UncertainGen: Scalable Uncertainty-Aware Representation Learning of DNA Sequences

Abdulkadir Çelikkanat[1] **Andres R. Masegosa**[1] **Mads Albertsen**[2] **Thomas D. Nielsen**[1]
Aalborg University
9000 Aalborg, Denmark
[1]{abce,arma,tdn}@cs.aau.dk, [2]ma@bio.aau.dk

Abstract

Learning effective representations of DNA sequences is fundamental to genomic data analysis, especially in the presence of high sequence similarity and inter-species DNA sharing. Existing approaches rely on deterministic embeddings, such as k-mer statistics or representations from large language models, which fail to capture the inherent uncertainty in short and ambiguous DNA fragments. We propose UNCERTAINGEN, a lightweight probabilistic representation learning framework that embeds DNA sequences as distributions in a latent space. The learned embedding variance captures positional ambiguity, increasing for sequences compatible with multiple genomic clusters. We establish theoretical guarantees on embedding distinguishability, and show that the probabilistic formulation induces a data-adaptive metric that expands the effective representational space. We evaluate UNCERTAINGEN in the metagenomic binning task to demonstrate the practical benefits of uncertainty-aware representations. Experiments show that probabilistic embeddings consistently outperform deterministic k-mer and large genome foundation models, while remaining lightweight and efficient.

1 Introduction

Genomic sequences encode the blueprint of life, and analyzing them is fundamental for understanding biological processes, evolutionary relationships, and microbial ecosystems [7, 25, 4]. In recent years, advances in high-throughput DNA sequencing technologies have enabled large-scale studies of complex microbial communities directly from environmental samples. However, these technologies typically produce fragmented DNA sequences (called *reads*) rather than complete genomes. This fragmentation poses a significant challenge: recovering the full DNA sequences of the microbes in a sample requires assembling these reads and organizing them according to their origin.

Organizing reads from a mixed microbial sample is known as *metagenomic binning*, which aims to group DNA fragments so that each group corresponds to a distinct genome [14]. Accurate binning is critical for downstream analyses, such as functional annotation, phylogenetic profiling, and strain-level variation studies [24, 16]. At its core, metagenomic binning relies on a representation of DNA fragments preserving genomic similarity and inter-species dissimilarity, enabling meaningful comparisons between reads or assembled contiguous sequences (i.e. contigs). This makes metagenomic binning a natural benchmark for evaluating DNA sequence representations, as it directly tests whether embeddings preserve genomic structure while offering well-defined ground truth.

Traditionally, DNA sequences are represented using k*-mer profiles*, where sequences are decomposed into substrings of length k to construct the feature vectors of DNA fragments (see Figure 1 (a-c)). Numerous methods leverage these k-mer–based representations to learn latent representations (i.e., *embeddings*) and later cluster the DNA fragments for metagenomic binning [23, 6, 21, 5, 9]. Recent studies employ Large Language Models (LLM) that operate directly on raw sequences by avoiding

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

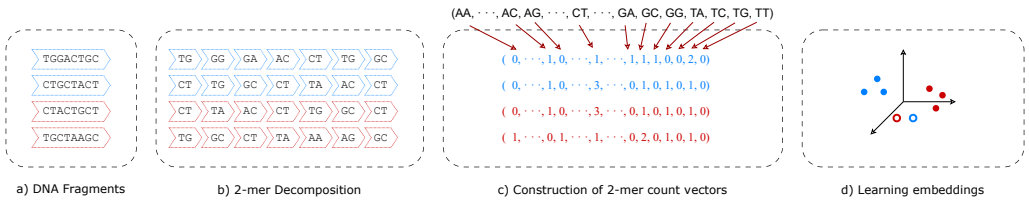


Figure 1: **Illustration of the metagenomic binning process**. Starting from a set of DNA sequences (a), the process ends with their two-dimensional embeddings derived from 2-mer profiles (d). In general, these embeddings allow the DNA fragments from two different species to be correctly clustered. However, the second and third DNA sequences in (a) pose an exception: although distinct, their k-mer representations shown in (c) are highly similar and, consequently, their embeddings are also very close (shown as the two empty circled points in (d)). Because the k-mer profiles of DNA within a species tend to be (locally) similar, the contrastive learning procedure attempts to position such fragments in both clusters, but, since this is not possible, ultimately places them between them.

explicit k-mer feature vectors to generate embeddings with the aim of capturing richer contextual information [18, 31, 32]. However, recent works also indicate that k-mer–based embeddings achieve comparable performance while offering orders-of-magnitude greater computational efficiency than large genome foundation models [5].

A shared characteristic of these methods is that they produce *deterministic embeddings*, mapping each DNA fragment to a single fixed point in the embedding space that is subsequently clustered and assigned to a single group representing the reconstructed species' DNA (Figure 1 illustrates this process). However, many DNA sequences can appear in multiple genomes, for instance, through horizontal (lateral) gene transfer [2] and should ideally be assigned to their correct clusters. But this is impossible for any clustering algorithm because such sequences will be represented by the same point in the embedding space. A further limitation arises with k-mer–based representations: distinct sequences, potentially belonging to different clusters, can exhibit highly similar k-mer profiles and are therefore projected to similar embedding vectors, making it very difficult for the clustering algorithm to assign them accurately. Figure 1 visually illustrates this point.

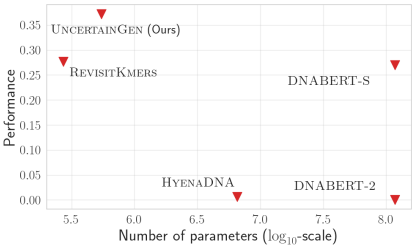


Figure 2: **Performance versus model size**. UncertainGen (our approach) achieves a favorable trade-off between model size (log-scale) and the performance in metagenomic binning tasks compared with existing lightweight and large foundation models.

In this work, we provide a mathematical formalization of the above issues and show that *deterministic embeddings* cannot resolve them. We therefore propose the use of *probabilistic embeddings* [22, 29, 11], where each fragment is mapped not to a single point but to a distribution (i.e., a region) in the embedding space. These distributions encode two sources of ambiguity: i) fragments may belong to multiple clusters (e.g., appear in the genomes of different species), and ii) identical k-mer profiles may arise from different species due to indistinguishable features.

Previous work on probabilistic embeddings in computer vision, NLP, and graph representation learning have typically relied on heuristically selected distributional distances, such as the Kullback–Leibler or Wasserstein divergence, to compare objects [17]. In contrast, our framework employs a contrastive-learning formulation in which non-Euclidean distances between embeddings emerge naturally from a probability distribution defined over the embedding space. This formulation yields closed-form expressions for the expected pairwise likelihood, enabling efficient and scalable optimization. Moreover, we present a theoretical analysis identifying the types of sequences producing large covariance terms, thereby offering insight into how and why the model captures uncertainty arising from ambiguous or multi-class sequences.

To the best of our knowledge, we propose the first framework for probabilistic embeddings of DNA sequences that extends the k-mer-based representation approaches [5, 21] to explicitly model ambiguity arising from DNA fragments shared across multiple species. We propose a lightweight framework for embedding DNA sequences, UncertainGen, that consistently outperforms existing

state-of-the-art approaches, spanning both lightweight methods and large foundation models as shown in Figure 2, offering both theoretical insights and practical performance gains. Our focus is on learning uncertainty-aware representations rather than developing a complete binning pipeline; we evaluate embedding quality through metagenomic binning benchmarks, leaving integration with coverage information and other pipeline-specific components to future work.

- We introduce a novel light-weight probabilistic embedding approach that models the uncertainty inherent in DNA sequences arising from inter-species DNA sharing and from fragments with highly similar representations.
- We derive theoretical guarantees on the distinguishability of both deterministic and probabilistic embeddings, showing how the probabilistic embeddings expand the feasible latent space and improve the model's capacity to separate DNA fragments.
- We empirically demonstrate the effectiveness of our approach on six metagenomic binning datasets, using clustering performance as a proxy to evaluate representation quality. Our method shows consistent improvements over deterministic k-mer and LLM-based embeddings.

The implementation of the proposed approach and further details are provided in the project repository.

2 Related Work

Embedding models for DNA sequences. Representations of DNA sequences have advanced rapidly in recent years. Classical approaches are based on k-mer profiles (frequency vectors of length-k substrings) [30, 10], and many practical genome analysis tools continue to rely on such features [19, 28, 15]. More recently, the field has seen a surge of genome foundation models that adapt transformers or other long-context architectures to genomic data. DNABERT [9] introduced a BERT-style encoder with k-mer tokenization, while DNABERT-2 [31] replaced fixed k-mers with byte-pair encoding (BPE) to improve efficiency and downstream performance. DNABERT-S [32] further refined this line of work by introducing a curriculum contrastive learning strategy with manifold instance mixup loss to address the metagenomic binning task. HYENADNA [18] extended the context window further by modeling single-nucleotide tokens with a long-range convolutional architecture, reducing the cost of dense attention. Other approaches explore alternative geometries, such as HCNN [12], which learns sequence representations in hyperbolic space.

In parallel, lightweight but task-specific models have been developed for metagenomics. SEMIBIN2 [21] and related methods [28] employ self-supervision and contrastive objectives tailored to binning, producing embeddings that cluster effectively by genome of origin. Recent work [5] has revisited the foundations of k-mer features, showing both their scalability and the limits of when k-mer profiles alone can separate genomes in practice. These results highlight a central trade-off: while foundation models offer expressive, context-aware representations, lightweight contrastive or k-mer-based approaches can rival or even outperform them in realistic binning scenarios.

Probabilistic embeddings for contrastive learning. Probabilistic embeddings have previously been explored, both generally [29, 11, 3] and within a task specific context [27, 22]. For example, in the context of face embeddings, each image is represented as a multivariate Gaussian distribution in embedding space, where a mutual likelihood score (MLS) is used to capture the likelihood of pairs of images belonging to the same person [22]. Property 1 in that work indicates that the proposed MLS score with fixed variance terms corresponds to a scaled and shifted negative squared Euclidean distance. Our results (Corollary 3.3.1) extend Property 1 given in this paper [22] by characterizing the (limited) expressivity of the equivalent of a fixed variance MLS score, while at the same time showing that modeling capacity can be increased by allowing for varying covariance terms.

Probabilistic embeddings have also been explored in a variational context. For instance, the work [20] learns probabilistic embeddings using a soft contrastive loss while relying on a variational information bottleneck principle [1]. The paper [8] reinterprets the InfoNCE loss as a reconstruction term in the ELBO objective through an approximation of the decoder model, which effectively also makes the representation decoder free. The study [13] posits a contrastive generative process and extends the InfoNCE loss to learn the correct posterior embedding distribution (up to rotation) for an unbounded number of negative samples; the correctness result relies on a known concentration parameter of the generative process for the positive samples. In contrast, we provide expressivity results related to model capacity, independent of any model-specific parameters defining the data-generating process.

3 Proposed Model

Let $\mathcal{S} \subset \Sigma^L$ be the set of sequences of length $L < \infty$ over alphabet $\Sigma := \{A, C, G, T\}$. In many genomic sequence clustering tasks, sequences originate from unknown genomes with only sparse pairwise similarity information; furthermore, the actual number of classes is unknown. In this regard, our goal is to learn an embedding function ϕ that captures the underlying cluster structure (i.e., *binning*) while also modeling the uncertainty in embedding space. Each cluster is expected to contain DNA fragments belonging to the same species.

Objective: We aim to learn an embedding function, ϕ, that maps sequences into a latent space, where the distance reflects cluster membership. Specifically, for a given threshold $\tau > 0$, we require:

$$\|\phi(\mathbf{s}) - \phi(\mathbf{r})\| < \tau \text{ if and only if } \ell(\mathbf{s}) = \ell(\mathbf{r}) = k, \tag{1}$$

for some $k \in [K]$ where $\ell(\mathbf{s})$ denotes the cluster label of the DNA sequence $\mathbf{s} \in \mathcal{S}$. Therefore, an embedding function satisfying this condition maps sequences from the same cluster close together, and sequences from different clusters remain well separated.

Lightweight Metagenomic Binning: Our work builds on the works [5, 21], which introduced a state-of-the-art metagenomic binning algorithm. These methods achieve competitive accuracy while being several orders of magnitude faster than large genomic foundation models because they construct embeddings from efficient k-mer representations rather than using heavy sequence transformers. We adopt this principle of lightweight non-linear embeddings as the starting point for our approach.

In the works [5, 21], each DNA sequence in the dataset is split into two equal-length segments to form a *positive* pair, while *negative* pairs are designed by choosing two distinct sequences at random. For every segment pair, both k-mer profiles are computed and passed through a shared neural network to map them into an embedding space. The contrastive loss used there encourages embeddings of positive pairs to be close and embeddings of negative pairs to be far apart, thereby learning a genome-aware representation without supervision. These embeddings are later clustered with a standard algorithm, and all DNA fragments in a cluster are assumed to belong to a single species.

Formally, let $\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^N$ denote the set of DNA sequences in our dataset, with $\mathbf{s}_i^{(l)}$ and $\mathbf{s}_i^{(r)}$ being the left and right halves of each sequence. We construct triplets $\big\{(\mathbf{s}_i^{(l)}, \mathbf{s}_j^{(r)}, y_{ij})\big\}_{(i,j)\in\mathcal{I}}$ where $\mathcal{I}$ is the set of sequence index pairs, and $y_{ij} = 1$ if the two segments originate from the same sequence (positive pair) and otherwise $y_{ij} = 0$ (negative pair). The neural network parameters Ω are trained by minimizing

$$\mathcal{L}(\Omega) = -\sum_{(i,j)\in\mathcal{I}} \Big[y_{ij} \log P(Y_{ij} = 1 \mid \mathbf{s}_i^{(l)}, \mathbf{s}_j^{(r)}) + (1 - y_{ij}) \log\big(1 - P(Y_{ij} = 1 \mid \mathbf{s}_i^{(l)}, \mathbf{s}_j^{(r)})\big) \Big], \tag{2}$$

with a success probability defined by $P(Y_{ij} = 1 \mid \mathbf{s}_i^{(l)}, \mathbf{s}_j^{(r)}) := \exp\big(-\big\|\mathbf{z}_i^{(l)} - \mathbf{z}_j^{(r)}\big\|_2^2\big)$ where $\mathbf{z}_i$ denotes the embedding of $\mathbf{s}_i$. Since we assume that our dataset contains many different genomes (i.e., classes), negative pairs constructed by sampling random segments are more likely to originate from different genomes. Similarly, the positive pairs consist of fragments from the same genome due to the nature of the construction procedure. Hence, the model learns to produce similar embeddings for k-mer profiles from the same genome and dissimilar embeddings for profiles from different genomes.

Motivation: While the above contrastive framework provides an efficient way to learn genome-aware embeddings, its reliance on squared Euclidean distances between deterministic points (non-uncertain representations) in latent space imposes a fundamental limitation. As discussed in the introduction, many DNA fragments do not belong exclusively to a single cluster: they may genuinely occur in multiple genomes (e.g., through horizontal gene transfer [2]) or, conversely, fragments from distinct genomes may yield indistinguishable k-mer features. In both cases, the fixed-point embeddings produced by the model above collapse such sequences to the same location in the latent space, preventing any clustering algorithm from assigning them consistently. Lemma 3.3 and Corollary 3.3.1 (below) formalize this limitation by showing that it is not always possible to satisfy all pairwise constraints under deterministic embeddings.

In particular, if the set of DNA sequences is sufficiently large, then the DNA fragments originating from different clusters cannot simultaneously be mapped close to these cluster centroids to represent their membership while also being placed far apart from one another to reflect their pairwise dissimilarities. This motivates our shift to probabilistic embeddings, where each fragment is represented by

a distribution, explicitly encoding ambiguity and thereby expanding the embedding space's flexibility to better separate multi-cluster or otherwise indistinguishable sequences.

Proposed model: Our approach represents each DNA fragment as a Gaussian distribution, $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{S})$ where two encoder networks output the mean–covariance pair $(\boldsymbol{\mu}, \mathbf{S})$. This provides a principled way of modelling sequence-level ambiguity rather than evaluating the success probability $p(Y_{ij} = 1 \mid \mathbf{s}_i, \mathbf{s}_j)$ at fixed points. We define a new conditional distribution that *marginalizes over the uncertainty of both embeddings* $(\mathbf{z}_i, \mathbf{z}_j)$,

$$P(Y_{ij} = 1 \mid \mathbf{s}_i, \mathbf{s}_j) = \mathbb{E}_{\substack{\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i) \\ \mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{S}_j)}} \left[p(Y_{ij} = 1 \mid \mathbf{z}_i, \mathbf{z}_j) \right]$$
$$= \mathbb{E}_{\substack{\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i) \\ \mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{S}_j)}} \left[\exp\left(-\frac{1}{2} (\mathbf{z}_i - \mathbf{z}_j)^\top \mathbf{K}_{ij}^{-1} (\mathbf{z}_i - \mathbf{z}_j) + \beta \right) \right], \tag{3}$$

where $\mathbf{K}_{ij} \succ 0$ is a positive definite matrix and β accounts for the volume of the latent Gaussian distributions (discussed below). The inner part of the expectation captures the (unnormalized) Gaussian likelihood of the embedding difference $\mathbf{z}_i - \mathbf{z}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{ij})$, and $\mathbf{K}_{ij}$ can therefore be seen as representing the uncertainty in the distance between $\mathbf{z}_i$ and $\mathbf{z}_j$ providing different weights to the differences across each embedding dimension; $\mathbf{K}_{ij}$ can thus also be interpreted as a local metric tensor. By Lemma 3.1, we can find the closed-form solution of the expectation term over embeddings (proofs of all formal results are placed in Appendix A.2):

Lemma 3.1. *(Closed-form expectation) Let* $\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i)$ *and* $\mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{S}_j)$ *be independent random variables. For a given positive definite matrix* $\mathbf{K}_{ij} \succ 0$*, the expectation in Eq. 3 is equal to*

$$\mathbb{E}_{\substack{\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i) \\ \mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{S}_j)}} \left[p(Y_{ij} = 1 \mid \mathbf{z}_i, \mathbf{z}_j) \right] = \frac{1}{\sqrt{|\mathbf{K}_{ij}^{-1} \mathbf{S}_{ij} + \mathbf{I}|}} \exp\left(-\frac{1}{2} \boldsymbol{\mu}_{ij}^\top (\mathbf{S}_{ij} + \mathbf{K}_{ij})^{-1} \boldsymbol{\mu}_{ij} + \beta \right), \tag{4}$$

where $\boldsymbol{\mu}_{ij} := \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ *and* $\mathbf{S}_{ij} := \mathbf{S}_i + \mathbf{S}_j$.

As $\mathbf{K}_{ij}$ (the local metric tensor) can be seen as representing the uncertainty in the distance between $\mathbf{z}_i$ and $\mathbf{z}_j$, we let $\mathbf{K}_{ij} := \alpha(\mathbf{S}_i + \mathbf{S}_j)$ so that $\mathbf{K}_{ij}$ reflects the point-wise uncertainty in the embeddings. The parameter α can be learnable, but in the remainder of the paper we set $\alpha = 1$; Appendix A.1 includes further insight into the role of $\alpha \in \mathbb{R}^+$. With this choice, the expectation in Eq. 3 simplifies algebraically to $2^{-D/2} \exp\left(-\frac{1}{4}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^\top (\mathbf{S}_i + \mathbf{S}_j)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \beta\right)$. In high-dimensional spaces, the overlap between two probability distributions shrinks rapidly as the dimensionality D grows. This causes the expected similarity to tend toward zero, even for two identical sequences. By setting $\beta := D \ln 2/2$ we always have that $P(Y_{ii} = 1 \mid \mathbf{s}_i, \mathbf{s}_i) = 1$, and Eq. 3 simplifies to:

$$P(Y_{ij} = 1 \mid \mathbf{s}_i, \mathbf{s}_j) = \exp\left(-\frac{1}{4} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^\top (\mathbf{S}_i + \mathbf{S}_j)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \right). \tag{5}$$

Here, it is worth emphasizing that both the mean vectors $(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$ and covariance matrices $(\mathbf{S}_i, \mathbf{S}_j)$ are parameterized by two simple neural networks denoted by ϕ_μ and ϕ_σ. They consist of a single hidden layer including 512 units with sigmoid activation functions, and the output dimension (i.e., D) is set to 256. We optimize these neural networks by minimizing the log-likelihood in Eq. 2 with the success probability in Eq. 5. Details about the optimization is provided in Section 4.

Definition 3.2 (ϵ-distinguishable)**.** *For a given* $\epsilon \in (0, 1/2)$*, a probabilistic mapping* $\phi = (\phi_\mu, \phi_\sigma) : \mathcal{S} \to \mathbb{R}^D \times \mathbb{R}_+^D$ *parametrized by the mean and variance functions* $\phi_\mu : \mathcal{S} \to \mathbb{R}^D$ *and* $\phi_\sigma : \mathcal{S} \to \mathbb{R}_+^D$ *is an* ***ϵ-distinguishable embedding*** *function for a set* $\{(\mathbf{s}_i, \mathbf{s}_j, y_{ij})\}_{(i,j) \in \mathcal{I}} \subseteq \mathcal{S}^2 \times \{0, 1\}$ *if it satisfies:*

$$P(Y_{ij} = y_{ij} \mid \mathbf{s}_i, \mathbf{s}_j) \geq (1 - \epsilon) \tag{6}$$

for all triplets $(\mathbf{s}_i, \mathbf{s}_j, y_{ij})$ *indexed by* $\mathcal{I}$*. We omit* y_{ij} *for brevity when it is clear from context.*

Intuitively, an ϵ-distinguishable embedding function guarantees that positive pairs remain close in the latent space, while negative pairs are sufficiently separated. Lemma 3.3 formalizes this relationship by providing explicit bounds on the Euclidean distance between the embedding means as a function of the corresponding variances, thus offering theoretical guarantees for pairwise distinguishability.

Lemma 3.3. *Let $\epsilon \in (0, 1/2)$, and let $\phi = (\phi_\mu, \phi_\sigma) : \mathcal{S} \to \mathbb{R}^D \times \mathbb{R}^D_+$ be an ϵ-distinguishable embedding function for a triplet $(\mathbf{s}_i, \mathbf{s}_j, y_{ij}) \in \mathcal{S}^2 \times \{0, 1\}$. Then the following bounds hold:*

$$\min_d \{(\phi_\sigma(\mathbf{s}_i) + \phi_\sigma(\mathbf{s}_j))_d\} \log\left(\frac{1}{\epsilon^4}\right) \leq \|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2 \qquad \text{if } y_{ij} = 0, \tag{7}$$

$$\max_d \{(\phi_\sigma(\mathbf{s}_i) + \phi_\sigma(\mathbf{s}_j))_d\} \log\left(\frac{1}{(1-\epsilon)^4}\right) \geq \|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2 \qquad \text{if } y_{ij} = 1. \tag{8}$$

These bounds highlight the critical role of variances in our setting. Intuitively, if a sequence is associated with multiple clusters, its mean representation ($i.e., \phi_\mu(\mathbf{s})$) cannot be simultaneously close to all corresponding centroids, since these centroids must remain sufficiently separated. For instance, Figure 3 depicts six blue points placed at distance τ from a central green point, while also being pairwise τ-separated. In this two-dimensional configuration, it is not possible to place an additional point (such as the red point) that lies within distance τ of the green point while remaining more than τ away from all the blue points. Hence, the covariance terms $\phi_\sigma(\mathbf{s})$ must adaptively increase to satisfy the distinguishability condition in Eq. 3.2.

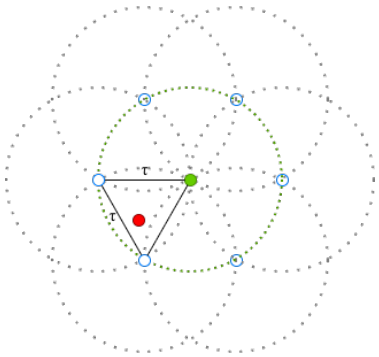


Figure 3: Packing number ($\mathcal{P}_\tau^D$).

Before stating the following corollary, we first introduce the *packing number* $\mathcal{P}_\tau^D$, defined as the maximum number of τ-separated distinct points in a ball of radius τ in D-dimensional space [26]. It will help us to formalize the limitations of the embedding function.

Corollary 3.3.1. *Let $\phi = (\phi_\mu, \phi_\sigma) : \mathcal{S} \to \mathbb{R}^D \times \mathbb{R}^D_+$ be an embedding function with constant variance, such that $\phi_\sigma(\mathbf{s})_d = \sigma^2$ is invariant for all $\mathbf{s} \in \mathcal{S}$ and $d \in [D]$. If there exists a collection of $N \geq \mathcal{P}_\tau^D + 1$ sequences $\{\mathbf{s}_1, \ldots, \mathbf{s}_N\} \subseteq \mathcal{S}$ that are each positive pairs with a common sequence $\mathbf{s}_0 \in \mathcal{S}$, but are pairwise negative with respect to each other, then ϕ cannot be an ϵ-distinguishable embedding function for any $\epsilon \in (0, 1/2)$.*

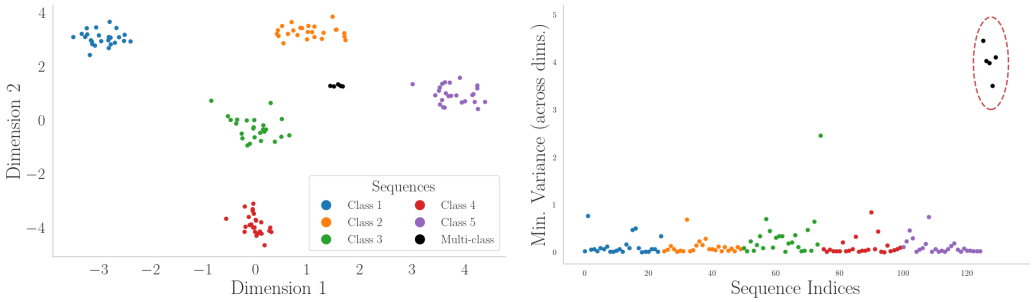
This packing construction is an expressivity result and does not imply that the extremal configuration is attained in the benchmark datasets. From Corollary 3.3.1, we see that fixed-variance embeddings have intrinsic limitations in expressiveness: they cannot simultaneously satisfy the pairwise constraints of a sufficiently large sequence set. Allowing covariance terms to vary introduces additional degrees of freedom, increasing the modeling capacity of the embedding function. Theorem 3.4 further characterizes how these variances must adapt when increasingly tight distinguishability is required for a sequence that forms positive pairs with multiple mutually negative sequences.

Theorem 3.4. *Let $\{\phi^t = (\phi_\mu, \phi_\sigma^t)\}_{t \geq 1}$ have fixed bounded means and $\max_d \phi_\sigma^t(\mathbf{s}_i)_d \leq \kappa \min_d \phi_\sigma^t(\mathbf{s}_i)_d \; \forall\, i \in [N]$, $t \geq 1$ for some $\kappa < \infty$. Consider $\{\mathbf{s}_i\}_{i=0}^N$, $N \geq 2$, such that $(\mathbf{s}_0, \mathbf{s}_i)$ are positive and $(\mathbf{s}_i, \mathbf{s}_j)$ are negative for all $i \neq j \in [N]$, with pairwise distinct mean embeddings. Assume that, for every $d \in [D]$, there exists $i \in [N]$ such that $\phi_\mu(\mathbf{s}_0)_d \neq \phi_\mu(\mathbf{s}_i)_d$. Then ϕ^t is eventually ϵ-distinguishable for every $\epsilon \in (0, 1/2)$ if and only if $\max_{i,d} \phi_\sigma^t(\mathbf{s}_i)_d \to 0$ and $\min_d \phi_\sigma^t(\mathbf{s}_0)_d \to \infty$.*

This result underscores the importance of probabilistic embeddings: by having covariance terms, the model can represent complex relationships among sequences that deterministic embeddings cannot capture. We note that we represent each DNA fragment as a single Gaussian distribution rather than a mixture model in order to preserve scalability and maintain the closed-form formulation in Eq. 5. Moreover, in metagenomic settings, the number of species a DNA fragment belongs to is unknown beforehand, making species-specific mixture representations more difficult to estimate and optimize.

4 Experiments

We evaluate UNCERTAINGEN using the same experimental setup as [32, 5] to ensure a fair comparison with previous deterministic and large genome foundation models. We also employ the same benchmark datasets, introduced by the work [32] and adopted in subsequent study [5], which



(a) Learned embeddings

(b) Variance distribution

Figure 4: Visualization of the learned embeddings and the variance distribution of the sequences.

provide well-defined ground truth labels essential for evaluating embedding quality. Reference 5/6 are synthetically constructed from fungal and viral reference genomes, whereas Plant 5/6 and Marine 5/6 are derived from the more complex CAMI2 settings; additional dataset statistics are provided in Appendix B. Although real metagenomic samples would test the full performance of the pipeline, controlled benchmarks allow us to isolate the contribution of the representation method from confounding factors such as assembly quality, coverage variation, and chimeric sequences. This controlled setting is particularly appropriate for our goal of evaluating whether probabilistic embeddings better capture genomic structure than deterministic alternatives. We provide further details about baselines and datasets in Appendix B.

Training procedure. For our method, training was performed within a contrastive learning framework by optimizing the objective function given in Eq. 2. Positive pairs were obtained by splitting each fragment into two halves, guaranteeing that both subsequences come from the same genome. Negative pairs were formed by randomly pairing fragments from the dataset; this procedure ensures, with high probability, that the paired sequences come from different genomes. From the original dataset consisting of two million pairs [32], we randomly subsample 10^5 pairs to demonstrate that our method is effective even with smaller training sets compared to large genome foundation models. For each positive pair, we generated 200 negative pairs, resulting in a total of $2,01 \times 10^7$ pairs.

We trained our model using the Adam optimizer with a learning rate of 10^{-2} and a batch size of 10^5 pairs. To improve stability, we first trained the mean encoder network alone for 50 epochs where we considered the squared Euclidean distance instead of Eq. 5, more specifically $\exp(-\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2^2/4)$. Then, we trained the variance network only for an additional 25 epochs with Eq. 5.

We employed a thresholding strategy to bound the embedding space and avoid unnecessary updates from already separated negative pairs ($y_{ij} = 0$). Specifically, for negative pairs, the distance inside the exponential term was clipped at a fixed threshold, so that the resulting gradient is zero once the pair is sufficiently far apart. This follows the standard margin intuition in contrastive metric learning: negative pairs need only satisfy a separation constraint, rather than be pushed indefinitely far apart.

It should be emphasized that this thresholding strategy remains entirely consistent with the theoretical guarantees established in Section 3. Our distinguishability framework (Definition 3.2) requires that for any pair, the success probability $P(Y_{ij} = y_{ij} \mid \mathbf{s}_i, s_j)$ meets or exceeds a boundary $(1 - \epsilon)$. By discarding the gradients once a negative pair is sufficiently separated, we simply recognize that the specific pairwise distinguishability constraint has been satisfied. Therefore, the optimization procedure prioritizes updates for pairs that do not yet meet the theoretical criteria for separation or proximity, ensuring that the final learned representations respect the distinguishability bounds. Further details and discussions can be found in Appendix B.

4.1 Toy Example with k-mer Dataset

To investigate our models' behavior in a controlled setting, we designed a synthetic dataset of 4-mer sequences. This setup allows us to assess whether the model can learn meaningful low-dimensional embeddings that reflect cluster structure and how it represents sequences spanning multiple classes.

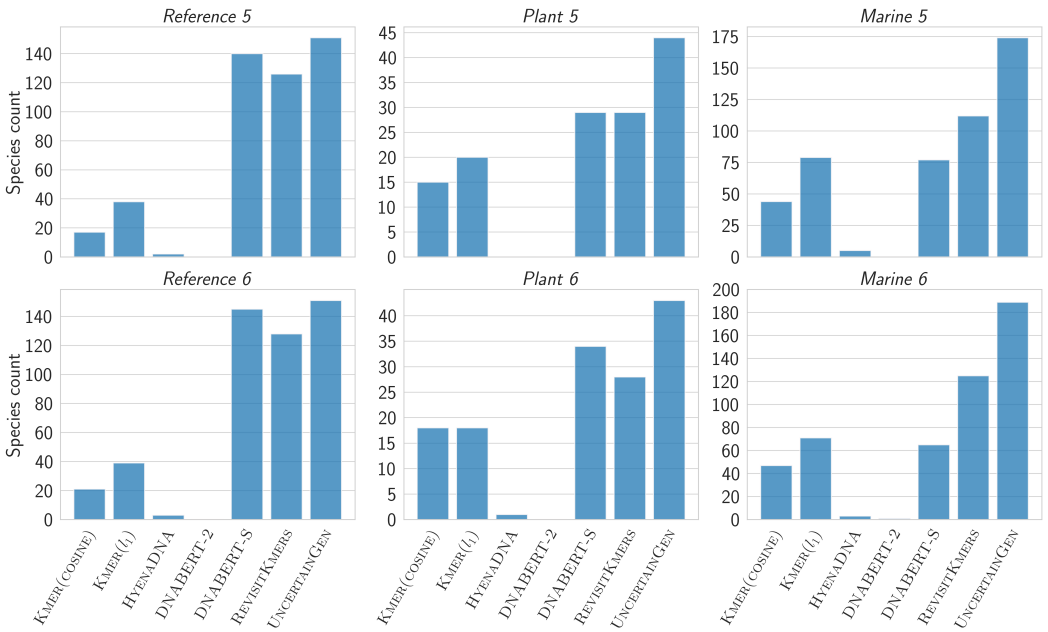


Figure 5: **Metagenomic binning results.** The number of detected HQ clusters ($F_1 > 0.9$).

We generated sequences of length 100 from multinomial distributions defining 5 distinct classes, each with a characteristic 4-mer compositions. To simulate ambiguity, we additionally generated 5 "multi-class" sequences by combining k-mer counts from multiple classes, representing inputs that do not belong exclusively to a single class. This design allows us to evaluate how the model handles both well-separated clusters and overlapping class memberships.

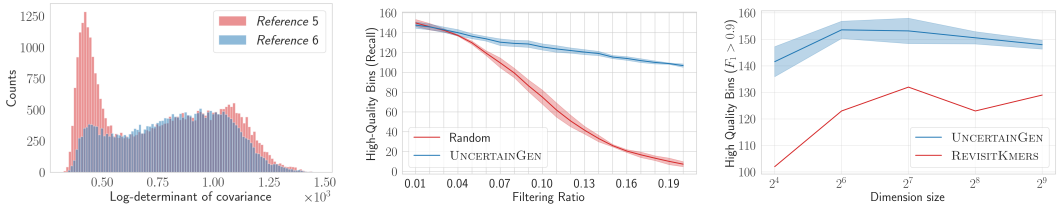
Positive sequence pairs were formed by sampling sequences from the same class, while negative pairs were drawn across different classes, which introduces the possibility of false negatives. For multi-class sequences, pairs were also constructed with sequences from their contributing classes, allowing the model to learn representations that account for both pure and mixed memberships. Due to space limitations, we provide the details in Appendix B.1.

We learn the embeddings in a 2-dimensional space, enabling direct visualization of the learned geometry without relying on dimensionality reduction techniques that could distort structural relationships (Figure 4a). The resulting embeddings reveal well-separated clusters corresponding to distinct species, while sequences that belong to multiple classes occupy intermediate regions (black points).

For each sequence, our model predicts a diagonal covariance matrix quantifying the degree of uncertainty in its placement. This uncertainty is particularly pronounced for multi-class sequences, as reflected in their larger covariance values in Figure 4b. In line with our theoretical results (Lemma 3.3 and Theorem 3.4), the minimum variance across dimensions (i.e. $\min_d\{(\phi_\sigma(\mathbf{s}))_d\}$) provides a lower bound on pairwise distances, and sequences associated with multiple classes indeed display a higher minimum variance. It shows that the model not only separates clusters effectively, but also encodes the uncertainty of ambiguous cases. We further compare relative distances in the embedding space in Appendix B.1, showing that accounting for uncertainty through the generalized Mahalanobis distance places ambiguous sequences substantially closer to their constituent classes than the Euclidean distance.

4.2 Metagenomics Binning

We evaluate our methods on the metagenomic binning task, where the objective is to cluster sequences into species-level groups without prior knowledge of the number of clusters. In this regard, we adopt the modified K-Medoid algorithm [32], which jointly estimates the cluster assignments and the underlying number of species. This setting is particularly challenging as it requires models to provide representations that are simultaneously discriminative and robust under unsupervised partitioning.



(a) Log-determinant of variances. (b) Sequence filtering (c) Impact of Dimension Size

Figure 6: Ablation studies examining the behavior of the proposed UNCERTAINGEN model.

For computing similarities between sequences, we employ cosine similarity for all genome-scale foundation models as well as the KMERS(COSINE) baseline. In contrast, we use an exponential kernel over the ℓ_1 distance for KMERS(ℓ_1), and ℓ_2 distance for REVITKMERS. We use an exponential kernel over the generalized Mahalanobis term in Eq.5 as a natural choice for our model.

For the evaluation of the models, we consider the number of detected high-quality bins (i.e., the number of clusters whose F_1 score is greater than 0.9), and a more detailed comparison is provided in the appendix. As shown in Figure 5, across all datasets, UNCERTAINGEN consistently outperforms its deterministic counterparts. Moreover, our method also surpasses the strongest competing genome-scale foundation model, DNABERT-S across all datasets, while at the same time providing a light-weight alternative to the foundation models. The latter property is illustrated in Figure 2, showing the relation between model size and the average number of high-quality bin ratios. UNCERTAINGEN also exhibits high performance stability; detailed five-run statistics are provided in Appendix B.2. These results demonstrate that our approach is not only competitive with state-of-the-art foundation models but also offers the added benefit of a principled probabilistic formulation, enabling more robust and interpretable clustering in metagenomic settings with a smaller number of parameters.

4.3 Ablation Studies

To better understand the behavior of our model, we perform a series of ablation studies. Due to space constraints, we only discuss the results for the *Reference* datasets in the main text, but a more comprehensive analysis across all datasets is provided in Appendix B.3.

Distribution of variances. We first analyze the distribution of the predicted variances for sequences in the test data. Specifically, we report $\sum_{d=1}^{D} \log\left(\phi_\sigma(\mathbf{s})_d + 1\right)$, which aggregates uncertainty across dimensions. As shown in Figure 6a, both variants of *Reference* exhibit multimodal distributions, suggesting that the model captures non-trivial heterogeneity in sequence-level uncertainty. This indicates that the variance estimates are not merely noise but encode meaningful structure about the underlying sequence distributions.

Sequence filtering. We filter out sequences with the largest log-determinant values to examine which sequences the model identifies as uncertain, and we report the number of clusters having a recall score ≥ 0.9 (Figure 6b). For comparison, we also include a random filtering baseline. To ensure fairness, removed sequences are assigned to a "garbage" label so that they do not artificially inflate false negatives. Our results show that filtering by model uncertainty consistently retains a larger number of high-recall clusters compared to random filtering. This shows that the model assigns higher uncertainty to sequences from low-quality bins or to those that would otherwise contribute to false negatives, thereby acting as an effective mechanism for uncertainty-aware sequence selection.

Dimension size. As discussed in the theoretical analysis of Section 3, incorporating covariance terms provides additional representational capacity to our model, UNCERTAINGEN, compared to its Euclidean distance version, REVISITKMERS. Beyond capturing predictive uncertainty, the covariance-dependent Mahalanobis term induces a data-adaptive, non-Euclidean geometry in the embedding space, which allows the model to better separate complex sequence structures. Empirical results in Figure 6c support this intuition. We observe consistent improvements when covariance terms are included, with the gains being especially pronounced in lower-dimensional settings where representational bottlenecks are most restrictive.

5 Conclusions, Limitations and Future Work

We introduced UNCERTAINGEN, the first probabilistic embedding framework for metagenomic binning. Unlike deterministic k-mer or genome foundation models, our method maps each DNA fragment to a distribution in latent space, explicitly encoding sequence-level uncertainty. Theoretical analysis showed the limitations of fixed-variance embeddings and how adaptive covariance terms enable rigid pairwise distinguishability, while experiments on both synthetic and real metagenomic data demonstrated consistent gains in binning quality over strong deterministic baselines.

Our study also reveals some limitations. We relied on a simplified self-supervised pairing strategy and two-layer networks; more expressive architectures or richer positive/negative sampling schemes may further enhance performance. In addition, although variance terms estimate uncertainty, whether these estimates are calibrated or not remains an open question. Moreover, while the toy experiment directly validates this behavior under controlled ambiguity, the benchmark datasets do not annotate conserved, horizontally transferred, or mobile genomic regions, preventing us from identifying the biological source of high learned variance.

Future work will extend UNCERTAINGEN beyond k-mer-based representations, explore hierarchical or non-Gaussian distributions for even richer uncertainty modeling, and integrate our embeddings into end-to-end pipelines for metagenome reconstruction. We hope this framework stimulates broader adoption of uncertainty-aware representations in computational genomics.

Acknowledgements

We thank the reviewers for their constructive feedback and insightful comments. We also thank all members of Albertsen Lab at Aalborg University and Katja Hose for the valuable discussions. This work was supported by a research grant (VIL50093) from VILLUM FONDEN.

References

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep Variational Information Bottleneck. In *ICLR*, 2017.

[2] Brian J. Arnold, I.-Ting Huang, and William P. Hanage. Horizontal gene transfer and adaptive evolution in bacteria. *Nature Reviews Microbiology*, 20(4):206–218, April 2022.

[3] Parikshit Bansal, Ali Kavis, and Sujay Sanghavi. Understanding Self-Supervised Learning via Gaussian Mixture Models, February 2025.

[4] Ricardo Cavicchioli et al. Scientists' warning to humanity: Microorganisms and climate change. *Nature Reviews Microbiology*, 17(9):569–586, September 2019.

[5] Abdulkadir Çelikkanat, Andres Masegosa, and Thomas Nielsen. Revisiting k-mer profile for effective and scalable genome representation learning. *Advances in Neural Information Processing Systems*, 37:118930–118952, 2024.

[6] Chon-Kit Kenneth Chan, Arthur L. Hsu, Saman K. Halgamuge, and Sen-Lin Tang. Binning sequences using very sparse labels within a metagenome. *BMC Bioinformatics*, 9(1):215, April 2008.

[7] Paul G. Falkowski, Tom Fenchel, and Edward F. Delong. The Microbial Engines That Drive Earth's Biogeochemical Cycles. *Science*, 320(5879):1034–1039, May 2008.

[8] Minoh Jeong, Seonho Kim, and Alfred O Hero. Probabilistic variational contrastive learning. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2025.

[9] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

[10] Dongwan D Kang, Feng Li, Edward Kirton, Ashleigh Thomas, Rob Egan, Hong An, and Zhong Wang. Metabat 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ*, 7:e7359, 2019.

[11] Ivan Karpukhin, Stanislav Dereka, and Sergey Kolesnikov. Probabilistic embeddings revisited. *The Visual Computer*, 40(6):4373–4386, June 2024.

[12] Raiyan R Khan, Philippe Chlenski, and Itsik Pe'er. Hyperbolic genome embeddings. *arXiv preprint arXiv:2507.21648*, 2025.

[13] Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic Contrastive Learning Recovers the Correct Aleatoric Uncertainty of Ambiguous Inputs. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17085–17104. PMLR, July 2023.

[14] Victor Kunin, Alex Copeland, Alla Lapidus, Konstantinos Mavromatis, and Philip Hugenholtz. A Bioinformatician's Guide to Metagenomics. *Microbiology and Molecular Biology Reviews*, 72(4):557–578, December 2008.

[15] Svetlana Kutuzova, Pau Piera, Knud Nor Nielsen, Nikoline S Olsen, Leise Riber, Alex Gobbi, Laura Milena Forero-Junco, Peter Erdmann Dougherty, Jesper Cairo Westergaard, Svend Christensen, et al. Binning meets taxonomy: Taxvamb improves metagenome binning using bi-modal variational autoencoder. *bioRxiv*, pages 2024–10, 2024.

[16] Fernando Meyer et al. Critical Assessment of Metagenome Interpretation: The second round of challenges. *Nature Methods*, 19(4):429–440, April 2022.

[17] Boris Muzellec and Marco Cuturi. Generalizing point embeddings using the wasserstein space of elliptical distributions. *Advances in Neural Information Processing Systems*, 31, 2018.

[18] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36:43177–43201, 2023.

[19] Jakob Nybo Nissen, Joachim Johansen, Rosa Lundbye Allesøe, Casper Kaae Sønderby, Jose Juan Almagro Armenteros, Christopher Heje Grønbech, Lars Juhl Jensen, Henrik Bjørn Nielsen, Thomas Nordahl Petersen, Ole Winther, et al. Improved metagenome binning and assembly using deep variational autoencoders. *Nature biotechnology*, 39(5):555–560, 2021.

[20] Seong Joon Oh, Kevin P. Murphy, Jiyan Pan, Joseph Roth, Florian Schroff, and Andrew C. Gallagher. Modeling Uncertainty with Hedged Instance Embeddings. In *International Conference on Learning Representations*, September 2019.

[21] Shaojun Pan, Xing-Ming Zhao, and Luis Pedro Coelho. Semibin2: self-supervised contrastive learning leads to better mags for short-and long-read sequencing. *Bioinformatics*, 39(Supplement_1):i21–i29, 2023.

[22] Yichun Shi and Anil Jain. Probabilistic Face Embeddings. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6901–6910, Seoul, Korea (South), October 2019. IEEE.

[23] Hanno Teeling, Anke Meyerdierks, Margarete Bauer, Rudolf Amann, and Frank Oliver Glöckner. Application of tetranucleotide frequencies for the assignment of genomic fragments. *Environmental Microbiology*, 6(9):938–947, 2004.

[24] Ben Temperton and Stephen J. Giovannoni. Metagenomics: Microbial diversity through a scratched lens. *Current Opinion in Microbiology*, 15(5):605–612, October 2012.

[25] Kenneth Timmis, Willem M. de Vos, Juan Luis Ramos, Siegfried E. Vlaeminck, Auxiliadora Prieto, Antoine Danchin, Willy Verstraete, Victor de Lorenzo, Sang Yup Lee, Harald Brüssow, James Kenneth Timmis, and Brajesh K. Singh. The contribution of microbial biotechnology to sustainable development goals. *Microbial Biotechnology*, 10(5):984–987, September 2017.

[26] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[27] Luke Vilnis and Andrew McCallum. Word Representations via Gaussian Embedding. In *ICLR*. arXiv, May 2015.

[28] Ziye Wang, Ronghui You, Haitao Han, Wei Liu, Fengzhu Sun, and Shanfeng Zhu. Effective binning of metagenomic contigs using contrastive multi-view representation learning. *Nature Communications*, 15(1):585, 2024.

[29] Frederik Warburg, Marco Miani, Silas Brack, and Søren Hauberg. Bayesian Metric Learning for Uncertainty Quantification in Image Retrieval. *Advances in Neural Information Processing Systems*, 36:69178–69190, December 2023.

[30] Yu-Wei Wu, Blake A Simmons, and Steven W Singer. Maxbin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics*, 32(4):605–607, 2016.

[31] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023.

[32] Zhihan Zhou, Weimin Wu, Harrison Ho, Jiayi Wang, Lizhen Shi, Ramana V Davuluri, Zhong Wang, and Han Liu. Dnabert-s: pioneering species differentiation with species-aware dna embeddings. *Bioinformatics*, 41(Supplement_1):i255–i264, 07 2025.

A Appendix

A.1 The role of α in $\mathbf{K}_{ij} = \alpha(\mathbf{S}_i + \mathbf{S}_j)$

Consider the closed form expectation of Lemma 3.1:

$$\mathbb{E}_{\substack{\mathbf{z}_i\sim\mathcal{N}(\boldsymbol{\mu}_i,\mathbf{S}_i)\\ \mathbf{z}_j\sim\mathcal{N}(\boldsymbol{\mu}_j,\mathbf{S}_j)}}\left[\exp\left(-\frac{1}{2}(\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{K}_{ij}^{-1}(\mathbf{z}_i-\mathbf{z}_j)+\beta\right)\right] = \frac{1}{\sqrt{|\mathbf{K}_{ij}^{-1}\mathbf{S}_{ij}+\mathbf{I}|}}\exp\left(-\frac{1}{2}\boldsymbol{\mu}_{ij}^\top(\mathbf{S}_{ij}+\mathbf{K}_{ij})^{-1}\boldsymbol{\mu}_{ij}+\beta\right). \quad (9)$$

where $\boldsymbol{\mu}_{ij} := \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ and $\mathbf{S}_{ij} := \mathbf{S}_i + \mathbf{S}_j$. By setting $\mathbf{K}_{ij} = \alpha(\mathbf{S}_i + \mathbf{S}_j) = \alpha\mathbf{S}_{ij}$, then we have

$$\frac{1}{\sqrt{|\mathbf{K}_{ij}^{-1}\mathbf{S}_{ij}+\mathbf{I}|}} = \frac{1}{\sqrt{|\alpha^{-1}\mathbf{I}+\mathbf{I}|}} = \frac{1}{\sqrt{|(1+\alpha^{-1})\mathbf{I}|}} = (1+\alpha^{-1})^{-D/2} \quad (10)$$

and

$$-\frac{1}{2}\boldsymbol{\mu}_{ij}^\top(\mathbf{S}_{ij}+\mathbf{K}_{ij})^{-1}\boldsymbol{\mu}_{ij} = -\frac{1}{2}\boldsymbol{\mu}_{ij}^\top(\mathbf{S}_{ij}+\alpha\mathbf{S}_{ij})^{-1}\boldsymbol{\mu}_{ij} \quad (11)$$

$$= -\frac{1}{2}\boldsymbol{\mu}_{ij}^\top((1+\alpha)\mathbf{S}_{ij})^{-1}\boldsymbol{\mu}_{ij} \quad (12)$$

$$= -\frac{1}{2(1+\alpha)}\boldsymbol{\mu}_{ij}^\top\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}. \quad (13)$$

To isolate the effect of α, we set $\beta = 0$ in the following analysis. Hence,

$$\mathbb{E}_{\substack{\mathbf{z}_i\sim\mathcal{N}(\boldsymbol{\mu}_i,\mathbf{S}_i)\\ \mathbf{z}_j\sim\mathcal{N}(\boldsymbol{\mu}_j,\mathbf{S}_j)}}\left[\exp\left(-\frac{1}{2}(\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{K}_{ij}^{-1}(\mathbf{z}_i-\mathbf{z}_j)+\beta\right)\right] \quad (14)$$

$$= \frac{1}{\sqrt{|\mathbf{K}_{ij}^{-1}\mathbf{S}_{ij}+\mathbf{I}|}}\exp\left(-\frac{1}{2}\boldsymbol{\mu}_{ij}^\top(\mathbf{S}_{ij}+\mathbf{K}_{ij})^{-1}\boldsymbol{\mu}_{ij}\right) \quad (15)$$

$$= (1+\alpha^{-1})^{-D/2}\exp\left(-\frac{1}{2(1+\alpha)}\boldsymbol{\mu}_{ij}^\top\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}\right) \quad (16)$$

$$\to \begin{cases}1 & \text{as } \alpha\to\infty\\ 0 & \text{as } \alpha\to 0\end{cases}. \quad (17)$$

A.2 Theoretical Analysis

Lemma (3.1). *(Closed-form expectation) Let $\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i)$ and $\mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{S}_j)$ be independent random variables. For a given positive definite matrix $\mathbf{K}_{ij} \succ 0$, the expectation is equal to*

$$\mathbb{E}_{\substack{\mathbf{z}_i\sim\mathcal{N}(\boldsymbol{\mu}_i,\mathbf{S}_i)\\ \mathbf{z}_j\sim\mathcal{N}(\boldsymbol{\mu}_j,\mathbf{S}_j)}}\left[p(Y_{ij}=1\mid\mathbf{z}_i,\mathbf{z}_j)\right] = \mathbb{E}_{\substack{\mathbf{z}_i\sim\mathcal{N}(\boldsymbol{\mu}_i,\mathbf{S}_i)\\ \mathbf{z}_j\sim\mathcal{N}(\boldsymbol{\mu}_j,\mathbf{S}_j)}}\left[\exp\left(-\frac{1}{2}(\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{K}_{ij}^{-1}(\mathbf{z}_i-\mathbf{z}_j)+\beta\right)\right]$$
$$= \frac{1}{\sqrt{|\mathbf{K}_{ij}^{-1}\mathbf{S}_{ij}+\mathbf{I}|}}\exp\left(-\frac{1}{2}\boldsymbol{\mu}_{ij}^\top(\mathbf{S}_{ij}+\mathbf{K}_{ij})^{-1}\boldsymbol{\mu}_{ij}+\beta\right) \quad (18)$$

where $\boldsymbol{\mu}_{ij} := \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ and $\mathbf{S}_{ij} := \mathbf{S}_i + \mathbf{S}_j$.

Proof. Since $\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i)$ and $\mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{S}_j)$ are independent random variables, the difference, $\mathbf{z}_i - \mathbf{z}_j$ is also normally distributed so we can write $\mathbf{z}_i - \mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j, \mathbf{S}_i + \mathbf{S}_j)$. In other words, $\mathbf{z}_{ij} \sim \mathcal{N}(\boldsymbol{\mu}_{ij}, \mathbf{S}_{ij})$ where $\boldsymbol{\mu}_{ij} := \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$, $\mathbf{z}_{ij} := \mathbf{z}_i - \mathbf{z}_j$, and $\mathbf{S}_{ij} := \mathbf{S}_i + \mathbf{S}_j$, then we can

write the expected term as follows:

$$\mathbb{E}_{\substack{\mathbf{z}_i\sim\mathcal{N}(\boldsymbol{\mu}_i,\mathbf{S}_i)\\ \mathbf{z}_j\sim\mathcal{N}(\boldsymbol{\mu}_j,\mathbf{S}_j)}}\left[\exp\left(-\frac{1}{2}(\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{K}_{ij}^{-1}(\mathbf{z}_i-\mathbf{z}_j)+\beta\right)\right] \tag{19}$$

$$=\mathbb{E}_{\mathbf{z}_i-\mathbf{z}_j\sim\mathcal{N}(\boldsymbol{\mu}_{ij},\mathbf{S}_{ij})}\left[\exp\left(-\frac{1}{2}(\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{K}_{ij}^{-1}(\mathbf{z}_i-\mathbf{z}_j)+\beta\right)\right] \tag{20}$$

$$=\mathbb{E}_{\mathbf{z}_{ij}\sim\mathcal{N}(\boldsymbol{\mu}_{ij},\mathbf{S}ij)}\left[\exp\left(-\frac{1}{2}\mathbf{z}_{ij}^\top\mathbf{K}_{ij}^{-1}\mathbf{z}_{ij}+\beta\right)\right] \tag{21}$$

$$=\int\frac{1}{\sqrt{(2\pi)^D|\mathbf{S}_{ij}|}}\exp\left(-\frac{1}{2}(\mathbf{z}_{ij}-\boldsymbol{\mu}_{ij})^\top\mathbf{S}_{ij}^{-1}(\mathbf{z}_{ij}-\boldsymbol{\mu}_{ij})\right)\exp\left(-\frac{1}{2}\mathbf{z}_{ij}^\top\mathbf{K}_{ij}^{-1}\mathbf{z}_{ij}+\beta\right)\mathrm{d}\mathbf{z}_{ij} \tag{22}$$

$$=\frac{\exp(\beta)}{\sqrt{(2\pi)^D|\mathbf{S}_{ij}|}}\int\exp\left(-\frac{1}{2}\left((\mathbf{z}_{ij}-\boldsymbol{\mu}_{ij})^\top\mathbf{S}_{ij}^{-1}(\mathbf{z}_{ij}-\boldsymbol{\mu}_{ij})\right)+\mathbf{z}_{ij}^\top\mathbf{K}_{ij}^{-1}\mathbf{z}_{ij}\right)\mathrm{d}\mathbf{z}_{ij} \tag{23}$$

By expanding and regrouping the terms in the integral, we can write that:

$$\int\exp\left(-\frac{1}{2}\left((\mathbf{z}_{ij}-\boldsymbol{\mu}_{ij})^\top\mathbf{S}_{ij}^{-1}(\mathbf{z}_{ij}-\boldsymbol{\mu}_{ij})+\mathbf{z}_{ij}^\top\mathbf{K}_{ij}^{-1}\mathbf{z}_{ij}\right)\right)\mathrm{d}\mathbf{z}_{ij} \tag{24}$$

$$=\int\exp\left(-\frac{1}{2}\left(\mathbf{z}_{ij}^\top\mathbf{K}_{ij}^{-1}\mathbf{z}_{ij}+\mathbf{z}_{ij}^\top\mathbf{S}_{ij}^{-1}\mathbf{z}_{ij}-2\mathbf{z}_{ij}^\top\mathbf{S}_{ij}^{-1}\mu_{ij}+\mu_{ij}^\top\mathbf{S}_{ij}^{-1}\mu_{ij}\right)\right)\mathrm{d}\mathbf{z}_{ij} \tag{25}$$

$$=\int\exp\left(-\frac{1}{2}\left(\mathbf{z}_{ij}^\top(\mathbf{K}_{ij}^{-1}+\mathbf{S}_{ij}^{-1})\mathbf{z}_{ij}-2\mathbf{z}_{ij}^\top\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}+\boldsymbol{\mu}_{ij}^\top\mathbf{S}_{ij}^{-1}\mu_{ij}\right)\right)\mathrm{d}\mathbf{z}_{ij} \tag{26}$$

$$=\int\exp\left(-\frac{1}{2}\left(\mathbf{z}_{ij}^\top\mathbf{A}_{ij}\mathbf{z}_{ij}-2\mathbf{z}_{ij}^\top\mathbf{S}_{ij}^{-1}\mu_{ij}+\mu_{ij}^\top\mathbf{S}_{ij}^{-1}\mu_{ij}\right)\right)\mathrm{d}\mathbf{z}_{ij} \tag{27}$$

$$=\int\exp\Bigg(-\frac{1}{2}\Big((\mathbf{z}_{ij}-\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij})^\top\mathbf{A}_{ij}(\mathbf{z}_{ij}-\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij})-\boldsymbol{\mu}_{ij}^\top\mathbf{S}_{ij}^{-1}\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}+\boldsymbol{\mu}_{ij}^\top\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}\Big)\Bigg)\mathrm{d}\mathbf{z}_{ij} \tag{28}$$

$$=\sqrt{(2\pi)^D|\mathbf{A}_{ij}^{-1}|}\exp\left(-\frac{1}{2}\left(-\mu_{ij}^\top\mathbf{S}_{ij}^{-1}\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\mu_{ij}+\boldsymbol{\mu}_{ij}^\top\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}\right)\right) \tag{29}$$

where $\mathbf{A}_{ij}:=\mathbf{K}_{ij}^{-1}+\mathbf{S}_{ij}^{-1}$. In Eq. 28, we add and substract the term $\boldsymbol{\mu}_{ij}^\top\mathbf{S}_{ij}^{-1}\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}$ so that the first component depends only on $\mathbf{z}_{ij}$. It also corresponds to the numerator of a normal distribution with mean $\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}$ and covariance $\mathbf{A}_{ij}^{-1}$. Hence, the last equality follows from the standard Gaussian integral:

$$\int\exp\left(-\frac{1}{2}(\mathbf{z}_{ij}-\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij})^\top\mathbf{A}_{ij}(\mathbf{z}_{ij}-\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij})\right)\mathrm{d}\mathbf{z}_{ij}=\sqrt{(2\pi)^D\left|\mathbf{A}_{ij}^{-1}\right|}$$

Therefore, the expectation term in Eq. 19 can then be rewritten as follows:

$$\mathbb{E}_{\substack{\mathbf{z}_i\sim\mathcal{N}(\boldsymbol{\mu}_i,\mathbf{S}_i)\\ \mathbf{z}_j\sim\mathcal{N}(\boldsymbol{\mu}_j,\mathbf{S}_j)}}\left[\exp\left(-\frac{1}{2}(\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{K}_{ij}^{-1}(\mathbf{z}_i-\mathbf{z}_j)+\beta\right)\right] \tag{30}$$

$$=\frac{\exp(\beta)}{\sqrt{(2\pi)^D|\mathbf{S}_{ij}|}}\int\exp\left(-\frac{1}{2}\left((\mathbf{z}_{ij}-\boldsymbol{\mu}_{ij})^\top\mathbf{S}_{ij}^{-1}(\mathbf{z}_{ij}-\boldsymbol{\mu}_{ij})+\mathbf{z}_{ij}^\top\mathbf{K}_{ij}^{-1}\mathbf{z}_{ij}\right)\right)\mathrm{d}\mathbf{z}_{ij} \tag{31}$$

$$=\frac{\exp(\beta)}{\sqrt{(2\pi)^D|\mathbf{S}_{ij}|}}\sqrt{(2\pi)^D|\mathbf{A}_{ij}^{-1}|}\exp\left(-\frac{1}{2}\left(-\boldsymbol{\mu}_{ij}^\top\mathbf{S}_{ij}^{-1}\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\mu_{ij}+\mu_{ij}^\top\mathbf{S}_{ij}^{-1}\mu_{ij}\right)\right) \tag{32}$$

$$=\frac{\exp(\beta)}{\sqrt{|\mathbf{A}_{ij}||\mathbf{S}_{ij}|}}\exp\left(-\frac{1}{2}\left(-\mu_{ij}^\top\mathbf{S}_{ij}^{-1}\mathbf{A}_{ij}^{-1}\mathbf{S}_{ij}^{-1}\mu_{ij}+\mu_{ij}^\top\mathbf{S}_{ij}^{-1}\mu_{ij}\right)\right) \tag{33}$$

where $|\mathbf{A}_{ij}^{-1}| = |\mathbf{A}_{ij}|^{-1}$. By substituting $\mathbf{A}_{ij}$ with $(\mathbf{K}_{ij}^{-1} + \mathbf{S}_{ij}^{-1})$ and applying the Woodbury Matrix Identity, we can obtain

$$\mathbb{E}_{\substack{\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i) \\ \mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{S}_j)}} \left[\exp\left(-\frac{1}{2}(\mathbf{z}_i - \mathbf{z}_j)^\top \mathbf{K}_{ij}^{-1}(\mathbf{z}_i - \mathbf{z}_j) + \beta \right) \right] \tag{34}$$

$$= \frac{\exp(\beta)}{\sqrt{|\mathbf{A}_{ij}||\mathbf{S}_{ij}|}} \exp\left(-\frac{1}{2}\left(-\mu_{ij}^\top \mathbf{S}_{ij}^{-1} \mathbf{A}_{ij}^{-1} \mathbf{S}_{ij}^{-1} \mu_{ij} + \mu_{ij}^\top \mathbf{S}_{ij}^{-1} \mu_{ij} \right) \right) \tag{35}$$

$$= \frac{\exp(\beta)}{\sqrt{|\mathbf{K}_{ij}^{-1} + \mathbf{S}_{ij}^{-1}||\mathbf{S}_{ij}|}} \exp\left(-\frac{1}{2}\left(-\mu_{ij}^\top \mathbf{S}_{ij}^{-1} (\mathbf{K}_{ij}^{-1} + \mathbf{S}_{ij}^{-1})^{-1} \mathbf{S}_{ij}^{-1} \mu_{ij} + \mu_{ij}^\top \mathbf{S}_{ij}^{-1} \mu_{ij} \right) \right) \tag{36}$$

$$= \frac{\exp(\beta)}{\sqrt{|\mathbf{K}_{ij}^{-1}\mathbf{S}_{ij} + \mathbf{I}|}} \exp\left(-\frac{1}{2} \mu_{ij}^\top \left(\mathbf{S}_{ij}^{-1} - \mathbf{S}_{ij}^{-1} (\mathbf{K}_{ij}^{-1} + \mathbf{S}_{ij}^{-1})^{-1} \mathbf{S}_{ij}^{-1} \right) \mu_{ij} \right) \tag{37}$$

$$= \frac{1}{\sqrt{|\mathbf{K}_{ij}^{-1}\mathbf{S}_{ij} + \mathbf{I}|}} \exp\left(-\frac{1}{2} \mu_{ij}^\top \left(\mathbf{S}_{ij} + \mathbf{K}_{ij} \right)^{-1} \mu_{ij} + \beta \right) \tag{38}$$

Finally, by substituting the terms, $\boldsymbol{\mu}_{ij}$ and $\mathbf{S}_{ij}$, we can conclude that

$$\mathbb{E}_{\substack{\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i) \\ \mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{S}_j)}} \left[\exp\left(-\frac{1}{2}(\mathbf{z}_i - \mathbf{z}_j)^\top \mathbf{K}_{ij}^{-1}(\mathbf{z}_i - \mathbf{z}_j) + \beta \right) \right] \tag{39}$$

$$= \frac{1}{\sqrt{|\mathbf{K}_{ij}^{-1}(\mathbf{S}_i + \mathbf{S}_j) + \mathbf{I}|}} \exp\left(-\frac{1}{2} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^\top \left(\mathbf{S}_i + \mathbf{S}_j + \mathbf{K}_{ij} \right)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \beta \right) \tag{40}$$

□

Lemma (3.3). *Let $\epsilon \in (0, 1/2)$, and let $\phi = (\phi_\mu, \phi_\sigma) : \mathcal{S} \to \mathbb{R}^D \times \mathbb{R}_+^D$ be an ϵ-distinguishable embedding function for a triplet $(\mathbf{s}_i, \mathbf{s}_j, y_{ij}) \in \mathcal{S}^2 \times \{0, 1\}$. Then the following bounds hold:*

$$\min_d \{(\phi_\sigma(\mathbf{s}_i) + \phi_\sigma(\mathbf{s}_j))_d\} \log\left(\frac{1}{\epsilon^4} \right) \leq \|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2 \qquad \text{if } y_{ij} = 0, \tag{41}$$

$$\max_d \{(\phi_\sigma(\mathbf{s}_i) + \phi_\sigma(\mathbf{s}_j))_d\} \log\left(\frac{1}{(1-\epsilon)^4} \right) \geq \|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2 \qquad \text{if } y_{ij} = 1. \tag{42}$$

Proof. Let us define $\mathbf{S}_{ij} := \text{diag}(\phi_\sigma(\mathbf{s}_i) + \phi_\sigma(\mathbf{s}_j))$ and $\boldsymbol{\mu}_{ij} = \phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)$. We will first establish the lower bound for a negative pair (i.e., $y_{ij} = 0$). By ϵ-distinguishability, we assume that $p(Y_{ij} = 0 \mid \mathbf{s}_i, \mathbf{s}_j) \geq 1 - \epsilon$ and it implies that $\exp\left(-\frac{1}{4}\boldsymbol{\mu}_{ij}^\top \mathbf{S}_{ij}^{-1} \boldsymbol{\mu}_{ij}\right) \leq \epsilon$. Hence, $\boldsymbol{\mu}_{ij}^\top \mathbf{S}_{ij}^{-1} \boldsymbol{\mu}_{ij} \geq \log\left(\epsilon^{-4}\right)$, and we can obtain

$$\min_d \{(\mathbf{S}_{ij})_d\} \log\left(\epsilon^{-4}\right) \leq \min_d \{(\mathbf{S}_{ij})_d\} \, \boldsymbol{\mu}_{ij}^\top \mathbf{S}_{ij}^{-1} \boldsymbol{\mu}_{ij} \tag{43}$$

$$= \min_d \{(\mathbf{S}_{ij})_d\} \sum_{d=1}^{D} \frac{(\boldsymbol{\mu}_{ij})_d^2}{(\mathbf{S}_{ij})_d} \tag{44}$$

$$\leq \min_d \{(\mathbf{S}_{ij})_d\} \left(\frac{1}{\min_d \{(\mathbf{S}_{ij})_d\}} \sum_{d=1}^{D} (\boldsymbol{\mu}_{ij})_d^2 \right) \tag{45}$$

$$= \sum_{d=1}^{D} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)_d^2 \tag{46}$$

$$= \|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2. \tag{47}$$

Since $\epsilon \in (0, 1/2)$, the left-hand side is strictly positive, giving a nontrivial lower bound.

For a positive pair (i.e., $y_{ij} = 1$), similarly, we have $\exp\left(-\frac{1}{4}\,\boldsymbol{\mu}_{ij}^\top \mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij}\right) \geq 1-\epsilon$, which implies $\boldsymbol{\mu}_{ij}^\top \mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij} \leq 4\log\left(\frac{1}{1-\epsilon}\right)$. Then, we can write that

$$\max_d\{(\mathbf{S}_{ij})_d\}\,\log\left(\frac{1}{(1-\epsilon)^4}\right) \geq \max_d\{(\mathbf{S}_{ij})_d\}\boldsymbol{\mu}_{ij}^\top \mathbf{S}_{ij}^{-1}\boldsymbol{\mu}_{ij} \tag{48}$$

$$= \max_d\{(\mathbf{S}_{ij})_d\}\sum_{d=1}^{D}\frac{(\boldsymbol{\mu}_{ij})_d^2}{(\mathbf{S}_{ij})_d} \tag{49}$$

$$\geq \max_d\{(\mathbf{S}_{ij})_d\}\left(\frac{1}{\max_d\{(\mathbf{S}_{ij})_d\}}\sum_{d=1}^{D}(\boldsymbol{\mu}_{ij})_d^2\right) \tag{50}$$

$$= \sum_{d=1}^{D}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)_d^2 \tag{51}$$

$$= \|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2. \tag{52}$$

This establishes the upper bound for the embedding distance when $y_{ij} = 1$. □

Lemma A.1. *Let $\tau > 0$ and $\mathbf{z}_0 \in \mathbb{R}^D$. Then, there exist at most $\mathcal{P}_\tau^{\mathcal{D}}$ distinct points $\{\mathbf{z}_i\}_{i\geq 1} \subset \mathbb{R}^D$ such that*

$$\|\mathbf{z}_i - \mathbf{z}_0\| \leq \tau \qquad \textit{and} \qquad \|\mathbf{z}_i - \mathbf{z}_j\| \geq \tau \quad \textit{for all } 1 \leq i < j \leq \mathcal{P}_\tau^{\mathcal{D}}, \tag{53}$$

where $\mathcal{P}_\tau^{\mathcal{D}}$ is the packing number *of a unit Euclidean ball in $\mathbb{R}^D$.*

Proof. By definition, the packing number, $\mathcal{P}_\tau^D$, is the maximal number of points that can fit in $B(\mathbf{z}_0, \tau)$ such that any two points are at least τ apart. Therefore, any set of points in $B(\mathbf{z}_0, \tau)$ satisfying $\|\mathbf{z}_i - \mathbf{z}_j\| \geq \tau$ can contain at most $\mathcal{P}_\tau^D$ points. □

Corollary (3.3.1). *Let $\phi = (\phi_\mu, \phi_\sigma) : \mathcal{S} \to \mathbb{R}^D \times \mathbb{R}_+^D$ be an embedding function with constant variance, such that $\phi_\sigma(\mathbf{s})_d = \sigma^2$ is invariant for all $\mathbf{s} \in \mathcal{S}$ and $d \in [D]$. If there exists a collection of $N \geq \mathcal{P}_\tau^D + 1$ sequences $\{\mathbf{s}_1, \ldots, \mathbf{s}_N\} \subseteq \mathcal{S}$ that are each positive pairs with a common sequence $\mathbf{s}_0 \in \mathcal{S}$, but are pairwise negative with respect to each other, then ϕ cannot be an ϵ-distinguishable embedding function for any $\epsilon \in (0, 1/2)$.*

Proof. Assume for contradiction that such an ϵ-distinguishable function $\phi : \mathcal{S} \to \mathbb{R}^D \times \mathbb{R}_+^D$ exists for some $\epsilon \in (0, 1/2)$ with constant variance, i.e., $\sigma^2 := \phi_\sigma(\mathbf{s}_i)_d = \phi_\sigma(\mathbf{s}_j)_d$ for all $1 \leq i < j \leq N$ pairs and all $d \in [D]$ so Lemma 3.3 implies that

$$2\sigma^2\log\left(\frac{1}{\epsilon^4}\right) \leq \|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2 \qquad \text{if } y_{ij} = 0 \tag{54}$$

$$2\sigma^2\log\left(\frac{1}{(1-\epsilon)^4}\right) \geq \|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2 \qquad \text{if } y_{ij} = 1. \tag{55}$$

for all positive and negative $0 \leq i < j \leq N$ pairs. Note that we have

$$\log\left(\frac{1}{\epsilon^4}\right) - \log\left(\frac{1}{(1-\epsilon)^4}\right) = \log\left(\frac{(1-\epsilon)^4}{\epsilon^4}\right) = 4\log\left(\epsilon^{-1} - 1\right) > 0 \tag{56}$$

so let's define $\tau_\epsilon := \left(\log(\epsilon^{-4}) + \log((1-\epsilon)^{-4})\right)/2$, then we can write

$$\|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2 > \sigma\sqrt{2\tau_\epsilon}, \tag{57}$$

for all negative $1 \leq i < j \leq N$ pairs and we have

$$\|\phi_\mu(\mathbf{s}_0) - \phi_\mu(\mathbf{s}_i)\|_2 < \sigma\sqrt{2\tau_\epsilon} \tag{58}$$

for all positive pairs $(\mathbf{s}_0, \mathbf{s}_i)$ where $1 \leq i \leq N$. In other words, each $\phi_\mu(\mathbf{s}_i)$ $(i \geq 1)$ lies within a ball of radius $\sigma\sqrt{2\tau_\epsilon}$ centered at $\phi_\mu(\mathbf{s}_0)$. However, the condition in Eq. 57 requires that all negative pairs have to be at least $\sigma\sqrt{2\tau_\epsilon}$ apart from each other at the same time but the maximum number of points that are $\sigma\sqrt{2\tau_\epsilon}$ apart in $B(\phi_\mu(\mathbf{s}_0), \sigma\sqrt{2\tau_\epsilon})$ is at most $\mathcal{P}_\tau^D$ by Lemma A.1. Therefore, we obtain a contradiction. □

Theorem (3.4). *Let $\{\phi^t = (\phi_\mu, \phi^t_\sigma)\}_{t\geq 1}$ have fixed bounded means and $\max_d \phi^t_\sigma(\mathbf{s}_i)_d \leq \kappa \min_d \phi^t_\sigma(\mathbf{s}_i)_d \; \forall\, i \in [N],\, t \geq 1$ for some $\kappa < \infty$. Consider $\{\mathbf{s}_i\}_{i=0}^N$, $N \geq 2$, such that $(\mathbf{s}_0, \mathbf{s}_i)$ are positive and $(\mathbf{s}_i, \mathbf{s}_j)$ are negative for all $i \neq j \in [N]$, with pairwise distinct mean embeddings. Assume that, for every $d \in [D]$, there exists $i \in [N]$ such that $\phi_\mu(\mathbf{s}_0)_d \neq \phi_\mu(\mathbf{s}_i)_d$. Then ϕ^t is eventually ϵ-distinguishable for every $\epsilon \in (0, 1/2)$ if and only if $\max_{i,d} \phi^t_\sigma(\mathbf{s}_i)_d \to 0$ and $\min_d \phi^t_\sigma(\mathbf{s}_0)_d \to \infty$.*

Proof. Let $\mathbf{S}^t_i := \operatorname{diag}(\phi^t_\sigma(\mathbf{s}_i))$ for all $0 \leq i \leq N$. Suppose first that ϕ^t is eventually ϵ-distinguishable for every $\epsilon \in (0, 1/2)$. Then, for every positive pair $(\mathbf{s}_0, \mathbf{s}_i)$ and all sufficiently large t,

$$Q^t_{0i} := \big(\phi_\mu(\mathbf{s}_0) - \phi_\mu(\mathbf{s}_i)\big)^\top (\mathbf{S}^t_0 + \mathbf{S}^t_i)^{-1} \big(\phi_\mu(\mathbf{s}_0) - \phi_\mu(\mathbf{s}_i)\big) \leq \log\big((1-\epsilon)^{-4}\big)\,, \tag{59}$$

while for every negative pair $(\mathbf{s}_i, \mathbf{s}_j)$,

$$Q^t_{ij} := \big(\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\big)^\top (\mathbf{S}^t_i + \mathbf{S}^t_j)^{-1} \big(\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\big) \geq \log\big(\epsilon^{-4}\big)\,. \tag{60}$$

Since these inequalities hold eventually for every $\epsilon \in (0, 1/2)$, and $\log((1-\epsilon)^{-4}) \to 0$ while $\log(\epsilon^{-4}) \to \infty$ as $\epsilon \to 0$, it follows that $Q^t_{0i} \to 0$ and $Q^t_{ij} \to \infty$. For any $i \in [N]$, choose $j \neq i$. Then

$$\frac{\|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2}{\min_d (\mathbf{S}^t_i)_d} \geq \sum_{d=1}^D \frac{\big(\phi_\mu(\mathbf{s}_i)_d - \phi_\mu(\mathbf{s}_j)_d\big)^2}{(\mathbf{S}^t_i + \mathbf{S}^t_j)_d} = Q^t_{ij} \to \infty. \tag{61}$$

Since the mean embeddings, $\{\phi_\mu(\mathbf{s}_i)\}_{i\geq 1}$, are pairwise distinct, the numerator is fixed, positive, and finite, and therefore $\min_d (\mathbf{S}^t_i)_d \to 0$ for every $i \in [N]$. By uniformly bounded covariance anisotropy, there exists $\kappa < \infty$ such that $\max_d (\mathbf{S}^t_i)_d \leq \kappa \min_d (\mathbf{S}^t_i)_d$, and hence $\max_{i,d} \phi^t_\sigma(\mathbf{s}_i)_d \to 0$.

Now fix $d \in [D]$. By assumption, there exists $i \in [N]$ such that $\phi_\mu(\mathbf{s}_0)_d \neq \phi_\mu(\mathbf{s}_i)_d$. Since all terms in Q^t_{0i} are nonnegative and $Q^t_{0i} \to 0$,

$$\frac{\big(\phi_\mu(\mathbf{s}_0)_d - \phi_\mu(\mathbf{s}_i)_d\big)^2}{(\mathbf{S}^t_0 + \mathbf{S}^t_i)_d} \to 0. \tag{62}$$

The numerator is fixed and strictly positive, so $(\mathbf{S}^t_0 + \mathbf{S}^t_i)_d \to \infty$. Since $(\mathbf{S}^t_i)_d \to 0$, it follows that $(\mathbf{S}^t_0)_d \to \infty$. As this holds for every $d \in [D]$, $\min_d \phi^t_\sigma(\mathbf{s}_0)_d \to \infty$.

For the other direction, assume that $\max_{i,d} \phi^t_\sigma(\mathbf{s}_i)_d \to 0$ and $\min_d \phi^t_\sigma(\mathbf{s}_0)_d \to \infty$. For every positive pair $(\mathbf{s}_0, \mathbf{s}_i)$,

$$Q^t_{0i} = \big(\phi_\mu(\mathbf{s}_0) - \phi_\mu(\mathbf{s}_i)\big)^\top (\mathbf{S}^t_0 + \mathbf{S}^t_i)^{-1} \big(\phi_\mu(\mathbf{s}_0) - \phi_\mu(\mathbf{s}_i)\big) \tag{63}$$

$$\leq \frac{\|\phi_\mu(\mathbf{s}_0) - \phi_\mu(\mathbf{s}_i)\|_2^2}{\min_d (\mathbf{S}^t_0)_d} \to 0. \tag{64}$$

Therefore, for any fixed $\epsilon \in (0, 1/2)$ and all sufficiently large t, $Q^t_{0i} \leq \log\big((1-\epsilon)^{-4}\big)$, which implies

$$P_t(Y_{0i} = 1 \mid \mathbf{s}_0, \mathbf{s}_i) \geq 1 - \epsilon. \tag{65}$$

Similarly, for every negative pair $(\mathbf{s}_i, \mathbf{s}_j)$,

$$Q^t_{ij} = \big(\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\big)^\top (\mathbf{S}^t_i + \mathbf{S}^t_j)^{-1} \big(\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\big) \tag{66}$$

$$\geq \frac{\|\phi_\mu(\mathbf{s}_i) - \phi_\mu(\mathbf{s}_j)\|_2^2}{\max_d (\mathbf{S}^t_i + \mathbf{S}^t_j)_d} \to \infty, \tag{67}$$

where the numerator is strictly positive because the mean embeddings, $\{\phi_\mu(\mathbf{s}_i)\}_{i\geq 1}$, are pairwise distinct. Hence, for all sufficiently large t, $Q^t_{ij} \geq \log\big(\epsilon^{-4}\big)$, which implies

$$P_t(Y_{ij} = 0 \mid \mathbf{s}_i, \mathbf{s}_j) \geq 1 - \epsilon. \tag{68}$$

Since the collection contains finitely many positive and negative pairs, for every fixed $\epsilon \in (0, 1/2)$ there exists a common sufficiently large t for which all pairwise distinguishability conditions hold simultaneously. Therefore, ϕ^t is eventually ϵ-distinguishable for every $\epsilon \in (0, 1/2)$. □

B Experiments

Datasets. For our experiments, we adopt the benchmark datasets introduced in prior work on the metagenomic binning task [32]. The datasets are constructed from reference genomes in GenBank and consist of viral, fungal, and bacterial sequences. The training data contains more than 2 million sequence pairs of length 1000bp. For testing, we have six datasets (*Reference 5/6*, *Plant 5/6*, and *Marine 5/6*) with species represented by highly variable numbers of sequences (10–4, 599), ranging from 2-20 kbp in length. While *Reference* datasets consist of DNA fragments from 250-330 fungal and viral genomes, *Marine* and *Plant*-associated environments contain 70k-125k sequences from roughly 180-520 species.

Baselines. KMER(COSINE) and KMER(ℓ_1) are the classical representations obtained by considering the 4-mer frequency vectors, where sequence similarity is computed using either cosine similarity or an exponential kernel over the ℓ_1 distance. HYENADNA [18] is a genome foundation model, operating at single-nucleotide resolution with context lengths up to 10^6 to efficiently capture long-range dependencies beyond the reach of standard transformers. DNABERT-2 [31] is also a foundation model that replaces fixed k-mer tokenization with Byte Pair Encoding (BPE) to improve modeling efficiency. DNABERT-S [32] leverages DNABERT-2 as a pretrained backbone, fine-tuned with contrastive objectives tailored to metagenomic binning. REVISTKMERS [5] is a lightweight model that learns sequence embeddings via a two-layer neural network applied to 4-mer profiles. It provides a strong deterministic baseline and can be viewed as the non-probabilistic counterpart of our approach. REVISTKMERS is thus also the main baseline in the experimental setup as the proposed model shares the lightweight characteristics of REVISTKMERS.

Architecture and training procedure. The training of UNCERTAINGEN is conducted within a self-supervised contrastive learning framework by optimizing the objective function provided in Eq. 2 with probabilities given as in Eq. 5. We optimized the model on a cluster equipped with NVIDIA A40 GPUs and 120GB of system memory.

The model utilizes two distinct encoder networks, ϕ_μ and ϕ_σ, which parameterize the mean vectors and diagonal covariance matrices of the multivariate Gaussian embeddings. Each encoder is implemented as a two-layer neural network designed to process input k-mer frequency profiles. Specifically, the mean and standard deviation branches both consist of a linear layer that projects the input k-mers into a 512-unit hidden space, followed by batch normalization and a sigmoid activation function. To prevent overfitting, dropout ($p = 0.2$) is applied before a final linear layer maps the features into the 256-dimensional latent space ($D = 256$). To ensure a valid probabilistic representation, we apply an element-wise exponential transformation to the output of the final linear layer in the variance encoder to guarantee a strictly positive diagonal covariance matrix.

We optimize the network parameters using the Adam optimizer with a learning rate of 10^{-2}. To enhance training stability, we adopt a sequential approach: the mean network is first trained alone for 50 epochs, followed by the variance network for an additional 25 epochs. Training is performed with a batch size of 10^5 pairs. Each positive pair is derived by splitting a single DNA fragment into two halves to ensure they originate from the same genome. We generate 200 negative pairs per positive sample via random sampling. During the initial mean-optimization stage, the success probability in Eq. 5 is parameterized as $\exp(-\|\mathbf{z}_i - \mathbf{z}_j\|_2^2/4)$ since we only optimize the mean encoder network. We additionally found it beneficial to adjust the scaling of the distance metric across optimization stages, and in the subsequent variance-optimization stage, the scaling factor in Eq. 5 is omitted.

To stabilize training, we apply a margin-style threshold only to negative pairs ($y_{ij} = 0$). When the distance term in the exponential exceeds the threshold 1, we replace its contribution by a constant, yielding zero gradient for negative pairs that are already sufficiently separated. The exact constant is insignificant, since it is used only to stop further updates from uninformative negative pairs. This follows the standard margin intuition in contrastive metric learning: negative pairs need only satisfy a separation constraint, rather than be pushed indefinitely far apart.

It is important to note that this thresholding strategy is fully aligned with the theoretical guarantees presented in Section 3 of the main paper. In our distinguishability framework (Definition 3.2), each pairwise relation must satisfy the condition $P(Y_{ij} = y_{ij} \mid \mathbf{s}_i, s_j) \geq 1 - \epsilon$. Once a negative pair becomes sufficiently separated, omitting its gradients simply indicates that the corresponding distinguishability requirement has already been fulfilled. Consequently, the optimization process

concentrates on pairs that have not yet reached the desired separation regime and is therefore compatible with the pairwise distinguishability criterion of Definition 3.2.

B.1 Toy Example with k-Mer Dataset

To evaluate the behavior of our model in a controlled setting, we designed a synthetic toy dataset of k-mer sequences. The goal of this study is twofold: (i) to verify that the model can learn meaningful low-dimensional embeddings that reflect the underlying class structure, and (ii) to assess how the model represents sequences belonging to multiple classes with their mean and variance terms.

The dataset consists of sequences generated from multinomial distributions with a clear class structure, enabling us to systematically analyze the effect of sequence overlap and class separability on the learned embeddings.

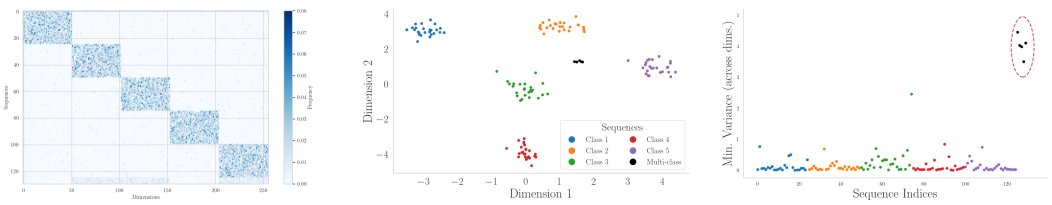
Data Generation. We generated sequences of length 100 composed of 4-mers, resulting in 256 possible k-mer types. The sequences were then divided into 5 classes, each containing 25 sequences. For each class, a multinomial distribution was defined such that the probability mass was concentrated on a distinct subset of k-mer dimensions, with a small uniform smoothing 10^{-2} to avoid zero probabilities. It ensured that sequences within the same class had similar k-mer compositions, while sequences from different classes were distinguishable (Figure 7a). To further test the model's ability to handle ambiguity, we introduced 5 "multi-class" sequences sampled by combining k-mer counts from multiple classes. This design simulates sequences that do not belong exclusively to a single class and allows us to probe how the model handles overlapping class structures.

Pair construction. We constructed positive sequence pairs by sampling two sequences from the same class, effectively assuming full access to positive examples (i.e., $y_{ij} = 1$). Negative pairs were generated using the same random sampling strategy described in Section 4, which naturally introduces the possibility of false negatives. For the multi-class sequences, we additionally constructed pairs with sequences from their contributing classes, enabling the model to learn representations that account for both pure and mixed class memberships.

Results. We learn sequence embeddings directly in a 2-dimensional latent space (Figure 7b), avoiding any dimensionality reduction steps that could distort the geometric structure of the embedding space. Visualization of the learned embeddings reveals that sequences from distinct classes form well-separated groups, while sequences belonging to multiple classes occupy intermediate regions. Importantly, our model also predicts a diagonal covariance matrix for each sequence, capturing the uncertainty associated with sequences that span multiple clusters.

Lemma 3.3 shows how the variance terms control pairwise distinguishability, while Theorem 3.4 characterizes their behavior for a sequence that must remain compatible with multiple mutually negative sequences. Consistent with this qualitative behavior, the multi-class sequences exhibit substantially larger variances in Figure 7c. This suggests that the model uses the covariance terms to represent ambiguity that cannot be captured by the mean embeddings alone.

Comparison of distance metrics for ambiguous fragments. Empirically, ambiguous fragments (labeled as "black points" in Figure 7b) lie in intermediate latent regions between clusters and are assigned significantly larger covariance (Figure 7c). While a single Gaussian distribution cannot explicitly represent multiple modes, this increased covariance adjusts the perceived distance in the embedding space. Under the standard Euclidean (ℓ_2) distance, these ambiguous points appear numerically far from the primary clusters. However, the generalized Mahalanobis distance used in



(a) k-mer Frequencies (b) Learned Embeddings (c) Variance Distribution

Figure 7: Visualization of the input k-mer features with learned embeddings and the variances.

Table 1: **Normalized average distances from ambiguous fragments to other clusters.** Values are computed by dividing the average inter-class distance by the intra-class distance of the target cluster.

Classes	Blue	Orange	Green	Red	Purple
ℓ_2-distance	80.3	10.1	9.0	87.9	12.7
Mahalanobis	10.7	1.5	2.0	11.5	1.7

UNCERTAINGEN reflects their inherent uncertainty, providing a more faithful representation. Table 1 presents the average pairwise distance between the set of ambiguous (black) fragments and each primary genomic class, normalized by the corresponding intra-class distance.

The normalization highlights the relative proximity: under Euclidean distance, ambiguous points are roughly 9.0 to 12.7 times further away from their neighboring classes than those classes' internal members are from each other. Under the Mahalanobis metric, this relative distance drops to between 1.5 and 2.0, indicating that when uncertainty is accounted for, the ambiguous fragments are recognized as being much closer to their constituent genomic distributions.

B.2 Metagenomics Binning

We evaluate our methods on the metagenomic binning task, which aims to cluster genomic sequences into species-level groups without prior knowledge of the number of clusters. To this end, we adopt the modified K-Medoid algorithm proposed by [32], which jointly infers cluster assignments and the underlying number of species. This setting is particularly challenging, as it requires learned representations to be highly discriminative while remaining resilient to noise and inaccuracies in the estimated clustering structure.

To compute pairwise similarities between sequences, we use cosine similarity for all genome-scale foundation models as well as for the KMERS(COSINE) baseline. For KMERS(ℓ_1), we employ an exponential kernel applied to the ℓ_1 distance, while REVISITKMERS uses the ℓ_2 distance (i.e., $\exp(-d_{\mathsf{X}}(\mathbf{z}_i, \mathbf{z}_j))$ where $\mathbf{z}_i$ and $\mathbf{z}_j$ are the k-mer features or sequence embeddings depending on the model, and $d_{\mathsf{X}}(\cdot, \cdot)$ is the corresponding metric). For our proposed method, we adopt an exponential kernel over the generalized Mahalanobis term defined in Eq. 5, which provides a natural similarity measure in the learned latent space.

Following established evaluation protocols, we stratify predicted clusters into five quality tiers based on their F_1 scores. High-quality bins, defined as clusters with $F_1 > 0.9$, are highlighted in dark blue

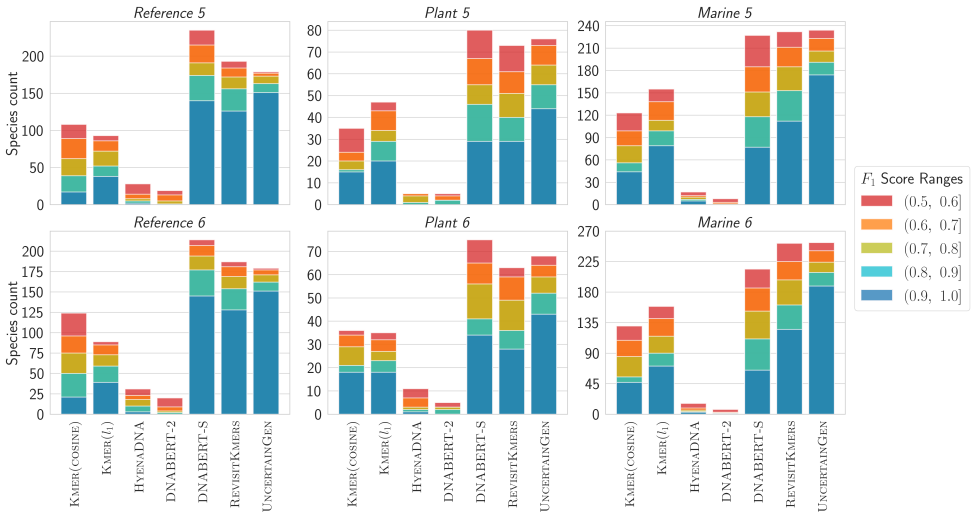


Figure 8: **Metagenomic binning results.** Cluster counts are segmented by F_1-score quality ranges. The dark blue portion highlights the highest-quality bins for each model-dataset combination.

in Figure 8. Across datasets, UNCERTAINGEN consistently outperforms its deterministic counterparts, especially the large genome foundation models.

Statistical Stability. To assess the reproducibility and reliability of our findings, we report the mean species counts and corresponding standard deviations across five independent experimental runs. The results, stratified by the F_1 quality ranges defined in Section 4.2, are summarized in Table 2. These results indicate that the performance gains achieved by UNCERTAINGEN are robust to initialization.

Table 2: Mean and standard deviation (Std) of detected species counts by UNCERTAINGEN.

		$F_1 > 0.5$	$F_1 > 0.6$	$F_1 > 0.7$	$F_1 > 0.8$	$F_1 > 0.9$
Reference 5	F_1	179	176.8	172.6	163.4	150.6
	Std	2.191	1.72	2.332	1.625	2.332
Reference 6	F_1	178.6	177	171	162.4	151.2
	Std	1.356	1.414	1.265	1.625	2.713
Plant 5	F_1	76.2	73.4	64.2	54.8	43.8
	Std	2.857	2.059	2.638	1.47	2.135
Plant 6	F_1	67.8	64	58.8	51.6	43.4
	Std	1.47	1.673	1.72	0.49	0.8
Marine 5	F_1	233.6	223	206.4	191.2	174.4
	Std	3.72	2.449	2.245	2.926	1.497
Marine 6	F_1	253.2	240.8	224.4	209.2	188.8
	Std	3.487	4.445	4.176	3.124	3.868

B.3 Ablation Studies

Distribution of variances. We begin our uncertainty analysis by inspecting the distribution of the predicted variance terms. Recall that for a given sequence $\mathbf{s} \in \mathcal{S}$, the model produces dimension-wise variance estimates $\phi_\sigma(\mathbf{s})$ corresponding to the diagonal entries of the covariance matrix. We compute

$$u(\mathbf{s}) = \sum_{d=1}^{D} \log\big(\phi_\sigma(\mathbf{s})_d + 1\big), \tag{69}$$

which is in fact the log-determinant of the covariance matrix, and we have the $+1$ term in order to ensure numerical stability. Therefore, it captures the sequence-level dispersion in the latent space.

Figure 9 shows the empirical distribution of $u(\mathbf{s})$ over sequences in the testing datasets. The multimodal distributions of $u(\mathbf{s})$ are very clear for *Reference 5/6* and *Plant 5/6* datasets. This indicates that the model partitions the sequence space into distinct regimes of predictive uncertainty, which might point out the two distinct sets of species (Please see Section 4 for the dataset details.). Importantly, the observed distributions are neither degenerate (collapsed near zero) nor uniform. Instead, they encode structured variability that reflects properties of the underlying data distribution. This finding supports the hypothesis that the variance terms carry semantically meaningful information rather than merely acting as nuisance parameters. Hence, these results establish that our model produces non-trivial and interpretable uncertainty estimates.

Sequence Filtering Across Datasets. To further probe the role of predictive uncertainty, we conduct a sequence filtering experiment in which we selectively remove sequences with the largest variance scores. Specifically, we sort sequences by their aggregated log-determinant values, $u(\mathbf{s})$ (as defined in Eq. 69,) and iteratively filter out the highest-uncertainty items. After filtering, we evaluate cluster quality by reporting the number of clusters that achieve a recall score ≥ 0.9. For comparison, we include a *random filtering* baseline in which the same number of sequences is removed uniformly at random. To ensure a fair comparison, all removed sequences are assigned to a dedicated "garbage" cluster so that they do not artificially inflate false negatives.

Figure 10 summarizes results for all the benchmarks, and we observe the consistent trends across all datasets. Filtering by predictive uncertainty consistently yields a larger number of clusters with recall ≥ 0.9 compared to the random baseline. This suggests that uncertainty-guided filtering preserves the

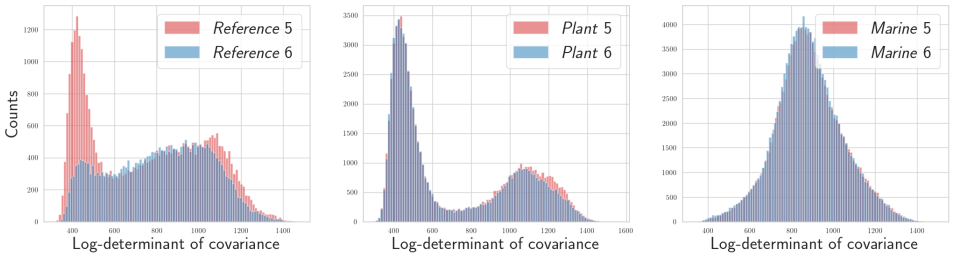


Figure 9: Distribution of the log-determinant of covariance matrices across datasets.

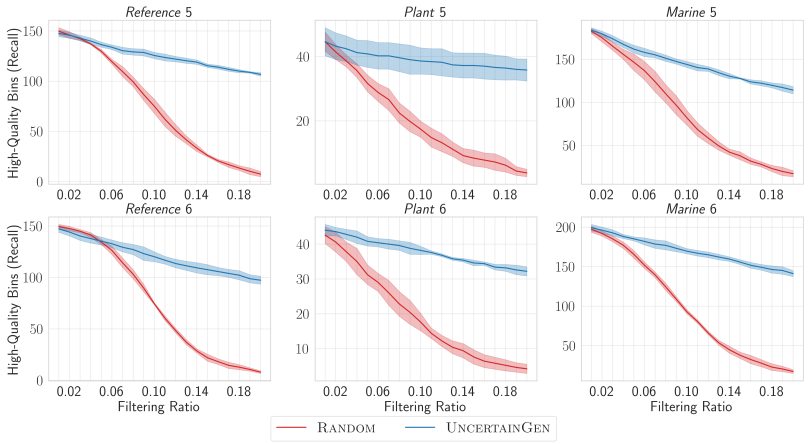


Figure 10: Filtering sequences with varying ratios over all testing datasets.

integrity of high-quality clusters while selectively removing sequences that would otherwise degrade cluster purity. The sequences assigned the highest variance values might typically originate from low-quality or noisy bins. These sequences might also tend to coincide with cases that contribute to false negatives in the unfiltered setting. By removing them, the model effectively reduces noise in the evaluation and highlights clusters that better reflect true structure in the data. Therefore, the model's variance estimates can serve as a practical mechanism for uncertainty-aware sequence selection and downstream decision making.

Effect of embedding dimension size. We analyzed the scalability of UNCERTAINGEN by evaluating the number of high-quality bins recovered across a range of latent dimension sizes, from 2^2 to 2^9, as illustrated in Figure 11. Across all six benchmarks, we observe a consistent trend where performance improves sharply as the dimensionality increases from 4 to 64. This suggests that a minimum threshold of representational capacity is required to effectively model the input sequences. Most datasets reach a peak or a stable plateau between 64 and 128 dimensions. The *Marine* benchmarks exhibit a more robust plateau, maintaining high performance even at the largest evaluated dimension size.

Computational Efficiency. Our model is intentionally lightweight, consisting of two small neural networks. In contrast to large genome foundation models, this design leads to substantial improvements in runtime efficiency. To quantitatively assess computational efficiency, we compare the number of Multiply-Accumulate operations (MACs) required by our model and several baselines. For reference, 1 MAC corresponds approximately to 2 FLOPs. All measurements are computed with `ptflops`[1] Python package. To ensure a fair comparison, all models were evaluated on a randomly generated DNA sequence of size 1000 bp. For foundation models, this was translated to the appropriate number of tokens based on their respective tokenization schemes (e.g., 1000 tokens for HYENADNA, 214 for DNABERT-2). Since REVISITKMERS and UNCERTAINGEN convert the given DNA sequence into a fixed 256-dimensional 4-mer frequency vector, the input size remains the same for these models regardless of the original input sequence length.

[1] https://github.com/LukasHedegaard/ptflops

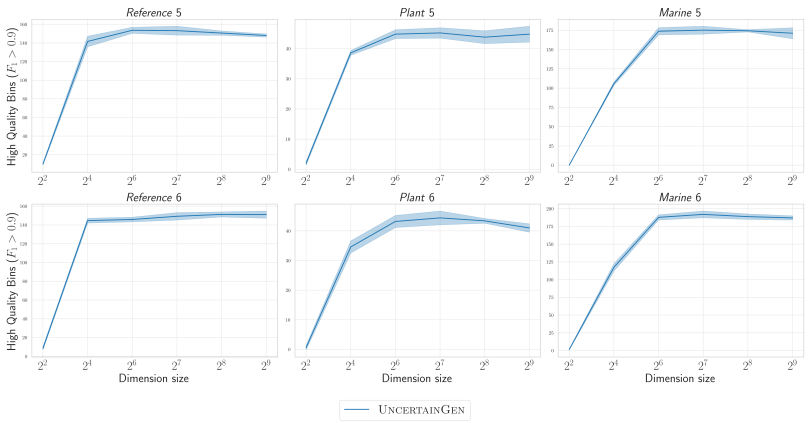


Figure 11: Impact of dimension size for different metrics.

As shown in Table 3, large genome foundation models require roughly between 6 to 25 Giga-MACs per sequence, which is over four orders of magnitude more computation than our lightweight alternatives. In contrast, both REVISITKMERS and our method, UNCERTAINGEN, achieve extremely low computational costs (placing in the roughly 10^{-4} to 10^{-3} Giga-MACs range). UNCERTAINGEN maintains this extreme efficiency while additionally modeling uncertainty in the embedding space. These results highlight that our approach provides a highly favorable trade-off between computational efficiency and representational power (Figure 2 of the main text), making it well-suited for large-scale genomic applications where both accuracy and scalability are critical.

Table 3: Comparison of computational cost in terms of Multiply-Accumulate operations (MACs) across different models. Lower values indicate higher computational efficiency.

Methods	*HyenaDNA*	*DNABERT-2*	*DNABERT-S*	*RevisitKMers*	*UncertainGen*
MACs (G)	6.55	25.92	25.12	0.26×10^{-3}	0.53×10^{-3}

C Impact Statement

This paper presents work whose primary goal is to advance machine learning methods for representation learning on genomic sequence data. We introduce an uncertainty-aware embedding framework for DNA sequences and use metagenomic binning solely as a benchmark task to evaluate the quality of the learned representations. We do not propose or deploy a full metagenomic binning pipeline, nor do we claim direct applicability to clinical or decision-critical settings.

The anticipated societal and ethical implications of this work are consistent with those commonly associated with methodological advances in machine learning and computational genomics, and we do not foresee any immediate negative impacts that require specific discussion.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the proposal of UNCERTAINGEN, its theoretical guarantees on distinguishability, and its performance evaluation on metagenomic binning datasets

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations in Section 5, specifically regarding the self-supervised pairing strategy, the use of two-layer networks, and the lack of calibration for variance estimates.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Section 3 provides the main theoretical statements (Lemma 3.1, 3.3, and Theorem 3.4), and full formal proofs including all necessary assumptions are provided in Appendix A.2.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed training procedures in Section 4 and architectural specifics in Appendix B, including optimizer choice, learning rates, and batch sizes.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide our implementation in the supplemental material and release the project' GitHub repository `https://github.com/abdcelikkanat/uncertaingen`. Additionally, the datasets utilized are publicly accessible via the DNABERT-S repository: `https://github.com/MAGICS-LAB/DNABERT_S`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: These details are thoroughly specified in Section 4 and further expanded in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper provides the mean and standard deviation of species counts across five independent experimental runs in Table 2 of Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: The paper discusses model size and efficiency relative to foundation models (Figure 2), and provides the hardware specifications for model training in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involves standard genomic data analysis and follows ethical research guidelines for computational biology.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: The paper focuses primarily on the technical and theoretical aspects of representation learning for genomics, and a dedicated broader impacts discussion is included in Appendix C.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The developed model for DNA sequence representation in metagenomic binning does not pose the high risks associated with the dual-use or misuse typical of large generative models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing assets, including datasets and baseline models like DNABERT-S and HyenaDNA, are cited appropriately in the References.

    Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper proposes a new method/architecture rather than introducing a new dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve human subjects or crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The research is entirely computational and based on public genomic data; no human subjects were involved.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We only used the LLMs for text editing purposes (e.g., grammar, spelling, word choice) as we also indicated on the paper submission platform.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.