1}^{N_1} f(X_i) \frac{p(X_i)}{q_{\theta^*}(X_i)} = \frac{1}{N_1} \sum_{X_i \in \mathcal{D}_0} f(X_i) \frac{p(X_i)}{q_{\theta^*}(X_i)}.$$

Unlike $\hat{I}^{(b)}$, this estimator is unbiased: $\mathbb{E}_{q_{\theta^*}}[f(X)p(X)/q_{\theta^*}(X)] = I$. However, $\hat{I}^{(c)}$ still takes advantage of a large number of samples, making possible lower variance than $\hat{I}^{(a)}$. In particular, we have

$$\frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \mathbb{E}_{q_{\theta^*}}[|\hat{I}^{(c)} - I|] \leq \frac{1}{\sqrt{N_1}} \sqrt{\textsc{chi}(p||q_{\theta^*})} \tag{C.18}$$

where CHI is the chi divergence, which can be defined as $\textsc{chi}(p||q) = \mathbb{V}_q[\frac{p(X)}{q(X)}]$. We can derive this result following the same analysis as in Equation C.15,

$$\frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \mathbb{E}_{q_{\theta^*}}[|\hat{I}^{(c)} - I|] \leq \frac{1}{f_{\max}} \sup_{f \in \mathcal{F}} \sqrt{\frac{\mathbb{V}_{q_{\theta^*}}[f(X)\frac{p(X)}{q_{\theta^*}(X)}]}{N_1}} \leq \frac{1}{\sqrt{N_1}} \sqrt{\textsc{chi}(p||q_{\theta^*})}. \tag{C.19}$$

Note that our suggested black-box optimization procedure for variational synthesis (Section 3.2.2) is intended to help ensure high discovery rates (maximizing $\sum_{i=1}^{N_1} f(X_i)$) but not to ensure accurate importance sampling estimates. In particular, the KL divergence does not provide a particularly tight bound on the CHI divergence (see e.g. Proposition 2 in Dragomir [60]), so it is likely preferable to (if possible) directly optimize the CHI divergence [53].

C.4.4 Proof of Corollary 3.4.2

Proof. We have

$$\mathbb{E}[N_1\hat{I}^{(b)} - N_0\hat{I}^{(a)}] \geq (N_1 - N_0)I - N_1 \sup_{f\in\mathcal{F}} \mathbb{E}[|\hat{I}^{(b)} - I|] - N_0 \sup_{f\in\mathcal{F}} \mathbb{E}[|\hat{I}^{(a)} - I|] \quad \text{(C.20)}$$

Applying Proposition 3.4.1 yields the result. □

C.4.5 Proof of Proposition 3.4.3

Before proving Proposition 3.4.3, we first prove a lemma that shows – as long as we are not using enzymatic mutagenesis – that we can construct templates that are arbitrarily close to a point mass while still having full support. We use $q_\theta(x|c)$ as shorthand for $q_\theta(x|C_i = c)$, and $\delta_{x'}(x)$ to denote the Kronecker delta function which takes value 1 if $x = x'$ and 0 otherwise.

Lemma C.4.1. *Assume we are using **arbitrary codon mixtures**, **finite codon mixtures** (with $A \geq 21$), or **finite nucleotide mixtures** (with $A \geq 4$). For any $\epsilon > 0$ sufficiently small, there exists some v such that:*

for all $\bar{x} \in \mathcal{X}$ there exists a $\bar{c}(\bar{x}) \in \mathcal{U}^L$ such that

$$q(\bar{x}|\bar{c}(\bar{x})) \geq 1 - \rho L\epsilon \quad \text{(C.21)}$$

where ρ is a positive constant, and $\text{supp}(q(x|\bar{c}(\bar{x}))) = \mathcal{X}$. *In particular, for arbitrary or finite codon mixtures, $\rho = 1$, while for finite nucleotide mixtures, $\rho = 3$.*

Proof. We start with the finite codon mixtures case; note that this immediately implies the arbitrary codon mixture case, since the space $\mathcal{U}$ for finite codon mixtures is a subset of the space $\mathcal{U}$ for arbitrary codon mixtures. We choose (arbitrarily) one codon for each amino acid and the stop symbol, and work with mixtures v over these 21 codons (setting the probability of all others to zero). For all $d \in \{1, \ldots, 21\}$, let $v_d = \mathbf{1}_{21} \frac{\epsilon}{21} + e_d^{(21)}(1-\epsilon)$ where $\mathbf{1}_D$ is the length D vector of all ones and $e_d^{(D)}$ is the length D vector of all zeros except a one at position d. Let $\ell(\bar{x})$ be the length of a protein sequence $\bar{x}$.[†] Given $\bar{x}$ we define the $L \times 21$ matrix $\bar{c}(\bar{x}) = \text{concatenate}(v_{\bar{x}_1}, \ldots, v_{\bar{x}_{\ell(\bar{x})}}, v_1, \ldots, v_1)$. Now note that

$$q(\bar{x}|\bar{c}(\bar{x})) \geq (1-\epsilon)^{\ell(\bar{x})} \geq 1 - L\epsilon. \tag{C.22}$$

Next we consider the finite nucleotide mixtures case, which works similarly. For all $b \in \{1, \ldots, 4\}$, let $v_b = \mathbf{1}_4 \frac{\epsilon}{4} + e_b^{(4)}(1-\epsilon)$. Given a protein sequence $\bar{x}$, choose a particular codon for each amino acid and the stop symbol. This defines a DNA sequence $\tilde{x}$, where $\tilde{x}_{j1}$ is the nucleotide in the first position of the codon for the amino acid at position j of $\bar{x}$, and likewise for $\tilde{x}_{j2}$ and $\tilde{x}_{j3}$. We can then choose nucleotide mixtures for each position of a template to match $\tilde{x}$, that is,

$$\bar{c}(\bar{x}) = \text{concatenate}(v_{\tilde{x}_{11}} \otimes v_{\tilde{x}_{12}} \otimes v_{\tilde{x}_{13}}, \ldots, v_{\tilde{x}_{\ell(\bar{x})1}} \otimes v_{\tilde{x}_{\ell(\bar{x})2}} \otimes v_{\tilde{x}_{\ell(\bar{x})3}}, v_1 \otimes v_1 \otimes v_1, \ldots, v_1 \otimes v_1 \otimes v_1).$$

Now we have

$$q(\bar{x}|\bar{c}(\bar{x})) \geq (1-\epsilon)^{3\ell(\bar{x})} \geq 1 - 3L\epsilon. \tag{C.23}$$

[†]Length is measured up to (and including) the first stop codon or L, whichever comes first.

□

We are now ready to prove Part 1 of Proposition 3.4.3. The basic idea is to construct a synthesis distribution q_{θ^*} that closely approximates p by convolving with p templates that are approximate point masses.

Part 1 of Proposition 3.4.3: *When using either **arbitrary codon mixtures**, **finite codon mixtures** (with $A \geq 21$), or **finite nucleotide mixtures** (with $A \geq 4$): for any $p \in \mathcal{P}(\mathcal{X})$ and $\eta > 0$ there exists some M and θ such that (1) $\text{KL}(p\|q_\theta) < \eta$ and (2) $\text{supp}(q_\theta(x|z)) = \mathcal{X}$ for all $z \in \{1, \ldots, M\}$.*

Proof. Let $M = |\mathcal{X}|$, that is, set the number of templates equal to the total number of sequences of length less than or equal to L. Since $\mathcal{X}$ is finite, we can construct for any $\epsilon > 0$ the synthesis distribution $q_\theta(x) = \mathbb{E}_{\bar{X}\sim p}[q(x|\bar{c}(\bar{X}))]$. In this synthesis distribution, the weights w of each mixture component are set by $p(x)$, and $\text{supp}(q_\theta(x|z)) = \mathcal{X}$ for all z by the construction of $\bar{c}$. We now have, applying Lemma C.4.1,

$$\begin{aligned}
\text{KL}(p\|q_\theta) &= \sum_{x\in\mathcal{X}} p(x)\log p(x) - \sum_{x\in\mathcal{X}} p(x)\log\Big[\sum_{\bar{x}\in\mathcal{X}} q(x|\bar{c}(\bar{x}))p(\bar{x})\Big] \\
&\leq \sum_{x\in\mathcal{X}} p(x)\log p(x) - \sum_{x\in\mathcal{X}} p(x)\log\left[q(x|\bar{c}(x))p(x)\right] \\
&\leq \sum_{x\in\mathcal{X}} p(x)\log p(x) - \sum_{x\in\mathcal{X}} p(x)\log\left[(1-\rho L\epsilon)p(x)\right] \\
&\leq -\log(1-\rho L\epsilon)
\end{aligned} \tag{C.24}$$

Thus we can choose ϵ sufficiently small that $\text{KL}(p\|q_\theta) < \eta$. □

One concerning aspect of this proof, practically, is that it requires very large M to form the approximation q_θ. How well can we do with smaller M? Combining Theorem 4.2 of Zhang [297] with the above result, we can say that for any $\eta > 0$ there exists an $\epsilon > 0$ such that $\text{KL}(p\|q_{\theta^*})$ converges to a value less than η at a $1/M$ rate. Note also that our setup differs from the more common case where a mixture model is used for density estimation based on finite data, since we can sample as much as we want from p. We therefore do not analyze the mismatch between a target p and model q_θ that may be caused by finite data.

Next, we prove the second part of Proposition 3.4.3, showing that enzymatic mutagenesis can fail to approximate arbitrary targets p. The basic idea is that when using enzymatic mutagenesis, the probability of a particular sequence cannot get arbitrarily close to 1, and so the KL divergence between p and q_θ cannot get arbitrarily close to 0.

Part 2 of Proposition 3.4.3: *When using **enzymatic mutagenesis**: there exists some $p \in \mathcal{P}(\mathcal{X})$ and $\eta > 0$ such that for all M and θ, we have $\text{KL}(p\|q_\theta) > \eta$.*

Proof. Since $\tau > 0$, and the entries of S are all positive, we can see that we are limited in how much mass an enzymatic mutagenesis model can concentrate on just one sequence, i.e.

$$\sup_{\tau>0, c\in\mathcal{U}^L} \sup_{x\in\mathcal{X}} q(x|c,\tau) < 1. \tag{C.25}$$

Choose $p(x) = \delta_{x'}(x)$ for some sequence $x' \in \mathcal{X}$, and let q_θ be an enzymatic mutagenesis synthesis

model with M templates. Then,

$$\inf_{\theta} \textsc{kl}(p\|q_\theta) \geq -\log \sup_{\tau>0, c\in\mathcal{U}^L} \sup_{x\in\mathcal{X}} q(x|c,\tau) > 0. \tag{C.26}$$

□

C.4.6 Proof of Proposition 3.4.4

Proof. Let $\tilde{L}_k$ denote the subset of positions generated by template k, i.e. $\tilde{L}_1 = \{1,\dots,L_1\}$, $\tilde{L}_2 = \{L_1+1,\dots,L_1+L_2\},\dots$. Let $p(x_{\tilde{L}_k})$ denote the marginal of p over these positions. We have, since templates are drawn independently from each pool, $q_\theta(x) = \prod_{k=1}^K q_\theta(x_{\tilde{L}_k})$, and so

$$\begin{aligned}
\textsc{kl}(p\|q_\theta) &= \sum_x p(x)\log p(x) - \sum_{k=1}^K \sum_{x_{\tilde{L}_k}} p(x_{\tilde{L}_k}) \log q_\theta(x_{\tilde{L}_k}) \\
&= \sum_x p(x)\log p(x) - \sum_{k=1}^K \sum_{x_{\tilde{L}_k}} p(x_{\tilde{L}_k}) \log p(x_{\tilde{L}_k}) + \sum_{k=1}^K \textsc{kl}(p(x_{\tilde{L}_k})\|q_\theta(x_{\tilde{L}_k})) \\
&\geq \textsc{kl}(p\|\prod_{k=1}^K p(x_{\tilde{L}_k})).
\end{aligned} \tag{C.27}$$

There exists p for which $\textsc{kl}(p\|\prod_{k=1}^K p(x_{\tilde{L}_k})) > 0$, in particular any p for which there is correlation between templates, proving the result. □

C.5 Results details

C.5.1 Datasets and target models

DHFR

We used a dataset of 3,629 sequences in the DHFR family collected using jackhmmer[69] from the Uniref100 dataset[252], and available as an example dataset from https://github.com/debbiemarkslab/plmc/tree/master/example/protein/DHFR.a2m. The multiple sequence alignment has a width of $L = 171$ amino acids. We trained a Potts model using pseudolikelihood maximization as in Hopf et al. [110], using the plmc package with the default hyperparameters https://github.com/debbiemarkslab/plmc. Gaps in the alignment were treated as missing data (not as separate symbols), following the default settings of plmc. The trained Potts model was the target p. We sampled sequences from p using Gibbs sampling, drawing 100,000 samples using 10 parallel chains with a burn-in of 200 steps per chain.

For the analysis of unaligned sequences (Figures 3.3D and C.5), we used the training dataset of 3,629 evolutionary sequences, with gap symbols excluded and stop symbols appended. We refer to this dataset as "DHFR raw".

GFP

We constructed a dataset of 722 sequences in the GFP family using jackhmmer and UniprotKB (07/2021)[201], starting from the seed sequence `GFP_AEQVI` with F64L (a stabilized variant used by

Sarkisyan et al. [226]), with a threshold of 0.3 bit score per residue. We trained an ICA model with a MuE output [284], which is available as an example in the Pyro probabilistic programming language [23] at `https://pyro.ai/examples/mue_factor.html`. The ICA model is similar to a probabilistic PCA model, but uses a Laplace prior on the latent variable instead of a Gaussian; the MuE output uses the default profile HMM-based architecture described in Weinstein & Marks [284]. We used 2 latent dimensions in the ICA model, a latent sequence length of 237 in the MuE, and default priors. The model was trained with stochastic variational inference, with a learning rate of 0.005 and batch size of 5 over 70 epochs, annealing the prior KL divergence linearly over 35 epochs. Using 20% of the data as heldout validation, the model achieves a per residue perplexity on the training set of 3.1 and on the test set of 4.6.

We used the ICA-MuE model to construct a target distribution p. In particular, let ψ be the latent alignment variable of the MuE (the state variable of the Markov chain). We estimated the maximum *a posteriori* value of ψ for the stabilized wild-type GFP (`GFP_AEQVI` with F64L), and then sampled new sequences conditional on this value $\hat{\psi}_{\text{ref}}$ – note that this procedure is a very weak form of supervision, since the stabilized wild-type is known to be functional and produce fluorescence. To limit the diversity of the library relative to the training data, we sampled from the posterior predictive over the latent representation given the observed data, rather than the prior. Explicitly, let $p_{\text{MuE}}(x|\psi, \kappa)$ denote the distribution of the learned ICA-MuE model conditional on the latent alignment ψ and latent representation κ. Let $X'_1, \ldots, X'_{N'}$ denote the training data, and let $p(\kappa|X')$ denote the posterior over the latent representation of a datapoint X' (which can be approx-

imated by the encoder/guide network). The complete generative process p is then defined as

$$\begin{aligned} \kappa_i &\sim \frac{1}{N'} \sum_{i=1}^{N'} p(\kappa | X'_i) \\ X_i &\sim p_{\text{MuE}}(x | \hat{\psi}_{\text{ref}}, \kappa_i). \end{aligned} \tag{C.28}$$

An important feature of this model is that we are *not* sampling from the conditional distribution of κ given $\hat{\psi}_{\text{ref}}$, that is, we are not sampling sequences with similar latent alignments. Unlike autoregressive models, for example, MuE models allow variation in sequence length and latent alignment to be treated as independent of variation at conserved sites. Thus, although the sequences generated from p are all of the same length, the pattern of amino acids at conserved sites reflects the full diversity of the dataset. Finally, note that in the jackhmmmer constructed-dataset, the first residue (M) of the wild-type sequence `GFP_AEQVI` was not included in the profile HMM envelope, but the sequence-to-function predictor expects this position to be included; we therefore prepended an M to each generated sequence, for a total length of $L = 238$.

TCR

We examined a dataset of 22,004 TCRβ sequences measured in Ramien et al. [207], taken from CD8+ T cells from a single healthy control patient (number HC12 in the study) in the 3rd trimester of pregnancy. We trained a ICA-MuE model as described above (Section C.5.1), with 5 latent dimensions and a latent sequence length of 170. We used stochastic variational inference, with a learning rate of 0.01 and batch size of 5 over 2 epochs, annealing the prior KL divergence linearly over

1 epoch. Using 20% of the data as heldout validation, the model achieves a per residue perplexity of 2.39 on both the training and test datasets. We sampled from the model using the same strategy as in Section C.5.1. The reference sequence used to construct $\hat{\psi}_{\text{ref}}$ was a randomly selected sequence from the dataset described in Section C.5.6, which Tcellmatch predicted to bind the influenza epitope; as with the GFP example, conditioning on $\hat{\psi}_{\text{ref}}$ is a very weak form of supervision, learning from only a single functional example. In particular, the reference sequence was $MSNQVLCCVVLCLLGANTVDGGITQSPKYLFRKEGQNVTLSCEQNLN$ $HDAMYWYRQDPGQGLRLIYYSQIVNDFQKGDIAEGYSVSREKKESFP$ $LTVTSAQKNPTAFYLCASSIRSAYEQYFGPGTRLTVTEDLKNVFPPE$ $VAVFEPSE$. The generated sequences had length $L = 149$.

C.5.2 Synthesis model hyperparameters

In this section we describe the details of our stochastic synthesis models and optimization procedure. We used $K = 5$ pools, with L_k of approximately the same length for each $k \in \{1, \ldots, K\}$ (the last template was shortened as necessary since L is not always a multiple of 5). This yields templates of length 29 to 48 amino acids across all the datasets considered, which is consistent with typical oligosynthesis lengths of ~ 150 nucleotides. We used $A = 8$ for finite nucleotide mixtures; this value is realistic, as the company IDT, for example, currently offers four custom mixtures per oligo plus preset mixtures and single nucleotides [194]. We used $A = 24$ for finite codon mixtures, which is similar to typical trimer-based synthesis projects, which use the 20 amino acids plus a few custom mixtures [172].

We set the mutation matrix S based on the ePCR enzyme Mutazyme II, available as part of Agilent's GeneMorph II Random Mutagenesis Kit https://www.chem-agilent.com/pdf/strata/200550.pdf. In particular, we converted the reported mutational spectra (Table II) into a substitution matrix, under the assumption that the test sequences are 50% A-T base pairs and 50% G-C base pairs: for instance, the probability of a particular base pair mutating per round of mutagenesis is given as 1% overall (10 bases per kilobase), and 50.7% of mutations happen to A-T base pairs, so the probability of a particular A-T base pair mutating is $0.01 \cdot 0.507/0.5$. Proceeding in this way, we find

$$S = \begin{pmatrix} 0.990 & 0.006 & 0.005 & 0.003 \\ 0.006 & 0.990 & 0.003 & 0.005 \\ 0.003 & 0.001 & 0.991 & 0.001 \\ 0.001 & 0.003 & 0.001 & 0.991 \end{pmatrix} \tag{C.29}$$

where the columns and rows are each in the order A, T, G, C. We also computed a mutation matrix S based on the Taq error prone polymerase (also in Table II of the Gene Morph II Random Mutagenesis Kit manual), but preliminary experiments suggested worse performance than Mutazyme II at matching the DHFR Potts target distribution, so we did not pursue it further. We limit the total number of rounds of mutagenesis τ to be less than 10, since large numbers of mutagenesis rounds are rarely used in practice.

Note that since we have chosen $A \geq 4$, the set of allowed values of $\mathcal{U}$ for enzymatic mutagenesis (that is, for all τ) is a strict subset of the set of allowed values of $\mathcal{U}$ for finite nucleotide mixtures (that is, for all v); thus, synthesis models using enzymatic mutagenesis are strictly less expressive than

those using finite nucleotide mixtures. Meanwhile, $\mathcal{U}$ for finite nucleotide or codon mixtures is a strict subset of $\mathcal{U}$ for arbitrary codon mixtures, regardless of the choice of v; so synthesis models using finite mixtures are strictly less expressive than those using arbitrary codon mixtures.

C.5.3 Baseline synthesis model

As a baseline stochastic synthesis approach, we considered a method motivated by a common heuristic for producing diversified libraries, which is to simply perform error prone PCR on an initial set of sequences. In particular, the baseline approach we consider is to do MC synthesis plus enzymatic mutagenesis: sample initial protein sequences $X'_1, \ldots, X'_M \sim p$, inverse-translate the protein sequences into DNA (sampling uniformly among all codons for the same amino acid), synthesize the DNA individually, and then mutagenize in the laboratory using ePCR. The distribution of resulting sequences can be described using a stochastic synthesis model for which we do *not* optimize the parameters. In particular, let $K = 1$, and for $m \in \{1, \ldots, M\}$ and $j \in \{1, \ldots, L\}$, let $\chi'_{mj1b} = 1$ if the sampled codon for X'_{mj} has base b at the first position, and $\chi_{mj1b} = 0$ otherwise. Likewise for χ_{mj2b} and χ_{mj3b}. Then, we set $u_{1mj} = S^\tau \chi_{mj1} \otimes S^\tau \chi_{mj2} \otimes S^\tau \chi_{mj3}$. We use fixed assembly, setting $w = (\frac{1}{M}, \ldots, \frac{1}{M})$. Then, the complete synthesis model (Equation 3.1) describes the distribution of sequences produced by the baseline approach.

Note that the baseline is effectively a kernel density estimate of p. It is thus unsurprising that the baseline underperforms relative to variational synthesis, since kernel density estimates typically underperform compared to mixture models.

Practically, we use S corresponding to a Mutazyme II enzyme (Section C.5.2) and set $\tau = 5$

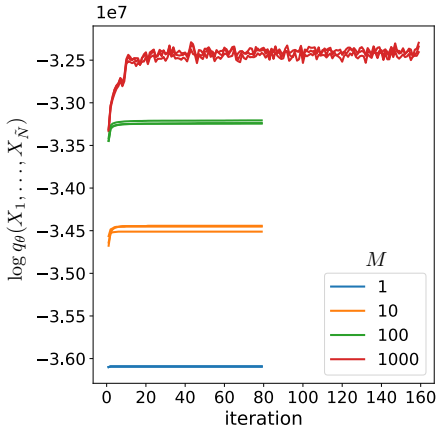


Figure C.1: Illustrative example of training curves for a stochastic synthesis model (enzymatic mutagenesis with fixed assembly) with different values of M. For each value of M, training is repeated with three initial seeds. The models are each trained on samples from the DHFR Potts model, as described in Section C.5.4.

as a typical value for proteins of the length considered here[290]. The samples from p used as initial sequences, $X'_1, \ldots, X'_M$, are subsampled from the same training dataset of 100,000 sequences used for variational synthesis (Section C.5.4). We examined the performance of the method averaged over 3 independent sets of initial sequences.

C.5.4 Optimization and perplexity evaluation

DHFR Potts, GFP, TCR To optimize synthesis models, we drew $\tilde{N} = 100,000$ samples from each target distribution p and applied stochastic EM, as described in Section C.2.2. We chose batch sizes to be as large as possible without running out of memory. In particular, we used batch sizes of 100,000 (the full dataset) with $M = 1$, $M = 10$ and $M = 100$, and batch sizes of 10,000 for $M = 1000$. We trained for 80 epochs with $M = 1$, $M = 10$ and $M = 100$, and 16 epochs for $M = 1000$. Training took 2-5 minutes for each target-synthesis pair using a Tesla V100 GPU.

Example training curves are shown in Figure C.1.

Each synthesis model was trained on the same set of $\tilde{N} = 100,000$ samples from each target distribution, and evaluated based on the average per residue perplexity on the training dataset, $\exp(-\frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} \frac{1}{\ell(X_i)} \log q_\theta(X_i))$, where $\ell(X_i)$ is the length of the sequence X_i. Note that we do not perform heldout evaluation, as our goal is to see how well each synthesis model can match a target library of size 100,000; overfitting the synthesis model is not a concern, and may even help downstream performance, as described in Section C.2.3. We initialized each optimization from three random seeds, and chose the result with the lowest perplexity.

DHFR raw For the DHFR raw dataset, we handle variable length sequences as described in Section C.2.4, and optimized each synthesis model using EM with batch size of 3,629 (the full dataset), for 100 epochs. We set L to be the maximum length of sequences in the dataset including the stop codon, 170. We evaluated using mean per residue perplexity on the full dataset. We initialized each model from three random seeds, and chose the result with the lowest perplexity.

C.5.5 BEAR two-sample test

We use the vanilla version of the BEAR two-sample test proposed in Section 5 of Amin et al. [12] to compare the target and the synthesis distributions. The test computes the Bayes factor comparing the hypothesis that two datasets $\{X_1, \ldots, X_{\tilde{N}}\}$ and $\{X'_1, \ldots, X'_{\tilde{N}'}\}$ come from the same underlying distribution versus different distributions. It uses $p_{\text{BEAR}}(X_1, \ldots, X_{\tilde{N}} | \alpha, \lambda)$, the probability of the dataset under a Bayesian Markov model with Dirichlet concentration parameter α and lag λ. In

particular, the Bayes factor is

$$\text{BF} = \frac{p_{\text{BEAR}}(X_1, \ldots, X_{\tilde{N}}, X'_1, \ldots, X'_{\tilde{N}'})}{p_{\text{BEAR}}(X_1, \ldots, X_{\tilde{N}}) p_{\text{BEAR}}(X'_1, \ldots, X'_{\tilde{N}'})} \tag{C.30}$$

where

$$p_{\text{BEAR}}(X_1, \ldots, X_{\tilde{N}}) = \frac{1}{\Lambda} \sum_{\lambda=1}^{\Lambda} p_{\text{BEAR}}(X_1, \ldots, X_{\tilde{N}} | \alpha, \lambda). \tag{C.31}$$

We used the training dataset of 100,000 samples from p as the first dataset in the two-sample test, and 100,000 independent samples drawn from the optimized synthesis model $q_{\hat{\theta}_{\tilde{N}}}$ as the second dataset. (In the case of DHFR raw, we used the 3,629 sequences as the target sample.) Note that the goal here is to understand whether the particular set of $\tilde{N}$ samples from p used for training look like a plausible set of samples from $q_{\hat{\theta}_{\tilde{N}}}$, following the logic of Section C.2.3, so we do not resample from p to compute the test. We use $\alpha = 0.5$ and $\Lambda = 8$; we found that in general the posterior over lags concentrated at values of λ below 8, suggesting the test has sufficiently high resolution. Computing the test took about 5-10 minutes for each target-synthesis pair, with 20 cores on an Intel Xeon E5 v3 CPU.

C.5.6 Sequence-to-function predictors

GFP: TAPE

We computed TAPE predictions of GFP fluorescence using the interface in the FLEXS package[239]. Sequences with internal stop codons were assigned the minimum log fluorescence in the Sarkisyan

et al.[226] dataset, 1.2. Variants with predicted log fluorescence above 3 were classified as hits, in line with the analysis of Sarkisyan et al.[226] who classify variants below 3 as dark.

TCR: Tcellmatch

We used Tcellmatch, trained on the same single-cell TCR sequencing data as in the original paper[3], with the suggested model architecture (1x1 convolutional embeddings based on BLOSUM50 and biGRU layers). We used the mean squared logarithmic error to evaluate the model's ability to predict MHC multimer binding counts. We trained the Tcellmatch model only on TCRβ sequences, since the target p was trained only on TCRβ sequences. The Tcellmatch model uses only the CDR3 region to make predictions. In general, techniques for identifying the CDR3 region in TCRs rely on nucleotide-level information, which is unavailable for generated amino acid sequences. However, we constructed the target p by conditioning on a latent alignment, which in turn is based on a reference sequence with nucleotide-level information (Section C.5.1). We thus use the positions corresponding to the CDR3 in the reference sequence (109:122, as annotated by the 10x pipeline) to define the CDR3 for each sampled sequence from p and q_θ. Although the Tcellmatch model can be used to predict many different antigens, we focused on predictions of the $GILGFVFTL$ influenza antigen, since the model had the most accurate predictions for this particular antigen (according to the R^2 metric used by Fischer et al.[77], in particular $R^2 = 0.70$). We conditioned on a single donor (donor 1) when making predictions with Tcellmatch. Sequences with internal stop codons were assigned zero counts. Variants with predicted counts above 10 were classified as hits, in line with the analysis of Fischer et al.[77].

Estimating Hit Rates

Given a dataset of indicators for whether or not each of 100,000 samples from q_θ was a hit or not, i.e. $\{f(X_1), \ldots, f(X_N)\}$ where $f : \mathcal{X} \mapsto \{0, 1\}$, we estimated the overall hit rate using a Beta$(0.5, 0.5)$ prior (Jeffreys prior). We report the standard deviation of the posterior in Figure 3.4C and G.

Estimating the Number of Unique Hits

Based on the hit rate (Section C.5.6), we can estimate the total number of hits for libraries of any size. However, we are also interested in the total number of unique hits, since discovering identical sequences is not as useful as discovering diverse sequences. Evaluating predictors on very large numbers of samples, though, can be impractical since predictors (especially TAPE) can be computationally expensive. Instead, we used a Good-Toulmin estimation strategy: we examined the hits from a sample of 100,000 sequences from q_θ and then extrapolated to estimate the number of unique hits in a library of 1,000,000 sequences. We used the smoothed Good-Toulmin estimator proposed by Orlitsky et al. [191], with the recommended Binomial model. Note that the estimator is considered trustworthy for datasets up to a factor of $\log N$ larger than the initial dataset; since $\log(10^5) = 11.5 \geq 10$, it is applicable here. We estimate the variance of the estimate under resampling using the jackknife, which can be efficiently computed for the smoothed Good-Toulmin estimator[71].

C.5.7 Error bars

In this section we summarize the calculation of the error bars in Figures 3.3 and 3.4. For the baseline model, we show the estimated standard deviation across independent samples of the initial M sequences from p (that is, the initial sequences that are mutagenized by ePCR). We use three independent samples for each value of M. For perplexity plots (Figures 3.3ACD and 3.4AE) we do not include any error estimates for non-baseline models, since we have exactly computed the total perplexity across the training dataset, and we are only interested in the match between the synthesis model and the training dataset, not in the synthesis model's generalization performance (as explained in Section C.2.3). Bayes factors are themselves measurements of statistical significance, so we do not include any error bars for non-baseline models in Figures 3.3B and 3.4BF. For plots of hit rate (Figure 3.4CG), error bars show the posterior standard deviation of the hit rate under a Beta$(0.5, 0.5)$ prior (the Jeffreys prior) (Section C.5.6). For plots of estimated unique hits (Figures 3.4DH), error bars show the jackknife estimate of the standard deviation (Section C.5.6) For the baseline model, in plots of both hit rate and unique hits, error bars include the variance across different initial sequences from p, and are computed using the law of total variance.

C.5.8 Additional results

DHFR

We further examined the match between stochastic synthesis models and the target DHFR Potts model, examining the difference in moments of each distribution. In particular, we looked at the dif-

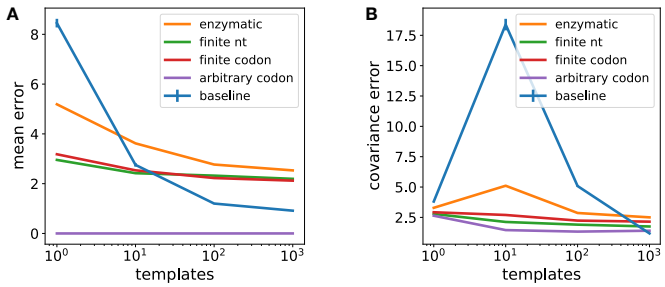


Figure C.2: (A) Difference in mean between synthesis and target models, for various stochastic synthesis models with fixed assembly applied to the DHFR Potts target. Mathematically, $\|\mathbb{E}_{q_\theta}[X] - \mathbb{E}_p[X]\|_2$ where X is represented as a one-hot encoding and $\|\cdot\|_2$ is the Euclidean distance. (B) Difference in position-wise covariance matrices between synthesis and target models. Mathematically, let $\tilde{C}^{(p)}_{j,j',d,d'} := \mathrm{Cov}_p(X_{j,d}, X_{j',d'})$ denote the covariance under p between the dth amino acid at position j and the d'th amino acid at position j'. The magnitude of the covariance between positions j and j' can be measured as $\mathcal{C}^{(p)} := \|\tilde{C}^{(p)}_{j,j'}\|_2$. Then we plot the position-wise covariance error $\|\mathcal{C}^{(p)} - \mathcal{C}^{(q_\theta)}\|_2$. In both plots, error bars for the baseline model are the standard deviation over initial sequences (Section C.5.7).

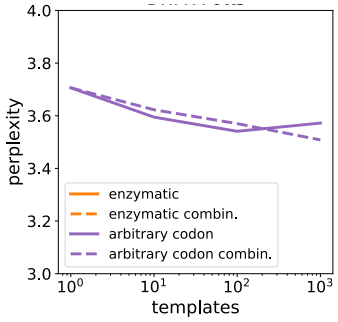


Figure C.3: Zoom in of Figure 3.3C.

ference in the mean sequence produced by the synthesis and target distributions, and the difference in covariance between positions of the sequences produced by the synthesis and target distributions (Figure C.2). Comparing different variational synthesis models, we see improved perplexity (Figure 3.3A) corresponds well with lower moment error (Figure C.2). Interestingly, the baseline synthesis method (Section C.5.3) yields comparatively low moment error for large M despite comparatively poor perplexity.

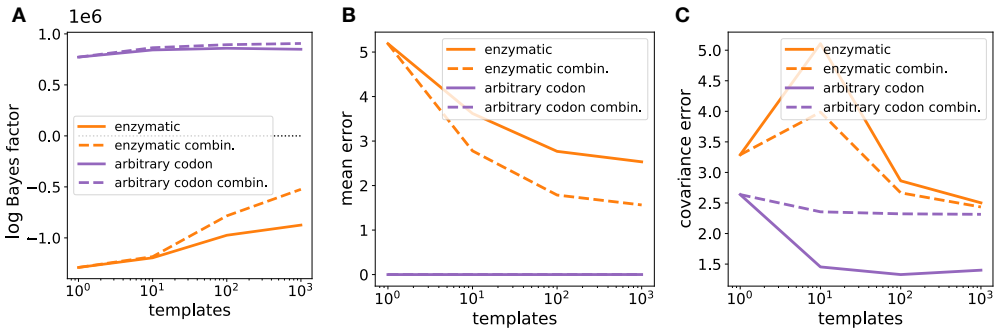


Figure C.4: Comparing fixed versus combinatorial assembly for the DHFR Potts target. (The perplexity comparison can be found in Figure 3.3C and C.3.) (A) Two-sample test Bayes factor. (B) Difference in mean between synthesis and target models, as defined in Figure C.2. (C) Difference in position-wise covariance matrices between synthesis and target models, as defined in Figure C.2.

We further examined the difference in performance between combinatorial and fixed assembly models. For enzymatic mutagenesis, switching from fixed to combinatorial assembly improves the two-sample test Bayes factor (Figure C.4A), mean (Figure C.4B) and covariance (Figure C.4C). For arbitrary codon synthesis, switching from fixed to combinatorial assembly slightly improves the Bayes factor (Figure C.4A), has no effect on the mean (as we expect mathematically and see in Figure C.4B), but substantially worsens the covariance (Figure C.4C). These results illustrate how the advantages of using fixed versus combinatorial assembly vary depending on the codon diversification technology.

We further examined the performance of different stochastic synthesis models applied to the DHFR raw dataset of unaligned evolutionary sequences. Applying the two-sample test, we find that using large numbers of templates with any codon diversification technology is better than using small numbers of templates with a very expressive codon diversification technology (Figure C.5), in line with the perplexity results (Figure 3.3D). We also see that variational synthesis is capable of

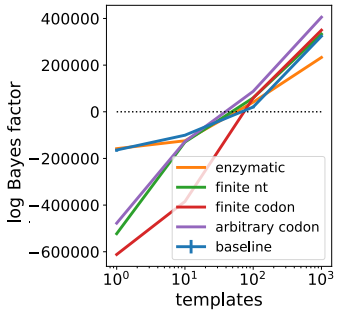


Figure C.5: Two-sample test Bayes factor for synthesis models with fixed assembly applied to the DHFR raw dataset. For perplexity comparison, see Figure 3.3D.

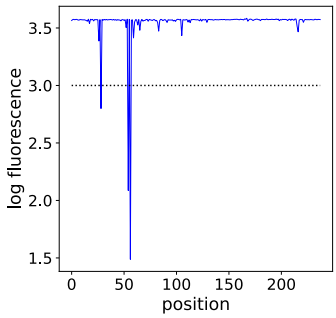


Figure C.6: Predicted mutational effects of substituting each position of the stabilized wild-type GFP sequence with an alanine (i.e. an *in silico* alanine scan). Dotted line shows the threshold for classifying a variant as functional.

matching the target closely enough to pass the two-sample test, but so is the baseline method in this case.

GFP

We examined the difference in moments between the target GFP distribution and the stochastic synthesis models. The results (Figure C.7) are qualitatively similar to those described for DHFR (Section C.5.8 and Figure C.2), with the baseline model performing better than its perplexity would suggest.

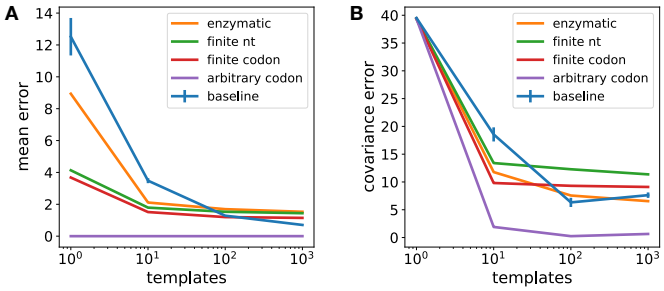


Figure C.7: (A) Difference in mean between GFP synthesis and target models, as defined in the caption of Figure C.2. (B) Difference in position-wise covariance matrices between GFP synthesis and target models, as defined in the caption of Figure C.2.

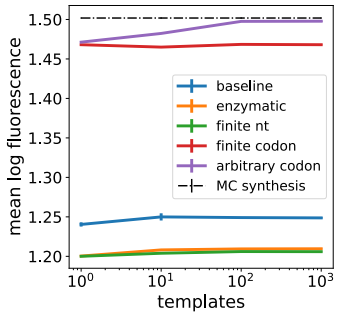


Figure C.8: Average log predicted fluorescence of samples from various stochastic synthesis models and from the GFP target p itself (MC synthesis). Error bars are estimates of the standard deviation of the mean (for the baseline model, this includes variance across different initial sequences, as described in Section C.5.7; for the rest of the models, it is just the standard error, and negligible in these plots).

We examined the difference in average log fluorescence between samples from various stochastic synthesis models, as compared to exact samples from the target (that is, as compared to the average log fluorescence under MC synthesis) (Figure C.8). Interestingly, we find that while using finite codon mixtures with $M = 1$ yields relatively low hit rates compared to arbitrary codon mixtures with $M = 1$ (Figure 3.4G), it yields nearly equivalent average log fluorescence (Figure C.8).

We examined the difference in performance between combinatorial and fixed assembly methods applied to the GFP target distribution. On statistical measures of the difference between the syn-

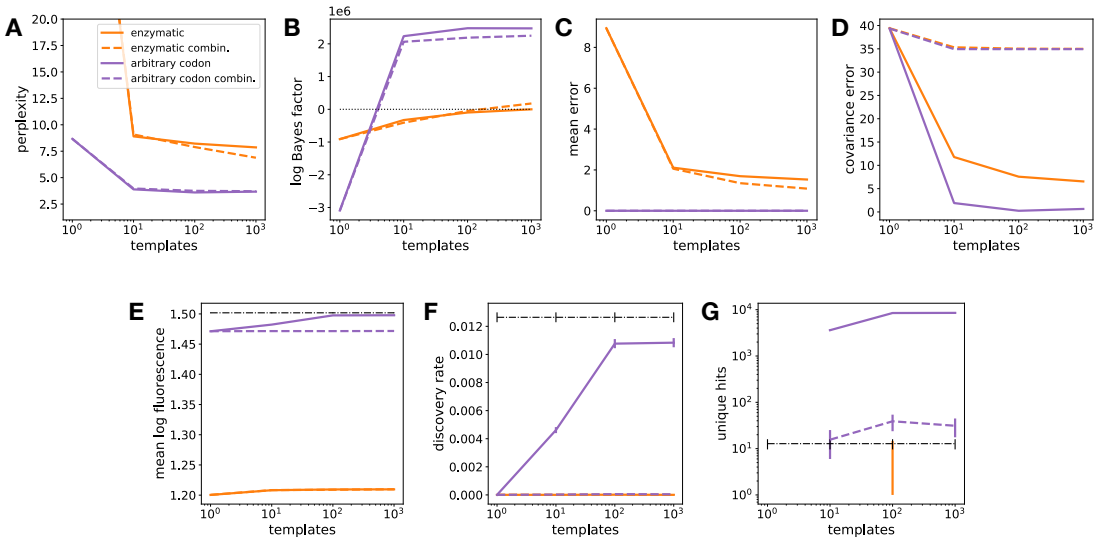


Figure C.9: Fixed versus combinatorial stochastic synthesis applied to the GFP target distribution. (A) perplexity, (B) two-sample test Bayes factor, (C) mean error (as defined in Figure C.2), (D) covariance error (as defined in Figure C.2), (E) average log fluorescence, (F) hit rate and (G) number of unique hits with $N_1 = 10^6$ and $N_0 = 10^3$. Error bars are as described in Section C.5.7.

thesis and target distribution (Figure C.9ABCD), we find broadly similar effects to those observed for DHFR Potts: for instance, we see moderate improvements in perplexity for enzymatic mutagenesis at large M when switching from fixed to combinatorial assembly, but little effect for arbitrary codon mixtures, and substantially worse covariance for arbitrary codon mixtures. On measures of function, using combinatorial assembly leads to dramatically worse performance (Figure C.9EFG): using arbitrary codon mixtures with combinatorial instead of fixed assembly drops the number of unique hits by three orders of magnitude. This result suggests that passing the BEAR two-sample test with large Bayes factors is not enough to ensure high hit rates when using combinatorial assembly; one should also inspect the covariance error.

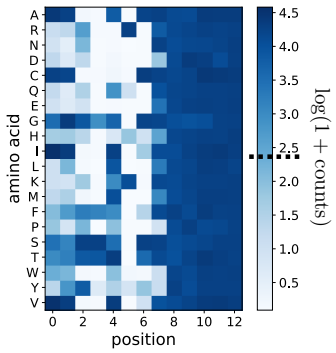


Figure C.10: Predicted binding effects of substituting each position of a natural CDR3 sequence ($CASSIRSAYEQYF$) with each of 20 amino acids (*in silico* deep mutational scan). The threshold for functionality (10 counts) is marked by a dotted line in the colorbar.

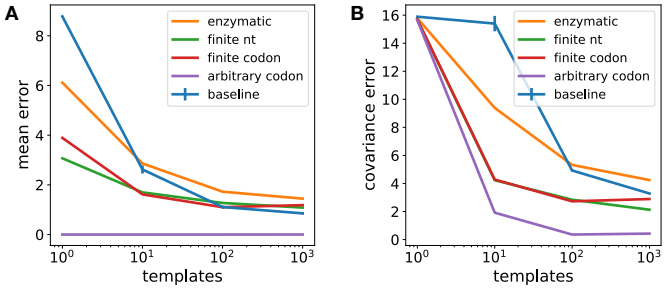


Figure C.11: (A) Difference in mean between TCR synthesis and target models, as defined in the caption of Figure C.2. (B) Difference in position-wise covariance matrices between TCR synthesis and target models, as defined in the caption of Figure C.2.

TCR

We examined the difference in moments between the target TCR distribution and the stochastic synthesis models. The results (Figure C.11) are qualitatively similar to those described for DHFR (Section C.5.8 and Figure C.2), with the baseline model performing better than its perplexity would suggest.

We examined the difference in average binding counts between samples from various stochastic synthesis models, as compared to exact samples from the target TCR model (that is, as compared to

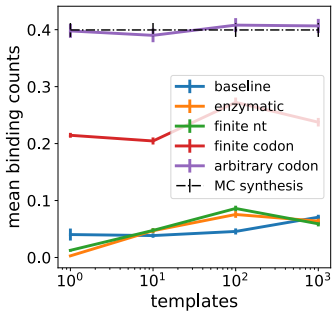


Figure C.12: Average predicted binding counts of samples from various stochastic synthesis models and from p itself (MC synthesis). Error bars are estimates of the standard deviation of the mean (for the baseline model, this includes variance across different initial sequences; for the rest of the models, it is just the standard error).

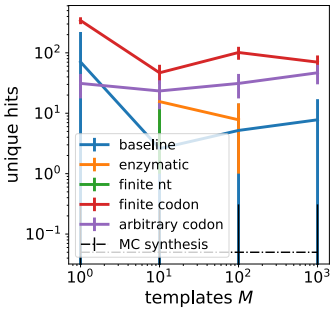


Figure C.13: Same as Figure 3.4H, but for the EBV epitope $RAKFKQLL$ instead of the influenza epitope.

MC synthesis) (Figure C.12). Unlike for GFP, we find that the average value of the assay output is roughly proportional to the hit rate.

We examined additional viral epitopes, besides the influenza epitope, for which decent Tcellmatch predictions were available. The second highest quality Tcellmatch predictor ($R^2 = 0.43$) was for an Epstein-Barr virus (EBV) epitope, $RAKFKQLL$. MC synthesis with $N_0 = 10^3$ generates just 0.05 hits on average across independent libraries, while variational synthesis with $N_1 = 10^6$, using arbitrary codon mixtures and $M = 10$, generates an expected 30 unique hits (Figure C.13). Here, variational synthesis makes the difference between likely failure and likely

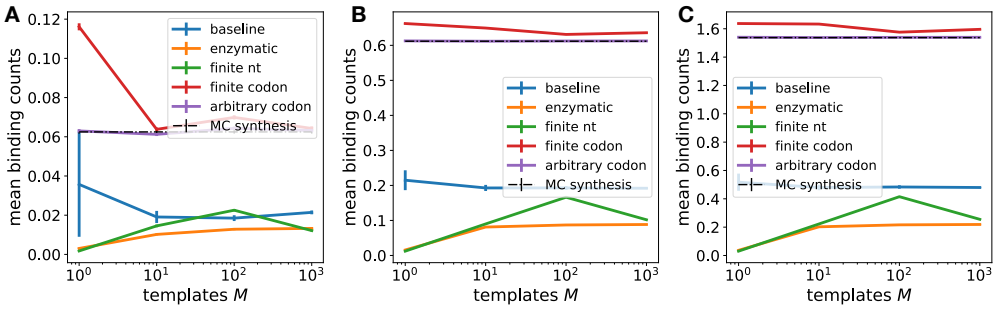


Figure C.14: Same as Figure C.12, but for the EBV epitope $RAKFKQLL$ (A), the EBV epitope $RLRAEAQVK$ (B) and the CMV epitope $KLGGALQAK$ (C).

success. We also examined two viral epitopes for which the target TCR model had an estimated hit rate of zero (based on a sample of 10^5 sequences from the model): a cytomegalovirus epitope ($KLGGALQAK$, Tcellmatch $R^2 = 0.40$) and another EBV epitope ($RLRAEAQVK$, Tcellmatch $R^2 = 0.36$). Note that a hit rate of close to zero is unsurprising, given that the Tcellmatch predictor has low accuracy, and that the individual patient which the TCR model was trained on may not have TCRs that bind these epitopes. For these two epitopes, we found that variational synthesis was still able to closely match the average binding counts under the target TCR model (Figure C.14).

D

Supplementary Material for Chapter 4

D.1 Evolutionary dynamics models

Application of the Sella & Hirsh [230] model (Eqn. 4.1) in JFPMs rests on a number of assumptions; we briefly the most relevant here.

When applying Eqn. 4.1 to amino acid sequences, as is typical for fitness estimation models, we ignore biases that come from the genetic code, which can modify the steady state probability of

amino acids (in the absence of fitness effects) away from a uniform distribution. This is justified practically by the small effect sizes: if at steady state an amino acid has probability $1/64$ instead of $1/20$, the total difference in log probability is $\log(1/20) - \log(1/64) \approx 1$, which is small compared to (for instance) the log probability differences relevant for disease risk prediction with fitness models, which are ≈ 10 (Frazer et al.[80], Extended data Fig. 3). Moreover, this bias only contributes an overall shift in amino acid probabilities, independent of position, and so does not change our main theoretical results. We ignore biases caused by asymmetric mutation rates for analogous reasons (though note they are often included in PMs in practice)[230].

The constant β depends on the effective population size, as well as the underlying population genetics model (Moran or Wright) and organismal ploidy (Sella & Hirsh[230], Table 1). Following standard practice, we treat β as fixed for simplicity, though in reality it may vary over time and across lineages. Taking into account these possible changes clearly would not contradict our main theoretical result, that fitness and phylogeny are non-identifiable.

D.2 Proofs

D.2.1 Proof of Proposition 4.2.3

N.b. this result is known in the literature (Ho & Ané[104], Eqn. 1) but we are unaware of a proof, so we provide one here for completeness.

Proof. For notational convenience, we will work with a standardized OUT, with $\mu = 0$ and $\sigma = 1$. The final result can be obtained by translating and scaling the distribution of leaves. The transition

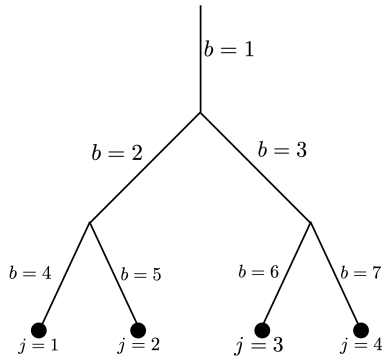


Figure D.1: Tree labeling for the proof of Proposition 4.2.3

distribution from point x' at time t' to point X at time t under the Ornstein-Uhlenbeck (OU) process is

$$X \sim \text{Normal}\left(x' e^{-\frac{1}{2}(t-t')}, 1 - e^{-(t-t')}\right). \tag{D.1}$$

This distribution can be reparameterized in location-scale form as

$$\epsilon \sim \text{Normal}(0, 1)$$
$$X = x' e^{-\frac{1}{2}(t-t')} + \sqrt{1 - e^{-(t-t')}}\epsilon.$$

As $t \to \infty$ we reach the stationary distribution Normal$(0, 1)$. Let $b \in \{1, ..., \mathbf{B}\}$ index the branches of the tree, let λ_b be the length of branch b, and let $j \in \{1, ..., N\}$ index the leaves (observed species or sequences); see Fig. D.1. We have assumed that the most recent common ancestor of the observed sequences was sampled from p^∞; this can be represented by adding a single branch length (indexed $b = 1$) to the root with length $\lambda_1 = \infty$. Let ϵ_b be the noise describing the OU diffusion over each branch. Let $\xi_{j,b}$ be the total time from leaf j to the nearest vertex

on branch b, so long as branch b is on the path from leaf j to the root; otherwise, set $\xi_{j,b} = \infty$. For instance, in the diagram in Figure D.1, we have $\xi_{1,4} = 0, \xi_{1,2} = \lambda_4, \xi_{1,1} = \lambda_4 + \lambda_2$, and $\xi_{1,5} = \xi_{1,6} = \xi_{1,7} = \xi_{1,3} = \infty$. We can now write the leaf position as

$$X_j = \sum_b e^{-\frac{1}{2}\xi_{j,b}} \sqrt{1 - e^{-\lambda_b}} \epsilon_b. \tag{D.2}$$

Define the matrix

$$M_{j,b} = e^{-\frac{1}{2}\xi_{j,b}} \sqrt{1 - e^{-\lambda_b}}, \tag{D.3}$$

such that $X_j = \sum_b M_{j,b}\epsilon_b$. We can now describe the complete leaf distribution as

$$\vec{\epsilon} \sim \text{MultivariateNormal}(0, I_{\mathbf{B}})$$

$$X_{1:N} = M \cdot \vec{\epsilon},$$

where $I_{\mathbf{B}}$ is the $\mathbf{B}$-dimensional identity matrix. Thus, according to the location-scale representation of the multivariate normal,

$$X_{1:N} \sim \text{MultivariateNormal}(0, MM^\top). \tag{D.4}$$

We can simplify the covariance matrix $\Sigma := MM^\top$. First

$$\Sigma_{j,j'} = \sum_b M_{j,b} M_{j',b} = \sum_b e^{-\frac{1}{2}(\xi_{j,b} + \xi_{j',b})} (1 - e^{-\lambda_b}).$$

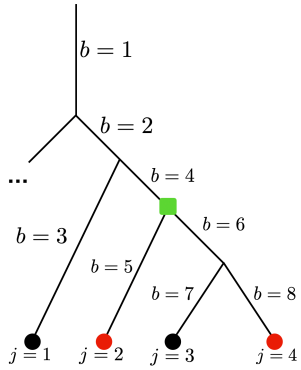


Figure D.2: In red are the leaves considered in the examples in the proof of Proposition 4.2.3; in green is their most recent common ancestor.

Before introducing the notation required to derive the general result, it's helpful to get a sense of how the derivation works; in the example tree (Figure D.1),

$$\begin{aligned}\Sigma_{1,2} &= e^{-\frac{1}{2}(\lambda_4+\lambda_5)}(1-e^{-\lambda_2}) + e^{-\frac{1}{2}(\lambda_4+\lambda_5+2\lambda_2)}(1-e^{-\lambda_1}) \\ &= e^{-\frac{1}{2}(\lambda_4+\lambda_5)} + (-e^{-\frac{1}{2}(\lambda_4+\lambda_5+2\lambda_2)} + e^{-\frac{1}{2}(\lambda_4+\lambda_5+2\lambda_2)}) - e^{-\frac{1}{2}(\lambda_4+\lambda_5+2\lambda_2-2\lambda_1)} \\ &= e^{-\frac{1}{2}(\lambda_4+\lambda_5)}.\end{aligned}$$

The sum over b telescopes, leaving only the initial term, which corresponds to the total time between leaf node 1 and leaf node 2. To construct the general result, define $\tilde{b}_{j,j'}$ as the branch whose later node is the most recent common ancestor of leaves j and j'. In the example in Figure D.2, $\tilde{b}_{2,4} = 4$. Let R be an ordered list of branches from $\tilde{b}_{j,j'}$ to $b = 1$, the earliest branch. In the example in Figure D.2, $R = [4, 2, 1]$. Finally, let $t_{jj'}$ be the length of the shortest path from leaf j to leaf j', the time from the most recent common ancestor to j plus the time to j'. In the example in Figure D.2,

$t_{2,4} = \lambda_5 + \lambda_6 + \lambda_8$. We now have

$$\begin{aligned}
\Sigma_{jj'} &= \sum_{b=1}^{B} e^{-\frac{1}{2}(\xi_{j,b}+\xi_{j',b})}(1 - e^{-\lambda_b}) \\
&= \sum_{b\in R} e^{-\frac{1}{2}(\xi_{j,b}+\xi_{j',b})}(1 - e^{-\lambda_b}) \\
&= e^{-\frac{1}{2}t_{jj'}} - e^{-\frac{1}{2}(t_{jj'}+2\lambda_{\tilde{b}_{j,j'}})} + \sum_{k=2}^{|R|} e^{-\frac{1}{2}(t_{jj'}+2\sum_{k'=1}^{k-1}\lambda_{R_{k'}})}(1 - e^{-\lambda_{R_k}}).
\end{aligned}$$

Breaking down the telescoping sum, and using the fact that the final element of R is $t_1 = \infty$,

$$= e^{-\frac{1}{2}t_{jj'}} - e^{-\frac{1}{2}(t_{jj'}+2\sum_{k'=1}^{|R|}\lambda_{R_{k'}})} = e^{-\frac{1}{2}t_{jj'}}.$$

So we have the simple result that the covariance matrix depends just on the divergence times between leaves,

$$\Sigma_{jj'} = e^{-\frac{1}{2}t_{jj'}}. \tag{D.5}$$

Translating the distribution Eqn. D.4 by μ and scaling by σ yields the result. □

D.2.2 Proof of Theorem 4.3.3

Before proving the result, we briefly clarify a definition in the statement of the theorem:

Definition D.2.1 (Exchangeable in leaves)**.** *Let* $\mathbf{H}$ *be a tree with countably infinite leaves and let* $\mathbf{H}_\pi$ *be a permutation of a phylogeny in its leaves, i.e. the same tree* $\mathbf{H}$ *with the leaves observed in a different order, according to a permutation* π*. A distribution over phylogenies is exchangeable in its leaves if*

$p(\mathbf{H}) = p(\mathbf{H}_\pi)$ *for any permutation* π.

Proof. Outline: First, using the results from Sarkar[225]*, we construct an embedding for each tree into the hyperbolic plane, being careful that the embedding preserves exchangeability. Second, we apply de Finetti's Theorem to obtain the conditionally independent representation of the joint distribution of* $Z_1, Z_2, \ldots$ *Third, we use the distortion bound from Sarkar*[225] *to bound the Wasserstein distance between* $p(\nu)$ *and* $p(\tilde{\nu})$.

First we describe the Sarkar[225] $(1+\epsilon)$ distortion embedding algorithm setup. Vertices in phylogenetic trees have maximum degree three, and, by assumption, the minimum edge length in a tree $\mathbf{H}$ is greater than $\eta > 0$ with probability one. For any $\epsilon' > 0$, choose a $\rho < \pi/3$ and a scale factor

$$\lambda > \left(\frac{1+\epsilon'}{\epsilon'}\right)\frac{2k}{\eta}\log\tan\frac{\rho}{2}, \tag{D.6}$$

where k is the Gaussian curvature of the hyperbolic plane $\mathbb{H}$ (for most hyperbolic geometry models, and in particular the Lorentz manifold, $k = -1$). Then, let $h_1(\mathbf{H}), h_2(\mathbf{H}), \ldots$ be the position of the leaves in the embedding of $\mathbf{H}$ produced by the $(1+\epsilon)$ distortion embedding algorithm in Sarkar[225], using edge scale factor λ, and ρ separated cones with cone angle $2\pi/3 - 2\rho$. Taking the last line of the proof of Theorem 6 in Sarkar[225], we are guaranteed that even for a countably infinite number of leaves,

$$\begin{aligned}\max_{i,i'}\frac{\lambda t_{ii'}(\mathbf{H})}{\tilde{d}(h_i(\mathbf{H}), h_{i'}(\mathbf{H}))} &\leq 1+\epsilon' \\ \max_{i,i'}\frac{\tilde{d}(h_i(\mathbf{H}), h_{i'}(\mathbf{H}))}{\lambda t_{ii'}(\mathbf{H})} &= 1,\end{aligned} \tag{D.7}$$

where $i, i' \in \mathbb{N} := \{1, 2, \ldots\}$, and $\tilde{d}(\cdot, \cdot)$ is the hyperbolic distance function.

Next we will modify the embedding function h to ensure that the distribution of embedded leaves is exchangeable. Let $[\mathbf{H}]$ be the set of phylogenetic trees that are equivalent to $\mathbf{H}$ up to re-ordering of the vertices. For each equivalence class $[\mathbf{H}]$ we choose one ordering of the vertices to be the canonical tree $\hat{\mathbf{H}}([\mathbf{H}])$, and for any tree $\mathbf{H}$ let $\pi^c(\mathbf{H})$ be the leaf permutation such that the reordered tree $\mathbf{H}_{\pi^c(\mathbf{H})} = \hat{\mathbf{H}}([\mathbf{H}])$. Now define the modified leaf embedding function $h'(\mathbf{H}) := h_{\pi(\mathbf{H})}(\mathbf{H}_{\pi^c(\mathbf{H})})$ where $\pi(\mathbf{H})$ is the inverse permutation of $\pi^c(\mathbf{H})$. Since by assumption the prior $p(\mathbf{H})$ on the phylogenetic tree is exchangeable, we can rewrite $p(\mathbf{H})$ using the induced distribution over equivalence classes $p([\mathbf{H}])$ as

$$\begin{aligned} [\mathbf{H}] &\sim p([\mathbf{H}]) \\ \pi &\sim \text{Permutation} \\ \mathbf{H} &:= \hat{\mathbf{H}}([\mathbf{H}])_\pi, \end{aligned}$$

where Permutation is the uniform distribution over all permutations of $\mathbb{N} := \{1, 2, \ldots\}$. We now define the distribution over leaf embeddings as

$$\begin{aligned} \mathbf{H} &\sim p(\mathbf{H}) \\ Z_{1:\infty} &:= h'_{1:\infty}(\mathbf{H}), \end{aligned} \tag{D.8}$$

which we can rewrite as

$$
\begin{aligned}
[\mathbf{H}] &\sim p([\mathbf{H}]) \\
\pi &\sim \text{Permutation} \\
Z_{1:\infty} &:= h_\pi(\hat{\mathbf{H}}([\mathbf{H}])).
\end{aligned}
$$

The distribution $p(Z_1, Z_2, \ldots)$ is therefore exchangeable. Applying de Finetti's Theorem [133] we have a.s.

$$
\begin{aligned}
G &\sim \mathcal{G} \\
Z_i &\overset{iid}{\sim} G \text{ for } i \in \{1, 2, \ldots\}
\end{aligned}
\tag{D.9}
$$

where G is a random measure distributed according to a prior $\mathcal{G}$. Moreover, the embedding distortion bounds (Eqn. D.7) are preserved for each $\mathbf{H}$, since

$$
\begin{aligned}
1 + \epsilon \geq \max_{i,i'} \frac{\lambda t_{ii'}(\hat{\mathbf{H}}([\mathbf{H}]))}{\tilde{d}(h_i(\hat{\mathbf{H}}([\mathbf{H}])), h_{i'}(\hat{\mathbf{H}}([\mathbf{H}])))} &= \max_{i,i'} \frac{\lambda t_{\pi_i \pi_{i'}}(\mathbf{H}_{\pi^c(\mathbf{H})})}{\tilde{d}(h_{\pi_i}(\mathbf{H}_{\pi^c(\mathbf{H})}), h_{\pi_{i'}}(\mathbf{H}_{\pi^c(\mathbf{H})}))} \\
&= \max_{i,i'} \frac{\lambda t_{ii'}(\mathbf{H})}{\tilde{d}(h'_i(\mathbf{H}), h'_{i'}(\mathbf{H}))},
\end{aligned}
\tag{D.10}
$$

and by the same logic

$$
1 = \max_{i,i'} \frac{\tilde{d}(h_i(\hat{\mathbf{H}}([\mathbf{H}])), h_{i'}(\hat{\mathbf{H}}([\mathbf{H}])))}{\lambda t_{ii'}(\hat{\mathbf{H}}([\mathbf{H}]))} = \max_{i,i'} \frac{\tilde{d}(h'_i(\mathbf{H}), h'_{i'}(\mathbf{H}))}{\lambda t_{ii'}(\mathbf{H})}. \tag{D.11}
$$

We will now construct the Wasserstein bound. Define the joint distribution over ν and $\tilde{\nu}$,

$$\begin{aligned} \mathbf{H} &\sim p(\mathbf{H}) \\ \nu_{ii'}(\mathbf{H}) &:= \log(\frac{1}{2} t_{ii'}(\mathbf{H})) \\ \tilde{\nu}_{ii'}(\mathbf{H}) &:= \log(d(h'_i(\mathbf{H}), h'_{i'}(\mathbf{H}))) \end{aligned} \tag{D.12}$$

where we have chosen $d(\cdot,\cdot) = \frac{1}{2\lambda}\tilde{d}(\cdot,\cdot)$. Note that the marginal distribution of ν matches its definition in the statement of the theorem, and that, applying Eqn. D.8 and Eqn. D.9, the marginal distribution of $\tilde{\nu}$ also matches its definition. Using the fact that log is a monotonically increasing function, Eqn. D.10 gives

$$\log \sup_{i,i'} \frac{\exp(\nu_{ii'}(\mathbf{H}))}{\exp(\tilde{\nu}_{ii'}(\mathbf{H}))} \leq \log(1+\epsilon)$$

$$\sup_{i,i'} [\nu_{ii'}(\mathbf{H}) - \tilde{\nu}_{ii'}(\mathbf{H})] \leq \epsilon,$$

and similarly using the bound from Eqn. D.11, $\sup_{i,i'}[\tilde{\nu}_{i,i'}(\mathbf{H}) - \nu_{i,i'}(\mathbf{H})] \leq 0$. Thus, with probability 1 under $p(\mathbf{H})$,

$$\|\nu(\mathbf{H}) - \tilde{\nu}(\mathbf{H})\|_\infty = \sup_{i,i'} |\nu_{ii'}(\mathbf{H}) - \tilde{\nu}_{ii'}(\mathbf{H})| \leq \epsilon.$$

Recall that the Wasserstein distance between the distribution of two random variables ν and $\tilde{\nu}$ can be written as

$$\mathcal{W}_1(p(\nu), p(\tilde{\nu})) = \inf_{\gamma \in \mathcal{J}} \mathbb{E}_\gamma[\|\nu - \tilde{\nu}\|_\infty]$$

where $\mathcal{J}$ is the set of joint distributions with marginals corresponding to the distributions of ν and $\tilde{\nu}$ (Dudley[64], Chap. 11.8). Using the joint distribution in Eqn. D.12, the Wasserstein distance is bounded by

$$\mathcal{W}_1(p(\nu), p(\tilde{\nu})) \leq \mathbb{E}_{\mathbf{H}\sim p(\mathbf{H})}[\|\nu(\mathbf{H}) - \tilde{\nu}(\mathbf{H})\|_\infty] \leq \epsilon. \tag{D.13}$$

Now consider the case where $\mathcal{W}_1(p(\nu), p(\tilde{\nu})) = 0$. (N.b. in this case, we do not need to assume that the minimum time between nodes in $\mathbf{H}$ is greater than $\eta > 0$.) Since the Wasserstein metric is a metric on the space of probability distributions (Dudley[64] Lemma 11.8.3), $p(\nu) = p(\tilde{\nu})$ a.e.. Using the standard properties of Gaussian processes (Williams & Rasmussen[289], Chap. 2), the GPLVM model (Eqn. 4.5) can be written as

$$\begin{aligned} G &\sim \mathcal{G} \\ Z_i &\overset{iid}{\sim} G \text{ for } i \in \mathbb{N} \\ \tilde{\nu}_{ii'} &:= \log d(Z_i, Z_{i'}) \\ X_{1:\infty} &\sim \text{MultivariateNormal}(\mu, \Sigma_{ii'} := \sigma^2 \exp(-\exp \tilde{\nu}_{ii'})), \end{aligned} \tag{D.14}$$

which is equivalent to the OUT distribution,

$$\begin{aligned} \mathbf{H} &\sim p(\mathbf{H}) \\ \nu_{ii'} &:= \log[\frac{1}{2} t_{ii'}(\mathbf{H})] \\ X'_{1:\infty} &\sim \text{MultivariateNormal}(\mu, \Sigma_{i,i'} := \sigma^2 \exp(-\exp \nu_{i,i'})). \end{aligned} \tag{D.15}$$

So the distribution $p(X_{1:\infty})$ produced by the GPLVM is equivalent to the distribution $p(X'_{1:\infty})$ produced by the OUT model a.e.. □

D.3 Simulation details

In both scenarios, we generated sequences of fixed length $|X| = 30$, with an alphabet size of $B + 1 = 4$ (corresponding to nucleotides).

Scenario 1 We simulated from a Potts model

$$p_{\text{POTTS}}(x) = \frac{1}{\mathcal{Z}} \exp\left(\sum_l \sum_b h_{lb} x_{lb} + \sum_l \sum_{l'>l} \sum_b \sum_{b'} e_{ll'bb'} x_{lb} x_{lb'}\right)$$

where h is the sitewise energies, e is the pairwise energies, x is a one-hot sequence encoding, l indexes sequence positions and b indexes letters. Following the simulations in Ingraham & Marks [121], which were intended to roughly match the statistics of typical real protein Potts models, we drew $h_{lb} \sim \text{InvGamma}(2, 0.8)$ and

$$A_{ll'} = \begin{cases} 1 & \text{if } l' = l + 1 \\ \text{Bernoulli}(0.1) & \text{otherwise} \end{cases}$$

$$B_{ll'bb'} \sim \text{Normal}(0, 1.2)$$

$$e_{ll'bb'} = A_{ll'} B_{ll'bb'}.$$

The energies h and e were drawn once, and the same values used across independent simulations.

We sampled from the model using a Gibbs sampler with 100 steps of burn-in and 10 parallel chains using the code from Ingraham & Marks [121] (https://github.com/debbiemarkslab/persistent-vi). We shuffled the resulting samples to remove autocorrelation.

Scenario 2 We used a site-wise independent fitness function:

$$f(x) = \sum_{l=1}^{30} \sum_{b} h_{lb} x_{lb},$$

with site-wise residue biases h_l, where x_l is a one-hot encoding of the letter at the l-th position of x. To generate phylogenetically correlated sequences, we sampled phylogenetic trees from a Kingman Coalescent (Bertoin [22], Def. 2.1) with rate 1. Starting from a random sequence drawn from the steady state distribution at the root, we evolved the sequence simulating a Wright process in a haploid population (Sella & Hirsh [230], Eqn. 3) according to the tree and fitness function. In particular, for sequences x_0, x that are one mutation away, the mutation rate is

$$\lim_{\tau \to 0} \frac{1}{\tau} P^{\tau}(x, x_0) = N_{\text{eff}} \frac{e^{2(f(x) - f(x_0))} - 1}{e^{2N_{\text{eff}}(f(x) - f(x_0))} - 1},$$

where we set the effective population size to $N_{\text{eff}} = 10000$. This stochastic process has steady state

$$p^{\infty}(x) \propto \exp\left(2(N_{\text{eff}} - 1) f(x)\right),$$

(Sella & Hirsh [230], Eqn. 7).

SWI model We fit the SWI model with maximum likelihood estimation.

BEAR model In these simulations, we used a vanilla BEAR model with a uniform embedded AR model (i.e. a Bayesian Markov model) for simplicity. We set the Dirichlet prior concentration to the constant $\alpha = 0.5$. Based on the theoretical analysis in Amin et al. [12] (Thm. 35), we used a prior on lags of the form

$$p(L) \propto \exp(-B^L) \tag{D.16}$$

where B is the alphabet size (4 for nucleotides). We inferred the prior via empirical Bayes, marginalizing over the transition probabilities following the protocol in Amin et al. [12]. Conditional on lag L, sampling from the posterior over the BEAR model is straightforward thanks to Dirichlet-Categorical conjugancy.

Evaluation We defined $\mathcal{S}_f$ following standard protocols for fitness estimation models. In particular, we let $\mathcal{S}_f(p)$ be the Spearman correlation between $p(x)$ and $f(x)$ for $x \in \Lambda$ where Λ consists of all possible single point mutations (i.e. single letter changes) of an initial ("wild-type") sequence. The wild-type sequence was chosen as the most likely sequence under p^∞, computed exactly for Scenario 2 and estimated based on the 10^6 samples for Scenario 1.

To estimate model perplexity (Fig. 4.4C and D.5B), we used $N = 10,000$ independent sequences from p_0 and computed the per-residue perplexity

$$\exp\left(-\frac{1}{\sum_{n=1}^N |X_n|}\sum_{n=1}^N \log p(X_n)\right), \tag{D.17}$$

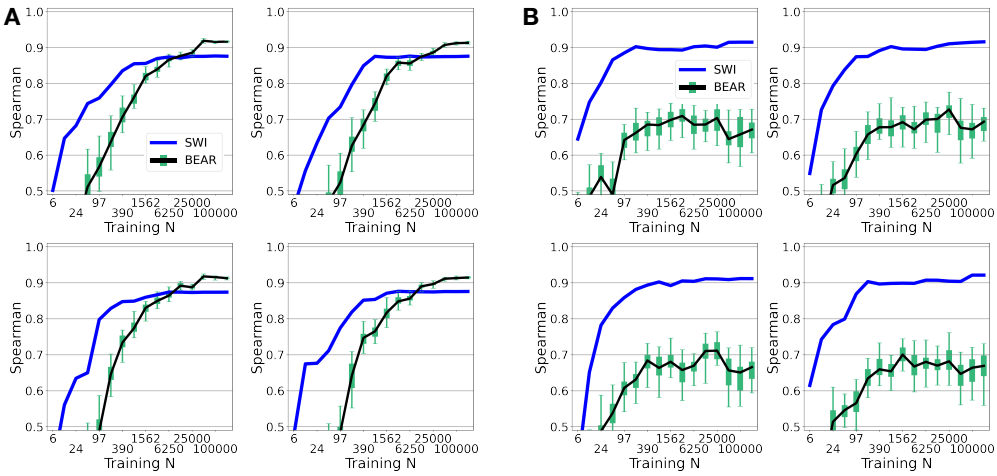


Figure D.3: (A) Same as Fig. 4.4A, for four independent simulations following Scenario 1. (B) Same as Fig. 4.4B, for four independent simulations following Scenario 2.

where $|X_n|$ is the sequence length and $p(X_n)$ is the probability of the sequence under the model.

To estimate the KL to the fitness distribution in Scenario 2 (Fig. 4.4D) , we sampled $N = 10,000$ independent sequences from p^∞, $\{X_1, \ldots, X_N\}$ and estimated

$$\text{KL}(p^\infty || p) \approx H(p^\infty) - \frac{1}{N} \sum_{n=1}^{N} \log p(X_n),$$

where $H(p^\infty)$ is the entropy of p^∞, which can be computed analytically. For BEAR, we plotted the KL to the posterior predictive, which, using Jensen's inequality can also be seen to lower bound

$$\mathbb{E}_{\Pi_{\text{BEAR}}(p|X_{\text{train}})}[\text{KL}(p^\infty || p)],$$

where $\Pi_{\text{BEAR}}(p|X_{\text{train}})$ is the BEAR posterior learned from the training dataset.

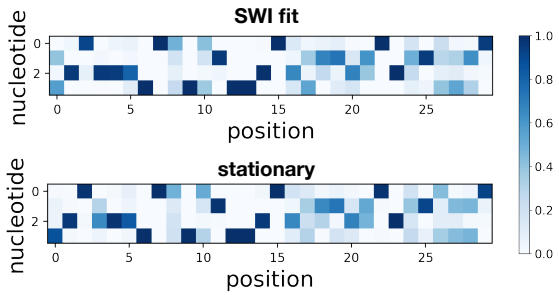


Figure D.4: Probability of each nucleotide at each position learned by the SWI model (above) and in the stationary distribution p^{∞} (below), for a simulation from Scenario 2.

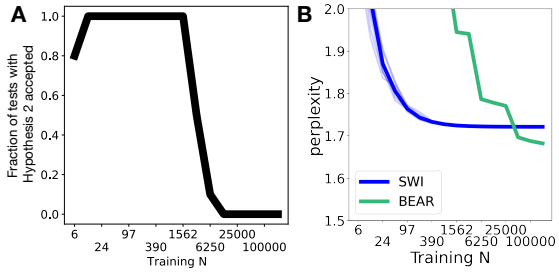


Figure D.5: (A) Fraction of independent simulations (out of 10 total), following Scenario 1 (Sec. 4.6), in which Hypothesis 2 was accepted at level $\alpha = 0.025$. (B) Perplexity on heldout data of the BEAR and the SWI models in Scenario 1. Thick line corresponds to the average over 10 individual simulations (thin lines).

D.4 Empirical results details

D.4.1 Data

Prediction task #1 (functional effect) Following standard practice, we report the absolute value of the Spearman correlation as $\mathcal{S}_f(p)$, since in some assays a negative change in the measured quantity corresponds to larger fitness (note that in all cases the predicted directionality of the effect under each model was correct). We focused on single amino acid substitutions, taking only those for which EVE was able to make a prediction (EVE is limited by its reliance on a multiple sequence alignment). We used the same data as in Shin et al. [235], Table 1, taking the 37 experiments performed on the following 32 proteins: UBC9_HUMAN, UBE4B_MOUSE, P84126_THETH, HIS7_YEAST, BLAT_ECOLX, IF1_ECOLI, PTEN_HUMAN, B3VI55_LIPST, GAL4_YEAST, POLG_HCVJF, PABP_YEAST, CALM1_HUMAN, AMIE_PSEAE, TRPC_THEMA, RASH_HUMAN, YAP1_HUMAN, TRPC_SULSO, DLG4_RAT, BG_STRSQ, KKA2_KLEPN, HSP82_YEAST, B3VI55_LIPST (stabilized), MK01_HUMAN, HIV BF520 env, SUMO1_HUMAN, RL401_YEAST, PA_FLU, HG_FLU, TPMT_HUMAN, HIV BG505 env, TPK1_HUMAN, and MTH3_HAEAE (stabilized).

Prediction task #2 (pathogenicity) We used the pathogenicity labels of single amino acid substitutions curated from ClinVar [148] in Frazer et al. [80]. We considered labels for 87 human proteins less than 250 amino acids in length: AICDA, AQP2, ATPF2, B9D2, CAH5A, CAV3, CD40L, CF410, CHC10, CIA30, CLD16, CLN8, COQ4, CRBB2, CRGD, CTRC, CXB1, CXB2, CXB3,

CXB4, CXB6, CY24A, DERM, DGUOK, DHDDS, EDAD, EFTS, ELNE, ETFB, ETHE1, EXOS3, FGF10, FGF23, FOXE3, FRDA, GP1BB, HBB, HEM4, HSPB1, HSPB8, IFM5, IFT27, JAGN1, KAD2, KCNE1, KCNE2, KITM, LITAF, MMAB, MMAC, MPU1, MYPR, NDP, NDUS8, NFU1, NKX25, NMNA1, OPA3, PAHX, PDYN, PMM2, PMP22, PNPH, PNPO, PROP1, PSPC, PTPS, RASH, RNH2A, S5A2, SAP3, SBDS, SCO1, SDHB, SDHF2, SIX1, SIX3, SOMA, TMM70, TNNT2, TPK1, TPM2, TR13B, TWST1, VHL, XLRS1, ZC4H2.

Training data All models were trained on datasets of protein sequences gathered as described in Shin et al. [235] for pathogenicity effect prediction tasks and as described in Frazer et al. [80] for functional effect prediction tasks. SWI and EVE were trained on the multiple sequence alignment, while Wavenet and BEAR were trained on the raw sequences as described in Shin et al. [235]. All datasets were uniformly subsampled to produce a 75%/25% train/test split.

D.4.2 Models and code

The SWI model was trained via maximum likelihood.

The Wavenet model was trained via maximum likelihood with the default architecture, hyperparameters and training protocol described in Shin et al. [235], for 100,000 steps. Code is from https://github.com/debbiemarkslab/SeqDesign. We did not apply the Wavenet model to the second prediction task, as it has only previously been developed for the first task.

The EVE model was trained via variational inference, using the same architecture, hyperparameters, and training protocol described in Frazer et al. [80]. Code is from https://github.com/debbiemarkslab/EVE. To match the protocol of the original paper, EVE was – unlike SWI, Wavenet

and BEAR – (a) trained on the full dataset rather than the training set alone, and (b) used a sequence reweighting heuristic.

The BEAR model used an embedded convolutional neural network (the same architecture as used in Amin et al. [12], with layer 1 width of 16, filter width of 5 and 30 filters total) and a uniform prior over lags 2, 3, 5, 7, and 9. Code is from `https://github.com/debbiemarkslab/BEAR`. The model was trained using empirical Bayes, as described in Amin et al. [12], for 500 steps with a batch size of 500000 kmers. To construct posterior credible intervals, we used 41 samples from the posterior for prediction task #1, and 1000 samples for prediction task #2.

We computed the heldout perplexity (Eqn. D.17) for the BEAR posterior predictive and for Wavenet to produce Fig. D.6.

D.4.3 Convergence experiments

To plot the convergence of the posterior over p_0 as a function of N (Fig. 4.5CD, D.7 and D.8), we used a vanilla BEAR model, a nonparametric Bayesian Markov model. Note that here we fixed the embedded AR model, rather than refitting with larger N, so that we could analyze the the convergence behavior with reference to the asymptotic results of Thm. 35 in Amin et al. [12], which does not take into account empirical Bayes. We set the Dirichlet concentration to 10 and used a prior over lags as in Eqn. D.16.

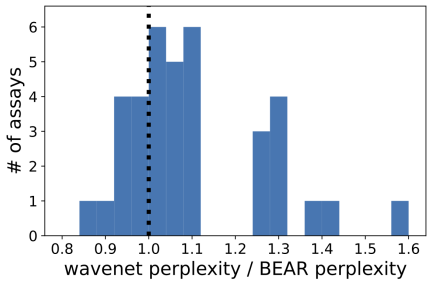


Figure D.6: Ratio of the per residue perplexity on heldout data of the Wavenet model and of the BEAR model posterior predictive, across the 37 assays used for the first prediction task. Note lower perplexity corresponds to better density estimation performance.

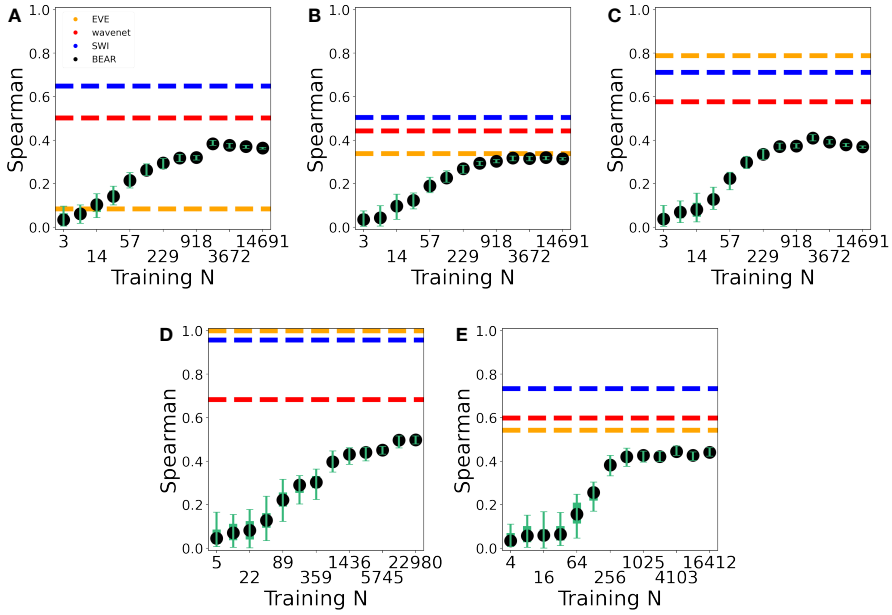


Figure D.7: Same as Fig. 4.5CD, for 5 additional assay examples. A-C are each distinct β-lactamase assays; D is from GAL4 (DNA-binding domain); E is from UBE4B (U-box domain).

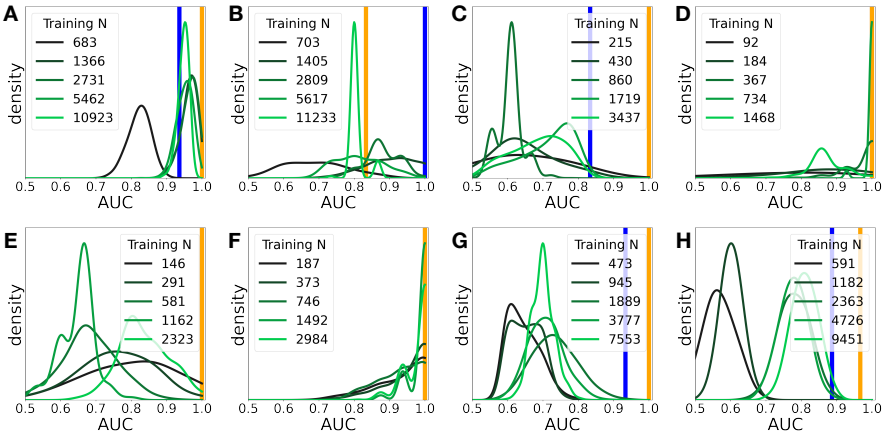


Figure D.8: Convergence of the BEAR posterior over AUCs with N (green distributions), compared to the AUC of SWI (blue line) and EVE (yellow line), for the second prediction task. (A) is for the *CXB1* gene, (B) *CXB6*, (C) *EXOS3*, (D) *FGF23*, (E) *OPA3*, (F) *PAHX*, (G) *PROP1*, (H) *S5A2*.

D.4.4 Interpolation experiments

We fit a BEAR model using the architecture and training protocol described in Sec. D.4.2, optimizing both the parameters of the AR model and h via empirical Bayes. We then varied h from its optimized value, and recalculated the total marginal likelihood and the posterior distribution over $\mathcal{S}_f(p)$ (Fig. 4.5EF and D.9). We also computed the value of $\mathcal{S}_f(q_{\hat{\theta}})$ for the fit BEAR model in the $h \to 0$ limit (Fig. D.10).

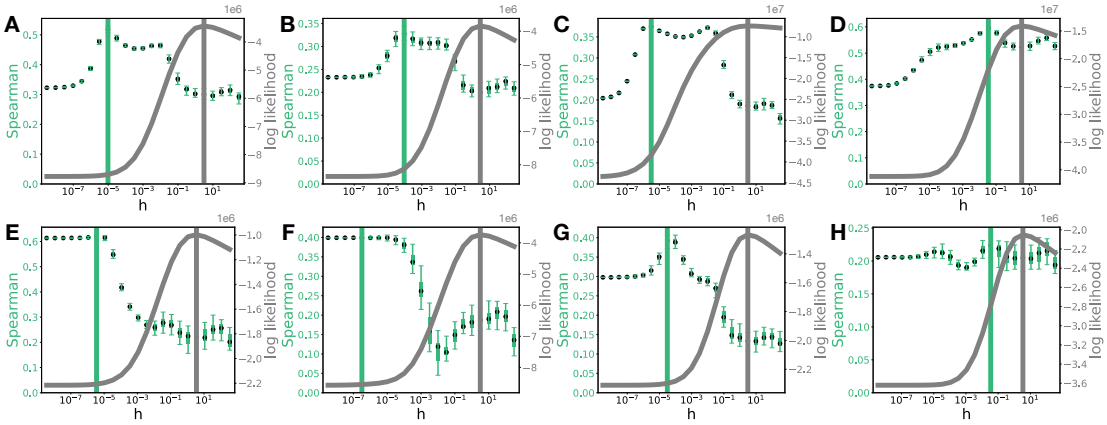


Figure D.9: Same as Fig. 4.5EF, for 8 additional assay examples. (A) Aliphatic amidase, (B) levoglucosan kinase (stabilized), (C) HIV env protein (BF520), (D) β-glucosidase, (E) UBE4B (U-box domain) (F) TIM barrel, (G) thiopurine S-methyltransferase, (H) thiamin pyrophosphokinase 1.

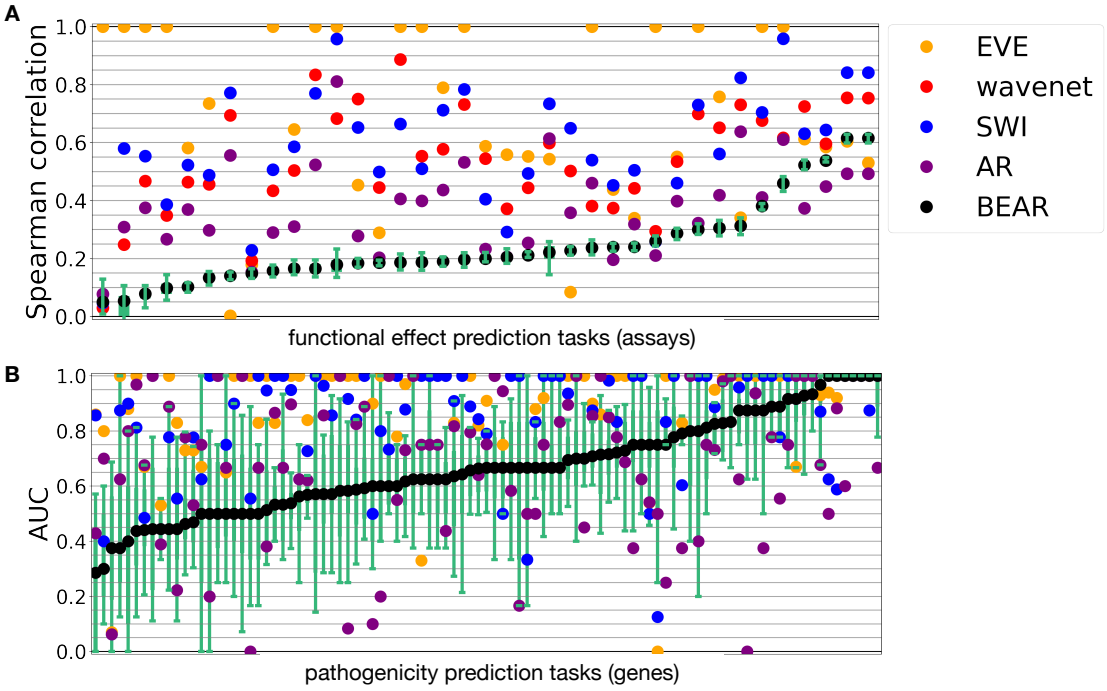


Figure D.10: Same as Fig. 4.5AB, with the addition of the AR model in the $h \to 0$ limit (purple). In prediction task 1 (A), Hypothesis 2 is accepted in 28/37 assays (75%) while Hypothesis 1 is accepted in 6/37 (16%) for the AR model. In prediction task 2 (B), Hypothesis 2 is accepted in 16/97 genes (16%) and Hypothesis 1 is accepted in 17/97 genes (18%) for the AR model.

E

Supplementary Material for Chapter 5

E.1 Methods details

E.1.1 Calibrating T

The SVC contains a hyperparameter $T > 0$. To choose an appropriate value of T, we aim, roughly, to match the coverage of the generalized posterior

$$\pi_N^{\text{SVC}}(\theta)d\theta = \frac{1}{z_N}\exp\Big(-\frac{N}{T}\widehat{\text{NKSD}}(p_0(x))\|q(x|\theta))\Big)\pi(\theta)d\theta$$

to the coverage of the standard Bayesian posterior

$$\pi_N^{\text{KL}}(\theta)d\theta = \frac{1}{q(X^{(1:N)})}\exp\Big(\sum_{i=1}^{N}\log q(X^{(i)}|\theta)\Big)\pi(\theta)d\theta$$

when the model is well-specified.

Let θ_* be the true parameter value, such that $p_0(x) = q(x|\theta_*)$ almost everywhere. Let $G^{\text{KL}}(\theta) := \nabla_\theta^2\mathbb{E}_{X\sim p_0}[-\log q(X|\theta)]$ and let $\theta_N^{\text{KL}} := \arg\max\sum_{i=1}^{N}\log q(X^{(i)}|\theta)$ be the maximum likelihood estimator. Let h_N^{KL} be the density of $\sqrt{N}(\theta - \theta_N^{\text{KL}})$ when $\theta \sim \pi_N^{\text{KL}}$. Under regularity conditions [176], according to the Bernstein–von Mises theorem, h_N^{KL} converges to a normal distribution in total variation,

$$\int_{\mathbb{R}^m}\Big|h_N^{\text{KL}}(x) - \mathcal{N}(x \mid 0, G^{\text{KL}}(\theta_*)^{-1})\Big|dx \xrightarrow[N\to\infty]{\text{a.s.}} 0.$$

According to Theorem 5.6.9, the generalized posterior associated with the SVC has analogous behavior. Let $G^{\text{SVC}}(\theta) := \nabla_\theta^2\frac{1}{T}\text{NKSD}(p_0(x)\|q(x|\theta))$ and let $\theta_N^{\text{SVC}} := \arg\min\widehat{\text{NKSD}}(p_0(x)\|q(x|\theta))$.

Let h_N^{SVC} be the density of $\sqrt{N}(\theta - \theta_N^{\text{SVC}})$ when $\theta \sim \pi_N^{\text{SVC}}$. Then by Theorem 5.6.9, h_N^{SVC} converges to a normal distribution in total variation,

$$\int_{\mathbb{R}^m} \left| h_N^{\text{SVC}}(x) - \mathcal{N}(x \mid 0, G^{\text{SVC}}(\theta_*)^{-1}) \right| dx \xrightarrow[N\to\infty]{\text{a.s.}} 0.$$

For the uncertainty in each posterior to be roughly the same order of magnitude, we want

$$\det G^{\text{KL}}(\theta_*) \approx \det G^{\text{SVC}}(\theta_*),$$

or equivalently,

$$T \approx \left(\frac{\det \left[\nabla_\theta^2 \big|_{\theta=\theta_*} \textsc{nksd}(p_0(x) \| q(x|\theta)) \right]}{\det \left[\nabla_\theta^2 \big|_{\theta=\theta_*} \mathbb{E}_{X\sim p_0}[-\log q(X|\theta)] \right]} \right)^{1/m}.$$

To choose a single T value, we simulate true parameters from the prior, generate data from each simulated true parameter, and take the median of the estimated T values. That is, we use the median $\hat{T}$ across samples drawn as

$$\begin{aligned} \theta_* &\sim \pi(\theta) \\ X^{(i)} &\overset{\text{iid}}{\sim} q(x|\theta_*) \\ \hat{T} &= \left(\frac{|\det \left[\nabla_\theta^2 \big|_{\theta=\theta_*} \widehat{\textsc{nksd}}(p_0(x) \| q(x|\theta)) \right]|}{|\det \left[\nabla_\theta^2 \big|_{\theta=\theta_*} \frac{1}{N} \sum_{i=1}^N -\log q(X^{(i)}|\theta) \right]|} \right)^{1/m}. \end{aligned} \tag{E.1}$$

In practice, we find that the order of magnitude of $\hat{T}$ is stable across samples θ_* from $\pi(\theta)$. See Section E.4.3 for an example.

E.1.2 Kernel recommendations

To obtain subsystem independence (Proposition 5.6.6), we suggest using a kernel that factors across subspaces, such that $k(X, Y) = k_{\mathcal{F}}(X_{\mathcal{F}}, Y_{\mathcal{F}})k_{\mathcal{B}}(X_{\mathcal{B}}, Y_{\mathcal{B}})$ where $k_{\mathcal{F}}$ and $k_{\mathcal{B}}$ are integrally strictly positive definite kernels. In the applications in Sections 5.7 and 5.8, we use the following kernel.

Definition E.1.1. *The* factored inverse multiquadric (IMQ) kernel *is defined as*

$$k(x, y) = \prod_{i=1}^{d} \left(c^2 + (x_i - y_i)^2\right)^{\beta/d}$$

for $x, y \in \mathbb{R}^d$, where $\beta \in [-1/2, 0)$ and $c > 0$.

Note that this kernel factors across any subset of dimensions, that is, if $S \subseteq \{1, \ldots, d\}$ and $S^c = \{1, \ldots, d\} \setminus S$, then we can write $k(x, y) = k_S(x_S, y_S)k_{S^c}(x_{S^c}, y_{S^c})$. Thus, if the foreground subspace $\mathcal{X}_{\mathcal{F}}$ is the span of a subset of the standard basis, such that $x_{\mathcal{F}} = V^\top x = x_S$ for some $S \subseteq \{1, \ldots, d\}$, then it follows that k factors as $k(x, y) = k_{\mathcal{F}}(x_{\mathcal{F}}, y_{\mathcal{F}})k_{\mathcal{B}}(x_{\mathcal{B}}, y_{\mathcal{B}})$. Along with this observation, the next result shows that the factored IMQ satisfies the conditions of Theorem 5.6.9 that pertain to k alone.

Proposition E.1.2. *The factored IMQ kernel is symmetric, positive, bounded, integrally strictly positive definite, and has continuous and bounded partial derivatives up to order 2.*

Proof. It is clear that $k(x, y) = k(y, x)$ and $k(x, y) > 0$. Next, we show that k has continuous and bounded partial derivatives up to order 2. Note that we can write $k(x, y) = \prod_{i=1}^{d} \psi(x_i - y_i)$

where $\psi(r) = (c^2 + r^2)^{\beta/d}$ for $r \in \mathbb{R}$. Differentiating, we have

$$\psi'(r) = \frac{\beta}{d}\frac{2r}{c^2 + r^2}\psi(r)$$
$$\psi''(r) = \Big(\frac{\beta^2}{d^2} - \frac{\beta}{d}\Big)\Big(\frac{2r}{c^2 + r^2}\Big)^2\psi(r) + \frac{\beta}{d}\frac{2}{c^2 + r^2}\psi(r).$$

Since $r^2 \geq 0$ and $\beta < 0$, $|\psi(r)| \leq c^{2\beta/d}$ for all $r \in \mathbb{R}$. Further, it is straightforward to verify that $|\psi'(r)|$ and $|\psi''(r)|$ are bounded on $\mathbb{R}$ by using the fact that $|r|/(c^2 + r^2) \leq 1/(2c)$. By the chain rule, it follows that for all i, j, the functions $k(x, y)$, $|\partial k/\partial x_i|$, and $|\partial^2 k/\partial x_i \partial y_j|$ are bounded. Thus, we conclude that k, $\|\nabla k\|$, and $\|\nabla^2 k\|$ are bounded.

Finally, we show that k is integrally strictly positive definite. First, for any d, for $x, y \in \mathbb{R}^d$, the function $(x, y) \mapsto (c^2 + \|x - y\|_2^2)^{\beta/d}$ is an integrally strictly positive definite kernel (see, for example, Section 3.1 of Sriperumbudur et al. [244]); we refer to this as the standard IMQ kernel. Since the factored IMQ is a product of one-dimensional standard IMQ kernels, it defines a kernel on $\mathbb{R}^d$ (Lemma 4.6 of Steinwart & Christmann [247]) and is positive definite (Theorem 4.16 of Steinwart & Christmann [247]). By Bochner's theorem (Theorem 3 of Sriperumbudur et al. [244]), a continuous positive definite kernel can be expressed in terms of the Fourier transform of a finite nonnegative Borel measure. In particular, applying Bochner's theorem to $\psi(r)$, we have

$$k(x, y) = \Psi(x - y) := \prod_{i=1}^{d} \psi(x_i - y_i) = \prod_{i=1}^{d} \int_{\mathbb{R}} \exp(-\sqrt{-1}(x_i - y_i)\omega_i) d\Lambda^0(\omega_i)$$
$$= \int_{\mathbb{R}^d} \exp(-\sqrt{-1}(x - y)^\top \omega) d\Lambda(\omega)$$

by Fubini's theorem, where Λ^0 is the finite nonnegative Borel measure on $\mathbb{R}$ associated with $\psi(r)$ and $\Lambda = \Lambda^0 \times \cdots \times \Lambda^0$ is the resulting product measure on $\mathbb{R}^d$. Applying Bochner's theorem in the other direction, we see that Ψ is a positive definite function. Moreover, since the standard IMQ kernel is characteristic (Theorem 7 of Sriperumbudur et al. [244]), it follows that the support of Λ^0 is $\mathbb{R}$ (Theorem 9 of Sriperumbudur et al. [244]), and thus the support of Λ is $\mathbb{R}^d$. This implies that the factored IMQ kernel k is characteristic (Theorem 9 of Sriperumbudur et al. [244]) and, since k is also translation invariant, k must be integrally strictly positive definite (Section 3.4 of Sriperumbudur et al. [243]). □

Our choice of the factored IMQ kernel is motivated by the analysis of Gorham & Mackey [93], which suggests that the standard IMQ is a good default choice for the kernelized Stein discrepancy, particularly when working with distributions that are (roughly speaking) very spread out. In particular, it is straightforward to show that the factored IMQ kernel, like the standard IMQ kernel, meets the conditions of Theorem 3.2 of Huggins & Mackey [116]. However, we do not pursue further the question of whether the NKSD with the factored IMQ detects convergence and non-convergence since our statistical setting is different from that of Gorham & Mackey [93], and we are assuming the data consists of i.i.d. samples from some underlying distribution rather than correlated samples from an MCMC chain which may or may not converge.

E.1.3 Exact solution for exponential families

Here, we show that when $q(x|\theta)$ is an exponential family, the estimated NKSD has the form

$$\widehat{\text{NKSD}}(p_0(x)\|q(x|\theta)) = \theta^\top A\,\theta + B^\top\theta + C \tag{E.2}$$

where A, B, and C depend on the data but not on θ. Since $q_\theta(x) = q(x|\theta) = \lambda(x)\exp(\theta^\top t(x) - \kappa(\theta))$, we have $s_{q_\theta}(x) = \nabla_x \log\lambda(x) + (\nabla_x t(x))^\top\theta$ where $(\nabla_x t(x))_{ij} = \partial t_i/\partial x_j$. Thus, we can write

$$\begin{aligned}
u_\theta(x,y) &:= s_{q_\theta}(x)^\top s_{q_\theta}(y)k(x,y) + s_{q_\theta}(x)^\top\nabla_y k(x,y) + s_{q_\theta}(y)^\top\nabla_x k(x,y) \qquad \text{(E.3)}\\
&\quad + \text{trace}(\nabla_x\nabla_y^\top k(x,y))\\
&= \theta^\top[(\nabla_x t(x))(\nabla_y t(y))^\top k(x,y)]\theta\\
&\quad + [(\nabla_x\log\lambda(x))^\top(\nabla_y t(y))^\top k(x,y) + (\nabla_y\log\lambda(y))^\top(\nabla_x t(x))^\top k(x,y)\\
&\quad\quad + (\nabla_x k(x,y))^\top(\nabla_y t(y))^\top + (\nabla_y k(x,y))^\top(\nabla_x t(x))^\top]\theta\\
&\quad + [(\nabla_x\log\lambda(x))^\top(\nabla_y\log\lambda(y))k(x,y) + (\nabla_y\log\lambda(y))^\top(\nabla_x k(x,y))\\
&\quad\quad + (\nabla_x\log\lambda(x))^\top(\nabla_y k(x,y)) + \text{trace}(\nabla_x\nabla_y^\top k(x,y))]. \qquad \text{(E.4)}
\end{aligned}$$

Then the estimated NKSD takes the form in Equation E.2 if we choose

$$
\begin{aligned}
A &:= \frac{1}{\sum_{i\neq j} k(X^{(i)}, X^{(j)})} \sum_{i\neq j} \nabla_x t(X^{(i)}) \nabla_x t(X^{(j)})^\top k(X^{(i)}, X^{(j)}) \\
B^\top &:= \frac{1}{\sum_{i\neq j} k(X^{(i)}, X^{(j)})} \sum_{i\neq j} \big[(\nabla_x \log \lambda(X^{(i)}))^\top \nabla_x t(X^{(j)})^\top k(X^{(i)}, X^{(j)}) \\
&\qquad + (\nabla_x \log \lambda(X^{(j)}))^\top \nabla_x t(X^{(i)})^\top k(X^{(i)}, X^{(j)}) \\
&\qquad + (\nabla_x k(X^{(i)}, X^{(j)}))^\top \nabla_x t(X^{(j)})^\top \\
&\qquad + (\nabla_y k(X^{(i)}, X^{(j)}))^\top \nabla_x t(X^{(i)})^\top \big] \\
C &:= \frac{1}{\sum_{i\neq j} k(X^{(i)}, X^{(j)})} \sum_{i\neq j} \big[(\nabla_x \log \lambda(X^{(i)}))^\top (\nabla_x \log \lambda(X^{(j)})) k(X^{(i)}, X^{(j)}) \\
&\qquad + (\nabla_x \log \lambda(X^{(j)}))^\top \nabla_x k(X^{(i)}, X^{(j)}) \\
&\qquad + (\nabla_x \log \lambda(X^{(i)}))^\top \nabla_y k(X^{(i)}, X^{(j)}) \\
&\qquad + \operatorname{trace}(\nabla_x \nabla_y^\top k(X^{(i)}, X^{(j)}))\big].
\end{aligned}
$$

If the prior on θ is $\mathcal{N}(\mu, \Sigma_0)$, then the SVC is

$$\begin{aligned}
\mathcal{K} &= \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} (2\pi)^{-m_{\mathcal{F}}/2} (\det \Sigma_0)^{-1/2} \\
&\quad \times \int \exp\Big(-\frac{N}{T}[\theta^\top A\,\theta + B^\top \theta + C]\Big) \exp\Big(-\frac{1}{2}(\theta-\mu)^\top \Sigma_0^{-1}(\theta-\mu)\Big) d\theta \\
&= \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} (2\pi)^{-m_{\mathcal{F}}/2} (\det \Sigma_0)^{-1/2} \\
&\quad \times \int \exp\Big(-\frac{1}{2}\theta^\top \Big(\frac{2N}{T}A + \Sigma_0^{-1}\Big)\theta + \Big(-\frac{N}{T}B^\top + \mu^\top \Sigma_0^{-1}\Big)\theta - \frac{N}{T}C - \frac{1}{2}\mu^\top \Sigma_0^{-1}\mu\Big) d\theta \\
&= \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} (\det \Sigma_0)^{-1/2} \Big(\det\Big(\frac{2N}{T}A + \Sigma_0^{-1}\Big)\Big)^{-1/2} \\
&\quad \times \exp\Big(\frac{1}{2}\Big(-\frac{N}{T}B^\top + \mu^\top \Sigma_0^{-1}\Big)^\top \Big(\frac{2N}{T}A + \Sigma_0^{-1}\Big)^{-1}\Big(-\frac{N}{T}B^\top + \mu^\top \Sigma_0^{-1}\Big) \\
&\qquad\qquad - \frac{N}{T}C - \frac{1}{2}\mu^\top \Sigma_0^{-1}\mu\Big).
\end{aligned}$$

Meanwhile, if $q(x|\theta) = \mathcal{N}(\theta, \Sigma)$ where Σ is a fixed covariance matrix, then we have $\nabla_x \log \lambda(x) = -\Sigma^{-1}x$ and $\nabla_x t(x) = \Sigma^{-1}$.

E.1.4 Comparing many foregrounds using approximate optima

Here, we justify the technique described in Section 5.2.3. As in Section 5.2.3, define $\ell_j(\theta) = \widehat{\text{NKSD}}(p_0(x_{\mathcal{F}_j}) \| q(x_{\mathcal{F}_j}|\theta))$ for $j \in \{1, 2\}$, and let $\theta_N(w) = \arg\min_\theta \mathcal{L}(w, \theta)$ where

$$\mathcal{L}(w, \theta) := \ell_1(\theta) + w(\ell_2(\theta) - \ell_1(\theta))$$

for $w \in [0,1]$. We assume that the conditions of Theorem 5.6.9 are met, over both $\mathcal{X}_{\mathcal{F}_1}$ and $\mathcal{X}_{\mathcal{F}_2}$. Since $(\partial\mathcal{L}/\partial\theta_i)(w, \theta_N(w)) = 0$, we have

$$0 = \frac{\partial}{\partial w}\Big(\frac{\partial \mathcal{L}}{\partial \theta_i}(w, \theta_N(w))\Big) = \frac{\partial^2 \mathcal{L}}{\partial w \partial \theta_i}(w, \theta_N(w)) + \sum_j \frac{\partial^2 \mathcal{L}}{\partial \theta_i \partial \theta_j}(w, \theta_N(w))\Big(\frac{\partial}{\partial w}\theta_{N,j}(w)\Big),$$

or equivalently, in matrix/vector notation,

$$0 = \nabla_w(\nabla_\theta \mathcal{L}(w, \theta_N(w))) = \nabla_\theta \nabla_w \mathcal{L}(w, \theta_N) + \nabla^2_\theta \mathcal{L}(w, \theta_N)\nabla_w(\theta_N(w)).$$

Rearranging, we have

$$\nabla_w \theta_N(w) = -\big(\nabla^2_\theta \mathcal{L}(w, \theta_N)\big)^{-1} \nabla_\theta \nabla_w \mathcal{L}(w, \theta_N).$$

At $w = 0$ we find, plugging back in the definition of $\mathcal{L}$,

$$\begin{aligned}\nabla_w \theta_N(0) &= -\nabla^2_\theta \ell_1(\theta_N(0))^{-1}(\nabla_\theta \ell_2(\theta_N(0)) - \nabla_\theta \ell_1(\theta_N(0))) \\ &= -\nabla^2_\theta \ell_1(\theta_N(0))^{-1} \nabla_\theta \ell_2(\theta_N(0)).\end{aligned}$$

Applying a first-order Taylor series expansion gives us $\theta_N(1) \approx \theta_N(0) + \nabla_w \theta_N(0)$, which yields Equation 5.13.

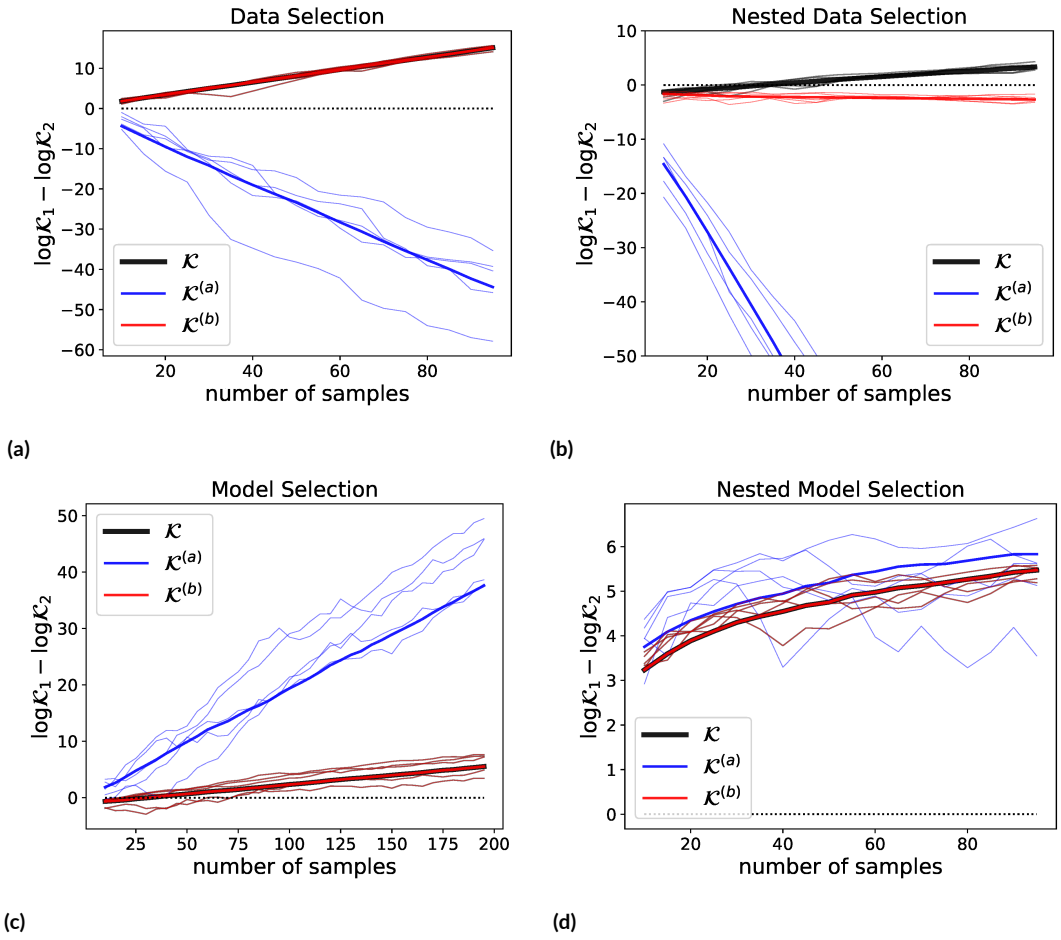


Figure E.1: Behavior of the Stein volume criterion $\mathcal{K}$, the foreground marginal likelihood with a background volume correction $\mathcal{K}^{(\mathrm{a})}$, and the foreground marginal nksd $\mathcal{K}^{(\mathrm{b})}$ on toy examples. The plots show the results for 5 randomly generated datasets (thin lines) and the average over 100 random datasets (bold lines). Here, unlike Figure 5.2, the Pitman-Yor expression for $m_{\mathcal{B}}$ is used, with $\alpha = 0.5, \theta = 1$, and $D = 0.2$.

E.2 Asymptotics of the alternative selection criteria

Theorem 5.6.17 shows that the SVC exhibits all four types of consistency: data selection, nested data selection, model selection, and nested model selection. In this section, we establish the consistency properties of the alternative criteria considered in Section 5.3.

E.2.1 Setup

We first review the asymptotics of the standard marginal likelihood, discussed in depth by Dawid [50] and Hong & Preston [108], for example. Define

$$f_N^{\mathrm{KL}}(\theta) := -\frac{1}{N}\sum_{i=1}^{N} \log q(X^{(i)}|\theta), \qquad \theta_N^{\mathrm{KL}} := \arg\min_\theta f_N^{\mathrm{KL}}(\theta),$$

$$f^{\mathrm{KL}}(\theta) := -\mathbb{E}_{X\sim p_0}[\log q(X|\theta)], \qquad \theta_*^{\mathrm{KL}} := \arg\min_\theta f^{\mathrm{KL}}(\theta).$$

Let m be the dimension of the parameter space. Under suitable regularity conditions [176], the Laplace approximation to the marginal likelihood is

$$q(X^{(1:N)}) = \int q(X^{(1:N)}|\theta)\pi(\theta)d\theta \sim \frac{\exp\big(-Nf_N^{\mathrm{KL}}(\theta_N^{\mathrm{KL}})\big)\pi(\theta_*^{\mathrm{KL}})}{\big|\det \nabla_\theta^2 f^{\mathrm{KL}}(\theta_*^{\mathrm{KL}})\big|^{1/2}} \left(\frac{2\pi}{N}\right)^{m/2} \tag{E.5}$$

almost surely, where $a_N \sim b_N$ indicates that $a_N/b_N \to 1$ as $N \to \infty$. We can rewrite this as

$$\begin{aligned} &\log q(X^{(1:N)}) + N(f_N^{\mathrm{KL}}(\theta_N^{\mathrm{KL}}) - f_N^{\mathrm{KL}}(\theta_*^{\mathrm{KL}})) \\ &\qquad + N(f_N^{\mathrm{KL}}(\theta_*^{\mathrm{KL}}) - f^{\mathrm{KL}}(\theta_*^{\mathrm{KL}})) + Nf^{\mathrm{KL}}(\theta_*^{\mathrm{KL}}) \\ &\qquad + \frac{m}{2}\log N - \log\left(\frac{\pi(\theta_*^{\mathrm{KL}})(2\pi)^{m/2}}{\big|\det \nabla_\theta^2 f^{\mathrm{KL}}(\theta_*^{\mathrm{KL}})\big|^{1/2}}\right) \xrightarrow[N\to\infty]{\text{a.s.}} 0. \end{aligned} \tag{E.6}$$

As shown by Dawid [50] and Hong & Preston [108], under regularity conditions,

$$\begin{aligned} N(f_N^{\mathrm{KL}}(\theta_N^{\mathrm{KL}}) - f_N^{\mathrm{KL}}(\theta_*^{\mathrm{KL}})) &= O_{P_0}(1) \\ N(f_N^{\mathrm{KL}}(\theta_*^{\mathrm{KL}}) - f^{\mathrm{KL}}(\theta_*^{\mathrm{KL}})) &= O_{P_0}(\sqrt{N}) \\ N f^{\mathrm{KL}}(\theta_*^{\mathrm{KL}}) &= O_{P_0}(N) \\ \log\left(\frac{\pi(\theta_*^{\mathrm{KL}})(2\pi)^{m/2}}{\left|\det \nabla_\theta^2 f^{\mathrm{KL}}(\theta_*^{\mathrm{KL}})\right|^{1/2}}\right) &= O_{P_0}(1). \end{aligned} \tag{E.7}$$

The NKSD marginal likelihood has a similar decomposition. Following Section 5.6, define

$$\begin{aligned} f_N^{\mathrm{NKSD}}(\theta) &:= \frac{1}{T}\widehat{\mathrm{NKSD}}(p_0(x)\|q(x|\theta)), & \theta_N^{\mathrm{NKSD}} &:= \arg\min_\theta f_N^{\mathrm{NKSD}}(\theta), \\ f^{\mathrm{NKSD}}(\theta) &:= \frac{1}{T}\mathrm{NKSD}(p_0(x)\|q(x|\theta)), & \theta_*^{\mathrm{NKSD}} &:= \arg\min_\theta f^{\mathrm{NKSD}}(\theta). \end{aligned}$$

As shown in Theorem 5.6.9,

$$z_N := \int \exp(-N f_N^{\mathrm{NKSD}}(\theta))\pi(\theta)d\theta \sim \frac{\exp(-N f_N^{\mathrm{NKSD}}(\theta_N^{\mathrm{NKSD}}))\pi(\theta_*^{\mathrm{NKSD}})}{\left|\det \nabla_\theta^2 f^{\mathrm{NKSD}}(\theta_*^{\mathrm{NKSD}})\right|^{1/2}}\left(\frac{2\pi}{N}\right)^{m/2}$$

almost surely as $N \to \infty$. As above, we can rewrite this as

$$\begin{aligned} &\log z_N + N(f_N^{\mathrm{NKSD}}(\theta_N^{\mathrm{NKSD}}) - f_N^{\mathrm{NKSD}}(\theta_*^{\mathrm{NKSD}})) \\ &\quad + N(f_N^{\mathrm{NKSD}}(\theta_*^{\mathrm{NKSD}}) - f^{\mathrm{NKSD}}(\theta_*^{\mathrm{NKSD}})) + N f^{\mathrm{NKSD}}(\theta_*^{\mathrm{NKSD}}) \\ &\quad + \frac{m}{2}\log N - \log\left(\frac{\pi(\theta_*^{\mathrm{NKSD}})(2\pi)^{m/2}}{\left|\det \nabla_\theta^2 f^{\mathrm{NKSD}}(\theta_*^{\mathrm{NKSD}})\right|^{1/2}}\right) \xrightarrow[N\to\infty]{\text{a.s.}} 0. \end{aligned} \tag{E.8}$$

By Theorem 5.6.12, we have

$$\begin{aligned} N(f_N^{\text{NKSD}}(\theta_N^{\text{NKSD}}) - f_N^{\text{NKSD}}(\theta_*^{\text{NKSD}})) &= O_{P_0}(1), \\ N(f_N^{\text{NKSD}}(\theta_*^{\text{NKSD}}) - f^{\text{NKSD}}(\theta_*^{\text{NKSD}})) &= O_{P_0}(\sqrt{N}), \\ N f^{\text{NKSD}}(\theta_*^{\text{NKSD}}) &= O_{P_0}(N), \\ \log\left(\frac{\pi(\theta_*^{\text{NKSD}})(2\pi)^{m/2}}{\left|\det \nabla_\theta^2 f^{\text{NKSD}}(\theta_*^{\text{NKSD}})\right|^{1/2}}\right) &= O_{P_0}(1), \end{aligned} \tag{E.9}$$

and further, when the model is well-specified, such that $\text{NKSD}(p_0(x)\|q(x|\theta_*^{\text{NKSD}})) = 0$,

$$N(f_N^{\text{NKSD}}(\theta_*^{\text{NKSD}}) - f^{\text{NKSD}}(\theta_*^{\text{NKSD}})) = O_{P_0}(1). \tag{E.10}$$

For ease of reference, here are the various scores that we consider for model/data selection.

Marginal likelihood of the augmented model (foreground+background):

$$\tilde{q}(X^{(1:N)}|\mathcal{F}) = \int\int q(X_{\mathcal{F}}^{(1:N)}|\theta)\, \tilde{q}(X_{\mathcal{B}}^{(1:N)}|X_{\mathcal{F}}^{(1:N)}, \phi_{\mathcal{B}})\pi(\theta)\pi_{\mathcal{B}}(\phi_{\mathcal{B}})d\theta d\phi_{\mathcal{B}}.$$

Foreground marginal NKSD, background volume correction (a.k.a. the SVC):

$$\mathcal{K} := \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} \int \exp\Big(-\frac{N}{T}\widehat{\text{NKSD}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\Big)\pi(\theta)d\theta.$$

Foreground marginal likelihood, background volume correction:

$$\mathcal{K}^{(\mathrm{a})} := \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} q(X_{\mathcal{F}}^{(1:N)}).$$

Foreground marginal NKSD:

$$\mathcal{K}^{(\mathrm{b})} := \int \exp\left(-\frac{N}{T}\widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\right)\pi(\theta)d\theta.$$

Foreground marginal KL, background volume correction:

$$\mathcal{K}^{(\mathrm{c})} := \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} \int \exp\left(-N\widehat{\mathrm{KL}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\right)\pi(\theta)d\theta.$$

Foreground NKSD, background volume correction:

$$\mathcal{K}^{(\mathrm{d})} := \left(\frac{2\pi}{N}\right)^{m_{\mathcal{B}}/2} \exp\Big(-\frac{N}{T}\min_{\theta}\widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\Big).$$

Foreground NKSD, foreground and background volume correction (a.k.a. BIC for SVC)

$$\mathcal{K}^{\mathrm{BIC}} := \left(\frac{2\pi}{N}\right)^{(m_{\mathcal{F}}+m_{\mathcal{B}})/2} \exp\Big(-\frac{N}{T}\min_{\theta}\widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q(x_{\mathcal{F}}|\theta))\Big).$$

E.2.2 Data selection

Assume $m_{\mathcal{B}_j} = o(N/\log N)$ for $j \in \{1, 2\}$. By Equations E.6 and E.7,

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{(\mathrm{a})}}{\mathcal{K}_2^{(\mathrm{a})}} \xrightarrow[N\to\infty]{P_0} \mathbb{E}_{X\sim p_0}[-\log q(X_{\mathcal{F}_2}|\theta_{2,*}^{\mathrm{KL}})] - \mathbb{E}_{X\sim p_0}[-\log q(X_{\mathcal{F}_1}|\theta_{1,*}^{\mathrm{KL}})] \tag{E.11}$$

$$= \mathrm{KL}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta_{2,*}^{\mathrm{KL}})) + H_{\mathcal{F}_2} - \mathrm{KL}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta_{1,*}^{\mathrm{KL}})) - H_{\mathcal{F}_1},$$

so $\mathcal{K}^{(\mathrm{a})}$ does not satisfy data selection consistency. The SVC satisfies data selection consistency by Theorem 5.6.17 (part 1). We show that the other scores also satisfy data selection consistency. Since $\mathcal{K}^{(\mathrm{b})} = (2\pi/N)^{-m_{\mathcal{B}}/2}\mathcal{K}$ where $\mathcal{K}$ is the SVC, by Theorem 5.6.17 (part 1),

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{(\mathrm{b})}}{\mathcal{K}_2^{(\mathrm{b})}} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\mathrm{NKSD}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta_{2,*}^{\mathrm{NKSD}})) - \frac{1}{T}\mathrm{NKSD}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta_{1,*}^{\mathrm{NKSD}})). \tag{E.12}$$

By Equation E.11 and the fact that $\mathcal{K}^{(\mathrm{c})} = \exp(NH_{\mathcal{F}})\mathcal{K}^{(\mathrm{a})}$, we have

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{(\mathrm{c})}}{\mathcal{K}_2^{(\mathrm{c})}} \xrightarrow[N\to\infty]{P_0} \mathrm{KL}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta_{2,*}^{\mathrm{KL}})) - \mathrm{KL}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta_{1,*}^{\mathrm{KL}})). \tag{E.13}$$

Since $\mathcal{K}^{(\mathrm{d})} = (2\pi/N)^{m_{\mathcal{B}}/2}\exp(-Nf_N^{\mathrm{NKSD}}(\theta_N^{\mathrm{NKSD}}))$, then by Equation E.9,

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{(\mathrm{d})}}{\mathcal{K}_2^{(\mathrm{d})}} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\mathrm{NKSD}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta_{2,*}^{\mathrm{NKSD}})) - \frac{1}{T}\mathrm{NKSD}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta_{1,*}^{\mathrm{NKSD}})). \tag{E.14}$$

Similarly, since $\mathcal{K}^{\text{BIC}} = (2\pi/N)^{m_{\mathcal{F}}/2}\mathcal{K}^{(\text{d})}$,

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{\text{BIC}}}{\mathcal{K}_2^{\text{BIC}}} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\textsc{nksd}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta_{2,*}^{\text{NKSD}})) - \frac{1}{T}\textsc{nksd}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta_{1,*}^{\text{NKSD}})). \tag{E.15}$$

These methods therefore satisfy data selection consistency. For the marginal likelihood of the augmented model, suppose $m_{\mathcal{B}_1}$ and $m_{\mathcal{B}_2}$ do not depend on N. Then by Equation E.6,

$$\begin{aligned}\frac{1}{N}\log\frac{\tilde{q}(X^{(1:N)}|\mathcal{F}_1)}{\tilde{q}(X^{(1:N)}|\mathcal{F}_2)} \xrightarrow[N\to\infty]{P_0} &\mathbb{E}_{X_{\mathcal{F}_2}\sim p_0}[-\log q(X_{\mathcal{F}_2}|\theta_{2,*}^{\text{KL}})] + \mathbb{E}_{X\sim p_0}[-\log\tilde{q}(X_{\mathcal{B}_2}|X_{\mathcal{F}_2},\phi_{2,*}^{\text{KL}})] \\ &- \mathbb{E}_{X_{\mathcal{F}_1}\sim p_0}[-\log q(X_{\mathcal{F}_1}|\theta_{1,*}^{\text{KL}})] - \mathbb{E}_{X\sim p_0}[-\log\tilde{q}(X_{\mathcal{B}_1}|X_{\mathcal{F}_1},\phi_{1,*}^{\text{KL}})]]\end{aligned} \tag{E.16}$$

We can rewrite this in terms of the KL divergence. First note the decomposition,

$$H = -\int p_0(x)\log p_0(x)dx = -\int p_0(x_{\mathcal{F}_j})\log p_0(x_{\mathcal{F}_j})dx_{\mathcal{F}_j} - \int p_0(x)\log p_0(x_{\mathcal{B}_j}|x_{\mathcal{F}_j})dx$$

for $j \in \{1,2\}$. Adding and subtracting the entropy H in Equation E.16, and using the fact that the background model is well-specified,

$$\begin{aligned}\frac{1}{N}\log\frac{\tilde{q}(X^{(1:N)}|\mathcal{F}_1)}{\tilde{q}(X^{(1:N)}|\mathcal{F}_2)} \xrightarrow[N\to\infty]{P_0} &\textsc{kl}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta_{2,*}^{\text{KL}})) + \textsc{kl}(p_0(x_{\mathcal{B}_2}|x_{\mathcal{F}_2})\|\tilde{q}(x_{\mathcal{B}_2}|x_{\mathcal{F}_2},\phi_{2,*}^{\text{KL}})) \\ &- \textsc{kl}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta_{1,*}^{\text{KL}})) - \textsc{kl}(p_0(x_{\mathcal{B}_1}|x_{\mathcal{F}_1})\|\tilde{q}(x_{\mathcal{B}_1}|x_{\mathcal{F}_1},\phi_{1,*}^{\text{KL}})) \\ = &\textsc{kl}(p_0(x_{\mathcal{F}_2})\|q(x_{\mathcal{F}_2}|\theta_{2,*}^{\text{KL}})) - \textsc{kl}(p_0(x_{\mathcal{F}_1})\|q(x_{\mathcal{F}_1}|\theta_{1,*}^{\text{KL}})).\end{aligned} \tag{E.17}$$

E.2.3 Nested data selection

In nested data selection, we are concerned with situations in which $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$ and the model is well-specified over both $\mathcal{X}_{\mathcal{F}_1}$ and $\mathcal{X}_{\mathcal{F}_2}$. Assume further that $m_{\mathcal{B}_2} - m_{\mathcal{B}_1}$ does not depend on N. First, consider $\mathcal{K}^{(\mathrm{d})}$ and $\mathcal{K}^{\mathrm{BIC}}$. Since $\mathcal{K}^{(\mathrm{d})} = (2\pi/N)^{m_{\mathcal{B}}/2} \exp(-N f_N^{\mathrm{NKSD}}(\theta_N^{\mathrm{NKSD}}))$ and by Theorem 5.6.12, $f_N^{\mathrm{NKSD}}(\theta_N^{\mathrm{NKSD}}) = O_{P_0}(1/N)$, we have

$$\frac{1}{\log N} \log \frac{\mathcal{K}_1^{(\mathrm{d})}}{\mathcal{K}_2^{(\mathrm{d})}} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{B}_2} - m_{\mathcal{B}_1}}{2}. \tag{E.18}$$

Likewise, since $\mathcal{K}^{\mathrm{BIC}} = (2\pi/N)^{m_{\mathcal{F}}/2} \mathcal{K}^{(\mathrm{d})}$, it follows that

$$\frac{1}{\log N} \log \frac{\mathcal{K}_1^{\mathrm{BIC}}}{\mathcal{K}_2^{\mathrm{BIC}}} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{F}_2} + m_{\mathcal{B}_2} - m_{\mathcal{F}_1} - m_{\mathcal{B}_1}}{2}. \tag{E.19}$$

As in Section 5.6.4, it is natural to assume $m_{\mathcal{B}_2} > m_{\mathcal{B}_1}$ and $m_{\mathcal{F}_2} + m_{\mathcal{B}_2} > m_{\mathcal{F}_1} + m_{\mathcal{B}_1}$, in which case these criteria satisfy nested data selection consistency.

None of $\mathcal{K}^{(\mathrm{a})}$, $\mathcal{K}^{(\mathrm{b})}$, and $\mathcal{K}^{(\mathrm{c})}$ are guaranteed to satisfy nested data selection consistency, because the contribution of background model complexity is negligible or nonexistent. To see this, note that assuming $m_{\mathcal{B}_j} = o(N/\log N)$, by Equation E.11 we have

$$\frac{1}{N} \log \frac{\mathcal{K}_1^{(\mathrm{a})}}{\mathcal{K}_2^{(\mathrm{a})}} \xrightarrow[N\to\infty]{P_0} H_{\mathcal{F}_2} - H_{\mathcal{F}_1}. \tag{E.20}$$

Meanwhile, since $\mathcal{K}^{(\mathrm{b})} = (2\pi/N)^{-m_{\mathcal{B}}/2}\mathcal{K}$ then by Theorem 5.6.17 (part 2),

$$\frac{1}{\log N}\log\frac{\mathcal{K}_1^{(\mathrm{b})}}{\mathcal{K}_2^{(\mathrm{b})}} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{F}_2} - m_{\mathcal{F}_1}}{2}. \tag{E.21}$$

Since $\mathcal{X}_{\mathcal{F}_2} \subset \mathcal{X}_{\mathcal{F}_1}$, we have $m_{\mathcal{F}_2} \leq m_{\mathcal{F}_1}$ except perhaps in highly contrived scenarios. If $m_{\mathcal{F}_2} < m_{\mathcal{F}_1}$ then Equation E.21 shows that $\log(\mathcal{K}_1^{(\mathrm{b})}/\mathcal{K}_2^{(\mathrm{b})}) \xrightarrow{P_0} -\infty$. On the other hand, if $m_{\mathcal{F}_2} = m_{\mathcal{F}_1}$, then by Equations E.8 and E.9, $\log(\mathcal{K}_1^{(\mathrm{b})}/\mathcal{K}_2^{(\mathrm{b})}) = O_{P_0}(1)$, so it is not possible to have $\log(\mathcal{K}_1^{(\mathrm{b})}/\mathcal{K}_2^{(\mathrm{b})}) \xrightarrow{P_0} \infty$. Therefore, $\mathcal{K}^{(\mathrm{b})}$ does not satisfy nested data selection consistency.

Since $\mathcal{K}^{(\mathrm{c})} = e^{NH_{\mathcal{F}}}\mathcal{K}^{(\mathrm{a})} = e^{NH_{\mathcal{F}}}(2\pi/N)^{m_{\mathcal{B}}/2}q(X_{\mathcal{F}}^{(1:N)})$, then by Equations E.6 and E.7,

$$\frac{1}{\sqrt{N}}\log\frac{\mathcal{K}_1^{(\mathrm{c})}}{\mathcal{K}_2^{(\mathrm{c})}} = \sqrt{N}\Big(\frac{1}{N}\sum_{i=1}^{N}\log\frac{p_0(X_{\mathcal{F}_1}^{(i)})}{p_0(X_{\mathcal{F}_2}^{(i)})} - \mathbb{E}\Big(\log\frac{p_0(X_{\mathcal{F}_1})}{p_0(X_{\mathcal{F}_2})}\Big)\Big) + O_{P_0}(N^{-1/2}\log N). \tag{E.22}$$

If $\sigma^2 := \mathbb{V}_{P_0}(\log p_0(X_{\mathcal{F}_1})/p_0(X_{\mathcal{F}_2}))$ is positive and finite, then by the central limit theorem and Slutsky's theorem, $N^{-1/2}\log(\mathcal{K}_1^{(\mathrm{c})}/\mathcal{K}_2^{(\mathrm{c})}) \xrightarrow{D} \mathcal{N}(0,\sigma^2)$. Thus, $\mathcal{K}^{(\mathrm{c})}$ randomly selects $\mathcal{F}_1$ or $\mathcal{F}_2$ with equal probability, and therefore, it does not satisfy nested data selection consistency.

For the marginal likelihood of the augmented model, suppose $m_{\mathcal{B}_1}$ and $m_{\mathcal{B}_2}$ do not depend on N. The marginal likelihood achieves nested data selection consistency because the augmented models are both well-specified and describe the complete data space $\mathcal{X}$; this guarantees that the $O_{P_0}(\sqrt{N})$ terms in the marginal likelihood decomposition cancel. Specifically, $p_0(x) = q(x \mid$

$\theta^{\text{KL}}_{j,*}, \phi^{\text{KL}}_{j,*}, \mathcal{F}_j)$ for $j \in \{1, 2\}$, and thus, by Equations E.6 and E.7 applied to the augmented model,

$$\frac{1}{\log N} \log \frac{\tilde{q}(X^{(1:N)}|\mathcal{F}_1)}{\tilde{q}(X^{(1:N)}|\mathcal{F}_2)} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{F}_2} + m_{\mathcal{B}_2} - m_{\mathcal{F}_1} - m_{\mathcal{B}_1}}{2}. \tag{E.23}$$

Nested data selection consistency follows assuming $m_{\mathcal{F}_2} + m_{\mathcal{B}_2} > m_{\mathcal{F}_1} + m_{\mathcal{B}_1}$ as before. This can be contrasted with Equation E.22, where although both foreground models are well-specified, they describe different data ($X^{(1:N)}_{\mathcal{F}_1}$ versus $X^{(1:N)}_{\mathcal{F}_2}$), so the $O_{P_0}(\sqrt{N})$ terms remain.

E.2.4 Model selection

All of the criteria we consider satisfy model selection consistency. To see this, we apply the same asymptotic analysis as used for data selection in Section E.2.2, under the same conditions on $m_{\mathcal{B}}$, obtaining

$$\frac{1}{N} \log \frac{\tilde{q}_1(X^{(1:N)}|\mathcal{F})}{\tilde{q}_2(X^{(1:N)}|\mathcal{F})} \xrightarrow[N\to\infty]{P_0} \text{KL}(p_0(x_{\mathcal{F}}) \| q_2(x_{\mathcal{F}}|\theta^{\text{KL}}_{2,*})) - \text{KL}(p_0(x_{\mathcal{F}}) \| q_1(x_{\mathcal{F}}|\theta^{\text{KL}}_{1,*})), \tag{E.24}$$

$$\frac{1}{N} \log \frac{\mathcal{K}^{(\text{a})}_1}{\mathcal{K}^{(\text{a})}_2} \xrightarrow[N\to\infty]{P_0} \text{KL}(p_0(x_{\mathcal{F}}) \| q_2(x_{\mathcal{F}}|\theta^{\text{KL}}_{2,*})) - \text{KL}(p_0(x_{\mathcal{F}}) \| q_1(x_{\mathcal{F}}|\theta^{\text{KL}}_{1,*})), \tag{E.25}$$

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{(\mathrm{b})}}{\mathcal{K}_2^{(\mathrm{b})}} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\textsc{nksd}(p_0(x_\mathcal{F})\|q_2(x_\mathcal{F}|\theta_{2,*}^{\textsc{nksd}})) - \frac{1}{T}\textsc{nksd}(p_0(x_\mathcal{F})\|q_1(x_\mathcal{F}|\theta_{1,*}^{\textsc{nksd}})), \tag{E.26}$$

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{(\mathrm{c})}}{\mathcal{K}_2^{(\mathrm{c})}} \xrightarrow[N\to\infty]{P_0} \textsc{kl}(p_0(x_\mathcal{F})\|q_2(x_\mathcal{F}|\theta_{2,*}^{\textsc{kl}})) - \textsc{kl}(p_0(x_\mathcal{F})\|q_1(x_\mathcal{F}|\theta_{1,*}^{\textsc{kl}})), \tag{E.27}$$

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{(\mathrm{d})}}{\mathcal{K}_2^{(\mathrm{d})}} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\textsc{nksd}(p_0(x_\mathcal{F})\|q_2(x_\mathcal{F}|\theta_{2,*}^{\textsc{nksd}})) - \frac{1}{T}\textsc{nksd}(p_0(x_\mathcal{F})\|q_1(x_\mathcal{F}|\theta_{1,*}^{\textsc{nksd}})), \tag{E.28}$$

$$\frac{1}{N}\log\frac{\mathcal{K}_1^{\mathrm{BIC}}}{\mathcal{K}_2^{\mathrm{BIC}}} \xrightarrow[N\to\infty]{P_0} \frac{1}{T}\textsc{nksd}(p_0(x_\mathcal{F})\|q_2(x_\mathcal{F}|\theta_{2,*}^{\textsc{nksd}})) - \frac{1}{T}\textsc{nksd}(p_0(x_\mathcal{F})\|q_1(x_\mathcal{F}|\theta_{1,*}^{\textsc{nksd}})). \tag{E.29}$$

Note that in contrast to the data selection case, $\mathcal{K}^{(\mathrm{a})}$ satisfies model selection consistency since the entropy terms $H_{\mathcal{F}_j}$ cancel due to the fact that $\mathcal{F}$ is fixed. We can think of this as a consequence of the KL divergence's subsystem independence; if we are just interested in modeling a fixed foreground space, there is no problem considering the foreground marginal likelihood alone[34,35,212].

E.2.5 Nested model selection

In nested model selection, since both models are well-specified, we have $q_j(x_{\mathcal{F}}|\theta^{\mathrm{KL}}_{j,*}) = p_0(x_{\mathcal{F}}) = q_j(x_{\mathcal{F}}|\theta^{\mathrm{NKSD}}_{j,*})$ for $j \in \{1,2\}$. Thus, the estimated divergences cancel:

$$\widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q_1(x_{\mathcal{F}}|\theta^{\mathrm{NKSD}}_{1,*})) = \widehat{\mathrm{NKSD}}(p_0(x_{\mathcal{F}})\|q_2(x_{\mathcal{F}}|\theta^{\mathrm{NKSD}}_{2,*})),$$
$$\sum_{i=1}^{N} \log q_1(X^{(i)}_{\mathcal{F}}|\theta^{\mathrm{KL}}_{1,*}) = \sum_{i=1}^{N} \log q_2(X^{(i)}_{\mathcal{F}}|\theta^{\mathrm{KL}}_{2,*}),$$
$$\widehat{\mathrm{KL}}(p_0(x_{\mathcal{F}})\|q_1(x_{\mathcal{F}}|\theta^{\mathrm{KL}}_{1,*})) = \widehat{\mathrm{KL}}(p_0(x_{\mathcal{F}})\|q_2(x_{\mathcal{F}}|\theta^{\mathrm{KL}}_{2,*})).$$

Using this along with Equations E.6–E.10, under the same conditions on $m_{\mathcal{B}}$ as in Section E.2.2,

$$\frac{1}{\log N} \log \frac{\tilde{q}_1(X^{(1:N)}|\mathcal{F})}{\tilde{q}_2(X^{(1:N)}|\mathcal{F})} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{F},2} - m_{\mathcal{F},1}}{2}, \tag{E.30}$$

$$\frac{1}{\log N} \log \frac{\mathcal{K}^{(\mathrm{a})}_1}{\mathcal{K}^{(\mathrm{a})}_2} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{F},2} - m_{\mathcal{F},1}}{2}, \tag{E.31}$$

$$\frac{1}{\log N} \log \frac{\mathcal{K}^{(\mathrm{b})}_1}{\mathcal{K}^{(\mathrm{b})}_2} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{F},2} - m_{\mathcal{F},1}}{2}, \tag{E.32}$$

$$\frac{1}{\log N} \log \frac{\mathcal{K}^{(\mathrm{c})}_1}{\mathcal{K}^{(\mathrm{c})}_2} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{F},2} - m_{\mathcal{F},1}}{2}, \tag{E.33}$$

$$\log \frac{\mathcal{K}_1^{(\mathrm{d})}}{\mathcal{K}_2^{(\mathrm{d})}} = O_{P_0}(1), \tag{E.34}$$

$$\frac{1}{\log N} \log \frac{\mathcal{K}_1^{\mathrm{BIC}}}{\mathcal{K}_2^{\mathrm{BIC}}} \xrightarrow[N\to\infty]{P_0} \frac{m_{\mathcal{F},2} - m_{\mathcal{F},1}}{2}, \tag{E.35}$$

where we are using the assumption that the background model is the same in the two augmented models $\tilde{q}_1$ and $\tilde{q}_2$ and so $m_{\mathcal{B},1} = m_{\mathcal{B},2}$. Only $\mathcal{K}^{(\mathrm{d})}$ fails to satisfy nested model selection consistency.

E.3 Proofs

E.3.1 Proofs of NKSD properties

Proof of Proposition 5.6.3. By assumption, the kernel is bounded, say $|k(x,y)| \leq B$, and $s_p, s_q \in L^1(P)$. Thus, by the Cauchy–Schwarz inequality,

$$\begin{aligned}&\left| \int_{\mathcal{X}} \int_{\mathcal{X}} (s_q(x) - s_p(x))^\top (s_q(y) - s_p(y)) k(x,y) p(x) p(y) dx dy \right| \\ &\qquad \leq B \Big(\textstyle\int_{\mathcal{X}} \|s_q(x) - s_p(x)\| p(x) dx \Big)^2 < \infty.\end{aligned}$$

Since the kernel is integrally strictly positive definite and $|k(x,y)| \leq B$,

$$0 < \int_{\mathcal{X}} \int_{\mathcal{X}} k(x,y) p(x) p(y) dx dy \leq B < \infty. \tag{E.36}$$

Thus, the NKSD is finite. Equation 5.30 follows from Theorem 3.6 of Liu et al. [158]. □

Proof of Proposition 5.6.4. The denominator of the NKSD is positive since k is integrally strictly positive definite. Defining $\delta(x) = s_q(x) - s_p(x)$, the numerator of the NKSD is

$$\int_{\mathcal{X}} \int_{\mathcal{X}} \delta(x)^\top \delta(y) k(x,y) p(x) p(y) dx dy = \sum_{i=1}^{d} \int_{\mathcal{X}} \int_{\mathcal{X}} \delta_i(x) \delta_i(y) k(x,y) p(x) p(y) dx dy. \quad \text{(E.37)}$$

If $\delta_i(x)p(x) = 0$ almost everywhere with respect to Lebesgue measure on $\mathcal{X}$, then the ith term on the right-hand side is zero. Meanwhile, if $\delta_i(x)p(x)$ is not a.e. zero, then $\int_{\mathcal{X}} |\delta_i(x)| p(x) dx > 0$, and hence, the ith term is positive since k is integrally strictly positive definite and $\delta_i \in L^1(P)$ by assumption. Hence, the NKSD is nonnegative, and equals zero if and only if $\delta(x)p(x) = 0$ almost everywhere.

Suppose $\delta(x)p(x) = 0$ almost everywhere. Since $p(x) > 0$ on $\mathcal{X}$ by assumption, this implies $s_p(x) = s_q(x)$ a.e., and in fact, $s_p(x) = s_q(x)$ for all $x \in \mathcal{X}$ by continuity. Since $\mathcal{X}$ is open and connected, then by the gradient theorem (that is, the fundamental theorem of calculus for line integrals), $p(x) \propto q(x)$, and hence, $p(x) = q(x)$ on $\mathcal{X}$. Conversely, if $p(x) = q(x)$ almost everywhere, then $\delta(x)p(x) = 0$ almost everywhere. □

Proof of Proposition 5.6.6. Define

$$\delta_1(x_1) := \nabla_{x_1} \log q(x) - \nabla_{x_1} \log p(x) = \nabla_{x_1} \log q(x_1) - \nabla_{x_1} \log p(x_1)$$

$$\delta_2(x_2) := \nabla_{x_2} \log q(x) - \nabla_{x_2} \log p(x) = \nabla_{x_2} \log q(x_2) - \nabla_{x_2} \log p(x_2).$$

Let $X, Y \sim p(x)$ independently. Note that $\mathbb{E}[k_1(X_1, Y_1)] > 0$ and $\mathbb{E}[k_2(X_2, Y_2)] > 0$ since k_1 and k_2 are integrally strictly positive definite by assumption. Therefore,

$$\begin{aligned}
\textsc{nksd}(p(x)\|q(x)) &= \frac{\mathbb{E}[(\nabla_x \log q(X) - \nabla_x \log p(X))^\top (\nabla_x \log q(Y) - \nabla_x \log p(Y)) k(X, Y)]}{\mathbb{E}[k(X, Y)]} \\
&= \frac{\mathbb{E}[\delta_1(X_1)^\top \delta_1(Y_1) k_1(X_1, Y_1)]\mathbb{E}[k_2(X_2, Y_2)]}{\mathbb{E}[k_1(X_1, Y_1)]\mathbb{E}[k_2(X_2, Y_2)]} + \frac{\mathbb{E}[\delta_2(X_2)^\top \delta_2(Y_2) k_2(X_2, Y_2)]\mathbb{E}[k_1(X_1, Y_1)]}{\mathbb{E}[k_1(X_1, Y_1)]\mathbb{E}[k_2(X_2, Y_2)]} \\
&= \frac{\mathbb{E}[\delta_1(X_1)^\top \delta_1(Y_1) k_1(X_1, Y_1)]}{\mathbb{E}[k_1(X_1, Y_1)]} + \frac{\mathbb{E}[\delta_2(X_2)^\top \delta_2(Y_2) k_2(X_2, Y_2)]}{\mathbb{E}[k_2(X_2, Y_2)]} \\
&= \textsc{nksd}(p(x_1)\|q(x_1)) + \textsc{nksd}(p(x_2)\|q(x_2)).
\end{aligned}$$

□

The following modified version applies to the estimator $\widehat{\textsc{nksd}}(p\|q)$ (Equation 5.5).

Proposition E.3.1.

$$\widehat{\textit{NKSD}}(p(x)\|q(x)) = \overline{\textit{NKSD}}(p(x_1)\|q(x_1)) + \overline{\textit{NKSD}}(p(x_2)\|q(x_2)) \tag{E.38}$$

where

$$\begin{aligned}
\overline{\textit{NKSD}}(p(x_1)\|q(x_1)) :=& \frac{\sum_{i \neq j} u_1(X_1^{(i)}, X_1^{(j)}) k_2(X_2^{(i)}, X_2^{(j)})}{\sum_{i \neq j} k_1(X_1^{(i)}, X_1^{(j)}) k_2(X_2^{(i)}, X_2^{(j)})} \\
u_1(x_1, y_1) :=& s_q(x_1)^\top s_q(y_1) k_1(x_1, y_1) + s_q(x_1)^\top \nabla_{y_1} k_1(x_1, y_1) + s_q(y_1)^\top \nabla_{x_1} k_1(x_1, y_1) \\
&+ \text{trace}(\nabla_{x_1} \nabla_{y_1}^\top k_1(x_1, y_1)) \\
s_q(x_1) :=& \nabla_{x_1} \log q(x_1),
\end{aligned}$$

and vice versa for $\overline{\text{NKSD}}(p(x_2)\|q(x_2))$ *with the roles of 1 and 2 swapped.*

Proof. Recall the definition of $\widehat{\text{NKSD}}(p(x)\|q(x))$ in Equation 5.5. Note that $\nabla_{x_1} k(x,y) = k_2(x_2,y_2)\nabla_{x_1} k_1(x_1,y_1)$ and $\nabla_{x_1} \log q(x) = \nabla_{x_1} \log q(x_1)$. Examining $u(x,y)$ term-by-term,

$$\begin{aligned}
\nabla_x \log q(x)^\top \nabla_y \log q(y) k(x,y) =& \left[\nabla_{x_1} \log q(x_1)^\top \nabla_{y_1} \log q(y_1) k_1(x_1,y_1)\right] k_2(x_2,y_2) \\
&+ \left[\nabla_{x_2} \log q(x_2)^\top \nabla_{y_2} \log q(y_2) k_2(x_2,y_2)\right] k_1(x_1,y_1), \\
\nabla_x \log q(x)^\top \nabla_y k(x,y) =& [\nabla_{x_1} \log q(x_1)^\top \nabla_{y_1} k_1(x_1,y_1)] k_2(x_2,y_2) \\
&+ [\nabla_{x_2} \log q(x_2)^\top \nabla_{y_2} k_2(x_2,y_2)] k_1(x_1,y_1), \\
\nabla_x k(x,y)^\top \nabla_y \log q(y) =& [\nabla_{x_1} k_1(x_1,y_1)^\top \nabla_{y_1} \log q(y_1)] k_2(x_2,y_2), \\
&+ [\nabla_{x_2} k_2(x_2,y_2)^\top \nabla_{y_2} \log q(y_2)] k_1(x_1,y_1) \\
\text{trace}(\nabla_x \nabla_y^\top k(x,y)) =& \text{trace}(\nabla_{x_1} \nabla_{y_1}^\top k_1(x_1,y_1)) k_2(x_2,y_2), \\
&+ \text{trace}(\nabla_{x_2} \nabla_{y_2}^\top k_2(x_2,y_2)) k_1(x_1,y_1).
\end{aligned}$$

Thus, defining u_1 and u_2 as in Proposition E.3.1, we have

$$\begin{aligned}
u(x,y) &= u_1(x_1,y_1)k_2(x_2,y_2) + u_2(x_2,y_2)k_1(x_1,y_1), \\
k(x,y) &= k_1(x_1,y_1)k_2(x_2,y_2).
\end{aligned}$$

The result follows. □

To interpret Proposition E.3.1, note that

$$\frac{\mathbb{E}_{X,Y\sim p}[u_1(X_1,Y_1)k_2(X_2,Y_2)]}{\mathbb{E}_{X,Y\sim p}[k_1(X_1,Y_1)k_2(X_2,Y_2)]} = \frac{\mathbb{E}_{X_1,Y_1\sim p(x_1)}[u_1(X_1,Y_1)]}{\mathbb{E}_{X_1,Y_1\sim p(x_1)}[k_1(X_1,Y_1)]} = \textsc{nksd}(p(x_1)\|q(x_1)),$$

so $\overline{\textsc{nksd}}(p(x_1)\|q(x_1))$ is an estimator of $\textsc{nksd}(p(x_1)\|q(x_1))$, and likewise for $\overline{\textsc{nksd}}(p(x_2)\|q(x_2))$.

E.3.2 Proof of Theorems 5.6.9 and 5.6.11

Our proofs in this section build on the proof of Theorem 3 of Barp et al. [17].

Proposition E.3.2. *Under the assumptions of Theorem 5.6.9, for any compact convex $C \subseteq \Theta$,*

$$\sup_{\theta\in C} |f_N(\theta) - f(\theta)| \xrightarrow{\text{a.s.}} 0. \tag{E.39}$$

Proof. First, we establish almost sure convergence for the denominator of $f_N(\theta)$. Since k is assumed to be bounded and to have bounded derivatives up to order two, we can choose $B < \infty$ such that $B \geq |k| + \|\nabla_x k\| + \|\nabla_x \nabla_y^\top k\|$. In particular, the expected value of the kernel is finite:

$$\int_{\mathcal{X}}\int_{\mathcal{X}} |k(x,y)|P_0(dx)P_0(dy) \leq B < \infty. \tag{E.40}$$

By the strong law of large numbers for U-statistics (Theorem 5.4A of Serfling [231]),

$$\frac{1}{N(N-1)}\sum_{i\neq j} k(X^{(i)},X^{(j)}) \xrightarrow[N\to\infty]{\text{a.s.}} \int_{\mathcal{X}}\int_{\mathcal{X}} k(x,y)P_0(dx)P_0(dy). \tag{E.41}$$

Note that the limit is positive since $k(x, y) > 0$ for all $x, y \in \mathcal{X}$. For the numerator, we establish bounds on u_θ and $\nabla_\theta u_\theta$. Let $C \subseteq \Theta$ be compact and convex. By Equation 5.5, for all $\theta \in C$ and all $x, y \in \mathcal{X}$,

$$\begin{aligned}
|u_\theta(x,y)| &\leq |s_{q_\theta}(x)^\top s_{q_\theta}(y) k(x,y)| + |s_{q_\theta}(x)^\top \nabla_y k(x,y)| \\
&\quad + |s_{q_\theta}(y)^\top \nabla_x k(x,y)| + |\operatorname{trace}(\nabla_x \nabla_y^\top k(x,y))| \\
&\leq \|s_{q_\theta}(x)\| \|s_{q_\theta}(y)\| B + \|s_{q_\theta}(x)\| B + \|s_{q_\theta}(y)\| B + Bd \\
&\leq g_{0,C}(x) g_{0,C}(y) B + g_{0,C}(x) B + g_{0,C}(y) B + Bd \\
&=: h_{0,C}(x,y).
\end{aligned} \tag{E.42}$$

Similarly, for all $\theta \in C$ and all $x, y \in \mathcal{X}$,

$$\begin{aligned}
\|\nabla_\theta u_\theta(x,y)\| &\leq \|\nabla_\theta (s_{q_\theta}(x)^\top s_{q_\theta}(y)) k(x,y)\| + \|\nabla_\theta (s_{q_\theta}(x)^\top \nabla_y k(x,y))\| \\
&\quad + \|\nabla_\theta (s_{q_\theta}(y)^\top \nabla_x k(x,y))\| + \|\nabla_\theta \operatorname{trace}(\nabla_x \nabla_y^\top k(x,y))\| \\
&\leq g_{0,C}(x) g_{1,C}(y) B + g_{0,C}(y) g_{1,C}(x) B + g_{1,C}(x) B + g_{1,C}(y) B \\
&=: h_{1,C}(x,y).
\end{aligned} \tag{E.43}$$

Note that $h_{0,C}$ and $h_{1,C}$ are continuous and belong to $L^1(P_0 \times P_0)$.

Let $S_1 \subseteq S_2 \subseteq \cdots \subseteq \mathcal{X}$ be a sequence of compact sets such that $\cup_{M=1}^\infty S_M = \mathcal{X}$. Note that this implies $\cup_{M=1}^\infty S_M \times S_M = \mathcal{X} \times \mathcal{X}$. Suppose for the moment that, for each M, the following collections of functions are equicontinuous on C: (A) $(\theta \mapsto u_\theta(x,y) : x, y \in S_M)$ and (B)

$(\theta \mapsto \int u_\theta(x,y)P_0(dy) : x \in S_M)$. Assuming this, Theorem 1 of Yeo & Johnson [296] shows that

$$\sup_{\theta\in C}\left|\frac{1}{N(N-1)}\sum_{i\neq j} u_\theta(X^{(i)},X^{(j)}) - \int_{\mathcal{X}}\int_{\mathcal{X}} u_\theta(x,y)P_0(dx)P_0(dy)\right| \xrightarrow[N\to\infty]{\text{a.s.}} 0, \tag{E.44}$$

and that $\theta \mapsto \int_{\mathcal{X}}\int_{\mathcal{X}} u_\theta(x,y)P_0(dx)P_0(dy)$ is continuous. (Note that although Yeo & Johnson [296] assume $\mathcal{X} = \mathbb{R}$, their proof goes through without further modification for any nonempty $\mathcal{X} \subseteq \mathbb{R}^d$.) Combining Equations E.41 and E.44, we have

$$\frac{\sup_{\theta\in C}\left|\frac{1}{N(N-1)}\sum_{i\neq j} u_\theta(X^{(i)},X^{(j)}) - \int\int u_\theta(x,y)P_0(dx)P_0(dy)\right|}{\frac{1}{N(N-1)}\sum_{i\neq j} k(X^{(i)},X^{(j)})} \xrightarrow[N\to\infty]{\text{a.s.}} 0.$$

Thus, it follows that $\sup_{\theta\in C}|f_N(\theta) - f(\theta)| \to 0$ a.s. by Equations E.41 and E.42. To complete the proof, we must show that (A) and (B) are equicontinuous on C.

(A) Since $\theta \mapsto u_\theta(x,y)$ is differentiable on C, then by the mean value theorem, we have that for all $\theta_1, \theta_2 \in C$ and all $x, y \in S_M$,

$$\begin{aligned}
|u_{\theta_1}(x,y) - u_{\theta_2}(x,y)| &\leq \|\nabla_\theta|_{\theta=\tilde{\theta}}\, u_\theta(x,y)\|\|\theta_1 - \theta_2\| \\
&\leq h_{1,C}(x,y)\|\theta_1 - \theta_2\| \\
&\leq \Big(\sup_{x,y\in S_M} h_{1,C}(x,y)\Big)\|\theta_1 - \theta_2\| < \infty
\end{aligned}$$

where $\tilde{\theta} = \gamma\theta_1 + (1-\gamma)\theta_2$ for some $\gamma \in [0,1]$. Here, the second inequality holds since $\tilde{\theta} \in C$ by the convexity of C, and the supremum is finite because a continuous function on a compact set

attains its maximum. Therefore, $(\theta \mapsto u_\theta(x,y) : x, y \in S_M)$ is equicontinuous on C.

(B) To see that $\left(\theta \mapsto \int u_\theta(x,y)P_0(dy) : x \in S_M\right)$ is equicontinuous on C, first note that

$$\int |u_\theta(x,y)|P_0(dy) \leq \int h_{0,C}(x,y)P_0(dy) < \infty.$$

Further, due to Equations E.42 and E.43, we can apply the Leibniz integral rule[78] Theorem 2.27 and find that $\nabla_\theta \int u_\theta(x,y)P_0(dy)$ exists and is equal to $\int \nabla_\theta u_\theta(x,y)P_0(dy)$. Now we apply the mean value theorem and the same reasoning as before to find that for all $\theta_1, \theta_2 \in C$ and all $x \in S_M$,

$$\begin{aligned}\left|\int u_{\theta_1}(x,y)P_0(dy) - \int u_{\theta_2}(x,y)P_0(dy)\right| &\leq \left\|\nabla_\theta|_{\theta=\tilde{\theta}} \int u_\theta(x,y)P_0(dy)\right\|\|\theta_1 - \theta_2\| \\ &\leq \|\theta_1 - \theta_2\| \int \left\|\nabla_\theta|_{\theta=\tilde{\theta}}\, u_\theta(x,y)\right\|P_0(dy) \\ &\leq \|\theta_1 - \theta_2\| \sup_{x \in S_M} \int h_{1,C}(x,y)P_0(dy) < \infty\end{aligned}$$

where $\tilde{\theta} = \gamma\theta_1 + (1-\gamma)\theta_2$ for some $\gamma \in [0,1]$. The supremum is finite since $x \mapsto \int h_{1,C}(x,y)P_0(dy)$ is continuous, which can easily be seen by plugging in the definition of $h_{1,C}$. Therefore, $(\theta \mapsto \int u_\theta(x,y)P_0(dy) : x \in S_M)$ is equicontinuous on C. □

Proposition E.3.3. *Under the assumptions of Theorem 5.6.9,* $(f_N''' : N \in \mathbb{N})$ *is uniformly bounded on* E.

Proof. First, for any $x, y \in \mathcal{X}$, if we define $g(\theta) = s_{q_\theta}(x)$ and $h(\theta) = s_{q_\theta}(y)$ then $u_\theta = (g^\top h)k + g^\top(\nabla_y k) + h^\top(\nabla_x k) + \text{trace}(\nabla_x \nabla_y^\top k)$. By differentiating, applying Minkowski's

inequality to the resulting sum of tensors, and applying the Cauchy–Schwarz inequality to each term, we have

$$\|\nabla_\theta^3 u_\theta(x,y)\| \le \|\nabla^3 g\|\|h\|k + 3\|\nabla^2 g\|\|\nabla h\|k + 3\|\nabla g\|\|\nabla^2 h\|k + \|g\|\|\nabla^3 h\|k + \|\nabla^3 g\|\|\nabla_y k\| + \|\nabla^3 h\|\|\nabla_x k\|.$$

Using the symmetry of the kernel to combine like terms, this yields that

$$\Big\|\sum_{i\neq j}\nabla_\theta^3 u_\theta(X^{(i)},X^{(j)})\Big\| \le \sum_{i\neq j}\Big(2\|\nabla_\theta^3 s_{q_\theta}(X^{(i)})\|\|s_{q_\theta}(X^{(j)})\|B + 6\|\nabla_\theta^2 s_{q_\theta}(X^{(i)})\|\|\nabla_\theta s_{q_\theta}(X^{(j)})\|B + 2\|\nabla_\theta^3 s_{q_\theta}(X^{(i)})\|B\Big)$$

where $B < \infty$ such that $B \ge |k| + \|\nabla_x k\| + \|\nabla_x \nabla_y^\top k\|$. Since $f_N(\theta) = 0$ when $N = 1$ by definition, we can assume without loss of generality that $N \ge 2$, so $\frac{1}{N-1} = \frac{1}{N}(1 + \frac{1}{N-1}) \le 2/N$.

Since each term is non-negative, we can add in the $i = j$ terms,

$$
\begin{aligned}
&\Big\| \frac{1}{N(N-1)} \sum_{i \neq j} \nabla_\theta^3 u_\theta(X^{(i)}, X^{(j)}) \Big\| \\
&\leq \frac{2B}{N^2} \sum_{i,j} \Big(2\|\nabla_\theta^3 s_{q_\theta}(X^{(i)})\| \|s_{q_\theta}(X^{(j)})\| + 6\|\nabla_\theta^2 s_{q_\theta}(X^{(i)})\| \|\nabla_\theta s_{q_\theta}(X^{(j)})\| + 2\|\nabla_\theta^3 s_{q_\theta}(X^{(i)})\| \Big) \\
&= 4B \Big(\frac{1}{N} \sum_i \|\nabla_\theta^3 s_{q_\theta}(X^{(i)})\| \Big) \Big(\frac{1}{N} \sum_j \|s_{q_\theta}(X^{(j)})\| \Big) \qquad \text{(E.45)} \\
&\quad + 12B \Big(\frac{1}{N} \sum_i \|\nabla_\theta^2 s_{q_\theta}(X^{(i)})\| \Big) \Big(\frac{1}{N} \sum_j \|\nabla_\theta s_{q_\theta}(X^{(j)})\| \Big) \\
&\quad + 4B \Big(\frac{1}{N} \sum_i \|\nabla_\theta^3 s_{q_\theta}(X^{(i)})\| \Big).
\end{aligned}
$$

By assumption, $\{ \frac{1}{N} \sum_i \|\nabla_\theta^2 s_{q_\theta}(X^{(i)})\| : N \in \mathbb{N}, \theta \in E \}$ is bounded with probability 1, and similarly for $\{ \frac{1}{N} \sum_i \|\nabla_\theta^3 s_{q_\theta}(X^{(i)})\| : N \in \mathbb{N}, \theta \in E \}$. We show the same for $\frac{1}{N} \sum_i \|s_{q_\theta}(X^{(i)})\|$ and $\frac{1}{N} \sum_i \|\nabla_\theta s_{q_\theta}(X^{(i)})\|$. By Equation 5.40, we have

$$
\int \sup_{\theta \in \bar{E}} \|s_{q_\theta}(x)\| P_0(dx) \leq \int g_{0,\bar{E}}(x) P_0(dx) < \infty.
$$

Hence, by Theorem 1.3.3 of Ghosh & Ramamoorthi [88], $\frac{1}{N} \sum_i \|s_{q_\theta}(X^{(i)})\|$ converges uniformly on $\bar{E}$, almost surely. In particular, $\frac{1}{N} \sum_i \|s_{q_\theta}(X^{(i)})\|$ is uniformly bounded on E, almost surely. The same argument holds for $\frac{1}{N} \sum_i \|\nabla_\theta s_{q_\theta}(X^{(i)})\|$ using $g_{1,\bar{E}}(x)$. Therefore, by Equation E.45, it follows that $\| \frac{1}{N(N-1)} \sum_{i \neq j} \nabla_\theta^3 u_\theta(X^{(i)}, X^{(j)}) \|$ is uniformly bounded on E. Since k is positive by assumption, $\frac{1}{N(N-1)} \sum_{i \neq j} k(X^{(i)}, X^{(j)}) > 0$ for all $N \geq 2$ and by Equations E.40 and E.41, $\frac{1}{N(N-1)} \sum_{i \neq j} k(X^{(i)}, X^{(j)})$ converges a.s. to a finite quantity greater than 0. We conclude that

almost surely,

$$\|f_N'''(\theta)\| = \frac{1}{T} \frac{\|\frac{1}{N(N-1)} \sum_{i \neq j} \nabla_\theta^3 u_\theta(X^{(i)}, X^{(j)})\|}{\frac{1}{N(N-1)} \sum_{i \neq j} k(X^{(i)}, X^{(j)})}$$

is uniformly bounded on E, for $N \in \{2, 3, \ldots\}$. Recall that for $N = 1$, $f_N(\theta) = 0$ by definition. Therefore, almost surely, $(f_N''' : N \in \mathbb{N})$ is uniformly bounded on E. □

Proof of Theorem 5.6.9. We show that the conditions of Theorem 3.2 of Miller [176] are met, from which the conclusions of this theorem follow immediately.

By Condition 5.6.10 and Equation 5.35, f_N has continuous third-order partial derivatives on Θ. Let E be the set from Condition 5.6.10. With probability 1, $f_N \to f$ uniformly on E (by Proposition E.3.2 with $C = \bar{E}$) and (f_N''') is uniformly bounded on E (by Proposition E.3.3). Note that f is finite on Θ by Proposition 5.6.3. Thus, by Theorem 3.4 of Miller [176], f' and f'' exist on E and $f_N'' \to f''$ uniformly on E with probability 1. Since θ_* is a minimizer of f and $\theta_* \in E$, we know that $f'(\theta_*) = 0$ and $f''(\theta_*)$ is positive semidefinite; thus, $f''(\theta_*)$ is positive definite since it is invertible by assumption.

Case (a): Now, consider the case where Θ is compact. Then almost surely, $f_N \to f$ uniformly on Θ by Proposition E.3.2 with $C = \Theta$. Since θ_* is a unique minimizer of f, we have $f(\theta) > f(\theta_*)$ for all $\theta \in \Theta \setminus \{\theta_*\}$. Let $H \subseteq E$ be an open set such that $\theta_* \in H$ and $\bar{H} \subseteq E$. We show that $\liminf_N \inf_{\theta \in \Theta \setminus \bar{H}} f_N(\theta) > f(\theta_*)$. Since $\Theta \setminus H$ is compact,

$$\inf_{\theta \in \Theta \setminus \bar{H}} f(\theta) - f(\theta_*) =: \epsilon > 0.$$

By uniform convergence, with probability 1, there exists N such that for all $N' > N$, $\sup_{\theta \in \Theta} |f_{N'}(\theta) - f(\theta)| \leq \epsilon/2$, and thus,

$$\inf_{\theta \in \Theta \setminus \bar{H}} f_{N'}(\theta) \geq \inf_{\theta \in \Theta \setminus \bar{H}} f(\theta) - \epsilon/2 = f(\theta_*) + \epsilon/2.$$

Hence, $\liminf_N \inf_{\theta \in \Theta \setminus \bar{H}} f_N(\theta) > f(\theta_*)$ almost surely. Applying Theorem 3.2 of Miller [176], the conclusion of the theorem follows. Note that $f''_N(\theta_N) \to f''(\theta_*)$ a.s. since $\theta_N \to \theta_*$ and $f''_N \to f''$ uniformly on E.

Case (b): Alternatively, consider the case where Θ is open and f_N is convex on Θ. For all $\theta \in \Theta$, with probability 1, $f_N(\theta) \to f(\theta)$ (by Proposition E.3.2 with $C = \{\theta\}$). However, we need to show that with probability 1, for all $\theta \in \Theta$, $f_N(\theta) \to f(\theta)$. We follow the argument in the proof of Theorem 6.3 of Miller [176]. Let W be a countable dense subset of Θ. Since W is countable, with probability 1, for all $\theta \in W$, $f_N(\theta) \to f(\theta)$. Since f_N is convex, then with probability 1, for all $\theta \in \Theta$, the limit $\tilde{f}(\theta) := \lim_N f_N(\theta)$ exists and is finite, and $\tilde{f}$ is convex (Theorem 10.8 of Rockafellar [219]). Since f_N is convex and $f(\theta)$ is finite, $f(\theta)$ is also convex. Since f and $\tilde{f}$ are convex, they are also continuous (Theorem 10.1 of Rockafellar [219]). Continuous functions that agree on a dense subset of points must be equal. Thus, with probability 1, for all $\theta \in \Theta$, $f_N(\theta) \to f(\theta)$. Applying Theorem 3.2 of Miller [176], the conclusion of the theorem follows. □

Proof of Theorem 5.6.11. Our proof builds on Appendix D.3 of Barp et al. [17], which establishes a central limit theorem for the KSD when the model is an exponential family. The outline of the proof is as follows. First, we establish bounds on s_{q_θ} and its derivatives, using the assumed bounds

on $\nabla_x t(x)$ and $\nabla_x \log \lambda(x)$. Second, we establish that $f''(\theta)$ is positive definite and independent of θ, and that $f''_N(\theta)$ converges to it almost surely; from this, we conclude that $f''(\theta_*)$ is invertible and $f_N(\theta)$ is convex. These results rely on the convergence properties of U-statistics and on Sylvester's criterion.

The assumption that $\log \lambda(x)$ is continuously differentiable on $\mathcal{X}$ implies that $\lambda(x) > 0$ for $x \in \mathcal{X}$. Since $q_\theta(x) = \lambda(x) \exp(\theta^\top t(x) - \kappa(\theta))$, we have

$$\begin{aligned} s_{q_\theta}(x) &= \nabla_x \log \lambda(x) + (\nabla_x t(x))^\top \theta \\ \nabla_\theta s_{q_\theta}(x) &= (\nabla_x t(x))^\top \in \mathbb{R}^{d \times m} \\ \nabla^2_\theta s_{q_\theta}(x) &= 0 \in \mathbb{R}^{d \times m \times m} \end{aligned}$$

where $(\nabla_x t(x))_{ij} = \partial t_i / \partial x_j$. Thus, $s_{q_\theta}(x)$ has continuous third-order partial derivatives with respect to θ, and Equations 5.41 and 5.42 are trivially satisfied. Equation 5.40 holds for all compact $C \subseteq \Theta$ since $\|\nabla_x \log \lambda(x)\|$ and $\|\nabla_x t(x)\|$ are continuous functions in $L^1(P_0)$ and

$$\begin{aligned} \|s_{q_\theta}(x)\| &= \|\nabla_x \log \lambda(x) + (\nabla_x t(x))^\top \theta\| \leq \|\nabla_x \log \lambda(x)\| + \|\nabla_x t(x)\| \|\theta\|, \\ \|\nabla_\theta s_{q_\theta}(x)\| &= \|\nabla_x t(x)\|. \end{aligned}$$

Hence, Condition 5.6.10 holds. By Equation 5.36 and Proposition 5.6.3,

$$f(\theta) = \frac{1}{T} \textsc{nksd}(p_0(x) \| q(x|\theta)) = \frac{1}{TK} \int_{\mathcal{X}} \int_{\mathcal{X}} u_\theta(x, y) P_0(dx) P_0(dy) \tag{E.46}$$

where $K := \int\int k(x,y)P_0(dx)P_0(dy)$. By Equation E.3,

$$u_\theta(x,y) = \theta^\top B_2(x,y)\theta + B_1(x,y)^\top\theta + B_0(x,y) \tag{E.47}$$

where

$$\begin{aligned}
B_2(x,y) &= (\nabla_x t(x))(\nabla_y t(y))^\top k(x,y),\\
B_1(x,y) &= (\nabla_y t(y))(\nabla_x \log\lambda(x))k(x,y) + (\nabla_x t(x))(\nabla_y \log\lambda(y))k(x,y)\\
&\quad + (\nabla_y t(y))(\nabla_x k(x,y)) + (\nabla_x t(x))(\nabla_y k(x,y)),\\
B_0(x,y) &= (\nabla_x \log\lambda(x))^\top(\nabla_y \log\lambda(y))k(x,y) + (\nabla_y \log\lambda(y))^\top(\nabla_x k(x,y))\\
&\quad + (\nabla_x \log\lambda(x))^\top(\nabla_y k(x,y)) + \text{trace}(\nabla_x\nabla_y^\top k(x,y)).
\end{aligned}$$

By Condition 5.6.7, $|k(x,y)|$, $\|\nabla_x k(x,y)\|$, and $\|\nabla_x\nabla_y^\top k(x,y)\|$ are bounded by a constant, say, $B < \infty$. Thus, it is straightforward to check that B_2, B_1, and B_0 belong to $L^1(P_0 \times P_0)$ since $\|\nabla_x t(x)\|$ and $\|\nabla_x \log\lambda(x)\|$ are in $L^1(P_0)$. Further, $0 < K < \infty$ since $0 < k(x,y) \le B < \infty$ by assumption. Thus,

$$f(\theta) = \frac{1}{TK}\int\int (\theta^\top B_2(x,y)\theta + B_1(x,y)^\top\theta + B_0(x,y))P_0(dx)P_0(dy) \in \mathbb{R}.$$

Since k is symmetric, $B_2(x,y)^\top = B_2(y,x)$. Hence, $\nabla_\theta(\theta^\top B_2(x,y)\theta) = (B_2(x,y) +$

$B_2(y,x))\theta$, so by Fubini's theorem,

$$f'(\theta) = \frac{1}{TK}\int\int (2B_2(x,y)\theta + B_1(x,y))P_0(dx)P_0(dy) \in \mathbb{R}^m,$$
$$f''(\theta) = \frac{2}{TK}\int\int B_2(x,y)P_0(dx)P_0(dy) \in \mathbb{R}^{m\times m}.$$

Here, differentiating under the integral sign is justified simply by linearity of the expectation. Note that $f''(\theta)$ is a symmetric matrix since $B_2(x,y)^\top = B_2(y,x)$. Next, to show $f''(\theta)$ is positive definite, let $v \in \mathbb{R}^m \setminus \{0\}$. By assumption, the rows of $\nabla_x t(x)$ are linearly independent with positive probability under P_0. Thus, there is a set $E \subseteq \mathcal{X}$ such that $P_0(E) > 0$ and $(\nabla_x t(x))^\top v \neq 0$ for all $x \in E$. Define $g(x) = (\nabla_x t(x))^\top v\, p_0(x) \in \mathbb{R}^d$. Then $\int_{\mathcal{X}} |g_i(x)|dx > 0$ for at least one i, and $\int_{\mathcal{X}} |g_i(x)|dx \leq \|v\| \int_{\mathcal{X}} \|\nabla_x t(x)\| p_0(x)dx < \infty$ for all i. Thus,

$$v^\top f''(\theta)v = \frac{2}{TK}\int\int g(x)^\top g(y)k(x,y)dxdy = \frac{2}{TK}\sum_{i=1}^{d}\int\int g_i(x)g_i(y)k(x,y)dxdy > 0$$

since k is integrally strictly positive definite. Therefore, $f''(\theta)$ is positive definite. In particular, $f''(\theta_*)$ is invertible.

Finally, we show that with probability 1, for all N sufficiently large, $f_N(\theta)$ is convex. By Equations 5.35 and E.47,

$$f_N(\theta) = \frac{1}{T}\frac{\sum_{i\neq j}\left[\theta^\top B_2(X^{(i)},X^{(j)})\theta + B_1(X^{(i)},X^{(j)})^\top\theta + B_0(X^{(i)},X^{(j)})\right]}{\sum_{i\neq j} k(X^{(i)},X^{(j)})}.$$

Thus,

$$f''_N(\theta) = \frac{2}{T} \frac{\sum_{i \neq j} B_2(X^{(i)}, X^{(j)})}{\sum_{i \neq j} k(X^{(i)}, X^{(j)})}.$$

By the strong law of large numbers for U-statistics (Theorem 5.4A of Serfling [231]), we have $f''_N(\theta) \to f''(\theta)$ almost surely, since $\int_{\mathcal{X}} \int_{\mathcal{X}} \|B_2(x, y)\| P_0(dx) P_0(dy) < \infty$ and $0 < K < \infty$. For a symmetric matrix A, let $\lambda_*(A)$ denote the smallest eigenvalue. Since $\lambda_*(A)$ is a continuous function of the entries of A, we have $\lambda_*(f''_N(\theta)) \to \lambda_*(f''(\theta))$ a.s. as $N \to \infty$. Thus, with probability 1, for all N sufficiently large, $f''_N(\theta)$ is positive definite, and hence, f_N is convex. Further, for such N, since f_N is a quadratic function with positive definite Hessian, we have $M_N := \inf_{\theta \in \Theta} f_N(\theta) > -\infty$ and $z_N = \int_\Theta \exp(-N f_N(\theta)) \pi(\theta) d\theta \leq \exp(-N M_N) < \infty$. □

E.3.3 Proof of Theorem 5.6.12

To establish Theorem 5.6.12, we use the properties of U-statistics described in Chapter 5.5 of Serfling [231]. When the data distribution matches the model distribution, $\widehat{\text{NKSD}}$ converges more quickly than when it does not match; this same property was used by Liu et al. [158] to develop a goodness-of-fit test based on the KSD.

Proof. We first study the asymptotics of $f'_N(\theta_*)$. Denoting $\nabla_\theta|_{\theta=\theta_*} u_\theta$ by $\nabla_\theta u_{\theta_*}$ for brevity,

$$f'_N(\theta_*) = \frac{1}{T} \frac{\frac{1}{N(N-1)} \sum_{i \neq j} \nabla_\theta u_{\theta_*}(X^{(i)}, X^{(j)})}{\frac{1}{N(N-1)} \sum_{i \neq j} k(X^{(i)}, X^{(j)})}.$$

The denominator converges a.s. to a finite positive constant, as in the proof of Proposition E.3.2. It is straightforward to verify that $\mathbb{E}_{X,Y\sim P_0}[\|\nabla_\theta u_{\theta_*}(X,Y)\|^2] < \infty$ since $s_{q_{\theta_*}}$ and $\nabla_\theta\big|_{\theta=\theta_*} s_{q_\theta}$ are in $L^2(P_0)$ by assumption. By Theorems 5.5.1A and 5.5.2 of Serfling[231],

$$\frac{1}{N(N-1)}\sum_{i\neq j}\nabla_\theta u_{\theta_*}(X^{(i)},X^{(j)}) - \mathbb{E}_{X,Y\sim P_0}[\nabla_\theta u_{\theta_*}(X,Y)] = O_{P_0}(N^{-1/2}).$$

Further, by the Leibniz integral rule[78] Theorem 2.27,

$$\mathbb{E}_{X,Y\sim P_0}[\nabla_\theta u_{\theta_*}(X,Y)] = \nabla_\theta\big|_{\theta=\theta_*}\mathbb{E}_{X,Y\sim P_0}[u_\theta(X,Y)] = T\,\mathbb{E}_{X,Y\sim P_0}[k(X,Y)]f'(\theta_*) = 0,$$

using the fact that $f'(\theta_*) = 0$ since θ_* is a minimizer of f. Thus,

$$f'_N(\theta_*) = O_{P_0}(N^{-1/2}). \tag{E.48}$$

Next, we examine the convergence of θ_N to θ_*. For all N sufficiently large, $f'_N(\theta_N) = 0$ by Theorem 5.6.9 (part 1), and thus, by Taylor's theorem,

$$0 = f'_N(\theta_N) = f'_N(\theta_*) + f''_N(\theta_N^+)(\theta_N - \theta_*),$$

where θ_N^+ is on the line between θ_N and θ_*. As in the proof of Theorem 5.6.9, $f''_N \to f''$ uniformly

on the set E defined in Condition 5.6.10. Thus, since f_N'' is continuous on E and $\theta_N^+ \to \theta_*$,

$$f_N''(\theta_N^+) \xrightarrow[N\to\infty]{\text{a.s.}} f''(\theta_*). \tag{E.49}$$

In particular, $f_N''(\theta_N^+)$ is invertible for all N sufficiently large, since $f''(\theta_*)$ is invertible by assumption. Hence,

$$\theta_N - \theta_* = -f_N''(\theta_N^+)^{-1} f_N'(\theta_*), \tag{E.50}$$

and therefore, by Equation E.48,

$$\|\theta_N - \theta_*\| \leq \|f_N''(\theta_N^+)^{-1}\| \|f_N'(\theta_*)\| = O_{P_0}(N^{-1/2}). \tag{E.51}$$

This result matches Theorem 4 in Barp et al. [17]. By Taylor's theorem,

$$\begin{aligned} f_N(\theta_*) - f_N(\theta_N) &= f_N'(\theta_N)^\top(\theta_* - \theta_N) + \frac{1}{2}(\theta_* - \theta_N)^\top f_N''(\theta_N^{++})(\theta_* - \theta_N) \\ &= \frac{1}{2}(\theta_* - \theta_N)^\top f_N''(\theta_N^{++})(\theta_* - \theta_N) \end{aligned}$$

for all N sufficiently large, where θ_N^{++} is on the line between θ_N and θ_*. Therefore, using the same reasoning as for Equations E.49 and E.51,

$$|f_N(\theta_*) - f_N(\theta_N)| \leq \frac{1}{2}\|f_N''(\theta_N^{++})\| \|\theta_* - \theta_N\|^2 = O_{P_0}(N^{-1}). \tag{E.52}$$

This proves the first part of the theorem (Equation 5.43). Next, consider $f_N(\theta_*) - f(\theta_*)$. Recall that

$$f_N(\theta_*) = \frac{1}{T} \frac{\frac{1}{N(N-1)} \sum_{i \neq j} u_{\theta_*}(X^{(i)}, X^{(j)})}{\frac{1}{N(N-1)} \sum_{i \neq j} k(X^{(i)}, X^{(j)})}.$$

It is straightforward to verify that $\mathbb{E}_{X,Y \sim P_0}[|u_{\theta_*}(X,Y)|^2] < \infty$ since $s_{q_{\theta_*}}$ is in $L^2(P_0)$. By Theorems 5.5.1A and 5.5.2 of Serfling [231],

$$\frac{1}{N(N-1)} \sum_{i \neq j} u_{\theta_*}(X^{(i)}, X^{(j)}) - \mathbb{E}_{X,Y \sim P_0}[u_{\theta_*}(X,Y)] = O_{P_0}(N^{-1/2}).$$

Similarly, since k is bounded,

$$\frac{1}{N(N-1)} \sum_{i \neq j} k(X^{(i)}, X^{(j)}) - \mathbb{E}_{X,Y \sim P_0}[k(X,Y)] = O_{P_0}(N^{-1/2}).$$

It is straightforward to check that the second part of the theorem (Equation 5.44) follows.

For the third part, our argument follows that of the proof of Theorem 4.1 of Liu et al. [158]. Suppose $\text{NKSD}(p_0(x) \| q(x|\theta_*)) = 0$, and note that $P_0(x) = Q_{\theta_*}(x)$ by Proposition 5.6.4. Given a differentiable function $g : \mathbb{R}^d \to \mathbb{R}^d$, define $\nabla_x^\top g(x) := \sum_{i=1}^d \partial g_i(x)/\partial x_i$. Then

$$\begin{aligned}
\mathbb{E}_{X \sim P_0}[u_{\theta_*}(X,y)] &= s_{p_0}(y)^\top \int_{\mathcal{X}} \Big((\nabla_x p_0(x)) k(x,y) + p_0(x)(\nabla_x k(x,y)) \Big) dx \\
&\quad + \int_{\mathcal{X}} \Big((\nabla_x p_0(x))^\top \nabla_y k(x,y) + p_0(x)(\nabla_x^\top \nabla_y k(x,y)) \Big) dx \\
&= s_{p_0}(y)^\top \int_{\mathcal{X}} \nabla_x (p_0(x) k(x,y)) dx + \int_{\mathcal{X}} \nabla_x^\top \nabla_y (p_0(x) k(x,y)) dx. \quad \text{(E.53)}
\end{aligned}$$

The first term on the right-hand side of Equation E.53 is zero since, by assumption, k is in the Stein class of P_0 (Condition 5.6.2). The second term is also zero since, by the Leibniz integral rule[78] Theorem 2.27, $\int \nabla_y^\top \nabla_x (p_0(x)k(x,y))dx = \nabla_y^\top \int \nabla_x (p_0(x)k(x,y))dx$, which again equals zero because k is in the Stein class of P_0. Therefore, $\mathbb{E}_{X\sim P_0}[u_{\theta_*}(X,y)] = 0$ for all $y \in \mathcal{X}$, and in particular, the variance of this expression is also zero: $\mathbb{V}_{Y\sim P_0}[\mathbb{E}_{X\sim P_0}[u_{\theta_*}(X,Y)]] = 0$. By Theorem 5.5.2 of Serfling[231], it follows that

$$\frac{1}{N(N-1)} \sum_{i\neq j} u_{\theta_*}(X^{(i)}, X^{(j)}) = O_{P_0}(N^{-1}) \tag{E.54}$$

since $\mathbb{E}_{X,Y\sim P_0}[u_{\theta_*}(X,Y)] = 0$. Although Serfling[231] requires $\mathbb{V}_{X,Y\sim P_0}[u_{\theta_*}(X,Y)] > 0$, Equation E.54 holds trivially if $\mathbb{V}_{X,Y\sim P_0}[u_{\theta_*}(X,Y)] = 0$. As before, since the denominator of $f_N(\theta_*)$ converges a.s. to a finite positive constant, we have that $f_N(\theta_*) = O_{P_0}(N^{-1})$. Equation 5.45 follows since $f(\theta_*) = 0$ when $\text{NKSD}(p_0(x)\|q(x|\theta_*)) = 0$. □

E.3.4 Proof of Theorem 5.6.17

Proof. Applying Theorem 5.6.9 (part 3) to each foreground model $j \in \{1,2\}$, we have

$$\log z_{j,N} + N f_{j,N}(\theta_{j,N}) - \log \pi(\theta_{j,*}) + \log|\det f_j''(\theta_{j,*})|^{1/2} - \frac{1}{2} m_{\mathcal{F}_j,j} \log(2\pi/N) \xrightarrow[N\to\infty]{\text{a.s.}} 0.$$

Since $\mathcal{K}_{j,N} = (2\pi/N)^{m_{\mathcal{B}_j}/2} z_{j,N}$, this implies

$$\log \mathcal{K}_{j,N} + N f_{j,N}(\theta_{j,N}) - \frac{1}{2}(m_{\mathcal{F}_j,j} + m_{\mathcal{B}_j}) \log(2\pi/N) + C_j \xrightarrow[N\to\infty]{\text{a.s.}} 0$$

where C_j is a constant that does not depend on N. Hence,

$$\begin{aligned} \log \frac{\mathcal{K}_{1,N}}{\mathcal{K}_{2,N}} &+ N(f_{1,N}(\theta_{1,N}) - f_{2,N}(\theta_{2,N})) \\ &- \frac{1}{2}(m_{\mathcal{F}_1,1} + m_{\mathcal{B}_1} - m_{\mathcal{F}_2,2} - m_{\mathcal{B}_2}) \log(2\pi/N) + C_1 - C_2 \xrightarrow[N\to\infty]{\text{a.s.}} 0. \end{aligned} \tag{E.55}$$

By Theorem 5.6.12, $f_{j,N}(\theta_{j,N}) \xrightarrow{P_0} f_j(\theta_{j,*})$, and therefore,

$$\frac{1}{N} \log \frac{\mathcal{K}_{1,N}}{\mathcal{K}_{2,N}} + f_1(\theta_{1,*}) - f_2(\theta_{2,*}) \xrightarrow[N\to\infty]{P_0} 0.$$

Plugging in the definition of f_j (Equation 5.36), this proves part 1 of the theorem.

For part 2, suppose $f_1(\theta_{1,*}) = f_2(\theta_{2,*}) = 0$ and $m_{\mathcal{B}_2} - m_{\mathcal{B}_1}$ does not depend on N. Then by Theorem 5.6.12, $f_{j,N}(\theta_{j,N}) = O_{P_0}(N^{-1})$. Using this in Equation E.55, we have

$$\frac{1}{\log N} \log \frac{\mathcal{K}_{1,N}}{\mathcal{K}_{2,N}} + \frac{1}{2}(m_{\mathcal{F}_1,1} + m_{\mathcal{B}_1} - m_{F_2,2} - m_{\mathcal{B}_2}) \xrightarrow[N\to\infty]{P_0} 0. \tag{E.56}$$

For part 3, suppose $f_1(\theta_{1,*}) = f_2(\theta_{2,*})$ and $m_{\mathcal{B}_j} = c_{\mathcal{B}_j}\sqrt{N}$. Then by Theorem 5.6.12,

$f_{j,N}(\theta_{j,N}) = f_j(\theta_{j,*}) + O_{P_0}(N^{-1/2})$. Using this in Equation E.55, we have

$$\frac{1}{\sqrt{N}\log N}\log\frac{\mathcal{K}_{1,N}}{\mathcal{K}_{2,N}} + \frac{1}{2}(c_{\mathcal{B}_1} - c_{\mathcal{B}_2}) \xrightarrow[N\to\infty]{P_0} 0. \tag{E.57}$$

□

E.4 Additional probabilistic PCA details

E.4.1 Optimizing the NKSD

Computing the Laplace or BIC approximation to the SVC requires finding the minimizer of $\widehat{\text{NKSD}}(p_0(x)\|q(x|\theta))$ with respect to θ. In this section, we describe how components of the NKSD can be pre-computed to speed up this optimization process. The generative model used for pPCA can be rewritten using the properties of multivariate normal distributions as

$$X \sim \mathcal{N}(0, HH^\top + vI_d). \tag{E.58}$$

The Stein score function for the pPCA model is then

$$s_{q_\theta}(x) = \nabla_x \log q(x|H, v) = -(HH^\top + vI_d)^{-1}x.$$

Define the matrices

$$K_{ij} := \mathbb{I}(i \neq j)\, k(X^{(i)}, X^{(j)}),$$
$$\dot{K}_{jb} := \sum_{i=1}^{N} \mathbb{I}(i \neq j)\, \frac{\partial k}{\partial x_b}(X^{(i)}, X^{(j)}),$$

where $\mathbb{I}(E)$ is the indicator function, which equals 1 when E is true and is 0 otherwise. Define the scalars

$$\bar{K} := \sum_{i,j=1}^{N} K_{ij},$$
$$\ddot{K} := \sum_{i,j=1}^{N} \sum_{b=1}^{d} \mathbb{I}(i \neq j)\, \frac{\partial^2 k}{\partial x_b \partial y_b}(X^{(i)}, X^{(j)}).$$

Letting $X \in \mathbb{R}^{N \times d}$ be the data matrix, the NKSD can be written as

$$\widehat{\text{NKSD}}(p_0(x) \| q(x|H, v)) = \frac{1}{\bar{K}} \big[\text{trace}(X^\top K X (HH^\top + vI_d)^{-1} (HH^\top + vI_d)^{-1}) - 2\ \text{trace}(X^\top \dot{K} (HH^\top + vI_d)^{-1}) + \ddot{K} \big],$$

where we have used the fact that the kernel is symmetric. The terms $X^\top K X$ and $X^\top \dot{K}$ are the only ones that include sums over the entire dataset; these can be pre-computed, before optimizing the parameters H and v.

To compute the matrix inversion $(HH^\top + vI_d)^{-1}$ we follow the strategy of Minka [179],

$$\begin{aligned}(HH^\top + vI_d)^{-1} - v^{-1}I_d &= (HH^\top + vI_d)^{-1}(I_d - v^{-1}(HH^\top + vI_d)) \\ &= -(HH^\top + vI_d)^{-1}HH^\top v^{-1} \\ &= -(U(L - vI_k)U^\top + vI_d)^{-1}U(L - vI_k)U^\top v^{-1}.\end{aligned}$$

Thus, applying the Woodbury matrix identity and using $I_d U = U = UI_kI_k = UI_kU^\top U$,

$$\begin{aligned}(HH^\top + vI_d)^{-1} - v^{-1}I_d &= -\big[v^{-1}I_d - v^{-2}U\big((L - vI_k)^{-1} + v^{-1}\big)^{-1}U^\top\big]U(L - vI_k)U^\top v^{-1} \\ &= -U[v^{-1}I_k - v^{-2}((L - v)^{-1} + v^{-1})^{-1}](L - vI_k)U^\top v^{-1} \\ &= -UL^{-1}(L - vI_k)U^\top v^{-1} \\ &= U(L^{-1} - v^{-1}I_k)U^\top.\end{aligned}$$

Therefore,

$$(HH^\top + vI_d)^{-1} = U(L^{-1} - v^{-1}I_k)U^\top + v^{-1}I_d.$$

Computing L^{-1} is trivial since the matrix is diagonal. Returning to the NKSD we have

$$
\begin{aligned}
&\widehat{\text{NKSD}}(p_0(x)\|q(x|U,L,v)) \\
&\quad= \frac{1}{\bar{K}}\Big[\text{trace}\left(X^\top K X[U(L^{-1}-v^{-1}I_k)^2U^\top + 2v^{-1}U(L^{-1}-v^{-1}I_k)U^\top + v^{-2}I_d]\right) \\
&\qquad\quad - 2\ \text{trace}\left(X^\top \dot{K}[U(L^{-1}-v^{-1}I_k)U^\top + v^{-1}I_d]\right) + \ddot{K}\Big] \\
&\quad= \frac{1}{\bar{K}}\Big[\text{trace}\left(U^\top X^\top K X U(L^{-1}-v^{-1}I_k)^2\right) \\
&\qquad\quad + \text{trace}\left(U^\top[2v^{-1}X^\top K X - 2X^\top \dot{K}]U(L^{-1}-v^{-1}I_k)\right) \\
&\qquad\quad + v^{-1}\ \text{trace}\left(v^{-1}X^\top K X - 2X^\top \dot{K}\right) + \ddot{K}\Big].
\end{aligned}
$$

We optimized U, L and v using the trust region method implemented in pymanopt[262].

E.4.2 Data selection with the SVC

We used the approximate optimum technique in Section 5.2.3 to estimate the SVC for different foreground subspaces. Following Section E.1.2, we used the factored IMQ kernel with $\beta = -0.5$ and $c = 1$.

We focused on foreground subspaces that correspond to subsets of the data dimensions. More specifically, recall that $X_{\mathcal{F}} = V^\top X$; then, we impose the restriction that each column of V is a standard basis vector $e^{(b)} \in \mathbb{R}^d$, where $e^{(b)}_b = 1$ and $e^{(b)}_{b'} = 0$ for $b' \neq b$. A subspace $\mathcal{X}_{\mathcal{F}}$ is then characterized by the set of included dimensions $S_{\mathcal{F}} \subseteq \{1, \ldots, d\}$. The marginal distribution of the model $q(x_{\mathcal{F}}|H, v)$ is now straightforward to compute based on Equation E.58 and the properties

of multivariate normals:

$$X_{\mathcal{F}} \sim \mathcal{N}(0, H_{S_{\mathcal{F}}} H_{S_{\mathcal{F}}}^{\top} + vI_{|S_{\mathcal{F}}|})$$

where $H_{S_{\mathcal{F}}}$ is the submatrix consisting of rows of H indexed by $S_{\mathcal{F}}$, and $|S_{\mathcal{F}}|$ is the size of the set $S_{\mathcal{F}}$.

In the projected model, some of the parameters are nuisance variables with no contribution to the likelihood. Since the dimension of a $d \times k$ matrix on the Stiefel manifold is $dk - k(k+1)/2$, the total dimension of the foreground model (including contributions from parameters U, L and v) is $m_{\mathcal{F}} = |S_{\mathcal{F}}|k - k(k+1)/2 + k + 1$, assuming $|S_{\mathcal{F}}| \geq k$.

Code is available at https://github.com/EWeinstein/data-selection.

E.4.3 Calibration

The T hyperparameter was calibrated as in Section E.1.1. In detail, we sampled 10 independent true parameter values from the prior, with $\alpha = 1$ and $d = 6$. (We used a slightly less disperse prior than during inference, where we set $\alpha = 0.1$, to avoid numerical instabilities in the $\hat{T}$ estimate.) Then, for each of the true parameter values, we simulated $N = 2000$ datapoints. For each simulated true parameter value, we tracked the trend in the $\hat{T}$ estimator (Equation E.1) with increasing N (Figure E.2). The median estimated T value at $N = 2000$ was 0.052 across the 10 runs.

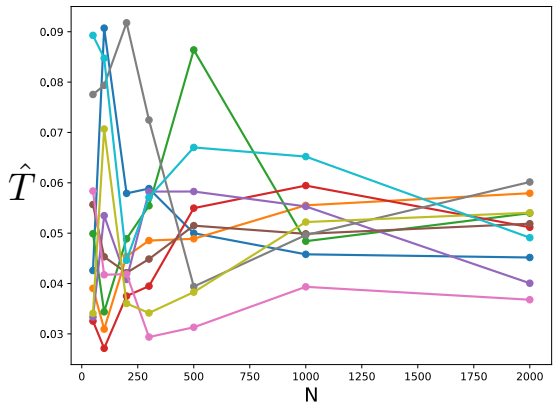


Figure E.2: Estimated T for increasing number of data samples, for 10 independent parameter samples from the prior. The median value at $N = 2000$ is $\hat{T} = 0.052$.

E.4.4 Pólya tree model

In this section, we describe the Pólya tree model[76,171,151] following the construction of Berger & Guglielmi[21]. Let $\underline{\epsilon}_n := (\epsilon_1, \ldots, \epsilon_n)$ denote a vector of length n, where each $\epsilon_j \in \{0, 1\}$. Each $\underline{\epsilon}_n$ vector indexes an interval in $\mathbb{R}$, given by

$$B_{\underline{\epsilon}_n} := \left[\tilde{F}^{-1}\left(\textstyle\sum_{j=1}^{n} \epsilon_j/2^j\right),\ \tilde{F}^{-1}\left(\textstyle\sum_{j=1}^{n} \epsilon_j/2^j + 1/2^n\right)\right],$$

where $\tilde{F}^{-1}$ is the inverse c.d.f. of some probability distribution. For all $n \in \{0, 1, 2, \ldots\}$ and all $\underline{\epsilon}_n \in \{0, 1\}^n$, let

$$Y_{\underline{\epsilon}_n} \sim \text{Beta}(\xi_{\underline{\epsilon}_n 0}, \xi_{\underline{\epsilon}_n 1}),$$

where the ξ's are hyperparameters. We say that a random variable $X \in \mathbb{R}$ is distributed according to a Pólya tree model if

$$P(X \in B_{\underline{\epsilon}_n}) = \prod_{j=1}^{n} (Y_{\underline{\epsilon}_{j-1}})^{\mathbb{I}(\epsilon_j = 0)} (1 - Y_{\underline{\epsilon}_{j-1}})^{\mathbb{I}(\epsilon_j = 1)},$$

where $\mathbb{I}(E)$ is the indicator function, which equals 1 when E is true and is 0 otherwise. We follow Berger & Guglielmi[21] and use

$$\begin{aligned}
\mu(B_{\underline{\epsilon}_n}) &:= F(\tilde{F}^{-1}(\textstyle\sum_{j=1}^{n} \epsilon_j/2^j + 1/2^n)) - F(\tilde{F}^{-1}(\textstyle\sum_{j=1}^{n} \epsilon_j/2^j)), \\
\rho(\underline{\epsilon}_n) &:= \frac{1}{\eta} \left(\frac{f(\tilde{F}^{-1}(\sum_{j=1}^{n} \epsilon_j/2^j + 1/2^{n+1}))}{\mu(B_{\underline{\epsilon}_n})} \right)^2, \\
\xi_{\underline{\epsilon}_n 0} &:= \rho(\underline{\epsilon}_n) \sqrt{\frac{\mu(B_{\underline{\epsilon}_n 0})}{\mu(B_{\underline{\epsilon}_n 1})}}, \\
\xi_{\underline{\epsilon}_n 1} &:= \rho(\underline{\epsilon}_n) \sqrt{\frac{\mu(B_{\underline{\epsilon}_n 1})}{\mu(B_{\underline{\epsilon}_n 0})}},
\end{aligned}$$

where F and f are the c.d.f. and p.d.f. respectively of some probability distribution, and $\eta > 0$ is a scale hyperparameter. We denote this complete model as $X \sim \text{PolyaTree}(F, \tilde{F}, \eta)$.

E.4.5 Datasets and preprocessing

We downloaded two publicly available datasets. The first dataset was taken from human peripheral blood mononuclear cells (PBMCs):

https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc3k. This is a standard dataset used in the tutorials for Seurat[249] and Scanpy[291], for exam-

ple. The second was taken from a dissociated extranodal marginal zone B-cell tumor, specifically a mucosa-associated lymphoid tissue (MALT) tumor: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/malt_10k_protein_v3.

We pre-processed the data using Scprep[90], following its example: we normalized the total expression of each cell to match the median total expression in the dataset, to account for variability in library size, and then square-root transformed the resulting normalized counts.

E.5 Additional glass model details

E.5.1 Glass model inference

We place a standard normal prior on each entry of H_j and a Laplace prior on each entry of $J_{jj'}$ with scale 0.1 to encourage sparsity. To enforce that $\mu \geq 0$ (since scRNAseq counts are nonnegative) and $\tau > 0$, we place priors on a transformed version of these parameters, as follows:

$$\tilde{\mu} \sim \mathcal{N}(0, 1)$$

$$\mu = \log(1 + \exp(\tilde{\mu}))$$

$$\tilde{\tau} \sim \mathcal{N}(0, 1)$$

$$\tau = \log(1 + \exp(\tilde{\tau})) + 1.$$

For posterior inference, we employ a mean-field variational approximation: independent normal distributions for the entries of H_j, normal distributions for $\tilde{\mu}$ and $\tilde{\tau}$, and Laplace distributions for each entry of $J_{jj'}$. We use the factored IMQ kernel for the NKSD, with $\beta = -0.5$ and $c = 1$.

To optimize the variational approximation (Equation 5.14), we construct stochastic estimates of its gradient. At each optimization step, the expectation $\mathbb{E}_{r_\zeta}\left[\widehat{\text{NKSD}}(p_0(x_\mathcal{F})\|q(x_\mathcal{F}|\theta))\right]$ is estimated using a minibatch of 200 randomly selected datapoints and a single sample from the variational approximation r_ζ. The rest of the variational inference algorithm follows standard practice in stochastic variational inference, as implemented in Pyro: automatic differentiation to compute gradients, reparameterization estimators for Monte Carlo expectations over the variational distribution, and the Adam optimizer[138,23].

We also used stochastic optimization to perform data selection, as follows. Let $I = (I_1, \ldots, I_d)^\top$ be an indicator variable that specifies for each gene j whether it is included in the foreground subspace ($I_j = 1$) or not ($I_j = 0$). We place a distribution on I such that $I_j \sim \text{Bernoulli}(1/(1 + \exp(-\phi_j)))$ for $j = 1, \ldots, d$ independently. Then, to perform data selection over all possible subsets of genes, we optimize

$$\text{argmax}_\phi \, \mathbb{E}(\mathcal{K}(I) \mid \phi) \tag{E.59}$$

where the expectation is taken with respect to I, where $\mathcal{K}(I)$ is the (estimated) SVC when genes with $I_j = 1$ are included in the foreground space, and $\phi = (\phi_1, \ldots, \phi_d)^\top \in \mathbb{R}^d$ is a vector of log-odds. This stochastic approach to discrete optimization has been used extensively in reinforcement learning and related fields. We use the Leave-One-Out REINFORCE (LOORF) estimator as described in Section 2.1 of Dimitriev & Zhou[54] to estimate gradients of ϕ, using 8 samples per step.

We interleave updates to the variational approximation and to ϕ, using the Adam optimizer with step size 0.01 for each. We ran the procedure with 4 random initial seeds, taking the result with the

largest final estimated SVC. We halt optimization using the stopping rule proposed in Grathwohl et al. [94], stopping when the estimated mean minus the estimated variance of the SVC begins to decrease, based on the average over 2000 steps.

Code is available at https://github.com/EWeinstein/data-selection.

E.5.2 Datasets and preprocessing

In addition to the two datasets in E.4.5, we also explored a dataset of E18 mouse neurons: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/neuron_10k_v3.

We preprocessed each dataset using Scprep [90] in the same way as in Section E.4.5. After preprocessing, we used the top 200 most highly expressed genes from among the top 500 most variable genes, according to the Scprep variability score. We log transform the counts, that is we define $x_{ij} = \log(1 + c_{ij})$ where c_{ij} is the expression count for gene j in cell i.

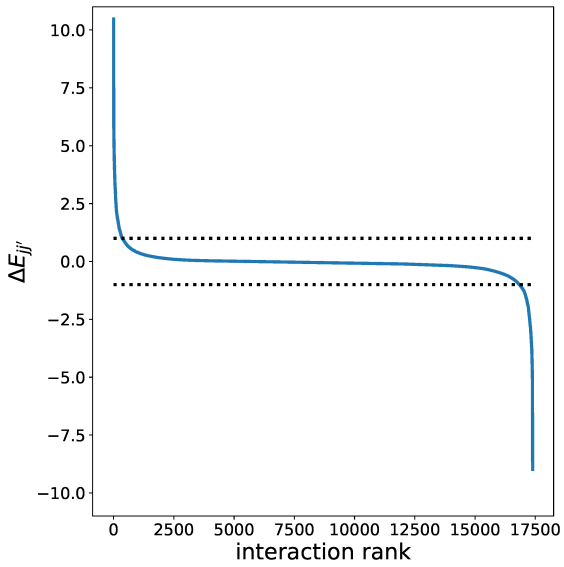


Figure E.3: Posterior mean interaction energies $\Delta E_{jj'}$ for all selected genes, sorted. Dotted lines show the thresholds for strong interactions (set by visual inspection).

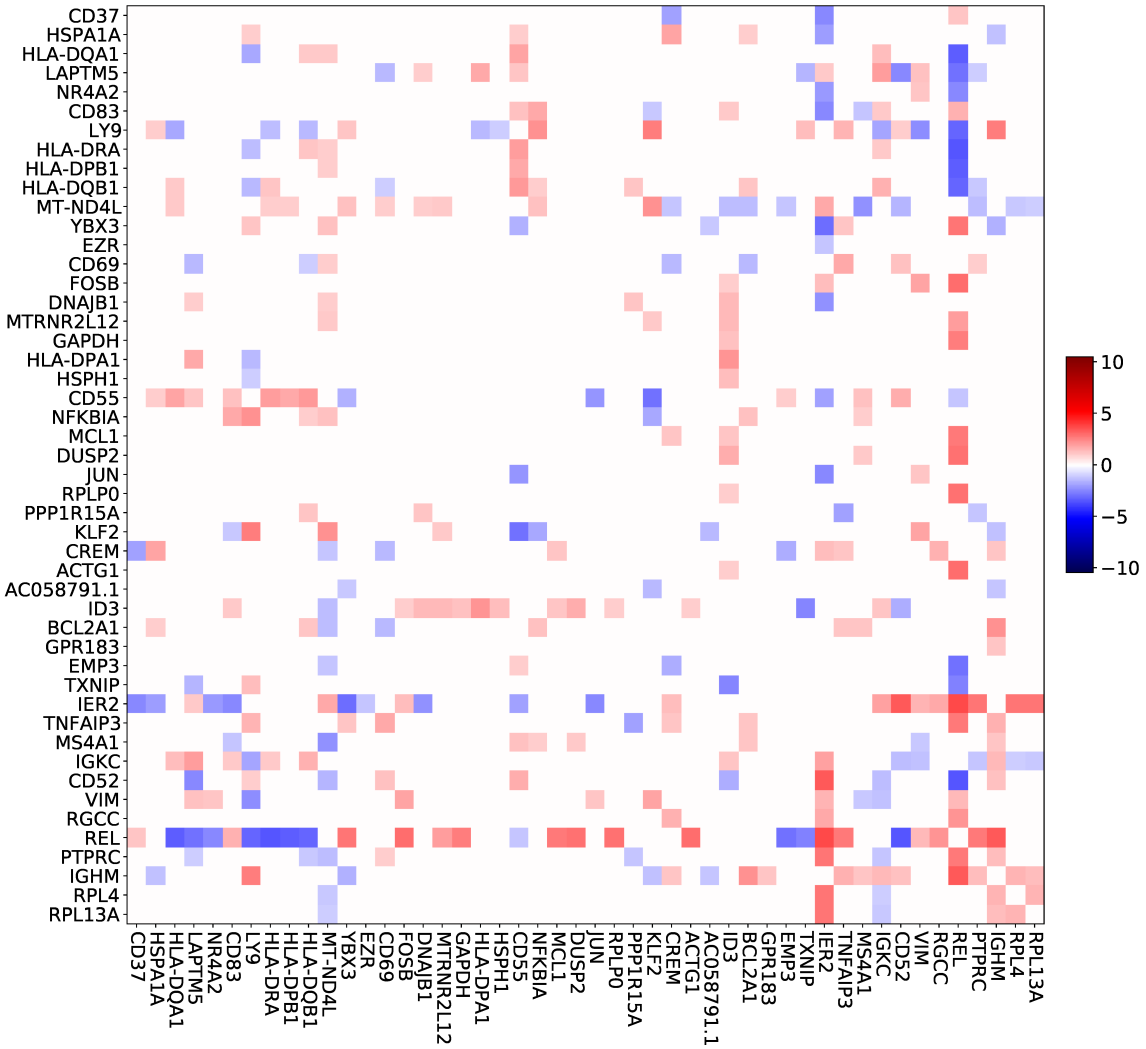


Figure E.4: Posterior mean interaction energies $\Delta E_{jj'}$ for the glass model applied to all 200 genes in the MALT dataset (rather than the selected 187). Genes shown are the same as in Figure 5.8, for visual comparison.

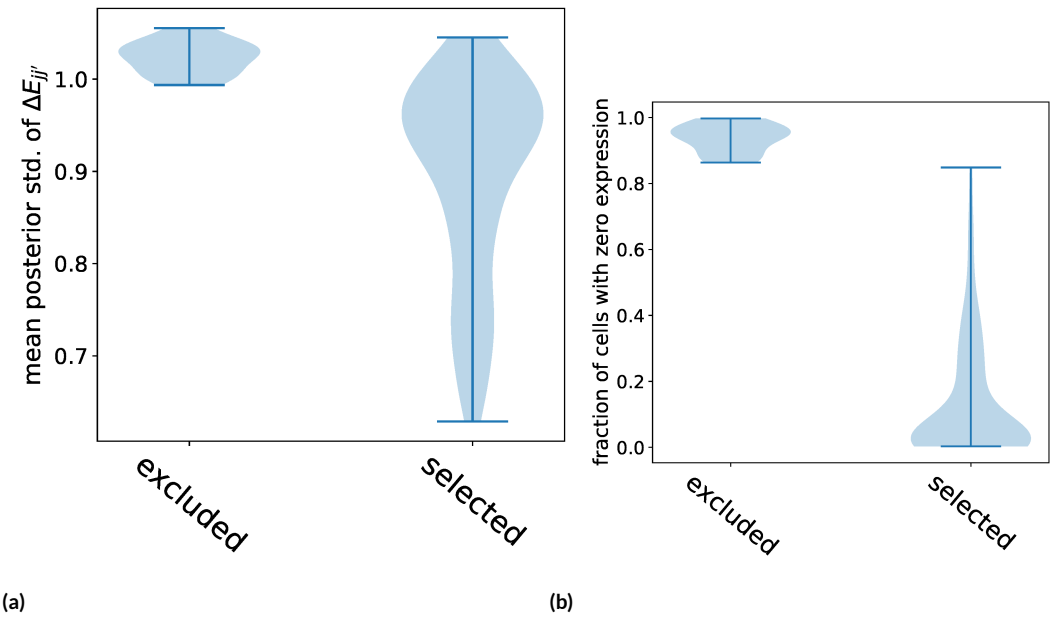


Figure E.5: Comparison of the 187 selected genes and 13 excluded genes using data selection. (a) Violin plot of $\bar{\sigma}_j$ over all excluded and selected genes j, respectively, when applying the model to all 200 genes, where $\bar{\sigma}_j$ is the mean posterior standard deviation of the interaction energies $\Delta E_{jj'}$ for gene j, that is, $\bar{\sigma}_j := \frac{1}{d-1} \sum_{j' \neq j} \mathrm{std}(\Delta E_{jj'} \mid \mathrm{data})$. (b) Violin plot of f_j over all excluded and selected genes j, respectively, where f_j is the fraction of cells with count equal to zero for gene j. The data selection procedure excluded all genes with more than 85% zeros and selected all genes with fewer than 85% zeros.

References

[1] 1000 Genomes Project Consortium, Auton, A., Brooks, L. D., Durbin, R. M., Garrison, E. P., Kang, H. M., Korbel, J. O., Marchini, J. L., McCarthy, S., McVean, G. A., & Abecasis, G. R. (2015). A global reference for human genetic variation. *Nature*, 526(7571), 68–74.

[2] 1001 Genomes Consortium (2016). 1,135 genomes reveal the global pattern of polymorphism in Arabidopsis thaliana. *Cell*, 166(2), 481–491.

[3] 10x Genomics (2019). A new way of exploring immunity - linking highly multiplexed antigen recognition to immune repertoire and phenotype.

[4] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265–283).

[5] Abbas, A. K., Lichtman, A. H., & Pillai, S. (2018). *Cellular and Molecular Immunology*. Elsevier, ninth edition.

[6] Abudayyeh, O. O., Gootenberg, J. S., Konermann, S., Joung, J., Slaymaker, I. M., Cox, D. B. T., Shmakov, S., Makarova, K. S., Semenova, E., Minakhin, L., Severinov, K., Regev, A., Lander, E. S., Koonin, E. V., & Zhang, F. (2016). C2c2 is a single-component programmable RNA-guided RNA-targeting CRISPR effector. *Science*, 353(6299), aaf5573.

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

[8] Alexandrov, L. B. & Stratton, M. R. (2014). Mutational signatures: the patterns of somatic mutations hidden in cancer genomes. *Curr. Opin. Genet. Dev.*, 24, 52–60.

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

[10] Alon, U. (2019). *An Introduction to Systems Biology: Design Principles of Biological Circuits*. CRC Press.

[11] Alsop, E. B. & Raymond, J. (2013). Resolving Prokaryotic Taxonomy without rRNA: Longer Oligonucleotide Word Lengths Improve Genome and Metagenome Taxonomic Classification. *PLoS ONE*, 8(7).

[12] Amin, A. N., Weinstein, E. N., & Marks, D. S. (2021). A generative nonparametric Bayesian model for whole genomes. In *Advances in Neural Information Processing Systems*, volume 34.

[13] Anastasiou, A., Barp, A., Briol, F.-X., Ebner, B., Gaunt, R. E., Ghaderinezhad, F., Gorham, J., Gretton, A., Ley, C., Liu, Q., Mackey, L., Oates, C. J., Reinert, G., & Swan, Y. (2021). Stein's method meets statistics: A review of some recent developments.

[14] Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization.

[15] Banerjee, O., Ghaoui, L. E., & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9(Mar), 485–516.

[16] Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D., Pyshkin, A. V., Sirotkin, A. V., Vyahhi, N., Tesler, G., Alekseyev, M. A., & Pevzner, P. A. (2012). SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5), 455–477.

[17] Barp, A., Briol, F.-X., Duncan, A. B., Girolami, M., & Mackey, L. (2019). Minimum stein discrepancy estimators. In *Advances in Neural Information Processing Systems*.

[18] Barron, A. R. (1989). Uniformly powerful goodness of fit tests. *The Annals of Statistics*, 17(1), 107–124.

[19] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.*, 18(153), 1–43.

[20] Bedbrook, C. N., Rice, A. J., Yang, K. K., Ding, X., Chen, S., LeProust, E. M., Gradinaru, V., & Arnold, F. H. (2017). Structure-guided SCHEMA recombination generates diverse chimeric channelrhodopsins. *Proc. Natl. Acad. Sci. U. S. A.*, 114(13), E2624–E2633.

[21] Berger, J. O. & Guglielmi, A. (2001). Bayesian and conditional frequentist testing of a parametric model versus nonparametric alternatives. *Journal of the American Statistical Association*, 96(453), 174–184.

[22] Bertoin, J. (2010). Exchangeable coalescents. Nachdiplom Lectures.

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

[24] Bissiri, P. G., Holmes, C. C., & Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 78(5), 1103–1130.

[25] Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M., & Church, G. M. (2021). Low-N protein engineering with data-efficient deep learning. *Nature Methods*, 18(4), 389–396.

[26] Blei, D. M. (2014). Build, compute, critique, repeat: Data analysis with latent variable models. *Annu. Rev. Stat. Appl.*, 1(1), 203–232.

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

[28] Boratyn, G. M., Camacho, C., Cooper, P. S., Coulouris, G., Fong, A., Ma, N., Madden, T. L., Matten, W. T., McGinnis, S. D., Merezhuk, Y., Raytselis, Y., Sayers, E. W., Tao, T., Ye, J., & Zaretskaya, I. (2013). BLAST: a more efficient report with usability improvements. *Nucleic Acids Research*, 41(Web Server issue), W29–33.

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

[30] Burke, D. F. & Smith, D. J. (2014). A recommended numbering scheme for influenza A HA subtypes. *PLoS One*, 9(11), e112302.

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

[32] Butler, M. A. & King, A. A. (2004). Phylogenetic comparative analysis: A modeling approach for adaptive evolution. *Am. Nat.*, 164(6), 683–695.

[33] Cappé, O. & Moulines, E. (2008). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society B*, 71, 593–613.

[34] Caticha, A. (2004). Relative entropy and inductive inference. *AIP Conference Proceedings*, 707(1), 75–96.

[35] Caticha, A. (2011). Entropic inference. *AIP Conference Proceedings*, 1305(1), 20–29.

[36] Champredon, D., Li, M., Bolker, B. M., & Dushoff, J. (2018). Two approaches to forecast ebola synthetic epidemics. *Epidemics*, 22, 36–42.

[37] Chen, H., Guo, J., Mishra, S. K., Robson, P., Niranjan, M., & Zheng, J. (2015). Single-cell transcriptional analysis to uncover regulatory circuits driving cell fate decisions in early mouse development. *Bioinformatics*, 31(7), 1060–1066.

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

[39] Chen, S. F. & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4), 359–394.

[40] Chevalier, A., Silva, D.-A., Rocklin, G. J., Hicks, D. R., Vergara, R., Murapa, P., Bernard, S. M., Zhang, L., Lam, K.-H., Yao, G., Bahl, C. D., Miyashita, S.-I., Goreshnik, I., Fuller, J. T., Koday, M. T., Jenkins, C. M., Colvin, T., Carter, L., Bohn, A., Bryan, C. M., Fernández-Velasco, D. A., Stewart, L., Dong, M., Huang, X., Jin, R., Wilson, I. A., Fuller, D. H., & Baker, D. (2017). Massively parallel de novo protein design for targeted therapeutics. *Nature*, 550(7674), 74–79.

[41] Chwialkowski, K., Strathmann, H., & Gretton, A. (2016). A kernel test of goodness of fit. In *International Conference on Machine Learning* (pp. 2606–2615).

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

[43] Colavin, A., Atolia, E., Bitbol, A.-F., & Huang, K. C. (2022). Extracting phylogenetic dimensions of coevolution reveals hidden functional signals. *Sci. Rep.*, 12(1), 820.

[44] Compeau, P. E. C., Pevzner, P. A., & Tesler, G. (2011). How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology*, 29(11), 987–991.

[45] Cortés-Ciriano, I., Lee, J. J.-K., Xi, R., Jain, D., Jung, Y. L., Yang, L., Gordenin, D., Klimczak, L. J., Zhang, C.-Z., Pellman, D. S., PCAWG Structural Variation Working Group, Park, P. J., & PCAWG Consortium (2020). Comprehensive analysis of chromothripsis in 2,658 human cancers using whole-genome sequencing. *Nature Genetics*, 52(3), 331–341.

[46] Csiszar, I. & Matus, F. (2003). Information projections revisited. *IEEE Trans. Inf. Theory*, 49(6), 1474–1490.

[47] Darmanis, S., Sloan, S. A., Croote, D., Mignardi, M., Chernikova, S., Samghababi, P., Zhang, Y., Neff, N., Kowarsky, M., Caneda, C., Li, G., Chang, S. D., Connolly, I. D., Li, Y., Barres, B. A., Gephart, M. H., & Quake, S. R. (2017). Single-Cell RNA-Seq analysis of infiltrating neoplastic cells at the migrating front of human glioblastoma. *Cell Reports*, 21(5), 1399–1410.

[48] David, L. A. & Alm, E. J. (2011). Rapid evolutionary innovation during an archaean genetic expansion. *Nature*, 469(7328), 93–96.

[49] Dawid, A. P. (1984). 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.

[50] Dawid, A. P. (2011). Posterior model probabilities. In P. S. Bandyopadhyay & M. R. Forster (Eds.), *Philosophy of Statistics*, volume 7 (pp. 607–630). Amsterdam: North-Holland.

[51] de Winde, C. M., Veenbergen, S., Young, K. H., Xu-Monette, Z. Y., Wang, X.-X., Xia, Y., Jabbar, K. J., van den Brand, M., van der Schaaf, A., Elfrink, S., van Houdt, I. S., Gijbels, M. J., van de Loo, F. A. J., Bennink, M. B., Hebeda, K. M., Groenen, P. J. T. A., van Krieken, J. H., Figdor, C. G., & van Spriel, A. B. (2016). Tetraspanin CD37 protects against the development of B cell lymphoma. *The Journal of Clinical Investigation*, 126(2), 653–666.

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

[53] Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., & Blei, D. (2017). Variational inference via chi upper bound minimization. In *Advances in Neural Information Processing Systems*.

[54] Dimitriev, A. & Zhou, M. (2021). ARMS: Antithetic-REINFORCE-Multi-Sample gradient for binary variables. In *Proceedings of the 38th International Conference on Machine Learning*.

[55] Ding, C. & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2), 185–205.

[56] Ding, D., Green, A. G., Wang, B., Lite, T.-L. V., Weinstein, E. N., Marks, D. S., & Laub, M. T. (2021). Coevolution of interacting proteins through non-contacting and non-specific mutations.

[57] Ding, X., Zou, Z., & Brooks III, C. L. (2019). Deciphering protein evolution and fitness landscapes with latent space models. *Nat. Commun.*, 10(1), 5644.

[58] Doksum, K. A. & Lo, A. Y. (1990). Consistent and robust Bayes procedures for location based on partial information. *The Annals of Statistics*, 18(1), 443–453.

[59] Donia, M. S., Cimermancic, P., Schulze, C. J., Wieland Brown, L. C., Martin, J., Mitreva, M., Clardy, J., Linington, R. G., & Fischbach, M. A. (2014). A systematic analysis of biosynthetic gene clusters in the human microbiome reveals a common family of antibiotics. *Cell*, 158(6), 1402–1414.

[60] Dragomir, S. S. (1999). Upper and lower bounds for Csiszar f-divergence in terms of the Kullback-Leibler distance and applications.

[61] Drummond, A. J. & Rambaut, A. (2007). BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol. Biol.*, 7, 214.

[62] Dubinkina, V. B., Ischenko, D. S., Ulyantsev, V. I., Tyakht, A. V., & Alexeev, D. G. (2016). Assessment of k-mer spectrum applicability for metagenomic dissimilarity analysis. *BMC Bioinformatics*, 17, 38.

[63] Duc Cao, M., Dix, T. I., Allison, L., & Mears, C. (2007). A simple statistical algorithm for biological sequence compression. In *2007 Data Compression Conference (DCC'07)* (pp. 43–52).

[64] Dudley, R. M. (2002). *Real Analysis and Probability*. Cambridge University Press.

[65] Dunn, S. D., Wahl, L. M., & Gloor, G. B. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3), 333–340.

[66] Dunstan, R. A., Pickard, D., Dougan, S., Goulding, D., Cormie, C., Hardy, J., Li, F., Grinter, R., Harcourt, K., Yu, L., Song, J., Schreiber, F., Choudhary, J., Clare, S., Coulibaly, F., Strugnell, R. A., Dougan, G., & Lithgow, T. (2019). The flagellotropic bacteriophage YSD1 targets salmonella typhi with a chi-like protein tail fibre. *Molecular Microbiology*, 112(6), 1831–1846.

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

[68] Duvenaud, D., Eaton, D., Murphy, K., & Schmidt, M. (2008). Causal learning without DAGs. In *Neural Information Processing Systems Workshop on Causality*.

[69] Eddy, S. R. (2011). Accelerated profile HMM searches. *PLoS Comput. Biol.*, 7(10), e1002195.

[70] Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, 32(5), 1792–1797.

[71] Efron, B. & Stein, C. (1981). The jackknife estimate of variance. *Ann. Stat.*, 9(3), 586–596.

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

[73] Endelman, J. B., Silberg, J. J., Wang, Z.-G., & Arnold, F. H. (2004). Site-directed protein recombination as a shortest-path problem. *Protein Eng. Des. Sel.*, 17(7), 589–594.

[74] Felsenstein, J. (1985). Phylogenies and the comparative method. *Am. Nat.*, 125(1), 1–15.

[75] Felsenstein, J. (2004). *Inferring phylogenies*. Sinauer associates, Sunderland, MA.

[76] Ferguson, T. S. (1974). Prior distributions on spaces of probability measures. *The Annals of Statistics*, 2(4), 615–629.

[77] Fischer, D. S., Wu, Y., Schubert, B., & Theis, F. J. (2020). Predicting antigen specificity of single T cells based on TCR CDR3 regions. *Mol. Syst. Biol.*, 16(8), e9416.

[78] Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons.

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

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

[81] Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659), 799–805.

[82] Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, 7(3-4), 601–620.

[83] Galiez, C., Siebert, M., Enault, F., Vincent, J., & Söding, J. (2017). WIsH: who is the host? predicting prokaryotic hosts from metagenomic phage contigs. *Bioinformatics*, 33(19), 3113–3114.

[84] Gasteiger, E., Hoogland, C., Gattiker, A., Duvaud, S., Wilkins, M. R., Appel, R. D., & Bairoch, A. (2005). Protein identification and analysis tools on the ExPASy server. In J. M. Walker (Ed.), *The Proteomics Protocols Handbook* (pp. 571–607). Totowa, NJ: Humana Press.

[85] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC.

[86] Geman, S. & Hwang, C.-R. (1982). Nonparametric maximum likelihood estimation by the method of sieves. *The Annals of Statistics*, 10(2), 401–414.

[87] Ghosal, S., Ghosh, J. K., & van der Vaart, A. W. (2000). Convergence rates of posterior distributions. *The Annals of Statistics*, 28(2), 500–531.

[88] Ghosh, J. K. & Ramamoorthi, R. V. (2003). *Bayesian Nonparametrics*. Series in Statistics. Springer.

[89] Gibson, D. G., Young, L., Chuang, R.-Y., Venter, J. C., Hutchison, 3rd, C. A., & Smith, H. O. (2009). Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat. Methods*, 6(5), 343–345.

[90] Gigante, S., Burkhardt, D., Dager, D., Stanley, J., & Tong, A. (2020). scprep. https://github.com/KrishnaswamyLab/scprep.

[91] Giordano, R., Stephenson, W., Liu, R., Jordan, M., & Broderick, T. (2019). A Swiss Army infinitesimal jackknife. In *International Conference on Artificial Intelligence and Statistics*.

[92] Gopalan, P., Hao, W., Blei, D. M., & Storey, J. D. (2016). Scaling probabilistic models of genetic variation to millions of humans. *Nature Genetics*, 48(12), 1587–1590.

[93] Gorham, J. & Mackey, L. (2017). Measuring sample quality with kernels. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17 (pp. 1292–1301). Sydney, NSW, Australia: JMLR.org.

[94] Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., & Zemel, R. (2020). Learning the Stein discrepancy for training and evaluating energy-based models without sampling. In *Proceedings of the 37th International Conference on Machine Learning*.

[95] Gray, R. M. (2011). *Entropy and Information Theory*. Springer Science & Business Media.

[96] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13, 723–773.

[97] Guo, F., Wang, X., Fan, K., Broderick, T., & Dunson, D. B. (2016). Boosting variational inference. In *NeurIPS Workshop on Advances in Approximate Bayesian Inference*.

[98] Györfi, L. & Van Der Meulen, E. C. (1991). A consistent goodness of fit test based on the total variation distance. In G. Roussas (Ed.), *Nonparametric Functional Estimation and Related Topics* (pp. 631–645). Dordrecht: Springer Netherlands.

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

[100] Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T. B., Dhariwal, P., Gray, S., Hallacy, C., Mann, B., Radford, A., Ramesh, A., Ryder, N., Ziegler,

D. M., Schulman, J., Amodei, D., & McCandlish, S. (2020). Scaling laws for autoregressive generative modeling.

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

[102] Hicks, S. C., Townes, F. W., Teng, M., & Irizarry, R. A. (2018). Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*, 19(4), 562–578.

[103] Hie, B., Zhong, E. D., Berger, B., & Bryson, B. (2021). Learning the language of viral evolution and escape. *Science*, 371(6526), 284–288.

[104] Ho, L. S. T. & Ané, C. (2013). Asymptotic theory with hierarchical autocorrelation: Ornstein–Uhlenbeck tree models. *Ann. Stat.*, 41(2), 957–981.

[105] Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14, 1303–1347.

[106] Holmes, C. C., Caron, F., Griffin, J. E., & Stephens, D. A. (2015). Two-sample bayesian nonparametric hypothesis testing. *Bayesian Anal.*, 10(2), 297–320.

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

[108] Hong, H. & Preston, B. (2005). Nonnested model selection criteria.

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

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

[111] Huang, W., Li, L., Myers, J. R., & Marth, G. T. (2012). ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4), 593–594.

[112] Huber, P. J. (1985). Projection pursuit. *The Annals of Statistics*, 13(2), 435–475.

[113] Huber, P. J. (1992). Robust estimation of a location parameter. In S. Kotz & N. L. Johnson (Eds.), *Breakthroughs in Statistics: Methodology and Distribution* (pp. 492–518). New York, NY: Springer New York.

[114] Huelsenbeck, J. P. & Ronquist, F. (2001). MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8), 754–755.

[115] Huggins, J., Kasprzak, M., Campbell, T., & Broderick, T. (2020). Validated variational inference via practical posterior error bounds. In S. Chiappa & R. Calandra (Eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research* (pp. 1792–1802).: PMLR.

[116] Huggins, J. H. & Mackey, L. (2018). Random feature Stein discrepancies. In *Advances in Neural Information Processing Systems*.

[117] Huggins, J. H. & Miller, J. W. (2020). Robust inference and model criticism using bagged posteriors.

[118] Huggins, J. H. & Miller, J. W. (2022). Reproducible model selection using bagged posteriors. *Bayesian Anal.*

[119] Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., & Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PLoS One*, 5(9).

[120] Imbens, G. W. & Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

[121] Ingraham, J. & Marks, D. (2017). Variational inference for sparse and undirected models. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 (pp. 1607–1616).: PMLR.

[122] Ingraham, J. B. (2018). *Probabilistic Models of Structure in Biological Sequences*. PhD thesis, Harvard Medical School.

[123] Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., & McVean, G. (2012). De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nature Genetics*, 44(2), 226–232.

[124] Iuliano, A. D., Roguski, K. M., Chang, H. H., Muscatello, D. J., Palekar, R., Tempia, S., Cohen, C., Gran, J. M., Schanzer, D., Cowling, B. J., Wu, P., Kyncl, J., Ang, L. W., Park,

M., Redlberger-Fritz, M., Yu, H., Espenhain, L., Krishnan, A., Emukule, G., van Asten, L., Pereira da Silva, S., Aungkulanon, S., Buchholz, U., Widdowson, M.-A., Bresee, J. S., & Global Seasonal Influenza-associated Mortality Collaborator Network (2018). Estimates of global seasonal influenza-associated respiratory mortality: a modelling study. *Lancet*, 391(10127), 1285–1300.

[125] Jacob, P. E., Murray, L. M., Holmes, C. C., & Robert, C. P. (2017). Better together? statistical learning in models made of modules.

[126] Jacobs, T. M., Yumerefendi, H., Kuhlman, B., & Leaver-Fay, A. (2015). SwiftLib: rapid degenerate-codon-library optimization through dynamic programming. *Nucleic Acids Res.*, 43(5), e34.

[127] Jewson, J., Smith, J. Q., & Holmes, C. (2018). Principles of Bayesian inference using general divergence criteria. *Entropy*, 20(6), 442.

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

[129] Jiang, W. & Tanner, M. A. (2008). Gibbs posterior for variable selection in high-dimensional classification and data mining. *The Annals of Statistics*, 36(5), 2207–2231.

[130] Johnson, L. S., Eddy, S. R., & Portugaly, E. (2010). Hidden markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics*, 11, 431.

[131] June, C. H., O'Connor, R. S., Kawalekar, O. U., Ghassemi, S., & Milone, M. C. (2018). CAR T cell immunotherapy for human cancer. *Science*, 359(6382), 1361–1365.

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

[133] Kallenberg, O. (2002). *Foundations of Modern Probability*. Springer Science & Business Media, 2 edition.

[134] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models.

[135] Kayushin, A., Korosteleva, M., & Miroshnikov, A. (2000). Large-scale solid-phase preparation of 3'-unprotected trinucleotide phosphotriesters–precursors for synthesis of trinucleotide phosphoramidites. *Nucleosides Nucleotides Nucleic Acids*, 19(10-12), 1967–1976.

[136] Kayushin, A. L., Korosteleva, M. D., Miroshnikov, A. I., Kosch, W., Zubov, D., & Piel, N. (1996). A convenient approach to the synthesis of trinucleotide phosphoramidites—synthons for the generation of oligonucleotide/peptide libraries. *Nucleic Acids Res.*, 24(1), 9–1996.

[137] Kim, D., Paggi, J. M., Park, C., Bennett, C., & Salzberg, S. L. (2019). Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature Biotechnology*, 37(8), 907–915.

[138] Kingma, D. P. & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

[139] Kingma, D. P. & Welling, M. (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*.

[140] Klima, J. C., Doyle, L. A., Lee, J. D., Rappleye, M., Gagnon, L. A., Lee, M. Y., Barros, E. P., Vorobieva, A. A., Dou, J., Bremner, S., Quon, J. S., Chow, C. M., Carter, L., Mack, D. L., Amaro, R. E., Vaughan, J. C., Berndt, A., Stoddard, B. L., & Baker, D. (2021). Incorporation of sensing modalities into de novo designed fluorescence-activating proteins. *Nat. Commun.*, 12(1), 856.

[141] Knoblauch, J., Jewson, J., & Damoulas, T. (2019). Generalized variational inference: Three arguments for deriving new posteriors.

[142] Kokot, M., Dlugosz, M., & Deorowicz, S. (2017). KMC 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17), 2759–2761.

[143] Kool, W., van Hoof, H., & Welling, M. (2019a). Buy 4 REINFORCE samples, get a baseline for free. In *ICLR Workshop: Deep Reinforcement Learning Meets Structured Prediction*.

[144] Kool, W., van Hoof, H., & Welling, M. (2019b). Stochastic beams and where to find them: The Gumbel-Top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning* (pp. 3499–3508).: PMLR.

[145] Kosuri, S. & Church, G. M. (2014). Large-scale de novo DNA synthesis: technologies and applications. *Nat. Methods*, 11(5), 499–507.

[146] Kryazhimskiy, S., Dieckmann, U., Levin, S. A., & Dushoff, J. (2007). On state-space reduction in multi-strain pathogen models, with an application to antigenic drift in influenza a. *PLoS Comput. Biol.*, 3(8), e159.

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

[148] Landrum, M. J., Lee, J. M., Benson, M., Brown, G. R., Chao, C., Chitipiralla, S., Gu, B., Hart, J., Hoffman, D., Jang, W., Karapetyan, K., Katz, K., Liu, C., Maddipatla, Z., Malheiro, A., McDaniel, K., Ovetsky, M., Riley, G., Zhou, G., Holmes, J. B., Kattman, B. L., & Maglott, D. R. (2018). ClinVar: Improving access to variant interpretations and supporting evidence. *Nucleic Acids Res.*, 46(D1), D1062–D1067.

[149] Lapedes, A. S., Giraud, B. G., Liu, L., & Stormo, G. (1999). Correlated mutations in models of protein sequences: phylogenetic and structural effects. *Statistics in Molecular Biology, IMS Lecture Notes - Monograph Series*, 33, 236–256.

[150] Laursen, N. S. & Wilson, I. A. (2013). Broadly neutralizing antibodies against influenza viruses. *Antiviral Res.*, 98(3), 476–483.

[151] Lavine, M. (1992). Some aspects of Polya tree distributions for statistical modelling. *The Annals of Statistics*, 20(3), 1222–1235.

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

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

[154] Lewis, J. R., MacEachern, S. N., & Lee, Y. (2021). Bayesian restricted likelihood methods: Conditioning on insufficient statistics in Bayesian regression. *Bayesian Analysis*, 1(1), 1–38.

[155] Li, Y., Swersky, K., & Zemel, R. (2015). Generative moment matching networks. In *International Conference on Machine Learning* (pp. 1718–1727).: PMLR.

[156] Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary Mathematics*, 80(1), 221–239.

[157] Liu, H., Lafferty, J., & Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10(Oct), 2295–2328.

[158] Liu, Q., Lee, J. D., & Jordan, M. (2016). A kernelized stein discrepancy for goodness-of-fit tests. *Proceedings of the International Conference on Machine Learning*, 33.

[159] Lloyd, J. R. & Ghahramani, Z. (2015). Statistical model criticism using kernel two sample tests. In *Advances in Neural Information Processing Systems* (pp. 829–837).

[160] Lloyd-Price, J., Arze, C., Ananthakrishnan, A. N., Schirmer, M., Avila-Pacheco, J., Poon, T. W., Andrews, E., Ajami, N. J., Bonham, K. S., Brislawn, C. J., Casero, D., Courtney, H., Gonzalez, A., Graeber, T. G., Hall, A. B., Lake, K., Landers, C. J., Mallick, H., Plichta, D. R., Prasad, M., Rahnavard, G., Sauk, J., Shungin, D., Vázquez-Baeza, Y., White, 3rd, R. A., IBDMDB Investigators, Braun, J., Denson, L. A., Jansson, J. K., Knight, R., Kugathasan, S., McGovern, D. P. B., Petrosino, J. F., Stappenbeck, T. S., Winter, H. S., Clish, C. B., Franzosa, E. A., Vlamakis, H., Xavier, R. J., & Huttenhower, C. (2019). Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. *Nature*, 569(7758), 655–662.

[161] Lloyd-Price, J., Mahurkar, A., Rahnavard, G., Crabtree, J., Orvis, J., Hall, A. B., Brady, A., Creasy, H. H., McCracken, C., Giglio, M. G., McDonald, D., Franzosa, E. A., Knight, R., White, O., & Huttenhower, C. (2017). Strains, functions and dynamics in the expanded human microbiome project. *Nature*, 550(7674), 61–66.

[162] Locatello, F., Khanna, R., Ghosh, J., & Ratsch, G. (2018). Boosting variational inference: an optimization perspective. In A. Storkey & F. Perez-Cruz (Eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research* (pp. 464–472).: PMLR.

[163] Luksza, M. & Lässig, M. (2014). A predictive fitness model for influenza. *Nature*, 507(7490), 57–61.

[164] Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos, J. L., Xiong, C., Sun, Z. Z., Socher, R., Fraser, J. S., & Naik, N. (2021). Deep neural language modeling enables functional protein generation across families.

[165] Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S., & Socher, R. (2020). ProGen: Language modeling for protein generation.

[166] Marçais, G. & Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6), 764–770.

[167] Mardia, K. V. & Marshall, R. J. (1984). Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1), 135–146.

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

[169] Matsubara, T., Knoblauch, J., Briol, F.-X., & Oates, C. J. (2021). Robust generalised bayesian inference for intractable likelihoods.

[170] Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M. S. H., Ko, S. B. H., Gouda, N., Hayashi, T., & Nikaido, I. (2017). SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. *Bioinformatics*, 33(15), 2314–2321.

[171] Mauldin, R. D., Sudderth, W. D., & Williams, S. C. (1992). Polya trees and random distributions. *The Annals of Statistics*, 20(3), 1203–1221.

[172] McMahon, C., Baier, A. S., Pascolutti, R., Wegrecki, M., Zheng, S., Ong, J. X., Erlandson, S. C., Hilger, D., Rasmussen, S. G. F., Ring, A. M., Manglik, A., & Kruse, A. C. (2018). Yeast surface display platform for rapid discovery of conformationally selective nanobodies. *Nat. Struct. Mol. Biol.*, 25(3), 289–296.

[173] Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., & Rives, A. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34.

[174] Mena, M. A. & Daugherty, P. S. (2005). Automated design of degenerate codon libraries. *Protein Eng. Des. Sel.*, 18(12), 559–561.

[175] Miller, A. C., Foti, N., & Adams, R. P. (2017). Variational boosting: Iteratively refining posterior approximations. In *International Conference on Machine Learning*.

[176] Miller, J. W. (2021). Asymptotic normality, concentration, and coverage of generalized posteriors. *Journal of Machine Learning Research*, 22(168), 1–53.

[177] Miller, J. W. & Dunson, D. B. (2019). Robust Bayesian inference via coarsening. *Journal of the American Statistical Association*, 114(527), 1113–1125.

[178] Minka, T. (2000a). Old and new matrix algebra useful for statistics.

[179] Minka, T. P. (2000b). Automatic choice of dimensionality for PCA.

[180] Mohamed, S. & Lakshminarayanan, B. (2016). Learning in implicit generative models.

[181] Moignard, V., Woodhouse, S., Haghverdi, L., Lilly, A. J., Tanaka, Y., Wilkinson, A. C., Buettner, F., Macaulay, I. C., Jawaid, W., Diamanti, E., Nishikawa, S.-I., Piterman, N., Kouskoff, V., Theis, F. J., Fisher, J., & Göttgens, B. (2015). Decoding the regulatory network of early blood development from single-cell gene expression measurements. *Nature Biotechnology*, 33(3), 269–276.

[182] Moore, G. L. & Maranas, C. D. (2000). Modeling DNA mutation and recombination for directed evolution experiments. *J. Theor. Biol.*, 205(3), 483–503.

[183] Mukamel, R. E., Handsaker, R. E., Sherman, M. A., Barton, A. R., Zheng, Y., McCarroll, S. A., & Loh, P.-R. (2021). Protein-coding repeat polymorphisms strongly shape diverse human phenotypes. *Science*, 373(6562), 1499–1505.

[184] Muñoz, E. T. & Deem, M. W. (2005). Epitope analysis for influenza vaccine design. *Vaccine*, 23(9), 1144–1148.

[185] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

[186] Narlikar, L., Mehta, N., Galande, S., & Arjunwadkar, M. (2013). One size does not fit all: on how Markov model order dictates performance of genomic sequence analyses. *Nucleic Acids Research*, 41(3), 1416–1424.

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

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

[189] Novembre, J. & Stephens, M. (2008). Interpreting principal component analyses of spatial population genetic variation. *Nat. Genet.*, 40(5), 646–649.

[190] O'Leary, N. A., Wright, M. W., Brister, J. R., Ciufo, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D., Astashyn, A., Badretdin, A., Bao, Y., Blinkova, O., Brover, V., Chetvernin, V., Choi, J., Cox, E., Ermolaeva, O., Farrell, C. M., Goldfarb, T., Gupta, T., Haft, D., Hatcher, E., Hlavina, W., Joardar, V. S., Kodali, V. K., Li, W., Maglott, D., Masterson, P., McGarvey, K. M., Murphy, M. R., O'Neill, K., Pujar, S., Rangwala, S. H., Rausch, D., Riddick, L. D., Schoch, C., Shkeda, A., Storz, S. S., Sun, H., Thibaud-Nissen, F., Tolstoy, I., Tully, R. E., Vatsan, A. R., Wallin, C., Webb, D., Wu, W., Landrum, M. J., Kimchi, A., Tatusova, T., DiCuccio, M., Kitts, P., Murphy, T. D., & Pruitt, K. D. (2016). Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 44(D1), D733–45.

[191] Orlitsky, A., Suresh, A. T., & Wu, Y. (2016). Optimal prediction of the number of unseen species. *Proc. Natl. Acad. Sci. U. S. A.*, 113(47), 13283–13288.

[192] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics* (pp. 311–318).

[193] Parker, A. S., Griswold, K. E., & Bailey-Kellogg, C. (2011). Optimization of combinatorial mutagenesis. *J. Comput. Biol.*, 18(11), 1743–1756.

[194] Pazdernik, N. & Bowersox, A. (2016). Need a library of related DNA or RNA oligo sequences? https://www.idtdna.com/pages/education/decoded/article/need-a-library-of-related-dna-or-rna-oligo-sequences. Accessed: 2020-8-25.

[195] Pearl, J. (2009). *Causality*. Cambridge University Press.

[196] Petrone, S., Rousseau, J., & Scricciolo, C. (2014). Bayes and empirical Bayes: do they merge? *Biometrika*, 101(2), 285–302.

[197] Pierson, E. & Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.*, 16, 241.

[198] Pinho, A. J., Ferreira, P. J. S. G., Neves, A. J. R., & Bastos, C. A. C. (2011). On the representability of complete genomes by multiple competing finite-context (Markov) models. *PLoS One*, 6(6), e21588.

[199] Pitman, J. (2002). *Combinatorial stochastic processes*. Technical Report 621, Dept of Statistics, UC Berkeley.

[200] Plesa, C., Sidore, A. M., Lubock, N. B., Zhang, D., & Kosuri, S. (2018). Multiplexed gene synthesis in emulsions for exploring protein functional landscapes. *Science*, 359(6373), 343–347.

[201] Potter, S. C., Luciani, A., Eddy, S. R., Park, Y., Lopez, R., & Finn, R. D. (2018). HMMER web server: 2018 update. *Nucleic Acids Res.*, 46(W1), W200–W204.

[202] Pratas, D., Hosseini, M., & Pinho, A. J. (2017). Substitutional tolerant markov models for relative compression of DNA sequences. In *International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB)* (pp. 265–272).

[203] Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics*, 155(2), 945–959.

[204] Pritchard, L., Corne, D., Kell, D., Rowland, J., & Winson, M. (2005). A general model of error-prone PCR. *J. Theor. Biol.*, 234(4), 497–509.

[205] Qin, C. & Colwell, L. J. (2018). Power law tails in phylogenetic systems. *Proc. Natl. Acad. Sci. U. S. A.*, 115(4), 690–695.

[206] Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H. A., & Trapnell, C. (2017). Reversed graph embedding resolves complex single-cell trajectories. *Nature Methods*, 14(10), 979.

[207] Ramien, C., Yusko, E. C., Engler, J. B., Gamradt, S., Patas, K., Schweingruber, N., Willing, A., Rosenkranz, S. C., Diemert, A., Harrison, A., Vignali, M., Sanders, C., Robins, H. S.,

Tolosa, E., Heesen, C., Arck, P. C., Scheffold, A., Chan, K., Emerson, R. O., Friese, M. A., & Gold, S. M. (2019). T cell repertoire dynamics during pregnancy in multiple sclerosis. *Cell Rep.*, 29(4), 810–815.e4.

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

[209] Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., & Song, Y. S. (2019). Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems*, volume 32 (pp. 9689–9701).

[210] Ravikumar, A., Arzumanyan, G. A., Obadi, M. K. A., Javanpour, A. A., & Liu, C. C. (2018). Scalable, continuous evolution of genes at mutation rates above genomic error thresholds. *Cell*, 175(7), 1946–1957.e13.

[211] Ren, J., Bai, X., Lu, Y. Y., Tang, K., Wang, Y., Reinert, G., & Sun, F. (2018). Alignment-free sequence analysis and applications. *Annual Review of Biomedical Data Science*, 1, 93–114.

[212] Rezende, D. J. (2018). Short notes on divergence measures.

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

[214] Richardson, E. & Weiss, Y. (2018). On GANs and GMMs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (pp. 5847–5858).

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

[216] Rivas, E., Clements, J., & Eddy, S. R. (2017). A statistical test for conserved RNA structure shows lack of evidence for structure in lncRNAs. *Nat. Methods*, 14(1), 45–48.

[217] Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.*, 118(15).

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

[219] Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press.

[220] Rodriguez Horta, E., Barrat-Charlaix, P., & Weigt, M. (2019). Toward inferring potts models for phylogenetically correlated sequence data. *Entropy*, 21(11), 1090.

[221] Rousseau, J. (2016). On the frequentist properties of Bayesian nonparametric methods. *Annual Review of Statistics and Its Application*, 3, 211–231.

[222] Rousseau, J. & Mengersen, K. (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B*, 73(5), 689–710.

[223] Rush, A. M. (2020). Torch-Struct: Deep structured prediction library.

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

[225] Sarkar, R. (2012). Low distortion delaunay embedding of trees in hyperbolic plane.

[226] Sarkisyan, K. S., Bolotin, D. A., Meer, M. V., Usmanova, D. R., Mishin, A. S., Sharonov, G. V., Ivankov, D. N., Bozhanova, N. G., Baranov, M. S., Soylemez, O., Bogatyreva, N. S., Vlasov, P. K., Egorov, E. S., Logacheva, M. D., Kondrashov, A. S., Chudakov, D. M., Putintseva, E. V., Mamedov, I. Z., Tawfik, D. S., Lukyanov, K. A., & Kondrashov, F. A. (2016). Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603), 397–401.

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

[228] Schirmer, M., Garner, A., Vlamakis, H., & Xavier, R. J. (2019). Microbial genes and pathways in inflammatory bowel disease. *Nat. Rev. Microbiol.*, 17(8), 497–511.

[229] Schreiber, P. W., Kufner, V., Hübel, K., Schmutz, S., Zagordi, O., Kaur, A., Bayard, C., Greiner, M., Zbinden, A., Capaul, R., Böni, J., Hirsch, H. H., Mueller, T. F., Mueller, N. J.,

Trkola, A., & Huber, M. (2019). Metagenomic virome sequencing in living donor and recipient kidney transplant pairs revealed JC polyomavirus transmission. *Clinical Infectious Diseases*, 69(6), 987–994.

[230] Sella, G. & Hirsh, A. E. (2005). The application of statistical physics to evolutionary biology. *Proceedings of the National Academy of Sciences*, 102(27), 9541–9546.

[231] Serfling, R. J. (2009). *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons.

[232] Shalek, A. K., Satija, R., Adiconis, X., Gertner, R. S., Gaublomme, J. T., Raychowdhury, R., Schwartz, S., Yosef, N., Malboeuf, C., Lu, D., Trombetta, J. J., Gennert, D., Gnirke, A., Goren, A., Hacohen, N., Levin, J. Z., Park, H., & Regev, A. (2013). Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498(7453), 236–240.

[233] Shao, S., Jacob, P. E., Ding, J., & Tarokh, V. (2018). Bayesian model comparison with the Hyvärinen score: Computation and consistency. *Journal of the American Statistical Association*, (pp. 1–24).

[234] Shimko, T. C., Fordyce, P. M., & Orenstein, Y. (2020). DeCoDe: degenerate codon design for complete protein-coding DNA libraries. *Bioinformatics*, 36(11), 3357–3364.

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

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

[237] Silva, M., Pratas, D., & Pinho, A. J. (2020). Efficient DNA sequence compression with neural networks. *Gigascience*, 9(11).

[238] Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. M., & Birol, I. (2009). ABySS: a parallel assembler for short read sequence data. *Genome Research*, 19(6), 1117–1123.

[239] Sinai, S., Wang, R., Whatley, A., Slocum, S., Locane, E., & Kelsic, E. D. (2020). AdaLead: A simple and robust adaptive greedy search algorithm for sequence design.

[240] Singer, Z. S., Yong, J., Tischler, J., Hackett, J. A., Altinok, A., Surani, M. A., Cai, L., & Elowitz, M. B. (2014). Dynamic heterogeneity and DNA methylation in embryonic stem cells. *Molecular Cell*, 55(2), 319–331.

[241] Skowronski, D. M., Janjua, N. Z., De Serres, G., Purych, D., Gilca, V., Scheifele, D. W., Dionne, M., Sabaiduc, S., Gardy, J. L., Li, G., Bastien, N., Petric, M., Boivin, G., & Li, Y. (2012). 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.

[242] Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., & Lanckriet, G. R. G. (2009). On integral probability metrics, φ-divergences and binary classification.

[243] Sriperumbudur, B. K., Fukumizu, K., & Lanckriet, G. R. G. (2011). Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12, 2389–2410.

[244] Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., & Lanckriet, G. R. G. (2010). Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11, 1517–1561.

[245] Stamatakis, A. (2006). RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21), 2688–2690.

[246] Steinhardt, J. (2018). *Robust Learning: Information Theory and Algorithms*. PhD thesis, Stanford University.

[247] Steinwart, I. & Christmann, A. (2008). *Support Vector Machines*. Springer Science & Business Media.

[248] Stephens, P. J., Greenman, C. D., Fu, B., Yang, F., Bignell, G. R., Mudie, L. J., Pleasance, E. D., Lau, K. W., Beare, D., Stebbings, L. A., McLaren, S., Lin, M.-L., McBride, D. J., Varela, I., Nik-Zainal, S., Leroy, C., Jia, M., Menzies, A., Butler, A. P., Teague, J. W., Quail, M. A., Burton, J., Swerdlow, H., Carter, N. P., Morsberger, L. A., Iacobuzio-Donahue, C., Follows, G. A., Green, A. R., Flanagan, A. M., Stratton, M. R., Futreal, P. A., & Campbell, P. J. (2011). Massive genomic rearrangement acquired in a single catastrophic event during cancer development. *Cell*, 144(1), 27–40.

[249] Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, 3rd, W. M., Hao, Y., Stoeckius, M., Smibert, P., & Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*, 177(7), 1888–1902.e21.

[250] Sutherland, D. J., Tung, H. Y., Strathmann, H., De, S., Ramdas, A., Smola, A., & Gretton, A. (2017). Generative models and model criticism via optimized maximum mean discrepancy. In *International Conference on Learning Representations*.

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

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

[253] Szpiro, A. A., Rice, K. M., & Lumley, T. (2010). Model-robust regression and a Bayesian "sandwich" estimator. *Ann. Appl. Stat.*, 4(4), 2099–2113.

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

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

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

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

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

[259] Tipping, M. E. & Bishop, C. M. (1999). Probabilistic principal component analysis. *J. R. Stat. Soc. Series B Stat. Methodol.*, 61(3), 611–622.

[260] Tomezsko, P. J., Corbin, V. D. A., Gupta, P., Swaminathan, H., Glasgow, M., Persad, S., Edwards, M. D., Mcintosh, L., Papenfuss, A. T., Emery, A., Swanstrom, R., Zang, T., Lan, T. C. T., Bieniasz, P., Kuritzkes, D. R., Tsibris, A., & Rouskin, S. (2020). Determination of RNA structural diversity and its role in HIV-1 RNA splicing. *Nature*, 582, 438–442.

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

[262] Townsend, J., Koep, N., & Weichwald, S. (2016). Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*.

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

[264] Twist Bioscience (2020). *Combinatorial Variant Libraries*.

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

[266] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio.

[267] van der Maaten, L. & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.

[268] van der Vaart, A. W. (1998). *Asymptotic Statistics*.

[269] van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., Burdziak, C., Moon, K. R., Chaffer, C. L., Pattabiraman, D., Bierie, B., Mazutis, L., Wolf, G., Krishnaswamy, S., & Pe'er, D. (2018). Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3), 716–729.e27.

[270] Van Noorden, B. Y. R., Maher, B., & Nuzzo, R. (2014). Nature explores the most-cited research of all time. *Nature*, 514, 550–553.

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

[272] Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, 21(1), 5–42.

[273] Verdinelli, I. & Wasserman, L. (1998). Bayesian goodness-of-fit testing using infinite-dimensional exponential families. *The Annals of Statistics*, 26(4), 1215–1241.

[274] Vershynin, R. (2020). *High-Dimensional Probability: An Introduction with Applications in Data Science*.

[275] Vikram, S., Hoffman, M. D., & Johnson, M. J. (2019). The LORACs prior for VAEs: Letting the trees speak for the data. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89 (pp. 3292–3301).: PMLR.

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

[277] Voichek, Y. & Weigel, D. (2020). Identifying genetic variants underlying phenotypic variation in plants without complete genomes. *Nature Genetics*, 52(5), 534–540.

[278] Voigt, C. A., Martinez, C., Wang, Z.-G., Mayo, S. L., & Arnold, F. H. (2002). Protein building blocks preserved by recombination. *Nat. Struct. Biol.*, 9(7), 553–558.

[279] Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica: Journal of the Econometric Society*, 57(2), 307–333.

[280] Weinreb, C., Riesselman, A. J., Ingraham, J. B., Gross, T., Sander, C., & Marks, D. S. (2016). 3D RNA and functional interactions from evolutionary couplings. *Cell*, 165(4), 963–975.

[281] Weinstein, E. N., Amin, A. N., Frazer, J., & Marks, D. (2022a). Non-identifiability and the blessings of misspecification in models of molecular fitness and phylogeny.

[282] Weinstein, E. N., Amin, A. N., Grathwohl, W., Kassler, D., Disset, J., & Marks, D. S. (2022b). Optimal design of stochastic DNA synthesis protocols based on generative sequence models. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*: PMLR.

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

[284] Weinstein, E. N. & Marks, D. S. (2021). A structured observation distribution for generative biological sequence prediction and forecasting. In *International Conference on Machine Learning*, 139 (pp. 11068–11079).: PMLR.

[285] Weinstein, E. N. & Miller, J. W. (2021). Bayesian data selection.

[286] Welling, M. & Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*.

[287] Wilburn, G. W. & Eddy, S. R. (2020). Remote homology search with hidden potts models. *PLoS Comput Biol*, 16(11).

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

[289] Williams, C. K. I. & Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. MIT press Cambridge, MA.

[290] Wilson, D. S. & Keefe, A. D. (2001). Random mutagenesis by PCR. *Curr. Protoc. Mol. Biol.*, Chapter 8, Unit8.3.

[291] Wolf, F. A., Angerer, P., & Theis, F. J. (2018). SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1), 15.

[292] Wu, M., Chatterji, S., & Eisen, J. A. (2012). Accounting for alignment uncertainty in phylogenomics. *PLoS One*, 7(1), e30288.

[293] Xu-Monette, Z. Y., Li, L., Byrd, J. C., Jabbar, K. J., Manyam, G. C., Maria de Winde, C., van den Brand, M., Tzankov, A., Visco, C., Wang, J., Dybkaer, K., Chiu, A., Orazi, A., Zu, Y., Bhagat, G., Richards, K. L., Hsi, E. D., Choi, W. W. L., Huh, J., Ponzoni, M., Ferreri, A. J. M., Møller, M. B., Parsons, B. M., Winter, J. N., Wang, M., Hagemeister, F. B., Piris, M. A., Han van Krieken, J., Medeiros, L. J., Li, Y., van Spriel, A. B., & Young, K. H. (2016). Assessment of CD37 B-cell antigen and cell of origin significantly improves risk prediction in diffuse large B-cell lymphoma. *Blood*, 128(26), 3083–3100.

[294] Yang, K. K., Chen, Y., Lee, A., & Yue, Y. (2019). Batched stochastic bayesian optimization via combinatorial constraints design. In *International Conference on Artificial Intelligence and Statistics*.

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

[296] Yeo, I.-K. & Johnson, R. A. (2001). A uniform strong law of large numbers for U-statistics with application to transforming to near symmetry. *Statistics & Probability Letters*, 51(1), 63–69.

[297] Zhang, T. (2003). Sequential greedy approximation for certain convex optimization problems. *IEEE Trans. Inf. Theory*, 49(3), 682–691.

[298] Zhang, T. (2006a). From ϵ-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, 34(5), 2180–2210.

[299] Zhang, T. (2006b). Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4), 1307–1321.

[300] Zuboff, S. (2019). *The Age of Surveillance Capitalism*. Public Affairs.