

Biological Sequence Kernels with Guaranteed Flexibility

Alan N. Amin

New York University

ALANAMIN@NYU.EDU

Debora S. Marks*

Harvard Medical School

Broad Institute of Harvard and MIT

DEBBIE@HMS.HARVARD.EDU

Eli N. Weinstein*

Technical University of Denmark

ENAW@DTU.DK

Editor: Jean-Philippe Vert

Abstract

Applying machine learning to biological sequences—DNA, RNA and protein—has enormous potential to advance human health and environmental sustainability. To support such high-stakes applications, it is important to develop models and evaluations that not only capture underlying biology, but also have theoretical guarantees of reliability and performance. In this article, we analyze kernel methods for biological sequences, including both hand-crafted kernels and deep neural network-based kernels. We show that popular biological kernels can severely fail at learning functions or distinguishing distributions. We then develop modified kernels that (1) are universal, characteristic, and metrize the space of distributions, and (2) preserve the underlying biological inductive biases and domain knowledge embedded in the original kernel. Our results rest on novel proof techniques for kernels that handle the structure of biological sequence space—discrete, variable length sequences—and biological notions of sequence similarity. We illustrate our theoretical results in simulation and on real biological data sets.

Keywords: kernel methods, sequences, biology, nonparametric statistics, representations

1. Introduction

DNA, RNA and protein sequence data have become a rich area of scientific application for modern machine learning, with tools emerging to diagnose genetic diseases, engineer proteins, and design therapeutics (Frazer et al., 2021; Yang et al., 2019; Notin et al., 2024). As high-stakes real-world applications proliferate, it becomes increasingly important to develop robust evaluations, reliable procedures, and accurate theoretical understanding.

One promising route to advancing these goals is through kernel methods. In principle, kernel methods can simultaneously offer: (1) rich inductive biases, allowing scientists to build in prior domain knowledge, (2) flexibility, ensuring models can capture arbitrary complexity in the data, and (3) theoretical tractability, enabling rigorous guarantees and understanding of learning procedures.

Computational biologists have developed kernels with biological inductive biases, that capture our basic understanding of sequence evolution and molecular function (Schweikert

*. Equal contribution.

et al., 2008; Toussaint et al., 2010; Leslie et al., 2002; Yang et al., 2018; Alley et al., 2019). The problem, as we will show, is that these kernels are inflexible. For example, they are not *universal*, so they cannot describe arbitrary functions. As a consequence, methods that use these kernels do not have strong theoretical guarantees on their success.

This article proposes modifications to popular biological sequence kernels that provide theoretical guarantees on flexibility, while preserving their biological inductive biases. Specifically, we construct kernels that are:

1. *Universal*. This ensures that kernel regression methods can capture any function on sequences. So, for instance, biologists can learn any sequence-to-phenotype relationship.
2. *Characteristic*. This ensures that the maximum mean discrepancy (MMD) between two distributions is zero if and only if the two distributions match exactly (Gretton et al., 2012). So, for instance, biologists can detect any mismatch between a deep generative model and real biological sequences.
3. *Metriize the space of distributions*. This ensures that minimizing the MMD with respect to the first distribution yields the second. So, for instance, biologists can train simulators to match real biological sequence data by minimizing the MMD (Li et al., 2017; Park et al., 2016; Dellaporta et al., 2022)

These properties appear widely as necessary conditions for kernel-based machine learning methods, across tasks from representation learning to causal inference (Pogodin et al., 2022; Singh et al., 2023).

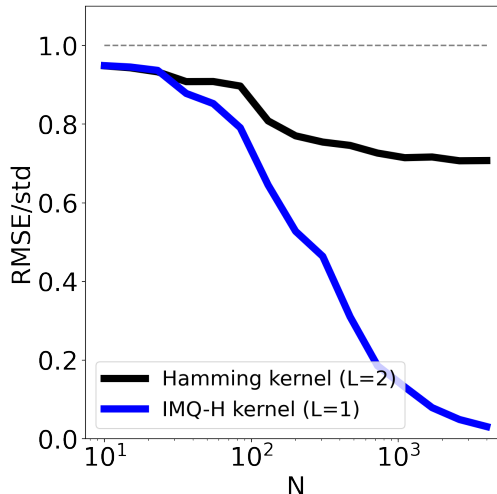
Figure 1 provides a toy illustration of the impact of our theory on a regression problem (details in Appendix A). We compare a standard kernel (black) to one of our modified kernels (blue). Both kernels judge sequence dissimilarity based on the Hamming distance, a common measure of mutational distance in biology. But the kernels differ in their functional form. As a result, the standard kernel fails to learn the true function, while our new universal kernel succeeds.

Studying kernels on biological sequence space presents unique theoretical challenges. Biological sequences are strings of characters (nucleotides or amino acids), and typically vary in length across a data set. Thus, following previous theoretical work on biological sequences, we take sequence space to be the set of all finite-length strings $\cup_{L=0}^{\infty} \mathcal{B}^L$ of an alphabet \mathcal{B} , where e.g. $\mathcal{B} = \{A, T, G, C\}$ for DNA (Amin et al., 2021; Weinstein, 2022). Biological sequence space is therefore discrete and infinite, while the vast majority of previous theoretical studies have been done on continuous or finite spaces. We present the first theoretical results on the flexibility of kernels for biological sequences.

Our first step is to show that kernels on biological sequences have guaranteed flexibility if they have discrete masses, i.e. if the associated Hilbert space contains delta functions (Section 4) (Jorgensen and Tian, 2015). Then we develop a suite of tools to prove that kernels have discrete masses (Section 5).

We apply these methods to study four key classes of biological sequence kernel that are in practical use by computational biologists. For each, we diagnose problems with current kernels, and develop modifications that grant flexibility while preserving biological inductive biases.

Figure 1: Least squares regression with Hamming and IMQ-H kernels. N is the size of the data set. Performance is measured by root mean squared error, normalized by the standard deviation of the outcome variable. Details in Appendix A



- *Position-wise comparison kernels (Section 6)*. These kernels compare sequences position-by-position, e.g. kernels based on the Hamming distance.
- *Alignment kernels (Section 7)*. These kernels compare sequences with a pairwise alignment, accounting for insertion and deletion mutations.
- *Kmer spectrum kernels (Section 8)*. These kernels compare sequences based on their sub-strings (kmers), accounting for similarities in the set of motifs in each sequence.
- *Embedding (Section 9)*. These kernels compare sequences based on an embedding into Euclidean space. The embedding can come from neural network, e.g. a representation from a “foundation” model.

Table 1 gives a general overview. We illustrate our results on both synthetic and real data in Section 10, showing they result in improved empirical performance.

2. Notation

In this section we establish notation for reference in the rest of the text.

We let S be a countably infinite set with the discrete topology. In sections 4 and 5 where we study guarantees for kernels on arbitrary infinite discrete spaces, S can be any countable infinite set; in the rest of the paper we will specifically be interested in the case when S is the space of sequences (defined below). For any finite set B we will define $|B|$ as its cardinality. We let \mathbb{N} be the set of natural numbers $\{0, 1, 2, \dots\}$. We also define $\mathbb{1}(P)$ to be the indicator function, which is 1 if P is true and 0 if it is false. If $y, z \in \mathbb{R}$, we define $y \vee z$ as their maximum and $y \wedge z$ as their minimum.

For $X \in S$, by δ_X we either mean the function or measure that is 1 on X and 0 everywhere else. We call $C_0(S)$ the space of functions on S that vanish at infinity and

Table 1: **Overview of standard kernels and proposed modifications.** Each section of the paper studies a different family of biological sequence kernels. In this table, we list an especially well-known instance of each family, and an example of an alternative we propose, which has guaranteed flexibility and similar biological inductive biases. We report the computational complexity of standard algorithms for evaluating each example on two sequences of length L . Here, k is k -mer length and d is the embedding dimension (we ignore the fixed cost of embedding each sequence and assume an RBF kernel in the embedding space).

| Kernel | Has discrete masses | Compute | Section |
|------------------------|---------------------|---------------------|---------|
| Hamming | No | $\mathcal{O}(L)$ | Sec. 6 |
| IMQ-H | Yes | $\mathcal{O}(L)$ | |
| Alignment | Sometimes | $\mathcal{O}(L^2)$ | Sec. 7 |
| Heavy-tailed alignment | Yes | $\mathcal{O}(L^3)$ | |
| Kmer spectrum | No | $\mathcal{O}(kL^2)$ | Sec. 8 |
| Infinite spectrum | Yes | $\mathcal{O}(L^2)$ | |
| Embedding | Usually no | $\mathcal{O}(d)$ | Sec. 9 |
| Scaled embedding | Usually yes | $\mathcal{O}(d)$ | |

$\|\cdot\|_\infty$ the infinity norm on $C_0(S)$. We define $C_C(S)$ to be the set of functions on S that are non-zero at only finitely many points. We define $\mathcal{P}(S)$ to be the space of probability distributions on S .

A kernel is a function $k : S \times S \rightarrow \mathbb{R}$ such that for any finite set of $Y_1, \dots, Y_N \in S$ and $\alpha_1, \dots, \alpha_N \in \mathbb{R}$, $\sum_{n,m} \alpha_n \alpha_m k(Y_n, Y_m) \geq 0$. We define the inner product $(\cdot|\cdot)_k$ on linear combinations of functions $k_Y = k(Y, \cdot)$ so that $(\sum_{n=1}^N \alpha_n k_{X_n} | \sum_{m=1}^M \beta_m k_{Y_m})_k = \sum_{n,m} \alpha_n \beta_m k(X_n, Y_m)$ for $X_n, Y_m \in S$ and $\alpha_n, \beta_m \in \mathbb{R}$. We denote the associated norm as $\|\cdot\|_k$. We say that a kernel is strictly positive definite if $\sum_{n=1}^N \alpha_n k_{X_n} \neq 0$ when the X_n are distinct and α_n are non-zero. We define the reproducing kernel Hilbert space (RKHS) of the kernel \mathcal{H}_k as the Hilbert space completion using the inner product $(\cdot|\cdot)_k$. We can write every $f \in \mathcal{H}_k$ as a function on S given by $f(X) = (f|k_X)_k$. For a set of vectors $\{v_\lambda\}_\lambda \subset \mathcal{H}_k$, we define $\text{span}\{v_\lambda\}_\lambda$ as the set of *finite* linear combinations of elements of $\{v_\lambda\}_\lambda$.

Say μ is a signed measure on S such that $\int d|\mu|(X) \sqrt{k(X, X)} < \infty$. Then there is a $f \in \mathcal{H}_k$ such that for all $g \in \mathcal{H}_k$ we have $(f|g)_k = \int d\mu(X) g(X)$ and $\|f\|_k^2 = \int d\mu(X) \int d\mu(Y) k(X, Y)$. This element of \mathcal{H}_k is called the “kernel embedding” of the measure μ , denoted $\int d\mu(X) k_X$.

Let \mathcal{B} be a finite set, the “alphabet” (for example, this would be the four nucleotides for DNA, or the twenty amino acids for proteins). We define the set of sequences as $\cup_{L=0}^\infty \mathcal{B}^L$ where \mathcal{B}^0 is defined to be the set containing just the sequence of length zero, \emptyset . For a sequence X , and numbers l, l' , we define $X_{(l)}$ as the first l letters in X , $X_{(l:l')}$ as the

$l' - l$ letters after the first l letters and $X_{(l)}$ as all letters after the first l letters, which is potentially the empty sequence \emptyset . We call $X_{(l)}$ the l -th letter of X , starting counting at 0. The concatenation of sequences X, Y is denoted $X + Y$. For any number l , the sequence X concatenated to itself l times is denoted $l \times X$. We call $d_H(X, Y)$ the Hamming distance between the sequences X and Y , that is the total number of mismatches between the two sequences after they have been padded with an infinite tail of stop symbols $\$$.

3. Related work

There is a large body of theoretical work studying kernel flexibility, much of it focused on the properties of universality, characteristicness, and metrizing probability distributions (Sriperumbudur et al., 2009, 2010, 2011; Fukumizu et al., 2008; Christmann and Steinwart, 2010). These results have been foundational in providing strong guarantees on a wide variety of kernel methods, including not only regression methods but also two-sample tests (Gretton et al., 2012), independence tests (Gretton et al., 2007), sample quality evaluation (Gorham and Mackey, 2017), learning implicit models (Li et al., 2017), learning models with intractable normalizing constants (Dai et al., 2019; Matsubara et al., 2022), causal inference (Singh et al., 2021), Bayesian model selection and data selection (Weinstein and Miller, 2023) and much more. However, virtually all existing flexibility guarantees are for kernels on particular topological spaces. Proof methods apply Fourier transforms to their domains (Sriperumbudur et al., 2011) or require compact spaces (Christmann and Steinwart, 2010). Thus, when it comes to applying these powerful kernel methods to infinite discrete spaces in general, and biological sequences in particular, theoretical justification is lacking.

The situation is particularly challenging for kernels that are used in practice, which must capture real-world structure (the underlying biology) and be tractable computationally. Indeed, it is not entirely obvious *a priori* that strong flexibility guarantees can even be established for practical biological sequence kernels: in the case of graphs, for instance, kernels that are characteristic are at least as hard to compute as the graph isomorphism problem, so no polynomial time algorithm exists (Kriege et al., 2020; Gärtner et al., 2003). In other words, tractability and flexibility are not necessarily compatible.

Jorgensen and Tian (2015) established pioneering results on kernel flexibility for infinite discrete spaces, exploring the idea of kernels with discrete masses. We extend the scope and implication of their results dramatically, by (a) connecting the discrete mass property to the notions of flexibility important for machine learning applications, and (b) studying kernels relevant to biological sequences. More broadly, we make the theory of Jorgensen and Tian (2015) more “user-friendly”, by establishing easier routes to proving the discrete mass property, and elaborating its practical implications.

Kernels for biological sequences are mathematically related to kernels for time series data. Many of our constructions below start with kernels k_s that compare individual letters of the alphabet; these can be swapped with arbitrary kernels to compare sequences of any type of object. As well, many popular kernels for biological sequences can be seen as discretizations of kernels for time series data (Király and Oberhauser, 2019). While the flexibility properties of continuous time series kernels are well established (Hambly and

Lyons, 2010), those relations unfortunately do not transfer to biological sequences kernels in a straightforward way.

4. The discrete mass property

In this section we introduce the discrete mass property and describe its implications for kernel flexibility. We focus on properties necessary for procedures useful in biological sequence analysis. In particular, we show that if a kernel has discrete masses, it is (1) universal (an important property for consistent regression), (2) characteristic (important for consistent two-sample testing (Gretton et al., 2012; Fukumizu et al., 2007)) and (3) metrizes the space of distributions (important for optimizing sequence distributions (Li et al., 2017; Pronzato and Zhigljavsky, 2020; Huszár and Duvenaud, 2012; Futami et al., 2019)). Later, we will design biological sequence kernels that have discrete masses to satisfy these three properties.

We say a kernel has discrete masses if its Hilbert space includes delta functions at all points in the space S . Note that in this section we take S to be an arbitrary discrete space; later (starting in Section 6) we will specialize to the setting where S is sequence space.

Definition 1 (The discrete mass property) *We say k has discrete masses if $C_C(S) \subset \mathcal{H}_k$, where $C_C(S)$ is the set of all functions on S that are non-zero at only finitely many points. Since \mathcal{H}_k is a linear space, this is equivalent to $\delta_X \in \mathcal{H}_k$ for all $X \in S$.*

For ease of exposition, in this section we will assume that $k(X, X) = 1$ for all $X \in S$.¹

4.1 Universal kernels and function approximation

If the RKHS is large enough to describe any function in some very general set, we say it is universal. In particular, we focus on C_0 -universality, which says that the RKHS can approximate any function in C_0 , the set of all functions on S that vanish at infinity (Sriperumbudur et al., 2011).

Definition 2 (C_0 -universality) *Say $k_X \in C_0(S)$ for all $X \in S$. We say k is C_0 -universal if for every $f \in C_0(S)$ and $\epsilon > 0$ there is a $g \in \mathcal{H}_k$ such that $\|f - g\|_\infty < \epsilon$.*

Are common biological sequence kernels universal? Many are not, because they fail a simple criteria: they do not have infinite features (Leslie et al., 2002; Tsuda et al., 2002; Jaakkola et al., 2000).

Example 1 (The kmer spectrum kernel is not universal) *The L -mer spectrum kernel uses as features the number of times each subsequence (kmer) up to length L occurs in a sequence (Leslie et al., 2002). It is defined as $k(X, Y) = (\Phi(X)|\Phi(Y))$ where $\Phi(X)$ is a vector with entries $\Phi(X)_V$ specifying the number of times the kmer V appears in X , for all $V \in S$ such that $|V| \leq L$. Since the total number of features is finite, it is not universal.*

Although infinite features are necessary for a kernel to be universal, they are not sufficient: in Fig 1 (details in App. A) we gave an example of a kernel which has infinite features but is

1. This does not result in a loss of generality as we can replace, in the arguments below, the space of distributions $\mathcal{P}(S)$ with $\{\mu \in \mathcal{P}(S) \mid \int d\mu(X) \sqrt{k(X, X)} < \infty\}$ and replacing the space of functions that vanish at infinity $C_0(S)$ with $\{X \mapsto f(X) \sqrt{k(X, X)} \mid f \in C_0(S)\}$.

not universal. Instead, to prove our proposed new biological sequence kernels are universal, we will prove that they have discrete masses: since $C_C(S)$ is dense in $C_0(S)$ we get the following proposition.

Proposition 3 *Kernels with discrete masses are C_0 -universal.*

4.2 Characteristic kernels for distribution comparison

Besides regression, kernels can also be used to compare distributions. We can measure the difference between two distributions μ and ν using the maximum mean discrepancy.

Definition 4 (Maximum mean discrepancy) *Recall the embedding of a measure μ , denoted $\int d\mu(X)k_X$, is the element of \mathcal{H}_k that satisfies $(\int d\mu(X)k_X|g)_k = \int d\mu(X)g(X)$ for all $g \in \mathcal{H}_k$. The MMD is the norm of the difference in embeddings of two distributions μ, ν ,*

$$\text{MMD}_k(\mu, \nu) = \max_{\{f \in \mathcal{H}_k : \|f\|_k \leq 1\}} E_\mu f - E_\nu f = \left\| \int d\mu(X)k_X - \int d\nu(Y)k_Y \right\|_k, \quad (1)$$

where $\|f\|_k^2 = (f|f)_k$.

In order for the MMD to be able to tell the difference between *any* two distributions, it must be characteristic.

Definition 5 (Characteristic kernel) *We say a kernel k is characteristic if $\mu \mapsto \int d\mu(X)k_X$ is injective on $\mathcal{P}(S)$, the space of probability distributions on S .*

Note characteristicness is a strictly weaker condition than universality in general (Sriperumbudur et al., 2011).

MMD is useful not only for comparing two given distributions μ and ν , but also as an optimization objective that we can minimize to find a distribution ν that matches μ . Ideally, a good optimization objective should not only tell us if we have reached the right answer, but also if we are headed in the right direction. For this to hold, the kernel must metrize the space of distributions $\mathcal{P}(S)$.

Definition 6 (Metrizing $\mathcal{P}(S)$) *We say a kernel k metrizes $\mathcal{P}(S)$ if for every $\mu \in \mathcal{P}(S)$ and sequence $\nu_1, \nu_2, \dots \in \mathcal{P}(S)$, it holds that $\text{MMD}_k(\mu, \nu_n) \rightarrow 0$ implies $\nu_n \rightarrow \mu$.*

Note, for kernels on continuous space, the analogous property is sometimes called “metrizing weak convergence” (Sriperumbudur et al., 2010). On $\mathcal{P}(S)$, however, the weak topology is equivalent to the total variation topology.

Many popular biological sequence kernels are not characteristic and do not metrize $\mathcal{P}(S)$. For example, kernels with finitely many features, in addition to not being universal, are also not characteristic. As with universality, having an infinite number of features is also not enough to make a kernel characteristic; one can easily show this for the Hamming kernel in Fig. 1 (details in App. A). Another class of kernels built off representation learning methods are a popular choice for optimizing biological sequences, since low dimensional continuous optimization can be more tractable than high dimensional discrete optimization (Yang et al., 2018; Stanton et al., 2022; Notin et al., 2021). However, even very faithful representations, that preserve large amounts of information about sequence space, may not be able to metrize $\mathcal{P}(S)$.

Example 2 (Embedding kernels are not guaranteed to metrize $\mathcal{P}(S)$) Consider $S = \{\emptyset, A, AA, \dots\}$. We define a target distribution $\mu = \delta_A$. To define a kernel, we consider the injective embedding $F(A) = 0$ and $F(n \times A) = 1/n$ for $n > 1$, and use a radial basis function kernel in the embedding space, such that the complete embedding kernel is $k(X, Y) = \exp(-(F(X) - F(Y))^2)$. Intuitively, since the radial basis function kernel is characteristic over \mathbb{R} , and since the embedding F is one-to-one, the embedding kernel is characteristic (we prove this rigorously in Section 9).

However, this embedding kernel does not metrize $\mathcal{P}(S)$. If we try to minimize $\text{MMD}_k(\mu, \delta_X)$ with respect to X , we find that choosing longer and longer sequences brings the objective closer and closer to zero, $\text{MMD}(\mu, \delta_{n \times A}) \rightarrow 0$ as $n \rightarrow \infty$.

To design new biological sequence kernels that are guaranteed to be characteristic and to metrize $\mathcal{P}(S)$, we again turn to discrete masses.

Proposition 7 (Kernels with discrete masses are characteristic and metrize $\mathcal{P}(S)$) Say k is a kernel such that $k(X, X) = 1$ and $k_X \in C_0(S)$ for all $X \in S$.

$$k \text{ has discrete masses} \implies \text{MMD}_k \text{ metrizes } \mathcal{P}(S) \implies k \text{ is characteristic.}$$

Proof The second implication is clear. Now say k has discrete masses and $\mu, \nu_1, \nu_2, \dots \in \mathcal{P}(S)$ such that $\text{MMD}_k(\mu, \nu_n) \rightarrow 0$. For each $X \in S$, $\delta_X, -\delta_X \in \mathcal{H}_k$, so by Equation 1 we have $\|\delta_X\|_k^{-1} |\mu(X) - \nu_n(X)| \leq \text{MMD}_k(\mu, \nu_n) \rightarrow 0$. Thus, $\nu_n \rightarrow \mu$. \blacksquare

The discrete mass property thus guarantees that a kernel is universal, characteristic and metrizes $\mathcal{P}(S)$, making the kernel a good choice for a wide range of applications.²

4.3 Degenerate examples

The remainder of the paper will be concerned with designing biological sequence kernels with discrete masses. Before studying complicated kernels, however, we first consider two simple but degenerate kernels with discrete masses, and explain why they are unsatisfactory.

One kernel with discrete masses is the identity kernel.

Example 3 (The identity kernel has discrete masses) Consider a kernel $k(X, Y) = \delta_X(Y)$ for all $X, Y \in S$. This kernel has discrete masses, since $k_X = \delta_X$ for all X . The problem is that it leads to poor generalization.

As this example demonstrates, the mere fact that a kernel has discrete masses does not mean it is a good modeling choice. Instead, we want kernels that not only have discrete masses but also capture biological notions of sequence similarity.

Another way to construct flexible kernels is to assume that S is finite.

Proposition 8 Say $|S| < \infty$. If k is strictly positive definite then it has discrete masses.

2. Note that the discrete mass property is stronger than all three of these properties, as there exist kernels that are universal and metrize $\mathcal{P}(S)$ but do not have discrete masses (Appendix B).

Proof For any $X, Y \in S$, $\delta_X(Y) = \sum_{X' \in S} \alpha_{X'} k(X', Y)$ for $\alpha = K^{-1} \delta_X$ where K is the Gram matrix of all points in S . \blacksquare

Focusing on a finite S , while tempting from the perspective of theoretical convenience, can lead to unreliable methods in practice.

Example 4 (Sequence space with bounded length) *If sequence space S includes only sequences X with length less than some maximum L_{\max} , that is $S = \cup_{L=0}^{L_{\max}} \mathcal{B}^L$, and the kernel is strictly positive definite, then the kernel has discrete masses. The problem is that we want our kernel methods to be reliable even as we observe or optimize longer and longer sequences. As well, if we choose L_{\max} to be astronomical, so that Proposition 8 is technically met, studying infinite sequence space $S = \cup_{L=0}^{\infty} \mathcal{B}^L$ is useful in that it forces us to consider the behavior of our methods as sequences grow in length.*

5. Characterizing and manipulating kernels with discrete masses

We have seen that kernels with discrete masses have strong guarantees on their ability to approximate functions and distinguish distributions. However only a small number of kernels, with limited relevance for biological sequences, have been shown previously to have discrete masses on infinite data spaces (Jorgensen and Tian, 2015). In this section we develop theoretical tools that can be used to prove that kernels have discrete masses. Section 5.1 describes conditions that are equivalent to having discrete masses; Section 5.2 describes transformations of kernels that preserve the discrete mass property. We will later apply these techniques to design new biological sequence kernels with discrete masses. Note in this section S is still an arbitrary infinite discrete space, not necessarily the space of sequences.

5.1 Equivalent formulations of the discrete mass property

In this section we describe two equivalent formulations of the discrete mass property.

5.1.1 CONDITIONS ON THE SPAN OF KERNEL EMBEDDINGS

The first formulation puts conditions on $\text{span}\{k_Y\}_{Y \neq X}$ that guarantee $\delta_X \in \mathcal{H}_k$. The intuition is as follows. Consider some element of the span, $\sum_{Y' \in S} \alpha_{Y'} k_{Y'} \in \text{span}\{k_Y\}_{Y \in S}$. If $\delta_X \in \mathcal{H}_k$ then $(\delta_X | \sum_{Y' \in S} \alpha_{Y'} k_{Y'})_k = \sum_{Y' \in S} \alpha_{Y'} \delta_X(Y') = \alpha_X$, that is, we can think of δ_X as a function that takes every element of $\text{span}\{k_Y\}_{Y \in S}$ to the coefficient in front of k_X . In order for this function to exist, it must be (1) well defined and (2) bounded. For the function to be well defined, k_X must be linearly independent from $\{k_Y\}_{Y \neq X}$. For the function to be bounded, it must also be difficult to approximate k_X using elements in $\text{span}\{k_Y\}_{Y \neq X}$. This intuition can be formalized as follows.

Proposition 9 *Let $X \in S$, and call \mathcal{K}_X the closure of $\text{span}\{k_Y\}_{Y \neq X}$. $\delta_X \in \mathcal{H}_k$ if and only if $\mathcal{H}_k \neq \mathcal{K}_X$, or in other words, $k_X \notin \mathcal{K}_X$.*

Proof Say $\delta_X \in \mathcal{H}_k$. If $Y \neq X$, $(\delta_X | k_Y)_k = \delta_X(Y) = 0$. On the other hand, $(\delta_X | k_X)_k = 1$. Thus, $k_X \notin \mathcal{K}_X$.

On the other hand, say $k_X \notin \mathcal{K}_X$. Then the orthogonal complement of \mathcal{K}_X is exactly one dimensional. Let $\phi : \mathcal{H}_k \rightarrow \mathbb{R}$ be the linear function projecting \mathcal{H}_k to the orthogonal

complement of \mathcal{K}_X , scaled so that $\phi(k_X) = 1$. Then $\phi(k_Y) = 0$ if $Y \neq X$, so, $\phi(k_Y) = \delta_X(Y)$ for all $Y \in S$. By the Riesz representation theorem, since ϕ is continuous, there is a $f \in \mathcal{H}_k$ such that for all $Y \in S$, $f(Y) = (f|k_Y)_k = \phi(k_Y) = \delta_X(Y)$, so $f = \delta_X$. \blacksquare

One implication of this result is that continuous kernels on Euclidean space cannot have discrete masses.

Proposition 10 (Continuous kernels on Euclidean space do not have discrete masses)

If the kernel $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ is a continuous function, it does not have discrete masses.

Proof Consider a sequence $Y_1, Y_2, \dots \in \mathbb{R}^D$ that converges to $X \in \mathbb{R}^D$. Then

$$\|k_{Y_n} - k_X\|_k^2 = k(X, X) + k(Y_n, Y_n) - 2k(X, Y_n) \rightarrow 2k(X, X) - 2k(X, X) = 0.$$

In other words, k_X is in the closure of $\text{span}\{k_Y\}_{Y \neq X}$. \blacksquare

The theory of kernels with discrete masses is thus unique to infinite discrete spaces such as biological sequence space.

5.1.2 CONDITIONS ON THE KERNEL'S GRAM MATRIX

A second formulation of discrete masses comes from pioneering work by Jorgensen and Tian (2015). We give a concise restatement of the proof of their result here. The basic idea builds off of Proposition 8, which says that on finite discrete spaces a strictly positive definite kernel has an invertible Gram matrix and thus discrete masses. On an infinite space, we consider the "invertibility" of a sequence of Gram matrices, defined on larger and larger finite subsets of S , to ensure the kernel has discrete masses.

Theorem 11 (Jorgensen and Tian, 2015) *Let $X \in S$ and let k be a strictly positive definite kernel on S . For a finite subset $B \subset S$ with $X \in B$, define the Gram matrix K_B indexed by B such that $K_{B,Y,Y'} = k(Y, Y')$ for $Y, Y' \in B$. Let $C_{B,X} = \sqrt{(K_B^{-1})_{X,X}}$. Then, $\delta_X \in \mathcal{H}_k$ if and only if $\sup_B C_{B,X} < \infty$ where the supremum is over all finite $B \subset S$. In this case, $\|\delta_X\|_k = \sup_B C_{B,X}$.*

Proof Define a functional ϕ_X on $\text{span}\{k_Y\}_{Y \in S}$ that is $\alpha_X k_X + \sum_{Y \neq X} \alpha_Y k_Y \mapsto \alpha_X$. Note ϕ_X is well defined since k is strictly positive definite. Now, if $\delta_X \in \mathcal{H}_k$, then for all $f \in \text{span}\{k_Y\}_{Y \in S}$, we have $(\delta_X|f)_k = \phi_X(f)$ so that ϕ_X is bounded. On the other hand, if ϕ_X is bounded, it can be continuously extended to all of \mathcal{H}_k and must be equal to $(\cdot|f)_k$ for some $f \in \mathcal{H}_k$ by the Riesz representation theorem. Then, $f(Y) = (k_Y|f)_k = \phi_X(k_Y) = \delta_X(Y)$, so, $f = \delta_X$. We will show that C_B is the norm of ϕ_X restricted to $\text{span}\{k_Y\}_{Y \in B}$ and the result will follow.

$\text{span}\{k_Y\}_{Y \in B}$ is a finite dimensional space with $\{k_Y\}_{Y \in B}$ as a basis, and inner product $(v|w) = v^T K_B w$ for $v, w \in \mathbb{R}^B$ in this basis. Calling e_X the indicator vector for X , $\phi_X(v) = (e_X^T K_B^{-1} | v)$. Finally, we see that the square norm of ϕ_X is $(e_X^T K_B^{-1} | e_X^T K_B^{-1}) = (K_B^{-1})_{X,X} = C_B^2$. \blacksquare

5.2 Transformations that preserve the discrete mass property

In this section we describe how to construct new kernels with discrete masses out of existing kernels with discrete masses. This allows us to construct large families of related kernels that all have flexibility guarantees. Many of the transformations we consider are standard for all kernels (as described in e.g. Schölkopf and Smola (2021)), and it will be straightforward to see they preserve discrete masses. We also introduce a new transformation — reparameterizing alphabets — that is particularly relevant for the analysis of biological sequence kernels.

5.2.1 SUMMING

First we consider linear combinations of kernels with discrete masses. The following result says that as long as one kernel in this sum has discrete masses, then so does the entire summed kernel.

Proposition 12 *Say Λ is a measurable space and $(k_\lambda)_{\lambda \in \Lambda}$ is a family of kernels. Assume for any $X, Y \in S$, $\lambda \mapsto k_\lambda(X, Y)$ is measurable. Say ν is a positive, nonzero measure on Λ with $\int d\nu(\lambda) k_\lambda(X, X) < \infty$ for all $X \in S$ and $\nu(\{\lambda \mid k_\lambda \text{ has discrete masses}\}) > 0$, i.e. ν has positive mass on kernels with discrete masses.³ Then, $k_\nu = \int k_\lambda d\nu(\lambda)$ is a kernel on S that has discrete masses.*

Proof We consider the possibility that k_ν does not have discrete masses and show that this leads to a contradiction. By Proposition 9, there is some $X \in S$ and sequence $\mu_l = \delta_X - \sum_{Y \neq X} \alpha_{l,Y} k_Y$, where $\sum_{Y \neq X} \alpha_{l,Y} k_Y \in \text{span}\{k_Y\}_{Y \neq X}$, such that $\|\int k_{\nu,Z} d\mu_l(Z)\|_{k_\nu} \rightarrow 0$ as $l \rightarrow \infty$. Then,

$$\begin{aligned} \left\| \int k_{\nu,Z} d\mu_l(Z) \right\|_{k_\nu}^2 &= \int d\mu_l(Z_1) \int d\mu_l(Z_2) k_\nu(Z_1, Z_2) \\ &= \int d\nu(\lambda) \int d\mu_l(Z_1) \int d\mu_l(Z_2) k_\lambda(Z_1, Z_2) \\ &= \int d\nu(\lambda) \left\| \int k_{\lambda,Z} d\mu_l(Z) \right\|_{k_\lambda}^2. \end{aligned} \tag{2}$$

Thus, by Fatou's lemma,

$$\int d\nu(\lambda) \liminf_{l \rightarrow \infty} \left\| \int k_{\lambda,Z} d\mu_l(Z) \right\|_{k_\lambda}^2 = 0.$$

In particular, there is a λ such that k_λ has discrete masses and $\liminf_{l \rightarrow \infty} \left\| \int k_{\lambda,Z} d\mu_l(Z) \right\|_{k_\lambda}^2 = 0$, which is a contradiction. \blacksquare

Corollary 13 *Let k_0, k_1, \dots, k_N be kernels such that k_0 has discrete masses and $\alpha_0, \dots, \alpha_N > 0$. Then $\sum_{n=0}^N \alpha_n k_n$ is a kernel with discrete masses.*

3. If $\{\lambda \mid k_\lambda \text{ has discrete masses}\}$ is not measurable, then we instead require that $\nu(C) > 0$ for all $C \supset \{\lambda \mid k_\lambda \text{ has discrete masses}\}$.

5.2.2 CHANGING DOMAINS

Next we consider using different kernels over different regions of sequence space. The following result says that if a kernel has discrete masses over all of S , it also has discrete masses when restricted to just one region of S . As well, if we have separate kernels with discrete masses over separate orthogonal regions of S , we can then combine them to construct a new kernel with discrete masses over all of S .

Proposition 14 *Say $\{W_V\}_V$ is a collection of disjoint subsets of S such that $S = \cup_V W_V$. If k has discrete masses, then it also has discrete masses when restricted to any W_V . On the other hand, if k has discrete masses when restricted to each W_V , and $k(X, X') = 0$ for any $X \in W_V, X' \in W_{V' \neq V}$, then k has discrete masses over S .*

Proof First assume k has discrete masses over S . A standard property of kernels, is that the Hilbert space of a kernel restricted to a domain W_V , that is $\mathcal{H}_{k|_{W_V}}$, is the closure of $\text{span}\{k_Y\}_{Y \in W_V}$ in the original Hilbert space \mathcal{H}_k . Now consider any $X \in W_V$. By Proposition 9, k_X is not in the closure of $\text{span}\{k_Y\}_{Y \in S \mid Y \neq X}$, so it's also not in the closure of $\text{span}\{k_Y\}_{Y \in W_V \mid Y \neq X}$. Thus, applying Proposition 9 again, k has discrete masses over W_V .

Now consider the case where k has discrete masses when restricted to each W_V . Assume $\delta_X \notin \mathcal{H}_k$ for some $X \in W_V$; we will show this leads to a contradiction. By Proposition 9, there exists a sequence of functions $f_1 = \sum_{Y \neq X} \alpha_{1,Y} k_Y, f_2 = \sum_{Y \neq X} \alpha_{2,Y} k_Y, \dots \in \text{span}\{k_Y\}_{Y \neq X}$ such that $f_m \rightarrow k_X$ as $m \rightarrow \infty$. Let $f_{V,m}$ be the orthogonal projection of f_m onto $\text{span}\{k_Y\}_{Y \in W_V}$, that is $f_{V,m} = \sum_{Y \in W_V} \mathbb{1}(Y \in W_V) \alpha_{m,Y} k_Y$. Note that for every m , we have $f_{V,m} \in \text{span}\{k_Y\}_{Y \in W_V \mid Y \neq X}$, so $f_{V,m}$ is in the Hilbert space $\mathcal{H}_{k|_{W_V}}$ of the kernel restricted to W_V . Define $f_{V,m}^\perp = f_m - f_{V,m} = \sum_{Y \neq X} \mathbb{1}(Y \notin W_V) \alpha_{m,Y} k_Y$. Now $(f_{V,m}^\perp | f_{V,m})_k = 0$ and $(f_{V,m}^\perp | k_X)_k = 0$, so

$$\|f_{V,m} - k_X\|_k^2 \leq \|f_{V,m}^\perp\|_k^2 + \|f_{V,m} - k_X\|_k^2 = \|f_m - k_X\|_k^2 \rightarrow 0.$$

In other words, k_X is in the closure of $\text{span}\{k_Y\}_{Y \in W_V \mid Y \neq X}$. This implies that the kernel k restricted to W_V does not have discrete masses, a contradiction. \blacksquare

5.2.3 TENSORIZING

We next consider tensorizing kernels, so that they can be applied to pairs of sequences (Fukumizu et al., 2007). If we have two kernels k_1 and k_2 on S , the tensorized kernel $k_1 \otimes k_2$ is $k_1 \otimes k_2((X_1, X_2), (Y_1, Y_2)) = k_1(X_1, Y_1)k_2(X_2, Y_2)$ for $X_1, X_2, Y_1, Y_2 \in S$. Tensorization preserves discrete masses.

Corollary 15 *Let k_1 and k_2 be kernels on S with discrete masses. Then $k_1 \otimes k_2$ is a kernel on S^2 with discrete masses.*

Proof A standard property of kernels, which follows from the results in Schölkopf and Smola (2021), says that if $f_1 \in \mathcal{H}_{k_1}$ and $f_2 \in \mathcal{H}_{k_2}$, there is a $f_1 \otimes f_2 \in \mathcal{H}_{k_1 \otimes k_2}$ such that for $(X_1, X_2) \in S^2$, we have $f_1 \otimes f_2(X_1, X_2) = f_1(X_1)f_2(X_2)$. Say $X, Y \in S$. Then

$$\delta_{(X,Y)} = \delta_X \otimes \delta_Y \in \mathcal{H}_{k_1 \otimes k_2}. \quad \blacksquare$$

5.2.4 TILTING

Next we consider re-weighting kernels to emphasize or de-emphasize certain areas of sequence space. More precisely, we consider *tilting* a kernel k by some function $A : S \rightarrow (0, \infty)$ to obtain a new kernel $k^A(X, Y) = A(X)k(X, Y)A(Y)$ for $X, Y \in S$. The discrete mass property is preserved after tilting.

Proposition 16 *If k is a kernel on S that has discrete masses, then k^A has discrete masses.*

Proof A standard property of kernels, which follows from the results in Schölkopf and Smola (2021), is that if we have a function $f(\cdot)$ in the original RKHS, \mathcal{H}_k , then the function $A(\cdot)f(\cdot)$ is in the RKHS of the tilted kernel, \mathcal{H}_{k^A} , and has the same norm, $\|f\|_k = \|Af\|_{k^A}$. In other words, $f \mapsto Af$ is an isometric isomorphism of \mathcal{H}_k to \mathcal{H}_{k^A} . So for any $X \in S$, we have $A\delta_X \in \mathcal{H}_{k^A}$. We can always multiply a function in an RKHS by a finite scalar. Thus, $\delta_X = \frac{1}{A(X)}A(\cdot)\delta_X(\cdot) \in \mathcal{H}_{k^A}$. \blacksquare

5.2.5 REPARAMETERIZING ALPHABETS

Finally, we look at a novel transformation of kernels, specific to biological sequences, that involves “reparameterizing” the alphabet \mathcal{B} . The basic idea starts from representing letters in the alphabet \mathcal{B} as one-hot encoded vectors, i.e. we treat each $b \in \mathcal{B}$ as a vector of length $|\mathcal{B}|$ with zeros everywhere except at one position. The alphabet \mathcal{B} thus forms a basis of $\mathbb{R}^{\mathcal{B}}$. However, we may also consider an alternative basis $\tilde{\mathcal{B}}$ of $\mathbb{R}^{\mathcal{B}}$. This alternative basis gives rise to an alternative set of sequences $\tilde{S} = \cup_{L=0}^{\infty} \tilde{\mathcal{B}}^L$. By treating S and \tilde{S} as sets of vectors, there will be a natural way to extend a kernel on S to \tilde{S} . We will see that the property of discrete masses is invariant to this change in basis: if the kernel has discrete masses over S , it also has discrete masses over \tilde{S} , and vice versa. (Note in this section and all following sections, S will be the set of sequences, rather than an arbitrary infinite discrete space.)

To be more precise, we must define what it means to apply a sequence kernel to vector encoding of a sequence. Any sequence Y in $S = \cup_{L=0}^{\infty} \mathcal{B}^L$ can be represented as a one-hot encoding, which consists of a vector $V \in \mathbb{R}^{|Y| \times \mathcal{B}}$ such that $V_{(l)} \in \mathbb{R}^{\mathcal{B}}$ is the one-hot encoding of the letter $Y_{(l)}$. If the vector $V \in \mathbb{R}^{L \times \mathcal{B}}$ is a one-hot encoding of a sequence then we define its embedding into \mathcal{H}_k as the embedding of the sequence it encodes. We can write this using the formula

$$k_V = \sum_{X \in S: |X|=L} \left(\prod_{l=0}^{L-1} V_{(l), X_{(l)}} \right) k_X. \quad (3)$$

Thus if V is a one-hot encoding of Y , we recover $k_V = k_Y$. The kernel has not changed; all we have done is rewrite it to embed vector encoded sequences.

We now apply the same kernel to different sequences that use different encodings, which are not one-hot. In particular, let $\tilde{\mathcal{B}}$ be an alternative basis of $\mathbb{R}^{\mathcal{B}}$. Each sequence \tilde{Y} in $\tilde{S} = \cup_{L=0}^{\infty} \tilde{\mathcal{B}}^L$ can be represented by a vector encoding $\tilde{V} \in \mathbb{R}^{|\tilde{Y}| \times \mathcal{B}}$, where $\tilde{V}_{(l)}$ is the encoding

of the letter $\tilde{Y}_{(l)}$. We define the kernel embedding of \tilde{Y} by plugging \tilde{V} into Equation 3; it consists of a linear combination of kernel embeddings k_X from each sequence $X \in S$ with $|X| = |\tilde{V}|$. We will show that if the kernel k has discrete masses over S , and we apply k to \tilde{S} , it will still have discrete masses. We call this shift from the alphabet \mathcal{B} to $\tilde{\mathcal{B}}$ a “reparameterization” of the kernel’s alphabet.

Intuitively, in the case of proteins, we can think of the reparameterized alphabet as a set of amino acid properties, such as mass, charge, etc. Each amino acid is a letter in the original alphabet \mathcal{B} , while each property is a letter in the reparameterized alphabet $\tilde{\mathcal{B}}$; since both alphabets form bases of $\mathbb{R}^{\mathcal{B}}$, each amino acid can be described as a linear combination of properties. We can analyze the flexibility of a kernel over sequences of amino acids by analyzing the flexibility of the same kernel applied to sequences of amino acid properties. This is useful theoretically for studying complex kernels.

Proposition 17 *Say k is a strictly positive definite kernel on S , and $\tilde{\mathcal{B}}$ is a basis of $\mathbb{R}^{\mathcal{B}}$. Then k has discrete masses as a kernel on \tilde{S} if and only if it has discrete masses as a kernel on S .*

Proof Both \mathcal{B} and $\tilde{\mathcal{B}}$ are bases of $\mathbb{R}^{\mathcal{B}}$, so the kernel over S is a reparameterization of the kernel over \tilde{S} and vice versa. Thus, we only need to show that k has discrete masses as a kernel on S if it has discrete masses as a kernel on \tilde{S} . First, note that reparameterizing the alphabet does not change the span of the kernel embedding vectors, $\text{span}(k_V)_{V \in (\tilde{\mathcal{B}})^L} = \text{span}(k_{V'})_{V' \in (\mathcal{B})^L}$, since both \mathcal{B} and $\tilde{\mathcal{B}}$ are bases of $\mathbb{R}^{\mathcal{B}}$. Now, consider some length L . If $X \in S$ with $|X| = L$, then $k_X = \sum_{Y \in \tilde{S} \mid |Y|=L} \alpha_{X,Y} k_Y$ for some $\alpha_{X,Y}$. As k is strictly positive definite, $\{k_X\}_{|X|=L}$ is a linearly independent set, so α is an invertible square matrix with dimension $|\mathcal{B}|^L \times |\mathcal{B}|^L$. Let $f_X = \sum_{Y \in \tilde{S} \mid |Y|=L} (\alpha^{-1})_{X,Y} \delta_Y$. Then $f_X(Z) = 0$ for $Z \in S$ with $|Z| \neq L$, and if $|Z| = L$, $f_X(Z) = \sum_{Y \in \tilde{S} \mid |Y|=L} (\alpha^{-1})_{X,Y} \alpha_{Z,Y} = \delta_X(Z)$. Thus $\delta_X \in \mathcal{H}_k$. ■

Throughout the rest of the paper, for any kernel on S and any $V, W \in \cup_{L=0}^{\infty} \mathbb{R}^{L \times \mathcal{B}}$ we will write $k(V, W) = (k_V | k_W)_k$ where k_V and k_W are defined by Eqn 3.

6. Position-wise comparison kernels

In this section, we design kernels with discrete masses that compare sequences position-by-position. We saw in Fig 1 and 4.2 that the Hamming kernel, the weighted degree kernel, and other related kernels that compare sequences position-by-position are neither universal nor characteristic. Here, we develop alternative kernels that capture the same biological ideas but are also highly flexible.

Position-wise sequence comparison is ubiquitous in biology, and has strong biological justification for many problems. For instance, a common observation is that nucleotides or amino acids at a specific position have a specific biological function; for instance, the amino acids at a few particular positions may chemically react to form a fluorophore, making the protein fluorescent. So, when predicting phenotype from sequence, a reasonable measure of sequence similarity is one that compares sequences position-by-position, as is done in the Hamming kernel. Moreover, a very common form of mutation during evolution is

a substitution, which switches one letter for another. Thus, a position-wise measure of sequence similarity can capture evolutionary distance as well as phenotypic distance, which may be desirable, for instance, when comparing sequence distributions.

Our new kernels preserve existing notions of position-wise sequence similarity. For example, as two sequences differ more and more according to the Hamming kernel, they will also differ more and more according to our proposed replacement for the Hamming kernel. What we modify is not the measure of sequence similarity but instead the functional form of the dependence of the kernel on sequence similarity, i.e. how exactly a change in the similarity of X and Y translates into a change in $k(X, Y)$. For existing kernels, the functional form rarely has strong biological justification, and instead is often motivated solely by convenience. Our results demonstrate that the functional form in fact matters a great deal for the reliability of kernel methods for biological sequences.

We start by studying a simple kernel that compares sequences position-by-position. We then use this “base” kernel to derive many other varieties of position-wise comparison kernels, using the transformations developed in Section 5.2. The base position-wise comparison kernel is defined as the product of individual kernels applied to the letters at each position.

Definition 18 (Base position-wise comparison kernel) *We represent each sequence $X \in S$ as terminated by an infinite tail of stop symbols $\$$. Let k_s be a strictly positive definite kernel on letters $\mathcal{B} \cup \{\$\}$ with $k_s(\$, \$) = 1$. Now, the base position-wise comparison kernel is*

$$k(X, Y) = \prod_{l=0}^{\infty} k_s(X_{(l)}, Y_{(l)}). \quad (4)$$

Note that because $k_s(\$, \$) = 1$, the infinite product is always finite. This kernel compares the sequences X and Y at each position l , according to k_s . For example, one natural choice of k_s is to set $k_s(b, b') = 1$ if $b = b'$ and $k_s(b, b') = e^{-\lambda}$ if $b \neq b'$, for $\lambda > 0$. This gives the “exponential Hamming kernel”,

$$k(X, Y) = \exp \left(-\lambda \sum_l \mathbb{1}(X_{(l)} \neq Y_{(l)}) \right) = \exp(-\lambda d_H(X, Y)),$$

where d_H is the Hamming distance. Recall that the Hamming kernel of lag $L = 1$ takes the form $k(X, Y) = |X| \vee |Y| - d_H(X, Y)$. Thus, the two kernels both measure sequence similarity using the Hamming distance, but differ in the functional form of their dependence.

The base position-wise comparison kernel, unlike the weighted degree kernel, has discrete masses.

Theorem 19 *The base position-wise comparison kernel has discrete masses.*

Proof Our proof strategy will be to reparameterize the alphabet \mathcal{B} (using Proposition 17) such that the RKHS \mathcal{H}_k decomposes into a product of orthogonal hyperplanes, each spanned by a subset of sequences. We prove that the kernel restricted to each of these hyperplanes takes a simple form, and has discrete masses. We then apply Proposition 14 to merge the separate hyperplanes and prove the result.

We start by reparameterizing the base position-wise comparison kernel, k . First note that k is a strictly positive definite kernel by the Schur product theorem. Define K as the

matrix with $K_{b',b} = k_s(b, b')$ for $b, b' \in \mathcal{B}$. Call $t \in \mathbb{R}^{\mathcal{B}}$ the vector with $t_b = k_s(b, \$)$ for $b \in \mathcal{B}$. Let $U = K^{-1}t$ and call $\sigma = U^T K U$. If $V, W \in \mathbb{R}^{\mathcal{B}}$ define $k_{s,V} = \sum_{b \in \mathcal{B}} V_b k_{s,b}$, $k_{s,W}$ analogously, $k_s(V, W) = (k_{s,V} | k_{s,W})_{k_s}$ and $k_s(V, \$) = (k_{s,V} | k_{s,\$})_{k_s}$. Then if $V \in \mathbb{R}^{\mathcal{B}}$ $k_s(V, U) = V^T K U = V^T t = k_s(V, \$)$. Note in particular that since k_s is strictly positive definite, we have the strict Cauchy-Schwartz inequality,

$$\sigma = k_s(U, U) = k_s(U, \$) < \sqrt{k_s(U, U) k_s(\$, \$)} = \sqrt{\sigma},$$

which can only be the case if $\sigma < 1$. Let $B_1, \dots, B_{|\mathcal{B}|-1} \in \mathbb{R}^{\mathcal{B}}$ be chosen so that $\{U, B_1, \dots, B_{|\mathcal{B}|-1}\}$ is an orthogonal basis of the vector space $\mathbb{R}^{\mathcal{B}}$ when using the dot product $(v|x) = v^T K w$, with $(B_i | B_i) = 1$ for all i . Thus, $k_s(B_i, B_j) = \delta_i(j)$ for all i, j and $k_s(U, B_i) = 0$ for all i . Call $\tilde{\mathcal{B}} = \{U, B_1, \dots, B_{|\mathcal{B}|-1}\}$ and \tilde{S} the set of sequences made up of letters in $\tilde{\mathcal{B}}$. By Proposition 17, if we can show that the kernel k has discrete masses on \tilde{S} , we know that it has discrete masses on S .

We now break up \tilde{S} into separate domains W_V . Recall the base position-wise comparison kernel takes the form $k(V, V') = \prod_{l=0}^{\infty} k_s(V_l, V'_l)$. When applied to sequences in the reparameterized alphabet, i.e. $V, V' \in \tilde{S}$, we have for all $b, b' \in \tilde{\mathcal{B}} \cup \{\$\}$ that $k_s(b, b') = 0$ if $b \neq b'$ except $k_s(U, \$) = \sigma$. Thus, if $|X| \geq |Y| = L$ for $X, Y \in \tilde{S}$ then $k(X, Y) \neq 0$ if and only if the first L letters of X and Y are identical and all the letters of X after position L are U . If $V \in \tilde{S}$ and does not end in U , define $W_V = \{V + m \times U\}_{m=0}^{\infty}$ to be all the sequences in \tilde{S} that start with V and have a tail of U s. Then if $V' \in \tilde{S}$ is a different sequence that does not end in U , $W_{V'}$ is orthogonal to W_V in \mathcal{H}_k . Thus \mathcal{H}_k is made up of orthogonal hyperplanes spanned by the sets $\{W_V\}_V$. By Proposition 14, k has discrete masses if and only if k has discrete masses when restricted to each W_V .

We now show that the kernel applied to each W_V has discrete masses. Note, since $k_s(U, \$) = k_s(U, U) = \sigma$, we have $k(V + m \times U, V + m' \times U) = k(V, V) k(m \times U, m' \times U) = k(V, V) \sigma^{m \vee m'}$. Thus k restricted to W_V is equivalent to the kernel $k^{(n)}(m, m') = \sigma^{m \vee m'}$ applied to $m, m' \in \mathbb{N}$, times a constant, $k(V, V)$. We therefore just need to show that $k^{(n)}(m, m')$, a one-dimensional kernel defined over the natural numbers, has discrete masses. We prove this by induction. First, noting $\sigma < 1$, let $f = (\sigma - 1)^{-1}(k_1^{(n)} - k_0^{(n)})$. We see that $\delta_0 = f$, so $\delta_0 \in \mathcal{H}_k$. Now assume $\delta_0, \dots, \delta_{x-1} \in \mathcal{H}_k$ for some $x \in \mathbb{N}$. Let $f_x = (\sigma - 1)^{-1}(k_{x+1}^{(n)} - k_x^{(n)})$. We see that $f_x(z) = 0$ if $z > x$ and $f_x(x) = 1$. Thus $\delta_x = f_x - \sum_{z=0}^{x-1} f(z) \delta_z$, so $\delta_x \in \mathcal{H}_k$. Invoking Proposition 14, the proof is complete. ■

We can leverage the base position-wise comparison kernel to construct more complex position-wise comparison kernels that address specific modeling challenges, and also have discrete masses. In each of these examples, we use for simplicity $k_s(b, b') = \exp(-\lambda \mathbb{1}(b \neq b'))$, but one can just as well use any strictly positive definite kernel k_s , for example one which compares b and b' based on amino acid similarity.

Example 5 (Heavy tailed Hamming kernels) *Practical challenge:* The exponential Hamming kernel is “thin-tailed” in the sense that it decreases quickly (exponentially) as the Hamming distance increases. Consider a data set $\{X_1, X_2, \dots\}$ consisting of distantly related sequences, and the Gram matrix K with entries $K_{ij} = k(X_i, X_j)$. Thin tails can lead to “diagonal dominance”, where the diagonal entries K_{ii} are much larger than the

off-diagonal entries, K_{ij} for $i \neq j$. The kernel may therefore behave much like the identity kernel (Example 3), and exhibit similarly poor generalization when used in a Gaussian process or other regression method. Fixing diagonal dominance by tuning λ is difficult; typically the only other regime available is one in which the matrix K is nearly constant, i.e. $K_{i,j} \approx \text{constant}$ for all i, j . In this case, the regression will make approximately the same prediction everywhere, and still exhibit poor generalization.

Proposed kernel: To address diagonal dominance, we propose a heavy-tailed kernel, whose value decays slowly as sequence distance increases. Take $\beta, C > 0$. Then the “inverse multiquadric Hamming kernel” is defined by integrating over the bandwidth parameter λ ,

$$k(X, Y) = (C + d_H(X, Y))^{-\beta} = \Gamma(\beta)^{-1} \int_0^\infty d\lambda \left(\lambda^\beta e^{-C\lambda} \right) e^{-\lambda d_H(X, Y)}, \quad (5)$$

where Γ is the gamma function. As Hamming distance increases, this kernel decreases according to a power-law instead of exponentially, i.e. its tails are very heavy. By Proposition 12, this kernel has discrete masses.

Example 6 (k-mer Hamming kernels) Practical challenge: The exponential Hamming kernel considers two sequences similar if they have the same letter in the same position. However, the biological function of a sequence can depend not just on whether it has a certain letter in a certain position, but, further, on whether it has a certain string of letters (such as a motif) at a certain position. It may therefore be more appropriate to judge the similarity of two sequences based on whether they have the same kmers at the same position, rather than single letters. This is the idea behind the Hamming kernel of lag $L > 1$. The problem is that this kernel is neither characteristic nor universal (Appendix A).

Proposed kernel We propose a kernel that uses the same measure of sequence similarity as the Hamming kernel of lag L , but takes a different functional form, which ensures it has discrete masses. Let L be the context size, and define $\tilde{\mathcal{B}} = \mathcal{B}^L$ to be the set of sequences of length L . We will treat $\tilde{\mathcal{B}}$ as an alphabet, and consider the set of sequences made from this alphabet, \tilde{S} . Now consider the kernel $\tilde{k}(X, Y) = (C + d_H(X, Y))^{-\beta}$ on \tilde{S} for some $C, \beta > 0$; recall that this kernel has discrete masses. Each sequence in S (the original sequence space) can be uniquely embedded in \tilde{S} by the mapping $i : S \rightarrow \tilde{S}$, defined as

$$h(X) = (X_{(:L)}, X_{(1:L+1)}, X_{(2:L+2)}, \dots),$$

that is, h maps each sequence to its sequence of L -mers in order. Note h is injective, but it is not surjective. By Proposition 14, \tilde{k} restricted to the image of h has discrete masses. Thus the “inverse multiquadric Hamming kernel of lag L ”,

$$k(X, Y) = \tilde{k}(h(X), h(Y)) = \left(C + \sum_{l=0}^{|X| \vee |Y| - 1} \mathbb{1}(X_{(l:l+L)} \neq Y_{(l:l+L)}) \right)^{-\beta},$$

which is defined for $X, Y \in S$, has discrete masses. Note that the sum in the above expression is precisely the Hamming kernel of lag L . In other words, the notion of sequence similarity has not changed, but the functional form of the kernel has, and this gives the new kernel discrete masses.

Example 7 (Centre-justified Hamming kernels) *Practical challenge* The exponential Hamming kernel compares sequences position-by-position starting from position one. This is sensible when position one is a good reference point, such as the start of a gene. In some cases, however, we are interested in a region of sequence to either side of a reference point, for instance the DNA sequence upstream as well as downstream of the start codon (i.e. both promoter and coding regions). For example, rather than comparing sequences as,

$$\begin{array}{c} |AACTTCT\$ \$ \$ \$ \dots \\ |GGACTTCTCT\$ \dots \end{array}$$

where $|$ marks the reference point, we want to compare them as,

$$\begin{array}{c} \dots \$ \$ AACT|TCT\$ \$ \$ \dots \\ \dots \$ \$ GACT|TCTCT\$ \dots \end{array}$$

Note that different sequences may now have variable lengths to either side of the reference point $|$.

Proposed kernel We can interpret each data point as consisting not of one sequence but rather a pair of two sequences, $(X^{(1)}, X^{(2)})$, where $X^{(1)}$ is the sequence to the left of the reference point and $X^{(2)}$ to the right. Thus, each data point is in S^2 rather than S . Starting with the exponential Hamming kernel (or any other kernel with discrete masses), we can extend it to S^2 using tensorization, and Proposition 15 guarantees it has discrete masses.

Example 8 (Hamming kernels with shifts) *Practical challenge* It is not always clear which positions exactly should be compared between two sequences. For instance, while protein sequences often have a clear starting position (the start codon), for other genetic elements (such as enhancers) the “beginning” of the sequence is less well-defined. One approach to this problem is to compare sequences under various offsets relative to one another, as in the Hamming kernel with shifts (Rätsch et al., 2005). The Hamming kernel with shifts, however, does not have discrete masses.

Proposed kernel We can define a shifted position-wise comparison kernel as

$$k_L(X, Y) = \sum_{l=0}^L k(X_{(l:)}, Y) + k(X, Y_{(l:)}),$$

where k is, for instance, the exponential Hamming kernel. This kernel compares X to Y under various offsets, i.e. starting from position $l = 0$, then position $l = 1$, etc., up to some maximum L ; it will be large if at least one of these offsets produces a good match. The kernel is a sum over individual kernels with discrete masses, so by Proposition 12 it has discrete masses.

Note each of these proposed kernels, in addition to having discrete masses, is computationally tractable, and in particular can be efficiently computed using the techniques developed for weighted degree kernels (Sonnenburg et al., 2007).

7. Alignment kernels

In this section, we design kernels with discrete masses that compare sequences based on pairwise alignments. Whereas position-wise comparison kernels judge two sequences X and Y to be similar only if there are a small number of substitutions that can transform X into Y , alignment kernels judge two sequences to be similar if there are a small number of substitutions, insertions and/or deletions that can transform X into Y . Biologically, sequences that are similar in this way often share similar phenotypes, and are closely related evolutionarily. However, this notion of similarity is poorly captured by position-wise comparison kernels. For example, the sequence $X = ATGC$ differs from the sequence $Y = TGC$ only by the insertion of a single A at the start of the sequence, but kernels based on the Hamming distance will judge the two sequences to be completely different, with distance $d_H(X, Y) = 4 = |X|$.

One popular, but problematic, technique for dealing with insertions and deletions is to pre-process the data set using a multiple sequence alignment algorithm, which adds gap symbols to each sequence to make it the same length. In essence, the idea behind multiple sequence alignment algorithms is to construct a point estimate of which letters in each sequence are evolutionarily related to one another via substitutions, and place each related letter in the same position, i.e. in the same column of the pre-processed data matrix. Once sequences have been aligned, it is more reasonable to apply a position-wise comparison kernel; insertions and deletions are compared by including the gap symbol in the alphabet \mathcal{B} . The problem with multiple sequence alignment pre-processing is that (a) it does not take into account uncertainty in which positions are related to one another, instead relying on a point estimate, and (b) it prevents downstream machine learning methods from generalizing to unseen sequences, since adding new sequences to the data set can change the multiple sequence alignment, for instance, if the new sequence is longer than those previously observed (Weinstein and Marks, 2021).

Alignment kernels offer an alternative approach to accounting for insertions and deletions (Haussler, 1999). Rather than transform the entire data set, they consider two sequences X and Y to be similar if they differ by a small number of insertions and deletions, as well as substitutions. Alignment kernels are of special relevance to problems involving sequences with high length variation, such as in the analysis of human antibodies or of distantly related evolutionary homologs.

Alignment kernels are typically computed with a dynamic programming procedure – comparing two sequences of length L costs $\mathcal{O}(L^2)$ (Watkins, 2000). However, the algorithm can be made much more efficient in practice with modest assumptions and the use of GPUs (Rush, 2020).

7.1 The alignment kernel

In this section we define the alignment kernel and show that it has discrete masses if and only if its hyperparameters are set to values in a certain range.

In biological sequence analysis, a pairwise alignment between two sequences is a matching between a subset of positions in each sequence, with the restrictions that (a) each position in each sequence can be matched to at most one position in the other sequence, and (b) matchings must be ordered, such that if site l_X in sequence X is matched to l_Y in

sequence Y , site $l'_X > l_X$ in X cannot match to a site $l'_Y < l_Y$ in Y (Durbin et al., 1998, chap. 2). For example, one alignment between the sequences $X = \text{ATGC}$ and $Y = \text{ACC}$ is,

$$\begin{array}{c} \text{ATGC} \\ | \quad | \quad | \\ \text{AC-C} \end{array}$$

where vertical lines $|$ denote a matching. Here, $-$ is a gap symbol, denoting the fact that the nucleotide G in X is unmatched; we can interpret the G as an insertion in X relative to Y , or, conversely, interpret the gap $-$ as a deletion in Y relative to X . The alignment kernel $k(X, Y)$ considers all possible pairwise alignments between X and Y , scores each alignment according to a position-wise comparison kernel applied to the aligned sequences, and sums up those scores to produce its value.

Mathematically, to define the alignment kernel, we start with two simpler kernels and then convolve them, following the construction of Haussler (1999). The first kernel, k_s , evaluates matches between aligned letters. The second kernel, k_I , evaluates insertions and deletions; it applies an “affine” gap penalty, which separately penalizes the presence of a gap and the gap’s total length. For any two kernels k_1 and k_2 on S , the convolution kernel $k_1 \star k_2$ is,

$$(k_1 \star k_2)(X, Y) = \sum_{X^{(1)} + X^{(2)} = X} \sum_{Y^{(1)} + Y^{(2)} = Y} k_1(X^{(1)}, Y^{(1)}) k_2(X^{(2)}, Y^{(2)})$$

where the sum is over all sequences $X^{(1)}, X^{(2)}, Y^{(1)}, Y^{(2)} \in S$ such that $X^{(1)} + X^{(2)} = X$ and $Y^{(1)} + Y^{(2)} = Y$. Recall that S includes the sequence of length zero, so the sum includes such terms as $X^{(1)} = X, X^{(2)} = \emptyset$.

Definition 20 (Alignment kernel) *Let k'_s be a strictly positive definite kernel on \mathcal{B} . Define k_s by extending k'_s to S with $k_s(X, Y) = 0$ if $|X| \neq 1$ or $|Y| \neq 1$.*

Let $\Delta\mu > 0$ be the penalty for starting an insertion and let $\mu > 0$ be the penalty for the insertion’s length. The insertion penalty function g is then defined as $g(X) = 1$ if $|X| = 0$ and $g(X) = \exp(-\Delta\mu - |X|\mu)$ otherwise. Define the non-negative-definite kernel $k_I(X, Y) = g(X)g(Y)$ for $X, Y \in S$.

The alignment kernel k is,

$$k = \sum_{l=0}^{\infty} k_I \star (k_s \star k_I)^{\star l},$$

where the exponent $\star l$ denotes the convolution of l kernels $k_s \star k_I$.

To see that this kernel indeed considers all possible pairwise alignments of $X, Y \in S$, note that we can rewrite $k(X, Y)$ as,

$$\sum_{l=0}^{\infty} \sum_{X^{(1)} + \dots + X^{(2l+1)} = X} \sum_{Y^{(1)} + \dots + Y^{(2l+1)} = Y} k_I(X^{(1)}, Y^{(1)}) \prod_{i=1}^l k_s(X^{(2i)}, Y^{(2i)}) k_I(X^{(2i+1)}, Y^{(2i+1)}). \quad (6)$$

Each term in the sum represents an alignment. Each alignment has l matches, in which the even subsequences $X^{(2)}, X^{(4)}, \dots, X^{(2l)}$ of X are matched to the even subsequences $Y^{(2)}, Y^{(4)}, \dots, Y^{(2l)}$ of Y . For the term to be non-zero, each of these subsequences must consist of a single letter, since otherwise $k_s(X^{(2i)}, Y^{(2i)}) = 0$. Between each of the match positions in X and Y there may be insertions, corresponding to the odd subsequences $X^{(1)}, X^{(3)}, \dots, X^{(2l+1)}$ and $Y^{(1)}, Y^{(3)}, \dots, Y^{(2l+1)}$. These subsequences can be of any length, including zero. The example alignment between $X = \text{ATGC}$ and $Y = \text{ACC}$ we saw earlier in this section corresponds to $X^{(1)} = \emptyset, X^{(2)} = \text{A}, X^{(3)} = \emptyset, X^{(4)} = \text{T}, X^{(5)} = \text{G}, X^{(6)} = \text{C}, X^{(7)} = \emptyset, Y^{(1)} = \emptyset, Y^{(2)} = \text{A}, Y^{(3)} = \emptyset, Y^{(4)} = \text{C}, Y^{(5)} = \emptyset, Y^{(6)} = \text{C}, Y^{(7)} = \emptyset$.

The kernel sums over all possible alignments: the outermost sum in Equation 6 is over the number of matches, $l \in \{0, 1, 2, \dots\}$, and the inner sums are over all possible choices of match positions in X and in Y . Each alignment is then scored according to k_s and k_I .

How flexible is the alignment kernel? Haussler (1999) showed it is strictly positive definite. Jorgensen and Tian (2015), motivated by problems outside biological sequence analysis, showed that the “binomial RKHS” does not have discrete masses; this corresponds to the alignment kernel with $\Delta\mu = \mu = 0$, $|\mathcal{B}| = 1$, and $k_s(b, b) = 1$. We now show that the alignment kernel can have discrete masses if and only if $\mu, \Delta\mu$ and k_s satisfy certain constraints.

Theorem 21 (Alignment kernels can have discrete masses) *Define K as the matrix with $K_{b,b'} = k_s(b, b')$ for $b, b' \in \mathcal{B}$, and $\sigma = \mathbf{1}_{\mathcal{B}}^T K^{-1} \mathbf{1}_{\mathcal{B}}$ where $\mathbf{1}_{\mathcal{B}} \in \mathbb{R}^{\mathcal{B}}$ is the vector with 1 in each entry. If $\Delta\mu > 0$, then k has discrete masses if and only if $2\mu \geq \log \sigma$. If $\Delta\mu = 0$, then k has discrete masses if and only if $2\mu > \log \sigma$.*

The proof is in Appendix C. It uses a similar strategy to that employed in proving that the position-wise comparison kernel has discrete masses (Theorem 19): reparameterize the alphabet, then break up sequence space into orthogonal hyperplanes. Over the orthogonal hyperplanes, the kernel takes a simpler form, which we prove has discrete masses using the alignment kernel’s feature representation; those features will be derived in Section 8.1 below.

Practically, Theorem 21 shows that if we set the hyperparameters of the alignment kernel naively, it may become unreliable, while if we set them appropriately, it has guaranteed flexibility. In particular, if we want flexibility, our choice of gap penalties (μ and $\Delta\mu$) is constrained by our choice of substitution penalties (k_s) through σ . If there is no penalty for starting a gap ($\Delta\mu = 0$) we require the penalty for extending a gap (μ) to be larger than $\frac{1}{2} \log(\sigma)$ in order for the kernel to have discrete masses. If there is a penalty for starting a gap ($\Delta\mu > 0$), the penalty for extending a gap can be slightly smaller, in that $\mu = \frac{1}{2} \log(\sigma)$ also gives discrete masses. In Appendix D we make sense of these conditions as arising from the requirement that the position-wise comparison kernel applied to individual alignments has discrete masses. In Appendix F we investigate relaxations of these conditions, such that the alignment kernel no longer has discrete masses but is still universal; we find that such relaxations are only valid given additional, problematic modifications of the kernel.

7.2 Heavy-tailed alignment kernels

Practically, when applied to data sets with highly diverse sequences, regression methods that use the alignment kernel often exhibit poor generalization due to diagonal dominance (Hausler, 1999; Saigo et al., 2004; Weston et al., 2003). Previous efforts to address this problem have encountered substantial difficulties; practitioners have gone so far as to introduce alternative “kernels” that are not positive semi-definite (Saigo et al., 2004). In this section, we diagnose possible sources of diagonal dominance in the alignment kernel, and propose heavy-tailed modifications of the alignment kernel that still possess discrete masses.

The key observation is that the alignment kernel, in effect, applies a position-wise comparison kernel to each pairwise alignment between sequences; diagonal dominance stems from the fact that the tail of this position-wise comparison kernel is thin. More precisely, consider an alignment kernel (Definition 20) with a letter kernel $k'_s(b, b') = \exp(-\lambda \mathbb{1}(b \neq b'))$; recall this choice gave us the exponential Hamming kernel in Section 6. Each term in the sum defining the alignment kernel (Equation 6) corresponds to an individual alignment, and contains a factor corresponding to the exponential Hamming kernel applied to the matched positions, namely $\prod_{i=1}^l k_s(X^{(2i)}, Y^{(2i)})$. As the Hamming distance between the matched positions increases, the term decays very quickly (exponentially), giving rise to diagonal dominance. To fix this problem, we follow the same strategy as in Example 5, and integrate over the bandwidth parameter λ to derive a thick-tailed alternative.

Example 9 (Heavy tailed alignment kernel on matches) *The “thick tailed alignment kernel on matches” is the kernel $\tilde{k}(X, Y) = \Gamma(\beta)^{-1} \int_0^\infty d\lambda (\lambda^\beta e^{-C\lambda}) k_\lambda(X, Y)$, where k_λ is the alignment kernel using k'_s with parameter λ , and $C, \beta > 0$. From Equation 5, we find that $\tilde{k}(X, Y)$ is,*

$$\sum_{l=0}^{\infty} \sum_{X^{(1)}+\dots+X^{(2l+1)}=X} \sum_{Y^{(1)}+\dots+Y^{(2l+1)}=Y} (C + d_H(X^{(2)} + \dots + X^{(2l)}, Y^{(2)} + \dots + Y^{(2l)}))^{-\beta} \\ \times k_I(X^{(1)}, Y^{(1)}) \prod_{i=1}^l k_I(X^{(2i+1)}, Y^{(2i+1)})$$

In this kernel, the dependence on the Hamming distance between matched positions follows a power law, rather than an exponential decay. If we define $K_{\lambda, b, b'} = k_\lambda(b, b')$ then $\sigma_\lambda = \mathbf{1}_B^T K_\lambda^{-1} \mathbf{1}_B = \left(\frac{1}{|B|} + \frac{|B|-1}{|B|} e^{-\lambda} \right)^{-1}$, which approaches 1 as $\lambda \rightarrow 0$. Thus, as long as $\mu > 0$, k_λ will have discrete masses for small λ , by Theorem 21. So, by Proposition 12, \tilde{k} has discrete masses. Note that this modified kernel can, like the standard alignment kernel, be computed efficiently using a dynamic programming algorithm (Appendix I) in $\mathcal{O}(L^3)$ time if $L = |X| = |Y|$. However the algorithm can be potentially be made more efficient with minor modifications and the use of GPUs (Rush, 2020).

Besides the substitution penalty λ , we can also consider the role of the insertion penalties μ and $\Delta\mu$. We find a similar issue: as the length of insertions increases, each term of the alignment kernel decays exponentially, since $k_I(X, Y) = \exp(-\mathbb{1}(|X| > 0)(\Delta\mu + \mu|X|) - \mathbb{1}(|Y| > 0)(\Delta\mu + \mu|Y|))$. Now, the parameter μ plays the role of the bandwidth. To produce thick-tails, we can perform the analogous transform, and take the integral

$\Gamma(\beta)^{-1} \int_0^\infty d\mu (\mu^\beta e^{-C\mu}) k_\mu(X, Y)$, where k_μ is the alignment kernel with parameter μ . This gives us a heavy-tailed alignment kernel on gaps, which has discrete masses by Proposition 12. We could even apply the same transform to the thick tailed alignment kernel on matches, yielding a thick tailed alignment kernel on both gaps and matches.

7.3 Local alignments

Genomes are very long, and so in practice machine learning is typically done only on specific genetic elements, such as genes, promoters, enhancers, etc. It is sometimes ambiguous where exactly these genetic elements start or end. This motivates the idea of a local alignment, which ignores insertions and deletions at the start and end of a sequence. Using a local alignment as a measure of similarity means two sequences that contain the same genetic element but different flanking regions will still be considered similar (Durbin et al., 1998, Chap. 2). In this section, we study a local alignment kernel, and establish conditions under which it has discrete masses (Vert et al., 2004; Saigo et al., 2004).

The local alignment kernel is a modification of the alignment kernel, which does not penalize the creation of gaps at the start or end pairwise alignments.

Definition 22 (Local alignment kernel) *Let k_I and k_s be as in Definition 20. Also define the modified insertion gap penalty kernel $\tilde{k}_I(X, Y) = \exp(-\mu(|X| + |Y|))$, which lacks the gap start penalty $\Delta\mu$. The local alignment kernel is,*

$$k_{\text{la}} = \tilde{k}_I + \sum_{l=1}^{\infty} \tilde{k}_I \star (k_s \star k_I)^{\star(l-1)} \star k_s \star \tilde{k}_I.$$

We can extend the logic of the proof of Theorem 21 to establish conditions under which local alignment kernels have discrete masses; these conditions turn out to be identical to those for the regular alignment kernel.

Theorem 23 *Define K as the matrix with $K_{b,b'} = k_s(b, b')$ for $b, b' \in \mathcal{B}$, and $\sigma = \mathbf{1}_{\mathcal{B}}^T K^{-1} \mathbf{1}_{\mathcal{B}}$ where $\mathbf{1}_{\mathcal{B}} \in \mathbb{R}^{\mathcal{B}}$ is the vector with 1 in each entry. If $\Delta\mu = 0$, k_{la} has discrete masses if and only if $2\mu > \log \sigma$. If $\infty > \Delta\mu > 0$, k_{la} has discrete masses if and only if $2\mu \geq \log \sigma$. If $\Delta\mu = \infty$, k_{la} has discrete masses regardless of the values of μ, σ .*

A proof is given in Appendix H. Note that for the standard, non-local alignment kernel, if we set the gap start penalty to infinity ($\Delta\mu = \infty$), we recover a position-wise comparison kernel. For local alignments, however, the situation is more complex, as there are different options for the number of gaps to put at the start and end of each sequence. We find that when $\Delta\mu = \infty$, the local alignment kernel has discrete masses regardless of the value of its other hyperparameters.

8. Kmer spectrum kernels

In this section we study kmer spectrum kernels. Kmer spectrum kernels, like alignment kernels, are motivated by the biological observation that sequences which differ by insertions, deletions, or other complex mutations often share similar phenotypes and are related

evolutionarily. Remarkably, we will find that the relationship between kmer spectrum kernels and alignment kernels runs deeper than their motivation: the two are in certain cases equivalent, up to an appropriate tilting.

Kmer spectrum kernels featurize sequences according to the presence and absence of subsequences (kmers). Since the position of the kmer within the sequence does not affect the value of the corresponding feature, kmer spectrum kernels can be relatively insensitive to insertions or deletions, as compared with position-wise comparison kernels. In Example 1, we saw a commonly used kmer spectrum kernel which had only a finite number of features, and so was neither universal nor characteristic. Here we will study infinite-dimensional kmer spectrum kernels, and show that they have discrete masses.

When k is not very small or gaps are included, the most popular algorithms to evaluate these kernels on two sequences of length L use dynamic programming and run in $\mathcal{O}(kL^2)$ (Lodhi et al., 2002).

8.1 Gapped kmer spectrum kernels

We start by connecting alignment kernels to kmer spectrum kernels. The basic connection between the two kernels has led to similar dynamic programming methods to compute each (Lodhi et al., 2002) and has been noted in a special case to motivate neural network architectures (Chen et al., 2019). Here we formally describe the connection in terms of an orthonormal basis of the RKHS. We show that, for a particular tilting, the features of the alignment kernel are a weighted sum of kmer counts, allowing for some kmer mismatches. More precisely, we show that the tilted alignment kernel is equivalent to an infinite-dimensional gapped kmer spectrum kernel. Since tilting preserves discrete masses (Proposition 16), this gapped kmer spectrum kernel has discrete masses.

Intuitively, a “gapped kmer” is a kmer that includes gaps; it matches another sequence if each of the non-gap characters match, regardless of the letters at the gap positions. For example, the gapped kmer sequence AT-G matches ATCG, ATGG, ATAG and ATTG. Formally, a gapped kmer of length L , with M non-gap letters, is defined by an ordered set of indices $J = (j_0, \dots, j_{M+1})$, with $-1 = j_{-1} < j_0 < j_2 < \dots < j_M < j_{M+1} = L$, and a sequence of M non-gap letters, $Z \in \mathcal{B}^M$. Each j_m for $m \in \{0, \dots, M-1\}$ indexes the location of the non-gap letter $Z_{(m)}$. There is a gap after the m th letter if $j_m + 1 \neq j_{m+1}$. Let $G(L, M, g)$ be the set of all gapped kmer indices J , with length L , M non-gap letters, and g gaps. Let $G(L, g) = \cup_M G(L, M, g)$ be the union over all possible M . For any sequence $X \in S$, and $J = (j_0, \dots, j_{M+1}) \in G(|X|, g)$, define $X_{(J)} = X_{(j_0)} + \dots + X_{(j_{M+1})}$.

The following theorem equates alignment kernels to gapped kmer spectrum kernels. In particular, it shows that we can represent alignment kernels in terms of features $u_V(X)$ that depend on the number of times a sequence $V \in S$ appears in X , allowing for gaps. In other words, each feature $u_V(X)$ depends on a sum over the counts of all gapped kmers with non-gap letters V .

Theorem 24 (The features of alignment kernels are gapped kmer counts) *Consider an alignment kernel k with $k_s(b, b') = \sigma^{-1}|\mathcal{B}|\delta_b(b')$ for $b, b' \in \mathcal{B}$. Tilting the kernel by $A(X) = e^{\mu|X|}$ gives $k^A(X, Y) = A(X)A(Y)k(X, Y)$. For all sequences $X \in S$, define the*

features $\{u_V(X)\}_{V \in S}$, given by

$$u_V(X) = e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} \mathbb{1}(X_{(J)} = V), \quad (7)$$

where $\zeta = 2\mu - \log \sigma + \log |\mathcal{B}|$. Now, $\{u_V(X)\}_{V \in S}$ is an orthonormal basis for \mathcal{H}_{k^A} . We call the kernel $k^A(X, Y)$ the “gapped kmer spectrum kernel”.

The proof is in Appendix E, including the case of general k_s . This theorem shows that the features of the tilted alignment kernel depend on a weighted sum over gapped kmer counts. The weights $e^{-\Delta\mu g}$ decrease as the number of gaps increases, thus penalizing gapped kmers with large numbers of gaps. Since tilting preserves discrete masses, we can conclude that the gapped kmer spectrum kernel has discrete masses if and only if the conditions of Theorem 21 are satisfied.

The tilting in Theorem 24 down-weights longer sequences, but does not change how the alignment kernel judges sequence similarity. Indeed, the theorem is straightforward to extend to other tiltings, such as the popular normalizing tilting $A(X) = \sqrt{k(X, X)}^{-1}$. In this case we find $k^A(X, Y) = \frac{u(X)^T u(Y)}{\|u(X)\| \|u(Y)\|}$, where $u(X)$ is the infinite dimensional vector indexed by S with $u_V(X)$ at position $V \in S$. The theorem can also be extended to non-diagonal k_s , in which case the features are the gapped kmer counts with a reparameterized alphabet (Appendix E).

Theorem 24 provides a practical example of a kmer spectrum kernel with discrete masses. It is also useful as a theoretical tool. Theorem 39 (Appendix G) uses Theorem 24 to establish that a simple version of the alignment kernel has discrete masses, which in turn allows us to prove that the full alignment kernel has discrete masses (Theorem 21). The proof technique used in Theorem 39 relies on a careful analysis of the combinatorics of biological sequence alignments, using the technique of generating functions and the theory of the Riordan group (Shapiro et al., 1991).

8.2 Heavy-tailed gapped kmer spectrum kernels

We now re-examine the diagonal dominance problem in alignment kernels, in light of Theorem 24. Equation 7 says that the feature weights, namely $\exp(\frac{1}{2}\zeta|V|)$, increase exponentially with kmer length $|V|$. This exponential scaling means the the kernel $k(X, Y)$ takes a very large value when applied to very similar sequences X and Y , as compared to its value when X and Y are very different and only have short kmers in common. This can result in diagonal dominance. Trying to avoid the problem by making ζ small leads to loss of flexibility, as the kernel no longer has discrete masses when $\zeta < \log |\mathcal{B}|$. In other words, ζ acts as a tuning parameter that trades off kernel flexibility and diagonal dominance. To address the problem, we propose a gapped kmer kernel with discrete masses in which the weighting on features scales as a power law with respect to kmer length, rather than exponentially.

Example 10 (Heavy tailed gapped kmer spectrum kernel) *Call the unscaled features $\tilde{u}_V(X) = e^{-|V|\zeta/2} u_V(X)$ and notice they do not depend on ζ . Consider a gapped kmer spectrum kernel tilted by $A(X) = e^{-\frac{1}{2}\zeta|X|}$,*

$$k_\zeta(X, Y) = e^{-\frac{1}{2}\zeta|X|} e^{-\frac{1}{2}\zeta|Y|} \sum_{V \in S} u_V(X) u_V(Y) = \sum_{V \in S} e^{-\frac{1}{2}(\zeta|X| + \zeta|Y|)} \tilde{u}_V(X) \tilde{u}_V(Y).$$

Then for $C, \beta > 0$, the “heavy tailed gapped kmer spectrum kernel” is defined as,

$$\begin{aligned}\tilde{k}(X, Y) &= \Gamma(\beta)^{-1} \int_0^\infty k_\zeta(X, Y) \left(\zeta^\beta e^{-C\zeta} \right) d\zeta \\ &= \sum_{V \in S} \left(C + \frac{1}{2}(|X| + |Y|) - |V| \right)^{-\beta} \tilde{u}_V(X) \tilde{u}_V(Y).\end{aligned}$$

In this kernel, shorter features V are down-weighted following a power law, rather than exponential decay. Proposition 12, Theorem 21 and Theorem 24 together ensure the kernel has discrete masses. The kernel can be computed efficiently using the dynamic programming algorithm given in Appendix I in $\mathcal{O}(L^3)$ time if $L = |X| = |Y|$.

Note that since ζ depends on μ , this kernel is closely related to the heavy tailed alignment kernel on gaps discussed in Section 7.2, which integrates over μ with the same base measure. The exponential decay in pairwise alignment scores with increasing gap length, and the exponential decay in feature weight for shorter kmers, are two ways of looking at the same diagonal dominance problem in alignment kernels. An analogous construction can also be used to create a heavy-tailed local alignment kernels.

8.3 Ungapped kmer spectrum kernels

We now consider ungapped kmer spectrum kernels, whose features correspond to the number of times a kmer appears exactly in a sequence. In particular, we show that a local alignment kernel with an infinite gap start penalty, $\Delta\mu = \infty$, is an ungapped kmer spectrum kernel, and that this kmer spectrum kernel therefore has discrete masses.

Proposition 25 (Infinite kmer spectrum kernel) *Consider a tilted local alignment kernel $k = k_{\text{la}}^A$, with $\Delta\mu = \infty$, $\zeta = 0$, and $A(X) = \exp(\mu|X|)$. This kernel has discrete masses and can be written as*

$$k(X, Y) = \sum_{V \in S} u_{\text{la}, V}(X) u_{\text{la}, V}(Y),$$

where the feature $u_{\text{la}, V}(X) = \sum_{l=0}^{|X|-|V|} \mathbb{1}(X_{(l:l+|V|)}) = V$ is the number of times the kmer V occurs in X .

Proof Define $G_{\text{la}}(L, M, g)$ as those elements in $G(L) = \cup_g G(L, g)$ that are of length M with g gaps, *not counting gaps at the beginning or end*. Using the same logic as Theorem 24, the features of the kernel, after tilting, are, for $V, X \in S$

$$u_{\text{la}, V}(X) = \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G_{\text{la}}(|X|, g)} \mathbb{1}(X_J = V).$$

Now if we set $\Delta\mu = \infty$, these features are simply the number of occurrences, without gaps, of V in X . By Theorem 23 and Proposition 16, the kernel has discrete masses. \blacksquare

The standard kmer spectrum kernel, by comparison, is $k(X, Y) = \sum_{|V| \leq L} u_{\text{la}, V}(X) u_{\text{la}, V}(Y)$ (Example 1). That is, it uses a finite rather than an infinite sum over kmers, and so does not possess any flexibility guarantees. The infinite kmer spectrum kernel has strong flexibility

guarantees, without sacrificing much computational efficiency: we can compute it using the same dynamic programming methods used for local alignment kernels, since Proposition 25 shows it *is* a local alignment kernel (Saigo et al., 2004). Note also that since the infinite spectrum kernel is an alignment kernel, it can be evaluated for two sequences of length L with compute $\mathcal{O}(L^2)$, faster than the standard kmer spectrum kernel.

9. Embedding kernels

In this section we study embedding kernels, which use sequence representations in Euclidean space to judge sequence similarity. A key motivation for such kernels is to avoid the need for hand-crafted similarity measures, and to instead learn a similarity measure from data. The data used to learn a representation need not be the same data the kernel is applied to; for example, a common transfer learning technique is to use a large, unlabeled sequence data set to learn a sequence representation, and then, on a small labeled data set, learn a regression from representations to labels (Yang et al., 2018; Alley et al., 2019; Biswas et al., 2021; Rao et al., 2019; Detlefsen et al., 2022). Popular modern unsupervised representation learning methods include variational autoencoders, recurrent neural networks, and transformers. Such techniques have often been found to produce low-dimensional representations that reflect biological properties well, in the sense that sequences with similar representations have similar phenotypes.

An embedding kernel consists of a continuous kernel applied to sequence representations.

Definition 26 (Embedding kernel) *Let $F : S \rightarrow \mathbb{R}^D$ be an embedding function, which maps sequences to representations. The embedding kernel is defined as $k(X, Y) = k_E(F(X), F(Y))$, where k_E is a kernel on $\mathbb{R}^D \times \mathbb{R}^D$.*

For example, if we use a variational autoencoder to learn a representation, and a Gaussian process to predict labels, F would be given by the encoder and k_E would be the kernel in the Gaussian process. Deep kernel learning also fits the form of an embedding kernel, with F a neural network (Wilson et al., 2016). If we regress directly from sequence to labels using a neural network, we can think of the map from sequence space to the final hidden layer of the network as the embedding function F , while the map from the hidden layer to the output is approximately a kernel regression, by the argument that single layer neural networks converge to Gaussian processes in the infinite width limit (Neal, 1996; Williams, 1996).

9.1 Universality and characteristicness

We first describe conditions under which embedding kernels are universal and characteristic. Intuitively, if we choose a kernel k_E in Euclidean space that is universal and characteristic, then the embedding kernel over sequence space will be universal and characteristic as well, so long as the embedding function F maps each sequence in S to a unique point in \mathbb{R}^D .

Proposition 27 (Universal and characteristic embedding kernels) *Assume $k_E(z, z') = \Psi(z - z')$ is a translation invariant kernel and that Ψ is a positive continuous function on \mathbb{R}^D*

that has a strictly positive Fourier transform.⁴ Then the embedding kernel k is characteristic and C_0 -universal if and only if F is injective.

The proof is in Appendix J. Note that the conditions on k_E ensure that it is universal and characteristic over \mathbb{R}^D (Sriperumbudur et al., 2010, 2011).

The condition that F is injective conflicts with some representation learning approaches, particularly those aimed at learning sparse, interpretable features. For the sake of interpretability, it is often desirable to learn a representation that ignores certain features of the data points entirely, such as those features which vary little in the training data. However, when this representation is applied to make predictions about held out data, those sequences that do vary in the ignored features can be mapped to the same point in the embedding space. Since the embedding F is therefore not injective, downstream supervised predictors will be limited in their flexibility. An example is in Appendix K.

By contrast, a completely “uninterpretable” representation, which makes no attempt to organize embedding space by a biological notion of sequence similarity, can easily yield universal and characteristic embedding kernels. In particular, if we use an embedding function that gives sequences random representations, distributed according to a continuous distribution on \mathbb{R}^D , then the chance of any two sequences having the same embedding is zero.

Proposition 28 (Random embeddings give universal and characteristic kernels)

Assume $\{F(X)\}_{X \in S}$ are i.i.d. random variables distributed according to a measure which is absolutely continuous with the Lebesgue measure. Then F is almost surely injective, and so an embedding kernel based on F is almost surely universal and characteristic.

Proof Number the elements of S , such that $\{X_1, X_2, \dots\} = S$. For any $n \in \mathbb{N}$, we have that $F(\{X_1, \dots, X_n\})$ is a set of measure zero, so $p(F(X_{n+1}) = F(X_i) \text{ for some } i \leq n) = 0$. Thus, $p(F(X_i) = F(X_j) \text{ for some } i, j) = 0$. ■

One implication of this proposition is that the “meaningless” randomness introduced through stochastic training of large neural networks may help them produce representations that are useful for downstream tasks.

9.2 Discrete masses

We have so far described conditions under which embedding kernels are characteristic and universal; we found that random embeddings suffice. However, we saw in Example 3 an embedding kernel that is universal and characteristic but cannot reliably be used for optimization, as it does not metrize the space of distributions. In this section, we address the question of when embedding kernels have discrete masses, and thus metrize $\mathcal{P}(S)$. We find that many standard representation learning techniques are likely to produce embedding kernels that lack discrete masses even when they are universal and characteristic, and propose a fix.

4. Ψ has strictly positive Fourier transform if $\hat{\Psi}(\xi) > 0$ for all ξ where $\hat{\Psi}$ is the Fourier transform of Ψ . For example, $\Psi(x) = \exp(-x^2)$ has a strictly positive Fourier transform.

For an embedding kernel to have discrete masses, the embedding function F must not only map different sequences to different points in \mathbb{R}^D —those points must also be sufficiently spread out.

Proposition 29 (Embedding kernels with discrete masses) *Consider an embedding kernel with an injective F that meets the conditions of Proposition 27. k_E metrizes $\mathcal{P}(S)$ if and only if k_E has discrete masses, which occurs if and only if $F(S)$ has no accumulation points, that is, there is no $X \in S$ such that $F(X)$ is in the closure of $F(S \setminus \{X\})$.*

The proof is given in Appendix J. In Example 2, we saw an embedding kernel that was universal and characteristic, but which had an accumulation point at the sequence A , and thus did not metrize $\mathcal{P}(S)$.

Unlike for universality and characteristicness, to construct an embedding kernel with discrete masses, it is not sufficient to use sequence representations that are drawn i.i.d. from a distribution on \mathbb{R}^D .

Proposition 30 (I.i.d. embeddings do not have discrete masses) *Assume $\{F(X)\}_{X \in S}$ are i.i.d. random variables. Then the embedding function has accumulation points almost surely, and so does not metrize $\mathcal{P}(S)$.*

Proof Consider any $X_0 \in S$, and note that almost surely, $p(\|F(X) - F(X_0)\| < \epsilon) > 0$ for any X, ϵ . Thus, $p(\|F(X) - F(X_0)\| < \epsilon \text{ for some } X) = 1$ for all ϵ . So, with probability 1 there are sequences X_{n_1}, X_{n_2}, \dots which are not X_0 but where $F(X_{n_k}) \rightarrow F(X_0)$ as $k \rightarrow \infty$. ■

Intuitively, to construct an embedding without accumulation points, one must somehow fit sequence space S into \mathbb{R}^D without letting the representations of each sequence get too close together. One approach is to make embeddings of longer and longer sequences more and more spread out, rather than sampling all the embeddings from the same distribution.

Proposition 31 (Scaled random embeddings have discrete masses) *Consider an initial embedding \tilde{F} where each $\tilde{F}(X)$ for $X \in S$ is drawn from the uniform distribution on the sphere, $\{x \in \mathbb{R}^D \mid \|x\| \leq 1\}$. Then, a kernel using the scaled embedding $F(X) = |\mathcal{B}|^{(1+\epsilon)|X|/D} \tilde{F}(X)$, for $\epsilon > 0$, metrizes $\mathcal{P}(S)$ almost surely.*

The proof can be found in Appendix L. Intuitively, the idea is that this rescaling gives each sequence representation “enough space”. If $V(B(1))$ is the volume of the unit ball in \mathbb{R}^D , we can think of F as embedding the set \mathcal{B}^L into a ball of radius $|\mathcal{B}|^{(1+\epsilon)L/D}$. Thus, the average amount of volume per sequence is $V(B(1))(|\mathcal{B}|^{(1+\epsilon)L/D})^D / |\mathcal{B}|^L = V(B(1))|\mathcal{B}|^{L\epsilon}$, which is growing with L . This is just enough space to ensure there is no accumulation.

Practically, Proposition 30 suggests that many representation learning approaches are unlikely to yield kernels that metrize $\mathcal{P}(S)$. Typically, as we embed more and more sequences, their representations stay clustered in a localized region of embedding space near the origin, rather than diverging away. Indeed, in many applications of representation learning, this behavior is desirable, as it makes the representations easier to visualize and interpret. One way to understand this behavior is as a consequence of the priors and regularizers typically used in representation learning. For instance, in latent variable models, we expect the set of embeddings $\{F(X)\}_{X \in S}$, taken as a whole, to look roughly like a set

of samples from the prior on the latent variable; since the prior is typically chosen to be a continuous distribution on \mathbb{R}^D , such as a normal distribution, Proposition 30 suggests the embedding kernel will not have discrete masses. The fact that we only observe a finite number of data points in the training data can also hurt: even if the representations of the training data are well spread out, the representations of sequences far from the training data are likely to look roughly like samples from the prior. In short, typical representation learning procedures are unlikely to give sequences enough space to produce embedding kernels with discrete masses.

Proposition 31 suggests a simple fix: rescale the embedding. Given an embedding \tilde{F} that does not become more spread out with increasing sequence length, modify it to $F(X) = |\mathcal{B}|^{(1+\epsilon)|X|/D} \tilde{F}(X)$, for $\epsilon > 0$. The resulting kernel will, by the reasoning of Proposition 31, likely have discrete masses. In Section 10.3 we illustrate the effectiveness of this trick in practice. Another approach would be to use a length-dependent prior, or length-dependent regularization, when learning the embedding in the first place.

10. Empirical Results

In this section we examine the performance, on real biological sequence datasets, of some of our proposed kernels with discrete masses. We compare each to an existing kernel that relies on a similar notion of sequence similarity but lacks discrete masses. We find that our theoretical guarantees on kernel flexibility lead to systematic improvements in performance on finite data. We consider eight kernels across each of the four families we studied, and three machine learning tasks. These include tasks where kernel methods are already standard in biology (regression, distribution comparison) and tasks where our guarantees enable novel methods that are new to biological applications (choosing representative sequences). Code can be found at <https://github.com/AlanNawzadAmin/Kernels-with-guarantees/>.

10.1 Predicting transcription factor binding

We first consider a regression problem, where the aim is to predict whether a transcription factor binds a DNA sequence. Understanding transcription factor binding is critical for understanding and controlling gene expression. The data consists of pairs of DNA sequences and transcription factor binding strengths, measured in terms of the intensity of a fluorescent signal in a micro-array assay (Barrera et al., 2016). We examined three separate data sets from three distinct transcription factors, the homeobox proteins Hox-D13, Nkx25, and Esx1. All the DNA sequences are length 8. We fit the data using kernel regression, and evaluate success using the root mean squared error on the training data. Our aim is to compare kernel flexibility; more flexible kernels should provide better fits to the data.

We first consider alignment kernels, and, applying Theorem 21, compare a kernel that has discrete masses ($2\mu < \log \sigma$) to one that does not ($2\mu > \log \sigma$). We find that the version with discrete masses has better performance in the large data regime across all three example datasets (Fig. 2). The version without discrete masses eventually starts to plateau with increasing data, while the version with discrete masses reliably approaches the correct answer in Fig. 2b and 2c.

We next consider kmer spectrum kernels, and again compare a kernel that has discrete masses (the infinite kmer spectrum kernel, Proposition 25) to one that does not (the finite

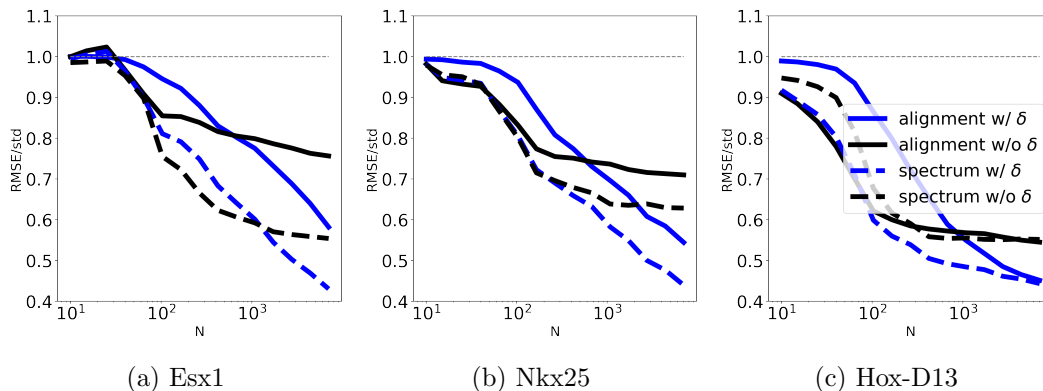


Figure 2: **Kernel regression applied to transcription factor binding** Kernel regression performance, comparing alignment and kmer spectrum kernels with and without discrete masses (“w/ δ ” versus “w/o δ ”). N is the size of the sub-sampled data set. Performance is measured by root mean squared error, normalized by the standard deviation of the outcome variable.

kmer spectrum kernel of Example 1, with $L = 3$). Here again we find that the version with discrete masses has better performance in the large data regime, while the version without can plateau at a substantial error level (Fig. 2). While in the case of the alignment kernels, the version with discrete masses showed moderately worse performance in the low data regime, here the version with discrete masses performed as well or better than the version without across the full range of data set sizes considered. In short, guaranteeing discrete masses leads to more reliable regression, especially but not exclusively in the large data regime.

10.2 Distinguishing immune repertoires

In this section we consider a testing problem, where the aim is to distinguish between patients’ T cell receptor (TCR) repertoires. T cells play a central role in the human adaptive immune system, recognizing foreign antigens through their TCR and then triggering an immune response. Here, we compare two different patients’ immune systems by comparing the distribution of their TCR sequences. For each patient, we have a data set of TCR CDR3 sequences, which vary in length from 10 to 19 amino acids (10x Genomics, 2022). We apply a two-sample test by performing a bootstrap for degenerate U-statistics on the MMD, with the null hypothesis that the patients’ TCRs are drawn from the same distribution (Arcones and Gine, 1992). We evaluate performance based on how often the test correctly rejects the null hypothesis at 90% confidence, averaging over 25 random sub-samples of a large data set. Our aim is to compare kernel flexibility; more flexible kernels should be able to distinguish between two patients’ repertoires more reliably.

We apply the same four kernels from Section 10.1. Although all kernels were able to distinguish the two distributions eventually with enough data, the kernels with discrete masses were able to distinguish reliably with much less data (Fig. 3). In other words, guaranteeing discrete masses improves the power of the two-sample test on real data sets.

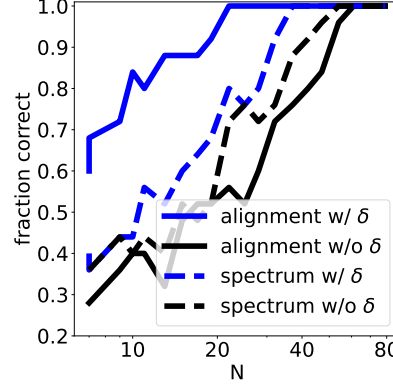


Figure 3: **MMD two sample test applied to patient TCR repertoires.** MMD two-sample test performance, comparing alignment and kmer spectrum kernels with and without discrete masses (“w/δ” versus “w/o δ”). N is the size of the sub-sampled data set. Performance is measured by the fraction of independent data sub-samples for which the null hypothesis is, correctly, rejected.

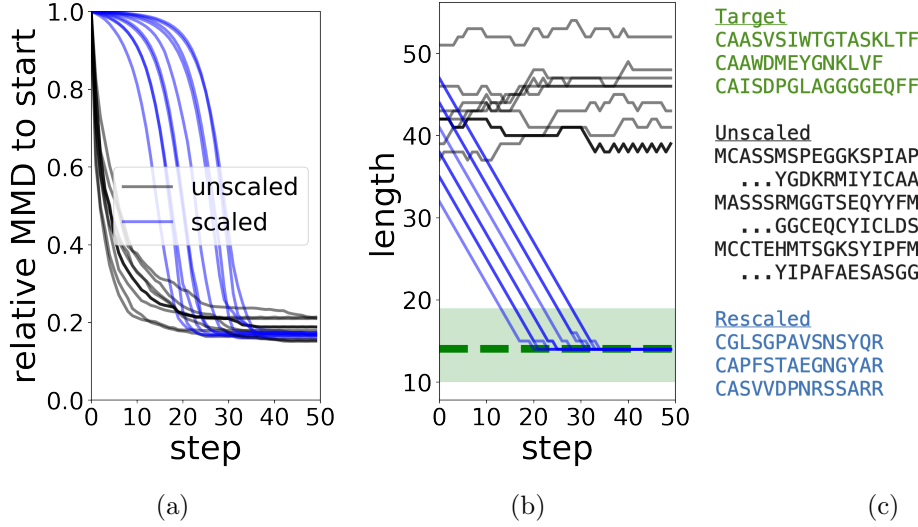


Figure 4: **Optimizing representative TCR sequences using embedding kernels.** Distribution optimization performance, comparing unscaled embedding kernels (likely to not have discrete masses) to scaled embedding kernels (likely to have discrete masses). (a) MMD per step during optimization. Here, the MMD is normalized to the MMD of the initial sequence X . Optimization was repeated with 10 different initial sequences. (b) Length of the optimized sequence during optimization. The maximum, minimum and average value of the target sequence lengths are shown in green. (c) Examples of sequences from the target set and of optimized sequences.

10.3 Optimizing representative receptors

In this section we consider a distribution optimization problem, where the aim is to find a TCR sequence that is representative of a larger set. Often biologists are interested in understanding the properties of many sequences—for instance, the TCRs from all the T cells that infiltrated a patient tumor—but have limited experimental resources, and so can only synthesize and test a small number in the laboratory. Here, we aim to find a single sequence X that is representative of a large set of TCR sequences, Y_1, \dots, Y_N . We do so by optimizing δ_X to be close to the empirical distribution $p_Y = \frac{1}{N} \sum_{n=1}^N \delta_{Y_n}$, with the optimization objective $\text{MMD}(\delta_X, p_Y)$. We use $N = 100$ human TCR CDR3 sequences, with lengths varying between 10 and 17, as the target set (10x Genomics, 2022). We initialize X at a sequence that is longer than those in the target set, but has similar amino acid composition and arrangement (in particular, a random CDR3 from the target set, concatenated to itself). We then optimize $\text{MMD}(\delta_X, p_Y)$ by taking the best substitution, insertion or deletion of a single amino acid at each step, for 100 steps. Our aim is to compare kernels’ ability to metrize $\mathcal{P}(S)$. Since the main difference between the initial X and the target set is its length, we are particularly interested in seeing if optimizing $\text{MMD}(\delta_X, p_Y)$ yields a sequence X that is similar to the target set in length.

We focus on embedding kernels. We use a pre-trained embedding function $\tilde{F} : S \rightarrow \mathbb{R}^{64}$, “UniRep”, learned by a deep recurrent neural network trained on a large data set of diverse protein sequences drawn from across evolution (Alley et al., 2019). We compare embedding kernels that use \tilde{F} directly (which are likely to be universal and characteristic by Proposition 27 but are unlikely to have discrete masses, by Proposition 30) to those that use the scaled embedding $F(X) = 20^{(1+\epsilon)|X|/64} \tilde{F}(X)$ with $\epsilon = 0.1$ (which are likely to have discrete masses, by Proposition 31). For both embedding functions, we construct an embedding kernel using the Euclidean inverse multiquadric kernel, $k_E(x, y) = (1 + \|x - y\|_2^2)^{-1}$, which satisfies the conditions of Proposition 27. For both embedding kernels, our optimization routine converges in MMD within the first 50 steps (Fig. 4A).

The unscaled kernel yields optimized sequences X that are much longer than those in the target set, while the scaled kernel, which is likely to have discrete masses, yields optimized sequences X at the target set’s average length (Fig. 4B). Visual inspection of the optimized X also suggests that the scaled kernel produces substantially more representative designs; for instance, the optimized sequences from the scaled kernel have a cysteine in the first position, like all the target sequences, but the optimized sequences from the unscaled kernel do not (Fig. 4C). In other words, rescaling the kernel to add discrete masses improves its ability to metrize $\mathcal{P}(S)$ in practice.

11. Discussion

In this article, we studied the flexibility of kernels for biological sequences. Our aim was to find kernels that are universal, characteristic, and metrize $\mathcal{P}(S)$, which guarantee they will be reliable when applied to machine learning problems involving regression, distribution comparison, and distribution optimization. We found that many existing kernels fail to meet one or all of these criteria, but that all three are met if the kernel has discrete masses. We then modified existing kernels to ensure they have discrete masses, imbuing them with strong theoretical guarantees and improved empirical performance.

In applications on Euclidean space, issues of kernel flexibility are now rarely discussed, and the reliability of kernel-based methods like Gaussian processes is often taken for granted. This is for a good reason: the default kernels implemented in packages and recommended in papers and tutorials are typically universal and characteristic at minimum (such as the Gaussian kernel) and often metrize weak convergence as well (such as the Matérn and inverse multiquadric kernels) (Sriperumbudur et al., 2010, 2011; Balandat et al., 2020). We have sought to provide a similarly stable foundation for kernel methods on biological sequence data.

We have considered a variety of different biological notions of sequence similarity, but in some settings, one may want others. Sequences can be joined at the end, as with cyclic peptides and circular RNA, or joined into large protein complexes, made up of many separate polypeptide chains. They can possess symmetries, like repeats, and they can be mutated in complex ways, such as in immune receptor development. One direction for future work is to construct new kernels with discrete masses that are suited to these or other biological phenomena. The tools we have developed for proving kernels have discrete masses (Section 5) provide a starting point.

We may also use these tools to better understand and improve deep learning methods (Alipanahi et al., 2015). On the one hand, kernels may be used as a lens for understanding the behavior of deep learning methods, by analyzing their neural tangent kernels (Jacot et al., 2018). On the other hand, kernels may inspire new neural network architectures with appropriate inductive biases (Lei et al., 2017; Chen et al., 2019).

Another direction is to develop methods to automatically learn complex kernels from data, while preserving the discrete mass property. Our proofs in Sections 6-9 demonstrate how, starting with a simple kernel with discrete masses, we can apply a series of discrete mass-preserving transformations (Section 5.2) to develop a more complicated kernel with discrete masses. A possible approach to learning new kernels is to optimize this series of transformations.

Broadly, our work contributes to the small but growing literature on the theoretical foundations of machine learning for biological sequences. Machine learning methods for biological sequences are of wide and increasing importance to human health and the environment, and are often applied to problems where trust and reliability are essential. However, they are sometimes mistakenly thought of as mere special cases of “general purpose” methods, invented for other types of data. Our results instead encourage further investigation into the specific theory of biological sequence analysis.

Appendix A. Illustrative example

We illustrate the practical use of our theoretical results via a toy example. We consider the problem of sequence regression, where the aim is to predict a sequence’s phenotype, e.g. fluorescent color, enzymatic activity, binding strength, etc. We show that kernel regression using a classic biological sequence kernel, a *Hamming kernel of lag L* (which is an instance of a *weighted degree kernel*) can fail entirely to fit the true sequence-to-phenotype map. We then introduce a small but non-obvious modification to the kernel, which our theory guarantees will allow kernel regression to succeed.

Hamming kernels compare sequences based on a sliding window, counting the number of times the subsequence (kmer) in the window matches exactly (Schweikert et al., 2008). More precisely, we consider a Hamming kernel that uses features $\Phi_{l,V}(X) = \mathbb{1}(X_{(l:l+L)} = V)$ which indicate if the L -mer at position l of X is V . The kernel is then $k_H(X, Y) = (\Phi(X) | \Phi(Y)) = \sum_{l=0}^{\infty} \sum_{V \in \mathcal{B}^L} \Phi_{l,V}(X) \Phi_{l,V}(Y)$ for sequences X, Y , i.e. it counts the number of L -mers found in the same position in X and Y . Note that in the case where $L = 1$, the representation $\Phi(X)$ reduces to the extremely popular *one-hot-encoding* representation of X , and the kernel takes the form $k_H(X, Y) = |X| \vee |Y| - d_H(X, Y)$, where d_H is the Hamming distance.

This kernel, however, might not be able to describe the true relationship between sequences and the phenotype that the scientist is interested in. The reason is that the kernel is not *universal*, and so has limited flexibility. To see this, we will show that functions f in \mathcal{H}_k are restricted to only take certain values. In particular, the values that a function f takes on set of sequences A_1 can be restrained by the values it takes on a subset $A_2 \subset A_1$. The problem arises as soon as the length of sequences in the data set exceed the length of the sliding window in the kernel, L . Define $A_1 = \{X\}_{|X|=L+1}$ to be the set of all sequences of length $L + 1$. Define $A_2 = \{X + X_{(0)}\}_{|X|=L}$ to be the set of those sequences of length $L + 1$ that end with the same letter they started with (namely, $X_{(0)}$). Now, when performing kernel regression, we are fitting a function $f = \sum_{n=1}^N \alpha_n k_H(Y_n, \cdot)$. We find that f must satisfy the constraint that its average value on A_1 equals its average value on A_2 ,

$$\begin{aligned} \frac{1}{|A_1|} \sum_{X \in A_1} f(X) &= \sum_{n=1}^N \alpha_n \left(\Phi(Y_n) \middle| \frac{1}{|A_1|} \sum_{X \in A_1} \Phi(X) \right) \\ &= \sum_{n=1}^N \alpha_n \left(\Phi(Y_n) \middle| \frac{1}{|A_2|} \sum_{X \in A_2} \Phi(X) \right) = \frac{1}{|A_2|} \sum_{X \in A_2} f(X). \end{aligned} \tag{8}$$

The reason is that for every L -mer $V \in \mathcal{B}^L$ and starting position $l \in \{0, 1\}$, the proportion of sequences that have V at position l is the same in A_2 as it is in A_1 , namely $|\mathcal{B}|^{-L}$. Thus, $\frac{1}{|A_1|} \sum_{X \in A_1} \Phi(X)_{l,V} = |\mathcal{B}|^{-L} = \frac{1}{|A_2|} \sum_{X \in A_2} \Phi(X)_{l,V}$ for all l, V , yielding Eqn. 8. The implication of Eqn. 8 is that functions in the RKHS are constrained, and cannot describe any possible relationship between sequences and their phenotypes; in other words, it says that the kernel is not universal. Note that this argument also generalizes to more complex versions of the Hamming kernel, such as those that consider all kmers up to length L , or those that score kmer similarity based on the biophysical properties of amino acids instead of exact matches (Schweikert et al., 2008; Toussaint et al., 2010).

The limited flexibility of the Hamming kernel can lead to serious practical failures, where the model is not just slightly wrong on some data points but instead makes terrible predictions for all data points. Say a scientist has taken measurements of all sequences of length $L + 1$, obtaining a data set $\{g(X) : X \in A_1\}$. They find that sequences $X \in A_2$ that start and end with the same letter have a score of $g(X) = |\mathcal{B}| - 1$, and those that do not (that is, $X \in A_1 \setminus A_2$) have a score of $g(X) = -1$. We assume there is no measurement error, i.e. $g(X)$ is the ground truth value of the phenotype. The least-squares fit for this big, clean data set, using kernel regression with the Hamming kernel, turns out to be awful: $f(X) = 0$ for all X . To see this, note that the least squares objective is minimized at $\alpha_1, \dots, \alpha_N = 0$,

$$\begin{aligned} \frac{d}{d\alpha} \sum_{X \in A_1} \frac{1}{2} (\alpha k_H(Y, X) - g(X))^2 \Big|_{\alpha=0} &= - \sum_{X \in A_1} k_H(Y, X) g(X) \\ &= \sum_{X \in A_1} k_H(Y, X) - |\mathcal{B}| \sum_{X \in A_2} k_H(Y, X) \\ &= |A_1| \left(\frac{1}{|A_1|} \sum_{X \in A_1} k_H(Y, X) - \frac{1}{|A_2|} \sum_{X \in A_2} k_H(Y, X) \right) = 0. \end{aligned}$$

The lack of kernel flexibility is thus a serious practical concern, as it can lead to terrible model performance even when we have large amounts of clean data.

Our theoretical results in Section 6 will provide a straightforward solution, in the form of a modified kernel: the inverse-multiquadratic Hamming (IMQ-H) kernel,

$$k_{\text{IMQ-H}}(X, Y) = \frac{1}{(1 + |X| \vee |Y| - (\Phi(X)|\Phi(Y)))^2}.$$

This kernel is just as tractable to compute as the original Hamming kernel, $k_H(X, Y) = (\Phi(X)|\Phi(Y))$. Like the original kernel, it treats sequences that have the same kmers in the same positions as similar to one another. Thus, if there is a biological reason to use a Hamming kernel for a specific problem, the same justification likely holds for a IMQ-H.⁵ However, we will prove that the IMQ-H has a fundamental advantage: it has discrete masses, which means that it is universal. If we perform a regression using the IMQ-H instead of the original Hamming kernel, we are guaranteed to be able to describe any sequence-to-property relationship.

We illustrate the benefits of our IMQ-H kernel in simulation. As the ground truth sequence-to-phenotype map, we consider

$$g(X) = \max_b \sum_{l=0}^{\infty} \mathbb{1}(X_l = b).$$

5. For instance, if we are examining promoters, whose biological activity depends on the arrangement of transcription factor binding sites, we have good biological reasons to believe that sequences with the same kmers in the same positions will have similar function, as transcription factors typically bind conserved sites of roughly 8 – 10 nucleotides. In this context, it makes sense to use $(\Phi(X)|\Phi(Y))$ as a similarity score, but the biology offers no particular reason to prefer k_H over $k_{\text{IMQ-H}}$.

which counts the number of occurrences of the most common letter in the sequence. We draw data points X_1, X_2, \dots from a uniform distribution on DNA sequences of length 4, and record noiseless observations of $g(X_n)$ for each. We estimate g using least-squares kernel regression, and compare two different kernels: the Hamming kernel with $L = 2$ and the IMQ-H kernel with $L = 1$. Naively, one might expect the IMQ-H kernel to perform more poorly, as it uses a smaller sliding window L . In fact, however, we find that regression with the Hamming kernel asymptotes at a high error as we collect more and more data, while with the IMQ-H the error converges to zero (Figure 1). Our theoretical guarantees can thus lead to dramatic performance gains at negligible computational cost.

Appendix B. A universal kernel that metrizes the space of distributions but lacks discrete masses

Here we show that having discrete masses is not strictly necessary in order to have a kernel that is universal or metrizes $\mathcal{P}(S)$. Recall Example 2, where the kernel failed to metrize $\mathcal{P}(S)$ because k_A could be well approximated by $k_{n \times A}$ in \mathcal{H}_k . To obtain a universal kernel that metrizes $\mathcal{P}(S)$ but lacks discrete masses we will create a kernel on the integers $\{0, 1, -1, 2, -1, \dots\}$ such that k_0 can be approximated by $\int d\nu_n(X)k_X$ for $\nu_n \in \mathcal{P}(S)$ but only if the total variation of ν_n blows up with n .

Let \mathcal{H}_k be a Hilbert space with an orthonormal basis indexed by the integers $\{e_0, e_1, e_{-1}, \dots\}$. Define $k_0 = e_0$ and for any positive n , $k_n = e_n$ and $k_{-n} = \sqrt{1 - \epsilon_n^2 - \epsilon_n^4}e_n + \epsilon_n^2 e_{-n} + \epsilon_n e_0$ for a decreasing sequence of positive ϵ_n that has $\epsilon_n \rightarrow 0$. Clearly k is a strictly positive definite kernel on the integers and $k(z, z) = 1$ for all integers z .

\mathcal{H}_k cannot contain δ_0 . To see this, say that $\delta_0 \in \mathcal{H}_k$. Then

$$\begin{aligned} 0 &= \frac{1}{\epsilon_n} (\delta_0(-n) - \delta_0(n)) \\ &= \left(\delta_0 \left| \frac{1}{\epsilon_n} (k_{-n} - k_n) \right. \right)_k \\ &= \left(\delta_0 \left| \frac{\sqrt{1 - \epsilon_n^2 - \epsilon_n^4} - 1}{\epsilon_n} e_n + \epsilon_n e_{-n} + e_0 \right. \right)_k \\ &\rightarrow (\delta_0 | e_0)_k = \delta_0(0) = 1, \end{aligned}$$

a contradiction.

However, k is C_0 -universal. To see this, say for some finite (possibly signed) measures on S , μ, ν , we have $\sum_z \nu(z)k_z = \sum_z \mu(z)k_z$. Call $\pi = \nu - \mu$. Note $0 = (\sum_z \pi(z)k_z | e_{-n})_k = \epsilon_n^2 \pi(-n)$ and $0 = (\sum_z \pi(z)k_z | e_n)_k = \pi(n) + \pi(-n)\sqrt{1 - \epsilon_n^2 - \epsilon_n^4}$. Thus $\pi(n) = 0$ for all $n \neq 0$. Finally, $0 = (\sum_z \pi(z)k_z | e_0)_k = \pi(0)$ so $\pi = 0$, i.e. $\mu = \nu$. By Proposition 2 of Sriperumbudur et al. (2011) k is universal.

k also metrizes $\mathcal{P}(S)$. To see this, say for some $\mu, \nu_1, \nu_2, \dots \in \mathcal{P}(S)$, we have $\sum_z \nu_m(z)k_z \rightarrow \sum_z \mu(z)k_z$. Call $\pi_m = \nu_m - \mu$. As above, $(\sum_z \pi_m(z)k_z | e_{-n})_k = \epsilon_n^2 \pi_m(-n)$ and $(\sum_z \pi_m(z)k_z | e_n)_k = \pi_m(n) + \pi_m(-n)\sqrt{1 - \epsilon_n^2 - \epsilon_n^4}$. For each n , the left hand sides of both of these equations go to 0 as $m \rightarrow \infty$ so, $\pi_m(n) \rightarrow 0$ for all $n \neq 0$. Finally note

$$\left(\sum_z \pi_m(z)k_z \left| e_0 \right. \right)_k = \pi_m(0) + \sum_{n=1}^{\infty} \pi_m(-n)\epsilon_n.$$

The left hand side goes to 0 and since $\sum_z |\pi_m(z)| \leq 2$ and $\epsilon_n \rightarrow 0$, the sum in the right hand side also goes to 0. Thus $\pi_m(0) \rightarrow 0$ and so $\nu_n \rightarrow \mu$.

Interestingly, although $\delta_0 \notin \mathcal{H}_k$, k is still able to approximate δ_0 arbitrarily well in infinity norm. First pick, for each n , an $f_n \in \text{span}\{k_0, k_n, k_{-n}\}$ such that $(f_n|k_0) = (f_n|k_n) = 0$ and $(f_n|k_{-n}) = 1$. Then $g_n = k_0 - \sum_{n=1}^N \epsilon_n f_n$ is k_0 on $\{0, N+1, -(N+1), N+2, -(N+2), \dots\}$ and 0 on $\{1, -1, \dots, N, -N\}$. As k_0 vanishes at infinity, g_n converges to δ_0 in infinity norm as $N \rightarrow \infty$. However, despite approximating δ_0 in infinity norm, $(g_n)_n$ does not approximate δ_0 in \mathcal{H}_k : $\|g_n\|_k$ grows unboundedly.⁶

Appendix C. Discrete masses in the alignment kernel

In this section we prove the alignment kernel has discrete masses under certain parameter settings.

We first establish some properties of sequences with encodings that are not one-hot. In particular, we show that convolution of kernels works in the same way when we reparameterize the alphabet. Note, for $V, W \in \cup_{L=0}^{\infty} \mathbb{R}^{L \times \mathcal{B}}$, we write $V_{(l)}, V_{(l:l')}$ to index positions along the first dimension and $V+W$ for concatenation along the first dimension, in analogy to the definitions for regular sequences in $S = \cup_{L=1}^{\infty} \mathcal{B}^L$.

Proposition 32 *Say k_1, k_2 are kernels on S and $V \in \mathbb{R}^{L \times \mathcal{B}}, V' \in \mathbb{R}^{L' \times \mathcal{B}}$.*

$$k_1 \star k_2(V, V') = \sum_{l_1+l_2=L, l'_1+l'_2=L'} k_1 \left(V_{(:l_1)}, V'_{(:l'_1)} \right) k_2 \left(V_{(l_1:)}, V'_{(l'_1:)} \right).$$

6. To see why $\|g_n\|_k$ must blow up, note that if this is not the case, then by the Banach-Alaoglu theorem a subsequence $(g_{n_k})_k$ must “weakly converge” to a $g \in \mathcal{H}_k$, that is, $g_{n_k}(z) = (g_{n_k}|k_z)_k \rightarrow (g|k_z)_k = g(z)$ for all z . Thus, $\delta_0 = g \in \mathcal{H}_k$, a contradiction.

Proof

$$\begin{aligned}
 & k_1 \star k_2(V, V') \\
 &= \sum_{|X|=L} \sum_{|Y|=L'} \left(\prod_{l=0}^{L-1} V_{l, X_{(l)}} \right) \left(\prod_{l=0}^{L'-1} V_{l, Y_{(l)}} \right) \sum_{l_1, l'_1} k_1(X_{(:, l_1)}, Y_{(:, l'_1)}) k_2(X_{(l_1, :)}, Y_{(l'_1, :)}) \\
 &= \sum_{l_1, l'_1} \sum_{|X|=L} \sum_{|Y|=L'} \left(\left(\prod_{l=0}^{l_1-1} V_{l, X_{(l)}} \right) \left(\prod_{l=0}^{l'_1-1} V_{l, Y_{(l)}} \right) k_1(X_{(:, l_1)}, Y_{(:, l'_1)}) \right) \\
 &\quad \times \left(\left(\prod_{l=l_1}^{L-1} V_{l, X_{(l)}} \right) \left(\prod_{l=l'_1}^{L'-1} V_{l, Y_{(l)}} \right) k_2(X_{(l_1, :)}, Y_{(l'_1, :)}) \right) \\
 &= \sum_{l_1, l'_1} \left(\sum_{|X|=l_1} \sum_{|Y|=l'_1} \left(\prod_{l=0}^{l_1-1} V_{l, X_{(l)}} \right) \left(\prod_{l=0}^{l'_1-1} V_{l, Y_{(l)}} \right) k_1(X, Y) \right) \\
 &\quad \times \left(\sum_{|X|=L-l_1} \sum_{|Y|=L'-l'_1} \left(\prod_{l=0}^{L-l_1-1} V_{l+l_1, X_{(l)}} \right) \left(\prod_{l=0}^{L'-l'_1-1} V_{l+l'_1, Y_{(l)}} \right) k_2(X, Y) \right) \\
 &= \sum_{l_1, l'_1} k_1(V_{(:, l_1)}, V'_{(:, l'_1)}) k_2(V_{(l_1, :)}, V'_{(l'_1, :)}).
 \end{aligned} \tag{9}$$

■

We now prove Theorem 21.

Theorem 33 (proof of Theorem 21) Define K as the matrix with $K_{b', b} = k_s(b, b')$ for $b, b' \in \mathcal{B}$, and $\sigma = \mathbf{1}_{\mathcal{B}}^T K^{-1} \mathbf{1}_{\mathcal{B}}$. If $\Delta\mu > 0$, then k has discrete masses if and only if $2\mu \geq \log \sigma$. If $\Delta\mu = 0$, then k has discrete masses if and only if $2\mu > \log \sigma$.

Proof The proof is organized as follows. First, we reparameterize \mathcal{B} such that \mathcal{H}_k decomposes into a set of orthogonal hyperplanes W_V . Second, we show that each of these hyperplanes is isomorphic to a tensor product of RKHSs of the alignment kernel with $|\mathcal{B}| = 1$. Finally, the result follows from Theorem 39, which establishes conditions under which alignment kernels with $|\mathcal{B}| = 1$ have discrete masses.

Part 1: We start by constructing the reparameterized alphabet. Call $\tilde{U} = K^{-1} \mathbf{1}_{\mathcal{B}}$, where $\mathbf{1}_{\mathcal{B}} \in \mathbb{R}^{\mathcal{B}}$ is the vector with 1 in each position, and let $U = \tilde{U} / k_s(\tilde{U}, \tilde{U})$, so, $U^T \mathbf{1}_{\mathcal{B}} = \tilde{U}^T K \tilde{U} / \tilde{U}^T K \tilde{U} = 1$ and $k_s(U, U) = (\tilde{U}^T K \tilde{U})^{-1} = k_s(\tilde{U}, \tilde{U})^{-1} = (\mathbf{1}_{\mathcal{B}}^T K^{-1} \mathbf{1}_{\mathcal{B}})^{-1} = \sigma^{-1}$. Let $B_1, \dots, B_{|\mathcal{B}|-1} \in \mathbb{R}^{\mathcal{B}}$ be chosen so that $\{U, B_1, \dots, B_{|\mathcal{B}|-1}\}$ is an orthogonal basis of the vector space $\mathbb{R}^{\mathcal{B}}$ with the dot product $(v|x) = v^T K w$, with $(B_i|B_i) = 1$ for all i . Thus, $k_s(B_i, B_j) = \delta_i(j)$ for all i, j and $k_s(U, B_i) = 0$ for all i . Now call $\tilde{\mathcal{B}} = \{U, B_1, \dots, B_{|\mathcal{B}|-1}\}$. We will use $\tilde{\mathcal{B}}$ as a reparameterized alphabet and note that by Proposition 17 the alignment kernel has discrete masses if and only if it has discrete masses when \mathcal{B} is replaced with $\tilde{\mathcal{B}}$.

Now, for any number L and $V \in \tilde{\mathcal{B}}^L$, the insertion kernel k_I applied to V is,

$$k_{I, V} = \sum_{|X|=L} \left(\prod_{l=0}^{L-1} V_{l, X_{(l)}} \right) e^{-\Delta\mu - L\mu} g = e^{-\Delta\mu - L\mu} \prod_{l=0}^{L-1} \left(\sum_{b \in \mathcal{B}} V_{l, b} \right) g.$$

Note for any $B \in \{B_1, \dots, B_{|\mathcal{B}|-1}\} \subset \tilde{\mathcal{B}}$, we have $\sum_{b \in \mathcal{B}} B_b = B^T \mathbf{1}_{\mathcal{B}} = B^T K \tilde{U} = k_s(B, U)/k_s(U, U) = \delta_U(B)$. Thus, $k_{I,V} = 0$ if any letters in V are in $\tilde{\mathcal{B}} \setminus \{U\}$. Now say $V \in \tilde{\mathcal{B}}^L$, $V' \in \tilde{\mathcal{B}}^{L'}$. By Proposition 32, $k(V, V')$ is given by

$$\sum_{l \in \mathbb{N}, V^{(1)} + \dots + V^{(2l+1)} = V, V'^{(1)} + \dots + V'^{(2l+1)} = V'} k_I(V^{(1)}, V'^{(1)}) \prod_{i=1}^l k_s(V^{(2i)}, V'^{(2i)}) k_I(V^{(2i+1)}, V'^{(2i+1)}).$$

Each term in the sum will be non-zero only if $V^{(m)}$ and $V'^{(m)}$ are composed only of the letter U for all odd m , and $V^{(m)} = V'^{(m)}$ and $|V^{(m)}| = 1$ for all even m . In particular, if $V \in (\tilde{\mathcal{B}} \setminus \{U\})^L$, sequences V' that satisfy $k(V, V') \neq 0$ must take the form

$$(t_0 \times U) + V_{(0)} + (t_1 \times U) + V_{(1)} + \dots + V_{(L-1)} + (t_L \times U)$$

for some integers t_0, \dots, t_L . Call this sequence $V[t]$. Let $W_V \subset S$ denote the set of all such sequences, i.e. $W_V = \{V[t] \mid t_0, \dots, t_N \in \mathbb{N}\}$. We see that W_V is orthogonal to any $W_{V'}$ for which $V' \notin W_V$. Each W_V thus defines an orthogonal hyperplane. Moreover, the union of all $\{W_V\}_V$ is the set of all sequences with letters in $\tilde{\mathcal{B}}$. Proposition 14 therefore says that if the alignment kernel has discrete masses when restricted to each W_V , it has discrete masses.

Part 2: In this section we find that, for the alignment kernel to have discrete masses over W_V , it is sufficient that a simpler alignment kernel, with an alphabet size of just one, have discrete masses. Say $V[t], V[t'] \in W_V$ for some $V \in (\tilde{\mathcal{B}} \setminus \{U\})^L$. Let $V[t_{1:}]$ denote $V[t]$ with the first $t_0 + 1$ letters dropped. Since the only alignments that have positive weight are those that align the i -th set of U s in $V[t]$ to the i -th set of U s in $V[t']$ for all i , we can decompose the alignment kernel into a product of alignment kernels comparing each of the strings of U s:

$$\begin{aligned} k(V[t], V[t']) &= \sum_{\substack{l \in \mathbb{N} \\ Y^{(1)} + \dots + Y^{(2l+1)} = t_0 \times U \\ Z^{(1)} + \dots + Z^{(2l+1)} = t'_0 \times U}} \sum_{\substack{l' \in \mathbb{N} \\ \tilde{Y}^{(1)} + \dots + \tilde{Y}^{(2l'+1)} = V[t_{1:}] \\ \tilde{Z}^{(1)} + \dots + \tilde{Z}^{(2l'+1)} = V[t'_{1:}]}} \left(\right. \\ &\quad \times \left(k_I(Y^{(1)}, Z^{(1)}) \prod_{i=1}^l k_s(Y^{(2i)}, Z^{(2i)}) k_I(Y^{(2i+1)}, Z^{(2i+1)}) \right) \\ &\quad \times k_s(V_{(0)}, V_{(0)}) \\ &\quad \times \left(k_I(\tilde{Y}^{(1)}, \tilde{Z}^{(1)}) \prod_{i=1}^{l'} k_s(\tilde{Y}^{(2i)}, \tilde{Z}^{(2i)}) k_I(\tilde{Y}^{(2i+1)}, \tilde{Z}^{(2i+1)}) \right) \left. \right) \\ &= k(t_0 \times U, t'_0 \times U) k(V[t_{1:}], V[t'_{1:}]) \\ &= \dots \\ &= \prod_{l=0}^L k(t_l \times U, t'_l \times U). \end{aligned} \tag{10}$$

Now call k_1 the alignment kernel when $\mathcal{B} = \{A\}$, with parameter $k_s(A, A) = \sigma^{-1}$. Since $k_s(U, U) = \sigma^{-1}$, we have $k_1(l \times A, l' \times A) = k(l \times U, l' \times U)$. Thus k on W_V acts as the

$L+1$ -times tensor product of k_1 . Thus, by Proposition 15, if k_1 has discrete masses, so does k on W_V for every V , so that k has discrete masses on S ; on the other hand, if k_1 does not have discrete masses, neither does k on W_\emptyset , so that k does not have discrete masses on S .

Finally, in Theorem 39 we will see that if $|\mathcal{B}| = 1$ then if $\Delta\mu > 0$, k_1 has discrete masses if and only if $2\mu + \log k(A, A) \geq 0$. Thus if $\Delta\mu > 0$, we find that k_1 , and therefore k , has discrete masses if and only if $2\mu \geq \log \sigma$. The result similarly follows for $\Delta\mu = 0$. \blacksquare

Note that this proof also extends to alignment kernels that have different penalties for inserting different letters. More precisely, we can modify k_I such that it differs for different letters by using $\tilde{g}(X) = \left(\prod_{l=0}^{|\mathcal{X}|-1} t(X_{(l)}) \right) g(X)$ for a function $t \in \mathbb{R}^{\mathcal{B}}$. Above we chose $t = \mathbf{1}_{\mathcal{B}}$. For general t define $U = K^{-1}t/t^T K^{-1}t$ and define $\tilde{\mathcal{B}} = \{U, B_1, \dots, B_{|\mathcal{B}|-1}\}$ the same way. Again, for any $B \in \tilde{\mathcal{B}}$, $\sum_b t(b)B_b = k_s(B, U)/k_s(U, U) = \delta_U(B)$, so, $k_{I,V} = 0$ for any sequence V that has any letters orthogonal to U . Calling $\sigma = t^T K^{-1}t$, we can repeat the proof of theorem 21 to reach the same conclusion.

Appendix D. Interpreting the conditions for discrete masses in alignment kernels

In Theorem 21 we saw that the alignment kernel has discrete masses only under certain conditions. An interesting aspect of these conditions is that they involve an interplay between the the kernel's insertion parameters (through μ) and the substitution parameters (through σ). Often these parameters are thought of as entirely distinct; after all, biologically, substitution and insertion mutations can stem from quite distinct molecular processes. Here we build some intuition for their interplay in determining discrete masses. We focus on the special case where $\Delta\mu = 0$.

The key idea is to rewrite the alignment kernel as a sum over a particular position-wise comparison kernel applied to each pairwise alignment; then, the conditions for the alignment kernel to have discrete masses turn out to be the same as the conditions for the corresponding position-wise kernel to have discrete masses. In detail, recall that Equation 6 decomposes the alignment kernel into a sum over pairwise alignments. For example, consider the alignment between $X = \text{ATGC}$ and $Y = \text{ACC}$ we saw in Section 7.1. We can rewrite the corresponding term in Equation 6 in the form

$$k'_s(\mathbf{A}, \mathbf{A})k'_s(\mathbf{T}, \mathbf{C}), k'_s(\mathbf{G}, -)k'_s(\mathbf{C}, \mathbf{C})k'_s(-, -)k'_s(-, -) \dots$$

where we define the extended letter comparison kernel k'_s as $k'_s(b, b') = k_s(b, b')$ for $b, b' \in \mathcal{B}$, $k'_s(b, -) = e^{-\mu}$ for $b \in \mathcal{B}$, and $k'_s(-, -) = 1$. This matches the form of a position-wise comparison kernel (Def. 18) between the sequences $\text{ATGC}--\dots$ and $\text{AC-C}--\dots$, with letter kernel k'_s , and where the gap symbol is also the stop symbol. In general, every term in the sum making up the alignment kernel (Equation 6) can be written in this way, with the same letter kernel.

Applying Theorem 19, the position-wise kernel has discrete masses if (and only if) k'_s is strictly positive definite.⁷ This, in turn, depends on the relationship between the

7. The “only if” direction is straightforward: if k'_s is not strictly positive definite, Theorem 11 is violated.

substitution and insertion parameters, since both go into k'_s . Let $V \in \mathbb{R}^{\mathcal{B} \cup \{-\}}$ such that $V_- = 1$. Call $\tilde{V} = (V_b)_{b \in \mathcal{B}}$ the components of V in \mathcal{B} . Define $\tilde{U} = K^{-1} \mathbf{1}_{\mathcal{B}}$ so $\tilde{U} K \tilde{U} = \sigma$. Now,

$$k'_s(V, V) = \tilde{V}^T K \tilde{V} + 2e^{-\mu} \tilde{V}^T \mathbf{1}_{\mathcal{B}} + 1 = \tilde{V}^T K \tilde{V} + 2e^{-\mu} \tilde{V}^T K \tilde{U} + 1.$$

Now decompose \tilde{V} as $\tilde{V} = \alpha \tilde{U} + \tilde{V}^\perp$ where the columns of \tilde{V}^\perp are orthogonal to those of \tilde{U} under the inner product $(v|w) = v^T K w$. Then,

$$k'_s(V, V) = \tilde{V}^{\perp T} K \tilde{V}^\perp + \alpha^2 \tilde{U} K \tilde{U} + 2\alpha e^{-\mu} \tilde{U} K \tilde{U} + 1 = \tilde{V}^{\perp T} K \tilde{V}^\perp + \sigma (\alpha^2 + 2e^{-\mu} \alpha) + 1.$$

This expression is minimized at $\alpha = -e^{-\mu}$, $\tilde{V}^\perp = 0$, in which case

$$k'_s(V, V) = -\sigma e^{-2\mu} + 1.$$

Thus k'_s is strictly positive definite if and only if $e^{-2\mu} \sigma < 1$, i.e. $2\mu > \log \sigma$. Therefore, the alignment kernel has discrete masses if and only if the position-wise comparison kernel has discrete masses.

At bottom, then, the flexibility of the alignment kernel depends on the flexibility of the kernel we use for comparing aligned sequences, and this in turn depends on the flexibility of the kernel we use for comparing letters – both real letters (nucleotides/amino acids) and gaps. The interplay between substitution and insertion parameters stems from the fact that we have in essence added the gap symbol to the alphabet, and need the letter kernel over this extended alphabet to still be strictly positive-definite.

Appendix E. Features of the alignment kernel

In this section, we derive a representation of the alignment kernel in terms of features. These features are gapped kmer counts.

Theorem 34 (*proof of Theorem 24*) *Consider an alignment kernel k with $k_s(b, b') = \sigma^{-1} |\mathcal{B}| \delta_{b'}(b)$ for $b, b' \in \mathcal{B}$. Tilting the kernel by $A(X) = e^{\mu|X|}$ gives $k^A(X, Y) = A(X)A(Y)k(X, Y)$. For all sequences $X \in S$, define the features $\{u_V(X)\}_{V \in S}$, given by*

$$u_V(X) = e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} \mathbb{1}(X_{(J)} = V), \quad (11)$$

where $\zeta = 2\mu - \log \sigma + \log |\mathcal{B}|$. Now we show that $\{u_V(X)\}_{V \in S}$ is an orthonormal basis for \mathcal{H}_{k^A} .

Proof We can represent alignments between two sequences $X \in S$ and $Y \in S$ by pairs of indices $J \in G(|X|, M)$, $J' \in G(|Y|, M)$ that describe which letters in X and Y are aligned. In particular, $k(X, Y)$ is equal to

$$\begin{aligned} \sum_{M=0}^{\infty} \sum_{(j_i)_{i=-1}^M \in G(|X|, M)} \sum_{(j'_i)_{i=-1}^M \in G(|Y|, M)} k_i(X_{(:j_0)}, Y_{(:j'_0)}) \\ \times \prod_{m=0}^{M-1} k_s(X_{(j_m)}, Y_{(j'_m)}) k_i(X_{(j_{m+1}:j_{m+1})}, Y_{(j'_{m+1}:j'_{m+1})}) \end{aligned}$$

By our assumption on k_s , for each alignment $J \in G(|X|, M, g)$, $J' \in G(|Y|, M, g')$, the contribution from the aligned letters, $\prod_{m=0}^{M-1} k_s(X_{(j_m)}, Y_{(j'_m)})$, is 0 if $X_{(j_m)} \neq Y_{(j'_m)}$ and is $\sigma^{-M} |\mathcal{B}|^M$ otherwise. On the other hand, the contribution from the insertions and deletions is $\prod_{m=0}^{M-1} k_I(X_{(j_m:j_{m+1})}, Y_{(j'_m+1:j'_{m+1})}) = e^{-\mu(|X|-M+|Y|-M)} e^{-\Delta\mu(g+g')}$. Thus, $k^A(X, Y)$ is equal to

$$\begin{aligned} & \sum_{M=0}^{\infty} (e^{2\mu} \sigma^{-1} |\mathcal{B}|)^M \sum_{g, g'} e^{-\Delta\mu(g+g')} \sum_{J \in G(|X|, M, g)} \sum_{J' \in G(|Y|, M, g')} \mathbb{1}(X_{(J)} = Y_{(J')}) \\ &= \sum_{M=0}^{\infty} e^{\zeta M} \sum_{V \in \mathcal{B}^M} \sum_{g, g'} e^{-\Delta\mu(g+g')} \sum_{J \in G(|X|, M, g)} \sum_{J' \in G(|Y|, M, g')} \mathbb{1}(X_{(J)} = V) \mathbb{1}(Y_{(J')} = V) \quad (12) \\ &= \sum_{M=0}^{\infty} \sum_{V \in \mathcal{B}^M} u_V(X) u_V(Y) \end{aligned}$$

Thus the features of k^A are $X \mapsto (u_V(X))_{V \in S}$.

Now we will show that $u_V \in \mathcal{H}_k$ for all $V \in S$ and that $(u_V)_V$ make up an orthonormal basis. Define the infinite matrix Q as $Q_{V,Z} = e^{-\frac{1}{2}\zeta|V|} u_V(Z)$ for all $V, Z \in S$. Pick some total ordering, \leq , of S such that $V \leq Z$ if $|V| \leq |Z|$. In this case, Q is an infinite upper triangular matrix with 1's along the diagonal. Thus Q has an infinite upper triangular matrix Q^{-1} as its inverse with 1s along its diagonal. In the remainder of the proof, we will represent any $\sum_Z \alpha_Z k_Z^A \in \text{span}\{k_Z^A\}_{Z \in S}$ as an infinite vector, indexed by elements of S , with α_Z in position Z . So, for any function $f \in \text{span}\{k_Z^A\}_{Z \in S}$, we can think of the matrix-vector product Qf as the featurization of f . Now, for any $f, g \in \text{span}\{k_Z^A\}_{Z \in S}$, we have from Equation 12,

$$(f|g)_{k^A} = f^T Q^T \text{diag}(e^{\zeta|V|})_{V \in S} Qg,$$

where $\text{diag}(e^{\zeta|V|})_{V \in S}$ denotes the infinite diagonal matrix with entries $e^{\zeta|V|}$.

We will next show that $u_X \in \mathcal{H}_{k^A}$, with the representation $f_X = e^{-\frac{1}{2}\zeta|X|} \sum_Z Q_{Z,X}^{-1} k_Z \in \text{span}\{k_Z^A\}_{Z \in S}$ (note the sum has only finitely many non-zero terms as $Q_{Z,X}^{-1} \neq 0$ for only finitely many Z with $Z \leq X$). We have,

$$\text{diag}(e^{\zeta|V|})_{V \in S} Q f_X = e^{\frac{1}{2}\zeta|X|} d_X,$$

where d_X is the infinite-dimensional vector indexed by S with 0's in every position except 1 in position X . For any $Y \in S$, $(f_X | k_Y^A)_{k^A} = e^{\frac{1}{2}\zeta|X|} Q_{X,Y} = u_X(Y)$. Thus, $u_X \in \mathcal{H}_{k^A}$. We also have,

$$(u_X | u_Y)_{k^A} = f_X^T Q^T \text{diag}(e^{\zeta|V|})_{V \in S} Q f_Y = d_X^T d_Y = d_X(Y).$$

Thus $\{u_X\}_{X \in S}$ is a set of orthogonal non-zero vectors, normalized such that $\|u_X\|_{k^A} = 1$ for all $X \in S$. Finally, by Equation 12,

$$k^A(X, Y) = \sum_{V \in S}^{\infty} (u_V | k_X^A)_{k^A} (u_V | k_Y^A)_{k^A}$$

for all $X, Y \in S$ so that $\{u_X\}_{X \in S}$ is also a basis. ■

We now consider the general case, allowing k_s to be any strictly positive definite kernel on letters, instead of requiring that it is diagonal. The first step is to show that any alignment kernel can be written as a reparameterization of an alignment kernel with diagonal k_s .

Proposition 35 *Consider an alignment kernel k , with parameters μ , $\Delta\mu$ and k_s . Let $K_{b,b'} = k_s(b, b')$ for $b, b' \in \mathcal{B}$ and $\sigma = \mathbf{1}_{\mathcal{B}}^T K^{-1} \mathbf{1}_{\mathcal{B}}$. There is an alphabet $\tilde{\mathcal{B}}$, inducing the set of sequences \tilde{S} , such that k applied to \tilde{S} is an alignment kernel with parameters $\mu, \Delta\mu$ and $k_s(\tilde{b}, \tilde{b}') = \sigma^{-1} |\mathcal{B}| \delta_{\tilde{b}}(\tilde{b}')$ for all $\tilde{b}, \tilde{b}' \in \tilde{\mathcal{B}}$. As well, for any $b \in \mathcal{B}$, $\sum_{\tilde{b} \in \tilde{\mathcal{B}}} k_s(\tilde{b}, b) = \sigma^{-1} |\mathcal{B}|$. We call $\tilde{\mathcal{B}}$ a rectified alphabet for k and \tilde{S} a rectified set of sequences for k .*

Proof Let $B_1 = K^{-1} \mathbf{1}_{\mathcal{B}} / \sqrt{\sigma}$ so $k_s(B_1, B_1) = 1$ and let $\{B_2, \dots, B_{|\mathcal{B}|}\}$ be an orthonormal set orthogonal to B_1 in $\mathbb{R}^{\mathcal{B}}$ with inner product k_s . Let $\Phi \in \mathbb{R}^{|\mathcal{B}| \times |\mathcal{B}|}$ be a unitary matrix with $|\mathcal{B}|^{-1/2}$ in every entry of its first column and define the letters $C_i = \sqrt{\sigma^{-1} |\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \Phi_{i,j} B_j$. Thus we have that $k_s(C_i, C_j) = \sigma^{-1} |\mathcal{B}| (\Phi_{i,\cdot} | \Phi_{j,\cdot}) = \sigma^{-1} |\mathcal{B}| \delta_i(j)$ and $\sum_{b \in \mathcal{B}} C_{i,b} = C_i^T \mathbf{1}_{\mathcal{B}} = \sqrt{\sigma} C_i^T K B_1 = \sqrt{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \Phi_{i,j} k_s(B_j, B_1) = \sqrt{|\mathcal{B}|} \Phi_{i,1} = 1$.

Call $\tilde{\mathcal{B}} = \{C_i\}_i$ and \tilde{S} the set of sequences with letters in $\tilde{\mathcal{B}}$. By Proposition 32, for $V, V' \in \tilde{S}$, $k(V, V')$ is given by

$$\sum_{l \in \mathbb{N}, V^{(1)} + \dots + V^{(2l+1)} = V, V'^{(1)} + \dots + V'^{(2l+1)} = V'} k_I(V^{(1)}, V'^{(1)}) \prod_{i=1}^l k_s(V^{(2i)}, V'^{(2i)}) k_I(V^{(2i+1)}, V'^{(2i+1)}).$$

For $\tilde{b}, \tilde{b}' \in \tilde{\mathcal{B}}$, $k_s(\tilde{b}, \tilde{b}') = \sigma^{-1} |\mathcal{B}| \delta_{\tilde{b}}(\tilde{b}')$. As well, for any number L and $V \in \tilde{\mathcal{B}}^L$, the insertion kernel k_I applied to V is,

$$k_{I,V} = \sum_{|X|=L} \left(\prod_{l=0}^{L-1} V_{l, X_{(l)}} \right) e^{-\Delta\mu - L\mu} g = e^{-\Delta\mu - L\mu} \prod_{l=0}^{L-1} \left(\sum_{b \in \mathcal{B}} V_{l,b} \right) g = e^{-\Delta\mu - L\mu} g.$$

Thus, for $V, V' \in \tilde{S}$ with $|V|, |V'| > 0$,

$$k_I(V, V') = e^{-\Delta\mu - |V|\mu} e^{-\Delta\mu - |V'|\mu}.$$

Thus k on \tilde{S} is the alignment kernel with the same parameters $\mu, \Delta\mu$ and with $k_s(\tilde{b}, \tilde{b}') = \sigma^{-1} |\mathcal{B}| \delta_{\tilde{b}}(\tilde{b}')$ for $\tilde{b}, \tilde{b}' \in \tilde{\mathcal{B}}$.

Finally, note, for any $b \in \mathcal{B}$,

$$\begin{aligned} \sum_i k_s(C_i, b) &= \sqrt{\sigma^{-1} |\mathcal{B}|} \sum_{i,j} \Phi_{i,j} k_s(B_j, b) \\ &= \sqrt{\sigma^{-1} |\mathcal{B}|} \sum_j \left(\mathbf{1}_{|\mathcal{B}|}^T \Phi_{\cdot,j} \right) k_s(B_j, b) \\ &= \sqrt{\sigma^{-1} |\mathcal{B}|} \sum_j \sqrt{|\mathcal{B}|} \delta_1(j) k_s(B_j, b) \\ &= \sqrt{\sigma^{-1} |\mathcal{B}|} k_s(B_1, b) \\ &= \sigma^{-1} |\mathcal{B}|. \end{aligned}$$

■

We can now describe the features of an alignment kernel with arbitrary k_s . When k_s was diagonal, each feature of a sequence X depended on the number of positions at which a gapped kmer matched X exactly. Now, instead of demanding an exact match, we score each match according to k_s .

Corollary 36 *Consider an alignment kernel k and let \tilde{S} be a rectified set of sequences. For $Y \in S$, $V \in \tilde{S}$ with $L = |Y| = |V|$, define the pairwise comparison using k_s as,*

$$c_{pw}(Y, V) = \prod_{l=0}^{L-1} \sigma |\mathcal{B}|^{-1} k_s(Y_{(l)}, V_{(l)}).$$

Tilting the kernel by $A(X) = e^{\mu|X|}$ gives $k^A(X, Y) = A(X)A(Y)k(X, Y)$. For all sequences $X \in S$, define the features $\{u_V(X)\}_{V \in \tilde{S}}$, given by

$$u_V(X) = e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} c_{pw}(X_{(J)}, V), \quad (13)$$

where $\zeta = 2\mu - \log \sigma + \log |\mathcal{B}|$. Now, $\{u_V(X)\}_{V \in \tilde{S}}$ is an orthonormal basis for \mathcal{H}_{k^A} .

Proof Let $\tilde{\mathcal{B}}$, \tilde{S} a rectified alphabet and a rectified set of sequences for k . By Theorem 24, there is an orthonormal basis of \mathcal{H}_{k^A} , $\{u_V\}_{V \in \tilde{S}}$ such that for any $V, X \in \tilde{S}$

$$(u_V | k_X^A)_{k^A} = e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} \mathbb{1}(X_{(J)} = V).$$

Note that reparameterization does not affect the RKHS \mathcal{H}_{k^A} , so that $\{u_V\}_{V \in \tilde{S}}$ is an orthonormal basis for the kernel k^A applied to the space S as well.

Now, consider a $Y \in S$, with $L = |Y|$. Since $\tilde{\mathcal{B}}$ is a basis for $\mathbb{R}^{\tilde{\mathcal{B}}}$, we can write each letter $Y_{(l)}$ as $Y_{(l)} = \sum_{\tilde{b} \in \tilde{\mathcal{B}}} Y_{(l), \tilde{b}} \tilde{b}$ for some scalar coefficients $\{Y_{(l), \tilde{b}}\}_{\tilde{b} \in \tilde{\mathcal{B}}}$ (to be clear, we are treating $Y_{(l)}$ here as a one-hot encoding of the l th letter of Y). To derive the coefficients $Y_{(l), \tilde{b}}$, recall that $\{\sqrt{\sigma|\mathcal{B}|^{-1}} \tilde{b}\}_{\tilde{b} \in \tilde{\mathcal{B}}}$ is an orthonormal basis of $\mathbb{R}^{\tilde{\mathcal{B}}}$, with inner product defined by k_s . Thus, we have $Y_{(l), \tilde{b}} = k_s(Y_{(l)}, \sqrt{\sigma|\mathcal{B}|^{-1}} \tilde{b}) \sqrt{\sigma|\mathcal{B}|^{-1}} = \sigma|\mathcal{B}|^{-1} k_s(Y_{(l)}, \tilde{b})$ for all $\tilde{b} \in \tilde{\mathcal{B}}$. We also have, applying the properties of the rectified alphabet in Proposition 35, $\sum_{\tilde{b} \in \tilde{\mathcal{B}}} Y_{(l), \tilde{b}} =$

$\sigma|\mathcal{B}|^{-1} \sum_{\tilde{b} \in \tilde{\mathcal{B}}} k_s(Y_{(l)}, \tilde{b}) = 1$. Thus,

$$\begin{aligned}
 (u_V | k_Y^A)_k^A &= \sum_{X \in \tilde{S} : |X|=L} \left(\prod_{l=0}^{L-1} Y_{(l), X_{(l)}} \right) e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} \mathbb{1}(X_{(J)} = V) \\
 &= e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} \sum_{X \in \tilde{S} : |X|=L} \left(\prod_{l=0}^{L-1} Y_{(l), X_{(l)}} \right) \prod_{m=0}^{M-1} \mathbb{1}(X_{(j_m)} = V_{(m)}) \\
 &= e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} \left(\prod_{l \notin J} \sum_{b \in \tilde{\mathcal{B}}} Y_{(l), b} \right) \prod_{m=0}^{M-1} Y_{(j_m), V_{(m)}} \\
 &= e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} 1 \times \prod_{m=0}^{M-1} \sigma|\mathcal{B}|^{-1} k_s(Y_{(j_m)}, V_{(m)}) \\
 &= e^{\frac{1}{2}\zeta|V|} \sum_{g=0}^{\infty} e^{-\Delta\mu g} \sum_{J \in G(|X|, g)} c_{pw}(Y_{(J)}, V)
 \end{aligned}$$

■

Appendix F. Alignment kernels are universal and characteristic for certain tiltings

In this section we show that all alignment kernels are, under an appropriate tilting, universal and characteristic—even if they do not have discrete masses. Note that while a kernel with discrete masses maintains discrete masses after any tilting, the same is not true for universal or characteristic kernels; here, the tilting matters.

Functions in C_0 vanish at infinity, so to be sure the alignment kernel can approximate any C_0 function, even when the kernel does not have discrete masses, the tilting will have to be sufficiently small. The normalizing tilting $A(X) = \sqrt{k(X, X)}^{-1}$ is not necessarily small enough. Instead, one would in general need to tilt so much that $k^A(X, X) \rightarrow 0$ as $|X| \rightarrow \infty$. From a modeling perspective, this is problematic: if f is drawn from a Gaussian process with kernel k^A , then its variance will go to zero as the length of sequences increases: $\text{Var}(f(X)) = k^A(X, X) \rightarrow 0$ as $|X| \rightarrow \infty$. Thus, on longer sequences, the value of f will be determined essentially just by the prior. So, although we can force alignment kernels that lack discrete masses to be universal if we tilt them enough, the tilted kernels will generalize poorly.

Proposition 37 *Say k is an alignment kernel and let \tilde{k} be an alignment kernel with the same values of $\mu, \Delta\mu$ as k but with $\tilde{k}_s(b, b') = \sigma e^{-2\mu+\epsilon} k_s(b, b')$ for some $\epsilon > 0$. Choose an $A : S \rightarrow (0, \infty)$ such that \tilde{k}^A is a C_0 -kernel, meaning $Y \mapsto A(Y)A(X)\tilde{k}(X, Y)$ is in $C_0(S)$ for every X and $\sup_{X \in S} A(X)A(X)\tilde{k}(X, X) < \infty$. Then the tilted alignment kernel k^A is C_0 -universal and characteristic.*

Proof Assume throughout that $2\mu \leq \log \sigma$, otherwise the result follows from Theorem 21 and Corollary 16. Note $\tilde{k}(X, Y) \geq k(X, Y)$ for all $X, Y \in S$. Thus, k^A is a C_0 -kernel. Also

note that, defining $\tilde{K}_{b,b'} = \tilde{k}_s(b, b')$, we have $\log(\mathbf{1}_B \tilde{K}^{-1} \mathbf{1}_B) = 2\mu - \epsilon < 2\mu$, so \tilde{k} has discrete masses by Theorem 21.

We now derive an orthogonal basis for $\mathcal{H}_{\tilde{k}^A}$. Let $\bar{A}(X) = \exp(-\mu|X|)$ and let \tilde{S} be a rectified set of sequences for k^A . Now, by Corollary 36, $\{A\bar{A}u_V\}_{V \in \tilde{S}}$ is an orthonormal basis of \mathcal{H}_{k^A} . Note \tilde{S} is also a rectified set of sequences for \tilde{k}^A , so let $\{A\bar{A}\tilde{u}_V\}_{V \in \tilde{S}}$ be the orthonormal basis of $\mathcal{H}_{\tilde{k}^A}$ from Corollary 36. Note c_{pw} from Corollary 36 is identical for k^A and \tilde{k}^A so that each \tilde{u}_V is a scaled version of u_V ; in particular, $\{A\bar{A}u_V\}_{V \in \tilde{S}}$ is an orthogonal basis for $\mathcal{H}_{\tilde{k}^A}$.

Since \tilde{k} has discrete masses, it is universal, so for any $f \in C_0(S)$ and $\epsilon > 0$ we can pick a $g \in \mathcal{H}_{\tilde{k}}$ such that $\|f - g\|_\infty < \epsilon$. Define $M = \sup_{X \in S} \sqrt{A(X)A(X)\tilde{k}(X, X)} < \infty$ and choose a $g' \in \text{span}\{\tilde{k}_X^A\}_X$ such that $\|g' - g\|_{\tilde{k}^A} \leq \epsilon/M$. Now, $g' \in C_0(S)$ and

$$|g'(X) - g(X)| = |(\tilde{k}_X^A |g' - g|_{\tilde{k}^A})| \leq \|\tilde{k}_X^A\|_{\tilde{k}^A} \|g' - g\|_{\tilde{k}^A} \leq M\epsilon/M = \epsilon$$

for all $X \in S$. Thus, $\|g' - f\|_\infty \leq 2\epsilon$. Thus, every function in $\mathcal{H}_{\tilde{k}^A}$ can be approximated in C_0 by a function in $\text{span}\{\tilde{k}_X^A\}_X$. Thus, $\text{span}\{\tilde{k}_X^A\}_X$ is dense in $C_0(S)$.

Notice the finite dimensional vector spaces $\text{span}\{\tilde{k}_X^A\}_{X \in S : |X| \leq L}$ and $\text{span}\{A\bar{A}u_X\}_{X \in \tilde{S} : |X| \leq L}$ are identical for all L (recall from the proof of Proposition 17 that reparameterizing the alphabet preserves the span of the kernel embedding vectors). As a result, $\text{span}\{\tilde{k}_X^A\}_{X \in S} = \text{span}\{A\bar{A}u_X\}_{X \in \tilde{S}}$. But this latter vector space is similarly equal to $\text{span}\{k_X^A\}_X$. Thus, $\text{span}\{k_X^A\}_X = \text{span}\{\tilde{k}_X^A\}_X$, so $\text{span}\{k_X^A\}_X$ is dense in $C_0(S)$. k^A is thus universal. Further, by proposition 2 of Sriperumbudur et al. (2011), k^A is also characteristic. ■

To interpret this result, and the tiltings it requires, note that we must choose a tilting A such that $\tilde{k}^A(X, X)$ is bounded. Roughly speaking, we expect $\frac{\tilde{k}(X, X)}{k(X, X)}$ to grow exponentially with the length $|X|$, since each match in each pairwise alignment has been up-weighted by a factor of $\sigma e^{-2\mu+\epsilon} > 1$. The normalizing tilting $A(X) = \sqrt{k(X, X)}^{-1}$ would help us bound $k^A(X, X)$, but to bound $\tilde{k}^A(X, X)$ we need to go further and exponentially down-weight longer sequences. This means choosing a tilting A such that $k^A(X, X) \rightarrow 0$ as $|X| \rightarrow 0$.

Appendix G. Discrete masses in the one-letter alignment kernel

In this section we show that alignment kernels have discrete masses under certain parameter settings *when the alphabet size is one*. This result is crucial for proving the more general result, covering arbitrary alphabet sizes, in Theorem 21 and Appendix C. It builds on the feature representation described in Theorem 24 and Appendix E.

The result will depend on the theory of the Riordan group, for which we will need to introduce some notation. If $f(x) = \sum_{i=0}^{\infty} C_i x^i$ is a formal power series, define $[x^n]f = C_n$. For two formal power series $f(y), g(y)$ we define the Riordan array of f and g as the infinite upper triangular matrix with entries

$$Q_{L', L} = [x^{L'} y^L] \frac{g(y)}{1 - xyf(y)} = [x^{L'} y^L] \sum_{l=0}^{\infty} g(y) x^l y^l f(y)^l = [y^L] y^{L'} g(y) f(y)^{L'}$$

for numbers $L, L' \geq 0$. We say that Q is in the Bell subgroup if $f = g$. The result we will need is that one can invert a Riordan array by inverting the power series f and g (Shapiro et al., 1991).

Proposition 38 (*Shapiro et al., 1991*) *Let*

$$Q_{L',L} = [x^{L'} y^L] \frac{g(y)}{1 - xyf(y)}$$

and let \bar{f} be the unique power series such that the equation $y\bar{f}(y)f(y\bar{f}(y)) = y$ holds.⁸ Then, letting $\bar{g}(y) = (g(y\bar{f}(y)))^{-1}$,

$$Q_{L',L}^{-1} = [x^{L'} y^L] \frac{\bar{g}(y)}{1 - xy\bar{f}(y)}.$$

Note if Q is in the Bell subgroup, $g(y) = f(y)$ and $\bar{f}(y)f(y\bar{f}(y)) = 1$ so $\bar{g}(y) = \bar{f}(y)$. Thus Q^{-1} is also in the Bell subgroup.

Proof

$$\begin{aligned} \sum_{l=0}^{\infty} Q_{L,l} Q_{l,L'}^{-1} &= \sum_{l=0}^{\infty} [y^l] g(y) (yf(y))^L [z^{L'}] \bar{g}(z) (z\bar{f}(z))^l \\ &= [z^{L'}] \bar{g}(z) \sum_{l=0}^{\infty} (z\bar{f}(z))^l [y^l] g(y) (yf(y))^L \\ &= [z^{L'}] \bar{g}(z) g(z\bar{f}(z)) (z\bar{f}(z)f(z\bar{f}(z)))^L \\ &= [z^{L'}] 1 \times z^L = \delta_{L'}(L). \end{aligned}$$

■

Theorem 39 *Say $|\mathcal{B}| = 1$ and call A the only letter in \mathcal{B} . If $\Delta\mu > 0$, k has discrete masses if and only if $2\mu + \log k_s(A, A) \geq 0$. If $\Delta\mu = 0$, k has discrete masses if and only if $2\mu + \log k_s(A, A) > 0$.*

Proof Note in this case $\zeta = 2\mu + \log k_s(A, A)$. We will replace k with $k^A(X, Y) = e^{\mu|X|} e^{\mu|Y|} k(X, Y)$ throughout. Clearly, this does not affect whether or not the kernel has discrete masses. Index $S = \{0, 1, \dots\}$. Call Q the infinite upper triangular matrix with $e^{-\frac{1}{2}\zeta L'} u_{L'}(L) = \sum_{g=0}^{\infty} e^{-g\Delta\mu} |G(L, L', g)|$ in position (L', L) . Note that the Gram matrix of k on $\{0, 1, \dots, L\}$ is $Q_{:L+1, :L+1}^T \text{diag}(e^{\zeta L'})_{L'=0}^L Q_{:L+1, :L+1}$. Since Q is infinite upper triangular with no zeros along its diagonal it has an infinite upper triangular inverse Q^{-1} with the property that $(Q_{:L, :L})^{-1} = (Q^{-1})_{:L, :L}$. Thus the inverse of the Gram matrix of k on $\{0, 1, \dots, L\}$ is $Q_{:L+1, :L+1}^{-1T} \text{diag}(e^{-\zeta L'})_{L'=0}^L Q_{:L+1, :L+1}^{-1}$ which has, in position (L', L') , $\sum_{l=L'}^L e^{-\zeta l} (Q_{L', l}^{-1})^2$. Thus, by Proposition 11, $\delta_{L'} \in \mathcal{H}_k$ if and only if

$$\sum_{l=L'}^{\infty} e^{-\zeta l} (Q_{L', l}^{-1})^2 < \infty. \quad (14)$$

8. Another way to write this condition is that $tf(t) = y$ when $t = y\bar{f}(y)$ so that yf is the compositional inverse of $y\bar{f}$.

The proof will now proceed in two parts using the technique of generating functions and the theory of the Riordan group. In part 1 we will equate $u_L(L')$ with the coefficients of a formal power series. We will then explicitly invert the matrix Q by inverting this power series. In part 2 we will equate Equation 14 with the convergence of a power series, which we will use to prove the result.

Part 1: Say g is an integer greater than or equal to 1, and $L \geq L'$. We want to find the size of $G(L, L', g)$. Each $J = (j_{-1}, \dots, j_{L'}) \in G(L, L', g)$ is uniquely characterized by the numbers j_{m_1}, \dots, j_{m_g} , which indicate the positions at which each of the g gaps start, and $j_{m_1+1}, \dots, j_{m_g+1}$, which indicate the position at which each of the g gaps end. In particular,

$$J = (-1, 0, 1, \dots, j_{m_1} - 1, j_{m_1}, j_{m_1+1}, j_{m_1+1} + 1, \dots, j_{m_2} - 1, j_{m_2}, j_{m_2+1}, \dots, L).$$

Thus each J is also uniquely characterized by the sizes of its g gaps $l_1 = j_{m_1+1} - j_{m_1} - 1, l_2 = j_{m_2+1} - j_{m_2} - 1, \dots$ as well as the sizes of its $g + 1$ contiguous regions $c_0 = j_{m_1} + 1, c_1 = j_{m_2} - j_{m_1+1} + 1, c_2 = j_{m_3} - j_{m_2+1} + 1, \dots, c_g = L - j_{m_g+1}$. On the other hand, given numbers $l_1, \dots, l_g, c_0, \dots, c_g$, one gets a unique valid $J \in G(L, L', g)$ if and only if $\sum_{i=0}^g c_i = L', \sum_{i=1}^g l_i = L - L', l_i \geq 1$ for all i , and $c_i \geq 1$ for all $i = 1, 2, \dots, g - 1$ ($c_0 = 0$ implies a gap at the beginning of the sequence while $c_g = 0$ implies a gap at the end). The number of $g + 1$ ordered numbers (c_0, \dots, c_g) with $\sum_{i=0}^g c_i = L'$ and $c_i \geq 1$ for all $i = 1, 2, \dots, g - 1$ is the L' -th coefficient of the power series

$$(1 + x + x^2 + \dots) (x + x^2 + \dots)^{g-1} (1 + x + x^2 + \dots) = \frac{x^{g-1}}{(1-x)^{g+1}}.$$

Similarly, the number of sets of number of g ordered numbers (l_1, \dots, l_g) with $\sum_{i=1}^g l_i = L - L'$ and $l_i \geq 1$ for all i is the $L - L'$ -th coefficient of the power series $\frac{y^g}{(1-y)^g}$. On the other hand if $g = 0$, $G(L, L', g)$ is empty if $L \neq L'$ otherwise it has one element. In particular, $G(L, L', 0)$ is (trivially) the coefficient in front of $x^{L'} y^{L-L'}$ of $1/(1-x)$. Thus, the size of $G(L, L', g)$ is the coefficient in front of $x^{L'} y^{L-L'} z^g$ of the power series

$$\begin{aligned} \sum_{g=1}^{\infty} z^g \frac{x^{g-1}}{(1-x)^{g+1}} \frac{y^g}{(1-y)^g} + \frac{1}{1-x} &= \frac{zy}{(1-x)^2(1-y)} \sum_{g=0}^{\infty} \left(z \frac{x}{1-x} \frac{y}{1-y} \right)^g + \frac{1}{1-x} \\ &= \frac{zy}{(1-x)^2(1-y)} \frac{(1-x)(1-y)}{(1-x)(1-y) - zxy} + \frac{1}{1-x} \\ &= \frac{1}{1-x} \left(\frac{zy}{(1-x)(1-y) - zxy} + 1 \right) \\ &= \frac{1}{1-x} \left(\frac{zy + 1 - x - y + xy - zxy}{(1-x)(1-y) - zxy} \right) \\ &= \frac{1}{1-x} \left(\frac{(zy + 1 - y)(1-x)}{(1-x)(1-y) - zxy} \right) \\ &= \frac{1 - (1-z)y}{(1-x)(1-y) - zxy} \\ &= \frac{\frac{1-(1-z)y}{1-y}}{1 - x \frac{1-(1-z)y}{1-y}}. \end{aligned}$$

Call $f_z(y) = \frac{1-(1-z)y}{1-y}$. Next we replace x with xy so that we get that $|G(L, L', g)|$ is $[(xy)^{L'} y^{L-L'} z^g] \frac{f_z(y)}{1-(xy)f_z(y)} = [x^{L'} y^L z^g] \frac{f_z(y)}{1-xyf_z(y)}$. Thus, substituting $z = e^{-\Delta\mu}$ and calling $f = f_{e^{-\Delta\mu}}$,

$$Q_{L',L} = \sum_{g=0}^{\infty} e^{-g\Delta\mu} |G(L, L', g)| = [x^{L'} y^L] \sum_{g=0}^{\infty} e^{-g\Delta\mu} [z^g] \frac{f_z(y)}{1-xyf_z(y)} = [x^{L'} y^L] \frac{f(y)}{1-xyf(y)}.$$

Q is in the Bell subgroup of Riordan arrays (Shapiro et al. (1991) and chapter 3 of Shapiro et al. (2022)). This means that by Proposition 38, defining \bar{f} as the unique power series with $y\bar{f}(y)f(y\bar{f}(y)) = y$,

$$Q_{L',L}^{-1} = [x^{L'} y^L] \frac{\bar{f}(y)}{1-xy\bar{f}(y)}.$$

Part 2: Since

$$Q_{L',L}^{-1} = [x^{L'} y^L] \frac{\bar{f}(y)}{1-xy\bar{f}(y)} = [y^L] \bar{f}(y) (y\bar{f}(y))^{L'} = [y^{L-L'}] \bar{f}(y)^{L'+1},$$

we have,

$$\sum_{l=L}^{\infty} e^{-\zeta l} (Q_{L,l}^{-1})^2 = e^{-\zeta L} \sum_{l=0}^{\infty} (e^{-\frac{1}{2}\zeta l} [y^l] \bar{f}(y)^{L+1})^2.$$

This sum is finite if $e^{-\frac{1}{2}\zeta}$ is smaller than the radius of convergence of $\bar{f}(y)^{L+1}$ (as the terms in the series decay exponentially) and infinite if it is larger (as the terms do not approach 0). We will now show that the radius of convergence of \bar{f} is exactly 1, so that if $\zeta < 0$ then $\delta_0 \notin \mathcal{H}_k$ and if $\zeta > 0$ then, since the radius of convergence of $\bar{f}(y)^L$ is also at least 1 for all L , k has discrete masses. We will afterwards carefully consider the case of $\zeta = 0$.

Let us first consider the case $\Delta\mu = 0$. In this case we have, from the definition of f , $yf(y) = \frac{y}{1-y}$, so $y\bar{f}(y) = \frac{y}{1+y}$. In this case \bar{f} clearly has radius of convergence 1. We can also see explicitly that $[y^l] \bar{f}(y)^{L+1} = (-1)^l \binom{L+1}{l}$. Thus Equation 14 also does not hold for any L when $\zeta = 0$.

We now assume $\Delta\mu > 0$. We will derive the form of \bar{f} . Call $\xi = 1 - e^{-\Delta\mu} \in (0, 1)$, and $t = y\bar{f}(y)$.

$$tf(t) = y \iff t(1 - \xi t) = (1 - t)y \iff 0 = \xi t^2 - (y + 1)t + y.$$

Thus, picking the root that is 0 when $y = 0$, we get

$$y\bar{f}(y) = \frac{1}{2\xi} \left(1 + y - \sqrt{(1+y)^2 - 4\xi y^2} \right).$$

The roots of the polynomial $(1+y)^2 - 4\xi y^2$ are

$$(2\xi - 1) \pm \sqrt{(2\xi - 1)^2 - 1} = (2\xi - 1) \pm i\sqrt{1 - (2\xi - 1)^2}.$$

Note both of these roots have norm 1 and can thus be written $e^{i\theta}, e^{-i\theta}$ for some $\pi > \theta > 0$. Thus, we may write

$$y\bar{f}(y) = \frac{1}{2\xi} \left(1 + y - \sqrt{(1 - e^{i\theta}y)(1 - e^{-i\theta}y)} \right).$$

$\sqrt{1 - e^{i\theta}y}$, $\sqrt{1 - e^{-i\theta}y}$ each have radius of convergence 1, so, $y\bar{f}$ has radius of convergence at least 1. On the other hand, the derivative of $y\bar{f}$ diverges when y approaches $e^{i\theta}$ or $e^{-i\theta}$, so that $y\bar{f}$, and thus \bar{f} , has radius of convergence exactly 1.

The case when $\zeta = 0$ is more delicate. First call $g(y) = \sqrt{(1+y)^2 - 4\xi y^2} = \sqrt{(1 - e^{i\theta}y)(1 - e^{-i\theta}y)}$. For real $0 < r < 1$, define

$$\begin{aligned} q_L(r) &= \int_0^1 dt g(\sqrt{r}e^{2\pi it})^L g(\sqrt{r}e^{-2\pi it})^L \\ &= \int_0^1 dt \sum_n r^{n/2} e^{2\pi nit} [y^n]g(y)^L \sum_m r^{m/2} e^{-2\pi mit} [z^m]g(z)^L \\ &= \sum_{n,m} ([y^n]g(y)^L [z^m]g(z)^L) r^{(n+m)/2} \int_0^1 dt e^{2\pi it(n-m)} \\ &= \sum_n ([y^n]g(y)^L)^2 r^n. \end{aligned}$$

g is analytic with real coefficients at 0 so $([y^n]g(y)^L)^2 \geq 0$ for all L, n . Thus, by the monotone convergence theorem, $\sum_n ([y^n]g(y)^L)^2 = \lim_{r \rightarrow 1^-} q_L(r)$. However,

$$g(\sqrt{r}e^{2\pi iz})^L g(\sqrt{r}e^{-2\pi iz})^L = \left| 1 - e^{i(\theta+2\pi z)}\sqrt{r} \right|^L \left| 1 - e^{i(\theta-2\pi z)}\sqrt{r} \right|^L \leq 4^L,$$

so $\sum_n ([y^n]g(y)^L)^2 < \infty$ for all L . Now note

$$\begin{aligned} \sum_n ([y^n](y\bar{f}(y))^L)^2 &= \sum_n \frac{1}{(2\xi)^{2L}} \left(\sum_{l=0}^L [y^n](1+y)^l (-1)^{L-l} g(y)^{L-l} \right)^2 \\ &= \sum_n \frac{1}{(2\xi)^{2L}} \left(\sum_{l=0}^L (-1)^{L-l} \sum_{l'=0}^l \binom{l}{l'} [y^{n-l'}]g(y)^{L-l} \right)^2 \\ &\leq \sum_n \frac{1}{(2\xi)^{2L}} \left(L^2 \max_{l,l'} \left| \binom{l}{l'} [y^{n-l'}]g(y)^{L-l} \right| \right)^2 \\ &\leq \frac{1}{(2\xi)^{2L}} (L^2)^2 \sum_n \max_{l,l'} \left| \binom{l}{l'} \right|^2 \left([y^{n-l'}]g(y)^{L-l} \right)^2 \\ &\leq \frac{1}{(2\xi)^{2L}} (L^2)^2 \sum_{l=0}^L \sum_{l'=0}^l \binom{l}{l'}^2 \sum_n \left([y^{n-l'}]g(y)^{L-l} \right)^2 < \infty. \end{aligned}$$

Thus k has discrete masses when $\zeta = 0$ as well. ■

Appendix H. Discrete masses in the local alignment kernel

In this section we prove that the local alignment kernel has discrete masses under certain parameter settings.

Theorem 40 (*proof of Theorem 23*) If $\Delta\mu = 0$, k_{la} has discrete masses if and only if $2\mu > \log \sigma$. If $\infty > \Delta\mu > 0$, k_{la} has discrete masses if and only if $2\mu \geq \log \sigma$. If $\Delta\mu = \infty$, k_{la} has discrete masses.

Proof We will only provide a sketch of the proof highlighting the differences to the case of the regular sequence kernel. The details are very similar to those of the proofs of Theorems 21 and 39.

First we repeat the logic of Theorem 21 to reduce to the case of $|\mathcal{B}| = 1$. Let $U, W_V, V[t], \tilde{B}$ as in Theorem 21. First note if $V = \emptyset$, W_V is simply k_{la} with $k_s(U, U) = \sigma^{-1}$ on a space with $|\mathcal{B}| = 1$. Thus, if this local alignment kernel does not have discrete masses, neither does k_{la} on the entire space. Next consider $|V| \geq 1$ and define the left-local alignment kernel

$$k_l(X, Y) = \sum_{l=0}^{\infty} \tilde{k}_l \star (k_s \star k_l)^l$$

and the right-local alignment kernel k_r similarly. Now we repeat the logic of Eqn 10:

$$\begin{aligned} k(V[t], V[t']) &= \sum_{\substack{l \in \mathbb{N} \\ Y^{(1)} + \dots + Y^{(2l+1)} = t_0 \times U \\ Z^{(1)} + \dots + Z^{(2l+1)} = t'_0 \times U}} \sum_{\substack{l' \in \mathbb{N} \\ \tilde{Y}^{(1)} + \dots + \tilde{Y}^{(2l'+1)} = V[t_1:] \\ \tilde{Z}^{(1)} + \dots + \tilde{Z}^{(2l'+1)} = V[t'_1:]}} \left(\right. \\ &\quad \times \left(\tilde{k}_l(Y^{(1)}, Z^{(1)}) \prod_{i=1}^l k_s(Y^{(2i)}, Z^{(2i)}) k_l(Y(2i+1), Z^{(2i+1)}) \right) \\ &\quad \times k_s(V_{(0)}, V_{(0)}) \\ &\quad \times \left(\prod_{i=1}^{l'} k_l(\tilde{Y}^{(2i-1)}, \tilde{Z}^{(2i-1)}) k_s(\tilde{Y}^{(2i)}, \tilde{Z}^{(2i)}) \right) \times \tilde{k}_{l'}(\tilde{Y}^{(1)}, \tilde{Z}^{(1)}) \Big) \\ &= k_l(t_0 \times U, t'_0 \times U) k_r(V[t_1:], V[t'_1:]) \\ &= \dots \\ &= k_l(t_0 \times U, t'_0 \times U) \left(\prod_{l=1}^{L-1} k_{\text{ali}}(t_l \times U, t'_l \times U) \right) k_r(t_L \times U, t'_L \times U). \end{aligned} \tag{15}$$

Thus, k_{la} has discrete masses on W_V if k_{ali} , k_l , and k_r with $k_s(U, U) = \sigma^{-1}$ and $|\mathcal{B}| = 1$ have discrete masses. Thus, the theorem holds if we prove the case where $|\mathcal{B}| = 1$ for k_{ali} , k_{la} , k_l , and k_r .

We therefore study the case where $|\mathcal{B}| = 1$. We've already shown the theorem for the regular alignment kernel. We now consider k_l . Let $G(L, L', g)$ be the number of length L' sub-strings of a sequence of length L with g gaps **not including a gap on the left end**. In this case, each element of $G(L, L', g)$ is characterized by the length of its $g+1$ contiguous regions, each of which must be at least length 1 except for the region after the last gap; and $g+1$ gap regions, each of which must be at least length 1 except the first gap. Thus by the same argument as in Theorem 39, we get that $|G(L, L', g)|$ is the coefficient in front

of $x^{L'}y^{L-L'}z^g$ of the power series

$$\begin{aligned} \sum_{g=0}^{\infty} z^g \frac{x^g}{(1-x)^{g+1}} \frac{y^g}{(1-y)^{g+1}} &= \frac{1}{(1-x)(1-y)} \frac{(1-x)(1-y)}{(1-x)(1-y) - xyz} \\ &= \frac{\frac{1}{1-y}}{1 - xf_z(y)}. \end{aligned}$$

Thus in this case, defining the matrix Q as in Theorem 39,

$$Q_{L',L} = [x^{L'}y^L] \frac{\frac{1}{1-y}}{1 - xyf(y)}.$$

We can invert this matrix using Proposition 38. In particular, the numerator $\bar{g}(y)$ of the power series corresponding to the matrix inverse is

$$\left(\left(\frac{1}{1-y} \right) \circ (y\bar{f}(y)) \right)^{-1} = 1 - y\bar{f}(y).$$

Thus,

$$Q_{L',L}^{-1} = [x^{L'}y^L] \frac{1 - y\bar{f}(y)}{1 - xy\bar{f}(y)}.$$

In this case, k_l has discrete masses if and only if for all L ,

$$\sum_{l=0}^{\infty} e^{-\zeta l} (Q_{L,l}^{-1})^2 = \sum_{l=0}^{\infty} \left(e^{-\frac{1}{2}\zeta l} [y^l] (1 - y\bar{f}(y)) (y\bar{f}(y))^L \right)^2 < \infty.$$

The proof of the theorem is almost identical in the case that $\Delta\mu < \infty$. If $\Delta\mu = \infty$ then $f(y) = 1$ so $\bar{f}(y) = 1$ and $Q_{L',L}^{-1} = \delta_{L'}(L) - \delta_{L'+1}(L)$. Thus, only finitely many terms of the sum are non-zero, and Equation 14 is satisfied. The case is identical for k_r .

Finally we look at k_{la} . Let $G(L, L', g)$ be the number of length L' sub-strings of a sequence of length L with g gaps **not including gaps on either end**. In this case, when $g > 0$, each element of $G(L, L', g)$ is characterized by the length of its $g + 1$ contiguous regions, each of which must be at least length 1; and $g + 2$ gap regions, each of which must be at least length 1 except the first and last gap. When $g = 0$ we must be sure to include the case where $L' = 0$, even though $L' = 0$ is impossible when $g > 0$. Thus, $|G(L, L', g)|$ is the coefficient in front of $x^{L'}y^{L-L'}z^g$ of the power series

$$\begin{aligned} \sum_{g=0}^{\infty} z^g \frac{x^{g+1}}{(1-x)^{g+1}} \frac{y^g}{(1-y)^{g+2}} + \frac{1}{1-y} &= \frac{x}{(1-x)(1-y)^2} \frac{(1-x)(1-y)}{(1-x)(1-y) - xyz} + \frac{1}{1-y} \\ &= x \frac{\left(\frac{1}{1-y} \right)^2}{1 - xf_z(y)} + \frac{1}{1-y}. \end{aligned}$$

So in this case,

$$Q_{L',L} = [x^{L'}y^L] xy \frac{\left(\frac{1}{1-y} \right)^2}{1 - xyf(y)} + \frac{1}{1-y}.$$

The first row of Q is just 1's and

$$(Q_{1:,1:})_{L',L} = [x^{L'} y^L] \frac{\left(\frac{1}{1-y}\right)^2}{1 - xyf(y)}.$$

$Q_{1:,1:}$ is a Riordan array with inverse

$$(Q_{1:,1:})_{L',L}^{-1} = [x^{L'} y^L] \frac{(1 - y\bar{f}(y))^2}{1 - xy\bar{f}(y)}.$$

Let R be the upper triangular metrix with $R_{1:,1:} = (Q_{1:,1:})^{-1}$, $R_{0,0} = 1$ and $R_{0,1:} = -\mathbf{1}^T (Q_{1:,1:})^{-1}$ where $\mathbf{1}$ is the infinite vector of ones. We will see that $R = Q^{-1}$. First, for $L, L' \geq 1$,

$$Q_{L,\cdot} R_{\cdot,L'} = \delta_L(L').$$

Next for $L \geq 0$,

$$Q_{L,\cdot} R_{\cdot,0} = \delta_L(0).$$

Finally, for $L' \geq 1$,

$$Q_{0,\cdot} R_{\cdot,L'} = -(\mathbf{1}^T (Q_{1:,1:})^{-1})_{L'-1} + \mathbf{1}^T (Q_{1:,1:})_{\cdot,L'-1}^{-1} = 0.$$

Thus $R = Q^{-1}$.

Since $\mathbf{1}_L = 1 = [x^L](1 - x)^{-1}$,

$$\begin{aligned} (\mathbf{1}^T (Q_{1:,1:})^{-1})_L &= \sum_{L'} (Q_{1:,1:})_{L',L}^{-1} [x^{L'}](1 - x)^{-1} \\ &= \sum_{L'} [y^{L'}] \left((1 - y\bar{f}(y))^2 (y\bar{f}(y))^{L'} \right) [x^{L'}](1 - x)^{-1} \\ &= [y^L] (1 - y\bar{f}(y))^2 \sum_{L'} (y\bar{f}(y))^{L'} [x^{L'}](1 - x)^{-1} \\ &= [y^L] (1 - y\bar{f}(y))^2 (1 - y\bar{f}(y))^{-1} \\ &= [y^L] (1 - y\bar{f}(y)) \end{aligned}$$

Thus,

$$Q_{L',L}^{-1} = [x^{L'} y^L] xy \frac{(1 - y\bar{f}(y))^2}{1 - xy\bar{f}(y)} + 1 - y(1 - y\bar{f}(y)).$$

The rest of the proof is again almost identical to the previous case when $\Delta\mu < \infty$. When $\Delta\mu = \infty$, $Q_{0,L}^{-1} = \delta_0(L) - \delta_1(L) + \delta_2(L)$ and

$$(Q_{1:,1:})_{L',L}^{-1} = [x^{L'} y^L] \frac{(1 - y)^2}{1 - xy} = \delta_{L'}(L) - 2\delta_{L'+1}(L) + \delta_{L'+2}(L).$$

Thus k_{la} again has discrete masses by Equation 14. ■

Appendix I. Computation of thick tailed alignment kernels

We now describe dynamic programming procedures that can be used to calculate the thick tailed alignment kernels described in Examples 9 and 10. We will deal with both examples as special cases of a single general dynamic programming algorithm. Let $\ell(b, b')$, taking values in $\{1, 0\}$, be a function for counting occurrences of a certain type of letter pairs. Call pairs (b, b') “ ℓ -type” if $\ell(b, b') = 1$.

Define

$$M(i, j, l) = \sum_{i=0}^{(l)} \prod_{i=0}^{c-1} k_I(X_{(:i)}^{(2i)}, Y_{(:j)}^{(2i)}) k_s(X_{(:i)}^{(2i+1)}, Y_{(:j)}^{(2i+1)})$$

Where the sum is over c and over partitions $X_{(:i)}^{(0)} + \dots + X_{(:i)}^{(2c+1)} = X_{(:i)}$, $Y_{(:j)}^{(0)} + \dots + Y_{(:j)}^{(2c+1)} = Y_{(:j)}$ for which $l = \sum_{i=1}^c \ell(X_{(:i)}^{(2i+1)}, Y_{(:j)}^{(2i+1)})$. Thus, $M(i, j, l)$ sums over all alignments between $X_{(:i)}$ and $Y_{(:j)}$ with l comparisons that are ℓ -type and which end in a match between $X_{(i-1)}$ and $Y_{(j-1)}$. Let $I_X(i, j, l)$ be defined similarly as the sum of alignment scores of $X_{(:i)}$ and $Y_{(:j)}$ that have l comparisons that are ℓ -type and end in an insertion for X but not Y . Finally let $I_Y(i, j, l)$ be defined similarly as the sum of alignment scores of $X_{(:i)}$ and $Y_{(:j)}$ that have l comparisons that are ℓ -type and end in an insertion for Y . Call the sum over all alignments of $X_{(:i)}$ and $Y_{(:j)}$ with l comparisons that are ℓ -type,

$$J(i, j, l) = M(i, j, l) + I_X(i, j, l) + I_Y(i, j, l).$$

We can now compute M , I_X and I_Y with dynamic programming. We initialize at,

$$\begin{aligned} M(i, 0, l) &= M(0, j, l) = 0 \text{ for } l, i, j \\ M(0, 0, 0) &= 1 \\ I_X(0, j, l) &= I_Y(i, 0, l) = 0 \text{ for } l, i, j. \end{aligned}$$

The update equations are,

$$\begin{aligned} M(i, j, l) &= \mathbb{1}(l > 0) \ell(X_{(i-1)}, Y_{(j-1)}) k_s(X_{(i-1)}, Y_{(j-1)}) J(i-1, j-1, l-1) \\ &\quad + (1 - \ell(X_{(i-1)}, Y_{(j-1)})) k_s(X_{(i-1)}, Y_{(j-1)}) J(i-1, j-1, l) \\ I_X(i, j, l) &= e^{-\Delta\mu - \mu} M(i-1, j, l) + e^{-\mu} I_X(i-1, j, l) \\ I_Y(i, j, l) &= e^{-\Delta\mu - \mu} M(i, j-1, l) + e^{-\Delta\mu - \mu} I_X(i, j-1, l) + e^{-\mu} I_Y(i-1, j, l). \end{aligned}$$

(We use the arbitrary convention that, between any two adjacent match positions, insertions in Y come before insertions in X ; thus I_Y does not appear in the update equation for I_X .) We can then define $R(L) = J(|X|, |Y|, L)$ as the sum over all alignments of X and Y with L comparisons that are ℓ -type. The dynamic programming algorithm calculates $R(L)$ with $O(|X||Y|(|X| \wedge |Y|))$ operations.

Now consider the choice $k_s(b, b') = 1$ and $\ell(b, b') = \mathbb{1}(b \neq b')$. In this case, $R(L)$ sums over all alignments where L of the matched positions have mismatched letters, i.e. where the Hamming distance between the matched positions is L . Thus, the kernel described in Example 9 can be computed as

$$\tilde{k}(X, Y) = \sum_{L=0}^{|X| \wedge |Y|} (C + L)^{-\beta} R(L).$$

Next set $k_s(b, b') = \delta_b(b')$ and $\ell(b, b') = 1$, so $R(L)$ sums over all alignments where there are L matched positions and all have matched letters. Recall the definition of \tilde{u}_V as in Example 10. For sequences V, X, Y , $\tilde{u}_V(X)\tilde{u}_V(Y)$ is the sum over all alignments of X and Y such that the matched positions are V in both X and Y , so, $R(L) = \sum_{|V|=L} \tilde{u}_V(X)\tilde{u}_V(Y)$. Thus, the kernel described in Example 10 can be computed as

$$\tilde{k}(X, Y) = \sum_{L=0}^{|X| \wedge |Y|} \left(C + \frac{1}{2}(|X| + |Y|) - L \right)^{-\beta} R(L).$$

Appendix J. Flexibility of embedding kernels

Proposition 41 (*proof of Propositions 27 and 29*) Assume $k(z, z') = \Psi(z - z')$ is a translation invariant kernel and that Ψ is a positive continuous function on \mathbb{R}^D that has a strictly positive Fourier transform. Then the embedding kernel k is characteristic and C_0 -universal if and only if F is injective. Furthermore, k metrizes $\mathcal{P}(S)$ if and only if k has discrete masses, which occurs if and only if $F(S)$ has no accumulation points, that is, there is no sequence $X \in S$ such that $F(X)$ is in the closure of $F(S \setminus \{X\})$.

Proof First we prove that k metrizes $\mathcal{P}(S)$ only if F is injective and $F(S)$ has no accumulation points. Say $F(X_n) \rightarrow F(X)$ for a sequence of points with $X_n \neq X$ for all n . $k(X_n, X_n) = \Psi(0) = k(X, X)$ and $k(X_n, Y) = \Psi(F(X_n) - F(Y)) \rightarrow \Psi(F(X) - F(Y)) = k(X, Y)$ for all Y . Thus

$$\left\| \int d\delta_{X_n}(Y)k_Y - \int d\delta_X(Y)k_Y \right\|_k^2 = \|k_{X_n} - k_X\|_k^2 = 2k(X, X) - 2k(X_n, X) \rightarrow 0$$

So k does not metrize $\mathcal{P}(S)$ as $\delta_{X_n} \not\rightarrow \delta_X$.

Now we show k has discrete masses if F is injective and $F(S)$ has no accumulation points. Say k does not have discrete masses, so that, by Proposition 9, $\sum_{m=1}^{M_n} \alpha_{n,m} k_{X_{n,m}} \rightarrow k_X$ in \mathcal{H}_k for $X_{n,m} \neq X$. Call $\mu_n = \sum_{m=1}^{M_n} \alpha_{n,m} \delta_{F(X_{n,m})} - \delta_{F(X)}$. Defining $\hat{\Psi}$ and $\hat{\mu}_n$ as the Fourier transforms of Ψ and μ_n , we have, as in Sriperumbudur et al. (2011), that

$$\int d\xi |\hat{\mu}_n(\xi)|^2 \hat{\Psi}(\xi) = \int \int d\mu_n(x) d\mu_n(y) \Psi(x - y) = \left\| \int d\mu_n(X) k_X \right\|_k^2 \rightarrow 0.$$

Thus there is a subsequence $(\hat{\mu}_{n_k})_k$ such that $\hat{\mu}_{n_k} \rightarrow 0$ pointwise almost everywhere with respect to the measure $\hat{\Psi}(\xi)d\xi$. Since $\hat{\Psi}$ is strictly positive, $\hat{\mu}_{n_k} \rightarrow 0$ pointwise almost everywhere with respect to the Lebesgue measure. This implies that $\mu_{n_k} \rightarrow 0$ in distribution. This can happen only if there is a sequence $F(X_{n_k, m_{n_k}}) \rightarrow F(X)$, i.e. $F(S)$ has an accumulation point.

Finally we show k is characteristic and universal if and only if F is injective. First, k is not characteristic if F is not injective, since this implies there are two sequences with the same kernel embedding. Now say F is injective and ν is a finite signed measure with $\int k_X d\nu(X) = 0$. Define $\mu(C) = \nu(F^{-1}(C))$ for all sets $C \subset \mathbb{R}^D$. Then by the logic above, $\int d\xi |\hat{\mu}(\xi)|^2 \hat{\Psi}(\xi) = \left\| \int k_X d\nu(X) \right\|_k^2 = 0$. Thus, $\hat{\mu} = 0$ so $\mu = 0$. Since F is injective, this implies that $\nu = 0$. Thus the map $\nu \mapsto \int d\nu(X)k_X$ is injective for all signed measures. This

is equivalent to C_0 -universality by Proposition 2 of Sriperumbudur et al. (2011). This also means that the mapping is injective for all probability distributions, so k is characteristic. ■

Appendix K. Sparse representation learning can yield inflexible embedding kernels

Consider learning a sequence representation using a matrix factorization model with sparsity-encouraging regularization. For example, consider the following objective,

$$\min_{W, B, Z} \sum_{i=1}^N \|X_i - \sum_d W_d Z_{id} - B\|_2^2 + \lambda \sum_{d=1}^D \sum_{l=0}^{\infty} \sum_{b \in \mathcal{B}} |W_{dlb}| + \beta \sum_{i=1}^D \|Z_i\|_2^2. \quad (16)$$

Here, as in the position-wise comparison kernel (Section 6), we add an infinite tail of stop symbols to each sequence $X \in S$. We treat sequences as one-hot encodings. The vectors $Z_i \in \mathbb{R}^D$ are the representations of the training data, and are mapped linearly, with slope W and offset B , to one-hot encoded sequence space. There is ℓ^2 regularization on the embedding Z , with weight β , and ℓ^1 regularization on W , with weight λ . This lasso regularization on W encourages each of its entries to be zero, and can make representations more interpretable, in the sense that each dimension of Z only affects a small number of sequence positions. Once the model has been trained, we can use it to define an embedding function $F(X) = \operatorname{argmin}_{Z \in \mathbb{R}^D} \|X - \sum_d \hat{W}_d Z_d - \hat{B}\|_2^2 + \beta \|Z\|_2^2$, where \hat{W} and \hat{B} are the parameter values estimated from the training data.

On a finite training data set, there are likely to be positions l in \hat{W} with more than one zero, i.e. for $b \neq b'$ and all $d \in \{1, \dots, D\}$ we have $W_{dlb} = W_{dlb'} = 0$. In this case, a sequence \tilde{X} with $\tilde{X}_{lb} = 1$ will map to the same representation as another sequence \tilde{X}' that is identical to \tilde{X} except with a different letter at position l , namely $\tilde{X}'_{lb'} = 1$. In other words, $F(\tilde{X}) = F(\tilde{X}')$ for $\tilde{X}' \neq \tilde{X}$, and so F is not injective, and an embedding kernel based on F will be neither universal nor characteristic.

Appendix L. Scaled random embeddings

Proposition 42 (*proof of Proposition 31*) *Consider an initial embedding \tilde{F} where each $\tilde{F}(X)$ for $X \in S$ is drawn from the uniform distribution on the sphere, $\{x \in \mathbb{R}^D \mid \|x\| \leq 1\}$. Then, a kernel using the scaled embedding $F(X) = |\mathcal{B}|^{(1+\epsilon)|X|/D} \tilde{F}(X)$, for $\epsilon > 0$, metrizes $\mathcal{P}(S)$ almost surely.*

Proof The probability that F is injective is 1, following the same logic as in the proof of Proposition 28. Now, for any N we will show that with probability 1, $\{X \mid \|F(X)\| \leq N\}$ is finite. This will imply that for any sequence of distinct sequences X_1, X_2, \dots , we must have $\|F(X_n)\|_{\infty} \rightarrow \infty$ as $n \rightarrow \infty$, and in particular the embedding $F(S)$ has no accumulation points. Now,

$$p(\|F(X)\| \leq N) = p(\|\tilde{F}(X)\| \leq N/|\mathcal{B}|^{(1+\epsilon)|X|/D}).$$

If $|\mathcal{B}|^{-(1+\epsilon)|X|/D}N \geq 1$ then this quantity is 1. Otherwise, calling V the Lebesgue measure and $B(1)$ the unit ball $\{x \in \mathbb{R}^D \mid \|x\| \leq 1\}$, then

$$p(\|F(X)\| \leq N) = \frac{V(|\mathcal{B}|^{-(1+\epsilon)|X|/D}NB(1))}{V(B(1))} = |\mathcal{B}|^{-(1+\epsilon)|X|}N^D.$$

Thus,

$$\sum_{X \in \mathcal{S}} p(\|F(X)\| \leq N) \leq N^D \sum_{X \in \mathcal{S}} |\mathcal{B}|^{-(1+\epsilon)|X|} = N^D \sum_{L=0}^{\infty} |\mathcal{B}|^{-L\epsilon} < \infty.$$

So by the Borel-Cantelli lemma, with probability 1, $F(S)$ has no accumulation points. \blacksquare

References

- 10x Genomics. A new way of exploring immunity: linking highly multiplexed antigen recognition to immune repertoire and phenotype. *10x Genomics*, 2022.
- Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, 33(8):831–838, 2015.
- Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, 16(12):1315–1322, December 2019.
- Alan N Amin, Eli N Weinstein, and Debora S Marks. A generative nonparametric bayesian model for whole genomes. *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, December 2021.
- Miguel A Arcones and Evarist Gine. On the bootstrap of U and V statistics. *Ann. Stat.*, 20(2):655–674, 1992.
- Balandat, Karrer, Jiang, Daulton, Letham, Wilson, and Bakshy. BoTorch: A framework for efficient Monte-Carlo bayesian optimization. *Adv. Neural Inf. Process. Syst.*, 2020.
- Luis A Barrera, Anastasia Vedenko, Jesse V Kurland, Julia M Rogers, Stephen S Gisselbrecht, Elizabeth J Rossin, Jaie Woodard, Luca Mariani, Kian Hong Kock, Sachi Inukai, Trevor Siggers, Leila Shokri, Raluca Gordân, Nidhi Sahni, Chris Cotsapas, Tong Hao, Song Yi, Manolis Kellis, Mark J Daly, Marc Vidal, David E Hill, and Martha L Bulky. Survey of variation in human transcription factors reveals prevalent DNA binding changes. *Science*, 351(6280):1450–1454, March 2016.
- Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-N protein engineering with data-efficient deep learning. *Nat. Methods*, 18(4):389–396, 2021.
- Dexiong Chen, Laurent Jacob, and Julien Mairal. Recurrent kernel networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 13453–13464. Curran Associates Inc., Red Hook, NY, USA, December 2019.

- Andreas Christmann and Ingo Steinwart. Universal kernels on Non-Standard input spaces. In J Lafferty, C Williams, J Shawe-Taylor, R Zemel, and A Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.
- Bo Dai, Hanjun Dai, Arthur Gretton, Le Song, Dale Schuurmans, and Niao He. Kernel exponential family estimation via doubly dual embedding. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2321–2330. PMLR, 2019.
- Charita Dellaporta, Jeremias Knoblauch, Theodoros Damoulas, and François-Xavier Briol. Robust bayesian inference for simulator-based models via the MMD posterior bootstrap. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, February 2022.
- Nicki Skafte Detlefsen, Søren Hauberg, and Wouter Boomsma. Learning meaningful representations of protein sequences. *Nat. Commun.*, 13(1):1914, April 2022.
- Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, April 1998.
- Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K Min, Kelly Brock, Yarin Gal, and Debora S Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.
- Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Neural Information Processing Systems (NeurIPS)*, 2007.
- Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Bharath K Sriperumbudur. Characteristic kernels on groups and semigroups. *Adv. Neural Inf. Process. Syst.*, 21, 2008.
- Futoshi Futami, Zhenghang Cui, Issei Sato, and Masashi Sugiyama. Bayesian posterior approximation via greedy particle optimization. *AAAI*, 33(01):3606–3613, July 2019.
- Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pages 129–143. Springer Berlin Heidelberg, 2003.
- Jackson Gorham and Lester Mackey. Measuring sample quality with kernels. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1292–1301. PMLR, 2017.
- Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Adv. Neural Inf. Process. Syst.*, 20, 2007.

- Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012.
- Ben Hambly and Terry Lyons. Uniqueness for the signature of a path of bounded variation and the reduced path group. *Ann. Math.*, 171(1):109–167, March 2010.
- David Haussler. Convolution kernels on discrete structures ucsc crl. 1999.
- Ferenc Huszár and David Duvenaud. Optimally-weighted herding is bayesian quadrature. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI’12, pages 377–386, Arlington, Virginia, USA, August 2012. AUAI Press.
- T Jaakkola, M Diekhans, and D Haussler. A discriminative framework for detecting remote protein homologies. *J. Comput. Biol.*, 7(1-2):95–114, February 2000.
- Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, pages 8580–8589, Red Hook, NY, USA, December 2018. Curran Associates Inc.
- P Jorgensen and F Tian. Discrete reproducing kernel hilbert spaces: sampling and distribution of dirac-masses. *J. Mach. Learn. Res.*, 2015.
- Franz J Kiraly and Harald Oberhauser. Kernels for sequentially ordered data. *J. Mach. Learn. Res.*, 20(31):1–45, 2019.
- Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, January 2020.
- Tao Lei, Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Deriving neural architectures from sequence and graph kernels. In *International Conference on Machine Learning*, pages 2024–2033. PMLR, July 2017.
- Christina Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: a string kernel for SVM protein classification. *Pac. Symp. Biocomput.*, pages 564–575, 2002.
- Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: towards deeper understanding of moment matching network. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, pages 2200–2210, Red Hook, NY, USA, December 2017. Curran Associates Inc.
- Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.
- Takuo Matsubara, Jeremias Knoblauch, François-Xavier Briol, and Chris J Oates. Robust generalised bayesian inference for intractable likelihoods. *J. R. Stat. Soc. Series B Stat. Methodol.*, 84(3):997–1022, July 2022.
- Radford M Neal. Priors for infinite networks. In Radford M Neal, editor, *Bayesian Learning for Neural Networks*, pages 29–53. Springer New York, New York, NY, 1996.

- Pascal Notin, José Miguel Hernández-Lobato, and Y Gal. Improving black-box optimization in VAE latent space using decoder uncertainty. *Adv. Neural Inf. Process. Syst.*, 2021.
- Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. Machine learning for functional protein design. *Nat. Biotechnol.*, 42(2):216–228, February 2024.
- Mijung Park, Wittawat Jitkrittum, and Dino Sejdinovic. K2-ABC: Approximate bayesian computation with kernel embeddings. In Arthur Gretton and Christian C Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 398–407, Cadiz, Spain, 2016. PMLR.
- Roman Pogodin, Namrata Deka, Yazhe Li, Danica J Sutherland, Victor Veitch, and Arthur Gretton. Efficient conditionally invariant representation learning. In *International Conference on Learning Representations (ICLR)*, September 2022.
- Luc Pronzato and Anatoly Zhigljavsky. Bayesian quadrature, energy minimization, and Space-Filling design. *SIAM/ASA J. Uncertainty Quantification*, 8(3):959–1011, January 2020.
- Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems*, volume 32, pages 9689–9701, 2019.
- G Rättsch, S Sonnenburg, and B Schölkopf. RASE: recognition of alternatively spliced exons in c.elegans. *Bioinformatics*, 21 Suppl 1:i369–77, June 2005.
- A M Rush. Torch-struct: Deep structured prediction library. *arXiv preprint arXiv:2002.00876*, 2020.
- Hiroto Saigo, Jean-Philippe Vert, Nobuhisa Ueda, and Tatsuya Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, July 2004.
- Gabriele Schweikert, Gunnar Rättsch, Christian Widmer, and Bernhard Schölkopf. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. *Adv. Neural Inf. Process. Syst.*, 21, 2008.
- Bernhard Schölkopf and Alexander J Smola. 13 designing kernels. In *Learning with Kernels*. The MIT Press, Massachusetts Institute of Technology, December 2021.
- Louis Shapiro, Renzo Sprugnoli, Paul Barry, Gi-Sang Cheon, Tian-Xiao He, Donatella Merlini, and Weiping Wang. *The Riordan Group and Applications*. Springer International Publishing, 2022.
- Louis W Shapiro, Seyoum Getu, Wen-Jin Woan, and Leon C Woodson. The riordan group. *Discrete Appl. Math.*, 34(1-3):229–239, November 1991.
- Rahul Singh, Liyuan Xu, and Arthur Gretton. Kernel methods for multistage causal inference: Mediation analysis and dynamic treatment effects. November 2021.

- Rahul Singh, Liyuan Xu, and Arthur Gretton. Kernel methods for causal functions: Dose, heterogeneous, and incremental response curves. *Biometrika*, 111(2):497–516, 2023.
- Sören Sonnenburg, Gunnar Rätsch, and Konrad Rieck. Large-scale learning with string kernels. In *Large-Scale Kernel Machines*. The MIT Press, 2007.
- Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert R G Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for RKHS embeddings of probability distributions. *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, pages 1750–1758, 2009.
- Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R G Lanckriet. Hilbert space embeddings and metrics on probability measures. *J. Mach. Learn. Res.*, 11(50):1517–1561, 2010.
- Bharath K Sriperumbudur, Kenji Fukumizu, and Gert R G Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *J. Mach. Learn. Res.*, 12(70):2389–2410, 2011.
- Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20459–20478. PMLR, 2022.
- Nora C Toussaint, Christian Widmer, Oliver Kohlbacher, and Gunnar Rätsch. Exploiting physico-chemical properties in string kernels. *BMC Bioinformatics*, 11 Suppl 8:S7, October 2010.
- Koji Tsuda, Taishin Kin, and Kiyoshi Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18 Suppl 1:S268–75, 2002.
- Jean-Philippe Vert, Hiroto Saigo, and Tatsuya Akutsu. Local alignment kernels for biological sequences. In *Kernel Methods in Computational Biology*, pages 131–153. MIT Press, 2004.
- Chris Watkins. Dynamic alignment kernels. In *Advances in Large-Margin Classifiers*, pages 39–50. The MIT Press, September 2000.
- Eli N Weinstein. *Generative Statistical Methods for Biological Sequences*. PhD thesis, Harvard University, Ann Arbor, United States, 2022.
- Eli N Weinstein and Debora S Marks. A structured observation distribution for generative biological sequence prediction and forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, 139, pages 11068–11079. PMLR, 2021.
- Eli N Weinstein and Jeffrey W Miller. Bayesian data selection. *J. Mach. Learn. Res.*, 24(23):1–72, 2023.

- Jason Weston, Bernhard Schölkopf, Eleazar Eskin, Christina Leslie, and William Stafford Noble. Dealing with large diagonals in kernel matrices. *Ann. Inst. Stat. Math.*, 55(2): 391–408, June 2003.
- Christopher Williams. Computing with infinite networks. *Adv. Neural Inf. Process. Syst.*, 9, 1996.
- Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In Arthur Gretton and Christian C Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain, 2016. PMLR.
- Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, August 2018.
- Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods*, 16(8):687–694, August 2019.