SegmentNT: annotating the genome at single-nucleotide resolution with DNA foundation models

Bernardo P. de Almeida*[1], Hugo Dalla-Torre*[1], Guillaume Richard[1], Christopher Blum[2],
Lorenz Hexemer[2], Maxence Gélard[1], Priyanka Pandey[2], Stefan Laurent[2], Alexandre Laterre[1],
Maren Lang[2], Uğur Şahin[2], Karim Beguir[1], Thomas Pierrot†[1]
[1]**InstaDeep Ltd, London, UK,** [2]**BioNTech, Mainz, Germany**

Abstract

Foundation models have achieved remarkable success in several fields such as natural language processing, computer vision and more recently biology. DNA foundation models in particular are emerging as a promising approach for genomics. However, so far no model has delivered granular, nucleotide-level predictions across a wide range of genomic and regulatory elements, limiting its practical usefulness. In this paper, we build on our previous work on the Nucleotide Transformer (NT) to develop a segmentation model, SegmentNT, that processes input DNA sequences up to 30kb length to predict 14 different classes of genomics elements at single nucleotide resolution. By utilizing pre-trained weights from NT, SegmentNT surpasses the performance of several ablation models, including convolution networks with one-hot encoded nucleotide sequences and models trained from scratch. SegmentNT can process multiple sequence lengths with zero-shot generalization for sequences of up to 50kbp. We show improved performance on the detection of splice sites throughout the genome and demonstrate strong nucleotide-level precision. Because it evaluates all gene elements simultaneously, SegmentNT can predict the impact of sequence variants not only on splice site changes but also on exon and intron rearrangements in transcript isoforms. Finally, we show that a SegmentNT model trained on human genomics elements can generalize to elements of different species and that a trained multispecies SegmentNT model achieves stronger generalization for all genic elements on unseen species. In summary, SegmentNT demonstrates that DNA foundation models can tackle complex, granular tasks in genomics at a single-nucleotide resolution. SegmentNT can be easily extended to additional genomics elements and species, thus representing a new paradigm on how we analyze and interpret DNA. We make our SegmentNT-30kb human and multispecies models available on our github repository in Jax and HuggingFace space in Pytorch.

Introduction

The intersection of genomics research and deep learning methods is profoundly changing our ability to understand the information encoded in each of the 3 billion nucleotides in the human genome and to accurately assess their influence with respect to different gene-regulatory activity layers, ranging from regulatory elements and transcriptional activation to splicing and polyadenylation [1, 2]. Sequence-based machine learning models trained on large-scale genomics data capture complex patterns in the sequence and can predict diverse molecular phenotypes with great accuracy. Recently, convolutional neural networks have demonstrated superior performance over other architectures across most sequence-based problems [3, 4, 5, 6, 7, 8, 9, 10, 11], sometimes combined with LSTMs [12, 13, 14, 15] or transformer layers [16, 17].

Most genomics models are built with a focus on only one specific task where one task is to annotate that a gene segment belongs to a specific group of genomic elements, for example detecting the presence of promoter elements in a given input sequence [18] or the binding of transcription factors [8]. Given the diversity and complexity of the different gene-regulatory activity processes, models that can tackle different types of tasks simultaneously will be easier to adopt by the community and should also obtain higher performance on each task by leveraging shared knowledge between tasks. Models compatible with different type of tasks have emerged using either multi-task supervised training schemes from scratch [5, 3, 19, 16, 17] or making use of large pre-trained DNA foundation models that are afterwards finetuned towards specific tasks [20, 21, 22, 23, 24, 25]. This last approach in particular is very promising for genomics given the ability of such foundation models to be trained on unlabeled data (e.g. raw genomes or experimental sequencing data), creating general-purpose representations capable of solving a multitude of downstream tasks, similarly to what has been observed in other fields such as natural language processing and computer vision [26, 27, 28, 29, 30].

---

*These authors contributed equally

†**Corresponding authors**: t.pierrot@instadeep.com

A second key feature of most genomics models trained on certain tasks, such as detecting promoter elements in an input sequence, is their limited resolution, usually predicting a single probability or quantitative score for the whole candidate sequence [18] or low-resolution continuous signals averaged across windows of 100–200 base pairs [16]. While framing such tasks as classification has practical advantages, this formulation has its limits in practice as we are interested in knowing precisely where elements are located in the sequence. In addition, it does not make use of additional information related to the spatial position of such elements. Models that make predictions at nucleotide-resolution were shown to improve performance and recover better features over previous deep learning classification approaches on tasks related to transcription factor binding [8], chromatin accessibility [31, 32] and RNA polyadenylation [10]. Developing models that can solve multiple tasks and at this nucleotide-level resolution is thus a promising avenue for the field.

Here we aim to train a model to predict the location of several types of genomics elements in a sequence at single-nucleotide resolution, both improving the model detection performance but also providing more refined annotations and predictions for an input sequence. Given the similarities between localizing elements at nucleotide resolution in a DNA sequence and localizing objects in images at pixel resolution, usually referred to as segmentation task [33, 34, 35], we adopted a segmentation architecture that proved useful in that field. More specifically, we built a DNA segmentation model, the Segment-Nucleotide Transformer (SegmentNT), that combines the pre-trained DNA foundation model Nucleotide Transformer (NT) [22] and a 1D U-Net [33] architecture, and trained it to predict the location of 14 types of human regulatory and gene elements in input sequences up to 30kb at single-nucleotide resolution. We show that SegmentNT achieves high nucleotide accuracy for all elements and generalizes to input sequences up to 50kb. We further finetuned our best SegmentNT-30kb model on multiple species and show improved generalization to unseen species.

To the extent of our knowledge, no model capable of predicting element locations at the nucleotide level for different sorts of elements, including gene and regulatory elements, has been developed so far, except for acceptor and donor splice sites identification [36, 37] or cross-species gene annotation [12]. Given the complexity of this task, this work demonstrates the benefit of leveraging pre-trained DNA foundation models over specialized methods trained from raw DNA sequences, showcasing the power of foundation models to tackle complex tasks in genomics and at single-nucleotide resolution.

Results

SegmentNT: finetuning Nucleotide Transformer for segmentation of DNA sequences at nucleotide resolution

SegmentNT is a DNA segmentation model that combines the pre-trained DNA foundation model Nucleotide Transformer (NT) [22] and a segmentation head to detect elements at different scales (Fig. 1a). As segmentation head we make use of a 1D U-Net architecture that downscales and upscales the foundation model embeddings of the input DNA sequence (Fig. 1b; see also Linder et al.[17] for a recent use-case of U-Net in genomics). This architecture is trained end-to-end on a dataset of genomic annotations to minimize a focal loss objective [34] to deal with element scarcity in the dataset (see Methods).

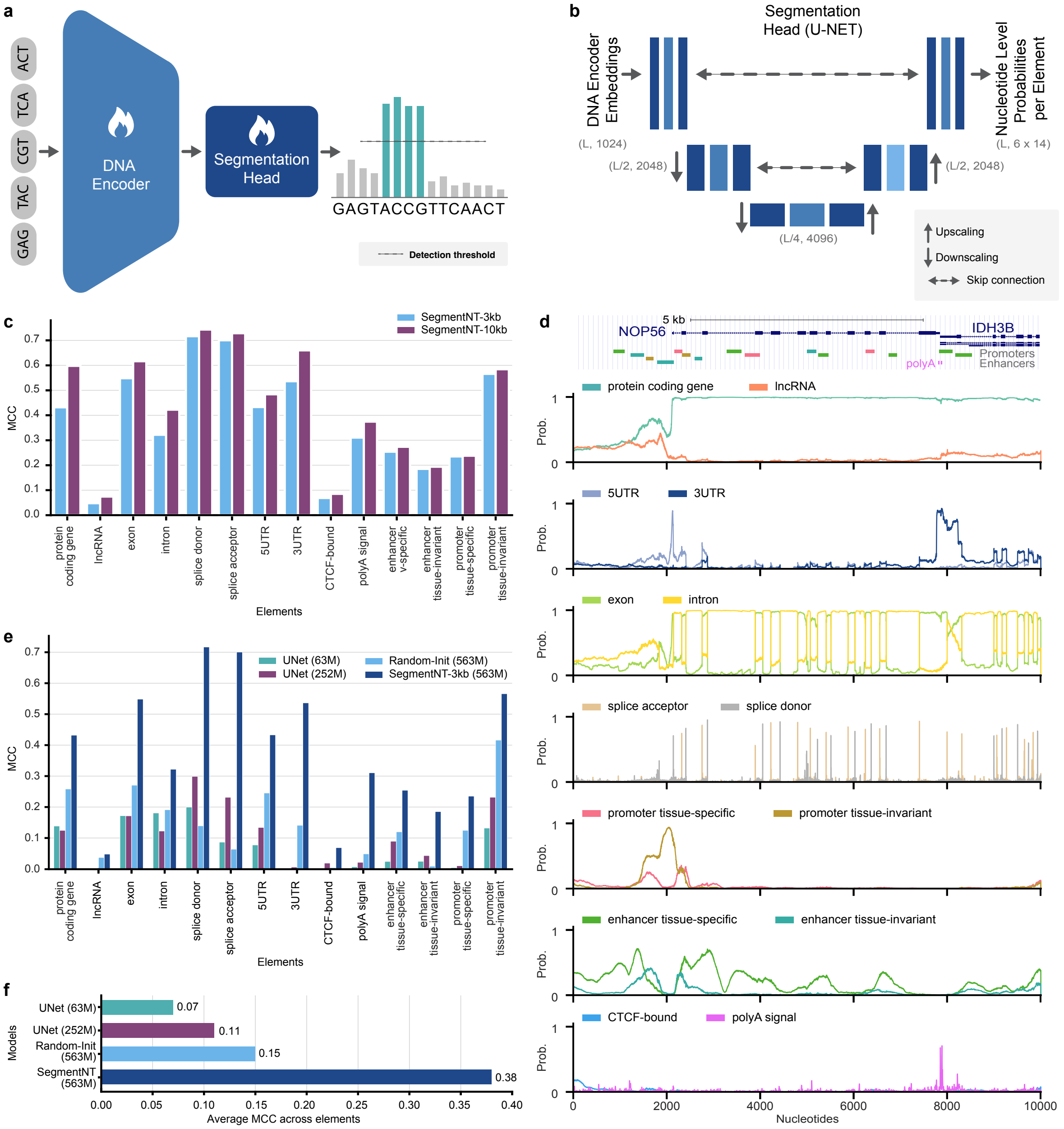


Figure 1: **SegmentNT localizes genomics elements at nucleotide resolution. a)** The SegmentNT neural network architecture consists of a pre-trained DNA encoder (here Nucleotide Transformer (NT) [22]) and a segmentation head (here a U-Net). The output are probabilities for each genomics element at nucleotide resolution. **b)** As segmentation head we use a 1D U-Net architecture with 2 downsampling and 2 upsampling convolutional blocks with matched U-Net connections. We added the dimensions of each layer. **c)** Performance of SegmentNT trained on 3kb and 10kb sequences on 14 types of genomics elements. We used as metric the Matthews correlation coefficient (MCC). **d)** Representative example of annotations and predicted probabilities of the 14 types of genomics elements at the *NOP56*/*IDH3B* gene locus located in the test set. Gene isoforms with respective exons and introns, as well as promoter and enhancer regulatory elements are shown. **e)** Comparison of MCC performance between SegmentNT and different architectures on 3kb input sequences. The number of parameters of each model in millions (M) is shown. **f)** Average MCC performance of the different architectures across the 14 elements.

To train SegmentNT we curated a dataset of annotations at nucleotide-level precision for 14 types of genomic elements in the human genome derived from GENCODE [38] and ENCODE [39], including gene elements (protein-coding genes, lncRNAs, 5'UTR, 3'UTR, exon, intron, splice acceptor and donor sites) and regulatory elements (polyA signal, tissue-invariant and tissue-specific promoters and enhancers, and CTCF-bound sites) (Supplementary Fig. 1; see Methods). Since these element annotations can overlap, SegmentNT predicts the probability of belonging to each of the genetic elements at nucleotide level. For example, in different gene transcript isoforms the same DNA region can be considered an exon or an intron, enhancers can also be found in gene regions, and polyA signals are usually in the gene's 3'UTRs. In addition, here we used the canonical definition of exons as any part of a gene that can be present in the final mature RNA after introns have been removed by RNA splicing, thus also overlapping with 5' and 3'UTRs. This allows the prediction of every genomics element independent of the other predictions. The annotation of all promoter and enhancer regions in the human genome was derived from the latest registry of candidate cis-regulatory elements by ENCODE [40]. It contains 790k enhancers and 34k promoters grouped by their activity in different tissues (Supplementary Fig. 1c).

We first trained a model to segment these distinct 14 genomics elements in input DNA sequences of 3kb (SegmentNT-3kb). This model was further finetuned on 10kb input sequences (SegmentNT-10kb) to extend its input length. This was achieved by initializing SegmentNT-10kb from the best checkpoint of the SegmentNT-3kb model for a more efficient training and length-adaptation. For a given input sequence, these models make $42,000$ and $140,000$ predictions, respectively, each being the probability of a given nucleotide to belong to a genomics element type. Model training, validation and performance evaluation were performed on different sets of chromosomes from the human genome to ensure no data leakage between the different sets in order for the test set to provide a robust evaluation of model performance. SegmentNT-3kb demonstrated high accuracy in localizing these elements to nucleotide precision, showing a Matthews correlation coefficient (MCC) on the test set above 0.5 for exons, splice sites, 3'UTRs and tissue-invariant promoter regions (Fig. 1c). LncRNA and CTCF-binding sites were the most difficult elements to predict, with test MCC values below 0.1. We observed superior performance of the model in sequences of 10kb (average MCC of 0.43) compared with 3kb (0.38), in particular for protein-coding genes, 3'UTRs, exons and introns, suggesting that these elements depend on longer sequence contexts (Fig. 1c).

To further evaluate predictive performance, we inspected regions of the held-out test chromosomes. Evaluating SegmentNT-10kb on a 10kb window that covers the gene *NOP56* on the positive strand and the end of the gene *IDH3B* on the negative strand shows that it accurately predicts the different genic elements of each gene (Fig. 1d). SegmentNT correctly predicts both genes as protein-coding, their 5'UTR and 3'UTR positions, their splice sites and exon-intron structure, and also the polyA signals. In addition, SegmentNT captures the promoter region of *NOP56*, both the tissue-specific and tissue-invariant ones. This region also contains multiple enhancers and some of those are correctly predicted by the model. Still, although our global performance metric for enhancers is good (MCC of 0.27 for tissue-specific and 0.19 for tissue-invariant for SegmentNT-10kb), we observe that enhancer predictions are more noisy. This could be related to their higher sequence complexity and diversity, and we expect that grouping them by cell type-specific activity should further improve model performance (see Discussion).

Using Nucleotide Transformer as a pre-trained DNA encoder is essential for efficient training and to achieve top performance

We next evaluated our model architecture and the importance of using the NT pre-trained foundation model as a DNA encoder. We compared the performance of SegmentNT with three different model architectures, using 3kb input sequences for a simpler comparison (see Methods). We removed the NT DNA encoder and trained two 1D U-Net architectures that take one-hot encoded DNA sequences directly as input instead of the NT embeddings. One with the same 63M parameters of the head of SegmentNT and a larger model with an additional downsampling/upsampling block featuring a total of 252M parameters. We selected for each model the checkpoint with the highest performance on the validation set and evaluated all on the same test set sequences. These two U-Net architectures demonstrated substantially reduced performance across all elements, with an average MCC of 0.07 (66M) and 0.11 (250M) compared with 0.38 for SegmentNT-3kb, demonstrating the value of using a pre-trained DNA encoder (Fig. 1e,f). We note that the largest U-Net architecture used here (252M) is around half the parameter size of SegmentNT (563M) with a size comparable to the Enformer [16].

To test the benefit of pretraining the NT foundation model, we trained a model version with the same architecture as SegmentNT but using a randomly initialized NT DNA encoder model, rather than the pre-trained one. We first note that while for SegmentNT-3kb we observed model convergence after 20M training sequences (10B tokens), the version with random initialized NT showed much slower convergence and have not converged yet even after 68M training sequences (34B tokens), a training more than three times longer. In addition, even after this longer training, performance of the randomly initialized model (average MCC 0.15) was substantially lower than SegmentNT-3kb (0.38) across all 14 genomics elements (Fig. 1e,f). In summary, SegmentNT demonstrates the value of DNA foundation models for solving challenging tasks in genomics such as localizing different types of genomics elements at a single nucleotide resolution.

SegmentNT outperforms alternative approaches in predicting regulatory elements with nucleotide-precision

We next focused on predicting regulatory elements and evaluated alternative approaches. To our knowledge there are no models that can predict the location of regulatory elements in an input sequence at nucleotide resolution. We considered two approaches that could be used to tackle this problem: sliding a binary classifier over the input 10kb sequence and using the Enformer [16] chromatin predictions as a surrogate for regulatory elements. For a more direct comparison with our model, we used as binary classifiers the Nucleotide Transformer models [22] finetuned on promoter or enhancer sequences (see Methods). We compared these approaches for the prediction of tissue-invariant promoters and tissue-specific enhancers on 10kb input sequences as these were the classes with the highest sequence predictive value (Fig. 1c).

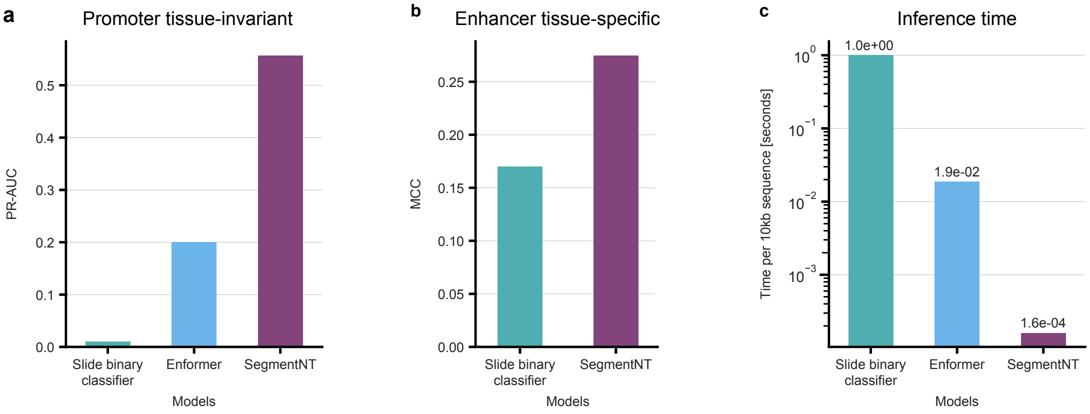


Figure 2: **Comparison between SegmentNT and alternative segmentation approaches for promoter and enhancer predictions.** **a)** Precision-Recall Area Under the Curve (PR-AUC) performance for sliding a binary promoter classifier, Enformer and SegmentNT for segmenting tissue-invariant promoters. **b)** MCC performance for sliding a binary enhancer classifier and SegmentNT for segmenting tissue-specific enhancers. **c)** Inference times on a 10kb sequences between Segment-NT, sliding a similar-size binary classifier model and Enformer.

On predicting promoters at nucleotide precision, SegmentNT-10kb outperformed both approaches (Fig. 2a,b). Using the promoter finetuned model from NT [22] and sliding it through each 10kb sequence yielded very low performance, which can be related with the different set of promoter sequences used for training such model. To use the Enformer, we calculated the predictions of DNA accessibility for 7 different cell lines for each 10kb sequence at its original 128bp resolution bins and averaged to get a more robust DNA regulatory activity metric. Despite the different approach and dataset, this resulted in a good performance of 0.21 Precision-Recall Area Under the Curve (PR-AUC) for the prediction of promoter regions, but still well below SegmentNT-10kb with a PR-AUC of 0.56 (Fig. 2a). For predicting enhancers we compared the NT model finetuned on human enhancers [22] and followed the same sliding window approach. Here the performance was better than for promoters, with an MCC of 0.17, but again SegmentNT-10kb achieved much better performance at 0.27 (Fig. 2b).

In addition to being a novel approach for predicting regulatory elements and achieving state-of-the-art performance, SegmentNT is also much faster on inference. Our SegmentNT-10kb model segments the 14 genomics elements in an input 10kb sequence (meaning $140,000$ predictions) in 0.16 milliseconds. This inference time is about 100x faster than running the Enformer model (18 milliseconds) and 5000x faster than sliding a similar-size binary classifier model over the sequence (1 second) (all times using Jax code and in a single A100 GPU; Fig. 2c). We note that for Enformer we had to pad the 10kb sequences for inference, since the original model predicts scores for $114,688$bp.

SegmentNT generalizes to sequences up to 50kb

We next investigated how to extend the sequence context length of SegmentNT, motivated by the improved results observed for SegmentNT-10kbp over SegmentNT-3kbp (Fig. 1c). However, NT uses rotary positional embeddings (RoPE; [41]) which was set to support sequences up to 12kb during its pre-training. As such, and given the periodic nature of RoPE encoding, using NT directly on sequences longer than 12kb, whether for finetuning or inference, would yield poor performance. To address this problem, we explored recent approaches that have been proposed for extending contexts of RoPE models by converting the problem of length extrapolation into one of ”interpolation”. Specifically, we employ a context length extension method first formally described in [42], where the frequency used in RoPE embeddings is re-scaled to account for longer sequences (see also [43, 44]). This approach can be used for extending the context length of SegmentNT during training to train it on sequences longer than 12kb but also for performing inference with SegmentNT models on sequences longer than the ones seen during training. We investigated both scenarios below.

We implemented context length extension in NT and trained two additional SegmentNT models that segment the 14 genomics elements in DNA sequences of 20kb (SegmentNT-20kb) and 30kb (SegmentNT-30kb) (see Methods). Evaluation on the same test chromosomes showed consistent improvements in performance with increased sequence length, in particular for the segmentation of protein-coding genes, 3’UTRs, exons and introns (Fig. 3a). The model with the best performance across all elements was SegmentNT-30kb with an average MCC of 0.46 (Fig. 3b).

Since it is computationally expensive to finetune SegmentNT on even longer sequence lengths, we tested if we could leverage context length extension to evaluate a model pre-trained on a given length on longer sequences. We tested this approach on the SegmentNT-10kb model and evaluated it with or without context length extension on the prediction of sequences up to 100kb from the same test chromosomes (Fig. 3c, Supplementary Fig. 2). Context length extension substantially improved the performance of the model on longer sequences, in particular on 100kb where the original model showed very poor performance (average MCC of 0.26 vs 0.07, respectively).

This motivated us to more systematically test how far our different SegmentNT models could be extended. To address that, we evaluated the performance of all trained SegmentNT models (3kb, 10kb, 20kb and 30kb) on input sequence lengths between 3 and 100kb using context length extension interpolation when needed. When averaging the performance across 14 elements, this revealed that the model trained on the longest context length (SegmentNT-30kb) achieved the best results when evaluated in all context lengths, including shorter sequences (Fig. 3d, Supplementary Fig. 3a,b). We observed top performance for 50kb input sequences (average MCC of 0.47; Fig. 3d) and a drop in performance for sequences longer than 50kb, although the model still has good performance on sequences of 100kb (0.45; Supplementary Fig. 3a). These results highlight the flexibility of SegmentNT

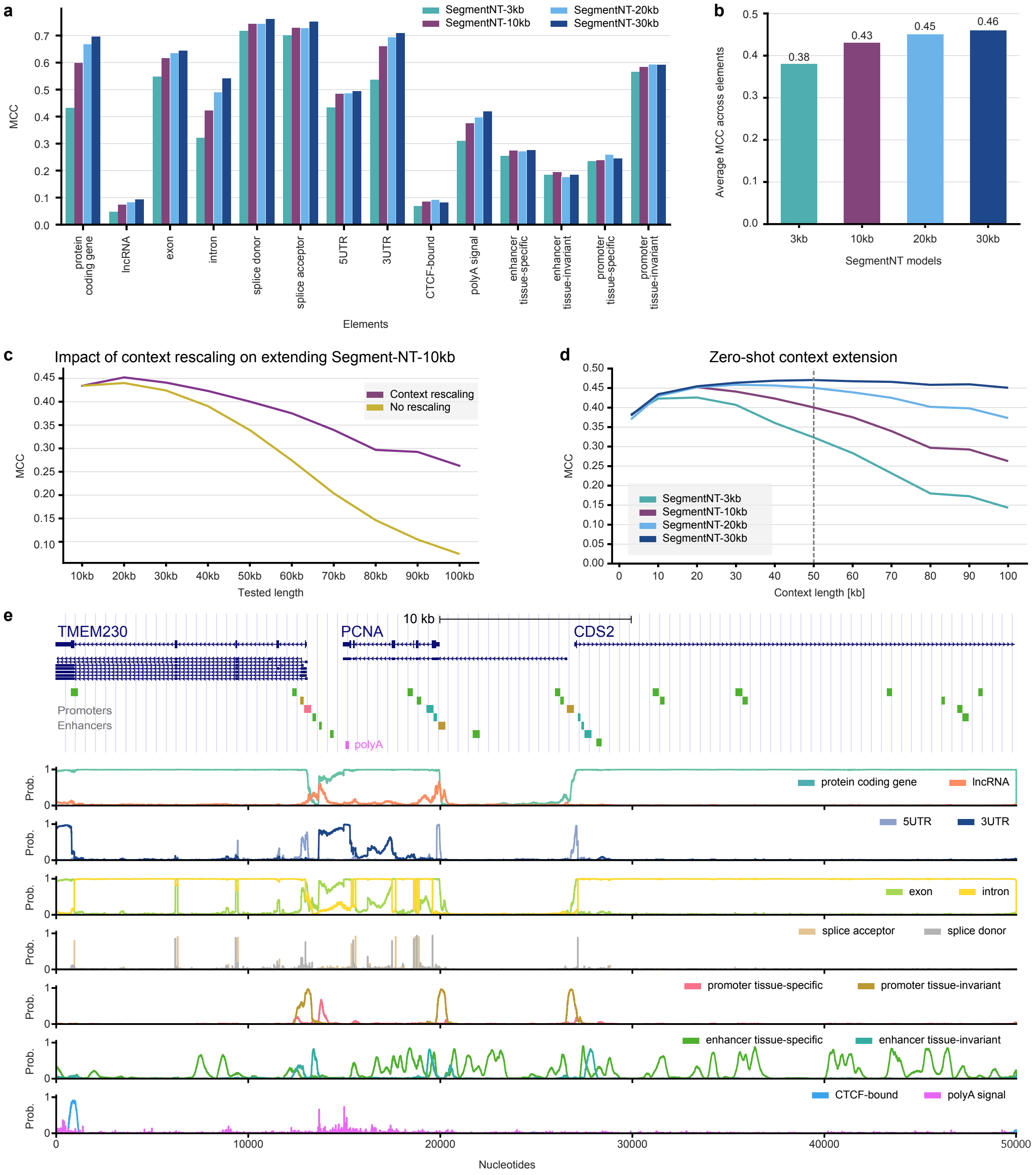


Figure 3: **Adaptation and zero-shot generalization of SegmentNT across multiple sequence lengths. a)** Performance of SegmentNT trained on 3kb, 10kb, 20kb and 30kb sequences on 14 types of genomics elements. We used as metric the MCC. **b)** Average MCC performance of the different models across the 14 elements. **c)** Context-length extension allows to rescale SegmentNT-10kb to 100kb sequences. Average MCC performance across the 14 elements for the SegmentNT-10kb model with and without context-length rescaling. **d)** Long-range models improve generalization on longer contexts while maintaining performance on shorter contexts. Average MCC performance across the 14 elements for the different SegmentNT models per input sequence length. **e)** Representative example of annotations and predicted probabilities of the 14 types of genomics elements for a 50kb region at the *TMEM230*/*PCNA*/*CDS2* gene locus located in the test set. Gene isoforms with respective exons and introns, as well as promoter and enhancer regulatory elements are shown.

and how it can be applied to sequences of different lengths. We note that the SegmentNT-30kb model when segmenting the 14 genomics elements in an 50kb input sequence makes 700, 000 predictions at once (14 x 50, 000), thus providing a very rich segmentation output. See an example of the SegmentNT-30kb predictions for a 50kb locus in the test set with three overlapping genes (Fig. 3e).

The results of the model extension per genomic element type also reveal their different sequence-dependencies (Supplementary Fig. 3c). Protein-coding gene was the element more dependent on longer sequence contexts, achieving its top prediction performance when evaluating 70kb sequences (MCC of 0.75, compared with 0.42 for 3kb and 0.58 for 10kb; Fig. 3e). We observed similar patterns for most elements, including introns (MCC of 0.6 for 90kb, compared with 0.31 for 3kb and 0.44 for 10kb). Splice sites were less dependent on longer contexts and showed minor improvements between 3kb (acceptor MCC: 0.71, donor: 0.72) and longer contexts (acceptor: 0.74 at 10kb and 0.75 at 60kb, donor: 0.75 and 0.76) (Supplementary Fig. 3a). These results highlight how different biological properties have specific dependencies on different sequence contexts.

Segment-NT accurately predicts splice sites and mutations

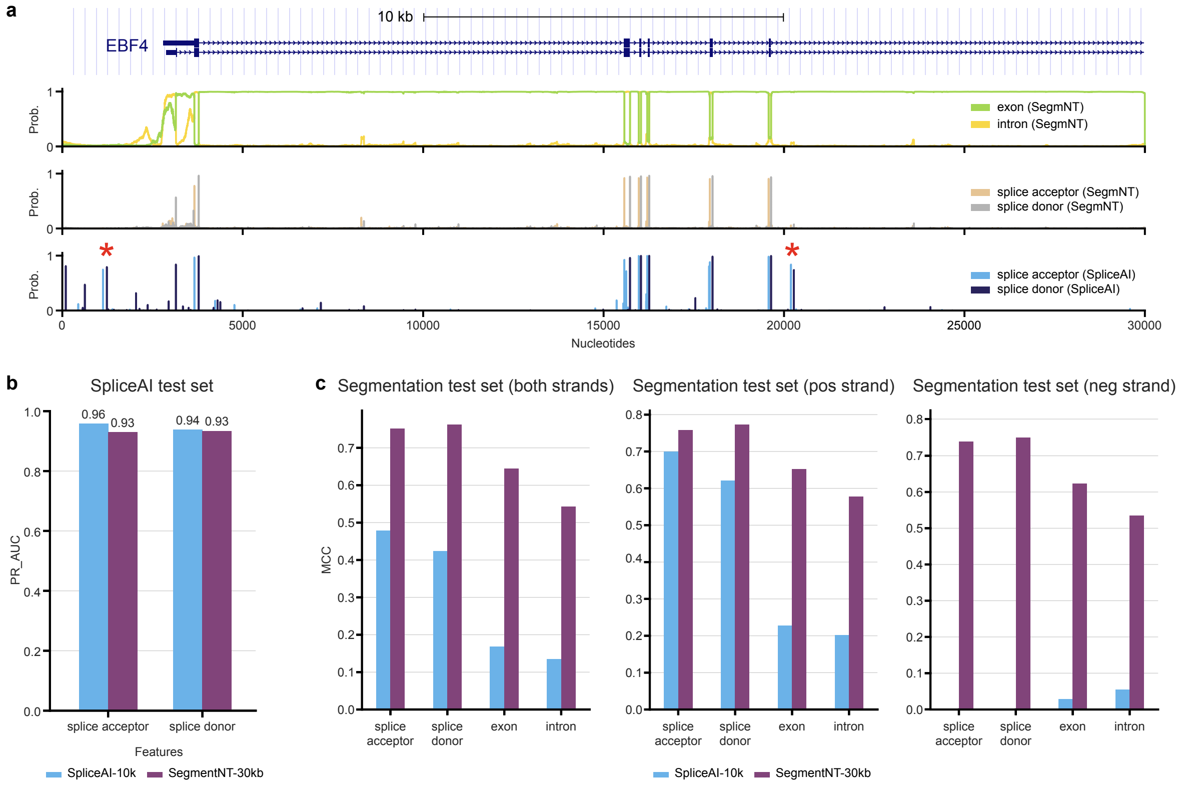


Figure 4: **Segment-NT achieves state-of-the performance on splice site prediction. a)** Representative example of gene annotations and predicted probabilities for splicing elements by SegmentNT-30kb and SpliceAI at the *EBF4* gene locus located in the test set. Gene isoforms with respective exons and introns, as well as promoter and enhancer regulatory elements are shown. SpliceAI mispredictions are highlighted with stars. **b)** Performance of SpliceAI and SegmentNT-30kb for splice acceptor and donor detection on SpliceAI's gene-centric test set. We used as metric the PR-AUC. **c)** Performance of SpliceAI and SegmentNT-30kb for splice acceptor and donor detection as well as exon and intron prediction on SegmentNT's whole chromosome test set. We used as metric the MCC and report performance for all regions (left), or regions containing genes only in the positive (middle) or negative strand (right).

One of the main nucleotide-level tasks in genomics that has been tackled by previous models is splice site detection, where SpliceAI is considered state-of-the-art [36]. We first compared our best

SegmentNT-30kb with the specialized SpliceAI-10kb model on detecting splice donor and acceptor nucleotides on a gene from our test set (*EBF4*; Fig. 4a). SegmentNT correctly predicts all exons and introns in addition to all splice sites, including the ones of the alternative exon at the gene start. When comparing both models we observe that SpliceAI predicts all existent splice sites but overpredicts additional sites (see red stars in Fig. 4a).

For a systematic comparison, we evaluated each model in both SpliceAI's test set and our test set given their differences. Specifically, SpliceAI was trained and tested solely on pre-mRNA transcripts from protein-coding genes, without intergenic sequences, and with transcript sequences always in the respective positive strand. In contrast, our training and test sets are more general and contain the whole DNA sequence of the respective chromosomes, including protein-coding genes and lncRNAs in both positive and negative orientation.

SegmentNT-30kb achieves comparable performance to SpliceAI on SpliceAI's test set: PR-AUC for acceptor sites of 0.93 vs 0.96, and for donor sites of 0.93 vs 0.94, respectively (Fig. 4b). The result is the same if using only 10kb input sequences, the length used for SpliceAI training (PR-AUC acceptor: 0.92 vs 0.94, donor: 0.92 vs 0.87, Supplementary Fig. 4a). On SegmentNT's whole genome test set our model achieves substantially improved performance when considering all genes (acceptor: 0.75 vs 0.48, donor: 0.76 vs 0.42) or only the ones in the positive orientation (acceptor: 0.76 vs 0.70, donor: 0.77 vs 0.62; Fig. 4c). As expected given its training data constraints, SpliceAI cannot predict splice sites when the gene is in the negative orientation, while SegmentNT maintains the same performance (acceptor: 0.74 vs 0.00, donor: 0.75 vs 0.00). Similar improvements were observed when considering 10kb sequences as input (Supplementary Fig. 4b). Overall, SegmentNT accurately detects splice donor and acceptor sites in both strands in any given input DNA sequence.

Another difference to SpliceAI is that SegmentNT also predicts the position of exons and introns. This can only be achieved with SpliceAI by combining the splice donor and acceptor predictions a posteriori into exon and intron segments. We use SpliceAI to predict the position of exons and introns and compare with the segmentation predictions of SegmentNT. Here, SegmentNT also showed improved performance (Fig. 4c).

This prediction of splice sites together with exon and intron segments by SegmentNT also allows for the direct prediction of potential transcript isoforms for a given DNA sequence. Given the accuracy of SegmentNT's predictions, we next tested if it could evaluate the effect of sequence variants on isoform structures. We used data from an experimental saturation mutagenesis splicing assay of the exon 11 of the gene *MST1R*, flanked by constitutive exons 10 and 12 and respective introns (data from Braun, Simon, et al. [45]; see Methods). This dataset contains a library of almost 5,800 randomly mutated minigenes of ∼700nt, where for each minigene variant it was evaluated the splicing of the alternative exon 11 in the respective mRNA molecules. We used this data to test if SegmentNT could predict the impact of those sequence variants on the respective splicing and transcript isoforms.

As a first check, SegmentNT correctly predicts this minigene as protein-coding, and the respective locations of all splice sites, the three exons and the two introns (Supplementary Fig. 5a). We next evaluated sequence variants with different experimentally measured impacts in the minigene transcripts. In Supplementary Figure 5b we show a minigene variant that leads to higher exon 11 inclusion, which is correctly predicted by SegmentNT- note the stronger exon prediction compared to the wildtype sequence, accompanied by stronger flanking intron and splice site predictions. In Supplementary Figure 5c we show a minigene variant where the exon is skipped with high frequency. SegmentNT correctly predicts the loss of splice sites and of the respective exon, with higher prediction of an intron at its place. Systematic correlations across all minigene variants revealed a strong agreement between the exon predictions by SegmentNT and the inclusion of the alternative exon 11 (PCC: 0.24; Supplementary Fig. 5d-g). These results show that the segmentation capabilities of SegmentNT can be used to predict complex gene rearrangements directly from the sequence, which should be a useful tool for the interpretation of sequence and structural variants that can affect gene regulation and disease.

Zero-shot generalization of SegmentNT across species

We next explored how SegmentNT trained on human genomics elements could generalize to other species (Fig. 5a). Gene annotations for more distant, less-studied species are less accurate, while annotations of regulatory elements such as promoters and enhancers are very scarce. Thus, models that can predict these elements for different species hold great potential. In addition, comparison of predictions across species should provide insights about the evolutionary constraints of each element.

For this analysis, we selected 17 additional species and for each one curated a dataset of annotations for the 7 main genomic elements available from Ensembl [46], namely protein-coding gene, 5'UTR, 3'UTR, intron, exon, splice acceptor and donor sites (see Methods). This allows us to evaluate the performance of the human model in each species on the 7 element types, while for the other 7 elements our predictions might be informative of potential regulatory regions. Similar to the human datasets, each dataset was split in train, validation and test chromosomes. We selected our best model trained on the human 14 genomics elements, SegmentNT-30kb, and evaluated it on each species test set.

We observe high zero-shot performance of the human SegmentNT-30kb model across species for exon and splice sites, correlating with their high evolutionary conservation (Fig. 5b,c). For the other elements the performance was good for related species like gorilla and macaque, but dropped for more evolutionary-distant animals. This shows that the SegmentNT-30kb model can generalize to some extent to other species, but that the performance depends on the evolutionary distance of the genomics elements and species.

Multispecies SegmentNT model shows improved species generalization

Since gene elements have evolved and therefore their sequence determinants might differ between species, we trained an additional, multispecies model (SegmentNT-30kb-multispecies) by finetuning the human SegmentNT-30kb model on the genic annotations of 5 selected species: mouse, chicken, fly, zebrafish and worm (see Methods). The remaining 12 species were kept as held-out test set species for comparing the generalization capabilities of the human and multispecies models. We note that since most training species have limited annotation of regulatory elements, we focused this multispecies model only on genic elements and therefore it should not be used for the prediction of regulatory elements. The performance of the SegmentNT-30kb-multispecies model improved quickly during finetuning, leveraging its knowledge of human elements. We observed an improved performance across species for the SegmentNT-30kb-multispecies model over the human SegmentNT-30kb model (Fig. 5d), showing that gene elements diverged between species and it is necessary to adjust the model accordingly.

Finally, we evaluated both human and multispecies SegmentNT-30kb models on the held-out set of 12 species, splitting them in two groups: 7 with an estimated divergence time from human of less than 100 million years (human-close species) and 5 more distant (more than 100 million years; human-distant; data from TimeTree) (Supplementary Fig. 6). The human model generalizes well for unseen species and showed better performance for human-close (average MCC of 0.62) than human-distant species (average MCC of 0.49; Fig. 5e,f). SegmentNT-30kb-multispecies demonstrated similarly good performance on human-close species (average MCC of 0.64) and improved performance on human-distant species (average MCC of 0.57) over the human model (0.49; Fig. 5e,f). This SegmentNT-30kb-multispecies model is thus more general and can generalize to species not included in the training set (Fig. 5g). Altogether, these results show that SegmentNT can be easily extended to additional genomics elements and species, which opens up promising new research directions to be explored in future work.

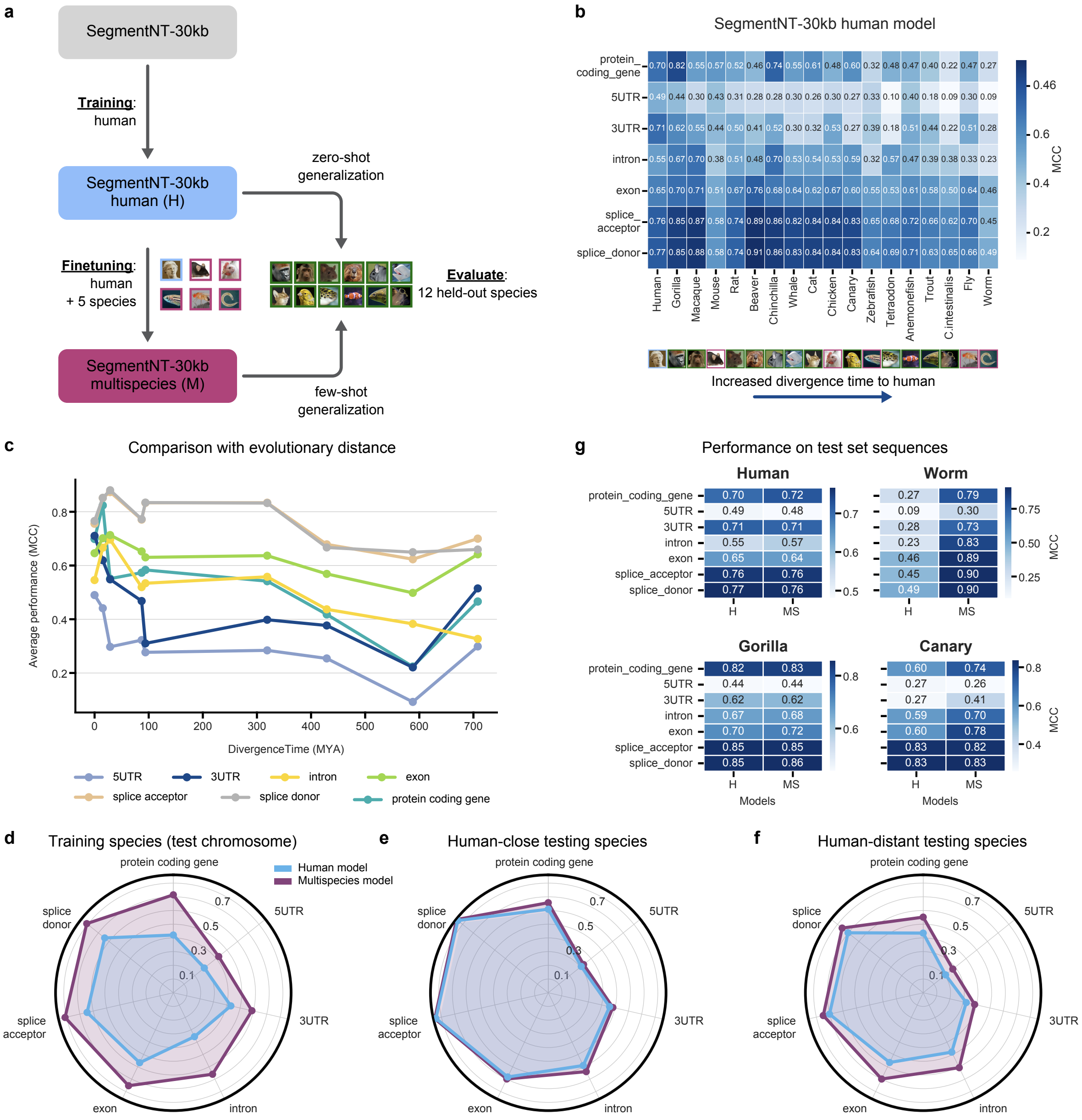


Figure 5: **SegmentNT generalizes across species.** **a)** Cartoon explaining zero-shot and few-shot specie generalization of SegmentNT. SegmentNT-30kb-multispecies was trained on genic elements from human plus 5 additional species. Both human (zero-shot) and multispecies (few-shot) models were evaluated on a held-out test set of 12 species. **b)** Performance of the human model on the gene elements of all species, sorted by divergence time to human. We used as metric the MCC. **c)** Comparison between MCC performance and divergence time per gene element. The MCC was averaged for species with the same evolutionary distance. **d-f)** Radar plot depicting the performance of the human and multispecies SegmentNT models per element for **(d)** species in the training set, **(e)** human-close species in test set, and **(f)** human-distant species in test set. **g)** Performance of the human (H) and multispecies (MS) model per element for 4 representative species.

Discussion

SegmentNT is an extension of the DNA foundation model NT towards predicting the location of several types of genomics elements in DNA sequences up to 50kb at nucleotide resolution. We show top performance for genic elements, including splice sites, and how each element depends on different context windows. For a given 50kb sequence, SegmentNT makes $700,000$ predictions at once in less than one millisecond allowing to annotate any input sequence in a very efficient way. SegmentNT trained on the human genome can already generalize to other species, but to make SegmentNT more broadly applicable to annotate sequences from different species we developed a multispecies version that improves generalization to unseen species. We make our best models (SegmentNT-30kb human and multispecies) available on our github repository and HuggingFace space.

SegmentNT provides strong evidence that DNA foundation models can tackle complex tasks in genomics at single-nucleotide resolution. Up until now, there is no consensus for the benefit of pre-trained foundation models for genomics. There has been limited improvements on most tasks where these models have been evaluated on [21, 22, 23, 25, 47, 43]. Here we focused on a more challenging task of segmenting genomc elements in DNA sequences at nucleotide resolution. Our results show that the highest performance is achieved by combining a pre-trained NT and a segmentation U-Net head, when compared with applying such segmentation architectures directly from one-hot encoded DNA sequences. This is a strong evidence for the value added by such pre-trained models and points to the need of expanding their applications and evaluations to more realistic tasks in genomics.

A current limitation of DNA foundation models is their limited context length. NT was the pre-trained model with the largest context length at its time, trained on sequences of up to 12kb [22]. Since then different approaches have been proposed to extend the context of such models, mostly by relying on novel state-space architectures to avoid the quadratic scaling of Transformers [23, 48, 44]. Here we took a different approach and extended the context of SegmentNT through context-length extrapolation in both training and evaluation phases, showing improved performance for sequences up to 50kb (see also [43]). We expect that extending the context of NT and SegmentNT models to longer sequences with efficient context-extension approaches will yield further improvements for DNA segmentation tasks. Many techniques have recently emerged in fields like natural language processing that manage to increase the input length of Transformer models to process hundreds of thousands of tokens at a time [49, 50, 51, 52]. These approaches together with the new developments of state-space models provide promising avenues to build the next generation of models.

We observed lower performance for the segmentation of promoter and enhancer regulatory elements compared with genic elements. Indeed, the sequence code of human regulatory elements is vastly more complex and unstructured, where for example the same element can encode different syntax in different cell types [53]. To account for some of this complexity we have split promoters and enhancers in tissue-invariant and tissue-specific classes each, and observed different predictive performances between the groups. In future work we expect that splitting promoters and enhancers by their specific cell types should allow the model to learn the different cell type-specific regulatory codes thus improving the performance on regulatory element prediction.

An important result of our work is the demonstration that SegmentNT trained on human genomics elements can generalize to unseen species. The generalization is stronger for splice sites and exons, likely due to their high conservation. In addition, we observed reduced generalization for species with longer divergence times to human. To improve the generalization to more distant species, we developed a SegmentNT-multispecies version that shows improved performance on unseen species. Thus, this model can be leveraged to annotate sequences up to 50kb of any species *de novo* which should be useful to explore the genomes of less-characterized species.

Overall, our work has several direct applications. First, the finetuned DNA encoder within SegmentNT should provide stronger representations of human genomics elements and could be used to improve performance on downstream tasks [54]. Second, interpreting the representations learned by SegmentNT could reveal insights about the genome and its encoded information. Third, the accuracy of SegmentNT predictions can be leveraged to evaluate the impact of sequence variants on the different types of genomics elements, as we showed for splicing isoforms. Thanks to the extended sequence context and the prediction of several types of genomics elements, we foresee important applications for the analysis of cancer genomes and their large structural variants. Fourth, SegmentNT-multispecies can be directly applicable to annotate and explore the genomes of different species. Fifth, SegmentNT's architecture can be easily applied to additional genomics annotations or nucleotide-level experimental data. Increasing the number of channels per nucleotides predicted by SegmentNT to include data coming from multiple experiments and biological processes should improve the transfer between tasks and lead to generalisation in a way similar to the segment anything model for images [55]. We ultimately hope that SegmentNT can be a useful tool for the genomics community and foster new developments in our understanding of the genome code.

References

[1] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, “Deep learning: new computational modelling techniques for genomics,” *Nature Reviews Genetics*, vol. 20, no. 7, pp. 389–403, 2019.

[2] T. Yue, Y. Wang, L. Zhang, C. Gu, H. Xue, W. Wang, Q. Lyu, and Y. Dun, “Deep learning for genomics: From early neural nets to modern large language models,” *International Journal of Molecular Sciences*, vol. 24, no. 21, p. 15858, 2023.

[3] J. Zhou and O. G. Troyanskaya, “Predicting effects of noncoding variants with deep learning–based sequence model,” *Nature methods*, vol. 12, no. 10, pp. 931–934, 2015.

[4] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, “Predicting the sequence specificities of dna- and rna-binding proteins by deep learning,” *Nature Biotechnology*, vol. 33, p. 831–838, 2015.

[5] D. R. Kelley, J. Snoek, and J. L. Rinn, “Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks,” *Genome research*, vol. 26, no. 7, pp. 990–999, 2016.

[6] D. R. Kelley, Y. A. Reshef, M. Bileschi, D. Belanger, C. Y. McLean, and J. Snoek, “Sequential regulatory activity prediction across chromosomes with convolutional neural networks,” *Genome Research*, vol. 28, pp. 739–750, Mar. 2018.

[7] D. R. Kelley, “Cross-species regulatory sequence activity prediction,” *PLOS Computational Biology*, vol. 16, p. e1008050, July 2020.

[8] Ž. Avsec, M. Weilert, A. Shrikumar, S. Krueger, A. Alexandari, K. Dalal, R. Fropf, C. McAnany, J. Gagneur, A. Kundaje, *et al.*, “Base-resolution models of transcription-factor binding reveal soft motif syntax,” *Nature Genetics*, vol. 53, no. 3, pp. 354–366, 2021.

[9] B. P. de Almeida, F. Reiter, M. Pagani, and A. Stark, “Deepstarr predicts enhancer activity from dna sequence and enables the de novo design of synthetic enhancers,” *Nature Genetics*, vol. 54, no. 5, pp. 613–624, 2022.

[10] J. Linder, S. E. Koplik, A. Kundaje, and G. Seelig, “Deciphering the impact of genetic variation on human polyadenylation using aparent2,” *Genome Biology*, vol. 23, p. 232, 2022.

[11] V. Agarwal and D. R. Kelley, “The genetic and biochemical determinants of mrna degradation rates in mammals,” *Genome Biology*, vol. 23, p. 245, 2022.

[12] F. Stiehler, M. Steinborn, S. Scholz, D. Dey, A. P. Weber, and A. K. Denton, “Helixer: cross-species gene annotation of large eukaryotic genomes using deep learning,” *Bioinformatics*, vol. 36, no. 22-23, pp. 5291–5298, 2020.

[13] M. R. Amin, A. Yurovsky, Y. Tian, and S. Skiena, “Deepannotator: genome annotation with deep learning,” in *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pp. 254–259, 2018.

[14] D. Quang and X. Xie, “Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences,” *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107, 2016.

[15] L. Minnoye, I. I. Taskiran, D. Mauduit, M. Fazio, L. V. Aerschot, G. Hulselmans, V. Christiaens, S. Makhzami, M. Seltenhammer, P. Karras, A. Primot, E. Cadieu, E. van Rooijen, J.-C. Marine, G. Egidy, G. E. Ghanem, L. Zon, J. Wouters, and S. Aerts, “Cross-species analysis of enhancer logic using deep learning,” *Genome Research*, vol. 30, pp. 1815–1834, 2020.

[16] Ž. Avsec, V. Agarwal, D. Visentin, J. R. Ledsam, A. Grabska-Barwinska, K. R. Taylor, Y. Assael, J. Jumper, P. Kohli, and D. R. Kelley, “Effective gene expression prediction from sequence by integrating long-range interactions,” *Nature methods*, vol. 18, no. 10, pp. 1196–1203, 2021.

[17] J. Linder, D. Srivastava, H. Yuan, V. Agarwal, and D. R. Kelley, "Predicting rna-seq coverage from dna sequence as a unifying model of gene regulation," *bioRxiv*, pp. 2023–08, 2023.

[18] M. Oubounyt, Z. Louadi, H. Tayara, and K. T. Chong, "Deepromoter: robust promoter predictor using deep learning," *Frontiers in genetics*, vol. 10, p. 286, 2019.

[19] K. M. Chen, A. K. Wong, O. G. Troyanskaya, and J. Zhou, "A sequence-based global map of regulatory activity for deciphering human genetics," *Nature genetics*, vol. 54, no. 7, pp. 940–949, 2022.

[20] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri, "Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome," *Bioinformatics*, vol. 37, no. 15, pp. 2112–2120, 2021.

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

[22] H. Dalla-Torre, L. Gonzalez, J. Mendoza-Revilla, N. L. Carranza, A. H. Grzywaczewski, F. Oteri, C. Dallago, E. Trop, B. P. de Almeida, H. Sirelkhatim, G. Richard, M. Skwark, K. Beguir, M. Lopez, and T. Pierrot, "The nucleotide transformer: Building and evaluating robust foundation models for human genomics," *bioRxiv*, 2023.

[23] E. Nguyen, M. Poli, M. Faizi, A. Thomas, C. Birch-Sykes, M. Wornow, A. Patel, C. Rabideau, S. Massaroli, Y. Bengio, *et al.*, "Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution," *arXiv preprint arXiv:2306.15794*, 2023.

[24] G. Benegas, S. S. Batra, and Y. S. Song, "Dna language models are powerful predictors of genome-wide variant effects," *Proceedings of the National Academy of Sciences*, vol. 120, no. 44, p. e2311219120, 2023.

[25] V. Fishman, Y. Kuratov, M. Petrov, A. Shmelev, D. Shepelin, N. Chekanov, O. Kardymon, and M. Burtsev, "Gena-lm: A family of open-source foundational models for long dna sequences," *bioRxiv*, pp. 2023–06, 2023.

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

[27] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[28] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," 2022.

[29] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations*, 2018.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[31] A. E. Trevino, F. Müller, J. Andersen, L. Sundaram, A. Kathiria, A. Shcherbina, K. Farh, H. Y. Chang, A. M. Pașca, A. Kundaje, *et al.*, "Chromatin and gene-regulatory dynamics of the developing human cerebral cortex at single-cell resolution," *Cell*, vol. 184, no. 19, pp. 5053–5069, 2021.

[32] S. Nair, M. Ameen, L. Sundaram, A. Pampari, J. Schreiber, A. Balsubramani, Y. X. Wang, D. Burns, H. M. Blau, I. Karakikes, K. C. Wang, and A. Kundaje, "Transcription factor stoichiometry, motif affinity and syntax regulate single-cell chromatin dynamics during fibroblast reprogramming to pluripotency," *bioRxiv*, 2023.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[36] K. Jaganathan, S. K. Panagiotopoulou, J. F. McRae, S. F. Darbandi, D. Knowles, Y. I. Li, J. A. Kosmicki, J. Arbelaez, W. Cui, G. B. Schwartz, *et al.*, "Predicting splicing from primary sequence with deep learning," *Cell*, vol. 176, no. 3, pp. 535–548, 2019.

[37] T. Zeng and Y. I. Li, "Predicting rna splicing from dna sequence using pangolin," *Genome biology*, vol. 23, no. 1, pp. 1–18, 2022.

[38] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, I. Barnes, A. Bignell, V. Boychenko, T. Hunt, M. Kay, G. Mukherjee, J. Rajan, G. Despacio-Reyes, G. Saunders, C. Steward, R. Harte, M. Lin, C. Howald, A. Tanzer, T. Derrien, J. Chrast, N. Walters, S. Balasubramanian, B. Pei, M. Tress, J. M. Rodriguez, I. Ezkurdia, J. van Baren, M. Brent, D. Haussler, M. Kellis, A. Valencia, A. Reymond, M. Gerstein, R. Guigó, and T. J. Hubbard, "Gencode: The reference human genome annotation for the encode project," *Genome Research*, vol. 22, pp. 1760–1774, 2012.

[39] The ENCODE Project Consortium, "An integrated encyclopedia of dna elements in the human genome," *Nature*, vol. 489, no. 7414, pp. 57–74, 2012.

[40] The ENCODE Project Consortium, "Expanded encyclopaedias of dna elements in the human and mouse genomes," *Nature*, vol. 583, no. 7818, p. 699–710, 2020.

[41] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *arXiv preprint arXiv:2104.09864*, 2021.

[42] B. Peng, J. Quesnelle, H. Fan, and E. Shippole, "Yarn: Efficient context window extension of large language models," *arXiv preprint arXiv:2309.00071*, 2023.

[43] E. Trop, C.-H. Kao, M. Polen, Y. Schiff, B. P. de Almeida, A. Gokaslan, T. Pierrot, and V. Kuleshov, "Advancing dna language models: The genomics long-range benchmark," *LLMs4Bio AAAI Workshop 2024*, 2024.

[44] Y. Schiff, C.-H. Kao, A. Gokaslan, T. Dao, A. Gu, and V. Kuleshov, "Caduceus: Bi-directional equivariant long-range dna sequence modeling," 2024.

[45] S. Braun, M. Enculescu, S. T. Setty, M. Cortés-López, B. P. de Almeida, F. R. Sutandy, L. Schulz, A. Busch, M. Seiler, S. Ebersberger, *et al.*, "Decoding a cancer-relevant splicing decision in the ron proto-oncogene using high-throughput mutagenesis," *Nature communications*, vol. 9, no. 1, p. 3315, 2018.

[46] F. J. Martin, M. R. Amode, A. Aneja, O. Austine-Orimoloye, A. G. Azov, I. Barnes, A. Becker, R. Bennett, A. Berry, J. Bhai, *et al.*, "Ensembl 2023," *Nucleic acids research*, vol. 51, no. D1, pp. D933–D941, 2023.

[47] F. I. Marin, F. Teufel, M. Horrender, D. Madsen, D. Pultz, O. Winther, and W. Boomsma, "Bend: Benchmarking dna language models on biologically meaningful tasks," *arXiv preprint arXiv:2311.12570*, 2023.

[48] E. Nguyen, M. Poli, M. G. Durrant, A. W. Thomas, B. Kang, J. Sullivan, M. Y. Ng, A. Lewis, A. Patel, A. Lou, S. Ermon, S. A. Baccus, T. Hernandez-Boussard, C. Ré, P. D. Hsu, and B. L. Hie, "Sequence modeling and design from molecular to genome scale with evo," *bioRxiv*, 2024.

[49] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[50] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, *et al.*, "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17283–17297, 2020.

[51] J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, N. Zheng, and F. Wei, "Longnet: Scaling transformers to 1,000,000,000 tokens," 2023.

[52] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "Realm: Retrieval-augmented language model pre-training," 2020.

[53] J. Janssens, S. Aibar, I. I. Taskiran, J. N. Ismail, A. E. Gomez, G. Aughey, K. I. Spanier, F. V. D. Rop, C. B. González-Blas, M. Dionne, K. Grimes, X. J. Quan, D. Papasokrati, G. Hulselmans, S. Makhzami, M. D. Waegeneer, V. Christiaens, T. Southall, and S. Aerts, "Decoding gene regulation in the fly brain," *Nature*, vol. 601, no. 7894, pp. 630–636, 2022.

[54] Z. Tang and P. K. Koo, "Evaluating the representational power of pre-trained dna language models for regulatory genomics," *bioRxiv*, 2024.

[55] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[56] S. Chen, S. Wong, L. Chen, and Y. Tian, "Extending context window of large language models via positional interpolation," *arXiv preprint arXiv:2306.15595*, 2023.

[57] kaiokendev, "Things I'm learning while training superhot." `https://kaiokendev.github.io/til#extending-context-to-8k`, 2023.

[58] "NTK-Aware Scaled RoPE." `https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/`, 2023.

[59] D. Kim, J. M. Paggi, C. Park, C. Bennett, and S. L. Salzberg, "Graph-based genome alignment and genotyping with hisat2 and hisat-genotype," *Nature biotechnology*, vol. 37, pp. 907–905, 2019.

[60] W. Meuleman, A. Muratov, E. Rynes, J. Halow, K. Lee, D. Bates, M. Diegel, D. Dunn, F. Neri, A. Teodosiadis, A. Reynolds, E. Haugen, J. Nelson, A. Johnson, M. Frerker, M. Buckley, R. Sandstrom, J. Vierstra, R. Kaul, and J. Stamatoyannopoulos, "Index and biological spectrum of human dnase i hypersensitive sites," *Nature*, vol. 584, pp. 244–251, 2020.

Methods

A Genome segmentation model

In this section, we introduce our approach to segment the genome, namely SegmentNT. We formulate this problem as the segmentation of a sequence of N nucleotides (for example $N = 3,000$ bp, 3kb, or $N = 10,000$ bp, 10kb) by predicting a probability for each nucleotide to be part of one of $K = 14$ elements: *protein-coding gene*, *lncRNA*, *5'UTR*, *3'UTR*, *exon*, *intron*, *splice donor site*, *splice acceptor site*, *polyA signal*, *promoter tissue-invariant*, *promoter tissue-specific*, *enhancer tissue-invariant*, *enhancer tissue-specific* or *CTCF-bound*.

A.1 SegmentNT architecture

Nucleotide Transformer (NT) can be used as a backbone for segmenting a sequence of nucleotides. SegmentNT uses the pre-trained *NT-Multispecies-v2 (500M)* model as DNA encoder to extract embeddings for each of the tokens yielded by a 6-mer tokenizer. We note N the number of nucleotides in the DNA sequence and L the number of DNA tokens (with roughly $L \approx N/6$). In order to segment the sequence, we replace its original language model head by a 1-dimensional *U-Net* segmentation head [33] made of 2 downsampling convolutional blocks and 2 upsampling convolutional blocks. Each of these blocks is made of 2 convolutional layers with $2,048$ and $4,096$ kernels respectively, and $L/2$ and $L/4$ sequence length. This accounts for 63 million parameters. The output of this layer is a $N \times K \times 2$ dimensional vector which gives K probabilities for each nucleotide corresponding to the probability that the nucleotide is part of each type of genomics element. We do not add further constraints on predictions such as the fact that one nucleotide belongs only to one element, and thus each nucleotide can be part of multiple elements.

A.2 Model training and evaluation

We train our model using Adam optimizer with $lr = 5e-5$. We use a batch size of 256 and trained the SegmentNT-3kb model for 10.24B tokens, meaning a total of 20.48M sequences seen during training. The training was done on a cluster of 8 GPU H100 over 20 hours. The 10kb, 20kb and 30kb models were initialized from the best checkpoint of the respective smaller model for faster adaptation to longer lengths. For example, SegmentNT-30kb model was initialized with the best SegmentNT-20kb checkpoint and finetuned for an additional 2.56B tokens (0.51M sequences). We use focal loss [34] with $\gamma = 2$ which helps the model to focus on "harder" samples, *ie* the sparse nucleotides that belong to an element.

We split our dataset between train, validation and test sets by chromosome. Namely, chromosomes 20 and 21 are used for test, chromosome 22 is used for validation and the remaining are used for training. During training, sequences are randomly sampled in the genome with associated annotations. We keep the sequences in the validation and test sets fixed by using a sliding window of length N over the respective chromosomes. The validation set was used to monitor training and for early stopping while the test set was used to evaluate model performance. We used Matthews correlation coefficient (MCC) as a validation metric and selected the best checkpoint based on the average score across all 14 genomics elements. During evaluation and testing, we predict for each sequence K probabilities per nucleotide, concatenate all predictions across all sequences into a single array per element predicted, and compute MCC and Precision-Recall Area Under the Curve (PR-AUC) for each genomics element over every nucleotide.

A.3 Comparison of different architectures

SegmentNT is made of a DNA encoder, Nucleotide Transformer, and a 1-dimensional U-Net segmentation head, as described above. To evaluate the added value of using a pre-trained backbone encoder, we compared it on the 3kb sequences with (1) two versions of the U-Net segmentation head alone, with 63M and 252M parameters respectively, which take one-hot encoded DNA sequences as input instead of the embeddings outputted by the DNA encoder; and (2) a SegmentNT model whose encoder is initialized with random weights. Since when using one-hot encoded input sequences there is no aggregation of the base pairs into 6-mers, the input to the first convolutional layers of the U-Net model has a length of $L = 3,000$ one-hot encoded base pairs instead of 500 token embeddings. As with the SegmentNT models, we monitor the training by validating on sequences from chromosome 22 and selected the best checkpoint based on the highest average MCC score across the 14 types of elements. For the randomly initialized SegmentNT model, we stopped training before this criteria was met because the training took significantly longer time and the performance on most of the genomics elements had plateaued. The 63M and 252M U-Net models converged after 14.3M and 18.4M sequences respectively, just before the SegmentNT-3kb model at 20.4M sequences. However, to reach this point, they take 12 hours and 36 hours respectively against 20 hours for SegmentNT-3kb.

A.4 Context Length Extension

Since the DNA encoder of SegmentNT is using rotary positional embeddings (RoPE) that have been trained on a maximum sequence length of $2,048$ tokens, its performance degrades very quickly when inferring on longer sequences. Several previous works have suggested adaptations to RoPE to better handle evaluation or fine-tuning on longer sequences, such as using Position Interpolation ([56, 57]) or ”NTK-aware” scaled Rope [58]. More recently, [42] formalized different methods and augmented them to propose a final adaptation of RoPE to unseen lengths called YaRN. After testing the different approaches, YaRN did not introduce improvements to extending Segment-NT lengths compared to simply using ”NTK-aware” RoPE. Since the latter is lighter to implement we decided to use it for extending the context of SegmentNT.

As described by Pend et al. [42], with the hidden layer set of hidden neurons denoted by D, and a sequence of vectors $x_1, \ldots x_L \in R^{|D|}$, ”NTK-aware” RoPE can be described by the following equation:

$$f'_w(\mathbf{x}_m, m, \theta_d) = f_{\mathbf{w}}(\mathbf{x}_m, g(m), h(\theta_d))$$

where d is the position along the embedding dimension, m is the position of the embedding in the sequence, f is the RoPE function (detailed in Eq.1 of [56]), $g(m) = m$, $h(\theta_d) = b'^{-2d/|D|}$, $b' = b.s^{\frac{|D|}{|D|-2}}$ and finally $\frac{2\pi}{\theta_d} = 2\pi b^{\frac{2d}{|D|}}$. The rescaling factor s is computed as $s = \frac{L'}{L}$ with L' the extended context length and L the training context length, which for the *NT-Multispecies-v2 (500M)* is 2048 tokens.

For SegmentNT models trained with ”NTK-aware” RoPE, all sequences with length inferior to their training length are evaluated with the same rescaling factor that was used during the training. Concretely, SegmentNT-30kb is trained with $s = 2.44$, and therefore inference on a sequence smaller than $30,000$bp is done with $s = 2.44$. When evaluated on a 50kb sequence, the rescaling factor becomes $s = 4.07$.

A.5 Multi-species training

We trained an additional, multispecies model (SegmentNT-10kb-multispecies) by finetuning the human SegmentNT-10kb model on the annotations of five species together (mouse, chicken, fly, zebrafish and worm). We used the same model hyperparameters and training parameters. Since the different species have different genome sizes, we balanced examples from each dataset with the following weights: 5 for human, 4 for mouse, 2 for chicken, fly and zebrafish, and 1 for worm. Similar to the human dataset, we split the chromosomes of each species intro training, validation and test set.

B Genome annotation data

B.1 Human genomics elements

The human segmentation dataset of genomics elements was created from 14 types of elements, divided in gene elements (protein-coding genes, lncRNAs, 5’UTR, 3’UTR, exon, intron, splice acceptor and donor sites) and regulatory elements (polyA signal, tissue-invariant and tissue-specific promoters and enhancers, and CTCF-bound sites). The final segmentation dataset was created by overlapping all 14 elements with every DNA sequence of length N nucleotides. Sequences with Ns were removed.

The location of all gene elements and polyA signals were obtained from GENCODE [38] V44 gene annotation. Annotations were filtered to exclude level 3 transcripts (automated annotation), so all training data was annotated by a human. We used *extract_splice_sites.py* from HISAT2 [59] (`https://github.com/DaehwanKimLab/hisat2/blob/master/hisat2_extract_splice_sites.py`) to extract respective intron and splice site annotations.

Promoter, enhancer and CTCF-bound sites were retrieved from ENCODE's SCREEN database (`https://screen.wenglab.org/`) [40]. Distal and proximal enhancers were combined. Promoters and enhancers were split in tissue-invariant and tissue-specific based on the vocabulary from Wouter Meuleman et al. [60] `https://www.meuleman.org/research/dhsindex/`. Enhancers or promoters overlapping regions classified as tissue-invariant were defined as that, while all other enhancers and promoters were defined as tissue-specific.

B.2 Multi-species dataset

To create segmentation datasets for additional species we focused only on the main gene elements: protein-coding genes, 5'UTR, 3'UTR, exon, intron, splice acceptor and donor sites. We obtained their annotations as described for the human dataset but retrieved from Ensembl databases (`https://www.ensembl.org`). We considered 5 species to train the multispecies model: mouse ($mm10$), chicken ($galGal6$), fly ($dm6$), zebrafish ($danRer11$) and worm ($ce11$). We created a held-out test set made of 12 species: gorilla ($gorGor4$), macaque ($Mnem\_1$), rat ($mRatBN7$), beaver (can_genome_v1), chinchilla ($ChiLan1$), whale ($ASM228892v3$), cat ($Felis\_catus\_9$), canary ($SCA1$), tetradon ($TETRAODON8$), anemonefish ($AmpOce1$), trout ($fSalTru1$) and Ciona intestinalis (KH). Evolutionary distance data was retrieved from Timetree of Life.

C Benchmarking for regulatory elements

C.1 Sliding Nucleotide Transformer finetuned models

We first compared SegmentNT-10kb with a sliding window approach, where a binary classifier is used to predict the output probability for multiple sliding windows of the input 10kb DNA sequence. We applied this approach for the segmentation of the two best classes of regulatory elements: promoter tissue-invariant and enhancer tissue-specific. As binary classifier we used the NT finetuned models on promoter and enhancer, respectively [22]. Sliding windows were created using a step size of 10 and the input size of the respective promoter (300nt) and enhancer (200nt) models. All inference times were calculated in a single A100 GPU.

C.2 Comparison with Enformer zero-shot predictions

We also compared SegmentNT with Enformer [16] for promoter predictions. Here we ported Enformer to our Jax codebase for a more direct comparison (previously done in [22]). For each 10kb input sequence, we padded the sequences as requested by the model input dimensions and computed all Enformer predictions at the original 128bp bin resolution and used the average over 7 selected ATAC-seq profiles for different human cell lines as quantitative score of regulatory activity. We report the PR-AUC metric for the predictive value of this quantitative score to identify promoters at nucleotide resolution. All inference times were calculated in a single A100 GPU.

D Splicing tasks

D.1 Comparison with SpliceAI

We compared SegmentNT with SpliceAI [36] on both SpliceAI's test set and SegmentNT's test set given their different settings. We used the scripts available at the Illumina Basespace platform [1] to reproduce the testing dataset presented in SpliceAI for both 10kb and 30kb input sequences without additional context. This test set contains only mRNA sequences and all in the forward strand (i.e. for genes in the reverse strand, the sequence is reversed to have the gene in the forward orientation). We also compared both models on the SegmentNT's 10kb and 30kb test sets, which contain all windows of the test chromosomes, including windows without genes or with genes in both the forward and reverse strand. We used as performance metrics both PR-AUC and MCC.

D.2 Sequence variants and transcript isoforms

We used data from an experimental saturation mutagenesis splicing assay of the exon 11 of the gene *MST1R*, flanked by constitutive exons 10 and 12 and respective introns (data from Braun, Simon, et al. [45]; see Methods). This dataset contains a library of almost $5,800$ randomly mutated minigenes of $\sim$700nt, where for each minigene variant it was evaluated the splicing of the alternative exon 11 in the respective mRNA molecules. We used this data to test if SegmentNT could predict the impact of those sequence variants on the respective splicing and transcript isoforms. We focused only on minigene variants composed of combinations of single-nucleotide mutations. We predicted all 14 genomics elements in the wildtype minigene sequence and all minigene variants. For a systematic comparison, we compared the predicted exon score for the region of the alternative exon 11 with the experimentally measured exon inclusion scores.

Data availability

The SegmentNT training data was obtained from publicly available resources. Gene annotations were obtained from GENCODE (`https://www.gencodegenes.org/`) and Ensembl databases (`https://www.ensembl.org`). Human regulatory elements were obtained from ENCODE's SCREEN database (`https://screen.wenglab.org/`). Evolutionary distance data was retrieved from Timetree of Life.

Code availability

Model weights of the human and multispecies SegmentNT models as well as inference code in Jax are available for research purposes at `https://github.com/instadeepai/nucleotide-transformer`. HuggingFace versions of the models, in PyTorch, can be found at `https://huggingface.co/InstaDeepAI`. Example notebooks are available on Google Colab at `https://colab.research.google.com/github/instadeepai/nucleotide-transformer/blob/main/examples/inference_segment_nt.ipynb`.

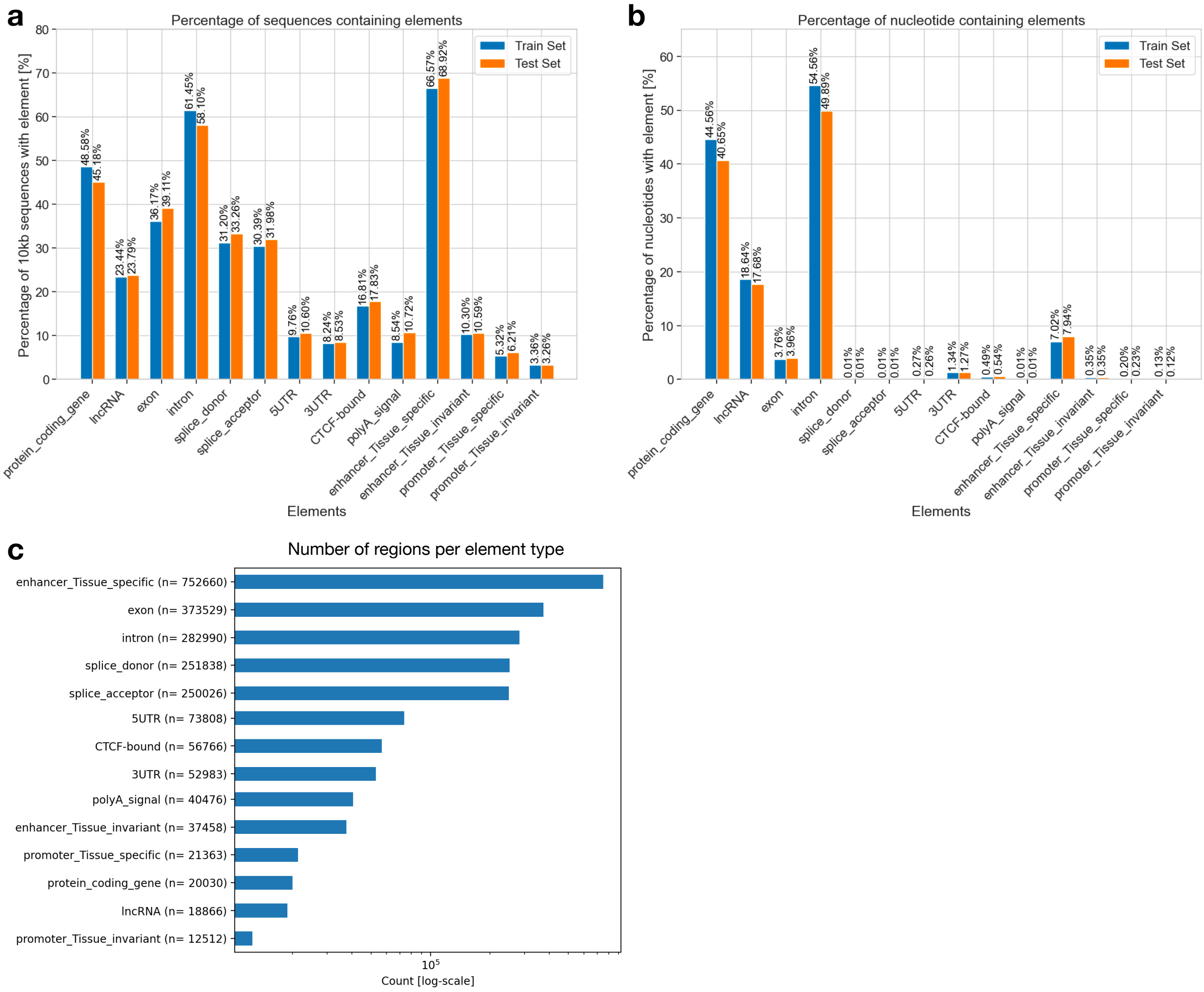
Acknowledgments

We thank Maxence Gélard for his assistance in identifying and running some baseline experiments. We also thank Aliou Kayantao for his help improving the figure panels. We would also like to thank Volodymyr Kuleshov and Yang Li for insightful discussions about context-length extension and important applications for DNA foundation models. Finally, we would like to thank Lida Rosselo for help on the project management side of this research project.

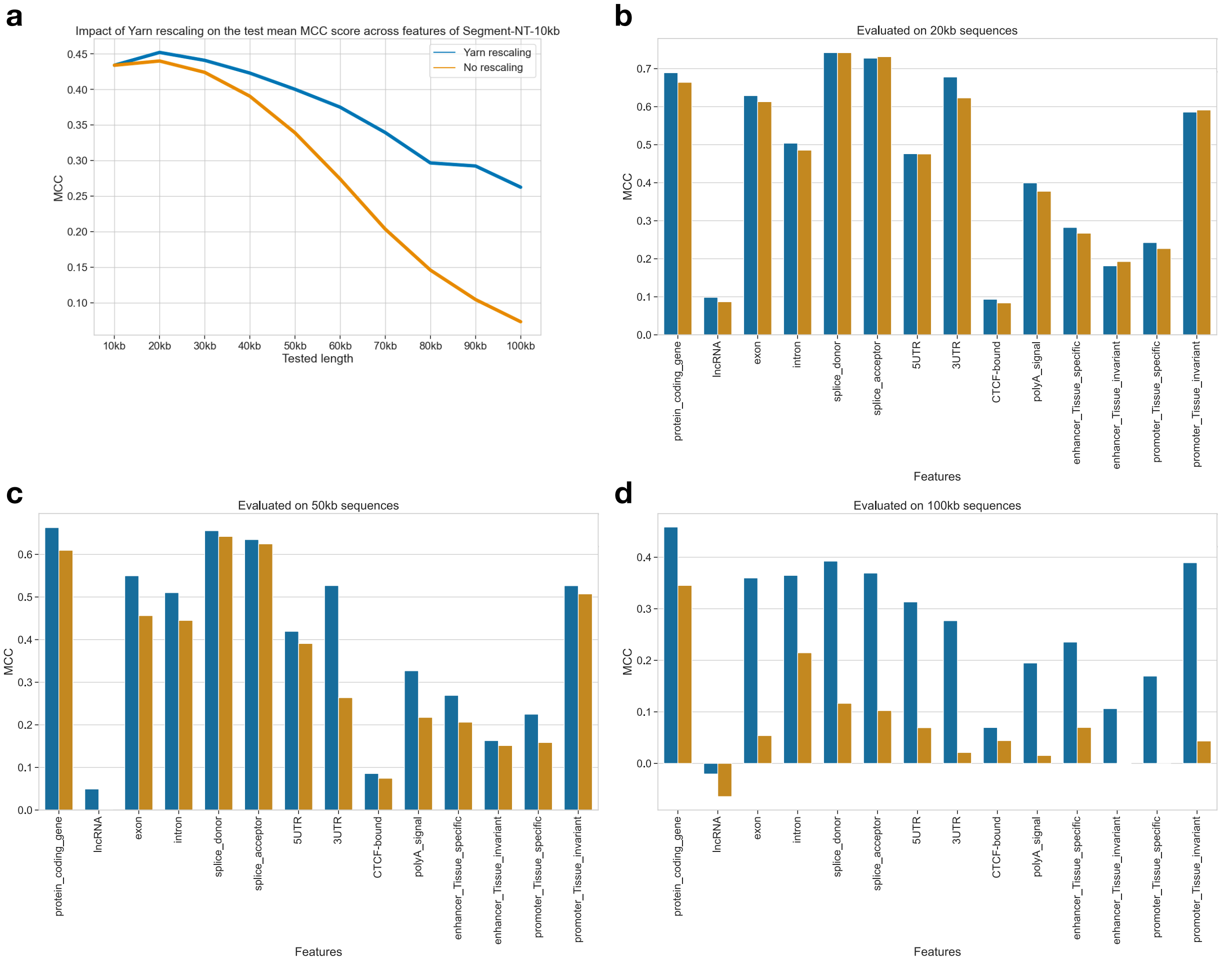
---

[1] `https://basespace.illumina.com/projects/66029966/`

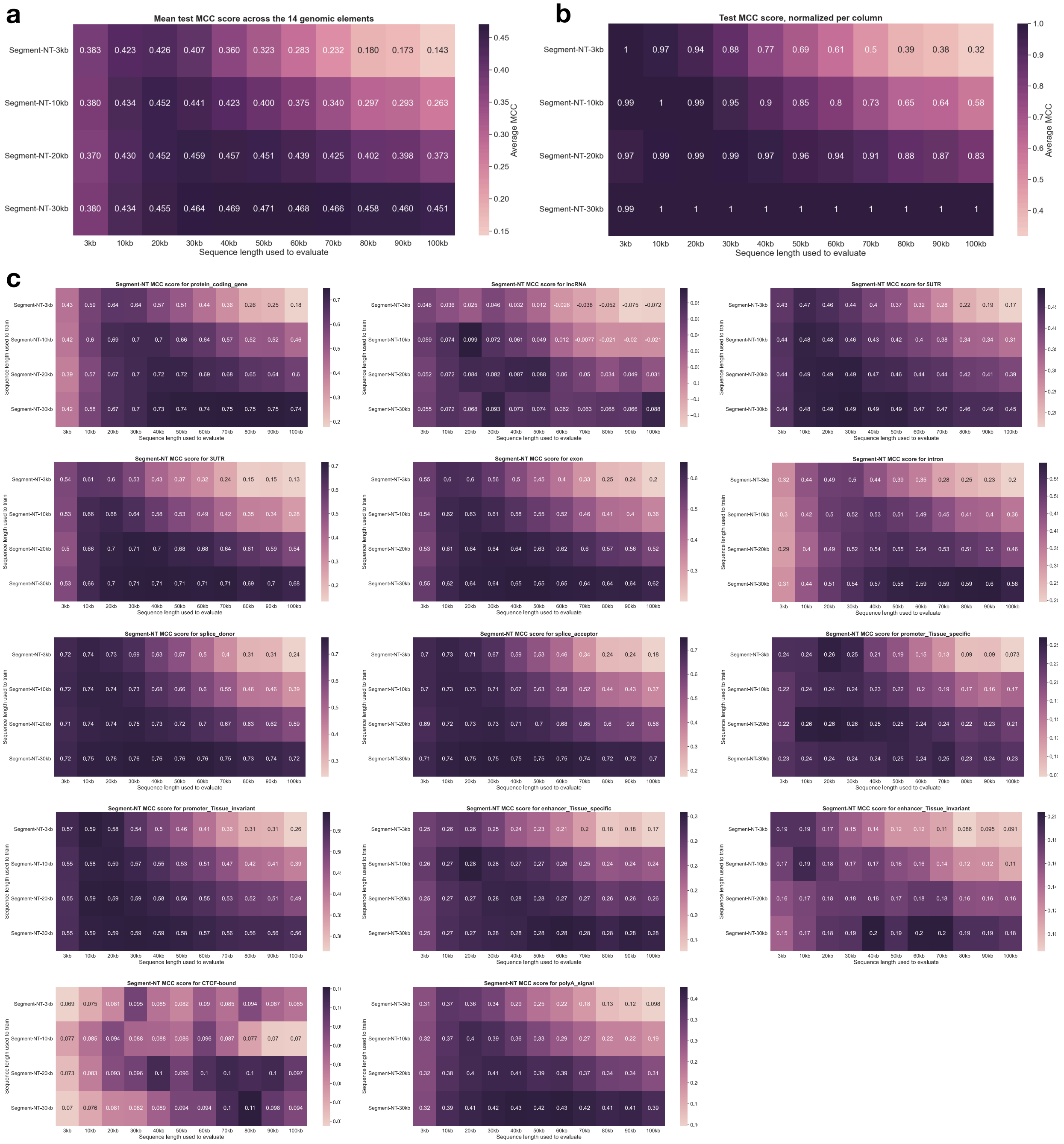
Supplementary Figures



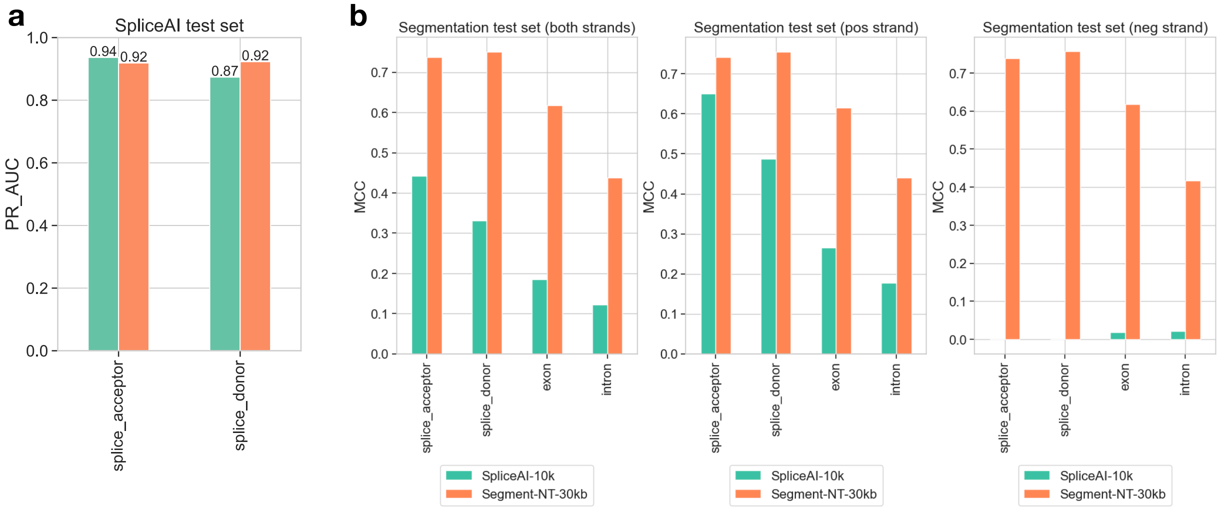
Supplementary Figure 1: **Data distribution per element type.** **a**) Percentage of sequences containing each element type in train and test 10kb dataset. **b**) Percentage of nucleotides containing each element type in train and test 10kb dataset. **c**) Number of regions per element type in the human annotation.



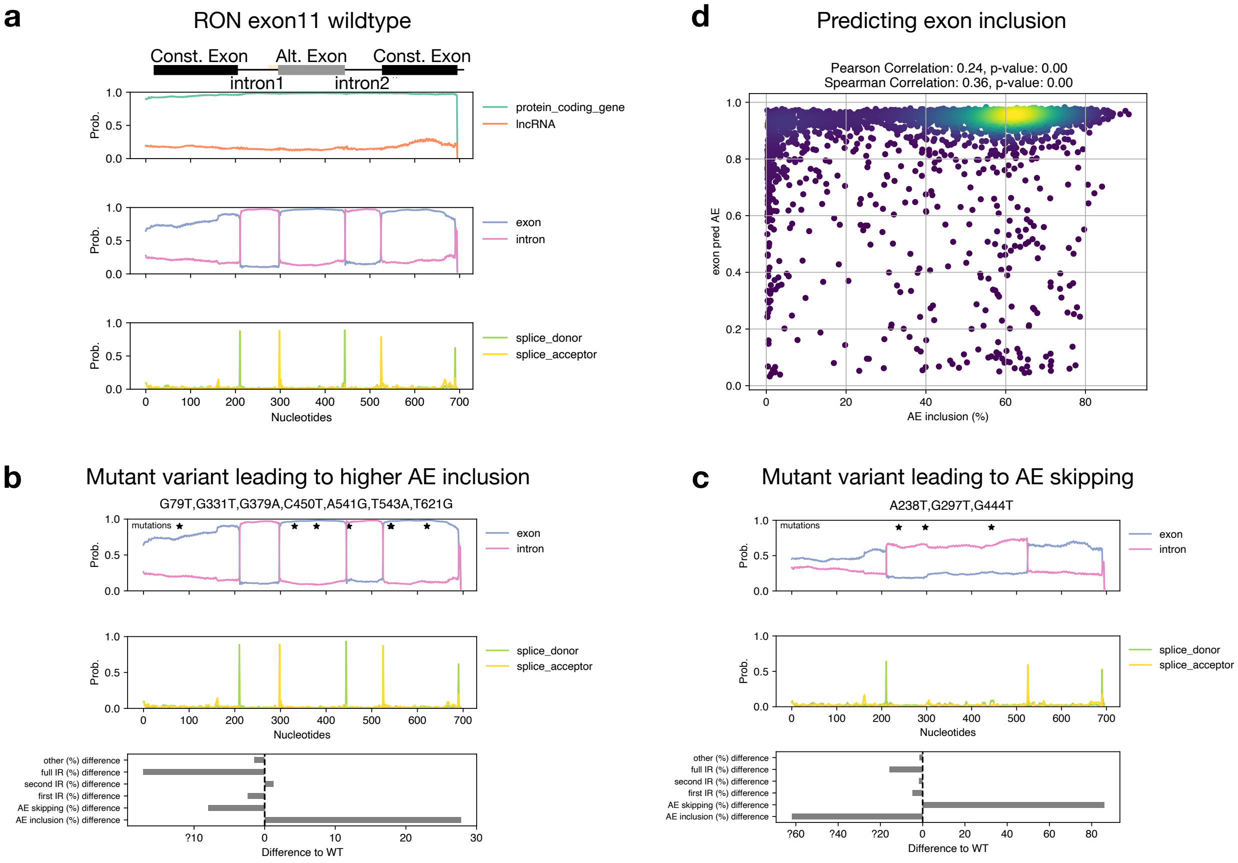
Supplementary Figure 2: **Context-length extension allows to rescale SegmentNT-10kb to 100kb sequences. a)** Average MCC performance across the 14 elements for the SegmentNT-10kb model with and without context-length rescaling. **b-d)** Performance on different input lengths without vs with context-length extension.



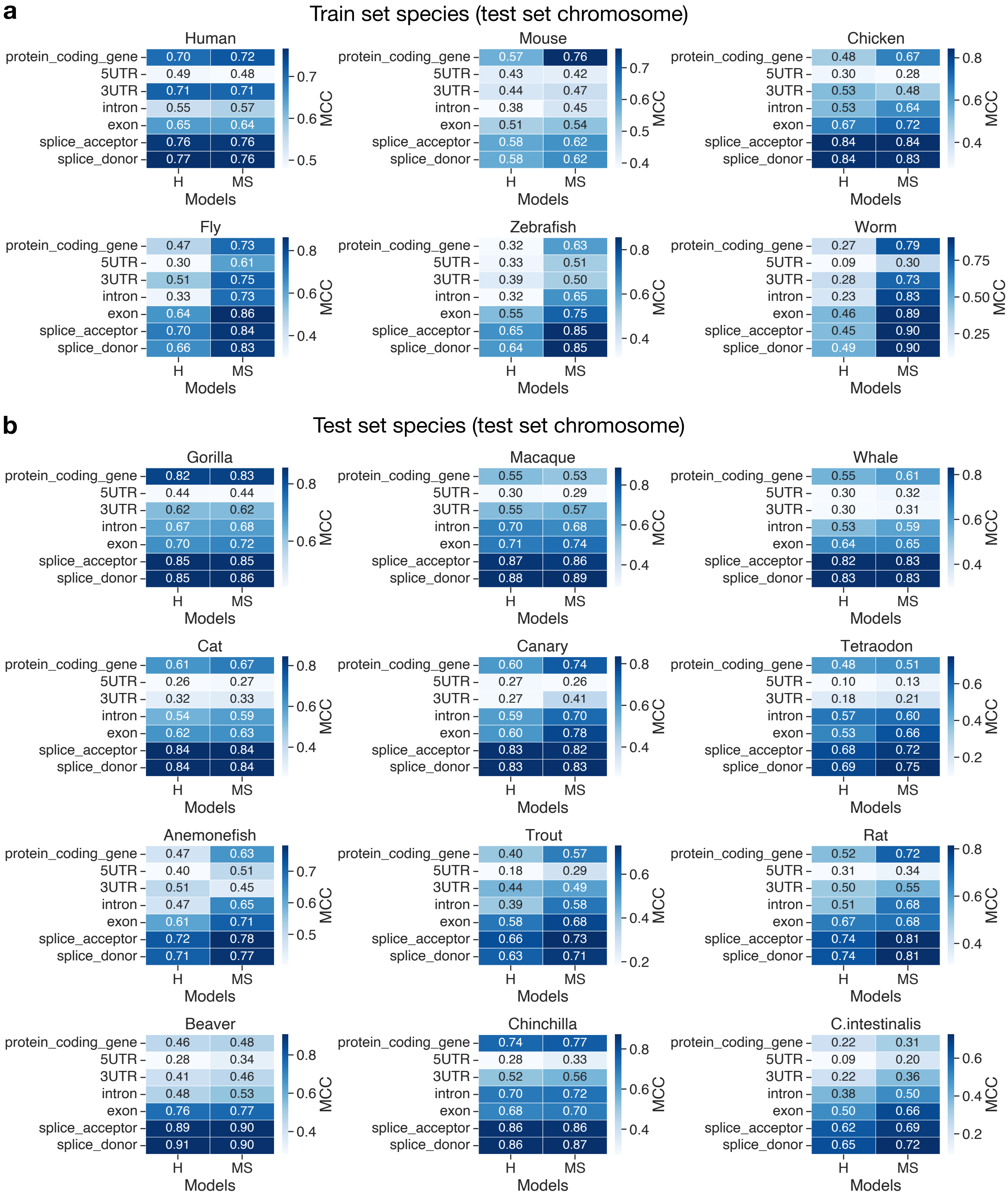
Supplementary Figure 3: **Evaluating Segment-NT models on different lengths**. **a)** Long-range models improve performance on longer contexts while maintaining performance on shorter contexts. Average MCC performance across the 14 elements for the different SegmentNT models per input sequence length. **b)** Same as in **(a)** but coloured by the difference to the best model per input sequence length. **c)** Heatmap with MCC performance per genomics element.



Supplementary Figure 4: **Comparison with SpliceAI on test set 10kb sequences.** **b)** Performance of SpliceAI and SegmentNT-30kb for splice acceptor and donor detection on SpliceAI's gene-centric test set of 10kb sequences. We used as metric the PR-AUC. **c)** Performance of SpliceAI and SegmentNT-30kb for splice acceptor and donor detection as well as exon and intron prediction on SegmentNT's whole chromosome test set of 10kb sequences. We used as metric the MCC and report performance for all regions (left), or regions containing genes only in the positive (middle) or negative strand (right).



Supplementary Figure 5: **Prediction of sequence variants affecting splicing isoforms.** **a)** SegmentNT prediction of gene elements of *RON* exon11 minigene wildtype sequence. **b-c)** SegmentNT prediction for a minigene variant leading to **(b)** higher exon inclusion and **(c)** higher exon skipping. Predictions for exon/intron (top) and splice sites (middle) are shown. Nucleotide mutations in each minigene variant are shown with black stars. Bottom bar-plot shows the experimental differences in isoform abundance relative to the wildtype minigene sequence. **d)** Scatter-plot comparing SegmentNT's exon prediction and the inclusion of the alternative exon 11 (AE) across all minigene variants. Pearson and Spearman correlation coefficients and p-values are shown.



Supplementary Figure 6: **Comparison of human and multispecies SegmentNT models. a-b)** Performance of the human (H) and multispecies (MS) model per element for **(a)** training set species and **(b)** test set species.