Motif-aware Tokenization of the Genome: Towards Interpretable Modeling of Gene Regulation

Parmida Davarmanesh
Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
parmida@mit.edu

Joshua Pereira
Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
joshuagp@mit.edu

Adityanarayanan Radhakrishnan
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139
aradha@mit.edu

Ruochi Zhang
Broad Institute of MIT and Harvard
Cambridge, MA 02139
zhangruo@broadinstitute.org

Jimin Tan
New York University Grossman School of Medicine
New York, NY 10016
tanjimin@broadinstitute.org

Ashia Wilson
Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
ashia07@mit.edu

Bo Xia
Broad Institute of MIT and Harvard
Cambridge, MA 02139
xiabo@broadinstitute.org

Abstract

Genomic Language Models achieve strong performance on biological tasks but rely on tokenization methods that overlook the complexity of the genome. We introduce a biologically grounded tokenization strategy that partitions the DNA sequence into meaningful "words" based on transcription factor (TF) motifs. Embedding biological insight into vocabulary design preserves predictive power while potentially improving interpretability and computational efficiency. Proof-of-concept results demonstrate that motif-informed tokens generate representations that better capture the language of gene regulation, opening the door to models that are both highly predictive and capable of decoding the regulatory genomic grammar vital for drug discovery and precision medicine. The code for tokenization and model training is available at https://github.com/pdavar/Genome_tokenization.

1 Introduction

Just as modeling language at the level of words or subwords, rather than individual letters, provides semantic and computational advantages, studying the regulatory genome can benefit from identifying meaningful units beyond single base pairs [1, 2]. To uncover the "grammar" of this biological language, we must move from treating DNA as a simple string of letters toward recognizing genomic "words" that carry regulatory meaning. Unlike prokaryotic genomes, where most of the DNA is coding and regulation

2nd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences

is relatively straightforward, the mammalian genome is 98.5% non-coding with a far more complex regulatory grammar [3]. Gene expression in mammals is orchestrated by layered control from these expansive non-coding regions, making the study of regulatory "language" essential for deciphering gene programs and, ultimately, advancing drug discovery, precision medicine, and personalized therapies.

Most state-of-the-art genomic language models tokenize DNA at the base-pair level and use convolutional layers to extract features from one-hot–encoded sequences [4–11]. Although this strategy achieves strong predictive performance on downstream tasks, it has notable limitations. First, because base-pair tokenization does not shorten the sequence, and the attention mechanism used in these models scales quadratically with sequence length, such models are computationally expensive and often impractical to train from scratch in academic settings. The long input lengths also constrain context windows, reducing the ability to capture long-range promoter–enhancer interactions [12]. Second, in analogy to NLP—where single-character tokenization impairs performance by forcing models to infer higher-level structure from minimal context [13], genomic models must similarly "spell out" regulatory words from scratch, potentially hindering the learning of meaningful semantic relationships.

To address this, some methods tokenize into fixed-length k-mers (commonly k=3–5) [14, 15]. While codons are biologically meaningful in protein-coding regions, over 98% of the genome is non-coding yet crucial for regulation, making fixed-length k-mers arbitrary and poorly aligned with biological structure. Others have adopted Byte Pair Encoding (BPE) to create variable-length tokens [16, 17]. BPE reduces sequence length in a data-driven manner, but has three key drawbacks: the vocabulary size must be set arbitrarily; it assumes explicit token boundaries ("spaces") absent in DNA; and it merges tokens solely by frequency, without biological knowledge. Inspired by the dual role of tokenization in NLP—both reducing data sparsity by composing unseen sequences from a finite vocabulary and aligning tokens with meaningful linguistic units—we sought to design a biologically grounded tokenization strategy for the genome that better captures the regulatory grammar of DNA.

2 Methodolgy

To design biology-informed tokens, we used Transcription Factor (TF) binding motifs—short DNA patterns recognized by TFs—as tokens, since TFs are central to gene regulation and >92% of the human genome (>99% excluding centromeres and unmapped regions) is covered by at least one motif. Motif annotations were obtained from JASPAR 2022 [18] yielding 1684 motif features across both DNA strands. While these empirical annotations do not represent the entirety of the complex regulatory landscape of the human genome, they encode far more domain knowledge beyond just the base pair sequence. Our tokenization strategy can be summarized two steps. The first step is **binning**, which uses the distribution of the TF motifs across the genome to partition the DNA into "bins" (Figure 1a). Each bin is then represented as a 1684-dimensional vector, with entries corresponding to the probability of a given TF binding to the region of the DNA in that bin. Next, to reduce redundancy among correlated motifs, we computed a genome-wide non-linear correlation (i.e. Inter-Dependence Scores [19]) across all 1684 motif features and applied hierarchical **clustering**, yielding 871 motif clusters (features). Details of our tokenization strategy are provided in the appendix A.

Linguistic laws Early efforts treating biological sequences as natural language have revealed preliminary linguistic features, statistical 'grammars' akin to those in human languages, emerging from genomic data [20, 21]. A well-known example is Zipf's law, which states that word frequency is inversely proportional to its rank [22]. Early studies 20–30 years ago applied this metaphorically by treating short k-mers as "words," reporting Zipf-like distributions in DNA [23, 24]. However, these analyses were small-scale, and even random sequences showed similar patterns, suggesting artifacts of nucleotide composition rather than biologically meaningful units.

Building on this foundation, we evaluated our motif-based tokenization against k-mers and BPE approaches. Our tokens follow Zipf's law closely (Pearson $r \approx -0.99$), outperforming BPE600 (–0.77), BPE4096 (–0.94), and k-mers (figure 1c). Additionally, motif coverage scores—computed via correlations between known TF motif PFMs and one-hot representations—showed that our method captures motifs better than others (figure 1d).

Model-agnostic visualization of genomic tokens Unlike one-hot or BPE encodings, our tokens are biologically interpretable even prior to modeling: each motif feature represents the probability of TF binding in that genomic region. This allows direct comparison of token similarities, a property lacking

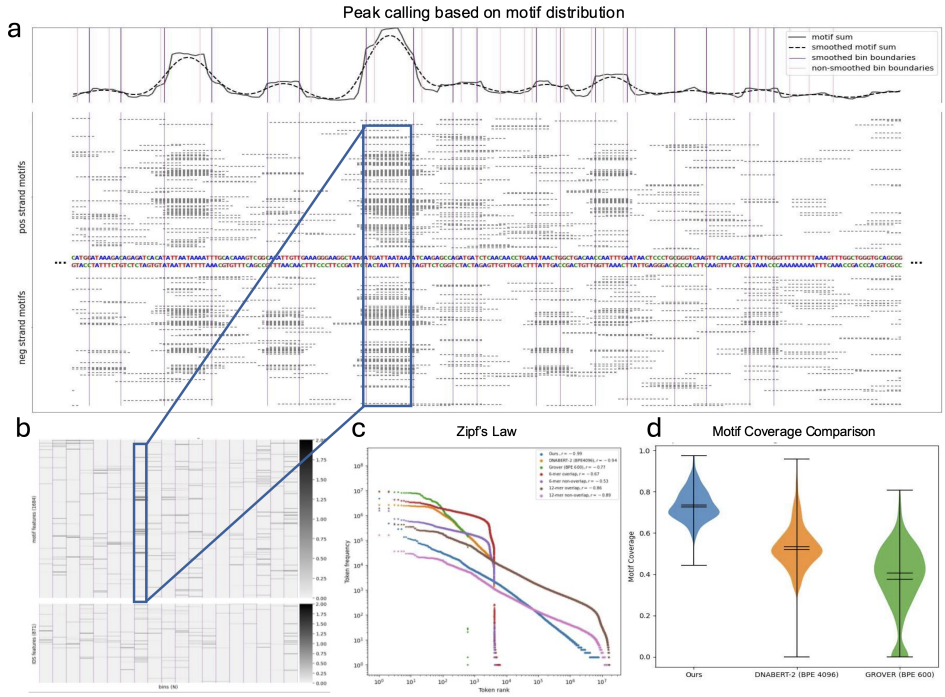

Figure 1: Motif-driven tokenization methodology and the linguistic laws. (a) partitioning the genome into bins using peak calling on motif sums. (b) constructing tokens by first stacking the motif-binding probabilities in each respective bins, then clustering similar motif features together. (c) Zipf's law (d) motifs capture comparison of common genome tokenizers

in other tokenization methods. Figure 2 shows a UMAP of all tokens on chromosome 1, colored by Louvain clusters (details in Appendix A).

3 Genomics Language Modeling

Recent advances in sequence-to-function genomic models offer a powerful way to model biological function using only the DNA sequence as input, providing scalable alternatives to costly experimental approaches [4–11]. These models can help interpret regulatory elements, prioritize disease-associated variants, and support synthetic biology. To test the utility of our tokenization strategy in sequence-to-function modeling, we sought to predict genome accessibility (obtained by ATAC-seq) from the DNA sequence. We compared the performance of a deep learning architecture trained using both our tokens and one-hot tokens. Figure 3a illustrates the architecture choices for these experiments where we tested both convolutional and transformer-based models. While the one-hot encoded model achieved a slightly higher correlation (0.73) compared to our tokenization method (0.68), qualitative evaluation of the predicted tracks indicates that our model effectively captures peak structures (Figures 3 b). Since these architectures are not optimized for learning from high dimensional sparse features like ours, unlocking the full potential of our tokenization method in the context of sequence-to-function modeling requires design of new model architectures which is beyond the scope of this work, yet a promising future direction to explore. Implementation details as well as additional results are provided in the appendix A.

To measure the interpretability of the models trained using both our and one-hot tokens, we conducted a systematic evaluation of DeepLift and Input x Gradient [25]. DeepLIFT attributes a model's prediction by comparing each input feature's activation to a reference input and propagating the contribution differences back to the input, while Input × Gradient attributes importance by multiplying each input feature's value by the gradient of the output with respect to that input, highlighting features where small changes strongly affect the prediction. Using a benchmarking dataset consisting of 41 TF binding sites [26], we obtained matching scores for each TF at a given genomic locus, and ranked the matching scores to obtain AUC for each TF (figure 3 c). Further analysis and evaluation metrics are provided

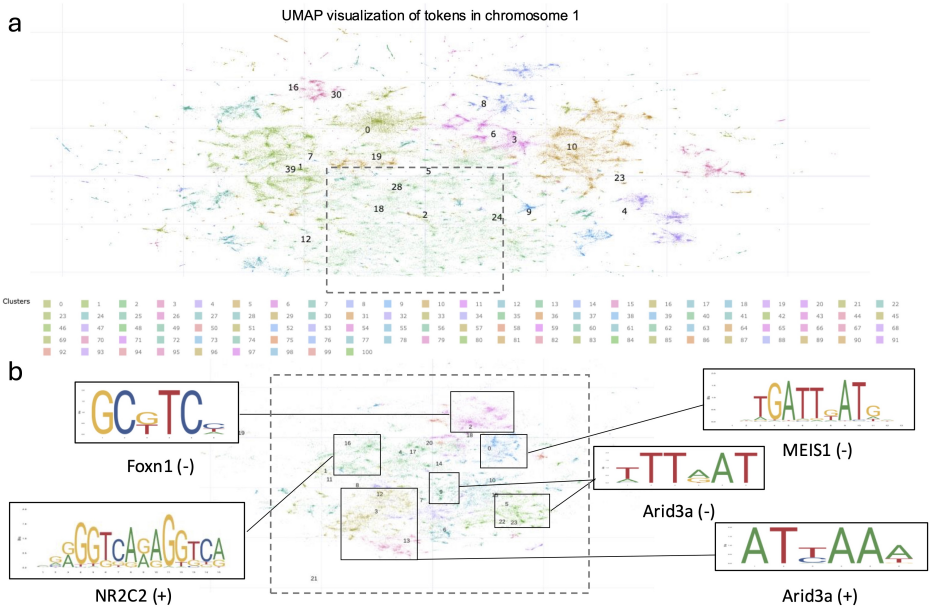

Figure 2: Token visualization (a) UMAP visualization of tokens, clustered by their sequence similarity. (b) Prominent motifs in sub-clusters derived from megacluster 2 in (a).

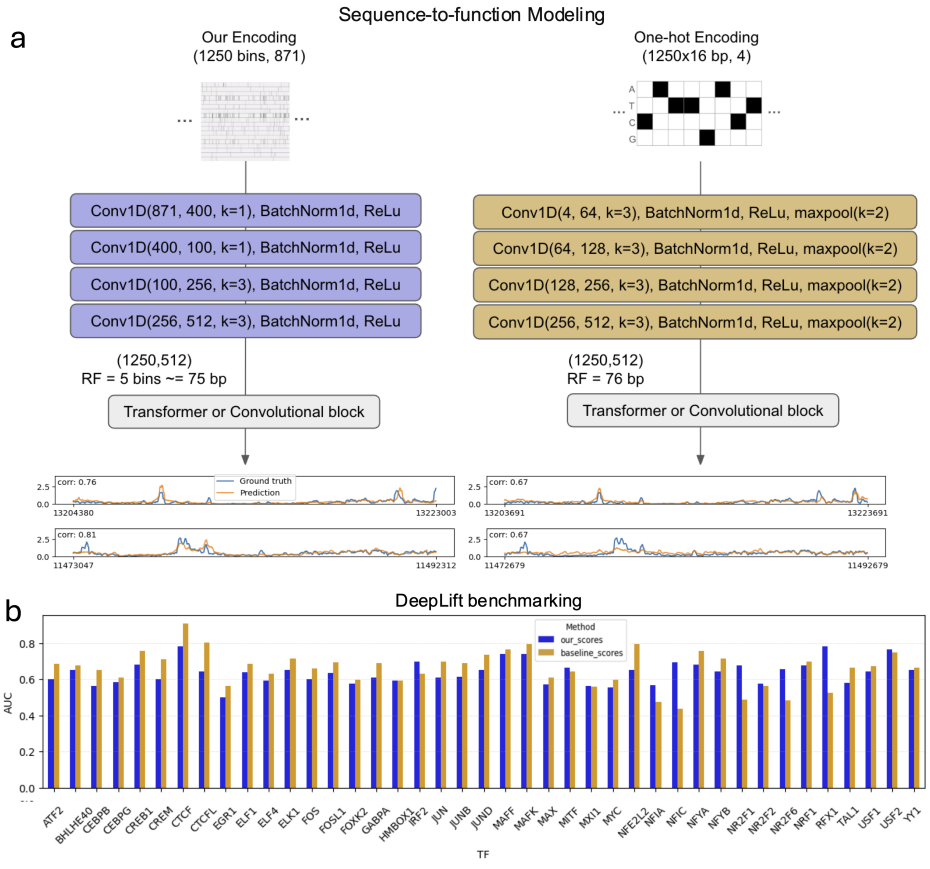

Figure 3: Sequence-to-function modeling. (a) The architectures for our (left) and one-hot (right) models. (b) Motif detection benchmarking using attribution scores taken from DeepLift applied to models trained on K562 cell lines.

in the appendix. We can see that for both DeepLift and Input x Gradient, our tokenization method performs similarly to the one hot tokenization for most of the TFs.

4 Conclusion and future work

Here, we presented the proof-of-concept of a dynamic tokenization framework of the genome, focusing on incorporating regulatory genomics domain knowledge through TF motifs. We showed that our biologically grounded genomic tokens not only offer a natural way of visualizing the genomic in low-dimensional manifolds, but importantly, they follow clear linguistic laws, providing strong evidence of DNA as a natural language hypothesis beyond the raw observation through K-mer analysis.

Even though our current tokenization procedure is not complete - as it relies on our incomplete empirical understanding of the motifs - we were able to achieve comparable performance and interpretability to one-hot tokenization in terms of sequence-to-function modeling while our approach offers a computational efficiency advantage due to shortening the sequence length. This highlights the potential of our approach and how we may be able to achieve better outcomes through enhanced empirical knowledge and designing customized architectures for these tokens.

References

[1] Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast wordpiece tokenization. *arXiv preprint arXiv:2012.15524*, 2020.

[2] Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91, 2022.

[3] Eric S. Lander, Lauren M. Linton, Bruce Birren, Chad Nusbaum, Michael C. Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michelle Doyle, William FitzHugh, Robert Funke, Daniel Gage, Kevin Harris, Anthony Heaford, Jody Howland, Larry Kann, Janet Lehoczky, Rita LeVine, Peter McEwan, Kevin McKernan, Jim Meldrim, Jill P. Mesirov, Christopher Miranda, William Morris, Jason Naylor, Christopher Raymond, Mary Rosetti, Raquel Santos, Andrew Sheridan, Claire Sougnez, Nicole Stange-Thomann, Nenad Stojanovic, Aparna Subramanian, David Wyman, Jane Rogers, John Sulston, Rachel Ainscough, Stephen Beck, David Bentley, Jeremy Burton, Christopher Clee, Nigel Carter, Alan Coulson, Richard Deadman, Panos Deloukas, Andrew Dunham, Ian Dunham, Richard Durbin, Laurie French, Darren Grafham, Sarah Gregory, Tim Hubbard, Sean Humphray, Adrian Hunt, Martin Jones, Claire Lloyd, Andrew McMurray, Lesley Matthews, Sarah Mercer, Stuart Milne, Jim C. Mullikin, Andrew Mungall, Richard Plumb, M. Ross, R. Shownkeen, Simon Sims, Robert H. Waterston, Robert K. Wilson, LaDeana W. Hillier, John D. McPherson, Marco A. Marra, Elaine R. Mardis, Lucinda A. Fulton, Asif T. Chinwalla, Kelly H. Pepin, Warren R. Gish, Stephanie L. Chissoe, Michael C. Wendl, Kevin D. Delehaunty, Tracie L. Miner, Allen Delehaunty, Jody B. Kramer, Lisa L. Cook, Robert S. Fulton, David L. Johnson, Patrick J. Minx, Sandra W. Clifton, Trevor Hawkins, Elbert Branscomb, Pawel Predki, Paul Richardson, Sabine Wenning, Thomas Slezak, Norman Doggett, J. F. Cheng, Arne Olsen, Susan Lucas, Chris Elkin, Edward Uberbacher, Marc Frazier, Richard A. Gibbs, Donna M. Muzny, Steven E. Scherer, Jonathan B. Bouck, Erica J. Sodergren, Kim C. Worley, Carolyn M. Rives, John H. Gorrell, Michael L. Metzker, Stephen L. Naylor, Raju S. Kucherlapati, David L. Nelson, George M. Weinstock, Yasuo Sakaki, Asao Fujiyama, Masahira Hattori, Toshihide Yada, Atsushi Toyoda, Tetsuo Itoh, Chikashi Kawagoe, Hiroyuki Watanabe, Yutaka Totoki, Timothy Taylor, Jean Weissenbach, Roland Heilig, Walter Saurin, Francois Artiguenave, Patrick Brottier, Thierry Bruls, Eric Pelletier, Catherine Robert, Patrick Wincker, David R. Smith, Lynn Doucette-Stamm, Marc Rubenfield, Ken Weinstock, Hae-Min Lee, Jennifer Dubois, Andrew Rosenthal, Matthias Platzer, Gero Nyakatura, Stefan Taudien, Andreas Rump, Huanming Yang, Jun Yu, Jian Wang, Guang Huang, Jian Gu, Leroy Hood, Lee Rowen, Anjana Madan, Shenglong Qin, Ronald W. Davis, Neal A. Federspiel, Andres P. Abola, Michael J. Proctor, Richard M. Myers, Jeremy Schmutz, Mark Dickson, Jane Grimwood, David R. Cox, Maynard V. Olson, Rajinder Kaul, Christopher Raymond, Nobuyoshi Shimizu, Kazuyoshi Kawasaki, Shinsei Minoshima, Glen A. Evans, Maria Athanasiou, Russell Schultz, Bruce A. Roe, Feng Chen, Heng Pan, Juliane Ramser, Hans Lehrach, Ralf Reinhardt, W. Richard McCombie, Marianne de la Bastide, N. Dedhia, Helmut Blöcker, Klaus Hornischer, Georg Nordsiek, Richa Agarwala, L. Aravind, Jeffrey A. Bailey, Alex Bateman, Serafim Batzoglou, Ewan Birney, Peer Bork, David G. Brown, Christopher B. Burge, Laurent Cerutti, Hsiu-Chuan Chen, Deanna Church, Michele Clamp, Richard R. Copley, Tobias Doerks, Sean R. Eddy, Evan E. Eichler, Terrence S. Furey, James Galagan, James G. Gilbert, Clint Harmon, Yoshihide Hayashizaki, David Haussler, Henning Hermjakob, Karsten Hokamp, Woong-Yang Jang, Leon S. Johnson, T. Andrew Jones, Simon Kasif, Adam Kaspryzk, Shane Kennedy, W. James Kent, Paul Kitts, Eugene V. Koonin, Ian Korf, David Kulp, Doron Lancet, Todd M. Lowe, Aoife McLysaght, Tarjei Mikkelsen, Jeremiah V. Moran, Nicola Mulder, Vincent J. Pollara, Chris P. Ponting, Gregory Schuler, Jörg Schultz, Guy Slater, Arian F. Smit, Elia Stupka, Joseph Szustakowski, Danielle Thierry-Mieg, Jean Thierry-Mieg, Lukas Wagner, John Wallis, Richard Wheeler, Arwen Williams, Yuri I. Wolf, Kenneth H. Wolfe, Shiaw-Pyng Yang, Richard F. Yeh, Francis Collins, Mark S. Guyer, Jane Peterson, Gary Felsenfeld, Kris A. Wetterstrand, Aristides Patrinos, Mark J. Morgan, Pieter de Jong, Joseph J. Catanese, Kazutoyo Osoegawa, Hiroaki Shizuya, S. Choi, Yan J. Chen, and International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001. doi: 10.1038/35057062. Errata: Nature 2001 Jun 7;411(6838):720 and Nature 2001 Aug 2;412(6846):565. Author correction: Szustakowski, J.

[4] David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research*, 28(5):739–750, 2018.

[5] Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R Kelley. Predicting rna-seq coverage from dna sequence as a unifying model of gene regulation. *Nature Genetics*, 57(4):949–961, 2025.

[6] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

[7] Jimin Tan, Nina Shenker-Tauris, Javier Rodriguez-Hernaez, Eric Wang, Theodore Sakellaropoulos, Francesco Boccalatte, Palaniraja Thandapani, Jane Skok, Iannis Aifantis, David Fenyö, et al. Cell-type-specific prediction of 3d chromatin organization enables high-throughput in silico genetic screening. *Nature biotechnology*, 41(8):1140–1150, 2023.

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

[9] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025.

[10] Garyk Brixi, Matthew G Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, et al. Genome modeling and design across all domains of life with evo 2. *BioRxiv*, pages 2025–02, 2025.

[11] Anusri Pampari, Anna Shcherbina, Evgeny Z Kvon, Michael Kosicki, Surag Nair, Soumya Kundu, Arwa S Kathiria, Viviana I Risca, Kristiina Kuningas, Kaur Alasoo, et al. Chrombpnet: bias factorized, base-resolution deep learning models of chromatin accessibility reveal cis-regulatory sequence syntax, transcription factor footprints and regulatory variants. *BioRxiv*, pages 2024–12, 2025.

[12] Micaela E Consens, Cameron Dufault, Michael Wainberg, Duncan Forster, Mehran Karimzadeh, Hani Goodarzi, Fabian J Theis, Alan Moses, and Bo Wang. Transformers and genome language models. *Nature Machine Intelligence*, pages 1–17, 2025.

[13] Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, et al. Tokenizer choice for llm training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924, 2024.

[14] Guangjian Zeng, Chengzhi Zhao, Guanpeng Li, Zhengyang Huang, Jinhu Zhuang, Xiaohua Liang, Xiaxia Yu, and Shenying Fang. Identifying somatic driver mutations in cancer with a language model of the human genome. *Computational and Structural Biotechnology Journal*, 27:531–540, 2025.

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

[16] Melissa Sanabria, Jonas Hirsch, Pierre M Joubert, and Anna R Poetsch. Dna language model grover learns sequence context in the human genome. *Nature Machine Intelligence*, 6(8): 911–923, 2024.

[17] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana V Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=oMLQB4EZE1`.

[18] Jaime A Castro-Mondragon, Rafael Riudavets-Puig, Ieva Rauluseviciute, Roza Berhanu Lemma, Laura Turchi, Romain Blanc-Mathieu, Jeremy Lucas, Paul Boddie, Aziz Khan, Nicolás Manosalva Pérez, et al. Jaspar 2022: the 9th release of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 50(D1):D165–D173, 2022.

[19] Adityanarayanan Radhakrishnan, Yajit Jain, Caroline Uhler, and Eric S Lander. Efficiently quantifying dependence in massive scientific datasets using interdependence scores. *Proceedings of the National Academy of Sciences*, 122(34):e2509860122, 2025.

[20] Stuart Semple, Ramon Ferrer-i Cancho, and Morgan L Gustison. Linguistic laws in biology. *Trends in Ecology & Evolution*, 37(1):53–66, 2022.

[21] Rosario N Mantegna, Sergey V Buldyrev, Ary L Goldberger, Shlomo Havlin, Chung-Kang Peng, Michael Simons, and H Eugene Stanley. Linguistic features of noncoding dna sequences. *Physical review letters*, 73(23):3169, 1994.

[22] Laurence Aitchison, Nicola Corradi, and Peter E Latham. Zipf's law arises naturally when there are underlying, unobserved variables. *PLoS computational biology*, 12(12):e1005110, 2016.

[23] Sebastian Bonhoeffer, Andreas VM Herz, Maarten C Boerlijst, Sean Nee, Martin A Nowak, and Robert M May. No signs of hidden language in noncoding dna. *Physical review letters*, 76(11): 1977, 1996.

[24] Andrzej K Konopka and Colin Martindale. Noncoding dna, zipf's law, and language. *Science*, 268(5212):789–789, 1995.

[25] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017.

[26] Yan Hu, Max A Horlbeck, Ruochi Zhang, Sai Ma, Rojesh Shrestha, Vinay K Kartha, Fabiana M Duarte, Conrad Hock, Rachel E Savage, Ajay Labade, et al. Multiscale footprints reveal the organization of cis-regulatory elements. *Nature*, 638(8051):779–786, 2025.

[27] W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. The human genome browser at ucsc. *Genome research*, 12(6): 996–1006, 2002.

[28] Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature genetics*, 54(7):940–949, 2022.

[29] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

A Technical Appendices and Supplementary Material

A.1 Tokenization Methodology

We used the JASPAR 2022 dataset to get the bed file annotations of the 841 known TF motifs in the hg38 human genome. We also used the CpG annotations from the UCSC database as an extra feature, which gave us a total of 1684 motif features across both strands. [27]

Binning The motif annotations on the genome have a high degree of overlap. First, we sought to partition the genome into dynamic bins where each bin will be used later on as a token. To this end, for a given genomic position, we added the number of motifs (on both + and - strands) that overlap that position (Figure 1a). This gave us a 1D "motif coverage" signal across the genome. We then used a peak calling algorithm on the smoothed motif coverage signal to identify regions of high motif density. The boundaries of such high intensity regions were used as our bin boundaries. Since some regions of the genome are unmapped (don't have any motif annotations), this led to large bins of length >50bps (0.5% of the total bins). Since we wanted to keep the bin length distribution close to Gaussian, we trimmed those large bins to bins of length 25 which corresponds to the length scale of large motifs. In this way, the genome was partitioned into 187M bins of an average length of 15 base pairs (which corresponds to the average motif length). Note that the exact positioning of the bin boundaries is not of big importance given how each bin is encoded into a token (explained next). Also, note that beyond using only the motif annotations that had a minimum score of 800 (max p-value of 10^{-8}), we did not utilize the match scores in our tokenization. We think incorporating these match scores into our tokenization is a promising direction for further exploration.

Continuous tokens Although tokens are discrete in their traditional sense, we sought to produce a continuous representation of the genome: each bin was encoded into a continuous valued vector (i.e. token) of size 1684 (2 strands, 842 motifs), where each element corresponds to the proportion of a given strand's motif that falls in that bin. The reason whythe exact bin boundary is irrelevant is that, for a given motif, the values of the consecutive bins that span that motif will always sum to 1. This gave us a continuous representation of the genome of size 187Mx1682 (figure 1b)

Clustering Since many motifs are highly correlated, we can reduce the number of features that are used to represent the genome by grouping similar motifs together. To this end, we used a metric called InterDependence Score (IDS) introduced in [19], which measures nonlinear relationships in large scales (e.g. whole genome) in a computatinally efficient manner. We computed the genome-wide IDS between the 1682 motif-strand features, and used hierarchical clustering to group similar motifs together. This gave us 871 motif clusters (i.e. features). This is analogous to using the JASPAR motif clusters (which are given by taking the pairwise correlation of the motif PFMs and putting pairs of motifs above a certain correlation threshold into the same cluster), but we decided to use our own correlation which takes into account the different strands and is using our own motif scores as opposed to the PFMs. Figure 4 illustrates the IDS matrix and how similar motif features are clustered into the same group.

A.2 UMAP visualization

In order to obtain the UMAP in figure 2 a, we started with the continuously embedded tokens, each represented by an 871-dimensional vector of motif-derived IDS features. Then, in order to filter out 'noisy' tokens, we first discretized the continuous tokens using a threshold of 0.5, and then retained only the tokens whose discrete chromosome-wide frequency was between 500 and 1M.

We then applied PCA (K=180) followed by standard scaling. UMAP was then applied to these 180-dimensional subsampled continuous tokens using the cosine distance to produce an initial two-dimensional embedding of the data. The UMAP coordinates were then refined via t-SNE, implemented with FFT-accelerated optimization, as suggested by [28]. Louvain community detection was then performed in two steps: first with a nearest neighbors parameter (K1= 320) to identify "megaclusters" of similar tokens, followed by a more granular clustering (K2=80) to find smaller "subclusters" within those megaclusters. Figure 2 b shows the prominent motifs in some of the illustrated sublclusters of the largest megaculster in chromosome 1.

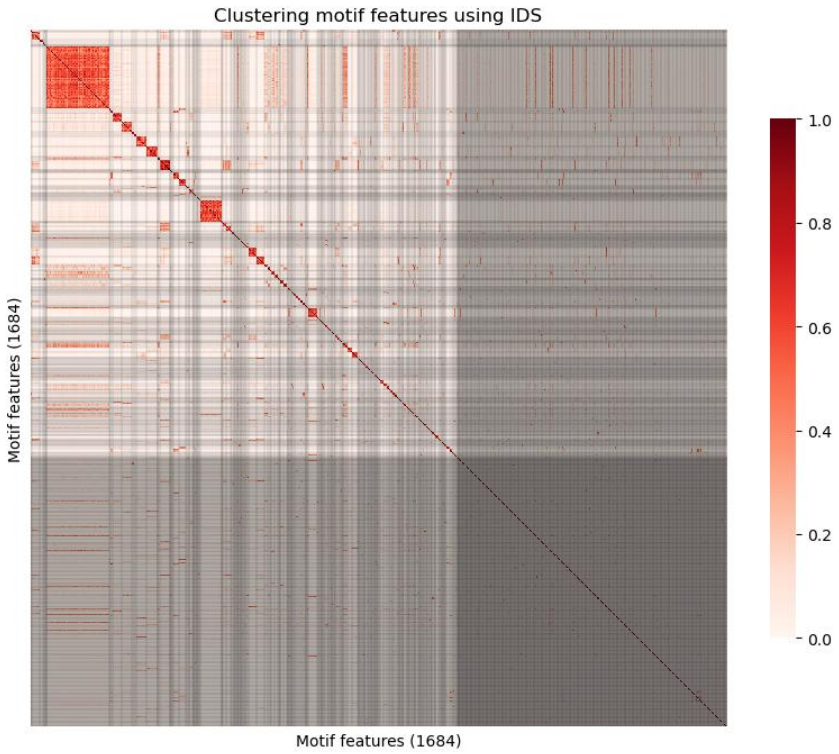

Figure 4: Genome-wide InterDependence Score (IDS) matrix

A.3 Sequence-to-Function modeling

To evaluate the utility of our tokenization method for sequence-to-function modeling, we focused on predicting ATAC-seq signal tracks from genomic sequence data in K562 cell lines. We benchmarked our tokenization approach against the standard one-hot encoding, which remains the predominant strategy in most sequence-to-function models. We evaluated performance using both a fully convolutional architecture and a transformer-based architecture, with detailed schematics provided in figure 3 a. We only report the performance results of the transformer models because they outperformed the fully convolutional models (average test set correlations were 0.69 and 0.65 for onehot and our encoding respectively). However, the convolutional model is still useful since we can use them for interpretability methods such as DeepLift (which cannot be readily applied to transformer architectures). To ensure a fair comparison, the architecture of the one-hot encoding model was carefully designed so that the output resolutions of both approaches were approximately matched. Specifically, the one-hot encoded model incorporated four average pooling layers, each with a window size of 2. Consequently, each value in the resulting ATAC-seq output track corresponded to $2^4 = 16$ base pairs, aligning closely with the average bin size employed in our tokenization scheme.

Data preparation and model training To prepare the training data, we utilized all autosomal chromosomes, designating chromosome 15 as the validation set, chromosome 10 as the held-out test set, and the remaining chromosomes for training. For each chromosome, we extracted genomic fragments of 1,250 bins in length (approximately 20 Kb) using a stride of 500 bins, resulting in approximately 344,000 training samples. Both input and output tracks were normalized to have zero mean and unit variance to facilitate stable training dynamics. For optimization, we employed a linear combination of a Poisson negative log-likelihood loss and a multinomial loss, similar to the approach used in the Borzoi model [5]. Models were trained for 15 epochs, with early stopping based on the Pearson correlation of the predicted and ground truth tracks in the validation set. Adam optimizer was used with a warm-up learning rate period of 5 epochs and linear decay after reaching the target learning rate of $1e^{-4}$. The models were trained on a GPU cluster with four NVIDIA L40S GPUs. The code for tokenization and model training will be made publicly available upon acceptance. While the one-hot

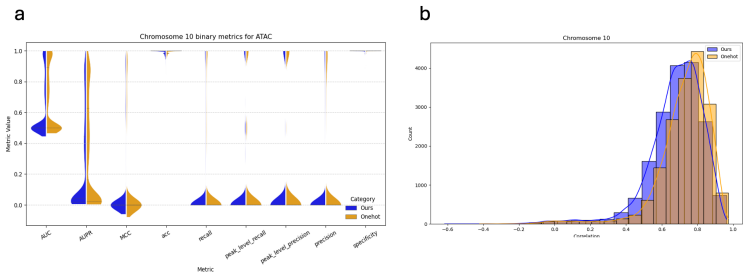

Figure 5: Binary peak detection metrics and pearson correlation on the test set

encoded model achieved higher overall performance in terms of correlation (average test set correlation = 0.73) compared to our tokenization method (average test set correlation = 0.68), qualitative evaluation of the predicted tracks indicates that our model effectively captures peak structures (Figure 3 a).

Binary metrics for peak detection Since Pearson correlation is highly susceptible to noise—and ATAC-seq tracks often exhibit substantial background noise—we aimed to assess which method more effectively captures peaks (i.e., open chromatin regions), as this represents a biologically meaningful objective. To this end, we binarized the ATAC-seq signals and employed binary classification metrics to evaluate the models' peak detection performance. Given that our predicted ATAC tracks have a resolution of 16 basepairs, standard peak calling algorithms such as MACS2, which require base-pair resolution inputs, cannot be directly applied. Consequently, we developed a custom peak calling approach utilizing the scipy package [29]. To determine optimal peak calling parameters, we performed a hyperparameter sweep on the ground truth ATAC-seq signal binned at 16 bp resolution, selecting the parameter set that maximized overlap with the default thresholded peak annotations provided by ENCODE (accession ENCFF948AFM). Using these optimized parameters, we applied our peak calling algorithm to the model predictions, annotating peak regions as 1 and non-peak regions as 0. This procedure yielded two binary vectors, enabling computation of various binary metrics for evaluating peak detection accuracy. Moreover, recognizing that traditional binary metrics may be sensitive to the size and precise boundaries of the called peaks—potentially introducing bias—we introduced two additional, more robust metrics: peak-level recall and peak-level precision. Peak-level recall quantifies the proportion of ground truth peaks overlapping with at least one predicted peak, whereas peak-level precision quantifies the proportion of predicted peaks that overlap with at least one ground truth peak. The results of our peak detection are reported in figure 4d. We can see that while our model slightly underperforms the onehot model, in terms of peak detection, the models have comparable performance. Figure 5 illustrates the test set performance for both the Pearson correlation metric and the peak detection binary metrics.

A.4 Interpretability analysis

In genomic language models, Input × Gradient and DeepLIFT are two common interpretability methods used to assign importance scores to individual bases in a DNA sequence, helping to reveal which sequence positions most influence a model's prediction [deeplift]. Input × Gradient computes the gradient of the output with respect to each input feature and multiplies it by the actual input value, making it straightforward to apply to one-hot encoded DNA but potentially sensitive to noisy gradients and unable to handle cases where gradients vanish due to activation saturation. DeepLIFT

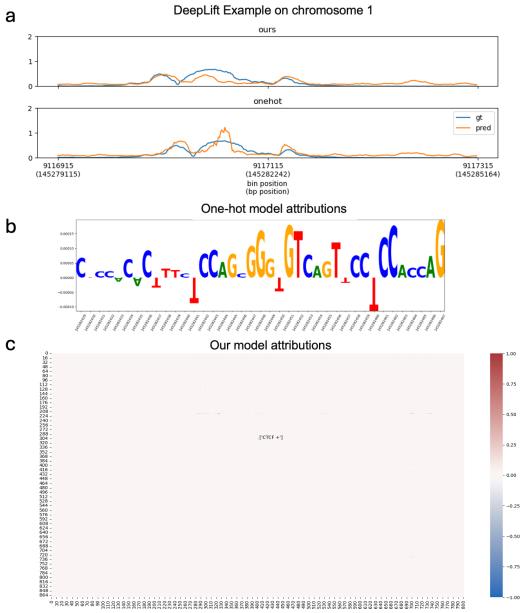

Figure 6: DeepLift visualization for a sample region of the genome

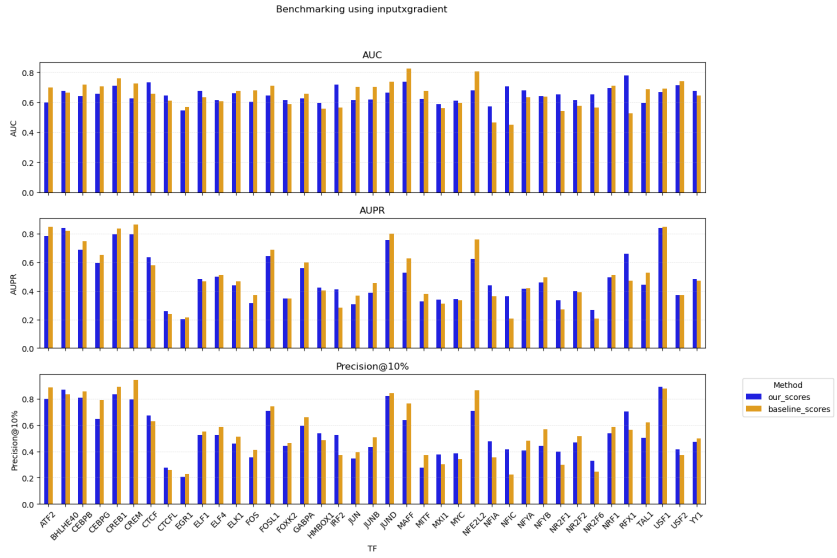

Figure 7: Input x Gradient benchmarking analysis

instead compares the model's activations for a real input versus a reference baseline sequence, then backpropagates these activation differences to the input, which can better capture importance when gradients are near zero. While Input x Gradient can be applied to any deep learning architecture, DeepLift is (at the time of this writing) not yet adapted for transformer architectures and can only be used with convolutional models. Figure 6 shows that for an open region chromating region that intersects with a CTCF peak, DeepLift applied to our model is able to correctly identify CTCF as the motif that's contributing the most to the model's prediction while the base level importance scores highlight only some of the basepairs that are present in a CTCF motif.

The TF benchmarking analysis using input x gradients is shown in figure 7.