MODERN GENE FINDERS: AB INITIO GENE DISCOVERY BENCHMARK WITH DNA LANGUAGE MODELS

Aleksei Shmelev[1,6], **Artem Shadskiy**[1,3], **Yuri Kuratov**[1,2], **Mikhail Burtsev**[4], **Olga Kardymon**[7], **Veniamin Fishman**[1,3,5]

[1]AXXX, Moscow, Russia
[2]Moscow Independent Research Institute of Artificial Intelligence, Moscow, Russia
[3]Sirius University, Sochi, Russia
[4]London Institute for Mathematical Sciences, London, UK
[5]Institute of Cytology and Genetics, Novosibirsk, Russia
[6]HSE University, Moscow, Russia
[7]Quantori, Belgrade, Serbia
minja.fishman@gmail.com

ABSTRACT

Detecting genes in DNA sequence is a fundamental step in enabling virtually any downstream analysis of the genome. A complete annotation pipeline must address two complementary tasks. One task is transcript position discovery, which determines where transcripts begin and end. The other task is transcript segmentation, which reconstructs exon intron structure within those intervals. In this work, we focus on transcript bondary discovery and treat it as an independent benchmarking problem. We introduce a mammalian benchmark that evaluates strand-aware localization of transcript boundaries for complete mRNA and lncRNA genes, using biologically grounded metrics based on transcription start sites and transcript termination sites. In addition, we introduce our own approach, which uses a DNA language model `ModernGENA` and a multi-stage pipeline to infer stranded transcript intervals and to recover multiple transcript boundary isoforms for the same gene.

1 INTRODUCTION

A genome's primary function is to encode genes — DNA segments that are transcribed into RNA. Mammalian genomes contain ∼25k protein-coding (mRNA) genes, whose products can be translated into proteins, and ∼20k long non-coding RNA (lncRNA) genes that do not give rise to a protein. Many genes have multiple alternative starts and ends, so a single gene can produce multiple RNA isoforms (alternative transcripts). Each transcript corresponds to a genomic interval bounded by a transcription start and end site. Because transcription is directional, a transcript can be specified unambiguously by three attributes: start, end, and strand (direction). Transcript boundary variation is often functionally relevant. It does not necessarily alter the encoded protein (although it may in some cases), since protein-coding transcripts are typically flanked by untranslated regions (UTRs) and shifts in transcript boundaries may leave the coding sequence unchanged. Nevertheless, even when the coding sequence is preserved, alternative transcripts can differ in stability, subcellular localization, and translation efficiency (Castillo-Hair et al., 2024).

DNA sequence alone does not identify where genes and transcripts are located. This information is essential for downstream analyses such as comparative genomics, functional interpretation, and biomedical research, and it must be inferred computationally when experimental annotation is unavailable. The need for reliable computational annotation is particularly evident in mammalian genomics. At present, NCBI provides $4\,582$ mammalian genome assemblies. Among these, about $1\,400$ assemblies are available at the chromosome level or higher, and 354 are designated as reference assemblies. However, only 166 assemblies currently have any form of gene annotation, and only 157 include complete and consistently curated RefSeq annotation. These numbers highlight why robust

ab initio annotation (annotation that does not require any experimental data beyond DNA sequence) remains necessary.

Ab initio annotation is commonly viewed as a multi-stage process. One stage is transcript position discovery, which determines where transcriptional units are located on the genome and where each transcript starts and ends. Another stage is transcript segmentation, which reconstructs exon–intron structure and coding organization within those transcript intervals. While both stages are required for a complete annotation, they pose different methodological challenges. In this work, we focus exclusively on transcript position discovery and leave transcript segmentation outside the scope of the present study.

Most existing *ab initio* annotation pipelines were originally designed around protein-coding gene structure and therefore emphasize coding regions. More recent neural and DNA language model based approaches can also predict untranslated regions for mRNAs and, in some cases, identify lncRNA transcripts, yet they are rarely evaluated under a common definition of transcript position. Here, we introduce a unified benchmark for *ab initio* transcript position discovery in mammalian genomes that evaluates strand-aware boundary localization for complete mRNA and lncRNA genes using biologically grounded metrics. We also introduce our own approach, which combines a custom DNA language model `ModernGENA` with a multi-stage pipeline to infer stranded transcript intervals and to recover multiple transcript boundary isoforms, addressing a setting where all existing methods output at most a single transcript per gene.

2 RELATED WORK

Recent years have seen increasing interest in large pretrained models for DNA sequence, driven by their ability to learn rich representations directly from genomic data. DNA language models such as DNABERT (Zhou et al., 2023), Nucleotide Transformer (Dalla-Torre et al., 2024; Boshar et al., 2025), GENA-LM (Fishman et al., 2025), Caduceus (Schiff et al., 2024), and Evo (Marchal, 2024; Nguyen et al., 2024) have demonstrated strong performance on a wide range of genomic prediction tasks. These models are trained on large collections of genomic sequence and can capture long-range dependencies that are relevant for identifying transcript boundaries, including transcription start sites and transcript termination sites.

Several works move more directly toward genome annotation style outputs. `SegmentNT` (de Almeida et al., 2025) builds on a `Nucleotide Transformer` backbone (Dalla-Torre et al., 2024) and produces nucleotide-resolution tracks for genic elements that can be post-processed into transcript intervals. `NTv3` (Boshar et al., 2025) further scales context length to the megabase range and targets genome-wide functional prediction across species. While these models provide dense predictions along the genome, their evaluation typically focuses on basewise metrics or task-specific labels, and it remains unclear how accurately complete transcript boundaries can be recovered at the interval level, especially in the presence of multiple isoforms.

Classical *ab initio* gene finders continue to play an important role in practical genome annotation because they output coherent gene models that satisfy biological constraints by construction. `AUGUSTUS` (Stanke et al., 2004) is a well-established standard that uses hidden Markov models (HMMs) with an explicit gene grammar and has been widely adopted in annotation pipelines. More recent neural systems aim to combine learnable sequence representations with structured decoding. `Helixer` (Holst et al., 2025) applies convolutional neural networks to predict gene annotation labels directly from sequence, while `Tiberius` (Gabriel et al., 2024) integrates a trainable neural backbone with an HMM-based decoder, and it represents one of the strongest modern baselines for protein-coding gene annotation.

Despite these advances, there is still no unified benchmark that isolates transcript position discovery as a standalone task. Existing evaluations typically assess coding regions only or use basewise labeling accuracy, which makes it difficult to compare methods that differ in how they represent internal gene structure, untranslated regions, or non-coding transcripts. In this work, we address this gap by benchmarking strand-aware transcript boundary localization directly, using biologically motivated interval-level metrics that apply consistently across classical HMM methods, hybrid systems, and modern DNA language model based approaches.

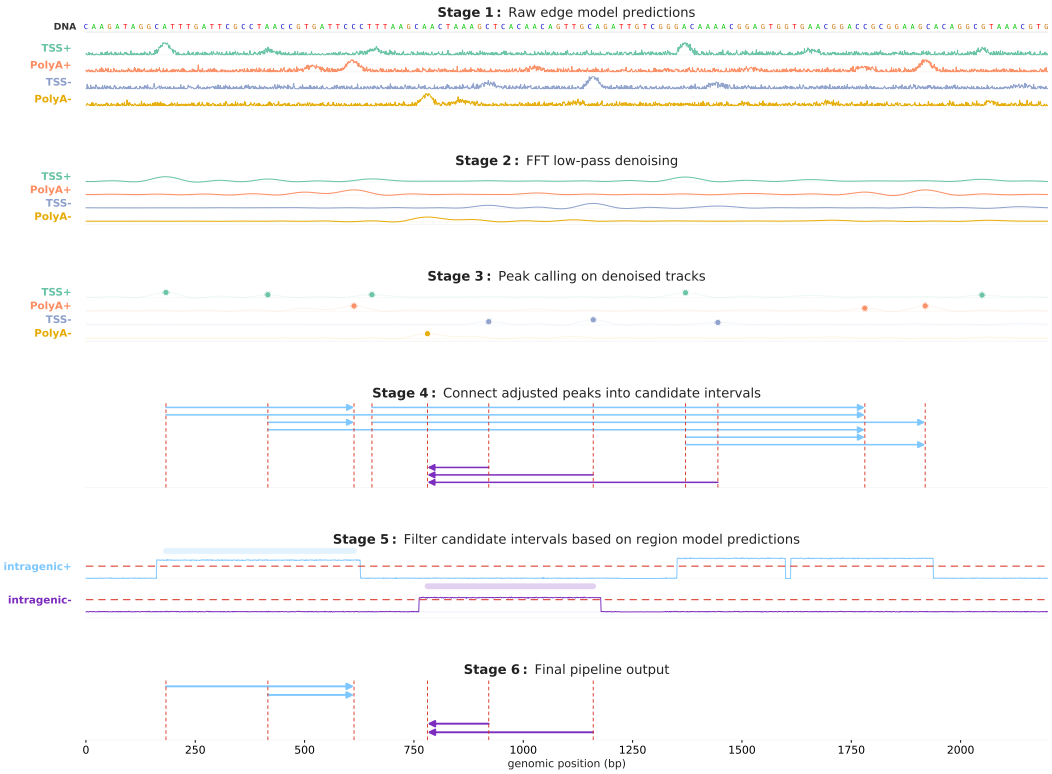

Figure 1: Overview of the proposed transcript discovery pipeline. Raw edge model boundary probability tracks for transcript $5'$ and $3'$ ends on both strands are shown first (Stage 1), then denoised using a Fourier based low pass filter (Stage 2) and converted into discrete boundary candidates by peak calling (Stage 3). Candidate stranded transcript intervals are formed by connecting $5'$ and $3'$ peaks in the transcription direction on the same strand (Stage 4), and then filtered using strand specific intragenic probability tracks produced by the region model (Stage 5). The remaining stranded intervals constitute the final pipeline output (Stage 6). See detailed description in subsection 3.3

3 METHODS

In this paper we present a custom approach for genome wide prediction of stranded transcript intervals, including overlapping isoforms with different lengths. The method combines two transformer based predictors, namely the *edge model* that predicts transcript boundaries, and the *region model* that predicts strand specific intragenic coverage. We use the edge model to generate multiple boundary hypotheses that can represent alternative isoforms, and we then use the region model to filter boundary based candidates using a more stable coverage signal. The full pipeline, including signal denoising and interval construction, is shown in Fig. 1.

3.1 BACKBONE AND FINE TUNING DATA

Both *edge* and *region* models use the ModernBERT encoder architecture (Warner et al., 2025) (Table A2). We first pretrained ModernBert similar to (Fishman et al., 2025), but using 443 vertebrate species (Appendix C) and longer context length (1024 tokens). All finetunning experiments were started from this pretrained checkpoint. For human-specific model we use human data only, for multispecies model we use data from 39 mammalian species (Appendix Table A5). In both cases we use transcripts annotated as mRNA or lncRNA only and held-out human chromosome 20 (GCF_009914755.1).

3.2 EDGE MODEL AND REGION MODEL

Reference annotation representation We use a reference annotation to define training targets, and we represent each transcript as a stranded interval while ignoring exon intron structure. For each transcript τ we use its strand $s_\tau \in \{+, -\}$. We define the transcript start site, denoted by t_τ, as the first nucleotide in transcription direction, and we define the transcript termination site, denoted by p_τ, as the last nucleotide in transcription direction. In other words, t_τ corresponds to the transcript $5'$ end and p_τ corresponds to the transcript $3'$ end, and both are interpreted in a strand aware manner.

Edge model The edge model is finetuned from the pretrained backbone with a context length of 1024 tokens. It predicts four strand specific boundary classes: $C_{\text{edge}} = \{\text{TSS}^+, \text{PolyA}^+, \text{TSS}^-, \text{PolyA}^-\}$. Here TSS denotes the transcript start site and PolyA denotes the transcript termination site, and the superscript indicates transcript strand. The edge model is trained to predict boundary probability tracks for transcript $5'$ and $3'$ ends within each input window.

Motivated by the imprecise nature of transcription initiation - experiments show that TSS position in real biological processes is slightly noisy (Seki et al., 2024) - we construct soft boundary targets at nucleotide resolution as follows. For each strand s, let $\mathcal{T}^s$ be the set of reference transcripts on that strand. For each boundary type $q \in \{\text{TSS}, \text{PolyA}\}$, we define the corresponding boundary position for transcript τ as $b_{\tau,q}$, where $b_{\tau,\text{TSS}} = t_\tau$ and $b_{\tau,\text{PolyA}} = p_\tau$ based on RefSeq genome annotation. For a nucleotide position j, we define a strand specific target as a union over transcripts:

$$y_q^s(j) = \max_{\tau \in \mathcal{T}^s} \exp\Big(- \frac{|j - b_{\tau,q}|}{w_q} \Big).$$

In our dataset construction we set $w_{\text{TSS}} = 10$ base pairs and $w_{\text{PolyA}} = 10$ base pairs. We then coerse nucleotide-level targets to tokens and train the model at BPE token resolution. For protein coding transcripts we additionally set boundary targets to zero within the coding span reported by the reference annotation, which prevents training signal from appearing inside coding regions. The edge model is trained with a binary cross entropy loss.

Region model The region model is finetuned independently from the edge model, starting from the same pretrained checkpoint, and it uses a longer context length of 8192 tokens. It predicts two strand specific intragenic classes: $C_{\text{reg}} = \{\text{Intra}^+, \text{Intra}^-\}$. A nucleotide position is labeled intragenic on strand s if it is covered by at least one reference transcript on strand s. The region prediction task therefore focuses on strand specific coverage rather than boundary localization. We use a longer context for this model because extended context improves stability of intragenic predictions and reduces fragmentation of long transcriptional units.

3.3 GENOME WIDE INFERENCE AND POST PROCESSING

Pipeline overview Our pipeline converts genome-wide predictions of the edge- and region-models into the final set of predicted transcript intervals, as shown in Fig. 1. First, we compute four nucleotide-level boundary probability tracks corresponding to the four classes in C_{edge}. Second, we denoise each boundary track using a Fourier-based low pass filter. Third, we perform peak calling on each denoised track to obtain single nucleotide boundary candidates. Then we construct stranded candidate transcript intervals by pairing TSS peaks with PolyA peaks in transcription direction, separately for each strand. Finally, we filter candidate intervals using the region model intragenic tracks from C_{reg} on the matching strand, and then define the remaining intervals as the final pipeline output. We detail each of these steps below.

Genome wide prediction Both models are intended to be applied to full chromosomes and scaffolds, so as a first step we compute genome wide prediction tracks. We apply each model in a sliding window manner across the whole given sequence. Adjacent windows overlap by half of the corresponding model context. Model outputs are produced at the BPE token level, and we map them back to nucleotide resolution by duplicating each token probability across the nucleotides covered by that token. For nucleotides covered by multiple windows we average the duplicated probabilities. We also run the same procedure on the reverse complemented sequence, but during this rev-comp pass, we swap strand specific channels. Finally, we average the forward and reverse complement tracks at each nucleotide. This is done for both the edge- and region-model tracks.

FFT low pass denoising Let $x[n]$ be a nucleotide level boundary track along a chromosome, where $n \in \{0, \ldots, L-1\}$ and L is the chromosome length in nucleotides. We compute the discrete Fourier transform:

$$X[k] = \sum_{n=0}^{L-1} x[n] \exp\Big(-2\pi i \frac{kn}{L} \Big),$$

where $k \in \{0, \ldots, L-1\}$ indexes frequency bins and i is the imaginary unit. We then keep only the lowest fraction of Fourier coefficients, using a cutoff fraction of 0.05 in our experiments, and we set the remaining coefficients to zero. We reconstruct the smoothed track by the inverse transform:

$$y[n] = \frac{1}{L} \sum_{k=0}^{L-1} X_{\mathrm{lp}}[k] \exp\Big(2\pi i \frac{kn}{L} \Big),$$

where X_{lp} denotes the truncated spectrum and $y[n]$ is the denoised boundary track. This operation is applied independently to each of the four boundary tracks. In our implementation we use `numpy.fft.rfft` and `numpy.fft.irfft` to compute the transform and its inverse.

Peak calling After denoising, the candidate boundary positions are identified by detecting local maxima in each smoothed track using `scipy.signal.find_peaks` peak detection procedure. We enforce a minimum prominence threshold and a minimum distance between peaks. In our experiments the minimum distance is fixed to 50 nucleotides, and we evaluate three prominence settings, namely 0.1, 0.15, and 0.3. We do not impose an explicit minimum peak height threshold. Each detected peak corresponds to a single nucleotide coordinate and is treated as a boundary candidate for the corresponding class and strand.

Interval construction We construct candidate transcript intervals separately for the forward strand and the reverse strand, and the procedure is the same for both strands. For each strand we iterate over TSS peaks, which represent candidate transcript $5'$ ends. For each such peak we search for PolyA peaks, which represent candidate transcript $3'$ ends, within a window of length $2 \cdot 10^6$ base pairs in transcription direction. For the forward strand this corresponds to the downstream direction in genomic coordinates, and for the reverse strand this corresponds to the upstream direction in genomic coordinates. Among the peaks in the window we keep up to 10 nearest partners by genomic distance. Each selected pair defines one candidate interval with endpoints at the two peak coordinates, then we reorder endpoints so that the interval start is smaller than the interval end, and remove duplicate intervals while preserving strand.

Filtering candidate intervals with the region model Finally we filter candidate intervals using the region model intragenic tracks. For each candidate interval we extract the region model probabilities on the matching strand, we binarize them using the probability threshold 0.5, and we compute the fraction of nucleotides inside the interval where the binarized intragenic signal is zero. We drop the interval if this fraction exceeds 0.01, otherwise we keep it. The remaining stranded intervals form the final output of the pipeline and are interpreted as predicted transcripts.

4 Constructing a benchmarking dataset and metrics

4.1 Benchmark dataset

We benchmark on human chromosome 20 from the T2T human assembly, containing 546 mRNA and 434 lncRNA genes. This chromosome is part of the held-out set for several widely used models, including SegmentNT, Tiberius, and Helixer, enabling an unbiased comparison.

4.2 Benchmark metrics

Many prior methods evaluate performance using per-nucleotide metrics. We argue that this framing is often misaligned with the underlying biology. Transcripts can span tens of kilobases, and predictions that shift transcript boundaries by several kilobases may still achieve a high average per-base score while substantially altering the resulting protein product. Moreover, per-nucleotide metrics do not capture transcript structure or the presence of multiple isoforms. Therefore, our benchmark introduces a comprehensive set of transcript-level metrics designed to evaluate boundary localization.

kX We first define the *kX* boundary quality metric. Let P denote the set of predicted genomic intervals, and let R denote the set of reference transcript intervals from mRNA and lncRNA annotations on the chromosome under evaluation, where each transcript belongs to a reference gene in a set G. For a tolerance k in base pairs, a predicted interval $p = [a, b)$ is counted as a match if there exists a reference transcript interval $r = [u, v)$ such that: $|a - u| \leq k \quad \text{and} \quad |b - v| \leq k$. This criterion evaluates whether a model localizes both transcript boundaries within the specified tolerance, independent of internal exon and intron structure. The motivation of having non-zero tolerance stems from the imprecision of transcription machinery described above: both start and end of the transcript are slightly noisy (Seki et al., 2024), whereas ground-truth annotation presents them as exact positions. We also define interval level true positives, $\mathrm{TP}_{\mathrm{int}}(k)$, as the number of predicted intervals that match at least one reference transcript under this rule, and interval level false positives, $\mathrm{FP}_{\mathrm{int}}(k)$, as the remaining predicted intervals. Hence, the precision is:

$$\mathrm{Precision}(k) = \frac{\mathrm{TP}_{\mathrm{int}}(k)}{\mathrm{TP}_{\mathrm{int}}(k) + \mathrm{FP}_{\mathrm{int}}(k)}.$$

To measure discovery quality at the gene level, we count a reference gene as detected if at least one of its transcripts is matched by at least one prediction. This yields gene level true positives, $\mathrm{TP}_{\mathrm{gene}}(k)$, and gene level false negatives, $\mathrm{FN}_{\mathrm{gene}}(k)$, which count reference genes with no matched transcript. Hence, the recall is:

$$\mathrm{Recall}(k) = \frac{\mathrm{TP}_{\mathrm{gene}}(k)}{\mathrm{TP}_{\mathrm{gene}}(k) + \mathrm{FN}_{\mathrm{gene}}(k)},$$

and the final F_1 score summarizing the precision and recall balance is:

$$F_1(k) = \frac{2\,\mathrm{Precision}(k)\,\mathrm{Recall}(k)}{\mathrm{Precision}(k) + \mathrm{Recall}(k)}.$$

Multi isoform We additionally quantify whether a model predicts multiple boundary isoforms for the same gene using a *multi isoform* score derived from the *kX* matching rule. For each reference transcript $r = [u, v)$ we define its length as $\ell(r) = v - u$. A reference gene is considered multi isoform eligible if it has at least two annotated transcripts and at least two distinct values of $\ell(r)$ among its transcripts. For an eligible gene, we count it as recovered with multiple isoforms at tolerance k if the predictions match transcripts corresponding to at least two distinct transcript lengths of that gene, and the supporting matches involve at least two distinct predicted intervals.

UTR-aware kX For mRNA transcripts we additionally define an *UTR-aware kX* metric that evaluates boundary placement while respecting coding sequence integrity. Each reference mRNA transcript is represented by its full transcript interval $[u_{\mathrm{tx}}, v_{\mathrm{tx}})$ and by a coding sequence core interval $[u_{\mathrm{cds}}, v_{\mathrm{cds}})$, defined by the first and last coding base of the transcript. A predicted interval $p = [a, b)$ can match a transcript only if it fully contains the coding core: $a \leq u_{\mathrm{cds}} \quad \text{and} \quad b \geq v_{\mathrm{cds}}$, which prevents truncation of coding sequence that could remove start or stop codons and thereby change the translated protein. We then allow limited extension into flanking intergenic sequence by requiring: $\max\big(u_{\mathrm{tx}} - a,\, b - v_{\mathrm{tx}}\big) \leq k$. Finally, we require boundaries to be located within exons, not introns. This constraint is biologically meaningful because mature mRNAs are produced by splicing, intronic sequence is removed from the final transcript, and transcript boundaries reported at nucleotide resolution are expected to lie in exonic sequence. A boundary placed within a UTR intron would correspond to sequence that is absent from the mature RNA, it would imply a noncanonical transcript architecture, and it can introduce unnecessary stop codons that shift open reading frame.

In our benchmark, this *UTR-aware kX* metric is the primary basis for comparisons that include both modern DNA LM models and classical *ab initio* gene finders, because many widely used tools primarily target coding gene structure and typically omit UTR boundary prediction, including `AUGUSTUS` (Stanke et al., 2004) and `Tiberius` (Gabriel et al., 2024). Precision, recall and the F_1 score are computed as in *kX* but for mRNA genes only.

Strand accuracy When strand predictions are available, we report an additional *strand accuracy* score at tolerance k for both *kX* and *UTR-aware kX*. We evaluate strand accuracy only for predicted intervals that are correctly matched to at least one reference transcript at tolerance k under the corresponding matching rule. A strand prediction is counted as correct only when it agrees with the strand of every reference transcript that matches the interval at tolerance k under particular rule.

Strand accuracy is defined as the fraction of true positive predicted intervals whose strand is correct among all true positive predicted intervals, and it is reported only for models that output strand information.

Accuracy To complement boundary based metrics, we also evaluate genome wide base level prediction quality using an *accuracy* score. We represent the full chromosome as a binary sequence, where a position is labeled 1 if it lies within any reference gene locus and 0 otherwise. The prediction is converted to the same representation using the union of predicted intervals, regardless of strand. A position is counted as correct when the predicted label matches the reference label, and we report the resulting basewise *accuracy*.

SOV score We also report the *SOV* metric, a segmentation overlap measure that evaluates agreement between predicted and reference gene loci at the level of contiguous genomic segments and their boundary placement, rather than only at individual base positions. SOV is computed by comparing a binary representation of the chromosome in which positions inside any reference gene locus are labeled as gene and all other positions are labeled as intergenic, with an analogous binary representation derived from the union of predicted intervals. We report two variants, *SOV_99* and *SOV_refine*, computed using default $\lambda = 1$, following recent practice in long context DNA language model evaluation such as `SegmentNT` (de Almeida et al., 2025), and standard SOV definitions (Liu & Wang, 2018).

5 RESULTS

We generated genome-wide predictions for all models using the developed benchmark (980 genes on the chromosome 20). For our `ModernGENA` pipeline we report three post-processing settings that correspond to the peak-calling prominence thresholds $p \in \{0.10, 0.15, 0.30\}$ (Fig. 1). Table 1 reports the main benchmark metrics at tolerance $k = 250$ bp. We additionally provide scores for both *kX* and UTR-aware *kX* over $k \in [0, 500]$ in Fig A1 and Fig A2 and per-class validation AUC scores for the `ModernGENA` edge- and region-models for both human and multispecies finetunning in Table A3.

Across all evaluated methods, `ModernGENA` achieves the strongest performance under *k250*. The highest *k250* F_1 in this benchmark is obtained by the multispecie model at $p = 0.30$ ($F_1 = 0.45$ with 354 matched intervals and 326 detected genes). When the prominence threshold is reduced, the pipeline predicts more intervals, which increases detection to 434 genes at $p = 0.10$ for the multispecie model. At the same time, the larger candidate set reduces precision and therefore limits the net improvement in F_1.

Training on multispecies dataset improves the `ModernGENA` pipeline relative to human-only training for all post-processing settings. At $p = 0.15$, the multispecie model improves F_1 from 0.36 to 0.43 compared with the human model, and it also improves SOV_{ref} from 0.47 to 0.57. At $p = 0.10$, the multispecie model increases detected genes from 401 to 434 and improves basewise accuracy from 0.82 to 0.87. At $p = 0.30$, the multispecies model improves F_1 from 0.41 to 0.45, while producing a similar number of predicted intervals. These results show that multispecies training improves the accuracy of transcript position prediction.

Under the UTR-aware *k250* metric on mRNA genes, which enforces preservation of the coding core while allowing imperfect recovery of untranslated boundaries, `Tiberius` achieves the highest F_1^{UTR} with 0.82. `ModernGENA` remains competitive, and the multispecies model at $p = 0.15$ reaches $F_1^{\text{UTR}} = 0.79$ while detecting 475 mRNA genes at $k = 250$, compared with 453 for `Tiberius`. The difference between `Tiberius` performance under UTR-aware and standard metrics reflects the fact that `Tiberius` is not designed to predict untranslated regions, and therefore its predictions can accurately cover proteain-coding gene parts while failing to localize transcript start and termination positions.

Genome-wide locus agreement shows a similar pattern. Basewise accuracy is relatively high for several methods, which is expected because intergenic regions dominate the genome-wide label distribution. SOV scores provide clearer separation, and `ModernGENA` achieves the strongest segmentation overlap, with $\text{SOV}_{99} = 0.65$ and $\text{SOV}_{\text{ref}} = 0.57$ at $p = 0.15$ for the multispecies model. Approaches that are evaluated through thresholded probability tracks, including `SegmentNT`,

Table 1: Gene finding performance on human chr20, reported at tolerance $k = 250$ bp. *Pred* is the number of predicted transcript intervals. TP_{int}, TP_{gene}, and F_1 are computed under *k250* on mRNA and lncRNA genes. $TP_{\text{int}}^{\text{UTR}}$, $TP_{\text{gene}}^{\text{UTR}}$, and F_1^{UTR} are computed under UTR-aware *kX* on mRNA genes only. *MI* is the number of multi-isoform genes recovered. *Acc* is standard basewise accuracy. SOV_{99} and SOV_{ref} correspond to *SOV_99* and *SOV_refine*. *Str* and Str^{UTR} are strand accuracy on corresponding true-positive intervals. For models whose names include a length in kb or Mb, the value denotes the sliding-window chunk size used during inference. For `SegmentNT`, the tags *hs* (human) and *ms* (multispecies) denote the fine-tuning data used by that model. For `ModernGENA`, *hs* denotes training on human only, and *39 sp* denotes multispecies finetuning (39 mammalian species from Appendix Table A5). The primary two metrics we are focusing on in this paper highlighted.

Model	Pred	TP_{int}	TP_{gene}	F_1	$TP_{\text{int}}^{\text{UTR}}$	$TP_{\text{gene}}^{\text{UTR}}$	F_1^{UTR}	MI	Acc	SOV_{99}	SOV_{ref}	Str	Str^{UTR}
max	3998	3998	980	1.00	3146	546	1.00	432	1.00	1.00	1.00	1.00	1.00
`ModernGENA` (39 sp, $p = 0.10$)	2 083	633	434	0.36	1 295	493	0.74	**66**	**0.87**	0.64	0.56	1.00	1.00
`ModernGENA` (39 sp, $p = 0.15$)	1 228	531	412	0.43	885	475	0.79	44	0.86	**0.65**	**0.57**	1.00	1.00
`ModernGENA` (39 sp, $p = 0.30$)	524	354	326	**0.45**	458	383	0.78	11	0.75	0.48	0.41	1.00	1.00
`ModernGENA` (hs, $p = 0.10$)	2 764	584	401	0.28	984	440	0.49	65	0.82	0.55	0.45	1.00	1.00
`ModernGENA` (hs, $p = 0.15$)	1 444	486	376	0.36	739	423	0.62	48	0.81	0.57	0.47	1.00	1.00
`ModernGENA` (hs, $p = 0.30$)	593	353	312	0.41	449	353	0.70	17	0.73	0.50	0.43	1.00	1.00
`Tiberius`	559	55	55	0.07	455	453	**0.82**	0	0.77	0.54	0.43	1.00	1.00
`Helixer`	563	252	252	0.33	347	346	0.62	0	0.86	0.33	0.28	1.00	1.00
`AUGUSTUS`	1 093	77	77	0.07	128	128	0.16	0	0.66	0.14	0.10	1.00	1.00
`SegmentNT` (hs, 30 kb)	92 337	90	86	0.00	108	108	0.00	0	0.80	0.00	0.00	NA	NA
`SegmentNT` (hs, 50 kb)	69 862	93	93	0.00	112	112	0.00	0	0.81	0.00	0.00	NA	NA
`SegmentNT` (ms, 30 kb)	92 359	90	86	0.00	108	108	0.00	0	0.80	0.00	0.00	NA	NA
`SegmentNT` (ms, 50 kb)	69 834	93	93	0.00	112	112	0.00	0	0.81	0.00	0.00	NA	NA
`SegmentBorzoi` (512 kb)	70 905	161	161	0.00	216	216	0.01	0	0.86	0.01	0.01	NA	NA
`SegmentEnformer` (192 kb)	96 327	160	160	0.00	214	214	0.00	0	0.85	0.00	0.00	NA	NA
`NTv3`-100M (32 kb)	21 399	0	0	0.00	0	0	0.00	0	0.51	0.01	0.00	NA	NA
`NTv3`-100M (1 Mb)	13 089	2	2	0.00	3	3	0.00	0	0.51	0.01	0.01	NA	NA
`NTv3`-650M (32 kb)	19 691	0	0	0.00	1	1	0.00	0	0.51	0.01	0.00	NA	NA
`NTv3`-650M (1 Mb)	9 735	4	4	0.00	5	5	0.00	0	0.51	0.01	0.01	NA	NA

`SegmentBorzoi`, `SegmentEnformer`, and `NTv3`, produce tens of thousands of short intervals on this chromosome and yield near-zero *kX* F_1 under the present interval-based benchmark.

Strand accuracy on true-positive intervals is 1.00 for all models that provide strand labels in our evaluation, and models that do not output strand are reported as NA in Table 1.

Only the `ModernGENA` pipeline achieves non-zero multi-isoform gene recovery, demonstrating state-of-the-art performance in this setting and making it the only evaluated model capable of identifying more than one transcript variant for the same gene locus. On chromosome 20 there are 432 eligible multi-isoform genes, and the multi-species model recovers 66 multi-isoform genes at $p = 0.10$, which corresponds to 15.3% of the eligible set. All other evaluated methods recover zero multi-isoform genes under this metric, indicating that they produce a single transcript per gene locus and therefore fail to capture isoform diversity.

6 Conclusion

This work introduces a biologically-aware benchmark of modern gene annotation methods capable of localizing gene regions within genomic sequences. This benchmarking approach facilitates an objective evaluation of the advantages and limitations of different models within a unified framework. We finetuned new models for the identification of gene positions within the genome. A key distinguishing feature of the proposed approach is its ability to detect multiple isoforms of the same gene that differ in their genomic coordinates. The majority of previously developed models formulate gene annotation as a deterministic mapping from a nucleotide sequence to a single annotation, in which each genomic position is assigned a label of only one gene isoform. However, genes can overlap and may exist as multiple isoforms with different lengths and boundaries. Consequently, multiple biologically valid interpretations may exist for the same genomic position. The models proposed in this work partially address this limitation by enabling the detection of gene isoforms with distinct genomic coordinates.

At the same time, the current approach has several limitations. In particular, the developed pipeline is not designed to identify isoforms arising from alternative splicing within genes. Such isoforms share identical transcription start and end sites but differ in their exon–intron structure. Future work will focus on extending the proposed approach toward gene segmentation and modeling of the alternative splicing.

REFERENCES

Sam Boshar, Benjamin Evans, Ziqi Tang, Armand Picard, Yanis Adel, Franziska K Lorbeer, Chandana Rajesh, Tristan Karch, Shawn Sidbon, David Emms, et al. A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction. *bioRxiv*, pp. 2025–12, 2025.

Sebastian Castillo-Hair, Stephen Fedak, Ban Wang, Johannes Linder, Kyle Havens, Michael Certo, and Georg Seelig. Optimizing 5'utrs for mrna-delivered gene editing using deep learning. *Nature Communications*, 15(1):5284, 2024.

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*, pp. 1–11, 2024.

Bernardo P de Almeida, Hugo Dalla-Torre, Guillaume Richard, Christopher Blum, Lorenz Hexemer, Maxence Gélard, Javier Mendoza-Revilla, Ziqi Tang, Frederikke I Marin, David M Emms, et al. Annotating the genome at single-nucleotide resolution with dna foundation models. *Nature Methods*, pp. 1–15, 2025.

Veniamin Fishman, Yuri Kuratov, Aleksei Shmelev, Maxim Petrov, Dmitry Penzar, Denis Shepelin, Nikolay Chekanov, Olga Kardymon, and Mikhail Burtsev. Gena-lm: a family of open-source foundational dna language models for long sequences. *Nucleic Acids Research*, 53(2):gkae1310, 2025.

Lars Gabriel, Felix Becker, Katharina J Hoff, and Mario Stanke. Tiberius: end-to-end deep learning with an hmm for gene prediction. *Bioinformatics*, 40(12):btae685, 2024.

Felix Holst, Anthony M Bolger, Felicitas Kindel, Christopher Günther, Janina Maß, Sebastian Triesch, Niklas Kiel, Nima Saadat, Oliver Ebenhöh, Björn Usadel, et al. Helixer: ab initio prediction of primary eukaryotic gene models combining deep learning and a hidden markov model. *Nature Methods*, pp. 1–8, 2025.

Tong Liu and Zheng Wang. Sov_refine: A further refined definition of segment overlap score and its significance for protein structure similarity. *Source code for biology and medicine*, 13(1):1, 2018.

Iris Marchal. Evo learns biological complexity from the molecular to genome scale. *nature biotechnology*, 42(12):1793–1793, 2024.

Eric Nguyen, Michael Poli, Matthew G Durrant, Brian Kang, Dhruva Katrekar, David B Li, Liam J Bartie, Armin W Thomas, Samuel H King, Garyk Brixi, et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723):eado9336, 2024.

Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling, 2024. URL `https://arxiv.org/abs/2403.03234`.

Masahide Seki, Yuta Kuze, Xiang Zhang, Ken-ichi Kurotani, Michitaka Notaguchi, Haruki Nishio, Hiroshi Kudoh, Takuya Suzaki, Satoko Yoshida, Sumio Sugano, et al. An improved method for the highly specific detection of transcription start sites. *Nucleic Acids Research*, 52(2):e7–e7, 2024.

Mario Stanke, Rasmus Steinkamp, Stephan Waack, and Burkhard Morgenstern. Augustus: a web server for gene finding in eukaryotes. *Nucleic acids research*, 32(suppl_2):W309–W312, 2004.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, 2025.

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

APPENDIX A. COMPARISON OF MODELS

Table A1: Frameworks of classical gene finders and recent foundation models used for *de novo* gene annotation or related genomic tasks.

Model	**Architecture (details)**	**N params, M**	**Input, Kb**	**Tokenization**	**Released**
AUGUSTUS	HMM	N/A	N/A	1-bp	(Stanke et al., 2004)
Tiberius	CNN + HMM	8	10	1-hot	(Gabriel et al., 2024)
Helixer	CNN + HMM	N/A	100	1-bp	(Holst et al., 2025)
SegmentNT	Transformer (RoPE) + U-Net	500	50	6-mer	(de Almeida et al., 2025)
SegmentBorzoi	CNN + U-Net	323	196	nucleotide	(de Almeida et al., 2025)
SegmentEnformer	Transformer + U-Net	379	196	nucleotide	(de Almeida et al., 2025)
NTv3	U-Net + Transformer	100	1000	nucleotide	(Boshar et al., 2025)
NTv3	U-Net + Transformer	650	1000	nucleotide	(Boshar et al., 2025)
ModernGENA	Transformer (ModernBERT)	135	4 (expandable)	BPE	this paper

Table A2: Architecture configuration for ModernGENA.

Parameter	ModernGENA
parameters	135M
num_hidden_layers	22
hidden_size	768
intermediate_size	1152
num_attention_heads	12

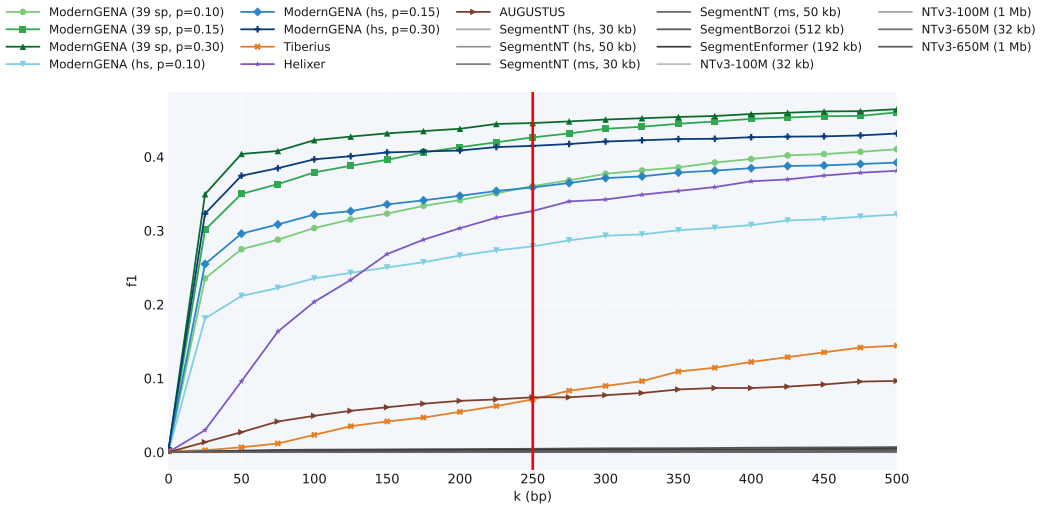

Figure A1: kX F_1 score as a function of the boundary tolerance k. Results are shown for all evaluated models. The vertical red line marks the operating point $k = 250$ bp used for the main benchmark results reported in Table 1.

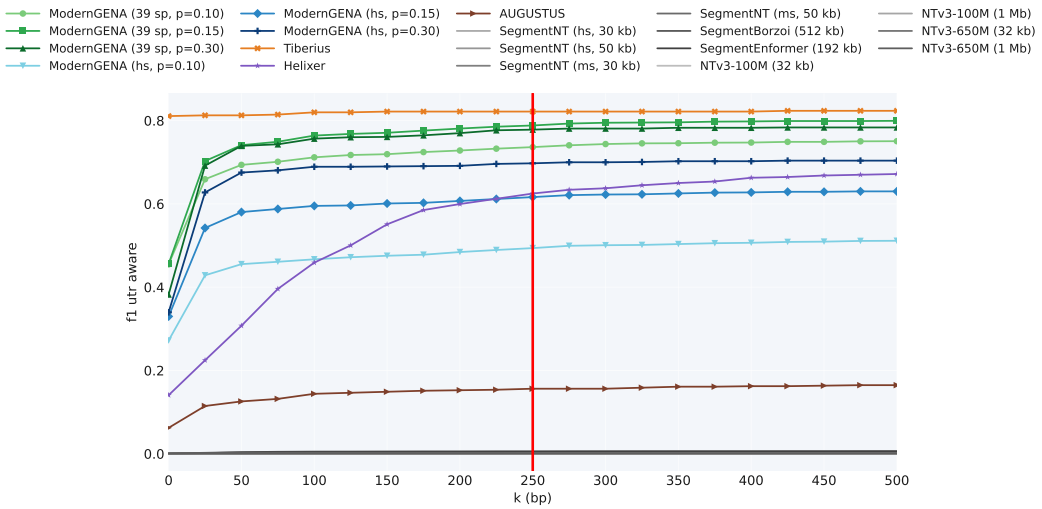

Figure A2: UTR-aware kX F_1 score as a function of the boundary tolerance k. Evaluation is restricted to mRNA genes. The vertical red line indicates $k = 250$ bp, which is used throughout the main text for more simple comparisons.

Table A3: Per-base ROC–AUC on human validation chromosome for the ModernGENA models used in the benchmark. Results are reported separately for the region model (intragenic classes) and the edge model (boundary classes), and for models trained on human only and multispecies datasets.

Model	Class	Strand	AUC
ModernGENA (hs, region)	Intragenic	+	0.9241
ModernGENA (hs, region)	Intragenic	−	0.9083
ModernGENA (hs, edge)	TSS	+	0.9484
ModernGENA (hs, edge)	TSS	−	0.9364
ModernGENA (hs, edge)	PolyA	+	0.9346
ModernGENA (hs, edge)	PolyA	−	0.9285
ModernGENA (39 sp, region)	Intragenic	+	0.9414
ModernGENA (39 sp, region)	Intragenic	−	0.9161
ModernGENA (39 sp, edge)	TSS	+	0.9534
ModernGENA (39 sp, edge)	TSS	−	0.9494
ModernGENA (39 sp, edge)	PolyA	+	0.9303
ModernGENA (39 sp, edge)	PolyA	−	0.9225

Table A4: Additional benchmark metrics reported at tolerance $k = 250$ bp. The *max* row reports the maximum attainable value for each metric on this chromosome given the reference annotation. TP_{mRNA} and TP_{lnc} denote the number of detected mRNA and lncRNA genes under *kX*. *Pr* and *Rc* denote *kX* precision and recall. Pr^{UTR} and Rc^{UTR} denote UTR-aware *kX* precision and recall on mRNA genes. *MI Rc* denotes the recall for multi-isoform genes among the eligible set.

Model	$\mathrm{TP}_{\mathrm{mRNA}}$	$\mathrm{TP}_{\mathrm{lnc}}$	Pr	Rc	$\mathrm{Pr}^{\mathrm{UTR}}$	$\mathrm{Rc}^{\mathrm{UTR}}$	MI Rc
max	546	434	1.00	1.00	1.00	1.00	1.00
`ModernBERT` (39 sp, $p = 0.10$)	420	14	0.30	**0.44**	0.62	**0.90**	**0.153**
`ModernBERT` (39 sp, $p = 0.15$)	402	10	0.43	0.42	0.72	0.87	0.102
`ModernBERT` (39 sp, $p = 0.30$)	323	3	**0.68**	0.33	**0.87**	0.70	0.025
`ModernBERT` (hs, $p = 0.10$)	386	15	0.21	0.41	0.36	0.81	0.150
`ModernBERT` (hs, $p = 0.15$)	367	9	0.34	0.38	0.51	0.77	0.111
`ModernBERT` (hs, $p = 0.30$)	308	4	0.60	0.32	0.76	0.65	0.039
`Tiberius`	55	0	0.10	0.06	0.81	0.83	0.000
`Helixer`	252	0	0.45	0.26	0.62	0.63	0.000
`AUGUSTUS`	76	1	0.07	0.08	0.12	0.23	0.000
`SegmentNT` (hs, 30 kb)	82	4	0.00	0.09	0.00	0.20	0.000
`SegmentNT` (hs, 50 kb)	89	4	0.00	0.09	0.00	0.21	0.000
`SegmentNT` (ms, 30 kb)	82	4	0.00	0.09	0.00	0.20	0.000
`SegmentNT` (ms, 50 kb)	89	4	0.00	0.09	0.00	0.21	0.000
`SegmentBorzoi` (512 kb)	158	3	0.00	0.16	0.00	0.40	0.000
`SegmentEnformer` (192 kb)	153	7	0.00	0.16	0.00	0.39	0.000
`NTv3`-100M (32 kb)	0	0	0.00	0.00	0.00	0.00	0.000
`NTv3`-100M (1 Mb)	2	0	0.00	0.00	0.00	0.01	0.000
`NTv3`-650M (32 kb)	0	0	0.00	0.00	0.00	0.00	0.000
`NTv3`-650M (1 Mb)	4	0	0.00	0.00	0.00	0.01	0.000

Table A5: List of genomic assemblies used to create the multispecies training dataset. List of genomic assemblies used to create the multispecies training dataset. Assembly names correspond to the annotation and genome names.

Assembly	Species
GCF_000952055.2	Aotus nancymaae
GCF_002263795.3	Bos taurus
GCF_000767855.1	Camelus bactrianus
GCF_000002285.3	Canis lupus familiaris
GCF_000151735.1	Cavia porcellus
GCF_001604975.1	Cebus imitator
GCF_000283155.1	Ceratotherium simum simum
GCF_000276665.1	Chinchilla lanigera
GCF_000260355.1	Condylura cristata
GCF_002940915.1	Desmodus rotundus
GCF_000151885.1	Dipodomys ordii
GCF_002288905.1	Enhydra lutris kenyon
GCF_000308155.1	Eptesicus fuscus
GCF_000002305.2	Equus caballus
GCF_018350175.1	Felis catus
GCF_000247695.1	Heterocephalus glaber
GCF_009914755.1	Homo sapiens
GCF_000236235.1	Ictidomys tridecemlineatus
GCF_000280705.1	Jaculus jaculus
GCF_000001905.1	Loxodonta africana
GCF_001458135.1	Marmota marmota
GCF_000165445.2	Microcebus murinus
GCF_000317375.1	Microtus ochrogaster
GCF_000001635.26	Mus musculus
GCF_900095145.1	Mus pahari
GCF_002201575.1	Neomonachus schauinslandi
GCF_000292845.1	Ochotona princeps
GCF_000260255.1	Octodon degus
GCF_000321225.1	Odobenus rosmarus divergens
GCF_009806435.1	Oryctolagus cuniculus
GCF_000181295.1	Otolemur garnettii
GCF_016772045.2	Ovis aries
GCF_000956105.1	Propithecus coquereli
GCF_003327715.1	Puma concolor
GCF_036323735.1	Rattus norvegicus
GCF_000235385.1	Saimiri boliviensis boliviensis
GCF_000181275.1	Sorex araneus
GCF_000003025.6	Sus scrofa
GCF_000243295.1	Trichechus manatus latirostris

APPENDIX B. MODELS SCORING AND BENCHMARKING

B.1 PROCESSING PREDICTIONS

`AUGUSTUS`, `Tiberius`, and `Helixer`, produce predictions in GFF format, where genomic coordinates can be explicitly reported for each predicted gene and transcript. These coordinates were used directly to derive intervals used by our metrics.

In contrast, models such as `SegmentNT` and `NTv3` output base-resolution prediction tracks for transcript classes (mRNA and lncRNA). For these models, a continuous genomic segment in which the predicted probability exceeded the threshold of 0.5 (following the authors' recommendations) was interpreted as a single predicted transcript.

B.2 BENCHMARKING

All predictions were generated by running the models on the full nucleotide sequence of human chromosome 20 (NC_060944.1 in GCF_009914755.1 genome assembly). For models whose names specify an input sequence length, the chromosome sequence was processed using a non-overlapping sliding window of the corresponding size.

APPENDIX C. PRETRAINING

The dataset contains 443 vertebrate genome assemblies (all vertebrate species that have available genome and transcripts annotation on NCBI RefSeq database). For each gene and pseudogene, regions extending 16 kbp upstream and 8 kbp downstream from each unique TSS were extracted. In cases of overlapping regions, these regions were merged using BEDTools to obtain non-overlapping intervals. Each resulting non-overlapping interval is represented in the dataset by two samples: the forward sequence and its reverse complement. Sequences containing ambiguous nucleotides (i.e., symbols other than A, C, T, or G) were entirely excluded. Each genome assembly, except for the human one (GCF_000001405.40), was split into training and validation sets, constituting approximately 90% and 10% of the total genome length, respectively. The partitioning was executed at the level of whole chromosomes. In the case of the human genome assembly, chromosomes 8, 20, and 21 were assigned to the validation set, with the remaining chromosomes allocated to the training set.

APPENDIX D. DECLARATION OF LLM USAGE.

Large Language Models (LLMs) were used solely to improve the readability and clarity of the manuscript text. No parts of the analysis, results, or conclusions were generated by LLMs.