Compression-Based Tokenization Improves Language Modeling of Hierarchical Genomic Structure

Yiping Wang[1,†], Jing Wang[2,3,4,†], Junhao Zhu[5], Fengyao Zhai[2,3,4], Hu Zhu[2,3,4], Ziwei Dai[5], Zengru Di[2,3], Da Zhou[1,6,*], Yu Liu[2,3,*]

[1]National Institute for Data Science in Health and Medicine, School of Medicine, Xiamen University, Xiamen 361102, China.
[2]Department of Systems Science, Faculty of Arts and Sciences, Beijing Normal University, Zhuhai 519087, China.
[3]International Academic Center of Complex Systems, Beijing Normal University, Zhuhai 519087, China.
[4]School of Systems Science, Beijing Normal University, Beijing 100875, China.
[5]Department of Systems Biology, School of Life Sciences, Southern University of Science and Technology, Shenzhen 518055, China.
[6]School of Mathematical Sciences, Xiamen University, Xiamen 361005, China.

*Corresponding author(s). E-mail(s): zhouda@xmu.edu.cn; yu.ernest.liu@bnu.edu.cn;
†These authors contributed equally to this work.

Abstract

Tokenization is a critical design choice in genomic language modeling. Widely used schemes—character-level encoding, fixed-length ***k***-mers, and greedy subword algorithms such as BPE—show intrinsic limitations on DNA that are magnified by the small four-letter alphabet. To address this, we adapt Ladderpath, an Algorithmic Information Theory method that identifies nested and hierarchical repetitions through optimal information reuse, into a tokenizer tailored for genomic sequences. Integrating this tokenizer into an 86-million-parameter Transformer yields the Ladderpath Tokenized Model (LTM), which surpasses the best existing models—including those several times larger—on 17 of 21 benchmarks. Comparisons with TF-IDF and other frequency-based baselines show that these gains extend beyond simple motif-frequency statistics. LTM's internal representations further exhibit biologically meaningful organization: token embeddings form coherent clusters, and sequence embeddings group promoters, enhancers, and histone-mark-associated regions without task-specific supervision, revealing an emergent structure of functional sequence classes. These findings show that strengthening the information-theoretic basis of tokenization provides a complementary path to architectural innovations and model scaling, enabling more compact and biologically aligned genomic foundation models.

Keywords: DNA Language Modeling, Tokenization, Algorithmic Information Theory, Ladderpath, Compression, Hierarchical Structure

1 Introduction

Recent advances in large language models (LLMs) have led to transformative progress in natural language processing (NLP), enabling capabilities ranging from text generation and translation to code synthesis and complex reasoning [1]. These models operate not on raw text, but on sequences of discrete symbols—tokens—derived through tokenization [2, 3]. Subword tokenization schemes such as Byte-Pair Encoding (BPE) [4], WordPiece [5], and Unigram [6] have become standard in mainstream LLMs.

Adapting these principles to DNA, however, presents unique challenges. Unlike natural languages, DNA lacks obvious "word" boundaries and is composed of only four nucleotides, yet spans millions to billions of bases [7, 8]. This combination of an extremely small alphabet and vast sequence length places substantial demands on memory and computation. Moreover, genomic information is encoded through overlapping features, multi-scale structural motifs, bidirectional transcription, and highly context-dependent regulatory logic—properties fundamentally distinct from human language. As a result, tokenization strategies directly transferred from NLP may fail to reflect the organizational principles and information-encoding mechanisms of DNA

[9]. Effective DNA tokenization may therefore require automatically discovering biologically meaningful sequence units, rather than relying solely on linguistic analogies [2].

Current DNA tokenization approaches lie along a spectrum. The most widely used strategy remains single-nucleotide tokenization, adopted by models such as Evo [10], HyenaDNA [11], and Enformer [12], which preserves maximal resolution but produces long token sequences that make capturing higher-order dependencies difficult [2]. k-mer tokenization offers an alternative: overlapping k-mers (e.g., DNABERT [13]) can introduce redundancy and potential information leakage [14], whereas non-overlapping k-mers (e.g., NT, the Nucleotide Transformer [15]) improve efficiency but may disrupt biologically coherent units. To increase flexibility, recent work has applied BPE—which learns variable-length tokens by iteratively merging frequent subword pairs—to genomic sequences [14, 16]. While BPE performs well in natural language, its suitability for DNA remains uncertain given the fundamental differences between genomic and linguistic structure [17–19].

A notable development in DNA tokenization is the incorporation of biological prior knowledge. The Genomic Tokenizer uses codons as basic token units, integrating the central dogma—including promoters, synonymous codons, and stop codons—into its vocabulary [7]. BioToken incorporates biologically relevant markers such as exons, introns, transcripts, and coding regions, leveraging genomic inductive biases to enable biologically grounded representations [20]. Recently, some methods have explored embedding tokenization within the model itself, allowing it to learn tokenization autonomously. VQDNA, based on the VQ-VAE method, uses a convolutional encoder and vector-quantized codebook for tokenization [21]. MxDNA, on the other hand, learns a tokenization strategy through gradient descent, overcoming the limitations of manual tokenization rules but still facing challenges with generalizability and interpretability [17].

These observations highlight the need for tokenization frameworks tailored to the modular and hierarchical structure of genomic sequences, rather than relying on frequency-based heuristics designed for natural language or directly embedding biological prior knowledge. To address this, we propose a deeper perspective: viewing tokenization as an effective process of information compression. Indeed, BPE, originally a compression algorithm, aims to reduce redundant information into a compact representation [22].

Algorithmic Information Theory (AIT) provides a solid theoretical foundation for understanding tokenization as information compression. The concept of Kolmogorov Complexity, which represents the minimum description of a system, reflects its intrinsic structure and complexity [23]. While Kolmogorov Complexity is theoretically incomputable, the recently proposed Ladderpath approach offers a practical approximation [24–26], and has already shown success in various domains [27, 28]. The core idea of Ladderpath is to measure the complexity of an object by analyzing the shortest path to reconstruct it step-by-step from its basic building blocks. The hierarchical decomposition and identification of repeated elements emphasized by the Ladderpath approach aligns well with the repetitive sequences and structures commonly found in DNA, such as transposable elements and repetitive motifs [29, 30].

In this paper, we explore the use of the Ladderpath approach for tokenizing DNA sequences and constructing a DNA language model, adapting the framework and training methods of the BPE-based GROVER model [16] by replacing the tokenizer. We anticipate that our Ladderpath tokenization strategy will generate biologically meaningful token representations, thereby enhancing our understanding of the "semantic" information embedded in DNA sequences. This, in turn, should improve the performance of DNA language models across various downstream tasks, providing new tools and perspectives for advanced research in genomics.

2 Results

2.1 Genome-Scale Ladderpath Construction and Token Vocabulary Derivation

The core idea of the Ladderpath approach is to reconstruct a target system—either a single sequence or a collection of sequences—through the minimal number of steps by systematically identifying, combining, and reusing repeated substructures [25]. The size of the shortest reconstruction map can thus be regarded as a quantitative indicator of the system's complexity, while the reused substructures themselves function analogously to words, subwords, or tokens that efficiently represent DNA sequences. For a given target system, Ladderpath decomposes its internal organization into a *directed acyclic graph* (DAG), which we term a *laddergraph*, as illustrated in Fig. 1 (highlighted in the red dashed box). For details on the computation of the Ladderpath, see Methods. This emphasis on nested and hierarchical reuse reflects biological reality, where complex functional regions of DNA sequences often originate from or involve recurring motifs and domains. Naturally, this suggests that the Ladderpath approach can be adapted into a tokenization strategy for genomic language models. In principle, the entire human genome—comprising approximately three billion characters—could be treated as a single sequence to be processed by the Ladderpath algorithm. However, the sheer scale of the data renders direct computation infeasible. To address this challenge, we designed a four-step pipeline, as outlined in Fig. 1.

The first step involved the pre-processing of the human reference genome (hg19). We employed the genome annotation file (RefSeq), ensuring that contiguous gene sequences were not disrupted, to segment the genome into 73,309 fragments, thereby preserving complete gene structures as much as possible. Building on this segmentation, we further partitioned the fragments into groups with a maximum cumulative

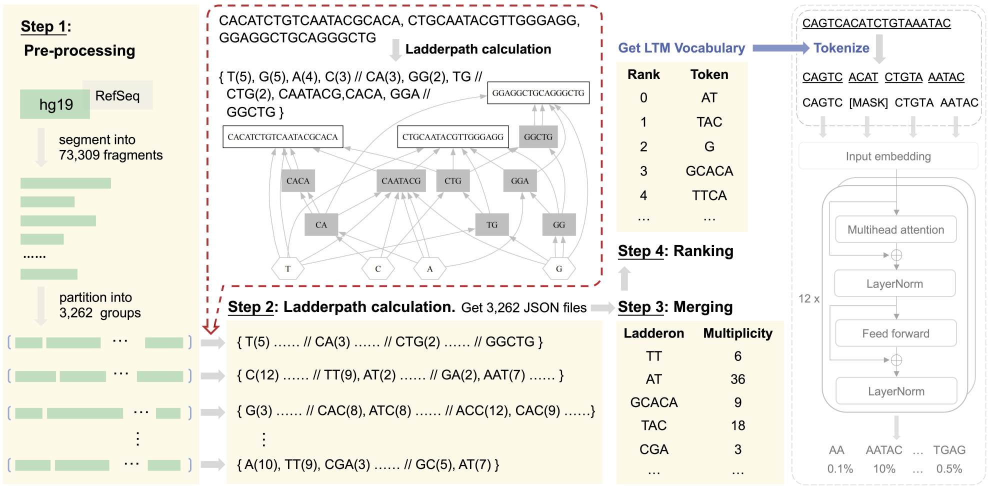

Step 1: Pre-processing
hg19
RefSeq
segment into 73,309 fragments
......
partition into 3,262 groups
CACATCTGTCAATACGCACA, CTGCAATACGTTGGGAGG, GGAGGCTGCAGGGCTG
Ladderpath calculation
{ T(5), G(5), A(4), C(3) // CA(3), GG(2), TG // CTG(2), CAATACG,CACA, GGA // GGCTG }
GGAGGCTGCAGGGCTG
CACATCTGTCAATACGCACA
CTGCAATACGTTGGGAGG
GGCTG
CACA
CAATACG
CTG
GGA
CA
TG
GG
T
C
A
G
Step 2: Ladderpath calculation. Get 3,262 JSON files
{ T(5) ...... // CA(3) ...... // CTG(2) ...... // GGCTG }
{ C(12) ...... // TT(9), AT(2) ...... // GA(2), AAT(7) ...... }
{ G(3) ...... // CAC(8), ATC(8) ...... // ACC(12), CAC(9) ......}
{ A(10), TT(9), CGA(3) ...... // GC(5), AT(7) }
Get LTM Vocabulary
Rank Token
0 AT
1 TAC
2 G
3 GCACA
4 TTCA
... ...
Step 4: Ranking
Step 3: Merging
Ladderon Multiplicity
TT 6
AT 36
GCACA 9
TAC 18
CGA 3
... ...
CAGTCACATCTGTAAATAC
Tokenize
CAGTC ACAT CTGTA AATAC
CAGTC [MASK] CTGTA AATAC
Input embedding
12 x
Multihead attention
LayerNorm
Feed forward
LayerNorm
AA AATAC ... TGAG
0.1% 10% ... 0.5%

Fig. 1: **Four-step pipeline for genome-scale Ladderpath tokenization.** The process consists of four steps: pre-processing, Ladderpath computation, merging, and ranking (see main text for details). The red dashed box provides a more detailed illustration of the Ladderpath computation, including the resulting *laddergraph*, which captures the nested and hierarchical relationships among repeated subsequences. In the final step, the top 1,000 ladderons are selected to form the LTM vocabulary, which is then directly used to tokenize genomic sequences for the downstream language model, as shown on the right.

length of one million characters, thus ensuring computational tractability. This strategy produced 3,262 fragment groups.

The second step was to compute the Ladderpaths for each of the 3,262 fragment groups. Each group was processed independently, producing 3,262 Ladderpath JSON files. Collectively, these files encode complete Ladderpath representations, capturing the nested and hierarchical relationships among repeated subsequences.

The third step involved merging the information contained in the 3,262 JSON files. In a Ladderpath, repeated subsequences are defined as *ladderons*, which serve as potential tokens in the vocabulary we aim to construct. An associated concept is *multiplicity*, denoting the number of times a given ladderon is reused in the original sequence. Ladderons with high multiplicity are therefore not only extensively reused across the genome but may also carry significant structural or functional importance.

Accordingly, we compiled all unique ladderons—25,917,682 in total—from the 3,262 Ladderpath files and aggregated their multiplicities across all files as a measure of importance.

The fourth step involved ranking the ladderons by their importance in descending order, after which we selected the top 1,000 to construct our vocabulary. With a size of 1,000, this vocabulary is designed to capture the most essential and representative compositional patterns of the genome while maintaining computational efficiency. The language model subsequently tokenized and pre-trained with this vocabulary is referred to as the *Ladderpath Tokenized Model* (LTM).

2.2 Comparative Analysis of Tokenization Vocabulary Structures

We began by examining the token frequency distributions generated by traditional fixed-length k-mer tokenization on the human reference genome. When k is small (e.g., $k = 2$ or 3), the resulting vocabulary is too limited to provide an expressive representation of long genomic sequences. However, as k increases to moderate values (e.g., $k = 4 - 6$ and above), the distribution becomes markedly skewed: a small subset of highly frequent k-mers dominates the corpus, while the majority occur only rarely. For example, as illustrated in Fig. 2a, in the top panel showing the distribution of 4-mers, the most frequent 16.4% of 4-mers already account for 30% of all occurrences in the genome; the top 49.6% account for 70% of the total, meaning that roughly half of the vocabulary consists of tokens with very low usage. This imbalance becomes even more pronounced for larger k-mers: for 5-mers (second panel of Fig. 2a), the top 28.2% of tokens cover 50% of all occurrences, and for 6-mers, the top 25% cover half of the corpus. Moreover, because the number of possible k-mers grows exponentially with k, the vocabulary size expands dramatically, further exacerbating the prevalence of low-frequency tokens. This imbalance introduces a severe rare-word problem and significantly reduces training efficiency.

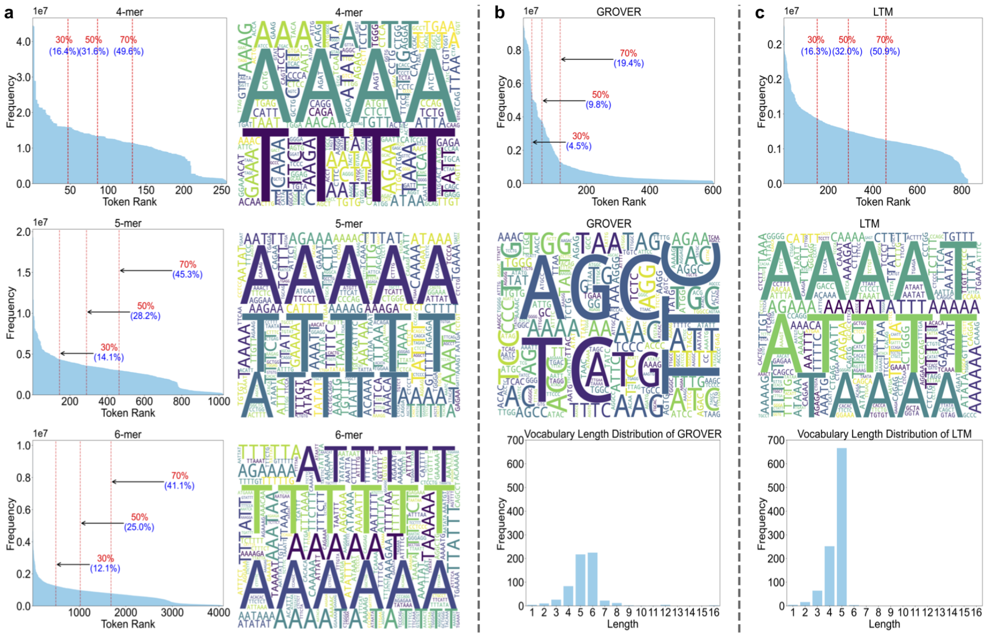


Fig. 2: **Token distribution patterns for *k*-mer, GROVER and LTM vocabularies.** (a) Fixed-length *k*-mers. The top panel shows the frequency distribution of all 4-mers in the human genome. The vertical red dashed lines indicate that the most frequent 16.4%, 31.6%, and 49.6% of 4-mers account for 30%, 50%, and 70% of all occurrences, respectively. The middle and bottom panels show the corresponding results for 5-mers and 6-mers, with word clouds on the right. (b) BPE tokenization used in GROVER. The first panel shows a highly skewed genome-wide token frequency distribution: the most frequent 19.4% of tokens cover 70% of all occurrences, indicating sparse usage of the remaining vocabulary. The second panel shows the GROVER word cloud, and the third displays the length distribution of the 600 BPE tokens, which range from 1 to 16 nucleotides, with 6-mers being the most frequent. (c) Ladderpath tokenization used in LTM. The panels correspond to those in (b). The final panel shows the length distribution of the 1,000 Ladderpath-derived tokens, which span 1 to 6 nucleotides, with 5-mers being the most common.

We next conducted a quantitative comparison between GROVER (Fig. 2b)—whose 600-token vocabulary was constructed through 600 iterations of the BPE algorithm—and our LTM vocabulary (Fig. 2c). First, GROVER exhibits a more severe usage imbalance: the most frequent 19.4% of tokens account for 70% of all occurrences, an even stronger skew than that observed for 6-mers, indicating sparse usage of the remaining vocabulary. We also calculated the average token length after tokenizing the human genome. For GROVER, the average token length is $L = 4.069$, computed as the total number of nucleotides divided by the total number of tokens—meaning that each token represents, on average, 4.069 nucleotides. In contrast, tokenization with LTM yields a longer average token length of $L = 4.883$, an increase of about 20% relative to GROVER. This implies that LTM can encode genomic sequences of the same length using fewer tokens, thereby expanding the effective sequence context accessible to the model under a fixed input length. Such extended context is crucial for capturing long-range dependencies across genomic regions.

Then, we conducted a detailed comparison between BPE—the tokenization method used by GROVER—and the Ladderpath tokenization employed in LTM. A key structural limitation of BPE stems from its greedy vocabulary construction: at each step, it merges only the most frequent adjacent pair, which can inadvertently exclude other high-frequency substructures from the vocabulary. For example, once BPE merges the frequent pair CC to form a token, sequences such as CCG are subsequently more likely to be segmented into CC and G, even if CG is itself a frequently occurring motif. Consequently, BPE systematically fails to capture certain short but meaningful repeated patterns whenever they do not lie along its greedy merge trajectory.

To examine this limitation more rigorously, we performed the following experiment. We sampled 100 sequences of length 5,000 from the human genome and generated 100 random sequences of the same length (with A, T, G, C sampled uniformly). For both datasets, we applied BPE and Ladderpath—using identical vocabulary sizes of 100—and counted the number of distinct 2-mers represented in their vocabularies. Because DNA consists of four nucleotides, there are 16 possible 2-mers. However, as shown in Fig. 3a,b, BPE consistently recovers only around 10, regardless of whether the input is random or genomic. This indicates that the missing 2-mers are a consequence of BPE's algorithmic bias rather than the underlying data distribution. In contrast, Ladderpath does not exhibit this limitation: as shown in Fig. 3c,d, it recovers nearly all 16 possible 2-mers for both random and genomic sequences.

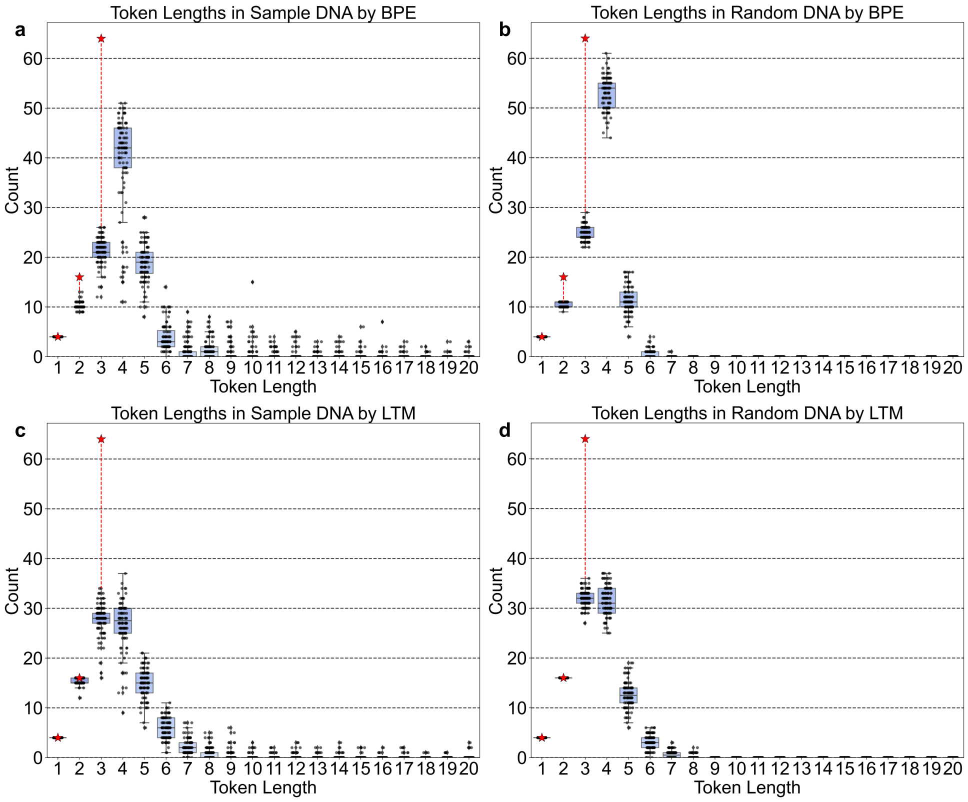


Fig. 3: **Comparison of tokenization vocabularies generated by BPE and Ladderpath methods.** Tokenization vocabularies produced by BPE (100 merge iterations) from the human genome (a) and from random sequences (b). The red star marks the total number of possible k-mers for the token length k, and the vertical red dashed line marks the gap between the number of possible k-mers and the subset recovered by this method. In contrast, panels (c) and (d) show the tokenization vocabularies generated by the Ladderpath method from the same human and random sequences, respectively.

It is important to emphasize that neither method is expected to recover all possible k-mers; the purpose of tokenization is to filter out uninformative patterns, and including every possible k-mer would defeat this purpose. From Fig. 3, we can see that, among the 64 possible 3-mers, BPE selects 22 and Ladderpath selects 28—neither achieves complete coverage, which is entirely expected. Nonetheless, the lower coverage observed for BPE is again likely attributable to its greedy merge mechanism, which can prevent certain 3-mers from ever appearing along its merge trajectory.

In summary, because of its inherently greedy design, BPE inevitably misses a subset of short k-mers and instead tends to favor the formation of longer tokens. This tendency can be problematic for biological sequences because it overlooks a crucial biological fact: many functional elements in DNA—such as transcription factor binding sites and splice site motifs—are built from highly conserved short nucleotide combinations [31]. As a result, BPE is intrinsically limited in its ability to capture biologically meaningful fine-grained signals. Finally, it is worth noting that this drawback is far less pronounced in natural language processing, where the basic character set is much larger than the four nucleotides of DNA (e.g., 26 letters in English and tens of thousands of characters in Chinese). Consequently, the limitations of BPE's greedy behavior become considerably more pronounced in genomic applications than in linguistic ones.

2.3 Evaluating Tokenization Effects in DNA Sequence Modeling

To comprehensively evaluate the performance of LTM, we benchmarked it against several representative DNA language models pre-trained on the human reference genome, including HyenaDNA [11], NT [15], DNABERT [13], and GROVER [16]. For the NT model, we specifically used the NT-HumanRef (500M) version, which was pre-trained exclusively on the human reference genome, making it the most appropriate choice for comparison in our setting.

LTM achieved superior performance in 17 of 21 downstream tasks, indicating robust generalization and predictive accuracy (Tab. 1). In particular, on histone modification prediction (Tasks 12-21), which reflects key epigenetic regulatory processes, LTM substantially surpassed all baseline models. Importantly, LTM and GROVER share the same architecture and training protocol; thus, the performance improvement arises solely from the tokenization strategy. This controlled comparison demonstrates that Ladderpath tokenization confers a significant advantage over the BPE-based approach for genomic sequence modeling.

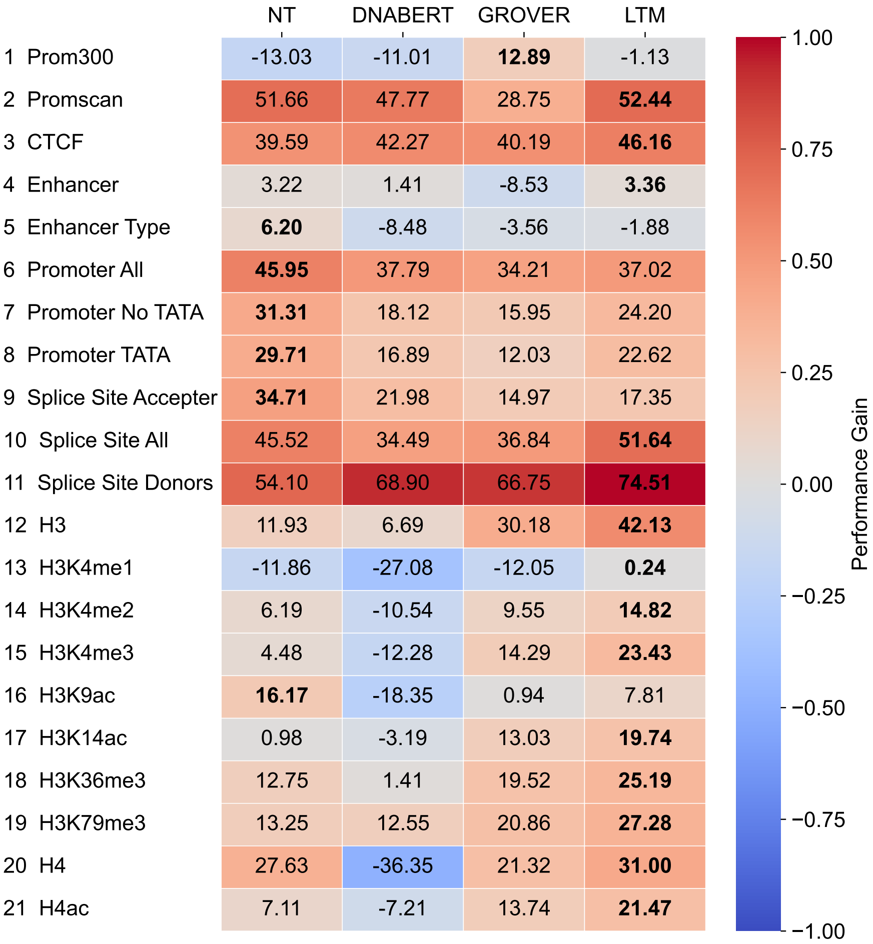
Lastly, to rigorously assess whether a language model truly learns the contextual syntax of DNA sequences—rather than relying solely on token frequency statistics—we constructed a TF-IDF (*Term Frequency–Inverse Document Frequency*) model as a negative control. TF-IDF is a classic *bag-of-words* approach that represents each sequence as a feature vector based on the frequency of a token within a sequence (TF) and its rarity across the corpus (IDF), while completely ignoring token order and interrelationships. Thus, its performance reflects the baseline achievable using frequency information alone. In many genomic prediction tasks, functional and non-functional sequences (e.g., promoters vs. background regions) can differ substantially in token frequency, enabling models to perform well simply by exploiting this imbalance rather than genuinely learning contextual meaning [32]. A truly effective DNA language model, however, should improve performance by capturing context and underlying biological knowledge. Accordingly, we propose the *performance gain*—the improvement of a language model relative to the TF-IDF baseline on downstream classification tasks (see Methods)—as a measure of tokenization quality. By comparing a model's performance with TF-IDF, we quantify how effectively it captures sequence context and higher-order biological patterns. The comparative results are shown in Fig. 4.

Table 1: Comparative performance of DNA language models on 21 genomic prediction tasks.

	Task	**HyenaDNA**	**NT**	**DNABERT**	**GROVER**	**LTM**
		1M	500M	118M	86M	86M
1	Prom300	77.26	84.61	81.38	**99.62**	96.13
2	Promscan	83.48	85.91	85.30	63.18	**87.02**
3	CTCF	13.18	57.03	63.46	60.13	**64.34**
4	Enhancer	51.25	57.00	55.00	58.50	**59.75**
5	Enhancer Type	42.47	50.27	52.72	46.47	**54.14**
6	Promoter All	90.83	90.02	92.90	91.47	**93.04**
7	Promoter No TATA	90.64	89.83	92.79	92.51	**93.55**
8	Promoter TATA	89.05	88.24	90.88	88.87	**90.98**
9	Splice site all	83.77	94.50	**96.75**	90.22	92.68
10	Splice site donor	90.08	95.04	**95.09**	86.82	93.10
11	Splice site accepter	85.17	92.30	**93.51**	87.65	93.24
12	H3	74.08	72.01	64.06	76.75	**80.76**
13	H3K4me1	38.07	38.83	35.83	45.86	**50.92**
14	H3K4me2	32.49	35.09	32.23	46.03	**46.11**
15	H3K4me3	24.38	25.35	20.71	43.75	**49.32**
16	H3K9ac	47.61	39.15	41.38	56.28	**58.33**
17	H3K14ac	38.70	33.10	35.28	53.68	**56.79**
18	H3K36me3	45.39	42.43	43.72	56.39	**58.31**
19	H3K79me3	55.48	49.20	52.01	58.77	**65.19**
20	H4	75.91	73.79	72.28	76.66	**81.52**
21	H4ac	34.93	29.88	30.79	51.50	**54.81**

*The downstream genomic prediction tasks are designed to evaluate a model's capacity to extract biologically meaningful patterns from DNA sequences and can be grouped into three categories: (1) Transcriptional regulation (Tasks 1-8): identification of gene regulatory elements, including promoters, enhancers, and binding sites for the chromatin organizer protein CTCF; (2) RNA splicing (Tasks 9-11): prediction of splice sites that demarcate intron-exon boundaries; and (3) Epigenetics (Tasks 12-21): prediction of histone proteins (H3, H4) and their post-translational modifications (e.g., H3K4me1/3, H3K9ac, H3K36me3), which serve as key epigenetic markers associated with chromatin accessibility and gene regulatory activity.

Across the 21 downstream tasks, LTM achieved the largest performance gain among all evaluated models in 14 of them. This demonstrates that LTM does not merely enable the model to memorize high-frequency functional "keywords." Instead, by providing a vocabulary that better reflects the intrinsic compositional and structural patterns of DNA, it advances the representation of biological sequences from frequency-based pattern recognition to genuine syntactic and contextual understanding. This widening performance gap provides compelling evidence that our



	NT	DNABERT	GROVER	LTM
1 Prom300	-13.03	-11.01	**12.89**	-1.13
2 Promscan	51.66	47.77	28.75	**52.44**
3 CTCF	39.59	42.27	40.19	**46.16**
4 Enhancer	3.22	1.41	-8.53	**3.36**
5 Enhancer Type	**6.20**	-8.48	-3.56	-1.88
6 Promoter All	**45.95**	37.79	34.21	37.02
7 Promoter No TATA	**31.31**	18.12	15.95	24.20
8 Promoter TATA	**29.71**	16.89	12.03	22.62
9 Splice Site Accepter	**34.71**	21.98	14.97	17.35
10 Splice Site All	45.52	34.49	36.84	**51.64**
11 Splice Site Donors	54.10	68.90	66.75	**74.51**
12 H3	11.93	6.69	30.18	**42.13**
13 H3K4me1	-11.86	-27.08	-12.05	**0.24**
14 H3K4me2	6.19	-10.54	9.55	**14.82**
15 H3K4me3	4.48	-12.28	14.29	**23.43**
16 H3K9ac	**16.17**	-18.35	0.94	7.81
17 H3K14ac	0.98	-3.19	13.03	**19.74**
18 H3K36me3	12.75	1.41	19.52	**25.19**
19 H3K79me3	13.25	12.55	20.86	**27.28**
20 H4	27.63	-36.35	21.32	**31.00**
21 H4ac	7.11	-7.21	13.74	**21.47**

Fig. 4: **Performance gains over the TF-IDF baseline across downstream tasks.** Heatmap of performance gains for NT, DNABERT, GROVER, and LTM across 21 tasks. HyenaDNA is excluded because it lacks a token-based language modeling framework, making a TF-IDF baseline unsuitable for fair comparison. Performance gain is defined as improvement over a TF-IDF baseline constructed with the same vocabulary, reflecting the contribution of contextual modeling. Red indicates a positive contribution (deeper red represents larger gains), while blue indicates the opposite.

tokenization strategy allows language models to capture deeper sequence relationships beyond surface-level frequency statistics.

2.4 Biological Meaning and Modular Structure of Tokens

2.4.1 Structure in Token Embedding Space

In language models, high-quality embeddings map semantically related tokens to proximate regions in the embedding space. When tokens with similar meanings or functions (e.g., color words or verbs) form coherent clusters, it reflects the model's ability to capture deep semantic relationships among tokens. Conversely, random or uninformative embeddings produce diffuse, unstructured distributions [33].

To examine whether DNA language models similarly organize their token representations, we applied UMAP to visualize the embedding vectors of all tokens in the vocabularies of NT, DNABERT, GROVER, and LTM (Fig. 5). HyenaDNA was excluded because it operates at the single-nucleotide level, resulting in only four tokens and therefore lacking meaningful clustering capacity.

The embedding space learned by LTM displays clear structural organization, with multiple well-separated clusters. In contrast, the embedding spaces of the other models appear more diffuse and show limited evidence of distinct cluster formation. These results suggest that Ladderpath tokenization produces tokens with stronger intrinsic semantic associations, supplying the model with a more structured and informative vocabulary. This richer representational basis facilitates learning higher-order sequence "grammar," enhances interpretability, and further supports the effectiveness of our tokenization strategy.

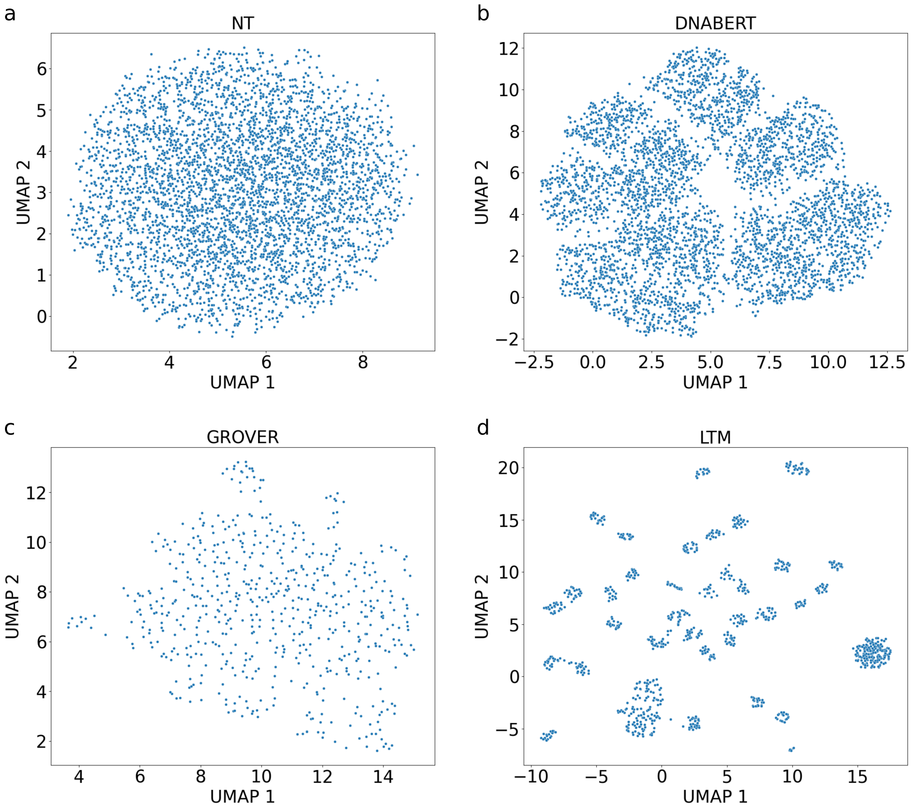


Fig. 5: **UMAP visualization of token embedding spaces.** UMAP was applied to the token embeddings of four models—(a) NT, (b) DNABERT, (c) GROVER, and (d) LTM—to examine the clustering structure within their embedding spaces. UMAP parameters: $n_{\text{neighbors}} = 5$, $\text{min}_{\text{dist}} = 0.5$.

2.4.2 Ladderpath Tokenization Reveals Modular and Compositional Structure in Genomic Sequences

Transcription factors (TFs) play a central role in gene regulation by binding to specific DNA sequences and modulating transcription. Their binding motifs are short DNA sequences that TFs recognize and bind, thereby encoding core principles of DNA-protein interactions. Consequently, an effective tokenizer vocabulary should be able to represent the diversity of TF binding motifs. To test this hypothesis, we evaluated the enrichment of known motifs from the JASPAR database [34] across different tokenizer vocabularies (see Methods for details). Counterintuitively, TF motifs are overrepresented in the GROVER vocabulary (9.46% known motifs vs. a background frequency of 6.56%; Fig. 6a) but underrepresented in the LTM vocabulary (5.47%, Fig. 6a), indicating that the performance of Ladderpath tokenization is not attributable to simple motif enrichment.

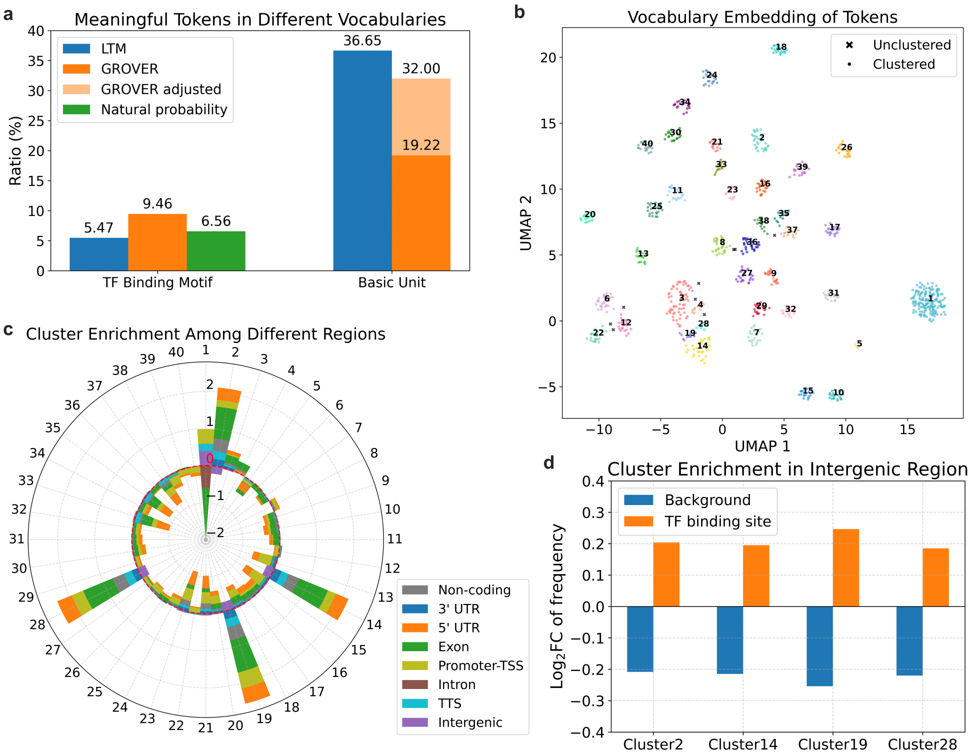


Fig. 6: **Ladderpath tokenization captures modular regulatory units and functional sequence organization.** (a) Frequency of TF binding motifs and their basic units in different vocabularies. (b) UMAP visualization of clustered LTM vocabulary embeddings. (c) Cluster enrichment in different genomic regions. The radius represents log2FC of different clusters in each genomic element. (d) Cluster enrichment in the entire intergenic region (blue) and intergenic region with TF binding sites.

We hypothesize that TF motifs exhibit hierarchical structure composed of recurrent basic sequence units that assemble into higher-order motifs, and that a tokenizer's ability to identify these fundamental units determines downstream performance. Analysis of 1,622 non-redundant motifs revealed that 36.68% can be reconstructed by recombining only 562 basic units, whereas this proportion drops to 13.47% in length-matched random sequences (see Methods). This demonstrates that TF motifs possess substantial inherent modularity. Notably, the LTM vocabulary captured 206 basic units (36.65%, Fig. 6a)—nearly twice as many as GROVER (108 units, 19.22%)—and remained higher after adjusting for vocabulary size (32.0%, adjusted). These results suggest that Ladderpath tokenization represents motifs not by memorizing complete instances, but by encoding general, combinatorially flexible subunits that compose diverse sequence patterns. This enables the model to generalize beyond known motifs and discover previously unrecognized sequence features.

Next, we examined whether tokens in the LTM vocabulary are functionally organized within the genome. Clustering analysis of the LTM embedding space identified 40 distinct clusters (Fig. 6b; see Methods). To assess biological relevance, we evaluated cluster enrichment across diverse genomic elements, including function-associated regions (e.g., promoters and exons) and traditionally considered nonfunctional intergenic regions. Clusters 2, 14, 19, and 28 were consistently enriched in regions of clear biological importance, whereas intronic regions showed near-zero enrichment (log2FC $\approx$ 0) across clusters (Fig. 6c). Enrichment was most pronounced in the 5' UTR and promoter-TSS regions, which are rich in regulatory signals, while intergenic regions showed the opposite trend. Interestingly, despite overall cluster depletion in intergenic sequences, these clusters remained significantly enriched within intergenic TF binding sites (Fig. 6d), further supporting their functional relevance.

In summary, these analyses demonstrate that Ladderpath tokenization captures modular sequence architecture at multiple levels. It identifies recurrent units within TF motifs and organizes functionally coherent token clusters, thereby providing a structured vocabulary that enhances interpretability and enables data-driven discovery of important genomic sequence patterns.

2.4.3 Zero-Shot Clustering of Functional Genomic Sequences

We next asked whether a tokenizer that captures modular genomic structure also induces zero-shot organization of functional sequence classes in embedding space. To examine this, we moved beyond token-level analysis and evaluated whole-sequence embeddings. We fed full genomic functional sequences into each model, extracted the [CLS] token embedding (the first-token representation commonly used as a global sequence embedding), and applied UMAP for dimensionality reduction and visualization. This enabled a direct comparison of the embedding distributions associated with promoters, enhancers, and histone marks.

Without any fine-tuning, LTM separated these categories into clear and coherent clusters in embedding space (Fig. 7). Although NT also exhibited meaningful cluster structure, it does so with a substantially larger model. By contrast, GROVER and DNABERT showed largely overlapping and poorly resolved distributions, indicating limited zero-shot discrimination capacity.

These results suggest that a tokenization scheme aligned with the compositional logic of genomic sequences can endow a model, through pre-training alone, with an emergent ability to organize high-level genomic functional classes. Rather than relying on explicit supervision or motif memorization, LTM appears to acquire a useful inductive bias toward functional genome organization. This unsupervised property provides a strong foundation for downstream supervised tasks, supporting efficient generalization and biologically informed representation learning.

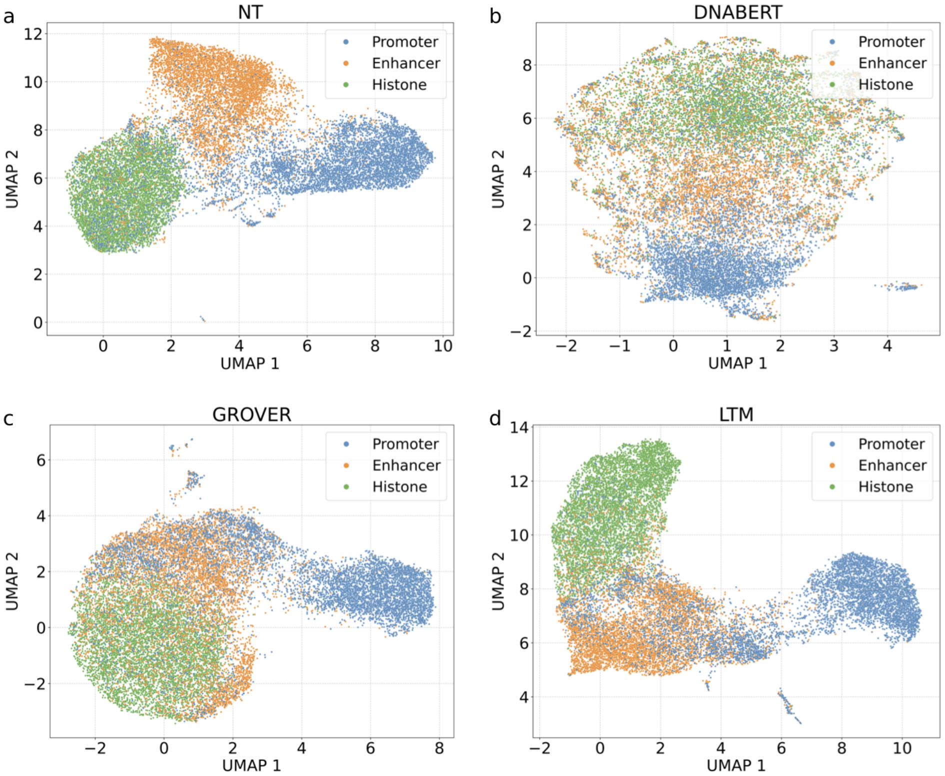


Fig. 7: **Zero-shot embedding structure of genomic functional elements.** UMAP visualizations of [CLS] sequence embeddings from four models—(a) NT, (b) DNABERT, (c) GROVER, and (d) LTM—for promoter, enhancer, and histone-mark sequences. LTM forms distinct, well-separated clusters without fine-tuning, whereas GROVER and DNABERT produce largely overlapping distributions; NT also shows clear structure but with a substantially larger model size.

3 Conclusion

In this work, we introduced Ladderpath, an information-theoretic tokenization strategy derived from Algorithmic Information Theory (AIT), and applied it to construct the DNA language model LTM. Our results demonstrate that reconceptualizing tokenization as a process of optimal information compression provides an effective means to capture the hierarchical and nested structures underlying the complex syntax and semantics of the genome. This approach represents a step beyond heuristic- or frequency-based segmentation, offering a principled framework that aligns more closely with the intrinsic organization of biological sequences.

The effectiveness of this framework is supported by extensive empirical evidence. Across 21 downstream tasks, LTM achieves superior performance in 17 (Tab. 1), surpassing not only the BPE-based GROVER model—which shares the same architecture—but also larger models such as NT-500M. These findings consistently support our central hypothesis: that a tokenization method aligned with the compositional and hierarchical nature of DNA enables the model to learn more generalizable and biologically meaningful representations. Furthermore, our performance gain analysis against a TF-IDF baseline shows that LTM's advantage does not stem from high-frequency token memorization, but from its ability to capture contextual and structural dependencies within genomic sequences (Fig. 4). In this sense, the Ladderpath strategy advances tokenization from surface-level frequency statistics toward a deeper modeling of contextual grammar in the genome, thereby expanding the model's effective "learning space."

Further analysis of the model reveals the biological significance of the representations it captures. The embedding space of LTM exhibits a highly organized structure in which functionally related tokens naturally cluster in semantic space (Fig. 5). Notably, LTM does not merely memorize complete transcription factor motifs but instead shows a preference for capturing combinatorial "modular units," reflecting the compositional logic of genomic organization (Fig. 6). Extending from motif-level modularity to sequence-level representation, LTM also organizes promoters, enhancers, and histone marks into distinct clusters without task-specific supervision (Fig. 7), demonstrating an emergent zero-shot understanding of genomic function. These findings also have broader implications. In an era dominated by models containing billions of parameters, the ability of our 86M-parameter LTM to outperform models several times larger underscores that improving data efficiency through a principled tokenization strategy may represent a more sustainable and interpretable path forward than indefinitely scaling model size.

Despite these encouraging results, several challenges remain. The computational cost of the Ladderpath algorithm is higher than that of frequency-based methods such as BPE, which can limit scalability to genome-wide datasets. Although we alleviated this issue through segmented computation, developing more efficient or parallelized implementations will be important for future improvement. In addition, this study focuses primarily on the human reference genome; extending the Ladderpath framework to multi-species data to construct a more generalizable pangenomic language model represents a promising direction for further exploration. Finally, future work should aim to interpret the learned representations in greater depth and connect them more directly to underlying biological mechanisms.

In summary, our work establishes Ladderpath tokenization as a theoretically grounded approach that enhances a model's ability to interpret DNA by providing a vocabulary that reflects the intrinsic modular and hierarchical organization of the genome. This framework moves toward a deeper and more systematic understanding of genomic information.

4 Methods

4.1 Ladderpath Approach

The Ladderpath approach is a computational framework for uncovering the generative structure of a target system (e.g., a sequence) by identifying the most efficient path to reconstruct it [25, 26]. Rather than segmenting a sequence based on local statistics, Ladderpath seeks the shortest generative path—the shortest path of operations required to construct the target sequence through the combination and reuse of repeated subsequences. This perspective emphasizes compositionality and hierarchical reuse, offering a global view of sequence organization.

This part provides a concrete example illustrating how a Ladderpath is computed (for detailed code implementations, please refer to our open-source GitHub repository). Consider the following three DNA sequences, each composed of the basic units A, T, G, and C: [′CACATCTGTCAATACGCACA′, ′GGAGGCTGCAGGGCTG′, ′CTGCAATACGTTGGGAGG′]. We treat these sequences collectively as the target system $\mathcal{X}$ and compute its Ladderpath. The resulting Ladderpath can be expressed in the form of a *partially ordered multiset* (POM) as: {T(5), G(5), A(4), C(3) // CA(3), GG(2), TG // CTG(2), CAATACG, CACA, GGA // GGCTG} (An equivalent representation is the *laddergraph*, shown in Fig. 1a.) This POM notation can be interpreted as follows:

1. Each element in the POM is a repetitive subsequence, which we refer to as a *ladderon*.
2. Ladderons separated by commas (,) belong to the same level of the construction; they are parallel components with no precedence constraints (e.g., CA and TG).
3. Ladderons separated by double slashes (//) belong to different levels, forming a hierarchical structure. Ladderons in a higher level must be constructed after those in preceding levels. For example, CTG depends on TG, and thus appears in a subsequent layer.
4. The integer in parentheses following each ladderon denotes its *multiplicity*, i.e., the number of times it is reused throughout the entire construction process. This quantity is also used to rank ladderons by importance.

Thus, the Ladderpath not only reveals the nested and hierarchical relationships among ladderons but also explicitly computes how many times each ladderon is reused, a property essential for vocabulary construction. The laddergraph representation (Fig. 1a) encodes the same information as the POM and provides a clearer visualization of these hierarchical dependencies.

In summary, the Ladderpath framework provides a complementary perspective to existing tokenization methods by modeling sequences through their globally optimal generative paths. By identifying substructures that are repeatedly used within this shortest-path reconstruction, it captures hierarchical and compositional relationships that are not explicitly encoded by frequency-based approaches (e.g., BPE). This perspective resonates with biological observations that functional genomic regions often exhibit recurrent and compositional patterns. Motivated by this alignment, we adapt the Ladderpath approach to DNA tokenization, using these reusable ladderons to construct a vocabulary that more faithfully reflects the intrinsic compositional logic of the genome.

4.2 Detailed Experimental Setup

We used the human reference genome hg19 as the primary dataset. Preprocessing began with a stringent filtering procedure that retained only contigs composed exclusively of the four canonical nucleotides (A, C, G, T), removing any sequences containing ambiguous characters. The resulting clean nucleotide sequences were then tokenized using the vocabulary derived from the Ladderpath approach (the procedure for constructing this vocabulary is described in detail in Section 2.1 of the main text).

To promote robustness to variable sequence lengths during pre-training, we adopted a bimodal length sampling strategy. Specifically, 50% of all sequence windows were fixed at 510 tokens, approximating the model's maximum effective input length (512 tokens minus the [CLS] and [SEP] tokens), while the remaining 50% were assigned a randomly sampled integer length drawn uniformly from the interval [20, 510). The complete corpus was then randomly partitioned into an 80% training set and a 20% held-out test set to ensure unbiased assessment of the model's generalization performance.

Our LTM follows the architecture described in the GROVER framework and was pre-trained using the Masked Language Modeling (MLM) objective. The model employs an embedding dimension of 768, with 12 attention heads and 12 transformer layers, amounting to approximately 86 million parameters. Inputs consist of sequences with a maximum length of 510 tokens. In addition to the genomic vocabulary, the tokenizer includes five special tokens: [CLS], [PAD], [UNK], [SEP], and [MASK].

During MLM pre-training, we masked 15% of all input tokens. Of these masked positions, 80% were replaced with the [MASK] token, 10% with a random token sampled from the vocabulary, and 10% were left unchanged. For model training, we extracted over five million sequence samples from the human reference genome. Training was performed using a batch size of 600, using the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1 \times 10^{-6}$, weight decay = 0.01). We applied a cosine annealing learning-rate schedule with a linear warmup over the initial 10% of training steps.

4.3 Downstream Tasks

For downstream evaluation, we trained on the Nucleotide Transformer (NT) benchmark for 20 epochs. The learning rate was linearly warmed up during the first epoch and then gradually decayed to zero over the remaining epochs. All models were fully fine-tuned with a batch size of 32 and a learning rate of 5×10^{-5}, using the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$), weight decay = 0.01).

Following findings from HyenaDNA, which reported optimal performance when the pre-training sequence length is 2-4 times longer than that of downstream inputs, we selected the official HyenaDNA-tiny model for comparison, as most downstream sequences consist of only a few hundred tokens [11]. To ensure fairness, we also included the NT-500M-human model—pre-trained specifically on the human reference genome—in our comparisons. Additionally, prior work on DNABERT demonstrated that pre-training with 6-mer tokenization yields the best downstream performance.

Therefore, we included a DNABERT model trained with 6-mer tokenization as another baseline.

For the CTCF, Prom300, and Promscan tasks, we followed the GROVER experimental setup and used the same training parameters as in the NT benchmark to ensure consistency across evaluations.

4.4 TF-IDF

To assess whether our language model learns information beyond simple token-frequency statistics, we constructed a TF-IDF (Term Frequency-Inverse Document Frequency) baseline model as a negative control. This control experiment quantifies the upper bound of performance that can be achieved using token frequencies alone. Specifically, we treated each DNA sequence as a “document” and the tokens produced by our Ladderpath tokenization method as “words.” We computed TF-IDF feature vectors for all sequences in the training set using the *TfidfVectorizer* implementation from the *scikit-learn* library. Based on these vectors, we trained a Random Forest classifier and tuned its performance through a grid search over a range of 100 to 2,000 estimators.

It is crucial to emphasize that this baseline model has no access to token order, syntax, or any contextual dependencies; its decisions rely exclusively on token frequency information. Therefore, comparing the performance of this TF-IDF model with that of our language model on downstream classification tasks allows us to quantify the true performance gains attributable to contextual learning.

4.5 Analysis of TF Binding Motifs

TF binding motifs were retrieved from the JASPAR database. Given that the LTM vocabulary primarily consists of 4-mers and 5-mers, only TF motifs with length 4 or 5 were used in the following analyses. For each vocabulary, we retrieved its subset containing 4/5-mers and calculated the proportion of TF motifs in that subset. The ratio of motifs among all possible 4/5-mers was set as natural probability. Next, to detect basic units of TF binding motifs, we generated rotations of 1,622 non-redundant motifs in JASPAR and compared the rotations to the original sequences of motifs. A rotation of a string s of length n is generated by splitting it into two substrings s_1 and s_2, and then swapping s_1 and s_2 to produce a new string. For a motif s in the JASPAR database, if one of its rotations is also a TF binding motif, the two substrings s_1 and s_2 were defined as two basic units.

4.6 Functional Analysis of Token Clusters

The embeddings generated by the LTM were clustered by the DBSCAN method, with the following parameters: eps = 0.5, min samples = 5. Information about genomic elements within the human hg19 genome was obtained based on Homer's genomic annotation file (hg19.basic.annotation) [35]. Subsequently, the frequency of tokens from each cluster in different genomic regions was calculated and compared to the genome-wide background frequency to obtain log2FC values. ChIP-seq peaks retrieved from the ENCODE project [36] were utilized to define TF binding sites, which were annotated using Homer to identify TF binding sites in intergenic regions.

Author Contributions

Conceptualization, Y.L. and D.Z.; methodology, Y.L., Y.W. and J.W.; software, Y.L., Y.W., J.W., F.Z. and H.Z.; validation, Y.W., J.W. and J.Z.; formal analysis, Y.W., J.Z. and ZW.D.; investigation, Y.W., J.W., F.Z., H.Z., ZW.D., D.Z., and Y.L.; resources, Y.L., ZW.D., ZR.D. and D.Z.; data curation, Y.W., J.W., and J.Z.; writing—original draft preparation, Y.W., J.W., J.Z., F.Z., H.Z., ZW.D., ZR.D., D.Z., and Y.L.; visualization, Y.W., J.W., J.Z., ZW.D., and Y.L.; supervision, ZW.D., ZR.D., D.Z. and Y.L.; project administration, Y.L. and D.Z.; funding acquisition, Y.L., ZW.D. and D.Z. All authors read and approved the final manuscript.

Data Availability

The complete pre-training corpus and gene regulatory sequence datasets are available at https://github.com/wyp178686/LTM_Data. For downstream evaluation, datasets were obtained from the following sources: the Nucleotide Transformer benchmarks hosted on Hugging Face (https://huggingface.co/spaces/InstaDeepAI/nucleotide_transformer_benchmark); the Prom300, Promscan, and CTCF datasets from Zenodo (https://zenodo.org/records/13374192); and the non-redundant position frequency matrices (PFMs) from the JASPAR database (https://jaspar.elixir.no/download/data/2024/CORE/JASPAR2024_CORE_non-redundant_pfms_jaspar.zip).

Code Availability

The complete source code required to reproduce all findings in this study is available on GitHub at https://github.com/yuernestliu/LTM. The core implementation of the Ladderpath algorithm is provided through the https://github.com/yuernestliu/lppack repository.

Declaration of Competing Interest

The authors declare no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Acknowledgments

We gratefully acknowledge Prof. Haijun Zhou for his insightful suggestions and discussions. This study was funded by the National Natural Science Foundation of China (Grant No. 12205012 to Y.L., Grant No. 12471475 to D.Z., Grant No. 12575033 to Z.D.), Basic and Applied Basic Research Foundation of Guangdong Province (Grant No. 2025A1515012923 to Y.L.), Shenzhen Medical Academy of Research and Translation (Grant No. D2401022 to ZW.D.) and Southern University of Science and Technology (Grant No. R011376020 to ZW.D.).

References

[1] Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 **1**(2) (2023)

[2] Vu, M.H., Akbar, R., Robert, P.A., Swiatczak, B., Sandve, G.K., Greiff, V., Haug, D.T.T.: Linguistically inspired roadmap for building biologically reliable protein language models. Nature Machine Intelligence **5**(5), 485–496 (2023)

[3] Gastaldi, J.L., Terilla, J., Malagutti, L., DuSell, B., Vieira, T., Cotterell, R.: The foundations of tokenization: Statistical and computational concerns. arXiv preprint arXiv:2407.11606 (2024)

[4] Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)

[5] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)

[6] Kudo, T.: Subword regularization: Improving neural network translation models with multiple subword candidates. arXiv preprint arXiv:1804.10959 (2018)

[7] Eapen, B.R.: Genomic tokenizer: Toward a biology-driven tokenization in transformer models for DNA sequences. bioRxiv, 2025–04 (2025)

[8] Marin, F.I., Teufel, F., Horlacher, M., Madsen, D., Pultz, D., Winther, O., Boomsma, W.: Bend: Benchmarking DNA language models on biologically meaningful tasks. arXiv preprint arXiv:2311.12570 (2023)

[9] Dotan, E., Jaschek, G., Pupko, T., Belinkov, Y.: Effect of tokenization on transformers for biological sequences. Bioinformatics **40**(4), 196 (2024)

[10] Nguyen, E., Poli, M., Durrant, M.G., Kang, B., Katrekar, D., Li, D.B., Bartie, L.J., Thomas, A.W., King, S.H., Brixi, G., *et al.*: Sequence modeling and design from molecular to genome scale with Evo. Science **386**(6723), 9336 (2024)

[11] Nguyen, E., Poli, M., Faizi, M., Thomas, A., Wornow, M., Birch-Sykes, C., Massaroli, S., Patel, A., Rabideau, C., Bengio, Y., *et al.*: Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. Advances in neural information processing systems **36**, 43177–43201 (2023)

[12] Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J.R., Grabska-Barwinska, A., Taylor, K.R., Assael, Y., Jumper, J., Kohli, P., Kelley, D.R.: Effective gene expression prediction from sequence by integrating long-range interactions. Nature methods **18**(10), 1196–1203 (2021)

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

[14] Zhou, Z., Ji, Y., Li, W., Dutta, P., Davuluri, R., Liu, H.: DNABERT-2: Efficient foundation model and benchmark for multi-species genome. arXiv preprint arXiv:2306.15006 (2023)

[15] Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., Lopez Carranza, N., Grzywaczewski, A.H., Oteri, F., Dallago, C., Trop, E., Almeida, B.P., Sirelkhatim, H., *et al.*: Nucleotide transformer: building and evaluating robust foundation models for human genomics. Nature Methods **22**(2), 287–297 (2025)

[16] Sanabria, M., Hirsch, J., Joubert, P.M., Poetsch, A.R.: DNA language model grover learns sequence context in the human genome. Nature Machine Intelligence **6**(8), 911–923 (2024)

[17] Qiao, L., Ye, P., Ren, Y., Bai, W., Liang, C., Ma, X., Dong, N., Ouyang, W.: Model decides how to tokenize: Adaptive DNA sequence tokenization with mxdna. Advances in Neural Information Processing Systems **37**, 66080–66107 (2024)

[18] Bostrom, K., Durrett, G.: Byte Pair Encoding is suboptimal for language model pretraining. arXiv preprint arXiv:2004.03720 (2020)

[19] Hofmann, V., Pierrehumbert, J.B., Schütze, H.: Superbizarre is not superb: Derivational morphology improves bert's interpretation of complex words. arXiv preprint arXiv:2101.00403 (2021)

[20] Medvedev, A., Viswanathan, K., Kanithi, P., Vishniakov, K., Munjal, P.,

Christophe, C., Pimentel, M.A., Rajan, R., Khan, S.: BioToken and BioFM–Biologically-Informed Tokenization enables accurate and efficient genomic foundation models. bioRxiv, 2025–03 (2025)

[21] Li, S., Wang, Z., Liu, Z., Wu, D., Tan, C., Zheng, J., Huang, Y., Li, S.Z.: VQDNA: Unleashing the power of vector quantization for multi-species genomic sequence modeling. arXiv preprint arXiv:2405.10812 (2024)

[22] Delétang, G., Ruoss, A., Duquenne, P.-A., Catt, E., Genewein, T., Mattern, C., Grau-Moya, J., Wenliang, L.K., Aitchison, M., Orseau, L., et al.: Language modeling is compression. arXiv preprint arXiv:2309.10668 (2023)

[23] Kolmogorov, A.N.: On tables of random numbers. Sankhyā: The Indian Journal of Statistics, Series A, 369–376 (1963)

[24] Liu, Y., Mathis, C., Bajczyk, M.D., Marshall, S.M., Wilbraham, L., Cronin, L.: Exploring and mapping chemical space with molecular assembly trees. Science Advances **7**(39), 2465 (2021)

[25] Liu, Y., Di, Z., Gerlee, P.: Ladderpath approach: how tinkering and reuse increase complexity and information. Entropy **24**(8), 1082 (2022)

[26] Zhang, Z., Liu, C., Zhu, Y., Peng, L., Qiu, W., Tang, Q., Liu, H., Zhang, K., Di, Z., Liu, Y.: Evolutionary tinkering enriches the hierarchical and nested structures in amino acid sequences. Physical Review Research **6**(2), 023215 (2024)

[27] Li, S., Peng, L., Chen, L., Que, L., Kang, W., Hu, X., Ma, J., Di, Z., Liu, Y.: Discovery of highly bioactive peptides through hierarchical structural information and molecular dynamics simulations. Journal of Chemical Information and Modeling **64**(21), 8164–8175 (2024)

[28] Xu, Z., Zhu, Y., Hong, B., Wu, X., Zhang, J., Cai, M., Zhou, D., Liu, Y.: Correlating measures of hierarchical structures in artificial neural networks with their performance. npj Complexity **1**(1), 15 (2024)

[29] Shapiro, J.A., Sternberg, R.: Why repetitive DNA is essential to genome function. Biological Reviews **80**(2), 227–250 (2005)

[30] Ouyang, Z., Wang, C., She, Z.-S.: Scaling and hierarchical structures in DNA sequences. Physical review letters **93**(7), 078103 (2004)

[31] Gutierrez-Vasques, X., Bentz, C., Sozinova, O., Samardzic, T.: From characters to words: the turning point of bpe merges. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 3454–3468 (2021)

[32] Orlov, Y.L., Orlova, N.G.: Bioinformatics tools for the sequence complexity estimates. Biophysical reviews **15**(5), 1367–1378 (2023)

[33] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (long and Short Papers), pp. 4171–4186 (2019)

[34] Bryne, J.C., Valen, E., Tang, M.-H.E., Marstrand, T., Winther, O., Piedade, I., Krogh, A., Lenhard, B., Sandelin, A.: JASPAR, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update. Nucleic acids research **36**(suppl_1), 102–106 (2007)

[35] Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y.C., Laslo, P., Cheng, J.X., Murre, C., Singh, H., Glass, C.K.: Simple combinations of lineage-determining

transcription factors prime cis-regulatory elements required for macrophage and b cell identities. Molecular cell **38**(4), 576–589 (2010)

[36] Consortium, E.P., *et al.*: An integrated encyclopedia of DNA elements in the human genome. Nature **489**(7414), 57 (2012)