VQDNA: Unleashing the Power of Vector Quantization for Multi-Species Genomic Sequence Modeling

Siyuan Li [*1 2] **Zedong Wang** [*1] **Zicheng Liu** [1 2] **Di Wu** [1 2] **Cheng Tan** [1 2] **Jiangbin Zheng** [1 2]
Yufei Huang [1 2] **Stan Z. Li** [† 1]

Abstract

Similar to natural language models, pre-trained genome language models are proposed to capture the underlying intricacies within genomes with unsupervised sequence modeling. They have become essential tools for researchers and practitioners in biology. However, the *hand-crafted* tokenization policies used in these models may not encode the most discriminative patterns from the limited vocabulary of genomic data. In this paper, we introduce VQDNA, a general-purpose framework that renovates genome tokenization from the perspective of genome vocabulary learning. By leveraging vector-quantized codebook as *learnable* vocabulary, VQDNA can adaptively tokenize genomes into *pattern-aware* embeddings in an end-to-end manner. To further push its limits, we propose Hierarchical Residual Quantization (HRQ), where varying scales of codebooks are designed in a hierarchy to enrich the genome vocabulary in a coarse-to-fine manner. Extensive experiments on 32 genome datasets demonstrate VQDNA's superiority and favorable parameter efficiency compared to existing genome language models. Notably, empirical analysis of SARS-CoV-2 mutations reveals the fine-grained pattern awareness and biological significance of learned HRQ vocabulary, highlighting its untapped potential for broader applications in genomics.

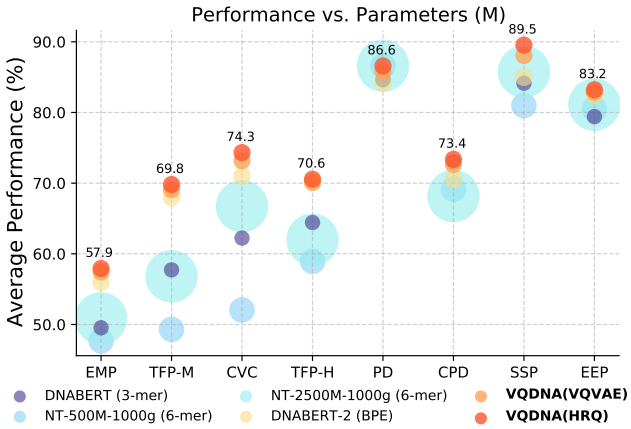


Figure 1. Performance of fine-tuned VQDNA and other genome language models across downstream tasks on 32 datasets, including Epigenetic Mark Prediction (EMP) for Yeast, Transcription Factor Prediction on mouse and human genome (TFP-M and TFP-H), Covid Variants Classification (CVC), Promoter Detection (PD), Core Promoter Detection (CPD), Splice Site Prediction (SSP), and Editing Efficiency Prediction (EEP). The circle size indicates the parameter scale of each model. Notably, NT-2500M-1000g is with 2537M model parameters, while our VQDNA has only 110M.

1. Introduction

Genomics, which refers to the study of genomes-the complete set of DNA instructions within an organism, enables scientists to delve into the molecular machinery of life (Yang et al., 2011; Moore et al., 2020). It provides critical insights into genetic coding and expression that orchestrate the development, functioning, and reproduction of living organisms, thereby prompting a paradigm shift in biological discovery, unlocking mysteries of multifactorial traits, genetic diseases, and evolution (Locke et al., 2015; Visscher et al., 2017; Andersson & Sandelin, 2020). By leveraging deep learning techniques, breakthroughs in genomics have burst onto the scene, showcasing their preeminence in addressing a broad spectrum of biological applications, such as splicing regulation and gene expression prediction (Kelley et al., 2015; Zhou & Troyanskaya, 2015; Žiga Avsec et al., 2021), DNA methylation prediction (Vidaki et al., 2017; Angermueller et al., 2017), chromatin accessibility (Min et al., 2017), promoter prediction (Lai et al., 2019; Le et al., 2022) and more.

[*]Equal contribution [1]AI Lab, Research Center for Industries of the Future, Westlake University, Hangzhou, 310024, China [2]College of Computer Science and Technology, Zhejiang University, Hangzhou, 310058, China. Correspondence to: Stan Z. Li <stan.z.li@westlake.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

In parallel, large-scale genomic data has been readily accumulated, which presents the opportunity, differing from specialized advancements, for digging out generalizable patterns that can be directly fine-tuned for various downstream tasks. Drawing inspiration from the success of natural language models (Devlin et al., 2019; Brown et al., 2020; Ouyang et al., 2022), genome language models have been introduced by representing genomes as languages for unsupervised genomic sequence modeling. DNABERT (Ji et al., 2021) first explores the language model-style pre-training on the human genome. Nucleotide Transformers (Dalla-Torre et al., 2023) is pre-trained on massive multi-species genomes, increasing cross-species diversity. HyenaDNA (Nguyen et al., 2023) targets the unique but challenging extra-long sequence issue and strikes remarkable accuracy-efficiency trade-offs. Very recently, DNABERT-2 (Zhou et al., 2024) pioneered the use of Byte Pair Encoding (BPE) (Sennrich et al., 2015) to iteratively merge the co-occurring nucleotides that might be relevant in genomics.

Along this line, tokenization has become an integral part of genome language models, significantly influencing the model's perception and interpretation of genomes (Zhou et al., 2024). The commonly used k-mer combines adjacent sets of k-length nucleotide bases through a sliding window of specified strides. BPE, however, iteratively merges the statistically most co-occurring segments regardless of the topological distance. Although more precise tokenization strategies have been introduced, these *hand-crafted* methods may not represent sufficient information from the limited vocabulary (A, T, C, and G, four nucleotide bases) of genomes and thus cannot guarantee the derived word embeddings encoding the most discriminative genomic patterns (Chelba et al., 2017). Thus, merging genome segments solely according to *hand-crafted* policies might mislead the training into subpar representations, resulting in inevitable sample inefficiency and non-generalizability. In this paper, we argue that if we can derive a *learnable* genome vocabulary that records the most discriminative patterns from input genomes, we can thus use it as an off-the-shelf weapon to tokenize genomes into pattern-aware embeddings for subsequent pre-training.

To this end, we reconstruct the genome tokenization into a **discriminative genome vocabulary learning** problem and propose VQDNA, a novel framework eschewing *hand-crafted* schemes and instead relying entirely on the VQ-VAE (Van Den Oord et al., 2017) tokenizer, which computes *pattern-aware* embeddings with the VQ codebook as *online-optimizable* genome vocabulary. Built upon this concept, we further conjecture that the limited original vocabulary of genomes may conceivably hamper discriminative codebook learning, resulting in the loss of fine-grained details trapped in the four nucleotides. To further push the limits of the VQ tokenizer, we present Hierarchical Residual Quantization (HRQ), where varying scales of codebooks are designed in a hierarchical structure with coarse-grained semantics concentrated in the lower layers and fine-grained details in the higher layers to expand the vocabulary for perceptually rich codebook learning in a coarse-to-fine progressive manner.

We comprehensively evaluate the effectiveness of VQDNA on GUE benchmark (Zhou et al., 2024) with 28 datasets and 4 additional genome datasets as illustrated in Figure 1 and Sec. 4.2 involving the input sequence lengths from 63 up to 32k. To further validate our methods on the unique but meaningful extra-long sequence issue, we extend the input length of VQDNA (HRQ) to a maximum of 32k, allowing fair comparisons with HyenaDNA in Species Classification (SC) tasks. Extensive experiments show that our VQDNA, as a general-purpose framework for multi-species genomic sequence modeling, can handle large and diverse genome analysis tasks and hits state-of-the-art across 32 datasets of varying input lengths while striking favorable complexity-accuracy trade-offs. More importantly, empirical analysis of the SARS-CoV-2 demonstrates the **fine-grained pattern-awareness** and **biological significance** of HRQ vocabulary, revealing its potential for broader applications in biology.

Our contributions can thus be summarized as follows:

- We push the boundaries of genome tokenization from the fresh perspective of genome vocabulary learning, presenting the VQDNA framework to learn a VQ codebook as discriminative genome vocabulary for pattern-aware genome language tokenization in an end-to-end manner.
- An HRQ tokenizer is designed to progressively enrich the originally limited genome vocabulary with a hierarchy of varying scales of codebooks in a coarse-to-fine manner. This hierarchical design delivers performance on par with the state-of-the-art models while using fewer parameters.
- Extensive experiments across 32 datasets verify the exceptional generalizability of VQDNA. Empirical study on SARS-CoV-2 mutations shows the biological significance and potential of VQDNA among existing models.

2. Related Work

2.1. Pre-trained Genome Language Models

Genomics has witnessed rapid advances in recent decades thanks to the emergence of new technologies that facilitate high-throughput DNA sequencing. This precipitous drop in the cost and time has led to an explosion of genomic data. The Human Genome Project and the 1000 Genomes Project (Byrska-Bishop et al., 2022) have successfully sequenced thousands of individual genomes, identifying millions of genetic variants. In addition to the human genome, genomes from other organisms have also been extensively sequenced and analyzed. The abundance of data provides the opportunity to explore pre-trained language models in

genomics that can be adapted to various downstream tasks.

DNABERT (Ji et al., 2021) introduces the first pre-trained genome language model based on BERT (Devlin et al., 2019) architecture. They pre-train the BERT Transformer solely on human genome to develop a general understanding of DNA and then fine-tune the models on task-specific datasets, including Eukaryotic Promoter Database for promoters, ENCODE ChIP-seq for TF binding sites, and more. Techniques are adjusted to suit the DNA characteristics, such as the masking scheme and next-sentence prediction. Similar to natural language models, tokenization is critical in the model's perception and interpretation of genomes. They utilize overlapping k-mer to incorporate contextual information from genomes in tokenization. Despite its shortcomings, DNABERT has inspired almost all the subsequent genome language models as a ground-breaker. Nucleotide Transformer (Dalla-Torre et al., 2023) proposes a new family of transformer-based genome language models. These models, ranging from 500M up to 2.5B parameters, are pre-trained on the human reference genome, 3,202 genetically diverse human genomes, and 850 multi-species genomes. It is a huge leap in terms of the volume and diversity of training data, directly leading to superior performance. As for tokenization, they first use non-overlapping k-mer instead of the overlapping version in DNABERT. They empirically show that tokenizing DNA into different mers exerts quite diverse performance which highlights the value of genome tokenization. Moreover, empirical analyses like attention maps confirm the learned representations can reconstruct human genetic variants and distinguish between key genomic elements like exons, promoters, and enhancers.

The newly emerged DNABERT-2 (Zhou et al., 2024) systematically discusses current genome tokenization techniques and first adapts SentencePiece (Kudo & Richardson, 2018) with BPE to tokenize genome sequences. They also integrate Attention with Linear Biases (ALiBi) (Press et al., 2021), FlashAttention (Dao et al., 2022), LoRA fine-tuning (Hu et al., 2021) and more practical techniques to overcome the architectural limitations of existing genome language models. Additionally, several genomic benchmarks are published (Fishman et al., 2023; Nguyen et al., 2023). MUSE (Marin et al., 2024) and Genome Understanding Evaluation (GUE) (Zhou et al., 2024) provide multi-species genome analysis with well-calibrated data separation, task setting, and evaluation metrics, resolving the lack of standard benchmarks for existing genome language models.

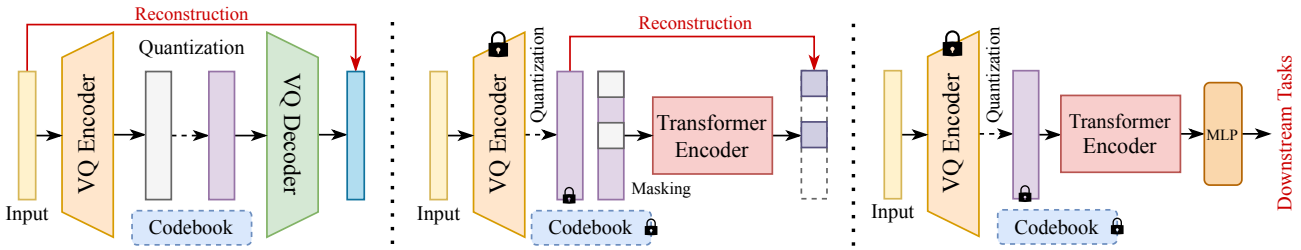
2.2. Vector Quantization

First pioneered in image compression (Gray, 1984), vector quantization (VQ) as a parametric method has demonstrated tremendous success in generating high-fidelity patterns by discretizing latent space. Holistically, VQ quantizes the continuous latent from the encoder into discrete vectors by replacing them with the closest embeddings from the *learnable* codebook. VQ-VAE (Van Den Oord et al., 2017) as the cornerstone first introduces a vector-quantized learning framework comprising training and generation phases. It encodes image pixels into latent features and then searches for the nearest token to each corresponding feature vector. The image is thereby reconstructed through the decoder. During training, an annealing procedure is employed to guide the quantization, helping avoid posterior collapse issues softly. Thereafter, a multi-scale variant (Razavi et al., 2019) is proposed. Dhariwal et al. (2020) adds random restart policy to avoid codebook collapse. VQGAN (Esser et al., 2021) leverages GPT-2 as the generator and employs adversarial loss and feature-level perceptual losses in the training stage, which shows improved reconstruction quality over VQ-VAE. As such, VQGAN-based variants have been adapted to video and more scenarios (Yu et al., 2023a;c). MaskGIT (Chang et al., 2022) proposes a new paradigm where masked tokens are predicted by attending to tokens from all directions. RQ (Lee et al., 2022) refines the latent feature by quantized residuals, and Huh et al. (2023a) examines critical challenges in VQ training. MAGE (Li et al., 2023b) predicts randomly masked VQ tokens in the latent space (Li et al., 2023a) that first combine both self-supervised pre-training (He et al., 2022; Li et al., 2024a) and image generation into one framework. LQAE (Liu et al., 2023) tokenizes the input into lexical representations with frozen BERT word embeddings. Recently, FSQ (Mentzer et al., 2023) boosts the quantization efficiency with finite-scalar implicit codebook. VQ techniques are also adopted in AI4S methods (Su et al., 2024; Wu et al., 2024b;a) to model both sequential and structural information.

As yet, the quantized posterior has proven effective in unleashing the full expressivity of complex multi-modal distributions such as images and videos. To the best of our knowledge, however, there is no such attempt to leverage VQ for genome language models. In the following sections, we will first incorporate VQ into genome tokenization in our proposed three-stage VQDNA framework. Next, we describe HRQ vocabulary learning architecture and discuss its advantages with extensive experimental results.

3. Methodology

This paper aims to develop a general-purpose framework by leveraging VQ codebook as *learnable* genome vocabulary that can adaptively tokenize inputs into *pattern-aware* word embeddings for genomic sequence modeling to serve multiple downstream tasks. The core idea behind this is to learn a discriminative genome vocabulary consisting of discrete code embeddings that can then get assigned to corresponding latent features via a nearest-neighbor lookup for



(a) Stage-1: DNA tokenization (b) Stage-2: Masked Code Modeling (c) Stage-3: Fine-tuning with LoRA

Figure 2. An overview of our three-stage training pipeline of VQDNA. (a) VQ genome vocabulary learning with large-scale multi-species genome sequences. (b) Masked modeling pre-training of the Transformer encoder with frozen genome vocabulary. (c) Fine-tuning the pre-trained encoder with an MLP head for various downstream genome analysis tasks.

genome tokenization. By optimizing this vocabulary to minimize quantization objectives, the codebook embeddings can essentially represent a dictionary of pattern-aware data clusters learned in a completely self-supervised paradigm.

In this section, we introduce our three-stage VQDNA training framework, as shown in Figure 2, to multi-species genomic sequence modeling. We first propose to incorporate the renowned VQ-VAE (Van Den Oord et al., 2017) instead of *hand-crafted* methods into tokenization for *pattern-aware* genome vocabulary learning, as illustrated in Sec. 3.1. Moreover, we further conjecture that the limited vocabulary of genome sequences may conceivably hamper discriminative codebook learning, resulting in the loss of fine-grained patterns trapped in the original four nucleotides. To tackle this problem, we propose hierarchical residual quantization (HRQ) in Sec. 3.2, to progressively enrich the genome vocabulary with a hierarchy of varying scales of codebooks in a coarse-to-fine manner. In Sec. 3.3, we describe the implementation details for VQDNA vocabulary learning.

3.1. Vector-Quantized Genome Vocabulary Learning

As aforementioned, we first parameterize the tokenization as a genome vocabulary learning problem and follow the VQ-VAE to take sequence reconstruction as pre-training objectives to concurrently optimize the codebook and the VQ encoder. We present this as the base version of VQDNA.

Given the input genome sequence $X \in \mathbb{R}^{L\times d}$, an encoder $E_\theta(\cdot)$ with parameters θ maps X into the latent space as $Z = E_\theta(X) \in \mathbb{R}^{L\times D}$. With a finite vocabulary of K key-value pairs as the VQ codebook, $\mathcal{C} = \{(k, e(k))\}_{k\in[K]}$, where each code (index) k owns its *learnable* code embedding vector $e(k) \in \mathbb{R}^D$, the representation Z can be quantized by the element-wise code mapping function $\mathcal{Q}(\cdot,\cdot)$:

$$M_i = \mathcal{Q}(Z_i;\mathcal{C}) = \text{argmin}_{k\in[K]}\|Z_i - e(k)\|_2, \quad (1)$$

where $1 \le i \le L$, $M \in [K]^L$ denotes code mapping indices. Thus, the latent Z_i can be indexed and quantized into discrete genome embeddings by the distance-wise closest 1-of-K embedding vectors within codebook $\mathcal{C}$ with assigned code M_i as $\hat{Z}_i = e(M_i)$. The decoder $G_\phi(\cdot)$ with parameters ϕ then maps the quantized embedding $\hat{Z}$ back to the input genome sequence space to reconstruct $\hat{X}$:

$$\hat{X} = G_\phi(\hat{Z}) = G_\phi(e(M)), \quad (2)$$

As differentiation through the quantization is ill-posed, the straight-through-estimator (STE) (Bengio et al., 2013) is employed as gradient approximation during backward computation. To optimize the overall framework, the overarching models aim to minimize the VQ-VAE loss $\mathcal{L}_{\text{VQ}}$:

$$\mathcal{L}_{\text{VQ}} = \underbrace{\mathcal{L}_{CE}(X,\hat{X})}_{\mathcal{L}_{\text{rec}}} + \underbrace{\|\text{sg}[Z] - \hat{Z}\|_2^2}_{\mathcal{L}_{\text{code}}} + \beta \underbrace{\|Z - \text{sg}[\hat{Z}]\|_2^2}_{\mathcal{L}_{\text{commit}}}, \quad (3)$$

where $\text{sg}[\cdot]$ refers to the aforementioned stop-gradient operator, and $\beta \in [0, 1]$ is a trade-off hyper-parameter (default to 0.5). Notably, the first term $\mathcal{L}_{\text{rec}}$ denotes the reconstruction loss to optimize the encoder and decoder in VQ-VAE vocabulary learning (Stage-1 in Figure 2). The middle term $\mathcal{L}_{\text{code}}$ takes a squared error as the codebook loss to update code embeddings by pushing embedding vectors toward the encoder outputs. The third term $\mathcal{L}_{\text{commit}}$ is a commitment loss, which ensures the training stability of code mapping $\mathcal{Q}(\cdot,\cdot)$. In this paper, we optimize the codebook $\mathcal{C}$ with the exponential moving average (EMA) of embeddings instead of the loss $\mathcal{L}_{\text{code}}$:

$$\hat{Z}_i = (1-\alpha)Z_i + \alpha\hat{Z}_i, \quad (4)$$

where α is the momentum coefficient. The EMA update of the codebook in Eq. (4) can reduce the training instability caused by updating conflicts of the certain code from latent tokens of different subjects (Razavi et al., 2019).

After obtaining the learned codebook $\mathcal{C}$, we can reuse it as an off-the-shelf genome vocabulary to tokenize genome sequences into *pattern-aware* genome embeddings for language model pre-training. Subsequently, we can store the tokenized data for stage-2 pre-training (described in Appendix A) and conduct the same masked pre-training and downstream fine-tuning as DNABERT-2 for our VQDNA with the trained VQ-VAE vocabulary (described in Sec. 4).

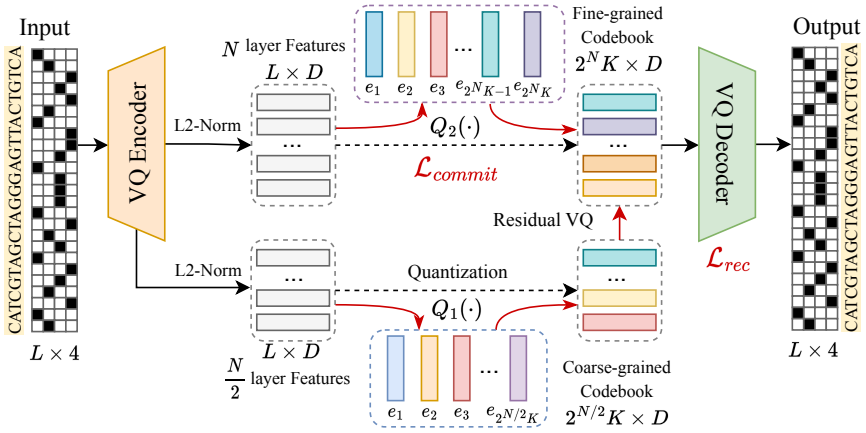


Figure 3. Illustration of our Hierarchical Residual Quantization (HRQ) as genome word embedding for VQDNA framework. We instantiate HRQ with a 6-layer encoder and decoder with two hierarchical codebooks after the output of 3-th and 6-th layers in practice.

Method	Tokenizer	Usage	Lin.	FT
DNABERT	6-mer	47	23.54	55.50
NT-2500M-1000g	6-mer (non)	47	23.54	66.73
HyenaDNA	one-hot	100	5.47	54.10
DNABERT-2	BPE (6-mer)	99	36.53	71.02
VQDNA	**VQVAE**	100	<u>44.76</u>	<u>73.16</u>
VQDNA	**HRQ**	100	**48.87**	**74.32**

Table 1. Analysis of tokenization efficiency. We report the tokenizer types, token usage (%), macro F1-score (%) of linear probing (Lin.), and fully fine-tuning (FT) for the Covid Variants Classification task, which is illustrated in Sec. 4.3. Note that 6-mer (non) utilizes non-overlapping 6-mer tokenization, and BPE (6-mer) iteratively merges the most concurrent codes in 6-mer tokenization. The token usage is the percentage of the used token in total for each tokenizer.

Now, we have reformulated genome tokenization from the perspective of VQ-VAE genome vocabulary learning with apparent benefits: **(i)** The sequence nature of genomic data comfortably fits the VQ computations. Nucleotide base pairs within genomes can not only form localized motifs like promoter elements but also modulate global chromatin states, which is much akin to that of image pixels in vision, where VQ has already established its dominance. The quantized posterior has proven effective in compressing intricate multi-modal distributions, making it well-primed to encode the most discriminative genomic patterns unshackled by *hand-crafted* rules and biases. **(ii)** Genomic context plays a vital role in genome analysis tasks. Contrary to existing tokenization methods that solely concern better merging intra-sequence nucleotides, VQ tokenizer naturally records the genomic context by incorporating whole inputs into its codebook optimization implicitly rather than just regarding the intra-sequence dependencies. Empirical analysis in Sec. 4.4 demonstrates both the intra- and inter-class pattern-awareness of VQ. The rest of this section expands on HRQ to further push the limits of genome vocabulary learning.

3.2. Hierarchical Residual Quantization

Although the VQ-VAE tokenizer can provide tangible benefits above, it expands its power primarily by enlarging the codebook size. To take a further step, however, simply expanding the codebook size is inefficient due to the codebook collapse problem, and more importantly, it may not be compatible with the nature of genomic data. Genomic data is essentially sequences consisting of four potential nucleotide bases, A, T, C, and G, at each site, which means the original vocabulary of genomes is much restricted compared to that of other modalities, such as images and natural languages. Through the lens of VQ tokenizer, such a limited vocabulary space might be too coarse-grained to present sufficient details for perceptually rich codebook learning. Therefore, we argue that it is necessary to design a specified protocol to disentangle such underlying intricacies within the restricted nucleotides for discriminative genome vocabulary learning.

Motivated by the success of multi-scale perception (Wang et al., 2020) in visual recognition, it is appealing that we can also transfer this success from computer vision to genomics, *i.e.*, to build varying scales of codebooks as multi-grained genome vocabulary and then tokenize different layers of inputs with corresponding vocabulary, which can be hierarchically aligned via residual techniques (Lee et al., 2022). To achieve this, we propose Hierarchical Residual Quantization (HRQ), where a hierarchy of codebooks is designed to expand the genome vocabulary in a coarse-to-fine manner.

As shown in Figure 3, the multiple scales of codebooks are designed in a hierarchical architecture with coarse-grained semantics concentrated in the lower layers and fine-grained details in the higher layers. Quantization is performed sequentially from encoder layer 1 to N. Given the hierarchical input $H^{(n)} \in \mathbb{R}^{L \times D}$ out of encoder layer n, a corresponding $2^n \cdot K$-size codebook $\mathcal{C}^{(n)} = \{(k^{(n)}, e(k^{(n)}))\}_{k \in [2^n K]}$ with each code embedding vector $e(k^{(n)}) \in \mathbb{R}^D$ is defined. Thus, each representation $H^{(n)}$ is quantized by the same code mapping operator $\mathcal{Q}(\cdot, \cdot)$ in Eq. (1):

$$\begin{aligned} M_i^{(n)} &= \mathcal{Q}(H_i^{(n)}; \mathcal{C}^{(n)}) \\ &= \text{argmin}_{k \in [2^n K]} \|H_i^{(n)} - e(k^{(n)})\|_2, \end{aligned} \tag{5}$$

where $1 \le i \le L$, $M^{(n)} \in [2^n K]^L$ indicates the HRQ code mapping indices of $H^{(n)}$. As such, we derive a hierarchy of codebooks with varying perceptual granularities for hierarchical genome tokenization in a coarse-to-fine manner. With assigned $M_i^{(n)}$, the latent features of layer n can be quantized as $\hat{H}_i^{(n)} = e(M_i^{(n)})$. However, one remaining challenge is that, given the output $Z^{(n)} \in \mathbb{R}^{L \times D}$ from en-

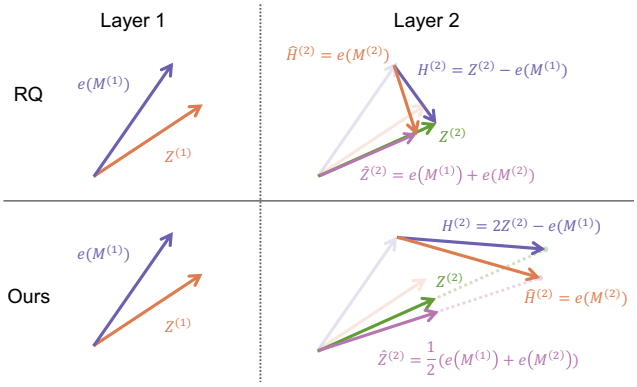


Figure 4. Illustration of RQ and our HRQ in a two-dimensional space with a two-layer quantization case. We use purple for the current hierarchical input $H^{(n)}$(or the residual in RQ), green for the second layer encoder output $Z^{(2)}$, orange for the input $Z^{(1)}$ and output hierarchical embeddings $\hat{H}^{(n)}$ per layer, and pale orchid for the ultimate embeddings $\hat{Z}^{(n)}$ after n-layer quantization.

coder layer n, how to associate the hierarchical input $H^{(n)}$ in Eq. (5) with $Z^{(n)}$ to form a unified HRQ architecture.

Although Lee et al. (2022) first introduces residual quantization (RQ) to harness the training of multiple codebooks, their method is essentially designed for recursive quantization with a single input, which has not addressed the above issue of multiple inputs. To resolve this problem, we define a strategy to associate $H_i^{(n)}$ with $Z_i^{(n)}$ formalized as:

$$H_i^{(n)} = \begin{cases} 2Z_i^{(n)} - e(M_i^{(n-1)}) & \text{for } n = 2, \cdots, N, \\ e(M_i^{(1)}) & \text{otherwise} \end{cases} \tag{6}$$

where $1 \le n \le N$, and $1 \le i \le L$. Starting with the initial quantization $H_i^{(1)} = e(M_i^{(1)})$, our HRQ calculates the code mapping $M^{(n)}$ in Eq. (5), which together with $Z_i^{(n+1)}$ yields the hierarchical input $H_i^{(n+1)}$ for next layer quantization. The motivation behind this is to resolve the alignment between $H_i^{(n)}$ and $Z_i^{(n)}$ while maintaining the scale consistency across HRQ layers, as this property has proven essential in ensuring better utilization of multiple codebooks (Yu et al., 2023b). As shown in Figure 4, we compare our proposed strategy with renowned RQ. Intuitively, representations computed by doubled inputs residual exhibit more favorable scale consistency across the layers.

Along this line, we obtain a hierarchy of learned codebooks. We can thereby use them as an off-the-shelf genome vocabulary to tokenize input genomes into a collection of hierarchical embeddings after N layers of quantization $\mathcal{HRQ}(\cdot,\cdot,\cdot)$:

$$\mathcal{HRQ}(Z_i, \mathcal{C}, N) = (\hat{H}_i^{(1)}, \cdots, \hat{H}_i^{(N)}), \tag{7}$$

where each $\hat{H}_i^{(n)} = e(M_i^{(n)}) \in \mathbb{R}^{L \times D}$ denotes the quantized genome embedding at layer n. As illustrated in Figure 4, we define the ultimate output embeddings of HRQ as $\hat{Z}_i = \frac{1}{N}(\sum_{n=1}^{N} \hat{H}_i^{(n)})$, which sums the hierarchical embeddings $\hat{H}^{(n)}$ from all N quantization layers in average for scale consistency. With the exponentially growing codebooks, the HRQ vocabulary can progressively capture the most discriminative coarse-grained semantics and the fine-grained details within input genome sequences for discriminative tokenization and subsequent masked pre-training.

Training of HRQ The overall learning objective of our proposed HRQ is defined as follows:

$$\mathcal{L}_{\mathcal{HRQ}} = \underbrace{\mathcal{L}_{CE}(X, \hat{X})}_{\mathcal{L}_{\text{rec}}} + \beta \underbrace{\sum_{n=1}^{N} \|Z^{(n)} - \text{sg}[\hat{Z}^{(n)}]\|_2^2}_{\mathcal{L}_{\text{commit}}}, \tag{8}$$

where $\beta > 0$ is the same hyper-parameter as in Eq. (3), and the first term is the reconstruction loss $\mathcal{L}_{\text{rec}}$. We also employ the widely-used EMA of the clustered embeddings to update codebook $\mathcal{C}$ instead of the codebook loss $\mathcal{L}_{\text{code}}$ in Eq. (3). The commitment loss $\mathcal{L}_{\text{commit}}$ in Eq. (8) is defined as the sum of squared errors from each layer n, which is different from VQ-VAE. It aims to make the quantized embeddings $\hat{Z}^{(n)}$ progressively reduce the squared error as n increases. In such a way, HRQ disentangles the underlying semantics in the limited genome vocabulary for perceptually rich codebook learning in a hierarchically coarse-to-fine manner. Empirical studies in Sec. 4.4 and Appendix B demonstrate the fine-grained pattern-awareness of the HRQ vocabulary.

3.3. Implementation Details

We adopt the network architecture of ConvNeXt variants (Liu et al., 2022; Li et al., 2024b) for our tokenizers, which have Transformer-like macro designs but are more efficient. The encoder network for VQVAE and HRQ consists of a stem module and 6 residual blocks, *i.e.*, N=6, and D=384. The stem projects the input data (one-hot encoded) to 256 dimensions by a 1D convolution layer with a kernel size of 5 and a stride of 1, followed by a LayerNorm (Ba et al., 2016) and GELU activation. Each residual block contains a 1D depth-wise convolution layer (the kernel size of 7) and 2 full-connected layers to form the inverted bottleneck (Sandler et al., 2018) (expanding 4 times). The architecture of the de-tokenizer (the decoder of the VQDNA tokenizer) is symmetrical to the tokenizer in Figure 3, except for us-

Table 2. Average performance ranking, tokenizer types, model parameters and FLOPs, and pre-training tokens on 32 genome downstream tasks.

Method	Date	Tokenizer	# Params. (M)	FLOPs (G)	Train (B)	Average Rank
DNABERT	BioInfo'2021	3-mer	86	3.3	122	5
NT-500M	biorxiv'2023	6-mer	480	3.2	50	6
NT-2500M	biorxiv'2023	6-mer	2537	19.4	300	4
DNABERT-2	ICLR'2024	BPE	117	1.0	262	3
VQDNA	**Ours**	**VQVAE**	86+16	1.1+0.5	262	<u>2</u>
VQDNA	**Ours**	**HRQ**	86+17	1.1+0.6	262	**1**

Table 3. MCC (in %) performance of Promoter Detection (PD), Core Promoter Detection (CPD), and Transcription Factor Prediction (TFP) tasks fine-tuned on GUE benchmarks.

Method	PD			CPD			TFP (Human)				
	all	notata	tata	all	notata	tata	0	1	2	3	4
DNABERT (3-mer)	90.44	93.61	69.83	70.92	69.82	78.15	67.95	70.90	60.51	53.03	69.76
NT-500M-1000g (6-mer)	89.76	91.75	78.23	66.70	67.17	73.52	63.64	70.17	52.73	45.24	62.82
NT-2500M-1000g (6-mer)	90.95	93.07	75.80	67.39	67.46	69.66	66.31	68.30	58.70	49.08	67.59
DNABERT-2 (BPE)	86.77	94.27	71.59	69.37	68.04	74.17	71.99	76.06	66.52	58.54	77.43
VQDNA	90.20	94.05	73.08	70.36	69.87	77.63	72.04	75.89	66.69	58.31	77.63
VQDNA (HRQ)	**90.75**	**94.48**	**74.52**	**71.02**	**70.58**	**78.50**	**72.48**	**76.43**	**66.85**	**58.92**	**78.10**

Table 4. Performance of Transcription Factor Prediction (TFP), Covid Variants Classification (CVC), Splice Site Prediction (SSP), and Editing Efficiency Prediction (EEP) tasks. TFP and SSP use MCC (%), while CVS and EEP report F1 (%) and MCC (%).

Method	TFP (Mouse)					CVC	SSP	EEP (gRNA)		
	0	1	2	3	4	Covid	Reconstruction	K562	Jurkat	H1
DNABERT (3-mer)	42.31	79.10	69.90	55.40	41.97	62.23	84.14	88.63	86.89	62.72
NT-500M-1000g (6-mer)	39.26	75.49	64.70	33.07	34.01	52.06	80.97	90.58	88.94	63.80
NT-2500M-1000g (6-mer)	48.31	80.02	70.14	42.25	43.40	66.73	85.78	90.90	89.34	66.87
DNABERT-2 (BPE)	56.76	84.77	79.32	66.47	52.66	71.02	84.99	91.02	89.27	66.91
VQDNA	57.52	85.36	79.78	68.45	54.10	73.16	88.06	91.16	89.83	67.56
VQDNA (HRQ)	**58.34**	**85.81**	**80.39**	**69.72**	**54.73**	**74.32**	**89.53**	**91.53**	**90.12**	**67.98**

ing 1D de-convolution layers instead. The output sequence length of the tokenizer is the same as the input. The standard VQVAE vocabulary learning uses a codebook size of 512, while the HRQ version uses the size of 384. In practice, we instantiate the HRQ decoder only with the 3-th and 6-th layers codebooks. The masked pre-training and downstream adaptation details are described in Sec. 4.

4. Experiments

4.1. Experimental Setup

In pre-training stages, we follow the pre-training recipes in DNABERT-2 (Zhou et al., 2024) that pre-training the VQ tokenizer and BERT-Base Transformer encoder on the human genome (Ji et al., 2021) with 2.75B nucleotide bases and the multi-species genome (Zhou et al., 2024) with 32.49B nucleotide bases. In the pre-training stage-1, VQDNA variants are pre-trained one epoch by AdamW (Loshchilov & Hutter, 2019) optimizer with a batch size of 1024 and a basic learning rate of 1×10^{-4} adjusted by a cosine scheduler with 8GPUs. In the pre-training stage-2, we apply masked language modeling (MLM) (Devlin et al., 2018) upon the tokenized VQ embeddings with a 25% random masking ratio for 500k steps. A similar pre-training setting is adopted, except the initial learning rate is 5×10^{-4} and the batch size of 2048. In stage 3 for downstream task adaptation, we also follow the fine-tuning evaluation setting in the GUE benchmark. The pre-trained Transformer encoder is fine-tuned by AdamW with LoRA on 28 GUE datasets (Zhou et al., 2024), 3 EEP datasets (Zhang et al., 2023), and the species classification dataset (Nguyen et al., 2023). The maximum length of the input nucleotide sequence is 512, in which case we report GFLOPs. The evaluation metrics of downstream tasks include top-1 accuracy (Acc), F1-score (F1), Matthews Correlation Coefficient (MCC), and Spearman Correlation (SC). All experiments are implemented with PyTorch, `transformers` library, and NVIDIA A100 GPUs. The average results of 3 trials are reported. View Appendix A and D for details.

4.2. Comparison Results

We take the popular genome language models into comparison, as shown in Table 2, including DNABERT (3-mer) (Ji et al., 2021), Nucleotide Transformer (NT) variants (Dalla-Torre et al., 2023), and DNABERT-2 (Zhou et al., 2024), where our VQDNA variants achieve the best and second best ranking of overall performances. We first evaluate VQDNA variants on the GUE benchmark, as shown in Table 5, Table 3, and Table 4, where 7 widely used genomic task are conducted, *i.e.*, Epigenetic Mark Prediction (EMP) for Yeast, Transcription Factor Prediction on mouse and human genome (TFP-M and TFP-H), Covid Variants Classification (CVC), Promoter Detection (PD), Core Promoter Detection (CPD), and Splice Site Prediction (SSP). Two versions of VQDNA consistently outperform the previous large-scale model NT-2500M-1000g and the efficient model DNABERT-2 with fewer parameters, while VQDNA (HRQ) further improves VQDNA (VQVAE) by a remarkable margin. We verify that VQDNA variants can also yield state-of-the-art performances on Editing Efficiency Prediction (EEP) with short genomic sequences in Table 4. Then, we scale up the sequence length as HyenaDNA and perform the 5-

Table 5. MCC (in %) performance of Epigenetic Marks Prediction tasks with different datasets fine-tuned on GUE benchmarks.

Method	Epigenetic Marks Prediction									
	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3	H3K79me3	H3K9ac	H4	H4ac
DNABERT (3-mer)	74.15	42.07	48.49	42.95	31.34	28.92	60.12	50.48	78.27	38.60
NT-500M-1000g (6-mer)	72.52	39.37	45.58	40.45	31.05	26.16	59.33	49.29	76.29	36.79
NT-2500M-1000g (6-mer)	74.61	44.08	50.86	43.10	30.28	30.87	61.20	52.36	79.76	41.46
DNABERT-2 (BPE)	78.27	52.57	56.88	50.52	31.13	36.27	67.39	55.63	80.71	50.43
VQDNA	78.56	53.93	60.62	52.84	33.73	38.49	68.15	56.28	81.32	50.33
VQDNA (HRQ)	**79.21**	**54.46**	**61.75**	**53.28**	**34.05**	**39.10**	**68.47**	**56.63**	**81.84**	**50.69**

Table 6. Top-1 accuracy (%) of species classification with scaling up sequence lengths, where N/A denotes out-of-memory.

Method	1k	20k	32k	250k	450k
HyenaDNA	61.13	87.42	93.42	97.90	99.40
DNABERT	39.61	76.21	91.93	N/A	N/A
DNABERT-2	61.04	86.83	99.28	N/A	N/A
VQDNA (HRQ)	**61.57**	**88.05**	**99.46**	N/A	N/A

species classification task in Table 6. Although HyenaDNA can fine-tune with extremely long sequences (*e.g.*, 450k), VQDNA (HRQ) achieves the best accuracy when the input sequence length is 32k (using FLASH Attention (Dao et al., 2022) and the gradient checkpoint technique), indicating that the learned VQDNA tokenizer can capture informative context and patterns for these extremely long-dependence tasks in genome analysis.

Table 7. Ablation study of the total codebook size in VQDNA tokenizers.

Code size	VQDNA		+HRQ	
	Rec.	Lin.	Rec.	Lin.
128	98.2	42.1	98.4	42.8
256	98.8	43.6	99.1	47.7
512	99.5	**44.8**	99.6	**48.9**
1024	**99.6**	44.5	**99.8**	48.2

Table 8. Ablation study of the codebook dimension (dim.) in VQDNA tokenizers.

Code dim.	VQDNA		+HRQ	
	Rec.	Lin.	Rec.	Lin.
256	99.4	44.3	99.5	48.2
384	99.5	**44.8**	99.6	**48.9**
768	99.6	44.6	99.6	**48.9**
1024	99.8	44.7	99.7	48.8

4.3. Ablation Study

Here, we ablate the VQ codebook settings and the mask ratio of MLM pre-training. Since applying the fine-tuning evaluation with the stage-1 tokenizers is too expensive, we report the reconstruction accuracy and the accuracy of linear probing (Lin.) (He et al., 2022) on VQDNA tokenized sequences of the CVC dataset. We first ablate the codebook dimension (dim.) and the total code size for VQDNA and HRQ. As shown in Table 7, we found that the size of 512 is an excellent trade-off between reconstruction and discrimination abilities for both VQDNA variants, capturing more intrinsic patterns. Then, Table 8 shows that the codebook dimension has less effect on the learned representation. Thus, we choose 384 as the default code dimension for efficiency. Then, we analyze the masking ratio in Table 9, reporting the fine-tuning results on the H3 and CVC datasets. We found that 25% can help VQDNA learn better representations than 15% or 20% in previous models (Ji et al., 2021). We hypothesize that VQDNA tokenizers may learn rich contextual information, allowing MLM to use large mask ratios to make the prediction task more difficult.

Table 9. Analysis of the mask ratio in the stage-2 MLM pre-training for our VQDNA.

Mask ratio	VQDNA		+HRQ	
	H3	CVC	H3	CVC
15%	77.9	72.6	78.3	73.7
20%	78.3	**73.4**	78.8	74.2
25%	**78.6**	73.2	**79.2**	**74.3**
30%	77.4	73.0	78.6	73.9

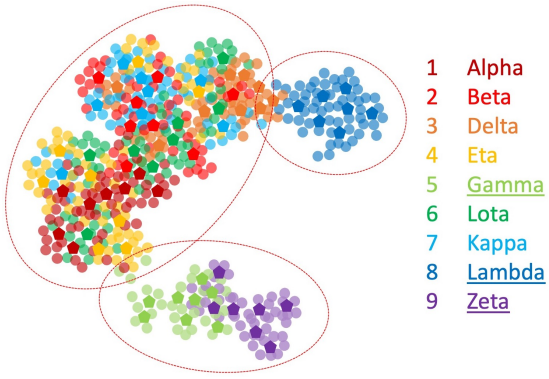


Figure 5. Visualization of the HRQ codebooks on CVC dataset by UMAP (McInnes et al., 2018). The label of each code is obtained by calculating the most relevant class with Grad-CAM (Selvaraju et al., 2017) of the linear classifier learned upon HRQ-tokenized sequences. The pentagon dots stand for codes of the layer-3 codebook, while the pale circle is the layer-6 ones. The result shows great intra- & inter-lineage pattern-awareness of HRQ vocabulary.

4.4. SARS-CoV-2 Analysis

SARS-CoV-2 is the cause of COVID-19, which has plunged our world into one of the gravest public health crises of the century. As the virus has proliferated globally with lightning speed, we have witnessed the rise of multiple SARS-CoV-2 variants from 2020∼2021, Alpha (B.1.1.7), Beta (B.1.351), Delta (B.1.617.2), Eta (B.1.525), Lota (B.1.526), Kappa (B.1.617.1), Lambda (C.37), Gamma (P.1), Zeta (P.2), each carrying unique mutations, which are identified by Pango lineage indicators (O'Toole et al., 2022). The rapid mutation in such a short time poses an urgent and formidable challenge as it may lead to variants that could evade immune responses and resist current vaccines and treatments. Given the real-world significance, we conduct an empirical analysis of this issue to validate the effectiveness of the VQ

tokenizer. Figure 5 shows that our HRQ tokenizer learns discriminative genome embeddings, where semantically close variants (same lineage) are clustered, and the semantically distinct ones (diverse lineage) are set apart, showcasing both the *intra-lineage* and *inter-lineage* pattern-aware ability. Moreover, the expanded codebook successfully captures fine-grained patterns. For example, Lambda is mutated from Delta with partially similar attributes but belongs to different lineages. Lambda circles in Figure 5 are closer to that of Delta, revealing the biological significance of HRQ. Refer to Appendix B for detailed background and analysis.

5. Conclusion and Discussion

Contributions. In this paper, we present VQDNA, a novel framework that leverages the VQ codebook as *learnable* genome vocabulary eschewing *hand-crafted* bias and rules for pattern-aware genome tokenization. To further push the limits of the VQ tokenizer, we propose HRQ, where varying scales of codebooks are designed in a hierarchy to enrich the limited genome vocabulary in a coarse-to-fine manner. Extensive experiments and analysis show the state-of-the-art performance of VQDNA across 32 datasets, highlighting its exceptional generalizability and biological significance.

Limitations and Future Works. There are several limitations in this work: (1) The superiority of VQDNA stems from its genome vocabulary learning, which is an additional training stage with extra costs compared to other models. Thus, there is still room for reducing its computational overhead to boost its applicability in multiple omics, as indicated by (Boshar et al., 2024). (2) Due to computational constraints, the model scale of VQDNA has not reached its maximum. It is worth exploring how to scale up VQDNA with model parameters and pre-training data to increase the gained merits. For example, employing an efficient encoder with linear attention mechanisms (Liu et al., 2024a;b) and pre-training with large-scale genomic databases (Dalla-Torre et al., 2023; Nguyen et al., 2024). (3) As the HRQ vocabulary has shown great biological significance in SARS-CoV-2 mutations, broader applications in genomics with VQDNA, such as generation tasks, deserve to be studied. Overall, all these avenues remain open for our future research.

Acknowledgement

This work was supported by Science and Technology Innovation 2030 - Major Project (No. 2021ZD0150100), National Natural Science Foundation of China Project (No. U21A20427), Project (No. WU2022A009) from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University and Integrated Bioengineering of Westlake University and Project (No. WU2023C019) from the Westlake University Industries of the Future Research Funding. We thank the Westlake University HPC Center for providing computational resources. This work was done by Zedong Wang during his research internship at Westlake University.

Impact Statement

The goal of this paper is to advance research in multi-species genomic sequence modeling by reconstructing the tokenization to end-to-end genome vocabulary learning tasks and further introducing the pattern-aware VQDNA and HRQ to form the three-stage VQDNA training pipeline. We have considered broader ethical impacts and do not foresee VQDNA directly leading to negative societal consequences. The genome datasets used are existing public resources that do not contain private or sensitive information. Thus, there are no privacy concerns in this study. Empirical analysis of SARS- CoV-2 mutations reveal the biological significance of the learned HRQ vocabulary, showcasing its potential for broader applications in genomics. We call on researchers in the community to extend this study to explore more applications with discriminative genome vocabulary learning.

References

Andersson, R. and Sandelin, A. Determinants of enhancer and promoter activities of regulatory elements. *Nature Reviews Genetics*, 21(2):71–87, 2020.

Angermueller, C., Lee, H. J., Reik, W., and Stegle, O. Deepcpg: accurate prediction of single-cell dna methylation states using deep learning. *Genome Biology*, 18, 2017.

Ba, J., Kiros, J. R., and Hinton, G. E. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Boshar, S., Trop, E., de Almeida, B. P., and PIERROT, T. Are genomic language models all you need? exploring genomic language models on protein downstream tasks. In *ICLR 2024 Workshop on Machine Learning for Genomics Explorations*, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Byrska-Bishop, M., Evani, U. S., Zhao, X., Basile, A. O., Abel, H. J., Regier, A. A., Corvelo, A., Clarke, W. E., Musunuri, R., Nagulapalli, K., et al. High-coverage whole-genome sequencing of the expanded

1000 genomes project cohort including 602 trios. *Cell*, 185(18):3426–3440, 2022.

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. Maskgit: Masked generative image transformer. *CVPR*, pp. 11305–11315, 2022.

Chelba, C., Norouzi, M., and Bengio, S. N-gram language modeling using recurrent neural network estimation. *arXiv preprint arXiv:1703.10724*, 2017.

Consortium, M. E., Stamatoyannopoulos, J. A., Snyder, M., Hardison, R., Ren, B., Gingeras, T., Gilbert, D. M., Groudine, M., Bender, M., Kaul, R., et al. An encyclopedia of mouse dna elements (mouse encode). *Genome biology*, 13:1–5, 2012a.

Consortium, M. E., Stamatoyannopoulos, J. A., Snyder, M., Hardison, R., Ren, B., Gingeras, T., Gilbert, D. M., Groudine, M., Bender, M., Kaul, R., et al. An encyclopedia of mouse dna elements (mouse encode). *Genome biology*, 13:1–5, 2012b.

Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., Carranza, N. L., Grzywaczewski, A. H., Oteri, F., Dallago, C., Trop, E., de Almeida, B. P., Sirelkhatim, H., et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pp. 2023–01, 2023.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Dreos, R., Ambrosini, G., Cavin Périer, R., and Bucher, P. Epd and epdnew, high-quality promoter resources in the next-generation sequencing era. *Nucleic acids research*, 41(D1):D157–D164, 2013.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, June 2021.

Fishman, V., Kuratov, Y., Petrov, M., Shmelev, A., Shepelin, D., Chekanov, N., Kardymon, O., and Burtsev, M. Gena-lm: A family of open-source foundational models for long dna sequences. biorxiv. 2023.

Gray, R. Vector quantization. *IEEE Assp Magazine*, 1(2): 4–29, 1984.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Huh, M., Cheung, B., Agrawal, P., and Isola, P. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023a.

Huh, M., Cheung, B., Agrawal, P., and Isola, P. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023b.

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

Kelley, D. R., Snoek, J., and Rinn, J. L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, 26: 990 – 999, 2015.

Khare, S., Gurry, C., Freitas, L., Schultz, M. B., Bach, G., Diallo, A., Akite, N., Ho, J., Lee, R. T., Yeo, W., et al. Gisaid's role in pandemic response. *China CDC weekly*, 3(49):1049, 2021.

Kudo, T. and Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

Lai, H.-Y., Zhang, Z.-Y., Su, Z.-D., Su, W., Ding, H., Chen, W., and Lin, H. iproep: a computational predictor for predicting promoter. *Molecular Therapy-Nucleic Acids*, 17:337–346, 2019.

Le, N. Q. K., Ho, Q.-T., Nguyen, V.-N., and Chang, J.-S. Bert-promoter: An improved sequence-based predictor of dna promoter using bert pre-trained model and shap

feature selection. *Computational Biology and Chemistry*, 99:107732, 2022.

Lee, D., Kim, C., Kim, S., Cho, M., and Han, W.-S. Autoregressive image generation using residual quantization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11523–11532, 2022.

Li, S., Zhang, L., Wang, Z., Wu, D., Wu, L., Liu, Z., Xia, J., Tan, C., Liu, Y., Sun, B., and Li, S. Z. Masked modeling for self-supervised representation learning on vision and beyond. *ArXiv*, abs/2401.00897, 2023a.

Li, S., Liu, Z., Zang, Z., Wu, D., Chen, Z., and Li, S. Z. Genurl: A general framework for unsupervised representation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024a.

Li, S., Wang, Z., Liu, Z., Tan, C., Lin, H., Wu, D., Chen, Z., Zheng, J., and Li, S. Z. Efficient multi-order gated aggregation network. In *International Conference on Learning Representations (ICLR)*, 2024b.

Li, T., Chang, H., Mishra, S. K., Zhang, H., Katabi, D., and Krishnan, D. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b.

Liu, H., Yan, W., and Abbeel, P. Language quantized autoencoders: Towards unsupervised text-image alignment. *arXiv preprint arXiv:2302.00902*, 2023.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Liu, Z., Li, S., Wang, L., Wang, Z., Liu, Y., and Li, S. Z. Short-long convolutions help hardware-efficient linear attention to focus on long sequences. In *International Conference on Machine Learning (ICML)*, 2024a.

Liu, Z., Li, W., Li, S., Wang, Z., Lin, H., and Li, S. Z. Longvq: Long sequence modeling with vector quantization on structured memory. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2024b.

Locke, A. E., Kahali, B., Berndt, S. I., Justice, A. E., Pers, T. H., Day, F. R., Powell, C., Vedantam, S., Buchkovich, M. L., Yang, J., et al. Genetic studies of body mass index yield new insights for obesity biology. *Nature*, 518(7538): 197–206, 2015.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Marin, F. I., Teufel, F., Horlacher, M., Madsen, D., Pultz, D., Winther, O., and Boomsma, W. Bend: Benchmarking dna language models on biologically meaningful tasks. In *International Conference on Learning Representations (ICLR)*, 2024.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Mentzer, F., Minnen, D., Agustsson, E., and Tschannen, M. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

Min, X., Zeng, W., Chen, N., Chen, T., and Jiang, R. Chromatin accessibility prediction via convolutional long short-term memory networks with k-mer embedding. *Bioinformatics*, 33:i92 – i101, 2017.

Moore, J. E., Purcaro, M. J., Pratt, H. E., Epstein, C. B., Shoresh, N., Adrian, J., Kawli, T., Davis, C. A., Dobin, A., et al. Expanded encyclopaedias of dna elements in the human and mouse genomes. *Nature*, 583(7818):699–710, 2020.

Nguyen, E., Poli, M., Durrant, M. G., Thomas, A. W., Kang, B., Sullivan, J., Ng, M. Y., Lewis, A., Patel, A., Lou, A., et al. Sequence modeling and design from molecular to genome scale with evo. *bioRxiv*, pp. 2024–02, 2024.

Nguyen, E. D., Poli, M., Faizi, M., Thomas, A. W., Birch-sykes, C. J., Wornow, M., Patel, A., Rabideau, C. M., Massaroli, S., Bengio, Y., Ermon, S., Baccus, S. A., and Ré, C. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *ArXiv*, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

O'Toole, Á., Pybus, O. G., Abram, M. E., Kelly, E. J., and Rambaut, A. Pango lineage designation and assignment using sars-cov-2 spike gene nucleotide sequences. *BMC genomics*, 23(1):1–13, 2022.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Razavi, A., Oord, A. V. D., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 618–626, 2017.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909, 2015.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. Saprot: Protein language modeling with structure-aware vocabulary. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Van Den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Vidaki, A., Ballard, D., Aliferi, A., Miller, T. H., Barron, L. P., and Court, D. S. Dna methylation-based forensic age prediction using artificial neural networks and next generation sequencing. *Forensic Science International. Genetics*, 28:225 – 236, 2017.

Visscher, P. M., Wray, N. R., Zhang, Q., Sklar, P., McCarthy, M. I., Brown, M. A., and Yang, J. 10 years of gwas discovery: biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

Wang, R., Wang, Z., Wang, J., and Li, S. Splicefinder: ab initio prediction of splice sites using convolutional neural network. *BMC bioinformatics*, 20:1–13, 2019.

Wu, L., Huang, Y., Tan, C., Gao, Z., Hu, B., Lin, H., Liu, Z., and Li, S. Z. Psc-cpi: Multi-scale protein sequence-structure contrasting for efficient and generalizable compound-protein interaction prediction. *arXiv preprint arXiv:2402.08198*, 2024a.

Wu, L., Tian, Y., Huang, Y., Li, S., Lin, H., Chawla, N. V., and Li, S. Z. Mape-ppi: Towards effective and efficient protein-protein interaction prediction via microenvironment-aware protein embedding. *arXiv preprint arXiv:2402.14391*, 2024b.

Yang, J., Lee, S. H., Goddard, M. E., and Visscher, P. M. Gcta: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82, 2011.

Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A. G., Yang, M.-H., Hao, Y., Essa, I., et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10459–10469, 2023a.

Yu, L., Cheng, Y., Wang, Z., Kumar, V., Macherey, W., Huang, Y., Ross, D. A., Essa, I., Bisk, Y., Yang, M.-H., Murphy, K. P., Hauptmann, A. G., and Jiang, L. SPAE: Semantic pyramid autoencoder for multimodal generation with frozen LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Yu, L., Lezama, J., Gundavarapu, N. B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Gupta, A., Gu, X., Hauptmann, A. G., et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023c.

Zhang, H., Yan, J., Lu, Z., Zhou, Y., Zhang, Q., Cui, T., Li, Y., Chen, H., and Ma, L. Deep sampling of grna in the human genome and deep-learning-informed prediction of grna activities. *Cell Discovery*, 9(1):48, 2023.

Zhou, J. and Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12:931–934, 2015.

Zhou, Z., Ji, Y., Li, W., Dutta, P., Davuluri, R., and Liu, H. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. In *International Conference on Learning Representations (ICLR)*, 2024.

Žiga Avsec, Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J. M., Kohli, P., and Kelley, D. R. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18:1196 – 1203, 2021.

Appendix

The appendix is structured as follows:

- In Appendix A, we provide implementation details of training schemes of pre-training and fine-tuning stages and hyperparameter settings.
- In Appendix B, we describe background knowledge of SARS-CoV-2 variant classification and analysis.
- In Appendix C and Appendix D, we provide detailed information for the pre-training nucleotide database and 32 genomic downstream tasks datasets.

A. Implementation Details

Pre-training. Since the two pre-training stages utilize different self-supervised methods, as shown in Figure 2, we can pre-train the VQDNA tokenizer and Transformer encoder separately using human genome and multi-species genome databases (mentioned in Appendix C). In pre-training stage 1, the VQDNA or HRQ model is optimized by AdamW (Loshchilov & Hutter, 2019) ($\beta_1 = 0.9$, $\beta_2 = 0.98$, and the weight decay of 0.01), a learning rate of 1×10^{-4}, and a batch size of 1024 for one epoch (around 1M steps). The hyper-parameter β in Eq. (3) and Eq. (8) is set to 0.5 and 0.9 to balance the codebook updating and reconstruction. To stabilize training, we apply 50k steps of linear warmup with a max sequence length of 128. Then, we use the max sequence length of 256. To further improve the clustering effects of codebooks in HRQ, we employ the Repeated K-means trick (Huh et al., 2023b) once after the warmup stage. In pre-training stage 2, we store the tokenized data (as binary files) by the pre-trained VQDNA and directly load the processed nucleotide sequences to save training budges. We perform Masked Language Modeling (MLM) pre-training (Ji et al., 2021; Wu et al., 2024b) for 500k steps with a batch size of 2048, the basic learning rate of 5×10^{-4} adjusted by the cosine annealing scheduler (decay to 1×10^{-6}), and a linear warmup of 10k steps. The masking ratio is 25%, and the maximum input length is 512.

Fine-tuning. During stage 3 for downstream tasks in Figure 2, the pre-trained VQDNA tokenizer and self-attention blocks in the Transformer encoder are frozen, while Low-Rank Adaptation (LoRA) machines are used for parameter-efficient fine-tuning optimized by AdamW optimizer with a batch size of 32 and a weight decay of 0.01. For each task, we choose the best combinations of the basic learning rate $\{1e-5, 3e-5, 5e-5\}$, the dropout rate $\{0, 0.05\}$, and the total fine-tuning epoch $\{4, 6, 8, 10\}$ on the validation set, because different tasks vary in convergence difficulty and the input length. Note that the maximum input length is set to 512 during fine-tuning and uses the maximum length of each dataset during inference. We use the default LoRA hyper-parameters (a LoRA alpha is 16 and a LoRA r of 8). We report the averaged results over three runs based on the optimal settings in Sec. 4.2.

B. SARS-CoV-2 Classification Analysis

Background. SARS-CoV-2 has continuously changed throughout the COVID-19 pandemic, resulting in multiple variants distinct from the original virus. This type of change is biologically termed a mutation-a single base change within a genome, which happens frequently but does not necessarily alter the genomic patterns of the virus. In other words, viruses with similar nucleotides may not necessarily present similar genomic patterns. To address the concerning variants, the World Health Organization (WHO) has categorized specific viral lineages-a group of genomically related viruses descended from a common ancestor based on shared essences and properties: Variants of Interest (VOI), Variants of Concern (VOC), Variants of High Consequence (VOHC), and Variants Being Monitored (VBM). This taxonomy distinguishes well between viruses with similar nucleotides but different characteristics as an ideal classification indicator.

SARS-CoV-2 Variants. For empirical analysis, we consider the following sublineages: Alpha variants (B.1.1.7), Beta variants (B.1.351), Delta (B.1.617.2), Eta (B.1.525), Lota (B.1.526), Kappa (B.1.617.1), Lambda (C.37), Gamma (P.1), Zeta (P.2) on CVC dataset. Based on the Pango lineage system (O'Toole et al., 2022), Alpha, Beta, Delta, Eta, Lota, and Kappa are sequentially mutated, forming the Omicron (BA) lineage with similar genomic attributes. Lambda (C.37), however, is biologically mutated from Delta (B.1.617.2) which shares diverse characteristics but with more similar patterns than the others. Gamma (P.1) and Zeta (P.2) are the other two lineages, where Zeta (P.2) is mutated from Gamma (P.1) with similar genomic patterns.

Analysis. Therefore, tokenizing a genome sequence only according to its intra-sequence nucleotide bases is sub-optimal without awareness of its high-level genomic patterns. This is exactly the problem that VQ tokenizer intends to resolve. Ideally, the learned codebook in VQDNA can record the underlying patterns of input genomes. Specifically, different code embeddings portray different high-dimensional semantics belonging to certain groups of lineages with common attributes and characteristics, and thus, pattern-aware genome embeddings can be computed with these discriminative codebooks. Moreover, as the proposed HRQ tokenizer intends to capture more fine-grained details for hierarchical codebook learning, it is expected to distinguish the above SARS-CoV-2 variants more precisely. We visualize the learned codebooks by UMAP (McInnes et al., 2018), where the labels are obtained by Grad-CAM of the

linear classifier with HRQ-tokenized genome sequences. As shown in Figure 5, the learned codebooks are capable of distinguishing all the tested SARS-CoV-2 variants with well-aligned biological correlations. Note that the pentagon dots stand for codes of the layer-3 codebook while the circle denotes the layer-6 ones. Code embeddings belonging to each lineage (Omicron (BA), Gamma (P.1) and Zeta (P.2)) are well clustered, which indicates that they shared similar patterns for encoding different same-type variants, showcasing the exceptional *intra-lineage* pattern-aware ability. More surprisingly, our HRQ codebooks can further capture the *inter-lineage* patterns as the Zeta (P.2) cluster is close to the Gamma (P.1) cluster, which well exhibits their mutation correlations. Furthermore, the Delta (B.1.617.2) cluster from the Omicron lineage is near the Lambda (C.37) one, demonstrating the *inter-lineage* pattern-aware capability of the HRQ codebooks. In addition, the expanded codebook successfully captures fine-grained patterns. For example, Lambda is mutated from Delta with partially similar attributes but belongs to different lineages. Lambda circles in Figure 5 are closer to that of Delta, which means the HRQ can capture the fine-grained patterns within genomes and lead to more precise tokenization. All this empirically confirms our claims on VQ tokenizer and reveals the biological significance of HRQ vocabulary.

C. Multi-Species Genome for Pre-Training

Following (Zhou et al., 2024), Table A1 lists the 135 species in 5 categories that we randomly selected for pre-training genome foundation models and presents the number of nucleotides collected from each species. We collected these pre-training data from the database of the National Center for Biotechnology Information (NCBI) at `https://www.ncbi.nlm.nih.gov/` based on Multi-species genome `https://huggingface.co/datasets/InstaDeepAI/multi_species_genomes` provided in Nucleotide Transformer (Dalla-Torre et al., 2023).

Table A1. Details statistics of the multi-species genome dataset for pre-training.

Category	**Species**	**Nucleotides (M)**
Fungi	Ceratobasidium, Claviceps Maximensis, Fusarium Annulatum, Melampsora, Metschnikowia, Mucor Saturninus, Penicillium Chermesinum, Saccharomyces Cerevisiae, Sporopachydermia Quercuum, Tranzscheliella Williamsii, Xylariales	3774
Protozoa	Phytophthora Sojae, Pythium Apiculatum	1244
Mammalian	Bubalus Bubalis, Camelus Dromedarius, Human, Macaca Assamensis, Macaca Nigra, Mus Musculus, Peromyscus Californicus	186931
Other Vertebrate	Anas Zonorhyncha, Coregonus Clupeaformis, Gnathonemus Longibarbis, Myxocyprinus Asiaticus, Rhipidura Dahli	79358
Bacteria	Aeromonas, Agrobacterium, Alcaligenaceae Bacterium, Aliivibrio, Alphaproteobacteria Bacterium, Amycolatopsis Antarctica, Anaerostipes Faecis, Arthrobacter, Atopobium, Bacillus Bc15, Bacillus Bs3 2021, Bacterium, Bacteroidetes Bacterium Qs, Breoghania Corrubedonensis, Caldicoprobacter Oshimai, Candidatus Cryptobacteroides Excrementipullorum, Candidatus Dadabacteria Bacterium Rbg Combo, Candidatus Dwaynia Gallinarum, Candidatus Falkowbacteria Bacterium, Candidatus Geothermincola Secundus, Candidatus Gottesmanbacteria Bacterium, Candidatus Nomurabacteria Bacterium Full, Candidatus Portnoybacteria Bacterium Big Fil Rev, Candidatus Regiella Insecticola, Candidatus Roizmanbacteria Bacterium Combo All, Candidatus Rokubacteria Bacterium, Candidatus Saccharibacteria Bacterium, Candidatus Staskawiczbacteria Bacterium Full, Christensenella, Clostridiaceae Bacterium, Clostridiales Bacterium, Clostridium Cag 505, Clostridium Mcc328, Clostridium Nexile, Clostridium Uba3521, Collinsella Urealyticum, Coprobacillus Cateniformis, Cyanobium, Dehalococcoidia Bacterium, Enterobacteriaceae Bacterium, Evtepia Gabavorous, Firmicutes Bacterium, Fulvivirga, Jeongeupia Chitinilytica, Legionella Endosymbiont Of Polyplax Serrata, Listeria Ilorinensis, Maribacter Cobaltidurans, Marinomonas, Mesorhizobium, Methyloceanibacter Caenitepidi, Microvirga, Mycolicibacter Engbaekii, Novosphingobium, Omnitrophica Wor Bacterium Rbg, Pantoea, Paraburkholderia Edwinii, Parerythrobacter Lutipelagi, Paulownia Witches Phytoplasma, Polaromonas Eurypsychrophila, Prevotella Ag 487 50 53, Prevotella Uba3619, Prevotella Uba634, Prochlorococcus Ag-321-I09, Prochlorococcus Ag-363-B18, Prochlorococcus Ag-402-L19, Prochlorococcus Scb243 498N4, Providencia, Pseudomonas 35 E 8, Pseudomonas Bigb0408, Pseudomonas P867, Pseudomonas Promysalinigenes, Roseobacter, Salinicola Peritrichatus, Salmonella S096 02912, Salmonella Zj-F75, Sinorhizobium, Sodalis Ligni, Sphaerochaeta, Sphingobacterium, Sphingomonas Carotinifaciens, Sphingomonas Mesophila, Sporosarcina Jiandibaonis, Sporosarcina Ureilytica, Staphylococcus Gdq20D1P, Staphylococcus M0911, Streptococcus, Streptomyces 8401, Streptomyces Di166, Streptomyces Durbertensis, Streptomyces Neau-Yj-81, Streptomyces Rk74B, Thermopetrobacter, Uncultured Kushneria, Uncultured Phascolarctobacterium, Uncultured Proteus, Verrucomicrobiales Bacterium, Vibrio, Victivallis Lenta, Virgibacillus Salexigens, Xanthomonadales Bacterium	3610

D. Downstream Datasets

D.1. GUE Benchmark

The GUE benchmark proposed by DNABERT-2 contains 28 datasets of 7 biological important genome analysis tasks for 4 different species. To comprehensively evaluate the genome foundation models in modeling variable-length sequences, we select tasks with input lengths ranging from 70 to 1000. Table A2 presents the detailed statistics of each evaluation dataset. The following descriptions of the supported tasks are included in the GUE benchmark (Zhou et al., 2024). We attach these resources here for illustration.

Promoter Detection (Human) focuses on identifying (proximal) promoter regions, crucial sequences in the human genome responsible for instigating transcription. As many primary regulatory elements are located in this region, accurately detecting these sites is instrumental in advancing our grasp of gene regulation mechanisms and pinpointing the genomic underpinnings of numerous diseases. The dataset is divided twofold, TATA and non-TATA, based on whether a TATA box motif is present in the sequence. We extract -249 +50 bp around the transcription start site (TSS) from TATA and non-TATA promoters downloaded from Eukaryotic Promoter Database (EPDnew) (Dreos et al., 2013) and use it as our promoter class. Meanwhile, we construct the non-promoter class with equal-sized randomly selected sequences outside of promoter regions but with TATA motif (TATA non-promoters) or randomly substituted sequences (non-TATA, non-promoters). We also combine the TATA and non-TATA datasets to obtain a combined dataset named *all*.

Core Promoter Detection (Human) is similar to proximal promoter detection with a focus on predicting the core promoter region only, the central region closest to the TSS and start codon. A much shorter context window (center -34 +35 bp around TSS) is provided, making this a more challenging task than proximal promoter prediction.

Transcription Factor Binding Site Prediction (Human) predicts binding sites of transcription factors (TF), the key proteins that regulate gene expression in the human genome. Their accurate prediction is key to deciphering complex genetic interactions and identifying potential targets for gene therapies. We accessed the legacy 690 ENCODE ChIP-seq experiments (Consortium et al., 2012b) via the UCSC genome browser, encompassing 161 TF binding profiles in 91 human cell lines. We extracted a 101-bp region around the center of each peak as the TFBS class and nonoverlapping sequences with the same length and GC content as the non-TFBS class. Finally, we randomly select 5 datasets out of a subset of 690 that we curated by heuristically filtering out tasks that are either too trivial (e.g., over 0.95 F1) or too challenging (e.g., less than 0.50 F1) for existing language models.

Splice Site Prediction (Human) predicts splice donor and acceptor sites, the exact locations in the human genome where alternative splicing occurs. This prediction is crucial to understanding protein diversity and the implications of aberrant splicing in genetic disorders. The dataset (Wang et al., 2019) consists of 400-bp-long sequences extracted from Ensembl GRCh38 human reference genome. As suggested by Ji et al. (2021), existing models can achieve almost perfect performance on the original dataset, containing 10,000 splice donors, acceptors, and non-splice site sequences, which is overly optimistic about detecting non-canonical sites in reality. As such, we reconstruct the dataset by iteratively adding adversarial examples (unseen false positive predictions in the hold-out set) in order to make this task more challenging.

Transcription Factor Binding Site Prediction (Mouse) predicts the binding site of transcription factors on mouse genomes. Like human binding site data, we obtain mouse ENCODE ChIP-seq data (Consortium et al., 2012a), the largest available collection on the UCSC genome browser (n=78). This time, the negative examples are created using dinucleotide shuffling while preserving relative frequencies, while all other settings stay the same as the human TFBS prediction dataset. We also randomly select 5 datasets out of the 78 datasets using the same process described above.

Epigenetic Marks Prediction (Yeast) predicts epigenetic marks in yeast, modifications on the genetic material that influence gene expression without altering the DNA sequence. Precise prediction of these marks aids in elucidating the role of epigenetics in yeast. We download the 10 datasets from `http://www.jaist.ac.jp/~tran/nucleosome/members.htm` and randomly split each dataset into training, validation, and test sets with a ratio of 8:1:1.

Covid Variant Prediction (Virus) aims to predict the variant type of the SARS_CoV_2 virus based on 1000-length genome sequences. We download the genomes from the EpiCoV database (Khare et al., 2021) of the Global Initiative on Sharing Avian Influenza Data (GISAID). We consider 9 types of SARS_CoV_2 variants, including *Alpha*, *Beta*, *Delta*, *Eta*, *Gamma*, *Iota*, *Kappa*, *Lambda* and *Zeta*.

D.2. Additional Datasets

Editing Efficiency Prediction Dataset Life science studies involving clustered, regularly interspaced short palindromic repeat (CRISPR) editing generally apply the best-performing guide RNA (gRNA) for a gene of interest in

Table A2. Statistics of tasks in GUE benchmark (Zhou et al., 2024), including the task name, evaluation metric, and the number of training, validation, and test samples in each dataset.

Task	Metric	Datasets	Train / Dev / Test	Class
Core Promoter Detection	MCC	tata	4904 / 613 / 613	2
		notata	42452 / 5307 / 5307	
		all	47356 / 5920 / 5920	
Promoter Detection	MCC	tata	4904 / 613 / 613	2
		notata	42452 / 5307 / 5307	
		all	47356 / 5920 / 5920	
Transcription Factor	MCC	wgEncodeEH000552	32378 / 1000 / 1000	2
		wgEncodeEH000606	30672 / 1000 / 1000	
		wgEncodeEH001546	19000 / 1000 / 1000	
		wgEncodeEH001776	27294 / 1000 / 1000	
		wgEncodeEH002829	19000 / 1000 / 1000	
Splice Site Prediction	MCC	reconstructed	36496 / 4562 / 4562	3
Transcription Factor	MCC	Ch12Nrf2Iggrab	6478 / 810 / 810	2
		Ch12Znf384hpa004051Iggrab	53952 / 6745 / 6745	
		MelJundIggrab	2620 / 328 / 328	
		MelMafkDm2p5dStd	1904 / 239 / 239	
		MelNelfeIggrab	15064 / 1883 / 1883	
Epigenetic Marks Prediction	MCC	H3	11971 / 1497 / 1497	2
		H3K14ac	26438 / 3305 / 3305	
		H3K36me3	27904 / 3488 / 3488	
		H3K4me1	25341 / 3168 / 3168	
		H3K4me2	24545 / 3069 / 3069	
		H3K4me3	29439 / 3680 / 3680	
		H3K79me3	23069 / 2884 / 2884	
		H3K9ac	22224 / 2779 / 2779	
		H4	11679 / 1461 / 1461	
		H4ac	27275 / 3410 / 3410	
Virus	F1	Covid variant classification	77669 / 7000 / 7000	9

human DNA. Three large-scale gRNA datasets with SpCas9/gRNA activities are provided in Zhang et al. (2023) on K562, Jurkat, and H1 cells. The gRNA sequence is a standardized 63-length RNA encoded with ACGT, and the activity of editing efficiency is a scaler (as the regression target). The Spearman correlation is adopted as the metric to indicate high and low editing samples. The K562 dataset contains 277,000 training data and 69,262 testing data. Jurkat dataset contains 285,150 and 71,297 training and testing data. The H1 dataset has 54,580 and 13,654 training and testing samples.

Species Classification Since the discriminative mutations of different species are located in various positions, long-range dependencies are essential for discriminating different species. It requires the model to process extremely long sequences, *e.g.*, up to 32k, to learn a distinct mutational profile for each species. To investigate this special issue, long-range species classification data is collected in Nguyen et al. (2023), which randomly select five species, including human (homo sapien), lemur (lemur catta), mouse (musculus), pig (sus scrofa), and hippo (hippopotamus amphibious). This dataset contains long genomic sequences up to 1 million lengths.