EnhancerDetector: Enhancer Discovery from Human to Fly via Interpretable Deep Learning

Luis M. Solis[1], Geyenna Sterling-Lentsch[2], Marc S. Halfon[2], Hani Z. Girgis[1*]

[1]The Bioinformatics Toolsmith Laboratory, Department of Electrical Engineering and Computer Science, Texas A&M University-Kingsville, Kingsville, TX, 78363, USA.
[2]Department of Biochemistry, University at Buffalo, State University of New York, Buffalo, NY, 14203, USA.

*Corresponding author(s). E-mail(s): hani.girgis@tamuk.edu;

Abstract

Deciphering how enhancers encode regulatory information in DNA remains a central genomics challenge, as sequencing outpaces functional annotation. A key question is whether enhancers possess an intrinsic, sequence-based "enhancerness" distinguishing them from other regions, independent of species, cell type, or assay. Confirming its existence and learnability is both biologically fundamental and essential for scalable genome annotation. We introduce *EnhancerDetector*, a convolutional neural network-based framework for cross-species enhancer prediction that combines high accuracy with biological interpretability. Trained on human data, *EnhancerDetector* achieves strong performance across human, mouse, and fly datasets, consistently outperforming existing methods in precision and F1. It generalizes to datasets generated using diverse experimental assays. Unlike chromatin feature-based predictors requiring complex post hoc thresholding, EnhancerDetector directly outputs enhancer probability scores from short sequence windows, simplifying enhancer discovery workflows. An ensemble strategy further improves prediction reliability by reducing false positives. *EnhancerDetector* supports fine-tuning on new species and retains strong performance even when adapted with as few as 20,000 enhancer sequences, making it ideal for newly sequenced genomes with limited experimental data. For interpretability and visualization, we apply class activation maps to identify sequence regions predictive of enhancer activity. Experimental validation in transgenic flies confirms the predictive power of *EnhancerDetector*: five of six tested candidates drove reporter expression, and four exhibited expression patterns supported by

prior literature. These analyses highlight distinct sequence and contextual features that confer what we term "enhancerness:" enhancer sequences possess a characteristic, identifiable signature.

Keywords: enhancer prediction, deep learning, convolutional neural network, interpretability, cross-species genomics

1 Introduction

Enhancers are regulatory elements that play a critical role in the transcriptional activation of genes. These non-coding DNA sequences can be located upstream, downstream, or even within the target gene itself, and their ability to function across long genomic distances is facilitated by the three-dimensional architecture of a genome [1, 2]. Enhancers play a crucial role in controlling gene activity that underlies lineage commitment and organismal formation [3]. Moreover, enhancer dysfunction is increasingly recognized as a contributor to various medical issues, collectively termed "enhanceropathies" [1]. Thus, accurate identification of enhancers is essential for understanding gene regulation in both physiology and pathology.

Despite their biological significance, identifying enhancers remains a considerable challenge. Enhancers exhibit high locational variability and are often found within extensive regions of non-coding DNA, making them difficult to detect [2]. This challenge is compounded by the cell-type-specific nature of enhancer activity: an enhancer active in one cellular context may be inactive in another [2]. Experimental methods—reviewed by Suryamohan and Halfon [4]—such as ChIP-seq, DNase-seq, ATAC-seq, FAIRE-seq, Hi-C, classical reporter assays, high-throughput reporter screens, CRISPR-based perturbation techniques, and eRNA-detection assays have advanced enhancer discovery [5–19]. However, each captures only a fraction of the enhancer landscape, requires substantial experimental resources, and is typically applied to a narrow set of species, tissues, and developmental stages. As a result, even in well-studied organisms, enhancer annotations remain incomplete.

The gap is even more striking in the rapidly expanding collection of newly sequenced genomes. Large-scale initiatives such as the Earth BioGenome Project [20], the Vertebrate Genomes Project [21], the European Reference Genome Atlas [22], and global marine surveys such as Tara Oceans [23] are generating reference genomes and environmental sequence resources far faster than functional genomic data can be annotated. Because the vast majority of these species lack ATAC-seq, ChIP-seq, or reporter assay data, their regulatory landscapes—especially enhancers—are effectively unannotated. Thus there is a critical need for scalable approaches that can identify putative enhancers from DNA sequence alone, enabling first-pass functional annotation across thousands of species.

A key question, therefore, is whether enhancers possess generalizable sequence-level properties that can be detected independently of cell type or species [24]. Biologically, enhancers are enriched for dense clusters of transcription factor binding sites

(TFBSs) [25, 26]. These binding sites are often degenerate and exhibit substantial similarity across transcription factor families—for example, the CANNTG core for bHLH factors, TAAT for homeobox proteins, and GGA/T for ETS proteins [27]. Although the precise combinations and arrangements of motifs that drive enhancer activity are cell-type specific, the enrichment of such degenerate motif clusters is a shared and fundamental property of enhancers [25]. These observations support the hypothesis that enhancers are defined by intrinsic, recurring sequence characteristics—suggestive of a general enhancer-defining signature—that distinguish them from other genomic regions, even when their regulatory outputs are context dependent [28, 29].

We therefore hypothesize that enhancers possess an intrinsic and conserved sequence signature—here termed "enhancerness"—that distinguishes enhancer DNA from non-enhancer DNA. This signature arises from the recurrent presence and density of related transcription factor motif families, their short-range organization, and supportive broader sequence context [30]. Under this hypothesis, a model trained on a diverse collection of enhancers from multiple cell types should be able to learn enhancer identity even when the specific transcriptional codes governing activity differ.

Existing computational methods have provided important advances in enhancer prediction but also exhibit several limitations. Early approaches relied on traditional machine learning models trained on manually designed sequence features (e.g., k-mer frequencies, GC content, or motif scores), exemplified by tools such as CrmMiner [31], iEnhancer-2L [32], EnhancerPred [33], iEnhancer-EL [34], and SCRMshaw [35]. More recent methods have adopted deep learning strategies, including convolutional, recurrent, generative, and attention-based architectures trained directly on DNA sequence [36–39]. In parallel, widely used models such as gkm-SVM and LS-GKM [40, 41], DeepSEA [42], deltaSVM [43], and Enformer [44] have demonstrated strong performance and broad utility in regulatory genomics; however, these approaches are typically optimized for predicting activity in specific cellular contexts rather than for identifying generalizable enhancer identity.

Despite these advances, many existing approaches rely on chromatin marks or handcrafted features, limiting their applicability to species and tissues with rich functional genomic datasets, while others require post hoc thresholding or show limited generalization across species, assays, or experimental contexts. Sequence-only deep learning methods offer a promising alternative, but many were not explicitly designed to capture generalizable enhancer identity or to provide interpretable insights into the sequence features underlying enhancer function [45].

To address these challenges and to directly test whether enhancers share a generalizable, sequence-encoded identity, we developed *EnhancerDetector*, a convolutional neural network-based framework for identifying enhancers from DNA sequence alone. *EnhancerDetector* is trained on curated enhancer datasets from human and is designed to generalize across species, datasets, and experimental assays. By learning sequence-level properties that distinguish enhancers from non-enhancers, the model provides a scalable solution for predicting putative enhancers in genomes lacking functional genomic data. Moreover, by incorporating class activation maps, *EnhancerDetector*

yields interpretable insights into the sequence features associated with enhancer identity. Experimental validation in transgenic flies demonstrates that these predictions correspond to biologically active enhancers.

In this study, we show that enhancers possess a detectable sequence signature and that this signature can be learned in a manner that generalizes across species—a property with significant implications for genome annotation. Our results provide evidence for the existence of enhancerness as a generalizable feature of enhancer sequences, and they introduce a practical framework for enhancer discovery at a genomic scale.

2 Results and Discussions

2.1 Main Contributions

This study presents several key contributions that advance the state of enhancer identification:

- We introduce *EnhancerDetector*, a deep learning framework for accurate and interpretable enhancer prediction across species. The code is freely available for academic use through github.com/BioinformaticsToolsmith/EnhancerDetector and doi.org/10.5281/zenodo.15531293.
- The model generalizes well to enhancers identified using diverse experimental assays, including CAGE, snATAC-seq, and DNase-seq.

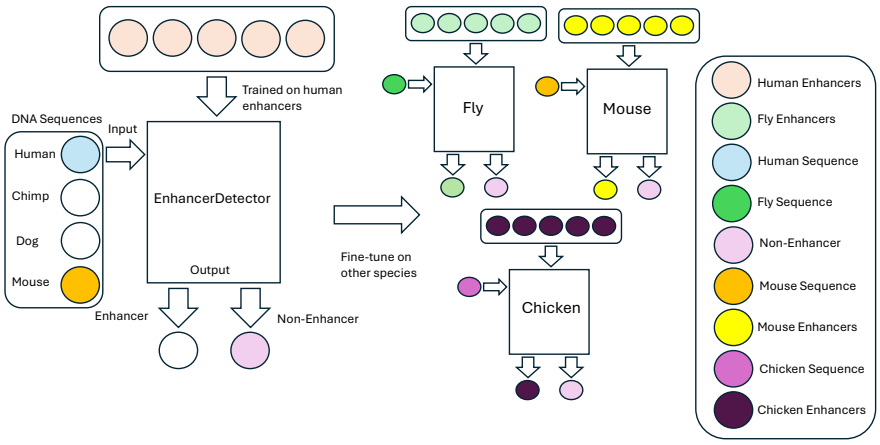


Fig. 1: General Architecture and Cross-Species Adaptability of *EnhancerDetector*. The left panel depicts the base model trained on the human CATlas dataset, which can be directly applied to closely related vertebrate species with minimal performance degradation. The right panel highlights *EnhancerDetector*'s support for fine-tuning, enabling the human-trained model to be adapted for more distantly related species. For illustrative purposes, fine-tuning is shown for fruit fly, mouse, and chicken, though the approach is broadly applicable to any species with sufficient enhancer data.

- It is easily fine-tuned to new species with limited data, achieving high performance with as few as 20,000 enhancer sequences.
- Experimental validation in transgenic flies confirms the predictive accuracy of *EnhancerDetector*, with five of six tested enhancers activating reporter expression.
- We utilize class activation maps to provide insights with *in silico* perturbation experiments. Our results demonstrate that (i) reversing CAM-identified core regions disrupts enhancer predictions, revealing direction-dependent internal sequence grammar despite the well-known orientation independence of enhancers relative to their target genes, and (ii) enhancer function emerges from an interplay between essential sequence motifs and their genomic context.
- *EnhancerDetector* consistently outperforms existing tools on both human and mouse datasets.
- Our tool identifies enhancer regions based on regulatory patterns that extend beyond simple sequence similarity.

2.2 Overview

EnhancerDetector is a flexible enhancer prediction framework that uses deep learning to classify genomic sequences as enhancers or non-enhancers. Given a sequence in FASTA format, the tool outputs a confidence score representing the probability that the input sequence functions as an enhancer. *EnhancerDetector* is centered on a convolutional neural network, with an optional ensemble mode to boost prediction confidence. These models are trained on curated enhancer datasets from both human and fruit fly genomes, and support fine-tuning for application to new species using limited enhancer data. *EnhancerDetector* also employs class activation maps, an explainable AI technique that highlights the most influential regions within input sequences. An overview of the architecture is shown in Figure 1.

2.3 *EnhancerDetector* Can Discover Human Enhancers

To assess our model's ability to recognize human enhancers, we trained a convolutional neural network on the CATlas human tissue dataset, which includes 400-bp sequences identified as putative enhancers using snATAC-seq across 222 cell types. We evaluated several network architectures by varying the number of convolutional filters, filter

Table 1: Training and validation performance of the human-trained convolutional network for enhancer discovery. Performance is reported on the full enhancer dataset and five control datasets: shuffled sequences, length-matched with repeats (LR), length-matched without repeats (LNR), length- and GC-matched with repeats (LGR), and length- and GC-matched without repeats (LGNR).

Dataset	Training					Validation				
	Accuracy (%)	**Recall** (%)	**Specificity** (%)	**Precision** (%)	**F1 Score** (%)	**Accuracy** (%)	**Recall** (%)	**Specificity** (%)	**Precision** (%)	**F1 Score** (%)
Overall	82.83	69.62	89.44	76.72	74.18	81.53	66.34	89.12	75.30	72.04
Shuffled	89.24	69.62	99.05	97.34	85.91	88.12	66.38	98.99	97.05	84.07
LR	81.92	69.59	88.09	74.50	72.76	80.59	66.34	87.72	72.98	70.59
LNR	82.23	69.60	88.55	75.25	73.24	80.99	66.41	88.28	73.91	71.20
LGR	80.64	69.63	86.15	71.53	70.86	79.29	66.40	85.74	69.96	68.70
LGNR	80.09	69.60	85.33	70.35	70.07	78.71	66.34	84.89	68.70	67.87

sizes, and dense layer neurons. The best-performing model consisted of 64 filters in the first convolutional layer (doubled in subsequent layers) with a filter size of 3 and 20 dense neurons. As shown in Table 1, our network achieves a validation F1 score of 72%, indicating a good balance between precision and recall. The validation recall score is 66%, demonstrating that our model is able to identify about two-thirds of human enhancers. The validation precision score is 75%, meaning that for every 100 positive predictions, 75 are true positives. Our human network has high specificity—89%—indicating that it can accurately identify non-enhancers 89 out of 100 times.

To test the impact of control sequence characteristics, we evaluated the model against five types of negative control datasets that varied in GC content and sequence repetitiveness, along with fully shuffled enhancer sequences. When sequences were shuffled, our network achieved a high precision and specificity of 99% and 97%, respectively. These high scores show that our network can reliably distinguish actual enhancers from random sequences with the same k-mer compositions. Additionally, we evaluated the model using negative control datasets designed to control for sequence length, GC content, and the presence or absence of repetitive elements: (i) LR (length- and repeat-matched), (ii) LNR (length-matched without repeats), (iii) LGR (length- and GC-matched with repeats), and (iv) LGNR (length- and GC-matched without repeats). We observed that the network performs best when using control sequences that do not match GC content, achieving an F1 score of 71% for both the LR and LNR datasets, with LNR having a slight advantage in precision and specificity scores of 74% and 88%, respectively, compared to LR. Conversely, the model struggles more when non-enhancer sequences match GC content but do not include repeated elements, resulting in an F1 score of 68% and the lowest precision and specificity scores of 69% and 85%. Collectively, these results show *EnhancerDetector* effectively learns sequence patterns characteristic of human enhancers and can predict them with high precision and specificity.

Table 2: Evaluation of the human-trained network (without and with fine-tuning) on the mouse dataset. In the top half of the table, training performance is omitted because the model was not trained on mouse enhancers—it was only evaluated on mouse data.

	Training					Validation				
Dataset	**Accuracy** (%)	**Recall** (%)	**Specificity** (%)	**Precision** (%)	**F1 Score** (%)	**Accuracy** (%)	**Recall** (%)	**Specificity** (%)	**Precision** (%)	**F1 Score** (%)
Human network with no fine-tuning										
Overall	—	—	—	—	—	79.07	66.42	85.40	69.46	68.37
Shuffled	—	—	—	—	—	86.51	66.82	96.36	90.17	80.72
LR	—	—	—	—	—	77.46	66.79	82.80	66.01	66.23
LNR	—	—	—	—	—	78.15	66.89	83.77	67.33	67.14
LGR	—	—	—	—	—	77.07	66.95	82.13	65.19	65.74
LGNR	—	—	—	—	—	77.21	67.00	82.31	65.45	65.92
Human network with fine-tuning										
Overall	84.39	75.05	89.06	77.42	76.45	83.02	71.72	88.67	75.99	74.35
Shuffled	88.24	75.09	94.82	87.88	83.13	86.92	71.79	94.49	86.69	81.05
LR	84.27	75.09	88.85	77.11	76.40	82.90	71.82	88.44	75.65	74.30
LNR	84.27	75.05	88.88	77.15	76.42	83.05	71.81	88.68	76.02	74.54
LGR	83.04	75.06	87.02	74.31	74.54	81.69	71.98	86.55	72.78	72.49
LGNR	82.16	75.12	85.68	72.40	73.25	80.82	71.79	85.34	71.00	71.24

2.4 *EnhancerDetector* Works on Another Vertebrate Species

To test our network's ability to discover enhancers in other species, we assembled a dataset of putative enhancers from the mouse (*Mus musculus*) genome. This dataset allows us to evaluate whether *EnhancerDetector* can generalize to another mammalian species. The Mouse ENCODE Project identifies putative enhancers based on DNase I hypersensitive sites. We obtained the mm10 ENCODE dataset from the UCSC Genome Table Browser. As with the human and fly enhancer datasets, we applied the same preprocessing procedure to the ENCODE dataset by removing promoters, exons, and insulators (regions containing CTCF binding sites). Promoters and exons were obtained from RefSeq and GENCODE (VM25) via the UCSC Genome Table Browser. After preprocessing, we selected potential enhancers shorter than 400 bp. Short regions were expanded bidirectionally to reach 400 bp in length. The final dataset contained 218,207 potential mouse enhancers. We then created five control datasets (shuffled, LR, LNR, LGR, and LGNR) using the same approach as for the human and fly controls. As shown in Table 2, the human-trained network achieved accuracy of 79%, recall of 66%, specificity of 85%, precision of 69%, and an F1 score of 68%. Using our base human network without any adjustments, we successfully identified two thirds of enhancers with approximately 70% precision. The model correctly excluded about 89% of non-enhancer regions. These results demonstrate that our network effectively generalizes enhancer patterns not only across different enhancer-detection biotechnologies but also across different vertebrate species.

To assess whether the predicted true positive enhancer sequences are conserved, we aligned them to the human genome using BLAST (Supplementary File 1). We found that only 6% of these sequences showed conservation (having at least 70% sequence identity and 70% alignment coverage). This low conservation rate demonstrates that our model identifies enhancer regions based on regulatory patterns that extend beyond simple sequence similarity. We next asked whether adapting the human-trained network via fine-tuning could improve performance on mouse enhancers.

2.5 Fine-Tuning Improves Our Mouse Enhancer Discovery

To further enhance our human network using the mouse dataset, we applied a fine-tuning technique with the potential mouse enhancers. Fine-tuning is a technique where a model that was already trained for one task is slightly adjusted to perform well on a new, related task, allowing it to keep what it has already learned while making small updates to adapt to the new data. This approach saves time and often leads to better performance, especially when the new task has less training data. We divided the enhancer and control datasets into training, validation, and testing sets using a 70%, 20%, and 10% split, respectively. To test this, we applied a fine-tuning strategy in which certain layers were frozen—their parameters remain fixed during training—while others were retrained on the mouse dataset. This technique allowed us to train a new network in which only the classifier was updated, ensuring that we retained features learned from the human enhancers while adapting to the mouse enhancers. As shown in Table 2, fine-tuning led to performance improvements across the board, compared to using the human-trained network without any adaptation. The fine-tuned

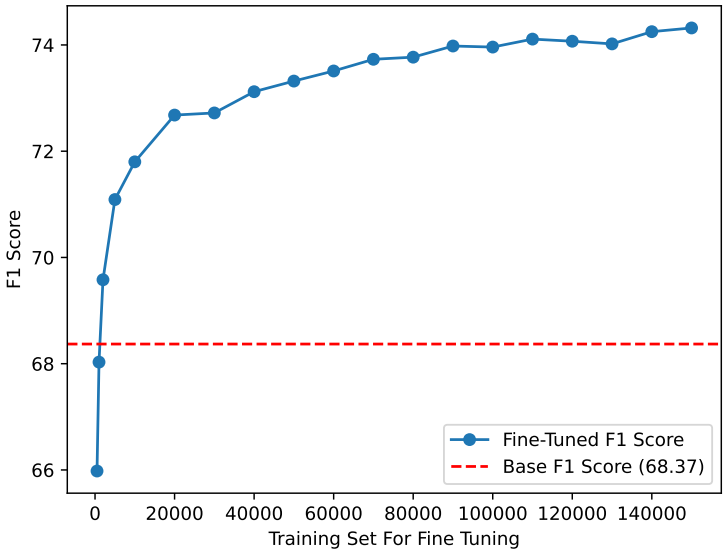


Fig. 2: Effect of training set size on fine-tuning performance. This plot shows the F1 score achieved by the human-trained network fine-tuned on mouse enhancer data using training subsets of increasing sizes. The x-axis indicates the number of mouse enhancer sequences used for fine-tuning, while the y-axis reports the corresponding F1 score on the validation set. Increasing the number of fine-tuning samples improves enhancer detection performance, with performance gains plateauing at larger dataset sizes.

network achieved an overall validation accuracy of 83%, compared to 79% for the original human network. It also showed higher recall (71% vs. 66%) and specificity (89% vs. 85%), along with improved precision and F1 scores (76% vs. 69% and 74% vs. 68%). These results highlight the effectiveness of fine-tuning in improving enhancer detection in mice, with notable gains in accuracy, recall, specificity, precision, and F1 score.

However, it is important to emphasize that even without fine-tuning, the human network remains a robust and reliable baseline. While fine-tuning provides substantial benefits when additional data is available, the human network alone offers a strong foundation for enhancer detection, delivering valuable results even in data-limited scenarios. Next, we asked: How much data are needed for effective fine-tuning?

2.6 The Human Network Can Be Successfully Fine-Tuned with Just 20K Mouse Enhancers

With the rise of large-scale genomic initiatives such as the 10,000 Vertebrate Genomes Project [46] and the Earth BioGenome Project [20], there is a growing demand for computational methods capable of predicting enhancers in newly sequenced genomes. Experimental approaches to enhancer identification are costly, labor-intensive, and time-consuming, making efficient *in silico* alternatives increasingly valuable. We have

shown that our human-trained network generalizes reasonably well to the mouse genome without fine-tuning. However, performance improves remarkably with fine-tuning on mouse enhancer data. Additionally, certain species—whether new or well-studied—may lack a sufficient number of confirmed enhancers for effective fine-tuning. This raises a practical question: How many enhancers are needed to effectively fine-tune the human network for use in a new species? To address this question, we tested multiple datasets with varying numbers of potential enhancers required for effective fine-tuning of the human network. We used potential enhancers from the mouse training dataset, which comprised 152,744 potential enhancers, and randomly selected subsets ranging from 500 to 150,000 enhancers for training. These enhancers were sampled from the ENCODE mouse dataset and were not stratified by cell type. For each subset of enhancers, we also randomly selected twice as many control sequences from the training dataset. These subsets of enhancers and control sequences allowed us to evaluate the impact of varying numbers of potential enhancers on the fine-tuning procedure.

F1 scores on the validation set for each model are shown in Figure 2. The baseline (using the human network without fine-tuning) achieved an F1 score of 68%. Models fine-tuned on just 500 or 1,000 mouse enhancers performed comparably to or slightly worse than the baseline (66% and 68%). Marked improvements were observed when the training set included 2,000 to 20,000 enhancers, reaching up to a 73% F1 score. These gains were achieved using only about 20% of the available mouse enhancer data. While increasing the dataset size beyond 20,000 led to marginal additional gains (e.g., up to 74%), the improvements became less pronounced, suggesting that, given pooled enhancer annotations without explicit cell-type stratification, approximately 20,000 enhancers provide a strong foundation for fine-tuning the human network to a new genome. These results demonstrate the potential of our fine-tuning approach for broader cross-species enhancer prediction. We next evaluated whether *EnhancerDetector* can also learn enhancer patterns in a more distant species, *Drosophila melanogaster*.

2.7 *EnhancerDetector* Can Learn Patterns in Fly Enhancers

To evaluate *EnhancerDetector*'s ability to identify enhancers in *Drosophila melanogaster*, we built multiple convolutional networks with varying numbers of filters, filter sizes, and dense layer neurons. We selected the two models that gave the best results (Table 3). These networks were trained using a combination of fly enhancer datasets: distal DNase I hypersensitive sites, distal scATAC-seq peaks, and cis-regulatory modules from REDFly. As with the human experiments, we used five negative datasets. Overall, both models had 67–68% validation F1 scores. Their validation recall scores ranged 63–68%, and specificity scores ranged 89–91%. Their precision scores were 67–70%. The scores on the training data were slightly higher than those obtained on the validation dataset but still comparable, demonstrating the lack of overfitting. The two networks performed well on the shuffled dataset. Both models showed strong specificity and precision when the control sequences did not contain repeats. Controlling for GC content made the classification task slightly more difficult: the F1 scores of the two networks on the LNR dataset (control for length only, excluding

repeats) were both 68%, whereas the scores on the LGNR (length and GC-controlled, no repeats) dataset were both 67%. The same trend—that controlling for GC content makes the task slightly more difficult—was observed on the datasets that include repeats. Compared to the performance of the human network, the trend related to GC content is consistent, but the trend related to the inclusion or exclusion of repeats is not. These results reaffirm previous findings [47] that enhancer sequences exhibit a discernible pattern and are characterized by a specific GC content.

2.8 Fine-Tuning the Human Network Achieved Comparable Performance to Fly Networks Trained Ab Initio

Having established strong *ab initio* performance on fly enhancers, we next tested whether a human-trained model can be adapted to fly via fine-tuning. Because the human training dataset was substantially larger than the fly dataset (8.9 million vs. 6.4 million samples), we hypothesized that the human-trained model had captured generalizable sequence features. Similar to our approach with the mouse dataset, we froze

Table 3: Training and validation performance of different neural networks on the fly enhancers. The models include two fly-trained convolutional networks, a fine-tuned human convolutional network on fly data, an ensemble combining ab initio and fine-tuned models for discovering fly enhancers, and a recurrent model specifically designed for fly enhancer sequences. Performance was also evaluated on five control datasets: four sets consisting of genomic sequences sampled to match the enhancer datasets in length and/or GC content, with or without repetitive elements (LR: Length Repeat; LNR: Length No Repeat; LGR: Length GC Repeat; LGNR: Length GC No Repeat), and a fifth set consisting of shuffled enhancer sequences.

	Training					Validation				
Dataset	**Accuracy** (%)	**Recall** (%)	**Specificity** (%)	**Precision** (%)	**F1 Score** (%)	**Accuracy** (%)	**Recall** (%)	**Specificity** (%)	**Precision** (%)	**F1 Score** (%)
Fly from Scratch (32 filters of size 3 and 20 dense units)										
Overall	85.96	66.72	92.38	74.47	71.67	83.82	68.45	88.94	67.35	67.69
Shuffled	90.63	66.75	98.59	94.05	82.73	90.88	68.49	98.35	93.25	83.19
LR	84.25	66.69	90.10	69.18	68.29	80.89	68.51	85.02	60.38	62.83
LNR	85.85	66.73	92.22	74.09	71.42	84.04	68.38	89.27	67.98	68.10
LGR	83.59	66.70	89.23	67.35	67.09	79.97	68.46	83.81	58.49	61.44
LGNR	85.47	66.70	91.73	72.90	70.67	83.28	68.47	88.23	65.96	66.74
Fly from Scratch (40 filters of size 3 and 20 dense units)										
Overall	84.92	59.21	93.49	75.21	68.96	83.94	62.92	90.94	69.84	67.34
Shuffled	89.44	59.22	99.52	97.61	80.21	90.26	62.91	99.37	97.09	82.16
LR	83.25	59.20	91.26	69.31	65.52	81.27	62.96	87.38	62.42	62.57
LNR	84.80	59.23	93.32	74.73	68.68	84.21	62.91	91.31	70.70	67.86
LGR	82.68	59.21	90.50	67.52	64.45	80.48	62.99	86.32	60.53	61.30
LGNR	84.48	59.28	92.89	73.53	68.02	83.54	62.96	90.40	68.60	66.59
Human Fine-tuned on Fly										
Overall	87.68	73.24	92.49	76.47	75.13	83.27	69.36	87.90	65.64	66.60
Shuffled	92.14	73.24	98.44	93.99	85.85	90.92	69.34	98.11	92.45	83.15
LR	86.04	73.25	90.31	71.58	72.10	80.14	69.40	83.73	58.70	61.83
LNR	87.61	73.26	92.39	76.24	75.18	83.49	69.47	88.16	66.18	67.20
LGR	85.39	73.25	89.43	69.80	70.87	79.14	69.40	82.39	56.77	60.38
LGNR	87.25	73.29	91.90	75.10	74.45	82.64	69.42	87.05	64.11	65.75
Ensemble of ab initio and Fine-tuned on Fly										
Overall	85.39	52.14	96.47	83.13	64.09	83.85	52.97	94.15	75.10	62.13
Shuffled	87.95	52.14	99.89	99.35	68.39	88.12	52.97	99.84	99.08	69.04
LR	84.33	52.14	95.06	77.88	62.46	81.95	52.97	91.62	67.80	59.48
LNR	85.43	52.14	96.52	83.32	64.14	84.11	52.97	94.48	76.20	62.50
LGR	84.00	52.14	94.62	76.36	61.97	81.40	52.97	90.87	65.92	58.74
LGNR	85.24	52.14	96.27	82.33	63.85	83.68	52.97	93.92	74.39	61.88
Recurrent Model for Fly Enhancers										
Overall	84.15	61.92	91.56	70.96	67.64	83.62	67.72	88.92	67.08	67.26
Shuffled	89.82	61.90	99.13	95.95	81.04	91.14	67.73	98.94	95.53	84.00
LR	82.27	61.90	89.04	65.33	64.11	80.71	67.70	85.04	60.14	62.41
LNR	83.76	61.89	91.05	69.74	66.87	83.71	67.66	89.05	67.33	67.38
LGR	81.52	61.93	88.04	63.32	62.81	79.79	67.77	83.79	58.23	61.06
LGNR	83.37	61.93	90.51	68.51	66.12	82.82	67.72	87.86	65.02	65.87

all layers during training except for the dense layers, which are responsible for making classifications rather than extracting features. This approach allowed the model to retain previously learned representations while adapting to fly-specific enhancer sequence patterns. As shown in Table 3, the fine-tuned network performed comparably to the two models trained from scratch on the fly dataset. The overall validation accuracy was 83%, compared to 84% for both fly-specific models. The fine-tuned network achieved a recall of 69%, slightly outperforming the two *ab initio* models (68% and 63%), while maintaining strong specificity (88% compared to 89% and 91%). Precision (66%) and F1 score (67%) were also closely aligned with those of the fly-trained models. These results demonstrate that fine-tuning a human-trained model can be as effective as training a model *ab initio*, offering a practical and efficient strategy for adapting *EnhancerDetector* to new species without the need for extensive retraining or large datasets.

2.9 A Network Ensemble Improves Prediction Confidence

After training three fly models—two trained from scratch on the fly dataset and one fine-tuned from the human-trained network—we observed that their individual performances were comparable, yet slightly different. To improve prediction confidence, we combined these models into an ensemble. In this approach, each model independently evaluates a sequence, and the sequence is classified as an enhancer only if all three models agree. This voting-based strategy prioritizes precision and specificity over recall by requiring unanimous consensus across models when it comes to positive predictions. As shown in Table 3, the ensemble achieved a specificity score of 94% and a precision score of 75%, outperforming the individual models (specificity scores of 89%, 91%, and 88%; precision scores of 67%, 70%, and 66%). Although recall dropped to 53%, indicating that the ensemble identified fewer true enhancers, it substantially reduced the number of false positives. These results demonstrate that the ensemble method provides a valuable option when higher confidence in predicted enhancers is needed, such as for experimental validation using transgenic animals.

2.10 *EnhancerDetector* Can Determine Enhancer Locations Within Longer Sequence Fragments

We selected 18 long sequences (median length 4 kb, range 3.2–7.1 kb) from the REDfly database. Each of these sequences has demonstrated regulatory activity and thus contains one or more individual enhancers. Our goal was to determine the locations of the enhancers within these long sequences. We cleaned the sequences by removing promoters, insulators, and exons (see Methods). We then split each long sequence into segments ranging from 460 to 500 base pairs and fed each segment into our ensemble of networks. Based on the confidence scores generated by the networks, we determined the locations of the highest-scoring segment(s) within each long sequence. We also shuffled the long sequences and ran them through the ensemble to ensure that our network could distinguish between real enhancer sequences and shuffled sequences. To test whether the network ensemble accurately predicted discrete enhancer locations within the longer sequences, we selected six predicted enhancers for *in vivo* testing

using reporter gene analysis in transgenic *Drosophila*. Results are shown in Figure 3. Three of the putative enhancers (HPC1–HPC3) were drawn from the same initial long sequence, while three others were taken from distinct long sequences (HPC4, HPC7, and HPC8). HPC1 drives weak reporter gene expression in the embryonic midgut and dorsal ectoderm (Figure 3 A, B), consistent with the activity of its enclosing longer sequence, enhancer ush_-12475/-6925 (REDflyID: RFRC:0000003436.004; Muratoglu et al. [48]). Interestingly, as reported in the original study [48], this expression appears to be ectopic relative to the endogenous expression of the assigned target gene, *ush*. HPC2 drives reporter gene expression in the midline of the embryonic ventral nervous system (Figure 3 C). This expression is not observed with the original longer enhancer. However, it is consistent with the known expression pattern of the *ush* gene, although a precise one-to-one correspondence between expressing cells remains to be determined [49]. Whether the discrepancy between the two reporter constructs

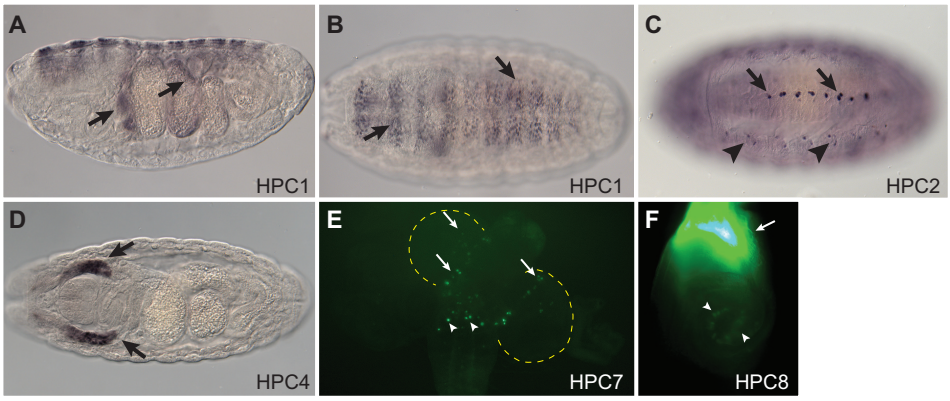


Fig. 3: *In vivo* validation of predicted enhancers. *Drosophila* enhancers of 460–500 bp in length were predicted from longer sequences (about 4,000 bp) with known regulatory activity and tested *in vivo* in a transgenic reporter gene assay. (A) HPC1 drives expression in the embryonic midgut (arrows) as well as (B) the dorsal ectoderm (arrows), consistent with described expression for the enclosing ush_-12475/-6925 enhancer [48]. (C) HPC2 drives reporter gene expression in the ventral midline (arrows), consistent with known expression of the *ush* gene but not the ush_-12475/-6925 enhancer. Arrowheads indicate expression driven by the plasmid vector, not the HPC2 enhancer. (D) HPC4 shows reporter gene expression in the embryonic salivary glands (arrows). Neither the larger enclosing enhancer (GMR26G07) nor its assigned target gene (*chn*) are reported to be expressed in this tissue. (E) HPC7 drives reporter expression in the larval brain (arrows; brain is outlined with yellow dotted lines) and ventral nerve cord (arrowheads). This resembles the described expression of the enclosing GMR52B12 enhancer. (F) HPC8 shows reporter gene expression in imaginal discs, including the leg discs (pictured). Very strong expression is observed in the presumptive proximal leg tissue (arrow), with weaker expression more distally (arrowheads), consistent with described expression of the *hth* target gene.

results from a mis-characterization of the original ush_-12475/-6925 reporter, is the result of a latent activity of the HPC2 fragment that is masked either *in vivo* or in the removed-from-context longer fragment, or is due to some other reason similarly remains to be explored. HPC3 failed to show reporter activity (data not shown). HPC4 exhibited strong reporter activity in the embryonic salivary gland (Figure 3 D). However, this is not a tissue where activity was observed with the original longer sequence (GMR26G07; REDflyID: RFRC:0000024999.002; Jenett et al. [14]). Moreover, ectopic expression in this tissue is not uncommon due to sequences in the reporter gene vector [50]. Thus, it is questionable whether this sequence should be considered a *bona fide* enhancer. HPC7 did not show significant embryonic activity but did drive limited gene expression in the larval brain and ventral nerve cord (Figure 3 E). This is consistent with the described activity of the longer enclosing sequence (GMR52B12; REDflyID: RFRC:0000027000; Jenett et al. [14]), although, again, an exact one-to-one correspondence has not been determined. HPC8 also lacked appreciable embryonic activity but was active in imaginal discs (Figure 3 F), consistent with part of the described activity of its longer enclosing sequence (GMR46A08; REDflyID: RFRC:0000030377; Jenett et al. [14]). In total, five of the six tested putative enhancers (83%) drove reporter gene activity, and four (67%) exhibited expression patterns consistent with those of their larger enclosing sequences and/or their likely target genes. These results strongly demonstrate that *EnhancerDetector* can successfully identify transcriptional enhancers.

2.11 Recurrent and Convolutional Networks Have Comparable Performance

To evaluate alternative architectures for enhancer prediction, we trained a Recurrent Neural Network (RNN) from scratch on the fly dataset and compared its performance to that of our three Convolutional Neural Networks (CNNs): two trained directly on fly data and one fine-tuned from the human-trained model. The RNN architecture employed gated recurrent units and followed a structure similar to the CNNs, including an embedding layer, two recurrent blocks, and two dense layers. As shown in Table 3, the RNN achieved validation performance comparable to that of the CNNs. Overall, the validation accuracy scores for the four networks were approximately the same (84% vs. 84%, 84%, and 83%). Recall scores were also similar (68% vs. 68%, 63%, and 69%), as were specificity (89% vs. 89%, 91%, and 88%) and precision (67% vs. 67%, 70%, and 66%). Importantly, the F1 scores—the harmonic mean of recall and precision—were nearly identical across all models (67% vs. 68%, 67%, and 67%). These results demonstrate that the RNN achieves performance comparable to the CNNs, indicating that both architectures are suitable for recognizing transcriptional enhancers. However, training the RNN was time-consuming. Each epoch required approximately 40 minutes, and training the network on the fly data took around 65 epochs, resulting in a total training time of about two days. In contrast, each epoch of the CNN took about 8 minutes, and training typically required 22 epochs on the fly data—resulting in a total training time of approximately three hours. Thus, the CNNs trained markedly faster while achieving similar performance. When we attempted to train an RNN on the human dataset, each epoch took nearly four hours, rendering the training process

impractically slow. For this reason, we did not pursue RNN training on the human data. In summary, our decision to prioritize CNNs over RNNs was clearly justified: they achieved comparable performance while drastically reducing the required training time.

2.12 *EnhancerDetector* Detects Putative Enhancers Identified by Alternative Biotechnologies

To evaluate our human network, we used an alternative set of potential enhancers identified using different methodologies. The FANTOM5 Project identifies putative enhancers based on cap analysis of gene expression, a method distinct from the snATAC-seq technology used to identify CATlas enhancers. We applied the same preprocessing pipeline to the FANTOM dataset as we did for the fly and human datasets, including the removal of promoters, exons, and insulators. Next, we extracted regions that are 400 bp or shorter; any region shorter than 400 bp was symmetrically expanded to 400 bp. This step was necessary because the human network—trained on CATlas enhancers—expects input sequences of 400 bp. After preprocessing, the dataset contained 41,553 potential enhancers. We first performed a baseline evaluation using our human network to assess its general ability to detect enhancers. To ensure a fair assessment, we then divided the FANTOM5 regions into four datasets: training, validation, testing, and non-overlapping. A region was assigned to the training, validation, or testing set if it overlapped a CATlas region in the corresponding set by at least 50%. The non-overlapping set included FANTOM regions that did not meet this overlap threshold with any CATlas training, validation, or testing region. This partition procedure resulted in 17,306, 4,896, 2,483, and 16,868 sequences in the respective datasets.

When evaluated on the full FANTOM5 dataset, the model achieved an accuracy of 73%, a recall of 50%, a specificity of 96%, and a precision of 93%. Although the recall was lower than in the CATlas-based evaluations, the high precision and specificity suggest that the model is stringent in its predictions, favoring precision over sensitivity and thereby reducing false positives. On the training, validation, and testing subsets of the FANTOM5 data, the network correctly identified 66–67% of the putative enhancers. Although the FANTOM5 testing subset contains some regions overlapping with CATlas, it is important to emphasize that the network had no access to this data during training or validation; hence, results on this set represent a true blind test. Detecting 67% of the putative enhancers in such a blind evaluation underscores our model's ability to generalize enhancer patterns across datasets.

On the non-overlapping subset—which contains sequences with no or minimal (<50%) overlap with CATlas—the network classified only 28% of the sequences as enhancers. While a 28% detection rate for the non-overlapping set may initially appear low, it is important to note that approximately 25–30% of sequences in the FANTOM5 dataset are not true enhancers. As reported by the FANTOM5 authors, "67.4–73.9% of the CAGE-defined enhancers showed significant reporter activity." In another *in vivo* validation using zebrafish embryos, they observed "tissue-specific enhancer activity with 3 of 5 fragments, which corresponded to the human enhancer tissue expression." Thus, the 28% recall rate on the non-overlapping set doesn't necessarily reflect poor model performance, but rather suggests that this subset predominantly comprises the

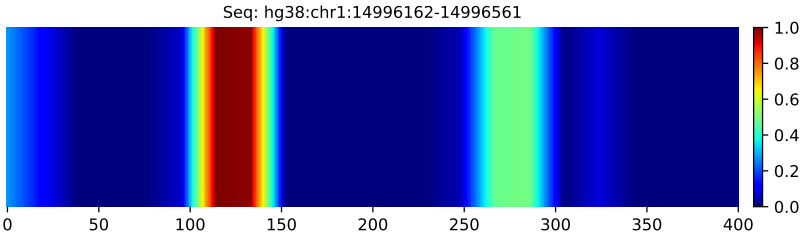


Fig. 4: Class Activation Map (CAM) visualization for a true enhancer—a sequence labeled as an enhancer and correctly classified as such by the neural network. CAM is a technique that assigns importance scores to each nucleotide in the sequence based on how much it contributed to the model's classification. These scores range from 0 (least important) to 1 (most important) and are visualized as a heat map: nucleotides with scores closer to 1 appear reddish, while those closer to 0 appear bluish. In this example, the dark red region spanning positions 110 to 140 indicates a strong enhancer signal and the green region spanning positions 260 to 290 indicates a moderate enhancer signal, suggesting that these portions of the sequence played a key role in the network's decision. The blue regions were less or not influential at all.

false positives within the FANTOM5 dataset, a hypothesis strongly supported by their minimal or absent overlap with the high-confidence single-cell data from CATlas.

These results demonstrate that our network effectively identifies a substantial portion of enhancers detected via both CAGE and snATAC-seq, highlighting its generalizability across datasets produced using different biotechnologies.

2.13 CAM-Based Knockout Experiments Reveal Biologically Significant Enhancer Regions

Class Activation Maps (CAMs) are visual tools that highlight the regions of an input sequence that are most influential in a neural network's classification decision. They help interpret which features the model has learned to associate with a given class. This technique outputs a score for each nucleotide in the sequence, typically ranging from 0 (least important) to 1 (most important). These scores can be visualized as a heat map: nucleotides with scores closer to 1 appear reddish, indicating high importance, while those closer to 0 appear bluish, indicating low importance. An example CAM is shown in Figure 4. In this figure, the dark red region spanning positions 110 to 140 highlights a portion of the sequence that strongly influences the network's classification. The green region spanning positions 260 to 290 highlights another portion of the sequence that moderately influences the network's classification. In contrast, the blue regions indicate areas with little to no relevance to the network's decision-making process.

To evaluate the biological significance of the regions highlighted by CAMs, we conducted a knockout experiment. The key idea was to disrupt ("knock out") the most important region of an enhancer sequence—identified using CAM—and assess whether the sequence was still classified as an enhancer by the network. We began with 25,000 true enhancer sequences and computed their CAM scores to identify regions of high importance. We defined important regions using a thresholding system with

values of 0.2, 0.3, 0.4, and 0.5. These thresholds represent the minimum average CAM score required for a region to be considered important. Based on this criterion, we selected sequences containing only one important region between 25 and 100 bp in length. This constraint was critical: destroying long important regions would disrupt a large portion of the sequence and likely lead to a negative classification, while regions shorter than 25 bp would be too small to meaningfully impact classification. For each selected sequence, we shuffled the nucleotides within the important region, effectively removing its recognizable pattern. As a control, we shuffled a region of the same length elsewhere in the sequence—ensuring it did not overlap with the important region. The classifier was then applied to modified sequences to observe the effect of the knockouts.

Finally, we assessed the significance of the results using a binomial test, which calculates p-values based on the number of sequences tested. This statistical evaluation helped confirm the observed differences in classification outcomes were not due to random chance, but rather to the disruption of biologically relevant regions identified by CAMs.

Shuffling the regions identified as important by CAMs significantly reduced the number of sequences classified as enhancers across all thresholds. At a threshold of 0.2, 1,409 sequences (94% of true enhancers with important regions averaging an intensity of 0.2) were reclassified as non-enhancers. This effect persisted across higher thresholds: 2,251 sequences (92%) at 0.3, 3,070 (90%) at 0.4, and 4,345 (87%) at 0.5. In total, disrupting CAM-identified regions led to loss of enhancer classification in 87–94% of cases. As a control, we shuffled a region outside the CAM-highlighted area within the same sequences. This had a smaller but still substantial effect: at thresholds of 0.2, 0.3, 0.4, and 0.5, the number of sequences reclassified as non-enhancers was 1,281 (85%), 1,928 (79%), 2,443 (72%), and 3,371 (68%). Next, we computed p-values comparing the proportion of negative classifications resulting from shuffling important versus non-important regions. The p-values for thresholds 0.2 through 0.5 were 6.87×10^{-26}, 8.94×10^{-71}, 1.34×10^{-151}, and 1.08×10^{-225}—all highly significant. We also evaluated two alternative knockout strategies: reverse sequencing (e.g., AGCT $\rightarrow$ TCGA) and transversion mutation, where each nucleotide (A, C, G, T) is randomly replaced with either of the two remaining non-complementary bases (e.g., A becomes G or C, determined by chance). Both approaches yielded reclassification rates comparable to the shuffling approach. At a threshold of 0.2, the reclassification rates to non-enhancer were 92% for the mutation approach and 95% for reverse sequencing vs. 94% for the shuffled knockout. At 0.3, the rates were 89% and 92% vs. 92%. For 0.4, we observed 86% and 91% vs. 90%. Finally, at 0.5, the rates were 84% and 88% vs. 87%. All of these percentages were statically significant.

These results demonstrate that the regions identified by CAMs are not only predictive but also biologically meaningful. Disrupting the important regions has a greater impact on enhancer classification than disturbing other regions. Notably, the experimental reversal of key enhancer regions strongly suggests a directional nature, independently confirming recent observations [51].

2.14 Insertion Experiment Further Supports the Biological Relevance of CAM-Identified Regions

To further demonstrate the importance of the CAM-identified regions, we conducted an insertion experiment. The goal was to test whether inserting a high-scoring region from a true enhancer into a non-enhancer sequence could cause the network to reclassify it as an enhancer. We began with 25,000 true enhancer sequences and applied the same selection criteria as in the knockout experiment: each sequence had to contain exactly one important region identified by CAM, with a length between 25 and 100 bp. For each selected enhancer, we paired it with a true negative sequence—one that was labeled and correctly classified as a non-enhancer. We then inserted the important region from the enhancer into the corresponding negative sequence at the same relative position. After this modification, the new sequence was passed through the network for reclassification. As a control, we performed a parallel experiment in which a randomly selected region of the same size–but from outside the CAM-highlighted area–was inserted into the negative sequence in the same location. This ensured that any observed changes in classification were specifically due to the informative region identified by CAM, rather than arbitrary sequence changes. Finally, we evaluated the significance of the observed classification changes using a binomial test, demonstrating that the effects were not due to random chance.

For CAM thresholds of 0.2, 0.3, 0.4, and 0.5, the number of sequences reclassified as enhancers was 1,132 (76%), 1,937 (79%), 2,745 (80%), and 4,030 (81%). As a control, we inserted regions outside of the CAM-identified important areas into true negatives. The number of sequences reclassified as enhancers in this case was 877 (59%), 1,485 (61%), 2,042 (60%), and 2,990 (60%) for the respective thresholds. We then calculated p-values comparing the reclassification rates resulting from inserting important versus non-important regions. The p-values for thresholds 0.2 through 0.5 were 1.70×10^{-43}, 4.01×10^{-85}, 3.87×10^{-146}, and 2.65×10^{-220}—all highly significant. Insertion of CAM-identified regions consistently increased enhancer classification rates across all thresholds, with stronger effects at higher CAM intensities. In contrast, insertion of non-important regions had moderate but notably weaker impact. These findings confirm the biological relevance of CAM-identified regions in enhancer sequences.

2.15 Sequence Context Surrounding Enhancer Regions Contributes to Classification

While our knockout and insertion experiments demonstrated the functional significance of CAM-identified regions, we next examined whether the surrounding sequence context also influences enhancer classification. Specifically, we asked: Is preserving only the important region sufficient for the model to recognize a sequence as an enhancer? We designed the experiment to be similar to the knockout experiment, except instead of shuffling the important region itself, we shuffled the adjacent context of same length as the important region. For example, if the important region is 100 base pairs long, we shuffle the 50 base pairs directly before and after it, while keeping the important region unchanged. This allowed us to test whether the network relies on the broader

sequence context, beyond the specific important region, for enhancer classification. As with the previous experiments, we selected 25,000 true enhancer sequences, applying the same criteria for selection: each sequence contained one important region between 25 and 100 bp in length. Once the sequences were chosen, we took the length of the important region and shuffled the regions on both adjacent sides of the important region to match. After modifying the sequences, we passed them through the network again to determine whether they would still be classified as enhancers.

At a CAM threshold of 0.2, 1,277 sequences (85% of the samples) were reclassified as non-enhancers. For thresholds of 0.3 and 0.4, the number of reclassified sequences was 1,959 (80%) and 2,524 (74%). At threshold 0.5, 3,553 sequences (71%) were reclassified as non-enhancers. Similar to the knockout experiment, we applied the following three disturbance approaches: (i) k-mer shuffling, (ii) transverse mutation, and (iii) sequence reverse. The results were comparable to those of the shuffled approach. At the 0.2 threshold, reclassification rates were 82% for mutation, 86% for reverse, and 85% for shuffled. At 0.3, the rates were 76% and 81% vs. 80%. For 0.4, we observed 71% and 75% vs. 74%. Finally, at 0.5, the rates were 69% and 74% vs. 71%.

These results indicate that preserving only the most predictive region is insufficient for accurate enhancer classification. Disrupting the surrounding sequence context often leads the model to reject the sequence as an enhancer, even when the core region remains intact. This finding is consistent with biological evidence suggesting that enhancer function depends not only on core transcription factor binding motifs but also on adjacent nucleotides that influence chromatin accessibility and higher-order regulatory architecture [52].

A comparison with the knockout experiments reveals that both the core region and its context contribute to enhancer activity. However, perturbing the core region resulted in a higher rate of loss-of-function predictions, indicating it plays a more central role. Together, these findings suggest that enhancer function emerges from an interplay between essential sequence motifs and their genomic context—both of which are effectively captured by our model.

2.16 *EnhancerDetector* Outperforms LS-GKM and DeepSEA on the CATlas Testing Set

We evaluated *EnhancerDetector* on the held-out CATlas test set, comparing its performance with the k-mer SVM LS-GKM and the chromatin-feature predictor DeepSEA. We could not evaluate other related tools—e.g., Enhancer-FRL, Enhancer-LSTMAtt, and PDCNN—on the CATlas dataset because these tools require 200-bp-long input sequences, whereas the CATlas sequences are 400 bp long; downsampling to 200 bp would likely remove informative context and yield an unfair comparison. All CATlas evaluations are based on a test set with a 1:10 enhancer-to-control ratio. The results are shown in Figure 5.

We first compared our network to LS-GKM trained on the CATlas dataset, using the same unseen test set for evaluation. Our model outperformed LS-GKM in overall accuracy (82% vs. 69%). While LS-GKM achieved higher recall (72% vs. 66%), our model showed superior specificity (89% vs. 69%), precision (75% vs. 19%), and F1 score (72% vs. 30%). These results suggest that LS-GKM tends to over-classify

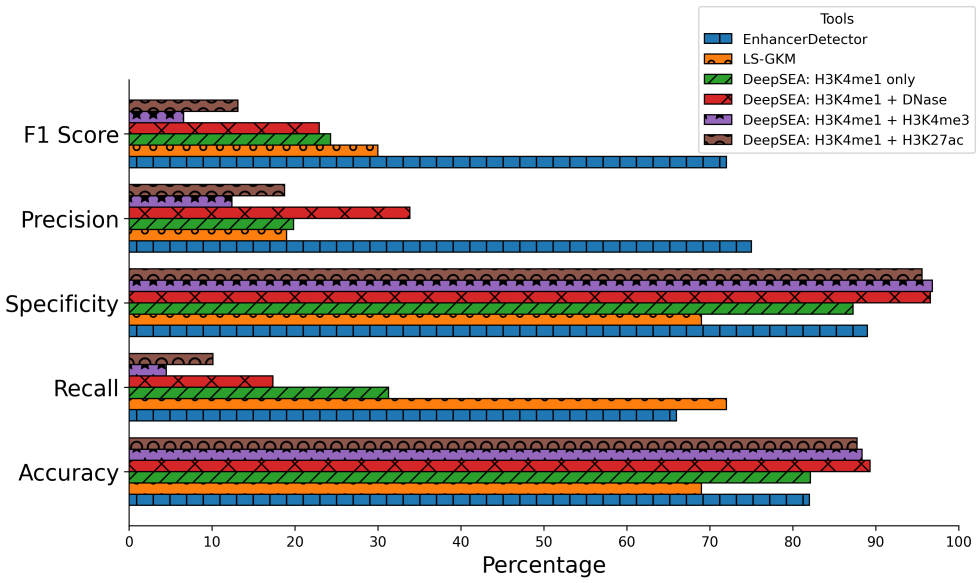


Fig. 5: Performance of *EnhancerDetector* compared with LS-GKM and DeepSEA on the CATlas human test set. Grouped horizontal bars showed accuracy, recall, specificity, precision, and F1 score for each tool. *EnhancerDetector* achieved the highest precision and F1 score while also maintaining strong recall and specificity. LS-GKM identified more positives but at the cost of substantially reduced precision. DeepSEA, evaluated using four chromatin-mark rules, was conservative, showing high specificity but low recall and precision.

sequences as enhancers, whereas *EnhancerDetector* achieves a more balanced performance, making it better suited for genome-scale enhancer discovery. To further assess performance beyond single-point metrics, we generated receiver operating characteristic (ROC) and precision-recall (PRC) curves. When trained with 50,000 enhancer and 50,000 control sequences, LS-GKM achieved an AUROC of 0.775 and an AUPRC of 0.275, reflecting its higher recall but limited precision. In contrast, *EnhancerDetector* achieved an AUROC of 0.879 and a substantially higher AUPRC of 0.507, demonstrating a more favorable balance between sensitivity and precision particularly important in the context of highly imbalanced genomic data.

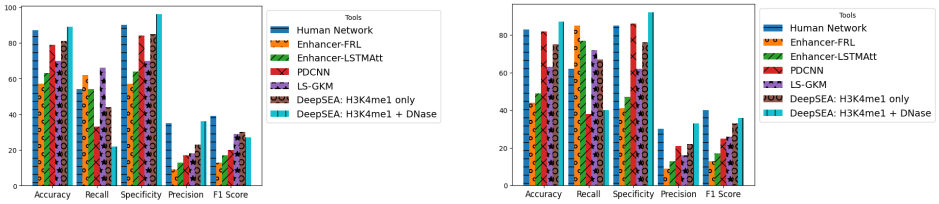
LS-GKM is based on support vector machines, which are known for their limited ability to scale on large data sets. For this reason, we further optimized LS-GKM; we trained additional models using progressively smaller subsets of the CATlas training data (ranging from 10k to 40k enhancers and controls). We observed a steady improvement in both AUROC and AUPRC with increasing training size: AUROC rose from 0.749 (10k) to 0.771 (40k), while AUPRC increased from 0.239 to 0.269. Despite these gains, the 50k model remained the best-performing LS-GKM configuration yet still did not surpass *EnhancerDetector*. We did not exceed 50k samples, as the LS-GKM

documentation recommends a maximum of 100,000 total sequences, and we aimed to remain within that constraint.

We then compared to DeepSEA, a widely used deep learning framework designed for regulatory variant prediction [42], specifically to their Beluga model [53]. While *EnhancerDetector* can distinguish enhancers from controls using direct sequence classification, we sought to assess how it performs relative to an alternative approach that infers enhancer activity through chromatin feature prediction. DeepSEA is not a dedicated enhancer detector; rather, it predicts cell-type specific open chromatin features such as DNase I hypersensitivity and histone modifications (e.g., H3K4me1, H3K27ac, H3K4me3) from DNA sequences. In this section, we evaluate DeepSEA on the CATlas human test set to enable a controlled comparison under identical conditions. In the next section, we extend this benchmarking framework to independent ENCODE human and mouse enhancer datasets.

Since DeepSEA requires 2000-bp-long input sequences, we symmetrically extended each 400 bp region to 2,000 bp by adding flanking sequence on both sides, keeping the original enhancer region centered. To interpret DeepSEA's predictions in the context of enhancer classification, we defined enhancers based on the presence of key chromatin marks. Specifically, we considered four scenarios: (1) H3K4me1 alone, a primary enhancer-associated mark; (2) H3K4me1 + DNase, reflecting accessible chromatin surrounding enhancer elements; (3) H3K4me1 + H3K27ac, indicative of active enhancers; and (4) H3K4me1 + H3K4me3, while traditionally associated with promoters, has shown signals in enhancers although less than H3K4me1 [54]. We defined a region as an enhancer if the predicted probability for any combination exceeded 0.5, regardless of cell type. This cell-agnostic thresholding reflects our focus on general enhancer potential rather than cell-specific activity.

We evaluated DeepSEA on the CATlas human test set. Across all chromatin mark combinations, DeepSEA achieved an overall accuracy ranging from 82% to 89% (H3K4me1 alone and H3K4me1 + DNase), with recall ranging from as low as 4% (for H3K4me1 + H3K4me3) to 31% (for H3K4me1 alone). Specificity ranged from 87% (H3K4me1 alone) to 97% (H3K4me1 + H3K4me3), indicating the model's tendency to classify sequences as negatives. Precision and F1 score were both low overall, for precision ranging from 12% (H3K4me1 + H3K4me3) to 20% (H3K4me1 alone) and F1 score ranging from 6% (H3K4me1 + H3K4me3) to 24% (H3K4me1 alone). In contrast, *EnhancerDetector* achieved substantially higher performance on recall, precision and F1 score while maintaining high specificity, with 82% accuracy, 66% recall, 89% specificity, 75% precision, and 72% F1 score. These results show that while DeepSEA effectively captures chromatin-associated signals from DNA sequence, its indirect approach to enhancer inference may limit its sensitivity for this specific task. In contrast, *EnhancerDetector* is optimized for enhancer classification and provides the best compromise between recall and precision, as evidenced by the highest F1 score, making it well suited for direct enhancer detection.



(a) ENCODE Human Enhancers (b) ENCODE Mouse Enhancers

Fig. 6: Performance comparison of enhancer classification tools on human and mouse ENCODE datasets. Bar plots show accuracy, recall, specificity, precision, and F1 score for six enhancer classification tools: our human network, Enhancer-FRL, Enhancer-LSTMAtt, PDCNN, LS-GKM, and two versions of DeepSEA, which is a sequence-to-expression tool. All tools were evaluated on the same test set for mouse enhancers. All tools except DeepSEA were evaluated on the same test set for the human enhancers. To evaluate DeepSEA, the human data were downsampled (40,000 enhancers and 400,000 controls) due to file size limits and the need to upload many files manually to DeepSEA's website. Our human network consistently achieved the highest F1 score while maintaining competitive accuracy, recall, specificity, and precision.

2.17 *EnhancerDetector* Outperforms Multiple Enhancer-Discovery Tools on Independent ENCODE Datasets

To evaluate performance on an independent benchmark distinct from CATlas, we next compared our human-trained network to multiple enhancer-classification tools on ENCODE human and mouse datasets. We compared our human-trained network to several other binary classification tools for human enhancer prediction: Enhancer-FRL, Enhancer-LSTMAtt, PDCNN, and LS-GKM (unlike the previous section, where LS-GKM was evaluated on CATlas, here all tools were evaluated on ENCODE-derived enhancer datasets based on DNase I hypersensitivity). We evaluated all tools on the ENCODE datasets for both human and mouse (Figure 6). All evaluations were performed on a test set with a 1:10 enhancer-to-control ratio, reflecting the natural sparsity of enhancers and imposing a strong penalty on precision and F1 score.

Against LS-GKM (trained on CATlas but evaluated here on ENCODE), our model achieved higher accuracy (87% vs. 70%), specificity (90% vs. 70%), precision (35% vs. 18%), and F1 score (42% vs. 29%) on the human data, though LS-GKM retained higher recall (66% vs. 54%). On the mouse data, LS-GKM again showed higher recall (72% vs. 62%) but substantially lower precision (16% vs. 30%) and F1 (26% vs. 40%).

Enhancer-FRL and Enhancer-LSTMAtt displayed a similar tradeoff: both achieved higher recall but much lower precision and specificity. For example, Enhancer-FRL reached 62% recall on the human data but only 9% precision, compared to our model's 52% recall and 35% precision (F1: 13% vs. 42%). Enhancer-LSTMAtt likewise showed

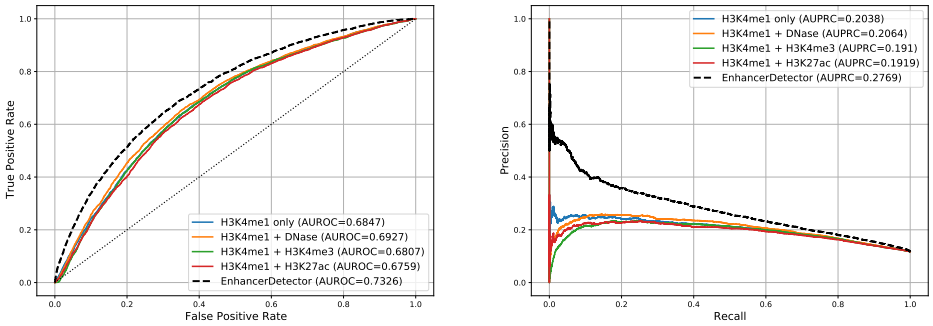
higher recall on the mouse data (77% vs. 62%) but far lower precision (13% vs. 30%), leading to a weaker F1 score (17% vs. 40%).

When compared to PDCNN, our network again performed better across metrics. On the human data, our model achieved 87% accuracy and 42% F1, compared to PDCNN's 79% accuracy and 20% F1. On the mouse data, PDCNN reached 82% accuracy and 25% F1, while our model matched its accuracy (83%) but achieved much stronger recall (62% vs. 38%) and F1 (40% vs. 25%).

In sum, our human network consistently outperforms Enhancer-FRL, Enhancer-LSTMAtt, PDCNN, and LS-GKM. It achieves superior F1 score on both the human and mouse datasets, providing the best balance between recall and precision. Despite being trained solely on human enhancers, our model generalizes strongly across species and experimental assays, including from snATAC-seq-derived CATlas enhancers to DNase I hypersensitivity-based ENCODE datasets, making it a reliable enhancer prediction tool.

2.18 *EnhancerDetector* Outperforms Sequence-to-Expression Models on ENCODE Datasets

Using the same ENCODE human and mouse enhancer datasets described in Section 2.17, we next compared *EnhancerDetector* to two general sequence-to-expression models, DeepSEA (previously evaluated on the CATlas human testing set under Section 2.16) and Enformer. These two tools predict chromatin features rather than enhancer labels directly. To ensure fairness, we adapted our evaluation to each



(a) ROC Curve: EnhancerDetector vs Enformer-based Rules

(b) Precision-Recall Curve: EnhancerDetector vs Enformer-based Rules

Fig. 7: Comparison of enhancer classification performance between *EnhancerDetector* and four feature-combination rules derived from Enformer's predictions (H3K4me1, H3K4me1 + DNase, H3K4me1 + H3K4me3, and H3K4me1 + H3K27ac). Panel (a) shows the receiver operating characteristic (ROC) curves and area under the ROC curve (AUROC) values; panel (b) shows the precision-recall curves (PRC) and area under the precision-recall curve (AUPRC) values. *EnhancerDetector* consistently outperforms all Enformer-based rule combinations in both AUROC and AUPRC.

model's input requirements while maintaining the same enhancer-to-control ratios as in our other benchmarks.

DeepSEA requires 2000 bp inputs, so each 400 bp sequence from the ENCODE human and mouse test sets was symmetrically extended to 2000 bp. For consistency, we evaluated DeepSEA alongside the other enhancer-discovery tools using the same four cell-agnostic rules (H3K4me1; H3K4me1+DNase; H3K4me1+H3K27ac; H3K4me1+H3K4me3). We preserved the 1:10 enhancer-to-control ratio throughout. We evaluated it on the entire mouse dataset, but had to downsample the human data (40,000 enhancers and 400,000 controls) due to file size limitations and the manual effort required to upload a large number of files to DeepSEA's website. Summary results are shown in Figure 6. On human data, DeepSEA achieved 81% and 89% accuracy and 15% and 30% F1 (depending on feature-rule choice), compared to *EnhancerDetector*'s 87% accuracy and 39% F1. On mouse data, DeepSEA reached 75% and 87% accuracy and 25% and 36% F1, while *EnhancerDetector* achieved 83% accuracy and 40% F1. Thus, given the 1:10 class imbalance where F1 is the most informative metric, *EnhancerDetector* consistently outperformed DeepSEA, even when their accuracy scores were similar.

To further benchmark *EnhancerDetector* against regulatory sequence models, we compared it to Enformer [44], a transformer-based deep learning model trained to predict thousands of functional genomic signals from long input sequences. Although Enformer is not explicitly designed as a binary enhancer classifier, it provides high-resolution predictions of chromatin accessibility and histone marks such as H3K4me1, H3K27ac, H3K4me3, and DNase hypersensitivity—features commonly associated with enhancers. We evaluated Enformer using our CATlas testing dataset. First, we identified transcription start sites (TSSs) where a 57,344 bp window on both sides (114,688 bp total) overlapped one or more annotated enhancer regions from our testing set. Matched control TSSs were selected with 114,688-bp-long windows. All selected sequences were expanded to 393,216 bp, as required by Enformer's input length. We then selected 1,000 non-overlapping TSS-centered regions and ran Enformer on each of them. From the resulting output matrix, we extracted the predictions corresponding to any cell type for the four enhancer-associated features: H3K4me1, H3K27ac, H3K4me3, and DNase. For each enhancer or control region, we identified the bins in the output that overlapped the region and computed the mean signal for each of the four features. This resulted in four averaged columns per region (one for each feature). To normalize across regions, we computed the mean and standard deviation of each column and then transformed these values into z-scores by subtracting the column's mean and dividing by the standard deviation. Using these normalized feature scores, we applied the same rule-based strategies as used for our DeepSEA assessment (H3K4me1 alone, H3K4me1 + H3K4me3, H3K4me1 + DNase and H3K4me1 + H3K27ac), for each rule, we computed both AUROC and AUPRC. Note that we did not compute traditional classification metrics (e.g., precision, recall at a threshold), as Enformer outputs continuous chromatin signal predictions rather than binary labels. Thus, interpreting a specific threshold as "active" is less meaningful and would introduce arbitrary cutoffs.

For *EnhancerDetector*, we used the same set of 114,688 bp TSS-centered regions and segmented them into overlapping 400 bp windows with a 25-bp stride. We then ran *EnhancerDetector* on each window and then took the windows that overlapped with the enhancer or control regions. We then calculated the average for those windows; from these average scores, we calculated both AUROC and AUPRC for *EnhancerDetector*. This setup allowed us to directly compare Enformer's inferred chromatin profiles to *EnhancerDetector*'s enhancer probabilities.

Using the normalized feature scores derived from Enformer's chromatin predictions, we evaluated the performance of four feature-based enhancer rules across the CATlas testing set. It is important to note that due to the relative scarcity of enhancer regions overlapping with TSS-centered regions, the selected dataset was skewed toward controls, resulting in an approximate enhancer-to-control ratio of 1:7.5. As shown in Figure 7, the combination of H3K4me1 and DNase yielded the highest performance among the Enformer-derived rules, with an AUROC of 0.6927 and an AUPRC of 0.2064. This was followed closely by the H3K4me1-only rule (AUROC = 0.6847, AUPRC = 0.2038). The lowest performance was observed for the H3K4me1 + H3K27ac rule (AUROC = 0.6759, AUPRC = 0.1919). In contrast, *EnhancerDetector* directly predicted enhancer probability scores across the same regions and achieved an AUROC of 0.7326 and an AUPRC of 0.2769, outperforming all Enformer-based rules. These results highlight that while Enformer provides rich chromatin signal predictions that could infer enhancer activity, it does not offer direct enhancer classification. Users must infer activity by combining multiple features and applying custom thresholds. In contrast, *EnhancerDetector* simplifies this process by directly predicting enhancer status from 400 bp sequences, making it a more streamlined and user friendly option for enhancer discovery.

We further evaluated both models using our ENCODE mouse testing set, following the same procedures as with the human dataset. However, due to the smaller number of mouse enhancers and controls, we increased the number of control TSS-centered sequences from 1,000 to nearly 5,000. This adjustment yielded a more realistic evaluation, producing an enhancer-to-control ratio of approximately 1:7.4 closely matching the 1:7.5 ratio used in the human testing. Enformer was evaluated using its mouse model and features, while *EnhancerDetector* was tested using its fine-tuned mouse model. *EnhancerDetector* achieved comparable results to Enformer with an AUROC of 0.8458. The best AUROC was obtained with the H3K4me1 + DNase combination (AUROC = 0.8621), followed by H3K4me1 + H3K27ac (0.8556), H3K4me1 + H3K4me3 (0.8548), and H3K4me1 alone (0.8540). In terms of AUPRC *EnhancerDetector* also showed comparable results to all Enformer based rules. The best AUPRC among the Enformer rules was 0.4038 (H3K4me1 + DNase), compared to 0.4256 due to *EnhancerDetector*. The results stayed comparable with 0.3967, 0.3945 and 0.3915 from H3K4me1 + H3K27ac, H3K4me1 only and H3K4me1 + H3K4me3, respectively. These results demonstrate that *EnhancerDetector* is able to match Enformer derived enhancer rules in both AUROC and AUPRC for mouse sequences, while offering a more direct and streamlined solution for enhancer prediction without the need for handcrafted feature combinations or post-hoc rule design.

Overall, *EnhancerDetector* delivers higher or comparable performance while offering a critical practical advantage: DeepSEA and Enformer require users to select among multiple chromatin-mark rules or thresholds, with the "best" choice varying by dataset. In contrast, *EnhancerDetector* outputs a direct enhancer probability from short 400-bp-long sequences, eliminating post hoc rule-tuning and providing a simpler, more consistent solution for enhancer discovery across species and datasets.

3 Conclusion

Understanding how enhancers encode regulatory information in DNA is a central challenge in genomics, particularly as genome sequencing continues to advance more rapidly than functional annotation. A key open question is whether enhancers share an intrinsic, sequence-encoded identity—here termed enhancerness—that distinguishes them from other genomic regions independently of species, cell type, or experimental assay. In this context, establishing the existence and learnability of enhancerness is both a fundamental biological problem and a practical requirement for scalable genome annotation. Below, we summarize how our results address this challenge and demonstrate the implications of learning enhancer identity directly from DNA sequence.

In this study, we introduced *EnhancerDetector*, a convolutional neural network-based framework for accurate and interpretable identification of transcriptional enhancers across cell types and across species. *EnhancerDetector* consistently outperforms existing enhancer prediction methods, achieving superior F1 scores on both human and mouse datasets. The tool maintains an excellent balance between recall and precision, which are critical for genome-wide applications.

EnhancerDetector directly classifies enhancers from DNA sequences, outperforming or achieving comparable performance to indirect sequence-to-expression models like DeepSEA and Enformer, which require additional thresholding and interpretation of chromatin feature predictions.

Our model generalizes well to independent datasets derived from diverse experimental assays, including snATAC-seq, CAGE, and DNase-seq.

A key strength of *EnhancerDetector* is its cross-species transferability: the model trained on human enhancers performs robustly on mouse and fly data and can be effectively fine-tuned using as few as 20,000 enhancers—a relatively small dataset in genomic studies. This capability makes it particularly valuable for enhancer discovery in newly sequenced genomes with limited experimental data.

We harnessed model interpretability through class activation maps to highlight sequence regions most predictive of enhancer activity. Computational perturbation analyses—including knockout, insertion, and sequence context disruption—confirmed the biological relevance of these regions. These analyses revealed that enhancers generally possess a core functional region supported by a broader sequence context. Our *in silico* experiments confirmed the directional nature of enhancers.

For experimental validation, we tested six putative enhancers in transgenic flies. Five (83%) drove reporter gene expression, and four (67%) showed expression patterns consistent with prior literature or their putative target genes.

In sum, *EnhancerDetector* offers a high-performing, biologically grounded, and easily adaptable platform for enhancer discovery. It lays a strong foundation for next-generation computational regulatory genomics, enabling accurate enhancer identification at scale—a critical capability as genomic data expand exponentially.

4 Materials and Methods

4.1 Enhancer Dataset

For the human classifier, we utilized a comprehensive set of putative enhancer sequences curated from the Cis-element Atlas (CATlas, which we will henceforth refer to as the CATlas dataset. This dataset was generated through the application of single-nucleus open chromatin accessibility sequencing (snATAC-seq) across 222 distinct cell types derived from both adult and fetal human tissues. The identified regions of open chromatin should encompass a diverse repertoire of regulatory elements, including transcriptional enhancers. The sequences within the CATlas dataset are uniformly 400 base pairs (bp) in length. For the *Drosophila melanogaster* classifier, we integrated distal DNase I hypersensitive sites identified by the Stamatoyannopoulos Laboratory [55], distal scATAC-seq regions from the Furlong Laboratory [56], and experimentally validated *cis*-regulatory modules (CRMs) sourced from REDfly [57].

4.2 Preprocessing

To ensure the high quality of our enhancer datasets, we implemented a stringent preprocessing pipeline to exclude potentially confounding genomic features, namely promoters, coding regions (exons), and insulators, from both the human and fly datasets. For the human data, we leveraged the RefSeq and GENCODE annotations to precisely delineate transcription start sites and the genomic coordinates of coding exons. Similarly, for the fly data, we employed the RefSeq and Ensembl databases to obtain analogous information. We defined a promoter region as a 1000 bp region centered on each annotated transcription start site. Human insulator regions were retrieved from the ENCODE Project via the UCSC Genome Table Browser, specifically selecting regions characterized by the presence of CTCF binding sites. For the fly, we incorporated putative insulator regions derived from ChIP-based binding profiles of known insulator-associated proteins derived from a previously published study [58]. All instances of overlap between these potentially confounding regions and our initial enhancer datasets were systematically removed using BEDTools [59]. Finally, to ensure uniformity in genomic coordinates for the fly data, all sequences were mapped to the DM6 genome assembly using the Liftover tool [60].

4.3 Control Datasets

To rigorously evaluate our model's ability to discriminate true enhancers from non-functional genomic regions, we generated five distinct control datasets representing negative examples. Four of these datasets were specifically designed to control for potential confounding factors: sequence length, repetitive element content, and

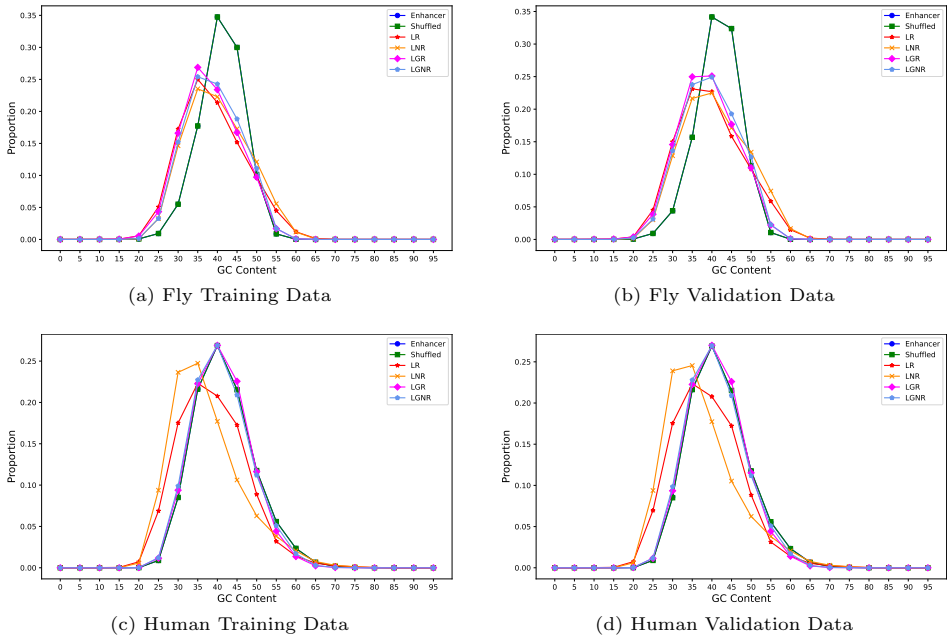
guanine-cytosine (GC) content. Based on these variables, we constructed the following datasets:

- Length Repeat (LR): Sequences in this control set were sampled to match the length distribution of the enhancer dataset, allowing for the presence of repetitive elements.
- Length No-Repeat (LNR): This dataset comprises sequences sampled to match the length distribution of the enhancer dataset, with the exclusion of repetitive elements.
- Length GC-Repeat (LGR): Sequences in this control set were sampled to match both the length and GC content distributions of the enhancer dataset, while permitting the inclusion of repetitive elements.
- Length GC-No-Repeat (LGNR): This dataset contains sequences sampled to match both the length and GC content distributions of the enhancer dataset, with the exclusion of repetitive elements.

In addition to these controlled datasets, we generated a fifth control set consisting of shuffled enhancer sequences. For each enhancer in our primary datasets, we generated shuffled sequences by randomly permuting its constituent k-mers (subsequences of length k), where k ranged from 1 to 6. This control aims to assess the model's capacity to distinguish true enhancers from sequences that merely share the same k-mer composition. Repetitive elements for both the human and *Drosophila melanogaster* genomes were identified using Red [61]. Control regions for both species were selected from their respective genomes, ensuring no overlap with the enhancer datasets. When generating control sequences for the human dataset, the length was precisely matched to 400 bp. For the *Drosophila melanogaster* dataset, control sequence lengths were randomly chosen from a uniform distribution between 460 and 500 bp for each corresponding enhancer, providing a broader range for scanning. A tolerance of 0.03 was permitted for the GC content of the selected control sequences in both the human and *Drosophila melanogaster* datasets.

4.4 Data Partitioning

Our enhancer and control datasets were split into training (70%), validation (20%), and testing (10%) datasets. Given the requirement for deep neural networks to be effectively trained with substantial amounts of data, we allocated 70% of our dataset for training purposes. An additional 20% of the data was designated as the validation dataset to detect and mitigate overfitting—the phenomenon where a model learns the training data too well but fails to generalize to new, unseen data (i.e., memorizing instead of learning). The remaining 10% of the data served as the testing dataset to ensure a comprehensive evaluation on previously unseen data. The data from both enhancers and controls in the human dataset were randomly allocated across these partitions. This random split ensures that each partition contains a representative mix of both enhancers and non-enhancers. However, for the fly dataset, a different approach was required. Since the enhancers for the fly were sampled from four different experiments there is potential overlap between the enhancers. If enhancers were split randomly, overlapping enhancers could appear across two or three partitions, leading to data leakage. As a remedy, sequences were sorted by chromosome and start position. After sorting, the data were split sequentially, ensuring that enhancers with



(a) Fly Training Data (b) Fly Validation Data

(c) Human Training Data (d) Human Validation Data

Fig. 8: Comparison of GC content in fly and human enhancer training and validation datasets. This figure illustrates the similarity in GC content distributions between control and enhancer datasets for both training and validation splits in human and fly. All four panels show closely aligned GC content distributions, demonstrating that the control sequences are well-matched to the enhancers in terms of GC content. Legend abbreviations: Enhancer is the true enhancer sequences. LR is control sequences matched for length with repeats allowed. LNR is length-matched controls with repeats removed. LGR is controls matched for length and GC content with repeats allowed. LGNR is length and GC-matched controls with repeats removed. Shuffled is enhancer sequences with permuted k-mer composition.

overlapping genomic coordinates were assigned to the same partition, thus eliminating potential data leakage between training, validation, and testing sets.

4.5 GC Distrubution

After partitioning the data, we ensured that the enhancer sequences and the control sequences were matched in terms of length and GC content. This prevents the model from oversimplifying the classification task by relying on differences in these basic sequence properties. As shown in Figure 8, the GC content distributions for both the fly and human datasets are well-matched between enhancers and controls. To quantify the similarity between the enhancer and control datasets, we used Kullback-Leibler (KL) divergence to compare the distributions of sequence length and GC content. A KL divergence of zero indicates identical distributions, with lower values reflecting greater similarity. For the fly datasets, we observed KL divergence values ranging from

0.1173 to 0.2216 for training and from 0.1272 to 0.2267 for validation. For the human datasets, values ranged from 0.0068 to 0.2233 for training and from 0.0066 to 0.2236 for validation, indicating strong similarity in both species. Shuffled control datasets yielded KL divergence values of 0.0 as they were generated directly from the enhancer sequences.

4.6 Data Indexing

To enable the neural network to learn a representation of each nucleotide, we assigned a unique integer index to each nucleotide. The specific values are unimportant to the network, as long as they are positive and unique. We modified the input sequences by assigning an integer to each nucleotide encountered in the genomic data. For example, the nucleotide A is assigned the index 2, with 0 and 1 reserved for padding and unknown bases, respectively. T is assigned the index 3, and so on, until every nucleotide has been mapped to a numerical value. Thus, a sequence such as ATAT is encoded as [2, 3, 2, 3], which can be used as input to the network.

4.7 Data Augmentation

To train the network effectively, we increased the amount of training data through a data augmentation strategy. We generated additional data by creating reverse complement counterparts of the original sequences and using them as input. This approach is valid because forward and reverse complement sequences are biologically equivalent. As a result, the model was exposed to more diverse training examples without requiring additional data collection. During training, each input sequence had a 50% chance of being replaced with its reverse complement. These reverse complements were indexed into numerical representations using the same scheme as the forward sequences. We applied an additional augmentation technique when preparing the fly dataset. To ensure that all sequences fell within a uniform length range (460–500 bp), we expanded or slightly trimmed sequences as needed. This was important as we planned to scan long sequences by dividing them into segments within this range. The procedure was the same for both expansion and trimming: we determined the midpoint of each sequence, then generated five sequence copies of randomly sampled lengths between 460 and 500 bp, all centered on the midpoint.

4.8 Convolutional Neural Network

For *EnhancerDetector*, the primary model used is a convolutional neural network. Convolutional neural networks are well-suited for processing data with a structured arrangement, such as DNA sequences, and are effective at identifying and learning hierarchical patterns. Our network (Figure 9) begins with an embedding layer, which learns numerical representations for nucleotides. The output of the embedding layer is passed to four custom convolutional blocks, each consisting of two one-dimensional convolutional layers followed by a batch normalization layer. The output of the final block is flattened and passed through two dense layers, with a batch normalization layer in between. The network ends with a single-neuron dense layer that outputs a probability score between 0 and 1, representing the likelihood that a given sequence

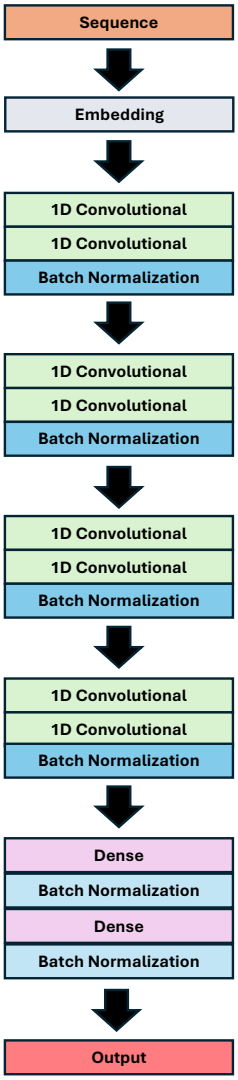


Fig. 9: An overview of the *EnhancerDetector* architecture. The network takes a 400 bp DNA sequence as input and passes it through an embedding layer, which learns a numerical representation for each nucleotide. The resulting sequence is then processed by four convolutional blocks, each consisting of two 1D convolutional layers (kernel size = 3), where the second uses a stride of 2 for downsampling. These are followed by batch normalization and a ReLU activation. The number of filters in each block increases with depth (64, 128, 256, and 512), allowing the network to extract hierarchical features of increasing complexity. Next, the output is flattened and passed through two dense layers (each with 20 units and ReLU activation), with a batch normalization layer in between, to classify whether a given sequence is an enhancer. The final output is a value between 0 and 1, where values above 0.5 indicate enhancers.

is an enhancer. Sequences with scores greater than or equal to 0.5 are classified as enhancers, while those below 0.5 are classified as non-enhancers. We optimized the following hyperparameters:

- Number of filters: Determines how many patterns each convolutional layer learns.
- Filter size: Specifies the width of the filters, e.g., a size of 3 corresponds to patterns spanning three consecutive positions.
- Number of dense neurons: Controls the number of neurons in the dense layers used for classification.

4.9 Recurrent Neural Network

Another model we considered was a recurrent neural network, which is particularly well-suited for sequential data, as it can capture temporal dependencies and patterns over time. This model uses a recurrent block consisting of a Gated Recurrent Unit (GRU) layer, followed by a batch normalization layer and an activation function. The architecture is similar to that of the convolutional neural network: it begins with an embedding layer, followed by two recurrent blocks, then two dense layers with a batch normalization layer in between. For this network, we optimized the following three hyperparameters: the number of GRU neurons, the activation function used in the recurrent block, and the number of dense neurons.

4.10 Evaluation Metrics

We evaluated the performance of *EnhancerDetector* and related tools using the following metrics (higher values indicate better performance across all metrics):

- Accuracy: Percentage of correctly predicted samples (enhancers or non-enhancers).
- Recall (Sensitivity): Percentage of true enhancers correctly identified.
- Specificity: Percentage of true non-enhancers correctly identified.
- Precision: Proportion of predicted enhancers that are correct. This is especially important for experimental validation in transgenic flies.
- F1 and Weighted F1 Scores: The F1 score is the harmonic mean of precision and recall. The weighted F1 score also incorporates precision and recall, but precision was weighted twice as much.

4.11 Guarding Against Overfitting

In addition to the data augmentation techniques described earlier, we applied early stopping to prevent overfitting—a situation where the model performs well on training data but poorly on unseen data. Early stopping is a regularization method that halts training when performance on validation data stops improving. Specifically, we monitored the weighted F1 score on the validation set after each epoch (one complete pass through the entire training dataset). Training continued as long as the score improved. If no improvement was observed for 25 epochs, training was stopped and the model with the highest validation weighted F1 score was selected.

4.12 Optimization

The models were evaluated using various combinations of hyperparameters to identify the optimal configuration. The following hyperparameter values were tested: number of filters (32, 40, 48, 64, and 72), filter size (3 and 5), and number of dense neurons (10, 20, and 40). The best configuration for the human enhancer classifier was a combination of 64 filters, a filter size of 3, and 20 dense neurons. Similarly, we identified two best networks for the fly enhancer classifier, which were optimized with combinations of 32 and 40 filters, a filter size of 3, and 20 dense neurons. The best models were primarily chosen based on a high weighted F1 score, due to the imbalance between confirmed enhancers and control data. Specificity was also considered to ensure the model does not incorrectly identify non-enhancers, and recall was assessed to verify the model's ability to correctly identify enhancers.

4.13 Fine-Tuning

We observed that the human model was trained on a comparatively larger dataset than the fly model (8,954,140 vs. 6,473,088), allowing it to learn certain characteristics and patterns more effectively. We therefore decided to adapt our trained human classifier to the fly dataset using a fine-tuning approach. The rationale was to adjust the model to optimize its performance on a new but related dataset. To fine-tune the classifier on the fly data, we employed a strategy of freezing and unfreezing layers during training. Only the unfrozen layers were updated with the new data, enabling the model to adapt while preserving knowledge gained from the human data. This approach allows us to retain useful features learned from the human dataset while tailoring the model to the specific characteristics of the fly data.

4.14 Ensemble

After training the fly networks and fine-tuning the model, we created an ensemble of these three networks. An ensemble combines multiple neural network models, each independently tasked with determining whether a given genomic region is an enhancer. The ensemble aggregates the predictions from all three models: if all agree that a region is an enhancer, it is classified as such; otherwise, it is not. This approach provides a stricter criterion for identifying enhancer regions, reducing the likelihood of false positives.

4.15 Comparison to Related Tools

To further validate *EnhancerDetector*, we compared its performance with that of other enhancer detection tools designed for binary classification. One such tool is Enhancer-FRL [62], which predicts human enhancers of 200 bp in length using feature representation learning and multi-view sequence information. We evaluated Enhancer-FRL using two datasets: the ENCODE Human dataset (obtained from the UCSC Genome Table Browser) and the ENCODE Mouse dataset. Both datasets were preprocessed as described earlier, including the removal of promoters, exons, and insulators. For the Enhancer-FRL evaluation, we selected potential enhancers that were 200 bp

or shorter. Regions shorter than 200 bp were symmetrically extended in both directions to reach exactly 200 bp in length. This same procedure was applied to mouse enhancers. Because our human network expects input sequences of 400 bp, we further extended the 200-bp enhancers to 400 bp. To ensure a fair comparison, we excluded any enhancer region that overlapped by at least 50% with our CATlas training enhancer dataset. This filtering ensured that none of the evaluated enhancers had been seen during training. For both the 200-bp and 400-bp enhancers in human and mouse datasets, we also generated matched control sequences (shuffled, LR, LNR, LGR, and LGNR). To ensure realistic and biologically relevant benchmarking, all model evaluations were conducted using testing datasets with a 1:10 enhancer-to-control ratio for both the human and mouse ENCODE datasets. This ratio reflects the natural sparsity of enhancer elements in the genome and more accurately simulates genome-wide classification scenarios. Enhancer-FRL was run using its default parameters.

Another tool we compared *EnhancerDetector* to was Enhancer-LSTMAtt [63]. This method detects 200-bp enhancers using a deep neural network that incorporates bidirectional Long Short-Term Memory (LSTM) layers and an attention mechanism. Enhancer-LSTMAtt was evaluated using the same 200-bp human and mouse ENCODE datasets, along with their corresponding control datasets. The model was executed using its default parameters.

A third tool we evaluated was the PDCNN model [64]. PDCNN is a convolutional neural network specifically designed for DNA sequence analysis. Unlike the other tools, PDCNN was trained on both human and mouse enhancers. The authors used sequences ranging from 50 bp to 300 bp for testing and recommended using 200-bp sequences for optimal performance. Accordingly, we evaluated PDCNN using the same 200-bp human and mouse enhancer datasets, along with their corresponding control datasets. PDCNN was run using its default parameters.

The last tool we assessed was gkm-SVM [40]. This sequence-based classifier uses gapped k-mer features and support vector machines to predict functional regulatory elements in genomic sequences. By allowing gaps within k-mers, gkm-SVM captures more flexible and informative sequence patterns compared to traditional k-mer methods. More specifically, we used LS-GKM [41], an optimized version of gkm-SVM that supports training on large-scale datasets (up to 100,000 sequences). Unlike the other tools we tested, LS-GKM uses a support vector machine instead of neural networks to detect enhancers. Additionally, this tool can be trained on a user-provided dataset, unlike the other three tools, which are distributed with pre-trained models. To train it, we randomly selected 50,000 positive sequences from the CATlas training dataset. For negative sequences, we randomly selected 10,000 sequences from each of the shuffled, LR, LNR, LGR, and LGNR training datasets, resulting in a total of 50,000 negative sequences. These 100,000 sequences were then used to train LS-GKM using its default parameters. We subsequently evaluated its performance on our testing dataset and compared the results to those of our human network. This comparison was fair, as both *EnhancerDetector* and LS-GKM were trained on the same dataset and evaluated on the same unseen testing sequences. The LS-GKM model trained on CATlas sequences was then evaluated on the 400-bp-long human and mouse ENCODE enhancers.

4.16 Reporter Gene Analysis

Putative enhancer sequences were synthesized as gBlocks (IDT, Coralville, IA, USA) and cloned into the plasmid pattBnucGFPs (details available upon request). Transgenic flies were generated by insertion into the third chromosome attP2 site (Rainbow Transgenics, Camarillo, CA, USA). For embryo analysis, homozygous transgenic embryos were collected, fixed, and stained using standard protocols. Larval expression was assessed by direct visualization of GFP following larval dissection.

Declarations

Funding

Research reported in this publication was supported by the National Human Genome Research Institute of the National Institutes of Health under award number: R21HG011507. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Conflict of interest/Competing interests

None declared.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Acknowledgments

We would like to thank Jack Leatherbarrow and James Weiser for their help with the experimental validations.

Data availability

Supplementary File 1: BLAST output of the true positive mouse enhancers versus the human genome.

Materials availability

Not applicable.

Code availability

The code is freely available for academic use through github.com/BioinformaticsToolsmith/EnhancerDetector and doi.org/10.5281/zenodo.15531292 under the Affero General Public License version 1. Restrictions on Use for Profit or Non-Academics: An alternative commercial license is required.

Author contribution

L. M. S. developed the code, analyzed the results, and wrote the manuscript. G. S.-L. performed the in vivo validation experiments. M. S. H. conceived the original idea, analyzed the results, interpreted the in vivo validation experiments, wrote, and reviewed the manuscript. H. Z. G. conceived the original idea, supervised the project, developed the code, analyzed the results, wrote, and reviewed the manuscript.

References

[1] Zaugg, J.B., Sahlén, P., Andersson, R., Alberich-Jorda, M., Laat, W., Deplancke, B., Ferrer, J., Mandrup, S., Natoli, G., Plewczynski, D., Rada-Iglesias, A., Spicuglia, S.: Current challenges in understanding the role of enhancers in disease. Nat Struct Mol Biol **29**(12), 1148–1158 (2022)

[2] Hariprakash, J.M., Ferrari, F.: Computational biology solutions to identify enhancers-target gene pairs. Comput Struct Biotechnol J **17**, 821–831 (2019)

[3] Zabidi, M.A., Stark, A.: Regulatory enhancer-core-promoter communication via transcription factors and cofactors. Trends Genet **32**(12) (2016)

[4] Suryamohan, K., Halfon, M.S.: Identifying transcriptional cis-regulatory modules in animal genomes. Wiley Interdiscip Rev Dev Biol **4**(2), 59–84 (2015)

[5] Hiromi, Y., Gehring, W.J.: Regulation and function of the *Drosophila* segmentation gene *fushi tarazu*. Cell **43**, 603–613 (1987)

[6] Goto, T., Macdonald, P., Maniatis, T.: Early and late periodic patterns of even skipped expression are controlled by distinct regulatory elements that respond to different spatial cues. Cell **57**(3), 413–422 (1989)

[7] Harding, K., Hoey, T., Warrior, R., Levine, M.: Autoregulatory and gap gene response elements of the even-skipped promoter of *Drosophila*. EMBO J **8**, 1205–1212 (1989)

[8] Boyle, A.P., Davis, S., Shulha, H.P., Meltzer, P., Margulies, E.H., Weng, Z., Sunyaev, S.R., Bernstein, I.D., Crawford, G.E., Furey, T.S.: High-resolution mapping and characterization of open chromatin across the genome. Cell **132**(2), 311–322 (2008)

[9] Park, P.J.: ChIP–seq: advantages and challenges of a maturing technology. Nature Rev Genet **10**(10), 669–680 (2009)

[10] Visel, A., Blow, M.J., Li, Z., Zhang, T., al.: ChIP-seq accurately predicts tissue-specific activity of enhancers. Nature **457**(7231), 854–858 (2009)

[11] Kim, T.-K., Hemberg, M., Gray, J.M., al.: Widespread transcription at neuronal activity-regulated enhancers. Nature **465**(7295), 182–187 (2010)

[12] Giresi, P.G., Kim, J., McDaniell, R.M., Iyer, V.R., Lieb, J.D.: Faire (formaldehyde-assisted isolation of regulatory elements) isolates active regulatory elements from human chromatin. Genome Res **17**(6), 877–885 (2007)

[13] Belton, J.-M., McCord, R.P., Gibcus, J.H., Naumova, N., Zhan, Y., Dekker, J.: Hi–c: A comprehensive technique to capture the conformation of genomes. Methods **58**(3), 268–276 (2012)

[14] Jenett, A., Rubin, G.M., Ngo, T.-T.B., Shepherd, D., Murphy, C., Dionne, H., Pfeiffer, B.D., Cavallaro, A., Hall, D., Jeter, J., Iyer, N., Fetter, D., Hausenfluck, J.H., Peng, H., Trautman, E.T., Svirskas, R.R., Myers, E.W., Iwinski, Z.R., Aso, Y., DePasquale, G.M., Enos, A., Hulamm, P., Lam, S.C.B., Li, H.-H., Laverty, T.R., Long, F., Qu, L., Murphy, S.D., Rokicki, K., Safford, T., Shaw, K., Simpson, J.H., Sowell, A., Tae, S., Yu, Y., Zugates, C.T.: A GAL4-driver line resource for *Drosophila* neurobiology. Cell Rep **2**(4), 991–1001 (2012)

[15] Arnold, C.D., Gerlach, D., Stelzer, C., Boryń, Ł.M., Rath, M., Stark, A.: Genome-wide quantitative enhancer activity maps identified by STARR-seq. Science **339**(6123), 1074–1077 (2013)

[16] Buenrostro, J.D., Wu, B., Chang, H.Y., Greenleaf, W.J.: Atac-seq: A method for assaying chromatin accessibility genome-wide. Curr Protoc Mol Biol **109**(1), 21–29121299 (2015)

[17] Tokusumi, T., Tokusumi, Y., Brahier, M.S., al.: Screening and analysis of Janelia FlyLight Project enhancer-gal4 strains identifies multiple gene enhancers active during hematopoiesis in normal and wasp-challenged *Drosophila* larvae. G3 **7**(2), 437–448 (2017)

[18] Catarino, R.R., Stark, A.: Assessing sufficiency and necessity of enhancer activities for gene expression and the mechanisms of transcription activation. Genes Dev **32**(3-4), 202–223 (2018)

[19] Klemm, S.L., Shipony, Z., Greenleaf, W.J.: Chromatin accessibility and the regulatory epigenome. Nat Rev Genet **20**(4), 207–220 (2019)

[20] Lewin, H.A., Robinson, G.E., Kress, W.J., Baker, W.J., Coddington, J., Crandall, K.A., Durbin, R., Edwards, S.V., Forest, F., Gilbert, M.T.P., Goldstein, M.M., Grigoriev, I.V., Hackett, K.J., Haussler, D., Jarvis, E.D., Johnson, W.E., Patrinos, A., Richards, S., Castilla-Rubio, J.C., Sluys, M.-A., Soltis, P.S., Xu, X., Yang, H., Zhang, G.: Earth BioGenome Project: Sequencing life for the future of life. Proc Nat Acad Sci USA **115**(17), 4325–4333 (2018)

[21] Rhie, A., McCarthy, S.A., Fedrigo, O., Damas, J., Formenti, G., Koren, S., Uliano-Silva, M., Chow, W., Fungtammasan, A., Kim, J., Lee, C., Ko, B.J., Chaisson, M., Gedman, G.L., Cantin, L.J., Thibaud-Nissen, F., Haggerty, L., Bista, I., Smith, M., Haase, B., Mountcastle, J., Winkler, S., Paez, S., Howard, J., Vernes, S.C.,

Lama, T.M., Grutzner, F., Warren, W.C., Balakrishnan, C.N., Burt, D., George, J.M., Biegler, M.T., Iorns, D., Digby, A., Eason, D., Robertson, B., Edwards, T., Wilkinson, M., Turner, G., Meyer, A., Kautt, A.F., Franchini, P., Detrich, H.W., Svardal, H., Wagner, M., Naylor, G.J.P., Pippel, M., Malinsky, M., Mooney, M., Simbirsky, M., Hannigan, B.T., Pesout, T., Houck, M., Misuraca, A., Kingan, S.B., Hall, R., Kronenberg, Z., Sović, I., Dunn, C., Ning, Z., Hastie, A., Lee, J., Selvaraj, S., Green, R.E., Putnam, N.H., Gut, I., Ghurye, J., Garrison, E., Sims, Y., Collins, J., Pelan, S., Torrance, J., Tracey, A., Wood, J., Dagnew, R.E., Guan, D., London, S.E., Clayton, D.F., Mello, C.V., Friedrich, S.R., Lovell, P.V., Osipova, E., Al-Ajli, F.O., Secomandi, S., Kim, H., Theofanopoulou, C., Hiller, M., Zhou, Y., Harris, R.S., Makova, K.D., Medvedev, P., Hoffman, J., Masterson, P., Clark, K., Martin, F., Howe, K., Flicek, P., Walenz, B.P., Kwak, W., Clawson, H., Diekhans, M., Nassar, L., Paten, B., Kraus, R.H.S., Crawford, A.J., Gilbert, M.T.P., Zhang, G., Venkatesh, B., Murphy, R.W., Koepfli, K.-P., Shapiro, B., Johnson, W.E., Di Palma, F., Marques-Bonet, T., Teeling, E.C., Warnow, T., Graves, J.M., Ryder, O.A., Haussler, D., O'Brien, S.J., Korlach, J., Lewin, H.A., Howe, K., Myers, E.W., Durbin, R., Phillippy, A.M., Jarvis, E.D.: Towards complete and error-free genome assemblies of all vertebrate species. Nature **592**(7856), 737–746 (2021)

[22] Mc Cartney, A.M., Formenti, G., Mouton, A., De Panis, D., Marins, L.S., Leitão, H.G., Diedericks, G., Kirangwa, J., Morselli, M., Salces-Ortiz, J., Escudero, N., Iannucci, A., Natali, C., Svardal, H., Fernández, R., De Pooter, T., Joris, G., Strazisar, M., Wood, J.M.D., Herron, K.E., Seehausen, O., Watts, P.C., Shaw, F., Davey, R.P., Minotto, A., Fernández, J., Böhne, A., Alegria, C., Alioto, T., Alves, P.C., Amorim, I.R., Aury, J.-M., Backstrom, N., Baldrian, P., Baltrunaite, L., Barta, E., BedHom, B., Belser, C., Bergsten, J., Bertrand, L., Bilandija, H., Binzer-Panchal, M., Bista, I., Blaxter, M., Borges, P.A.V., Dias, G.B., Bosse, M., Brown, T., Bruggmann, R., Buena-Atienza, E., Burgin, J., Buzan, E., Cariani, A., Casadei, N., Chiara, M., Chozas, S., Čiampor, F., Crottini, A., Cruaud, C., Cruz, F., Dalen, L., De Biase, A., Campo, J., Delic, T., Dennis, A.B., Derks, M.F.L., Diroma, M.A., Djan, M., Duprat, S., Eleftheriadi, K., Feulner, P.G.D., Flot, J.-F., Forni, G., Fosso, B., Fournier, P., Fournier-Chambrillon, C., Gabaldon, T., Garg, S., Gissi, C., Giupponi, L., Gomez-Garrido, J., González, J., Grilo, M.L., Grüning, B., Guerin, T., Guiglielmoni, N., Gut, M., Haesler, M.P., Hahn, C., Halpern, B., Harrison, P.W., Heintz, J., Hindrikson, M., Höglund, J., Howe, K., Hughes, G.M., Istace, B., Cock, M.J., Janžekovič, F., Jonsson, Z.O., Joye-Dind, S., Koskimäki, J.J., Krystufek, B., Kubacka, J., Kuhl, H., Kusza, S., Labadie, K., Lähteenaro, M., Lantz, H., Lavrinienko, A., Leclère, L., Lopes, R.J., Madsen, O., Magdelenat, G., Magoga, G., Manousaki, T., Mappes, T., Marques, J.P., Redondo, G.I.M., Maumus, F., McCarthy, S.A., Megens, H.-J., Melo-Ferreira, J., Mendes, S.L., Montagna, M., Moreno, J., Mosbech, M.-B., Moura, M., Musilova, Z., Myers, E., Nash, W.J., Nater, A., Nicholson, P., Niell, M., Nijland, R., Noel, B., Noren, K., Oliveira, P.H., Olsen, R.-A., Ometto, L., Oomen, R.A., Ossowski, S., Palinauskas, V., Palsson, S., Panibe, J.P., Pauperio, J., Pavlek, M., Payen, E., Pawlowska, J., Pellicer, J., Pesole, G., Pimenta, J., Pippel, M., Pirttilä, A.M.,

Poulakakis, N., Rajan, J., M. C. Rego, R., Resendes, R., Resl, P., Riesgo, A., Rodin-Morch, P., Soares, A.E.R., Fernandes, C.R., Romeiras, M.M., Roxo, G., Rüber, L., Ruiz-Lopez, M.J., Saarma, U., Silva, L.P., Sim-Sim, M., Soler, L., Sousa, V.C., Santos, C.S., Spada, A., Stefanovic, M., Steger, V., Stiller, J., Stöck, M., Struck, T.H., Sudasinghe, H., Tapanainen, R., Tellgren-Roth, C., Trindade, H., Tukalenko, Y., Urso, I., Vacherie, B., Van Belleghem, S.M., Van Oers, K., Vargas-Chavez, C., Velickovic, N., Vella, N., Vella, A., Vernesi, C., Vicente, S., Villa, S., Pettersson, O.V., Volckaert, F.A.M., Voros, J., Wincker, P., Winkler, S., Ciofi, C., Waterhouse, R.M., Mazzoni, C.J.: The European Reference Genome Atlas: piloting a decentralised approach to equitable biodiversity genomics. npj Biodivers **3**(1), 28 (2024)

[23] Sunagawa, S., Acinas, S.G., Bork, P., Bowler, C., Acinas, S.G., Babin, M., Boss, E., Cochrane, G., Vargas, C., Follows, M., Gorsky, G., Grimsley, N., Guidi, L., Hingamp, P., Iudicone, D., Jaillon, O., Kandels, S., Karp-Boss, L., Karsenti, E., Lescot, M., Not, F., Ogata, H., Pesant, S., Poulton, N., Raes, J., Sardet, C., Sieracki, M., Speich, S., Stemmann, L., Sullivan, M.B., Wincker, P., Eveillard, D., Lombard, F., Pesant, S., Sullivan, M.B., Coordinators, T.O.: Tara Oceans: towards global ocean ecosystems biology. Nat Rev Microbiol **18**(8), 428–445 (2020)

[24] Chen, L., Fish, A.E., Capra, J.A.: Prediction of gene regulatory enhancers across species reveals evolutionarily conserved sequence properties. PLoS Comput Biol **14**(10), 1006484 (2018)

[25] Kulkarni, M.M., Arnosti, D.N.: Information display by transcriptional enhancers. Development **130**(26), 6569–6575 (2003)

[26] Spitz, F., Furlong, E.E.M.: Transcription factors: from enhancer binding to developmental control. Nat Rev Genet **13**(9), 613–626 (2012)

[27] Noyes, M.B., Christensen, R.G., Wakabayashi, A., Stormo, G.D., Brodsky, M.H., Wolfe, S.A.: Analysis of homeodomain specificities allows the family-wide prediction of preferred recognition sites. Cell **133**(7), 1277–1289 (2008)

[28] Colbran, L.L., Chen, L., Capra, J.A.: Sequence characteristics distinguish transcribed enhancers from promoters and predict their breadth of activity. Genetics **211**(4), 1205–1217 (2019)

[29] Jindal, G.A., Farley, E.K.: Enhancer grammar in development, evolution, and disease: dependencies and interplay. Dev Cell **56**(5), 575–587 (2021)

[30] Asma, H., Halfon, M.S.: Computational enhancer prediction: evaluation and improvements. BMC Bioinformatics **20**(1), 174 (2019)

[31] Girgis, H.Z., Ovcharenko, I.: Predicting tissue specific cis-regulatory modules in the human genome using pairs of co-occurring motifs. BMC Bioinformatics **13**(1),

25 (2012)

[32] Liu, B., Fang, L., Long, R., Lan, X., Chou, K.-C.: iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. Bioinformatics **32**(3), 362–369 (2015)

[33] Jia, C., He, W.: Enhancerpred: a predictor for discovering enhancers based on the combination and selection of multiple features. Sci Rep **6**(1), 38741 (2016)

[34] Liu, B., Li, K., Huang, D.-S., Chou, K.-C.: iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach. Bioinformatics **34**(22), 3835–3842 (2018)

[35] Asma, H., Liu, L., Halfon, M.S.: SCRMshaw: Supervised cis-regulatory module prediction for insect genomes. PLoS One **19**(12), 1–8 (2024)

[36] Asim, M.N., Ibrahim, M.A., Malik, M.I., Dengel, A., Ahmed, S.: Enhancer-DSNet: A supervisedly prepared enriched sequence representation for the identification of enhancers and their strength. In: Yang, H., Pasupa, K., Leung, A.C.-S., Kwok, J.T., Chan, J.H., King, I. (eds.) Proc. Neurips, pp. 38–48. Springer, Cham (2020)

[37] Yang, R., Wu, F., Zhang, C., Zhang, L.: iEnhancer-GAN: A deep learning framework in combination with word embedding and sequence generative adversarial net to identify enhancers and their strength. Int J Mol Sci **22**(7) (2021)

[38] Liao, M., Zhao, J.-p., Tian, J., Zheng, C.-H.: ienhancer-dcla: using the original sequence to identify enhancers and their strength based on a deep learning framework. BMC Bioinformatics **23**(1), 480 (2022)

[39] Mehmood, F., Arshad, S., Shoaib, M.: ADH-Enhancer: an attention-based deep hybrid framework for enhancer identification and strength prediction. Brief Bioinform **25**(2), 030 (2024)

[40] Ghandi, M., Lee, D., Mohammad-Noori, M., Beer, M.A.: Enhanced regulatory sequence prediction using gapped k-mer features. PLoS Comput Biol **10**(7), 1003711 (2014)

[41] Lee, D.: Ls-gkm: A new gkm-svm for large-scale datasets. Bioinformatics **32**(14), 2196–2198 (2016)

[42] Zhou, J., Troyanskaya, O.G.: Predicting effects of noncoding variants with deep learning–based sequence model. Nat Methods **12**(10), 931–934 (2015)

[43] Lee, D., Gorkin, D.U., Baker, M., Strober, B.J., Asoni, A.L., McCallion, A.S., Beer, M.A.: A method to predict the impact of regulatory variants from DNA sequence. Nat Genet **47**(8), 955–961 (2015)

[44] Avsec, Agarwal, V., Visentin, D., Ledsam, J.R., Grabska-Barwinska, A., Taylor,

K.R., Assael, Y., Jumper, J., Kohli, P., Kelley, D.R.: Effective gene expression prediction from sequence by integrating long-range interactions. Nat Methods **18**(10), 1196–1203 (2021)

[45] Winter, S.D., Konstantakos, V., Aerts, S.: Modelling and design of transcriptional enhancers. Nat Rev Bioeng **3**(5), 374–389 (2025)

[46] Genome 10K Community of Scientists: Genome 10k: A proposal to obtain whole-genome sequence for 10000 vertebrate species. J Hered **100**(6), 659–674 (2009)

[47] Li, L., Zhu, Q., He, X., Sinha, S., Halfon, M.S.: Large-scale analysis of transcriptional cis-regulatory modules reveals both common features and distinct subclasses. Genome Biol **8**(6), 101 (2007)

[48] Muratoglu, S., Garratt, B., Hyman, K., Gajewski, K., Schulz, R.A., Fossett, N.: Regulation of *Drosophila* friend of GATA gene, u-shaped, during hematopoiesis: A direct role for Serpent and Lozenge. Dev Biol **296**(2), 561–579 (2006)

[49] Fossett, N., Zhang, Q., Gajewski, K., Choi, C.Y., Kim, Y., Schulz, R.A.: The multitype zinc-finger protein U-shaped functions in heart cell specification in the *Drosophila* embryo. Proc Nat Acad Sci USA **97**(13), 7348–7353 (2000)

[50] Zhu, Q., Halfon, M.S.: Vector-dependent gene expression driven by insulated p-element reporter vectors. Fly **1**(1), 55–56 (2007)

[51] Agrawal, S., Kanamaru, E., Saito, Y., Ishikawa, F., Hoon, M.: Cell type-dependent directional transcription at enhancers. NAR Genom Bioinform **7**(1), 007 (2025)

[52] Grossman, S.R., Zhang, X., Wang, L., Engreitz, J., Melnikov, A., Rogov, P., Tewhey, R., Isakova, A., Deplancke, B., Bernstein, B.E., Mikkelsen, T.S., Lander, E.S.: Systematic dissection of genomic features determining transcription factor binding and enhancer function. Proc Nat Acad Sci USA **114**(7), 1291–1300 (2017)

[53] Zhou, J., Theesfeld, C.L., Yao, K., Chen, K.M., Wong, A.K., Troyanskaya, O.G.: Deep learning sequence-based *ab initio* prediction of variant effects on expression and disease risk. Nat Genet **50**(8), 1171–1179 (2018)

[54] Girgis, H.Z., Velasco, A., Reyes, Z.E.: HebbPlot: an intelligent tool for learning and visualizing chromatin mark signatures. BMC Bioinformatics **19**(1), 310 (2018)

[55] Thomas, S., Li, X.-Y., Sabo, P.J., al.: Dynamic reprogramming of chromatin accessibility during *Drosophila* embryo development. Genome Biol **12**(5), 43 (2011)

[56] Cusanovich, D.A., Reddington, J.P., Garfield, D.A., Daza, R.M., Aghamirzaie, D.,

Marco-Ferreres, R., Pliner, H.A., Christiansen, L., Qiu, X., Steemers, F.J., Trapnell, C., Shendure, J., Furlong, E.E.M.: The cis-regulatory dynamics of embryonic development at single-cell resolution. Nature **555**(7697), 538–542 (2018)

[57] Keränen, S.V.E., Rivera, J., Gallo, S.M., Halfon, M.S.: Redfly: the transcriptional regulatory element database for *Drosophila*. Nucleic Acids Res **47**(D1), 828–834 (2019)

[58] Négre, N., Brown, C.D., Shah, P.K., Kheradpour, P., Morrison, C.A., Henikoff, J.G., Feng, X., Ahmad, K., Russell, S., White, R.A.H., Stein, L., Henikoff, S., Kellis, M., White, K.P.: A comprehensive map of insulator elements for the drosophila genome. PLoS Genet **6**(1), 1–13 (2010)

[59] Quinlan, A.R., Hall, I.M.: Bedtools: a flexible suite of utilities for comparing genomic features. Bioinformatics **26**(6), 841–842 (2010)

[60] Hinrichs, A.S., Karolchik, D., Baertsch, R., Barber, G.P., Bejerano, G., Clawson, H., Diekhans, M., Furey, T.S., Harte, R.A., Hsu, F., *et al.*: The UCSC genome browser database: update 2006. Nucleic Acids Res **34**(Database issue), 590–598 (2006)

[61] Girgis, H.Z.: Red: an intelligent, rapid, accurate tool for detecting repeats de-novo on the genomic scale. BMC Bioinformatics **16**(1), 227 (2015)

[62] Wang, C., Zou, Q., Ju, Y., Shi, H.: Enhancer-frl: Improved and robust identification of enhancers and their activities using feature representation learning. IEEE/ACM Trans Comput Biol Bioinform **20**(2), 967–975 (2023)

[63] Huang, G., Luo, W., Zhang, G., Zheng, P., Yao, Y., Lyu, J., Liu, Y., Wei, D.-Q.: Enhancer-lstmatt: A bi-lstm and attention-based deep learning method for enhancer recognition. Biomolecules **12**(7) (2022)

[64] Hu, W., Li, Y., Wu, Y., Guan, L., Li, M.: A deep learning model for dna enhancer prediction based on nucleotide position aware feature encoding. iScience **27**(6), 110030 (2024)