Exploring the Representational Power of Genomic Deep Learning Models

ZIQI AMBER TANG

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

School of Biological Sciences
Cold Spring Harbor Laboratory

April 3. 2024

“*Any sufficiently advanced technology is indistinguishable from magic.*”

Arthur C. Clarke

Acknowledgements

First and foremost I want to thank my advisor, Peter Koo, for his support throughout my Ph.D. time. None of this work would have been possible without his guidance. I'm grateful for his faith in me even when I had no experience in this field, and all the mentorship he provided to help me grow as a scientist. He has also showed me the influence a supportive and empathetic mentor can have beyond research.

I would also like to thank my committee members, Jesse Gillis, Adrian Krainer, Justin Kinney, and Alea Mills, for the advice they have provided during all the committee meetings. All of your insights has encouraged me to think about science from different perspectives, and helped me to improve my research. Also thanks to everyone at the CSHL School of Biological Sciences, Alyson Kass-Eisler, Kimberly Creteur, Catherine Perez, Brianna Campmier, Monn Monn Myat, Zach Lippman and Alex Gann. For always being supportive and caring.

Many thanks to the lab members in the Koo Lab. I feel lucky to have the chance of working with everyone. Through my time I have learnt and benefited from all of you. Also to everyone in Koch, for making this wonderful community.

Special thanks to all of my friends for going through the highs and lows with me. Your support, advice and the time we spent together are so valuable for me. I can't convey how grateful I am to have all of you in my life. Lastly to my family, for your belief in me and for never letting me doubt myself.

Contents

List of Figures

List of Tables

List of Abbreviations

CNN	**C**onvolutional **N**eural **N**etwork
CREs	**C**is-**R**egulatory **E**lements
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
GIA	**G**lobal **I**mportance **A**nalysis
gLM	**g**enomic **L**anguage **M**odel
LLM	**L**arge **L**anguage **M**odel
MPRA	**M**assively **P**arallel **R**eporter **A**ssays
PWM	**P**osition **W**eight **M**atrix
SNP	**S**ingle **N**ucleotide **P**olymorphisms
TF	**T**ranscription **F**actors

Chapter 1

Introduction to Deep Learning and genomic DL models

Cis-regulatory codes are fundamental to the regulation of gene expression, representing a significant portion of the functional complexity in biological systems. Understanding the roles and interactions of CREs remains a highly active field of research, crucial for exploring the mechanisms underlying biological functions. Their inherent complexity makes CREs a prime subject for the application of DL methods. This chapter establishes a background for the development of DL techniques and their applications to regulatory genomic functions. The chapter begins by introducing the function and complexity of cis-regulatory codes (Chapter 1.1) and the various DL related methods that's been applied to study them (Chapter 1.2, 1.3). Then we discuss the technical innovations in DL the enabled these developments (Chapter 1.4).

1.1 Cis-regulatory code

The human body is composed of around 37 trillion cells (Chen et al., 2022c; Panigrahi and O'Malley, 2021) that are vastly different in morphology and function but largely contain identical genetic information. Differential gene expression is one of the fundamental processes underlying the formation of different cell types during embryogenesis through precise spatio-temporal expression of genes (Panigrahi and O'Malley, 2021; Ong and Corces, 2011). Moreover, dysregulation of gene expression can lead to cells not carrying on their normal function at the right place and at the right time (Jindal and Farley, 2021). This can be caused by various sources of mutations in the non-coding, regulatory genome. For instance, the majority of disease-causing SNPs are mapped to non-coding genome (Wong et al., 2021). Therefore, understanding gene expression regulation is fundamental to the study of normal development and disease biology.

Cells achieve precise activation and repression of genes specific to a cell type through *cis*- and *trans*-regulatory elements (Wittkopp, Haerum, and Clark, 2004). CREs are non-coding sequences that activate or repress transcription from a gene. CREs regulate gene expression by binding of *trans*-regulatory factors (Wittkopp, Haerum, and Clark, 2004), such as TFs. TFs are proteins that contain a DNA binding domain that recognizes and binds to specific sequence elements or sequence motifs within CREs, leading to activation of transcription through various mechanisms. Factors other than sequence motif, e.g. motif flanking nucleotides, nucleosomal context of the DNA, presence of cofactors and co-binding TFs can affect TF binding (Inukai, Kock, and Bulyk, 2017).

The main types of CREs are promoters and enhancers, which activate gene expression and silencers which downregulate expression (Panigrahi and O'Malley, 2021). A 72bp region in Simian Virus 40 (SV40) genome was the first element demonstrated to increase gene expression levels of a given promoter (β globin gene) independently from its orientation and genomic distance to the promoter (Banerji, Rusconi, and Schaffner, 1981). In the original manuscript detailing this, the authors refer to the newly discovered element as an enhancer "for convenience". Subsequently, stemming from this, the classical definition of an enhancer focuses on sufficiency to activate transcription as well as orientation and distance independence of an element.

Since then, the study of enhancers has resulted in various further definitions. These are fundamentally different because of the methodology used to identify enhancers (Gasperini, Tome, and Shendure, 2020). For instance, a common method of defining enhancers is biochemical characterization through various assays that probe the chromatin state (Gasperini, Tome, and Shendure, 2020). High-throughput chromatin profiling assays such as DNase I hypersensitive site sequencing (DNase-seq) (Song and Crawford, 2010) can be used to define accessible or DNase hypersensitive sites (DHSs) in the genome. Active TFBSs are more likely to be within those regions and, therefore, DHS regions are often tagged as putative CREs. Another common method to characterize candidate regulatory elements is the histone code (Heintzman et al., 2009)– the set of post-translational modifications of histone proteins that can change the biophysical state of the chromatin. These can be measured using histone Chromatin immunoprecipitation followed by sequencing (ChIP-seq) experiments (Ren et al., 2000). For instance, H3K27ac is a common mark of enhancer regions. Together, such assays describe the state of the chromatin at a given gene but do not directly characterize the sequence determinants of these functional elements. Moreover, these methods only identify candidate regulatory elements and do not link them to the genes they target.

Another common strategy is identification of elements through perturbation experiments, i.e., measuring gene expression changes upon modifications to the DNA sequence. Certain kinds of MPRAs (Kinney et al., 2010; Levo and Segal, 2014), e.g. self-transcribing active regulatory region sequencing (STARR-seq) (Arnold et al., 2013), allow testing if an element is sufficient to activate transcription from a minimal promoter (similar to the classical definition). Others approaches, such as clustered regularly interspaced short palindromic repeats or CRISPR interference coupled with dead Cas9 (Diao et al., 2016; Fulco et al., 2016; Gasperini et al., 2017) can epigenetically inactivate a specific element (of around 1000 basepair (bp) length) and measure any changes in gene expression profile. In contrast to the MPRA experiments, this approach essentially measures if an enhancer is necessary for the expression of a given gene.

All of these methods provide valuable information about the complicated set of elements that orchestrate gene expression. However, they target different biological questions and unsurprisingly identify different and largely non-overlapping subsets of elements as enhancers. Moreover, only some of them can link an enhancer to its target gene in 'native' conditions – i.e. within the genome. For instance, although assays such as ATAC-seq or Histone ChIP-seq provide valuable insights into the state of the chromatin, and where the regions with higher activity are, they do not identify enhancer-promoter pairs. Assays such as STARR-seq do provide such a mapping, i.e. which enhancers can activate a given promoter. However, this is done in an episomal assay, and therefore the contribution of factors such as distance between the candidate enhancer and a promoter or the influence of other CREs in the genomic context is not taken into account.

CREs can be kilobases away from the target promoter through looping and

3D structures in the genome therefore making the search space for putative regulatory elements larger (Fulco et al., 2019). Additionally, they can act in a quantitative way (Kim and Wysocka, 2023; Zeitlinger, 2020) instead of simply turning promoters on or off, leading to a dynamic range of activities. Finally, for the majority of genes there are multiple elements acting within the 'neighborhood' of a gene and it is unclear how they interact in order to drive promoter activity (Lin et al., 2022a; Jin et al., 2013; Zeitlinger, 2020). All of these make the study of the regulatory code and even the definition of what an enhancer is very complicated. Additionally, the genomic context of enhancers has been shown to affect enhancers as well. For instance, so called 'shadow' enhancers are groups of redundant enhancers that completely or partially share functionality and can activate the gene even after one in the set is inactivated (e.g. enhancers of Atoh7 gene (Miesfeld et al., 2020) in retinal development). Similarly, cooperativity among enhancers has also been demonstrated. For instance, (Lin et al., 2022a) conducted pairwise CRISPRi inactivation experiments of the 7 MYC enhancers showing complex, epistatic and hierarchical relationships between pairs of enhancer. Thus, enhancer-enhancer relationships can also complicate how enhancers are studied and defined as the presence of one enhancer can change the behavior of the others.

On the other hand, less is known about silencers, i.e. elements that reduce transcriptional activity (Doni Jayavelu et al., 2020). These are elements that contain repressor binding sites or form heterochromatin. Most of the screens for CRE elements are conducted within DHS regions of the genome which might bias the inclusion of these elements as candidates. Therefore, it is currently unclear where we can efficiently search for silencers within the genome and the extent to which they affect gene expression regulation.

In summary, many genes contain a complex landscape of enhancing and silencing elements that potentially interact with each other and lead to the precise and tissue and/or developmental time specific expression levels of a given gene. Each of these elements contains a set of TF binding sites that in turn interact hierarchically to orchestrate regulation of transcriptional rate. The study of CREs has led to contradicting conclusions and debates about the exact definition of an enhancer. Moreover, the complex hierarchical interactions, tissue-specificity, quantitative binding of TFs and vast non-coding space where enhancer or silencer CREs can reside complicate the study of the cis-regulatory grammar. Models of sequence-to-function relationships, i.e. how sequence determinants lead to regulatory activity of elements can shed light on this complex CRE grammar. However, the complexity of the problem requires flexible models that can approximate the complex sequence-to-function landscape. DL models are well suited for this problem.

1.2 Advancing genomics with deep learning

1.2.1 What is deep learning?

DL models are now used for understanding complex systems in various fields, including in genomics. The reason behind their versatility is that DL models are often considered as universal function approximators, capable of uniformly approximating any continuous function (Cybenko, 1989). Compared to traditional machine learning, this capability originates from the layered structure of neural networks. Each layer employs a number of operators applying non-linear transformations to its input data and passing on to the next layer. Through successive applications of non-linearities, DL models can learn to identify useful features directly from data, and learn complex, high-order functions.

According to the Universal Approximation Theorem, a neural network with at least one hidden layer and a sufficient number of neurons can approximate any continuous function to a desired degree of accuracy, given appropriate activation functions (Cybenko, 1989). But in practice, deeper models and more complicated operators are usually used. Expressivity is typically used to describe the capability of a model to represent a wide variety of complex functions. The expressivity of DL models is impacted by many design factors including width and depth, types of layers and activation functions, optimization algorithms, etc (Raghu et al., 2017; Shaham, Cloninger, and Coifman, 2018). The design choices also reflect our assumptions about the data structure which is often refereed to as an inductive bias. The inductive bias of a model determines its effectiveness in learning from the training data and its ability to generalize well to new, unseen data. A well-chosen inductive bias will make it easier for the model to learn efficient representations, extracting features helpful for the predictive tasks.

1.2.2 Brief history of deep learning

Despite the recent craze in genomic DL, DNNs have been used to understand biological problems since the 1980s and 1990s. They were initially applied to relatively simpler tasks such as identifying *E. coli* translational initiation sites (Stormo et al., 1982), annotating coding regions (Snyder and Stormo, 1993), aligning regulatory sites (Heumann, Lapedes, and Stormo, 1995) and recognizing protein domains (Bengio and Pouliot, 1990).

DL, despite its long history of study, experienced a period of relative stagnation until 2012, marked by the publication and success of AlexNet (Krizhevsky, Sutskever, and Hinton, 2012). AlexNet, a deep CNN was submitted to the ImageNet Large Scale Visual Recognition Challenge. It significantly outperformed the then state-of-the-art (SOTA) models based on support vector machines by a large margin. This success underscored the potential of DNNs to tackle complex predictive tasks in an end-to-end manner. Following AlexNet, there was a significant resurgence of interest in DL. This led to the development of diverse model structures and training algorithms, catalyzing a 'revival' in DL research and applications (LeCun, Bengio, and Hinton, 2015).

1.2.3 The rise of genomic deep learning

Modeling of genomics sequence-to-function relationships underwent a similar revolution after the success of DL models in other fields such as image classification. This was also fueled by the need for advanced methods given the complexity of the cis-regulatory code and the ability of DNNs to model complex hierarchical relationships in the data.

One of the first models in the field is DeepBind (Alipanahi et al., 2015b) - a CNN for prediction of protein biding score from both microarray and TF ChIP-seq experiments using 14–101 nucleotides of DNA sequence as input. DeepBind used a simple model structure with a single convolutional layer followed by a ReLU activation and max pooling. The convolutional filters were thought to fit well to the task given their conceptual similarity to PWMs [see section 1.2.4 on PWMs]. The ReLU layer allowed to prune out positions with negative activation (corresponding to positions that did not match the learnt filters) and the maxpooling layer summarized the activation across the input sequence. With this simple design DeepBind outperformed the other methods, e.g. based on kmer content.

DeepSea (Zhou and Troyanskaya, 2015a) is another early genomics DNN that expanded the architecture to include 3 layers of convolutional and maxpooling blocks which allowed learning features at multiple scales. These layers transformed the input features (DNA sequence of length 1 Kb) to predict a vector of binary labels corresponding to multiple epigenetic assays, specifically TF and histone modification ChIP-seq and DNase-seq (975 tasks in total). In addition to these innovations the model was proposed and used to score SNPs for their potential to be causal of epigenetic changes. Similarly, Basset (Kelley, Snoek, and Rinn, 2016) utilized 3 blocks of convolution, ReLU and maxpooling to predict DNase-seq peaks (binary labels) of 164 cell lines from 600bp input DNA.

These models set the foundations for genomic DNNs introducing the use of multiple convolutional layers to learn motifs and multi-scale interactions, predicting various assay output (multi-tasking) and expanding input DNA length to incorporate genomic context into the predictions. Each model was shown to outperform traditional bioinformatics methods which are described next.

1.2.4 Advantage over traditional PWMs/k-mer methods.

Traditionally, statistical methods such as PWMs have been used for understanding sequence-function relationships. PWMs quantify the variation of nucleotides at specific positions within a set of aligned sequences known to share a common function such as protein binding (Stormo et al., 1982; Berg and Hippel, 1987; Stormo, 2000; Foat, Morozov, and Bussemaker, 2006). However, they are convolution-based models that assume an additive effects model, treating each nucleotide independently and ignoring motif positioning within sequences. Various extensions to the PWM have been proposed to incorporate pairwise interaction terms between adjacent positions in the motif (Bulyk, Johnson, and Church, 2002; Siddharthan, 2010; Berger et al., 2006) or between all positions in the motif (Omidi et al., 2017; Zhao et al., 2012; Tomovic and Oakeley, 2007); allow multiple PWMs for a single motif (Hannenhalli and Wang, 2005); or extend the PWM model in a Markovian manner by allowing the probability at each position to depend on some number of previous positions (Siebert and Söding, 2016; Mathelier and Wasserman, 2013; Ge et al., 2021), in addition to more expressive Bayesian networks (Barash et al., 2003; Keilwagen and Grau, 2015; Pudimat, Schukat-Talamazzini, and Backofen, 2005). A key limitation of previous PWM-based methods is that they do not consider long-range dependencies between motifs or sequence context, beyond the core binding sites – factors that can influence non-additive functional activity via cooperativity (Venkatesh et al., 2021; Slattery et al., 2011; Monahan et al., 2017; Huang et al., 2015), competition (Baeza-Centurion et al., 2019; Kwasnieski et al., 2012), or other dependencies, such as nucleosome positioning (Zhu et al., 2018; Segal et al., 2006).

Comparatively, CNNs can be considered as a generalization of PWMs, with each layer serving as a PWM-like model. Thus, deeper layers take as input PWM-like scans from the previous layer, enabling them to learn non-linear interactions within and across binding sites. Also, unlike the rigid binary motif scans employed by traditional PWMs, CNNs adopt a more flexible assumptions about the expressive representation of motifs, capturing a wider range of affinities than PWMs. Additionally, DNNs have a larger receptive field, i.e. longer input area that affects the calculation of a given output unit (Araujo, Norris, and Sim, 2019). Therefore, more of the surrounding genetic context and distance dependencies between detected motif pattern are considered during prediction. Collectively, these features make DNNs more beneficial for understanding sequence to function relationships.

1.2.5 The second wave of genomic DNN applications

With the growing interest in using DNNs to model sequence-function relationships in genomics several early innovations followed the initial attempts (outlined in section 1.2.3). Below is a brief review of the important innovations in the early stages following these pioneering works. For more extensive reviews refer to (Angermueller et al., 2016; Eraslan et al., 2019; Ching et al., 2018; Zou et al., 2019; Koo and Ploenzke, 2020).

One of the earlier models, DanQ (Quang and Xie, 2016) was developed to predict 919 binary target vectors from ChIP-seq and DNase-seq assays for each input of 1Kb DNA sequence. It introduced bi-directional long short-term memory network (BLSTM) on top of a convolutional and maxpooling layer block as an innovation in genomics models. This was motivated by the need to allow motif positioning and interactions to be learnt across the length of the sequence. Coda (Koh, Pierson, and Kundaje, 2017) was another model introducing a novel approach to denoising histone ChIP-seq and DNase-seq datasets using two separate CNNs - one for binary classification of presence or absence of a peak and the other for the regression task of normalized read counts (per 25bp bin). Following the popularity of DNNs in genomics (Zeng et al., 2016) compared simple architectural choices (e.g. number of convolutional layers) and benchmarked their utility for prediction of DNA-protein binding.

Following the early success of DNNs in other tasks (Cuperus et al., 2017) aimed to model the expression levels of a functional protein required for growth based on a library of 5’ UTR sequences in yeast *Saccharomyces cerevisiae*. This was one of the earliest attempts at predicting expression (through the massively parallel growth selection assay outputs) as well as overcoming the limitations of the inherently small size of the training set by designing new synthetic sequences. Ref.(Angermueller et al., 2017) used DNNs to impute missing single cell data using DNA sequence and single cell CpG methylation data using bidirectional gated recurrent networks. This was the first use of DNNs to predict the methylation state of the chromosome. Similarly, SpliceAI (Jaganathan et al., 2019) was the first DNN trained to clasify genomic positions as splicing donor, acceptor or neither of those classes using a 10Kb sequence as input. This work also compared different depths of DNNs and scales of input sequences in addition to using dilated convolutions and residual connections. Overall, this wave of models explored a diversity of new prediction tasks and some early innovations of architectural choices.

1.3 A maturing field: major breakthroughs in genomic DNN applications since 2017

Following the early success of genomic DNNs, there has been a surge of applications, pushing the boundaries of what can be predicted from DNA sequences. Building on the early models which mostly used simple blocks of convolutions and maxpooling the next generation of DNNs used various architectural tricks to improve the models [see section 1.4.1 for architectural innovations]. These allowed for larger input sizes, better performance outcomes and (in theory) improved flow of information between different parts of the input sequence. This is crucial for certain modeling tasks such as gene expression prediction which depend on distal regulators. Many of these models were also trained to predict quantitative readouts of larger sets of epigenetic marks (at various resolutions). In addition, new data modalities were introduced, e.g. prediction of regulatory activity chromatin structure or gene expression. Overall, this generation of DNNs applied innovations in architecture and explored various target datasets as discussed next.

1.3.1 Chromatin profiling: TF binding and histone marks and chromatin accessibility

DNNs have been used for prediction of cell-type specific chromatin profiles using DNA sequence only or coupled with other data modalities. Initial efforts

(Quang and Xie, 2019; Li, Quang, and Guan, 2019; Asif and Orenstein, 2020) were towards prediction of binarized target labels (binding or no binding events based on peak-calling algorithm outputs). To make models generalizable to cell types not included in the training dataset some of the models incorporated ATAC-seq or DNase-seq as cell-type specific input to the models (Quang and Xie, 2019; Cazares et al., 2023). More recently modeling efforts have shifted to quantitative or base-resolution prediction of the TF binding profiles.

Deviating from this pattern, BPNet (Avsec et al., 2021a) was trained to predict ChIP-Nexus profiles from DNA sequences for 4 TFs of interest. The model was then used to perform *in silico* experiments to address questions about TF co-binding depending on distance revealing known patterns of TF-TF interaction. Similarly, DNNs were initially trained to predict accessibility of binarized DNase-seq (Kelley, Snoek, and Rinn, 2016; Minnoye et al., 2020) and histone modification ChIP-seq data (Yin et al., 2019). The trend in this task group is the same in moving from binary to quantitative profile predictions. Moreover, a series of models by (Kelley et al., 2018a; Kelley, 2020; Avsec et al., 2021c; Linder et al., 2023) introduced quantitative and parallel prediction of binned (lower than base-resolution) accessibility, histone and TF ChIP-seq and CAGE-seq track predictions.

1.3.2 Regulatory element activity

Many of the introduced models have been benchmarked using datasets of perturbation-based functional genomics assays, such as CRISPRi or MPRA. A subset of models has also focused on directly modeling the regulatory function of DNA elements based on data from such assays. For instance, STARR-seq measures enhancer activity in a massively parallel episomal assay and DeepSTARR (Almeida et al., 2022) was trained to predict this activity based on input DNA sequence of the putative regulatory element discovering different syntax rules governing housekeeping and tissue-specific gene regulation. Similarly, lentiMPRA measures regulatory activity of DNA segments inserted into the genome at random positions and (Agarwal et al., 2023) trained MPRAnn to predict the regulatory activity of a given element in a specific cell line and identified elements with cell type specific activity. Regulatory element activity was also modelled in yeast (Vaishnav et al., 2022) in order to uncover evolutionary dynamics of these elements.

1.3.3 DNNs for deciphering the cis-regulatory code

In the past, DNNs have been used to model various data types – e.g. ATAC-seq, ChIP-seq, and CAGE. Although not all the models have been used or interpreted to their full potential, DNNs and interpretability methods have been used to dissect the CREs in specific cell lines. For instance, Kim et al. (Kim et al., 2020), used this approach to model the regulatory grammar of epithelial cells, discovering a combinatorial and dynamic set of motifs that determine tissue differentiation. In a similar work, Nair, et al. (Nair et al., 2023), used DNNs to uncover transiently active and non-canonical motifs of TFs involved in fibroblast differentiation. Others have applied this to study cardiogenesis (Ameen et al., 2022), DNA accessibility in Drosophila embryos (Brennan et al., 2023), binding affinities in yeast and mammalian TFs (Le et al., 2018; Alexandari et al., 2023), etc. Here I will summarize the main directions of DNN model application classified by the target assay modality.

1.3.4 Gene expression

One of the main challenges in predicting the amount of steady-state mRNA expression levels through RNA-seq from DNA sequence as input is the multitude of factors affecting the rates of RNA production, stability and degradation as well as presence of long-range silencer- or enhancer-promoter interactions etc.

Although many of these factors are believed to have sequence determinants, these can be located kilobases away from the promoter sequence. Early models were limited to shorter input sequences because of computational and technical constraints. Therefore, the initial efforts focused on using only the promoter sequence albeit with limited success. This was pioneered by (Agarwal and Shendure, 2020) and (Zhou et al., 2018a) who developed models that predicted a scalar value of gene expression as log RPKM (Reads Per Kilobase per Million) from bulk RNA-seq per input DNA sequence. Afterwards the efforts of predicting gene expression shifted to predicting quantitative CAGE-seq profiles (Kelley et al., 2018a; Kelley, 2020; Avsec et al., 2021c; Karbalayghareh, Sahin, and Leslie, 2022) as a proxy for gene expression achieving performance almost equal to replicate-replicate agreement for gene-centered loci. Similarly, (Dudnyk, Shi, and Zhou, 2023) used DNNs to predict readouts of transcription initiation (through CAGE, GRO-cap, PRO-seq, RAMPAGE) using DNA sequence. Notably, (Karbalayghareh, Sahin, and Leslie, 2022) incorporated the 3D structure of the genome in the predictions essentially forcing the model to focus on biologically important information - i.e. elements in the spacial proximity, into the predictions. Finally, (Linder et al., 2023) reversed the trend in the field by developing Borzoi - a large scale DNN that predicts RNA-seq coverage values at 32bp resolution for more than half a megabase DNA sequence. This was mainly possible due to architectural innovations which allowed the inclusion of a larger input size sequence while maintaining a large receptive field (see section 1.4.1 on technical details). Overall, one of the trends in the field of genomic sequence-to-function DNNs, especially in expression prediction is incorporation of more data modalities and longer input sequences.

1.3.5 Chromatin structure

Vertebrate genomes are organized in a hierarchical way in the 3D space so that long-range interactions between enhancers and silencers can take place. Although the extent to which this influences or is causal to transcription is unclear this is an important part in gene regulation. Moreover, mistakes in genome folding can lead to enhancer 'hijacking' which in turn can cause disease (Della Chiara et al., 2023). The 3D structure of the genome can be studied using Chromosome Conformation Capture or 3C (Dekker et al., 2002) assays such as Hi-C (Belton et al., 2012) or Micro-C (Hsieh et al., 2015) that map the high contact regions representing them as a heatmap or a 2D matrix. DNNs have been used to uncover the sequence determinants of the 3D genome structure in a cell-type specific manner by modeling such matrices based on input DNA sequences (Yang et al., 2023; Fudenberg, Kelley, and Pollard, 2020; Schwessinger et al., 2020; Tan et al., 2023; Zhou, 2022). These models, notably Orca (Zhou, 2022) operate at chromosome length scales and have shown aptitude to predict genome folding given structural variants. However, very few models (Karbalayghareh, Sahin, and Leslie, 2022) have incorporated 3D genome structure into gene expression prediction.

1.3.6 Disease associated variants

Predicting the implications of putative disease causing variants is a major downstream application of sequence-to-function models (Wong et al., 2021). In an ideal scenario, a DNN trained to predict epigenetic assays or gene expression levels would capture the biological mechanisms in sequences that lead to normal expression of transcripts or their post-transcriptional regulation (e.g. splicing, RNA-stability, etc). In order to assess how well models capture the effect of point mutations (Avsec et al., 2021c) used the CAGI5 (Critical Assessment of Genome

Interpretation) dataset of saturation mutagenesis (Kircher et al., 2019), to score how well the introduced model, i.e. Enformer, predicts changes in expression caused by point mutations. Generally, models like Enformer would then be used to score non-coding sequence mutations and rank them according to how detrimental they would be to normal expression levels. Alternatively, these could be used to score known GWAS or eQTL variants to fine-map variants of interest among candidate positions. Lastly, DNNs could be used to predict implications of genetic variation for a particular gene from personalized genomes. However, (Huang et al., 2023) and (Sasse et al., 2023) have shown that current state-of-the-art models struggle to do this as summarized in (Tang, Toneyan, and Koo, 2023).

DNNs in regulatory sequence design. Sequence design is the rational design of new sequence elements that satisfy or improve certain desiderata. An example of this is adeno-associated virus or AAV capsid sequence design where there are multiple objectives or goals, e.g. packaging efficiency. This is usually done at the DNA sequence level as libraries are constructed for viral assembly. Machine learning or deep learning can then be used to predict the packaging efficiency and therefore propose mutated versions of the sequence that will yield a higher enrichment for successfully packaged viruses (Zhu et al., 2024; Ogden et al., 2019). Similarly, DNNs can be used for *de novo* discovery of enhancer elements for tissue-specific expression of a given gene (Almeida et al., 2022; Almeida et al., 2024).

1.4 Innovations that are empowering breakthroughs

1.4.1 Architectures

Although the majority of sequence to function models still rely on convolutional networks without architectural innovations, many modern techniques that can achieve better performances have be developed. Among these, dilated convolutional layers (Yu, Koltun, and Funkhouser, 2017) were designed to broaden the receptive field without an increase in parameter count. By introducing a dilation rate that skips input values at a given interval, dilated convolutions allow the network to collect information over larger areas of the input data, enabling more efficient detection of long-range interactions (Yu and Koltun, 2015; Kelley et al., 2018b; Jaganathan et al., 2019). Similarly, various pooling layers have been developed to reduce the dimensions of input features while retaining critical information. Techniques like max pooling and average pooling are commonly used to maintain either the most significant or average feature statistics learned by the network (Boureau, Ponce, and LeCun, 2010). In contrast, attention pooling offers a dynamic focus during dimensionality reduction (Er et al., 2016). In addition to expanded reception field, skipped connection was developed to address the diminishing gradient problem of traditional linear model structure, simplifying model training (He et al., 2016). These architectural innovations combined constitute squeeze-and-excitation (SE) networks. SE blocks can adaptively weigh channel-wise feature importance by explicitly modelling interdependencies between channels (Hu, Shen, and Sun, 2018). There have also been innovations more specific to features of biological data. Reverse-complement equivariant CNNs were developed to ensure that the DNNs learn the inherent symmetry of reverse complement DNA sequences (Mallet and Vert, 2021).

Beyond CNNs, other types of neural networks has also been applied to sequence data. Recurrent neural network (RNN) architectures such as Bidirectional Long Short-Term Memory (bi-LSTM) layers are specifically designed to capture long-distance interactions within data. This is achieved by maintaining an internal state that ’memorizes’ contextual inputs over time. When applied to sequence models, the convolution layer captures regulatory motifs, while the recurrent layer latter captures long-term dependencies between the motifs in order to learn a regulatory ‘grammar’ to improve predictions (Quang and Xie, 2016; Zhang et al., 2020). Additionally, there has been a resurgence of RNN based state-space models. These new layers are more often used in self-supervised sequence models (Nguyen et al., 2024a; Gu and Dao, 2023).

Lastly, self-attention layers, capable of learning dependencies across the full span of the input sequence regardless of distance, offer a significant advantage in contexts where comprehensive contextual understanding is critical. Initially developed for natural language processing, this approach has been applied to diverse fields including biological tasks, offering promising performances (Avsec et al., 2021c; Dalla-Torre et al., 2023; Lin et al., 2022b; Zhou et al., 2023).

1.4.2 Training methods

In addition to architectural innovations training strategy choices can also affect the DNN and various methods have been used in genomics. Below is an outline of the main components of training methods.

Activation function Activation function is (most of the time) a non-linear function that is used to produce the final outputs of units or neurons. Mirroring the success of rectified linear unit (ReLU) activation function in image classification (Krizhevsky, Sutskever, and Hinton, 2012), genomics models also most commonly use ReLU which clamps negative values to zero. Other functions such as tanh, softplus, Gaussian Error Linear Unit (GELU) (Hendrycks and Gimpel, 2016) etc. have also been deployed. Interestingly, (Koo and Ploenzke, 2021) showed that using exponential activation function instead of ReLU improves the interpretability of the genomic models illustrating the importance of nuances in training method choices.

Initialization Initialization of weights, i.e. how the weights are set to start the training, is another important aspect of modeling. Note, that all initialization strategies sample from different distributions instead of setting all the values to the same number to ensure symmetry breaking - i.e. units learning non-redundant features. Interestingly, early work (Glorot and Bengio, 2010; Pennington, Schoenholz, and Ganguli, 2017; He et al., 2015) on how to sample these values demonstrated that there is interaction between the choice of initialization and activation function, illustrating that the modeling choices do not act independently. Despite the relative lack of theoretical basis, the most commonly used initializations in genomics are He (He et al., 2015) and Glorot (Glorot and Bengio, 2010) due to their empirical success.

Regularization Regularization strategies aim to reduce overfitting and increase model generalization to held-out data. Some approaches, such as dropout and early stopping are widely adopted in genomics and elsewhere. Dropout (Srivastava, 2013) randomly removes a subset of nodes (and connections to those nodes) during training, essentially training an ensemble of multiple sub-models and averaging those to make a prediction. Another approach commonly used in genomics is weight decay, where the loss calculation is modified to include a penalty term to reduce the number of non-zero parameters (L1-norm) or reduce the values of weights (L2-norm). Batch normalization (Ioffe and Szegedy, 2015) which normalizes the activations of each layer across data points and layer normalization (Ba, Kiros, and Hinton, 2016) which normalizes across activation values in the layer per data point are also commonly used to stabilize the model gradients and prevent overfitting.

Optimizers and learning rate The loss surface – i.e. the error or loss value at each possible combination of weights or parameters of most DNNs is extremely complex due to over-parameterization of DNNs (much larger number of DNN parameters than data points (Wu, Wang, and Su, 2022). Because of the lack of an analytical solution to the optimal parameter set, DNNs are optimized through gradient-based, iterative optimization methods. One of the simplest methods is stochastic gradient descent (SGD) (Robbins and Monro, 1951) where the parameters are updated by computing the gradient of the loss (w.r.t. weights) given a data point (or a mini-batch) and subtracting this gradient multiplied by the learning rate. The learning rate determines the step size of the changes in the parameters. Although certain genomic DNNs use SGD, the most popular method is Adam (Diederik, 2014). This method uses momentum combined with root mean square propagation (RMSProp). Intuitively, momentum allows parameter updates to take into account the 'inertia' of the weight change in the prior step which reduces oscillations on the loss surface, especially when the gradients in one direction are very large. RMSProp adjusts the learning rate for each parameter by dividing it with the exponentially weighted sum of all previous gradients. Additionally, various, more advanced tricks have been proposed, e.g. cosine annealing (Loshchilov and Hutter, 2016) or stochastic weight averaging (Izmailov et al., 2018) which require further exploration in genomic DNNs.

With a similar logic, (Lee et al., 2023) developed EvoAug – a new suite of evolution-inspired data augmentations, e.g. inversions, deletions, etc. This dramatically increases the number of data points and reduced overfitting leading to superior performance on held-out test set across different tasks. Similarly, (Duncan, Mitchell, and Moses, 2023) introduced phylogenetic augmentation which uses phylogenetic information to enlarge the training dataset and incorporate more sequences from related species.

Transfer learning Transfer learning is a machine learning technique where knowledge learnt from pre-training task(s) is used to improve performance or efficiency of a related target task. It is particularly helpful when the dataset size of the target task is the main limiting factor for prediction performance. It is believed that by harnessing relevant knowledge from pre-training, the transferred model will not only provide better performance, but also have better generalizability to new tasks and data (Crawshaw, 2020; Jain et al., 2023). It has also been successful for sequence to function models. Through using models pre-trained on large sets of general biological functions, downstream models focusing on specific biological mechanism or disease have shown better performances (Avsec et al., 2021c; Chen et al., 2022a).

Binary vs quantitative The readouts from bulk sequencing assays such as TF ChIP-seq are the number of reads that pile up at a given genomic region. These are often normalized using a control experiment and are represented as quantitative profiles showing various heights and shapes of peaks at regions of high activity. To answer the question if a region is active or not (e.g. if a TF binds to the region or not) various peak calling algorithms have been developed that use statistically determined thresholds for binarizing the data. This has the potential to reduce the noise in the data and filter for important regions. However, it also reduces the information about a peak (shape and magnitude) into a single value. Moreover, given the lack of ground truth it is unclear which thresholding strategy is the best and whether intermediate intensity peaks should be considered as well. So, there are two representations of such experiments used widely - binary peak locations or the quantitative readouts of the assays. In genomics DNN models both have been used as modeling targets or labels. Initially, given the easier nature of the data type, the field used this to frame the modeling as a classification task with two possible outputs (peak or no peak). Subsequently, with the development of more advanced models the field moved to quantitative models framing the problem as a regression task. Some of these use base-resolution readouts as targets for prediction while others bin the data into various lower resolutions to simplify the task.

Loss function Depending on the task formulation the loss function choice is different between binary and quantitative models. In case of a classification task with binary or peak-based labels binary cross entropy loss is used which is a standard approach in classification problems. In case of the regression based quantitiative readouts the choice of the loss function is more diverse. Models such as (Kelley et al., 2018a; Kelley et al., 2018b; Avsec et al., 2021c) used poisson negative log likelihood (NLL) which assumes each base position or bin read count is independent from the neighboring ones. In contrast, BPNet (Avsec et al., 2021b) split the loss function to focus on two parts: (i) profile shape via multinomial NLL and (ii) total read counts per input sequence via mean squared error.

Multi-task learning Multi-tasking models are trained to predict multiple different tasks in parallel. In genomics this can include prediction of the same epigenetic assay outputs across cell lines, prediction of different epigenetic tracks (e.g. ATAC-seq and ChIP-seq) or both. Multi-tasking is advantageous because (i) the model is less likely to overfit, (ii) can learn a common repertoire of representations (e.g. convolutional filters representing motifs) that are shared between tasks and (iii) for a large number of tasks it can be efficient to train one unified model instead of one model per task. However, depending on the combination of tasks this can lead to adverse effects such as negative transfer, i.e. lower performance in certain tasks compared to single task learning. The majority of genomic DNNs (Zhou and Troyanskaya, 2015b; Avsec et al., 2021d; Kelley et al., 2018b; Kelley et al., 2018a; Chen et al., 2022b) utilize multi-tasking, with the number of tasks ranging from a few ((Avsec et al., 2021b) used 4 cell line datasets) to thousands ((Avsec et al., 2021c) trained Enformer to predict almost 7,000 tracks from human and mouse biosamples). However, it is unclear if combining all the tasks without any grouping or changes in the training configuration is the most optimal approach and requires further investigation.

1.4.3 Model interpretability

Model interpretability has become a crucial aspect of applying DL within genomics, as it aids in understanding the predictive models' decision-making processes. Standard interpretation methods include techniques like filter visualization and performing attribution analysis.

Filter visualization A standard approach for model interpretation is filter visualization (Alipanahi et al., 2015a), which involves analyzing the first convolution layer of a model. This layer scans the input sequence with a set of pattern-recognizing filters, whose weights are optimized during training. Typically, an accessible site on DNA, where regulatory proteins bind, is of interest. Predictive models of accessibility are expected to capture these binding sites or motifs by learning their sequence. However, it's important to note that filters in the first layer are not guaranteed to learn motifs. Nonetheless, certain design principles enable filters to capture motifs effectively (Koo and Eddy, 2019; Koo and Ploenzke, 2021; Novakovsky et al., 2023). By identifying inputs from the test set that maximally activate a filter, aligning them, and converting them into position frequency matrices, these can be compared to known motifs in databases (Castro-Mondragon et al., 2021).

Attribution analysis Attribution analysis, methods for attributing importance scores to each nucleotide based on DNN predictions, further aids post-hoc interpretation of genomic DNNs. These scores form attribution maps, which can help to identify motifs, especially TF binding sites, and their dependencies. Other observed CREs include nucleosome positioning signals and interactions between TFs. Among the plethora of attribution methods, each bears unique advantages and considerations.

In Silico Mutagenesis (ISM) (Zhou and Troyanskaya, 2015b) stands out for its direct correlation with single-nucleotide saturation mutagenesis, offering a straightforward attribution approach in genomic contexts. By comparing the predictions for in silico single-nucleotide variants (SNVs) against the wild-type sequence, ISM constructs an attribution matrix that reveals the impact of each variant, mirroring traditional genomics experiments.

Saliency maps offer a simpler approach, computing input attributions by returning the gradient of the output with respect to the input (Simonyan, Vedaldi, and Zisserman, 2014). This method can be seen as a first-order Taylor expansion of the network at the input, with gradients indicating feature importance. Saliency analysis uses the entire network to identify influential signals in the input data leading to task-specific predictions. This method reveals the importance of each nucleotide in a sequence by computing the gradient of a class prediction with respect to the inputs, allowing the generation of sequence logos highlighting nucleotide significance. Further enriching the interpretability toolkit are methods like Integrated Gradients (Sundararajan, Taly, and Yan, 2017), SmoothGrad (Smilkov et al., 2017), DeepSHAP (Lundberg and Lee, 2017), and DeepLIFT (Shrikumar, Greenside, and Kundaje, 2017).

Despite these methods, interpreting models remains complex since different methods may yield varying insights, shaped by the characteristics of the functions learned by the DNN. Following up with tools like TF-MoDISco (Shrikumar et al., 2018) or in silico experiments, and ultimately experimental validation, can confirm these hypotheses. For instance, TF-MoDISco clusters attribution maps into segments, offering averaged representations for each cluster that can be linked to task-type-specific regulation.

Global interpretations Global interpretability methods such as GIA (Koo et al., 2021) extends interpretability to a broader scale, quantifying the effect size of putative patterns on model predictions across a population. This method tests hypotheses on patterns and their interactions, mapping specific functions learned by the network.

In summary, interpretability methods in genomics provide essential insights into the predictive models' workings. These methods not only help uncover motifs and CREs but also facilitate a deeper understanding of cell-type-specific transcription regulation.

1.5 Overview of thesis chapters

As discussed in this chapter, technical innovations in DL have led to significant advancements for a diverse set of biological inquiries. However, the growing number of model design choices and training techniques has made evaluating their utility, particularly within the domain of regulatory genomics more complex. Since we not only care about training task performance, but also a model's ability to generalize to experimental design and clinical applications, as well as its interpretability for understanding regulation mechanisms. It is unclear how well fitted some of the newest model innovations are for our research interests. In this thesis, I explored the representational capabilities of current DL methods. In Chapter 2, we will introduce a framework designed to assess supervised models, focusing only on prediction performance but also robustness and generalizability. And in Chapter 3, we shifts the focus towards the evaluation of unsupervised models, and explore whether language models can extract meaningful representations from DNA sequences. Finally in Chapter 4, we will discuss the broader implications of DNNs in biology, discussing their generalizability, the limitations we've encountered in our research, and the potential paths forward.

Chapter 2

Evaluating deep learning for predicting epigenomic profiles

Deep learning has been successful at predicting epigenomic profiles from DNA sequences. Most approaches frame this task as a binary classification relying on peak callers to define functional activity. Recently, quantitative models have emerged to directly predict the experimental coverage values as a regression. As new models continue to emerge with different architectures and training configurations, a major bottleneck is forming due to the lack of ability to fairly assess the novelty of proposed models and their utility for downstream biological discovery. Here we introduce a unified evaluation framework and use it to compare various binary and quantitative models trained to predict chromatin accessibility data. We highlight various modeling choices that affect generalization performance, including a downstream application of predicting variant effects. In addition, we introduce a robustness metric that can be used to enhance model selection and improve variant effect predictions. Our empirical study largely supports that quantitative modeling of epigenomic profiles leads to better generalizability and interpretability.

2.1 Introduction

DL has achieved considerable success in predicting epigenomic profiles from DNA sequences, including transcription factor binding (Quang and Xie, 2019; Li, Quang, and Guan, 2019; Zheng et al., 2021), chromatin accessibility(Kelley, Snoek, and Rinn, 2016; Minnoye et al., 2020), methylation (Angermueller et al., 2017), and histone marks (Zhou and Troyanskaya, 2015b; Yin et al., 2019). By learning a sequence-function relationship, trained DL models have been utilized on various downstream tasks, such as predicting the functional effects of single-nucleotide variants associated with human diseases (Dey et al., 2020; Cheng et al., 2021; Zhou and Troyanskaya, 2015b; Zhou et al., 2019; Park et al., 2021; Kelley, Snoek, and Rinn, 2016; Zhou et al., 2018b).

Over the past several years, the variety of DL models proposed to address regulatory genomic prediction tasks has increased substantially(Kim et al., 2021; Novakovsky et al., 2021; Atak et al., 2021; Li et al., 2021; Karbalayghareh, Sahin, and Leslie, 2022; Chen et al., 2022b; Janssens et al., 2022; Vaishnav et al., 2022; Zhou, 2022). The wide variety of proposed models, the datasets they are trained on, how the datasets are processed, and the tricks used to train the models make it challenging to assess which innovations are driving performance gains. A direct comparison of model performance cannot always be made easily due to the variations of how the prediction tasks are framed. For instance, previous approaches typically frame the task as a *binary* classification, where binary labels represent functional activity based on a peak caller. However, in collapsing the amplitude and shape of a peak into a binary label, information about differential *cis*-regulatory mechanisms potentially encoded in these attributes is lost. Recently, *quantitative* models(Kelley et al., 2018b; Kelley, 2020; Maslova et al., 2020; Avsec et al., 2021b; Avsec et al., 2021d) have emerged, similarly taking DNA sequences as input but now directly predicting experimental read coverage values as a regression task, thus bypassing the need for a peak caller and preserving quantitative information of epigenomic tracks. Since standard metrics differ across classification and regression tasks, it remains unclear how to directly compare models trained on binary tasks versus quantitative tasks.

To address this issue, Kelley et al(Kelley et al., 2018b) propose to 'binarize' their quantitative predictions using a peak caller, which enables a comparison of the overlapping regions with binary labels. However, this approach narrowly focuses evaluation on regions of the genome that have been annotated as functional according to a peak caller, which is noisy and sensitive to parameter choices of the peak caller(Koohy et al., 2014). Alternatively, Avsec et al(Avsec et al., 2021b) compared the performance of a binary model with an augmented version of the binary model that appends an output-head that simultaneously predicts quantitative profiles. While this measures the added benefit of quantitative modeling, this approach requires retraining multiple versions of the model, which can be sensitive to initialization, and it does not easily extend to comparisons with existing models.

Moreover, other modeling choices within a prediction task make it challenging to directly make fair comparisons. For instance, existing quantitative models predict different resolutions of the epigenetic profiles. Basenji (Kelley et al., 2018b) predicts non-overlapping binned epigenomic profiles with a resolution of 128 base-pairs (bp), while BPNet(Avsec et al., 2021b) predicts at base-resolution. Comparing models across different resolutions is not straightforward, because binning affects the smoothness of the coverage values which, in turn, can influence performance metrics. Moreover, existing methods employ different data augmentations and analyze different subsets of training and test data, further complicating any direct comparisons.

As the number of applications continues to grow, a bottleneck of modeling innovations is forming as we lack the ability to perform a critical assessment of newly proposed models. Here, we propose an evaluation framework for DL models trained on regulatory genomics data that enables a systematic comparison of prediction performance and model interpretability, irrespective of how the prediction task is framed. Using this framework, we perform a critical assessment of quantitative models and binary models on a chromatin accessibility prediction task to elucidate beneficial factors in model architecture, training procedure, and data augmentation strategies. Moving beyond predictive performance, we assess each model with additional criteria: 1) robustness of predictions to small perturbations to the input sequence, 2) variant effect predictions, and 3) interpretability of the learned representations. Our evaluation framework is packaged in a python-based software, called GOPHER (GenOmic Profile-model compreHensive EvaluatoR).

2.2 Results

Many newly proposed DL models are accompanied with custom software; however, their scope is often limited to employing a specific pipeline, making it difficult to mix-and-match innovations across methods. To gain deeper insights into the factors that drive model performance, it is critical to be able to make a systematic and fair comparison across existing and newly proposed DL models. To address this gap, we developed a new, integrative software package called GOPHER that consists of high-level Tensorflow/Keras-based APIs for data processing, data augmentation strategies, and comprehensive model evaluation, including variant effect predictions and model interpretability, for binary and quantitative modeling of epigenomic profiles (Fig. 2.1).

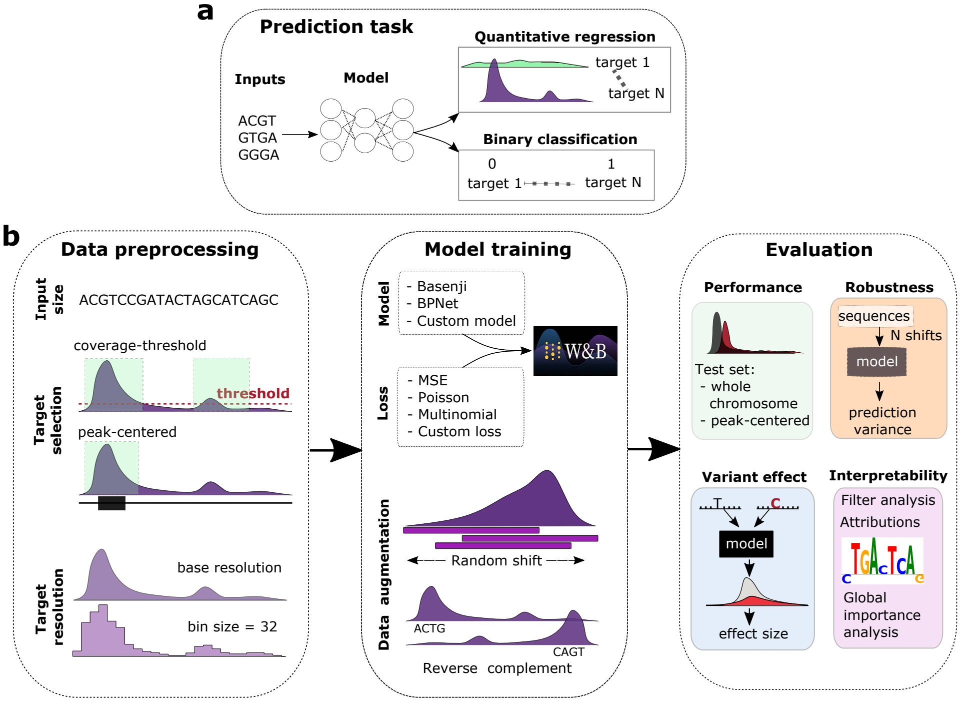


FIGURE 2.1: GOPHER overview. (**a**) Comparison of binary and quantitative prediction tasks for regulatory genomics. (**b**) Illustration of the 3 main components of DL analysis: data preprocessing (i.e. input size, target selection and resolution), model training (i.e. model architecture, loss and data augmentations) and evaluation (i.e. generalization performance, robustness, interpretability and variant effect predictions).

2.2.1 Performance evaluation of best-in-class quantitative models

Prominent quantitative models for regulatory genomics are Basenji(Kelley et al., 2018b) and BPNet(Avsec et al., 2021b). Each employ different strategies for model design, data processing, loss function, evaluation metric, and data augmentations (Supplementary Table A.1), which makes it difficult to identify the key factors that drive performance gains. Thus, we performed a systematic comparison of Basenji- and BPNet-inspired models on a multi-task quantitative prediction of chromatin accessibility ATAC-seq data across 15 human cell lines (see Methods). This dataset provides a sufficient challenge in deciphering the complexity of enhancer activity across cell types but maintains a dataset size that is amenable to the scale of comprehensive evaluations performed in this study.

For each base model, we used GOPHER to search for optimal hyperparameters using each model's original *target resolution* and *training set selections* (Supplementary Fig. B.1). Target resolution defines the bin size of the prediction task, which is used to create non-overlapping windows of coverage values, with the lowest resolution being a bin size of the entire input sequence (i.e. predicting a single quantitative output) while the highest resolution is a bin size of 1 (i.e. base-resolution). BPNet was trained at base resolution on *peak-centered* data (BPNet-base), which consists of a training set selection of genomic regions that contain at least 1 peak from a target cell-type. On the other hand, Basenji was trained at 128 bin-resolution (Basenji-128) with *coverage-threshold* data, which consists of training set selection based on segmenting each chromosome into non-overlapping regions and then sub-selecting the regions that have a max coverage value above a set threshold. Details of the default choices for the dataset and training parameters are detailed in Methods.

Overall, the quality of the model predictions were in line with previous studies (Fig. 2.2a). Using the optimized models as a baseline, we compared the impact of various factors that influence prediction performance, including loss function, target resolution, training set selection, and test set selection.

Loss function. The choice of loss function for quantitative models is not as straightforward as binary models, which is typically a binary cross-entropy. Loss functions can penalize the shapes (e.g. Pearson's r) or the magnitudes (e.g. mean-squared-error (MSE)). BPNet employs a combination of MSE for the magnitude and multinomial loss for the shape. On the other hand, Basenji employs a single loss, Poisson negative log-likelihood (NLL). To explore the effect of loss function on quantitative modeling, we systematically evaluated Basenji-based models and BPNet-based models (using the optimized parameter setting from hyperparameter search) across 5 different loss functions at 8 different target resolutions (Fig. 2.2b). Evidently, Poisson NLL outperformed the other losses at all tested bin-resolutions, i.e. lower MSE and higher Pearson's r on the held out test set. On the other hand, Pearson's r and the combination of Pearson's r and MSE loss yielded the second best overall performance. Interestingly, higher bin sizes tend to yield better performance up to a bin size of about 1 kb for Basenji-based models, which is roughly the width of an ATAC-seq peak. Surprisingly, BPNet-based models yielded a different trend, where base-resolution models performed the best, albeit Poisson NLL remained the best loss (Supplementary Fig. B.2a). This is expected (to a degree) as each of these models were optimized for different resolutions. This suggests that model design can be optimized for a given resolution but may not necessarily generalize across resolutions.

Target resolution. Quantitative models that employ different target resolutions cannot be directly compared, because the bin sizes serve to down-sample the

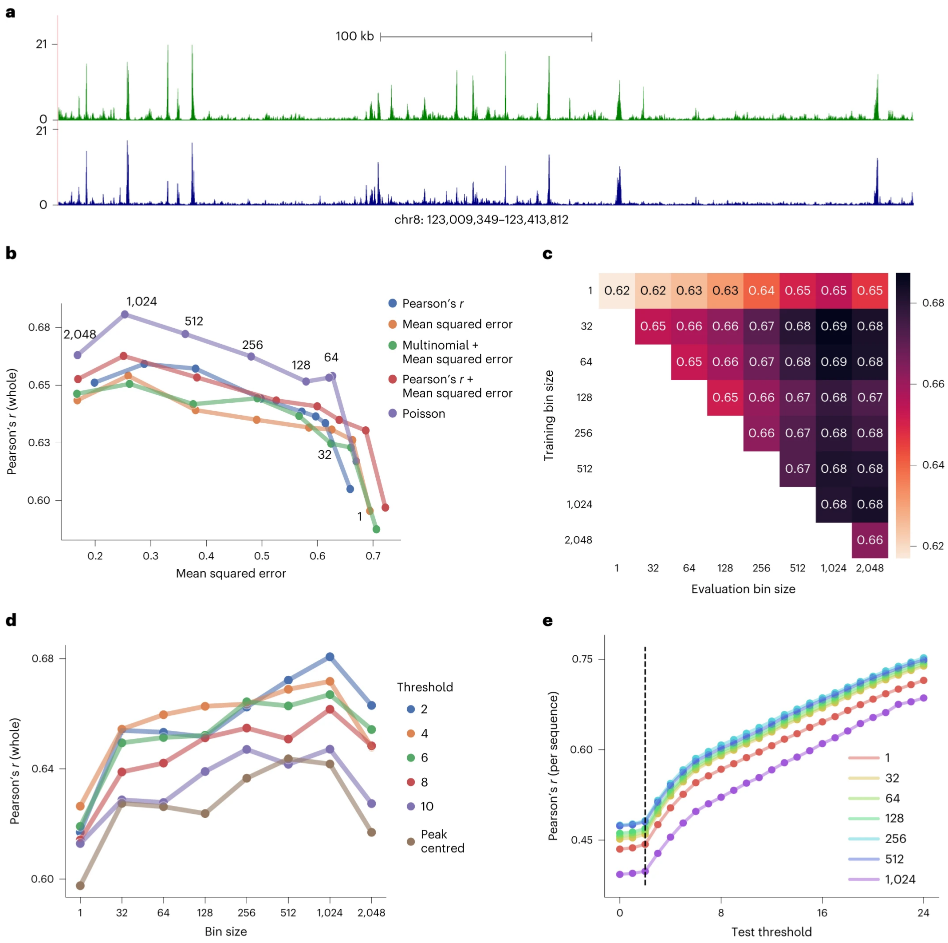


FIGURE 2.2: Evaluation of Basenji-based quantitative models. (**a**) Example visualization of bigWig tracks for experimental measurements and model predictions for GM23338 cell line on a test chromosome for Basenji-128. (**b**) *Loss function analysis.* Scatter plot of the whole-chromosome Pearson's r versus the MSE for different loss functions and target resolutions. Predictions were scaled using ground truth mean coverage (Methods). (**c**) *Target resolution analysis.* Heatmap of the whole-chromosome Pearson's r for models trained on a given bin size with predictions down-sampled to a lower resolution for evaluation. (**d**) *Training set selection analysis*. Scatter plot of whole-chromosome Pearson's r versus different target resolutions for Baenji-based models trained on datasets with a different coverage threshold. Peak-centered represents when model is trained only on genomic regions identified as a peak. (**e**) *Test set selection analysis*. Scatter plot of the thresholded Pearson's r, which is average of per sequence correlation in the thresholded test set, versus different coverage thresholds applied to the test set for different resolution Basenji-based models trained on default data. (**b**-**e**) Pearson's r represents the average across cell lines.

number of data points, which can affect performance metrics based on correlation and binning effectively smoothens high-frequency noise. To explore whether the observed relationship between a higher bin size (i.e. lower resolution) and higher prediction performance for Basenji-based models is due to more accurate predictions or because of the statistical artefacts from binning, we developed an evaluation scheme that enables a direct comparison across target resolutions. Specifically, we binned the predictions of the higher resolution models to match the lower resolution predictions. This effectively provides an avenue to fairly compare the performance across target resolutions. As expected, models trained at a given target resolution yield a higher Pearson's r with increased smoothing, despite that the biology underlying the predictions remains unchanged (Fig. 2.2c). A similar observation was made for BPNet-based models (Supplementary Fig. B.2b). To further demonstrate the sensitivity of Pearson's r on smoothness properties, we systematically smoothed predictions while maintaining the predicted resolution by applying a box-car filter with window sizes that matched a lower-resolution bin and observed a similar trend (Supplementary Fig. B.3).

Training set selection. One major component of generalization performance depends on the composition of training data. To explore the impact of training set selection, we systematically trained Basenji-based models at different resolutions on new datasets with increasing stringency of a coverage threshold, which serves to modulate the balance between the original BPNet's peak-centered training approach and the original Basenji's whole-chromosome training approach. For comparison, we also trained each model on peak-centered data. By evaluating each model on the whole-chromosome test set for consistency, we found that the training set with the lowest threshold yielded the best performance, while peak-centered models performed the worst (Fig. 2.2d). Limiting the model to only higher functional activity reduces the data set size, and hence the number of examples the model has to learn from, which may explain some of the drop in performance. On the other hand, providing too many inactive regions may imbalance the model's focus on features within inactive sites, though we did not observe this undesirable behavior.

Test set selection. Choice of test set can influence the measure of generalization performance. The common approach is to process the training set and test set in the same way and split them via random splits or held-out chromosome(s). Alternatively, predictions across the whole chromosome (via tiled predictions) puts a greater emphasis on generalization to regions of non-functional sites. To explore the influence of test set selection on model performance, we generated new test sets with progressively stringent coverage thresholds to modulate between the two extremes. To compare performance across test thresholds, we calculated the average Pearson's r per sequence, instead of calculating a single Pearson's r across tiled predictions. Interestingly, Basenji-base's performance monotonically increases with the coverage threshold on the test set (Fig. 2.2e). A similar trend was observed across other resolutions. This illustrates that predictions are more accurate for higher coverage regions and thus focusing only on high-activity regions can inflate test performance.

In addition, we performed a more targeted evaluation of the performance at high-activity regions across models trained on either peak-centered or coverage-threshold datasets (Supplementary Table A.2). We observed a consistent trend where models trained on peak-centered data had a slight performance advantage over models trained on coverage-threshold data in terms of scaling the heights of the reads (i.e. lower MSE). However, models trained on coverage-threshold data yielded substantially better peak detection performance across the whole chromosome (i.e. higher Pearson's r). Thus, there is a slight trade-off in scaling the predictions when training on coverage-threshold data, but it leads to lower false positive activity predictions genome-wide.

2.2.2 Robustness test to identify models with fragile predictions

Robustness to input perturbations is a widely used criterion for evaluating the trustworthiness of DL models(Madry et al., 2017; Cohen, Rosenfeld, and Kolter, 2019). Adversarial attacks using small, targeted noise to the inputs dramatically affects the prediction of non-robust models(Goodfellow, Shlens, and Szegedy, 2014). These noise-based perturbations do not naturally extend to genomic data. Alternative perturbations, such as single-nucleotide mutations, can affect function and hence are also inappropriate. We developed a robustness test to measure the sensitivity of model predictions to translational shifts of the input sequences, whose function is largely maintained by also shifting the target predictions (see Methods). Specifically, our robustness test provides a variation score for a given input sequence that is randomly translated N times – the predictions are aligned and only overlapping regions are considered for the variation summary statistic (Fig. 2.3a).

To test the robustness of models across augmentation strategies and choice of training sets, we compared how different combinations of augmentations, including random reverse-complement (RC) transformations and random shifts of the input sequence (up to 1024 bp), affect the model's robustness properties for BPNet-128 and Basenji-128 trained on either peak-centered or coverage-threshold training data. We opted to compare BPNet-128 at a lower resolution (instead of base-resolution) to make a direct comparison across models, since the robustness metric is sensitive to bin-resolution. Indeed, models trained with augmentations

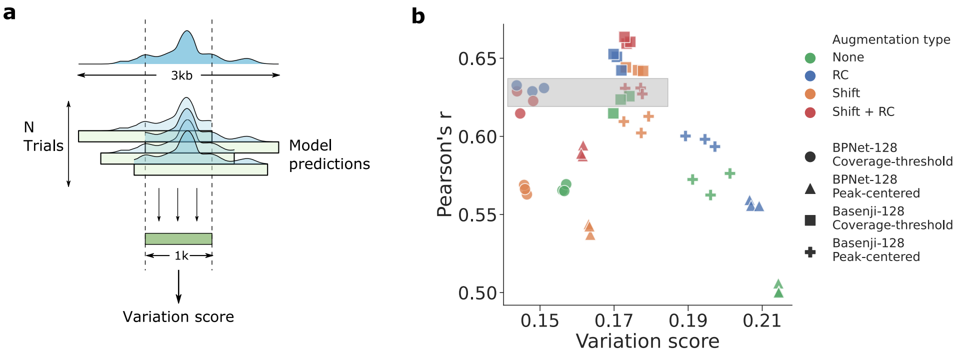


FIGURE 2.3: Testing model robustness against translational shifts. (**a**) Schematic overview of robustness test. For each 3 kb sequence, N random 2 kb sub-sequences were extracted, and the standard deviation across predictions within the overlapping regions is calculated. Average variation score of predictions (i.e. average per-position standard deviation of coverage values normalized by the total mean coverage value) was used as a measure of model robustness. (**b**) Scatter plot of the Pearson's r (averaged across a per-sequence analysis) versus the robustness variation score across models with different augmentation methods (shown in a different color). Each 128 bin-resolution model (shown in a different marker) was trained 3 times with different random initializations. Pearson's r represents the average across cell lines.

yielded improved robustness compared to models without augmentations, especially when trained on peak-centered data (Fig. 2.3b). On the other hand, models that were trained on coverage-threshold data already benefited to a large degree on the non-centered, random nature of the epigenomic profiles. This could explain the observation that in BPNet-128 with RC augmentation alone was sufficient, although Basenji-128 still benefited from both augmentations. Surprisingly, we observed that models with similar prediction performance could yield large differences in their robustness levels, demonstrating that prediction performance and model robustness are not strictly correlated. This suggests that robustness can be utilized along with generalization performance as an additional metric to facilitate model selection.

2.2.3 Comparing quantitative model architectures

The space of binary model architectures has been well explored; however, the exploration in quantitative models has been more limited. Existing quantitative models often have complex designs, including dilated convolutions with skipped connections and task-specific output-heads. It remains unclear to what extent that complex model structures are needed to fit quantitative data. We therefore wanted to address two questions: (1) could standard convolutional neural networks (CNNs) that were successful on binary classifications have similar success at quantitative regressions, and (2) could further exploration of the architectures improve performance?

To address these questions, we benchmarked a baseline CNN – with 3 convolutional layers and 1 fully connected hidden layer – at base and 32 bin-resolutions. We created 2 versions of each model where predictions are made based on task-specific output-heads, where each task is given a nonlinear prediction module or all predictions are based on a linear mapping from a single representation,

			Pearson's r (whole)	
Model	Activation	Output-head	Base	32 Bin
BPNet	ReLU	Task-specific	0.601	0.583
Basenji	GELU	Single	0.617	0.654
CNN	ReLU	Single	0.600	0.605
		Task-specific	0.607	0.604
	Exponential	Single	0.599	0.600
		Task-specific	0.599	0.601
ResidualBind	ReLU	Single	0.642	0.655
		Task-specific	0.637	0.670
	Exponential	Single	0.655	0.665
		Task-specific	0.654	0.677

TABLE 2.1: Table shows the whole-chromosome Pearson's r (averaged across cell lines) for various quantitative models with different activations, output-heads, and trained on different target resolutions. For activations, Exponential refers to the application of it to only the first-layer filters while ReLU is used in deeper layers.

i.e. the common multi-task approach. In addition, we set the first-layer activations of each model either to be rectified linear units (ReLU) or exponential activations, which has been shown to improve the quality of learned motif representations(Koo and Ploenzke, 2021). To test the benefits of a wider receptive field to give context to the patterns learned in lower layers, we created an augmented version of the baseline models by adding a residual block (He et al., 2016), each with 4 dilated convolutional layers (Yu and Koltun, 2015; Yu, Koltun, and Funkhouser, 2017), after each of the first two convolutional layers in a manner similar to ResidualBind (Koo et al., 2021). Together, this results in a total of 8 custom models (see Methods for details).

Surprisingly, we found that baseline CNNs perform on par with Basenji- and BPNet-based models, with the exception of Basenji-32 (Table 2.1). This shows that simple model architectures can be effective at predicting epigenomic profiles as a quantitative regression. On the other hand, including dilated residual blocks, but with components arranged differently than Basenji, substantially improved performance at both tested resolutions for ResidualBind. Interestingly, task-specific output-heads consistently yielded better performance versus a single output-head, albeit the effect was variable and small. Moreover, exponential activations yielded comparable results to ReLU-based models, suggesting that high-divergence activations do not negatively affect the ability to make quantitative predictions. Together, this demonstrates that design considerations for quantitative models are largely under-explored and can greatly improve performance.

2.2.4 Benchmarking model performance across binary and quantitative models

Although quantitative models were developed with the aim of preserving more information about epigenomic profiles, directly comparing the different prediction formats between binary and quantitative tasks is not straightforward. To bridge this gap, we developed a way to directly compare binary models to quantitative models by converting predictions from one format to the other (Fig. 2.4a). To convert binary predictions to a quantitative format, we treated the logits (i.e. before the output activation function) as the predicted coverage values. While binary models are not trained to learn signal strength, the model's confidence can be encoded in the unbound logits. Thus, binary models can now be evaluated with quantitative metrics. Moreover, to convert quantitative predictions to a binary format, we calculated the average coverage predictions at positive regions and negative regions based on corresponding binary-labelled data. These two distributions can be used to calculate standard classification metrics, such as the area-under the precision-recall curve (AUPR) and area-under the receiver-operating-characteristic curve (AUROC).

Using this task-conversion evaluation framework, we directly compared the performance of various quantitative models with various binary models (Supplementary Table A.3). Interestingly, when evaluating on peak-centered data, several binary models yielded similar (if not better) AUROC and AUPR compared to quantitative models (Fig. 2.4b and Supplementary Fig. B.4a). However, when converting the binary models to quantitative metrics, quantitative models outperformed all binary models. This effect became more pronounced when evaluation was extended across the whole chromosome, where all quantitative models yielded better performance across both metrics (Fig. 2.4c and Supplementary Fig. B.4b).

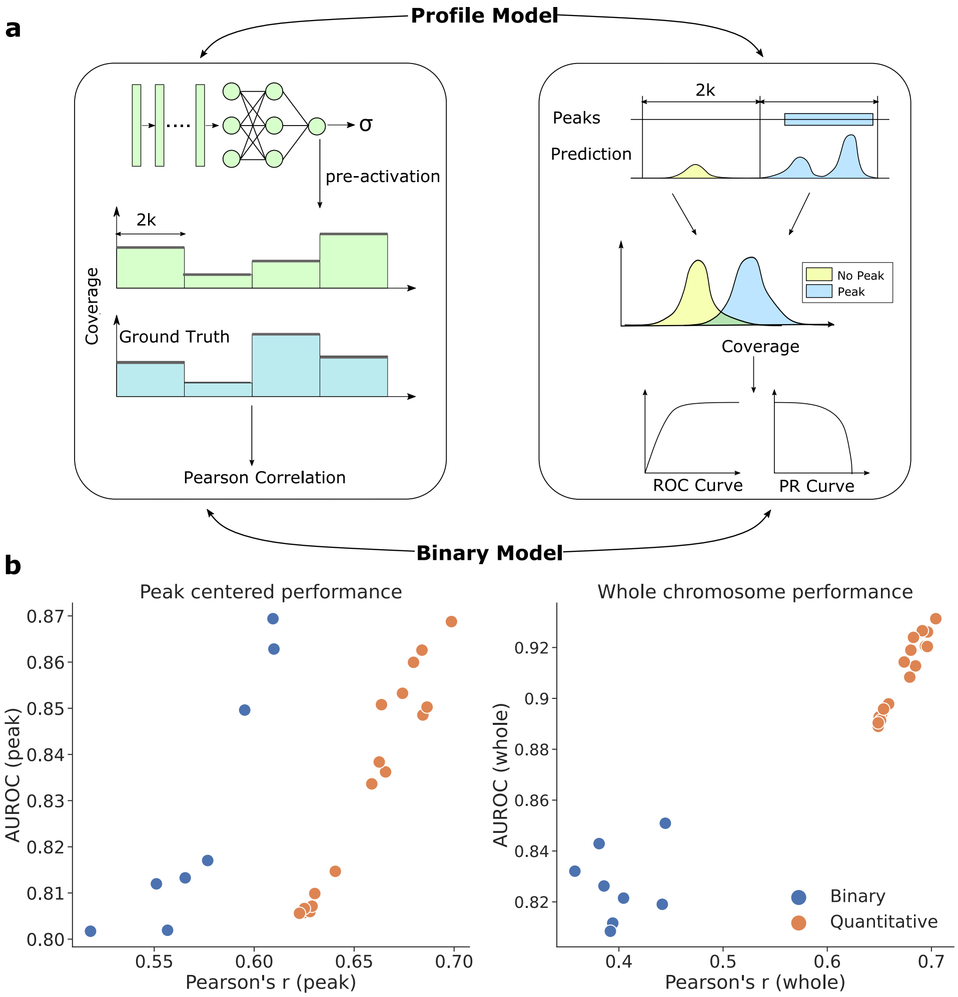


FIGURE 2.4: Performance comparison between binary and quantitative models. (**a**) Schematic overview of prediction task conversion. Binary models can use logits to generate continuous 'coverage-like' values to calculate regression metrics. On the other hand, the coverage predictions of quantitative models can be grouped according to binary labels (i.e. peak and no peak groups) to calculate standard classification metrics. (**b**) Scatter plot of the classification-based AUROC versus the regression-based Pearson's r for various binary models (blue) and quantitative models (orange) on peak-centered test data (left) and whole-chromosome test data (right). Metrics represent the averaged value across cell lines.

Together, this demonstrates that while some binary models can be competitive with quantitative models within high-activity functional sites, quantitative models tend to yield better overall performance across whole chromosomes.

2.2.5 Out-of-distribution generalization: variant effect prediction

A major downstream application for DL models that learn sequence-function relationships is to utilize them to score the functional effects of mutations. High-performing models can inform how their predictions change relative to wild-type when queried with a new mutated sequence. Thus, we benchmarked each model on this out-of-distribution (OOD) generalization task by validating predictions with experimental data from the CAGI5 Competition (Kircher et al., 2019; Shigaki et al., 2019a), similar to previous studies (Minnoye et al., 2020; Avsec et al., 2021d). CAGI5 dataset consists of massively parallel reporter assays (MPRAs) that measure the effect size of single-nucleotide variants through saturation mutagenesis of 15 different regulatory elements across different cell-types. Instead of a standard approach that makes a single prediction based on a sequence centered on the variant-of-interest, robust predictions were calculated by introducing random translations and averaging the central overlapping region, similar to our robustness test (see Methods). Robust predictions were calculated separately for reference and alternative alleles and the effect size was calculated based on their log2 fold change. An anecdotal visualization shows that the variant effect predictions by quantitative models are qualitatively effective despite being trained on OOD data – i.e. chromatin accessibility in different cell lines (Fig. 2.5a).

By benchmarking various models, we found that quantitative models consistently outperformed binary models (Fig. 2.5b). In addition, by cross-comparison

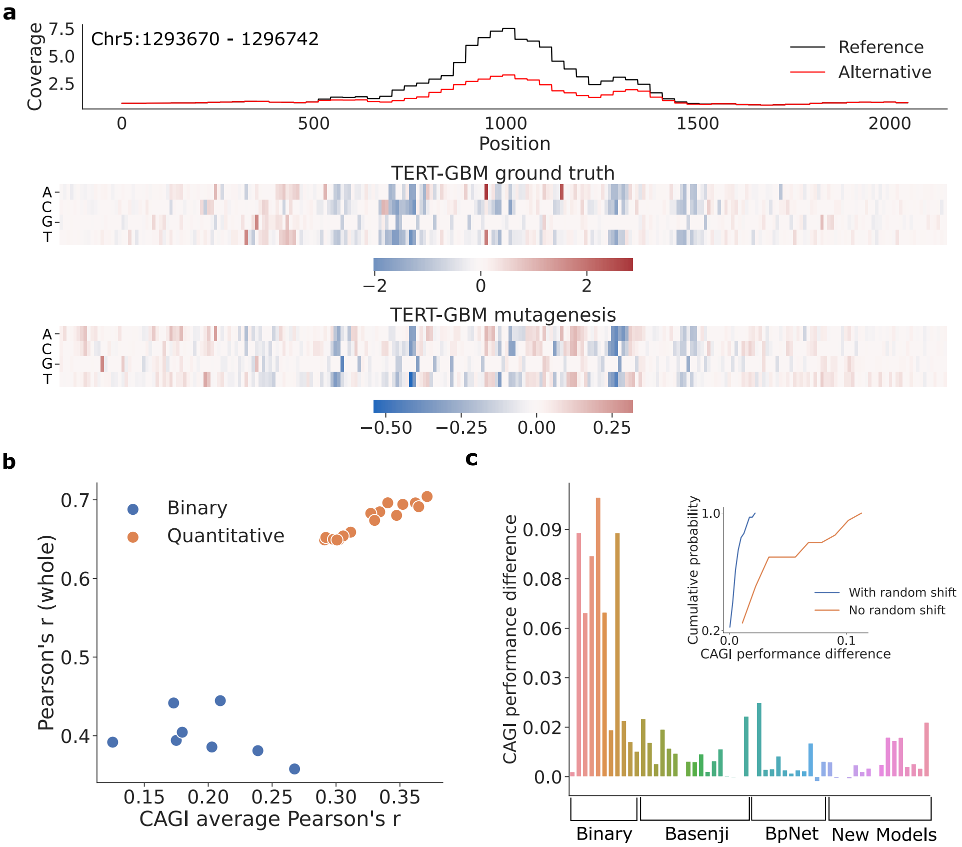


FIGURE 2.5: Comparison of functional effect predictions. (**a**) Example visualization of predictions of a sequence with a reference allele (black curve) and an alternative allele (red curve) for a given mutation. Below, heat maps show the experimental measurements of variant effects for the TERT promoter in a GBM cell line (ground truth) and the predicted variant effects from ResidualBind-32. (**b**) Scatter plot of the prediction performance across whole-chromosome test set (y-axis) and the average CAGI5 performance (x-axis). Each dot represents a different model. (**c**) Bar plot shows the CAGI5 performance difference between robust predictions minus standard predictions. Each bar represents a different model. Groups of models represent different training strategies or target resolutions. Inset shows the cumulative distribution of variant effect performance differences for models trained with and without random shift data augmentation.

with prediction performance, we found that whole-chromosome generalization is a reliable metric for variant effect performance. As a control, we also compared whether robust predictions are beneficial for predicting variant effects compared to the standard approach of employing a single-pass prediction centered on the variant-site. Strikingly, we found that 50 out of 56 models performed better using robust predictions (Fig. 2.5c). Upon further investigation, models that did not employ random shift data augmentations were the ones that indeed benefited the most from robust predictions (Fig. 2.5c, inset). This suggests that robust models yield more trustworthy variant effect predictions, but our post hoc workaround of making predictions more robust could improve the efficacy of less robust models.

2.2.6 Model interpretation

A major downstream application of genomic DL models is interpretability analysis, which can lead to the discovery of functional motifs and their complex interactions (Koo and Ploenzke, 2020). Here we compare binary and quantitative models across several common interpretability approaches: motif discovery through filter visualizations, foot-printing motifs at base-resolution using attribution methods, and quantitatively testing hypotheses *in silico* using global importance analysis (GIA).

Filter interpretability. First-layer convolutional filters provide an inductive bias to learn translational patterns such as motifs. However, the extent that they learn interpretable motifs largely depends on design choices, such as the max-pool size (Koo and Eddy, 2019), activation function (Koo and Ploenzke, 2021), or even the utilization of batch normalization (Ghotra et al., 2021). However, it is not clear whether the same design principles established for binary models extends to quantitative models. To evaluate which models yield better motif representations, we visualized the first-layer filters of various models according to activation-based alignments(Alipanahi et al., 2015a; Kelley, Snoek, and Rinn, 2016) and compared how well they match motifs in the JASPAR database(Castro-Mondragon et al., 2021) using Tomtom (Gupta et al., 2007), a motif search comparison tool. Since the absolute number of hits can be misleading because of low-quality hits from partial motifs, we also consider the q-value, which specifies the confidence level of motif-filter matches. We found that among models that employ ReLU-based activations, binary models generally yield a lower hit-ratio as well as higher q-values, which indicates poor quality hits (Supplementary Table A.3). However, models that employ exponential activations in the first convolutional layer yield higher hit-ratios and higher quality hits for both binary and quantitative models. We did not notice any significant differences in learning motif representations across architectures for a given activation. This suggests that the improved predictions do not necessitate learning better motif representations in the first layer, but the design principle using exponential activations can greatly improve the interpretability of learned motifs in first layer filters for both binary and quantitative models.

Embedding representations and attribution maps. To visualize structure in the data as seen through the lens of the model, we embedded the penultimate (or bottleneck) representations of test sequences for a given class using Uniform Manifold Approximation and Projection (UMAP)(McInnes, Healy, and Melville, 2018). ResidualBind-32 with exponential activation yielded distinct UMAP structures and clustered data points with similar profile distributions in each cell line (Supplementary Fig. B.5). By exploring different regions of the UMAP embeddings for the PC-3 cell line as an example, we found that ResidualBind-32 is largely encoding the magnitudes and locations of the read distributions (Supplementary Fig. B.6). We generated attribution maps (based on saliency analysis) from different regions

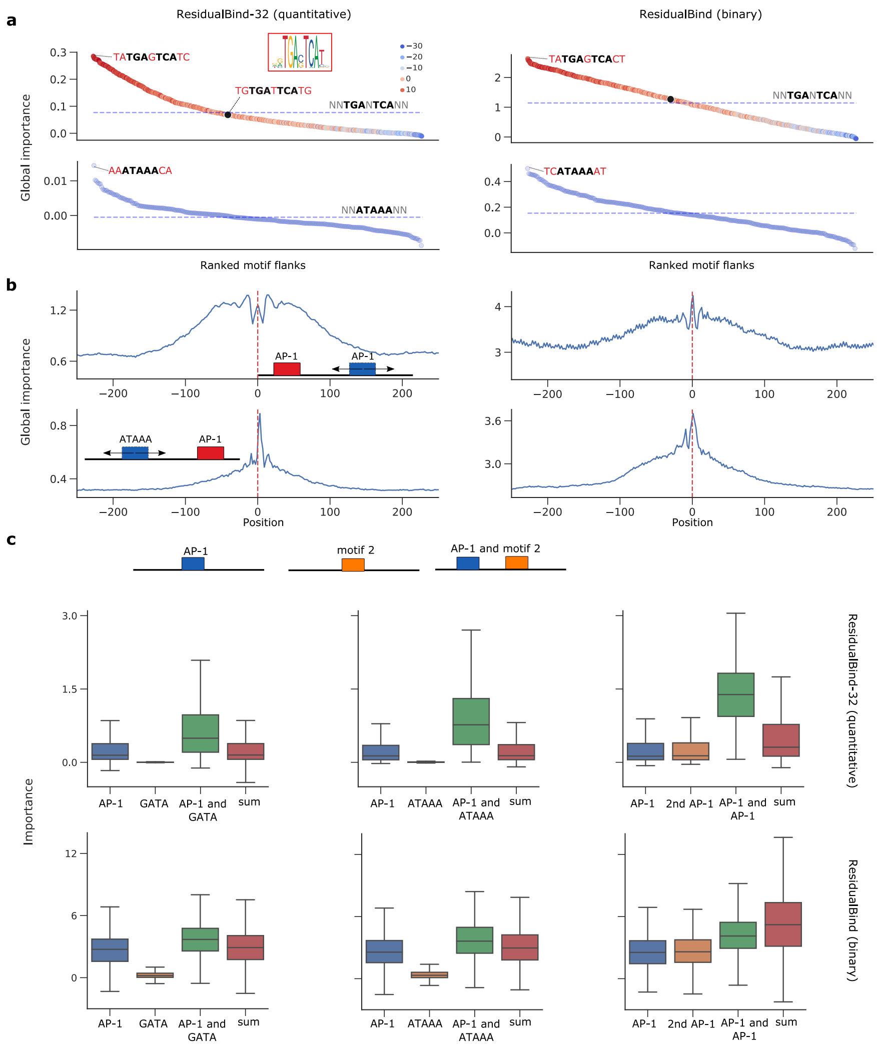


FIGURE 2.6: Interpretability analysis for ResidualBind-32 on PC-3 cell line. (**a**) GIA for optimal flanking nucleotides. Ranked plot of the global importance for each tested flank. Dashed line represents the global importance of the core motif with random flanks. The hue represents the position-weight-matrix score for an AP-1 motif from the JASPAR database. The black dot indicates a high position-weight-matrix score motif that yields a global importance close to the core motif with randomized flanks. (**b**) GIA for distance dependence between two motifs with optimized flanks. Global importance plot for sequences with an AP-1 motif fixed at the center of the sequence and another motif that is systematically placed in different locations.(**c**) GIA for cooperative interactions between AP-1 and another motif. Each subplot shows a box-plot of the global importance when each motif is placed in random background sequences individually and in combinations.

and found that the ResidualBind-32 has learned many known regulatory motifs, such as AP-1, SP1, and GABPA, among others (Supplementary Fig. B.6). Interestingly, many accessible regions with high functional activity for PC-3 contained repeated clusters of AP-1, suggesting that our model considers AP-1 to be a critical motif for accessible sites for PC-3. We also observed an unknown motif (ATAAA) that flanked the AP-1 motif in many attribution maps potentially corresponding to a Forkhead family transcription factor binding site (Castro-Mondragon et al., 2021). Many of these motifs were observed in attribution maps of other models, but due to the lack of ground truth, a quantitative comparison remains difficult.

Global importance analysis. While attribution maps can help to identify and footprint putative motifs, they cannot quantify the importance of motifs beyond individual nucleotides. GIA is an interpretability approach that enables direct testing of hypotheses gained from attribution analysis(Koo et al., 2021). GIA computes the effect size, or global importance score, of hypothesis patterns that are embedded within a population of background sequence, where the other positions are effectively randomized. This approach essentially marginalizes out any confounding patterns within any individual sequence, revealing the global importance of only the embedded patterns on model predictions. Using GIA, we test various hypotheses of AP-1, ATAAA and GATA motifs.

First, we used GIA to explore how the two flanking nucleotides adjacent to the core motif on either side and the central nucleotide in AP-1 influences accessibility predictions in PC-3 cells. We compared the results from our high performing models within each prediction task, quantitative ResidualBind-32 and ResidualBind-binary, both with exponential activations. Strikingly, we find that flanking nucleotide combinations relative to the AP-1's core binding site can drive predictions by a factor of 2 to 3 (Fig. 2.6a), similar to what was observed previously(Mauduit et al., 2021; Almeida et al., 2022). A position-weight-matrix-based approach(Stormo et al., 1982), which considers each position independently, would score many AP-1 binding sites the same, despite their wide spread in functional activity. A similar observation was made for other models, both quantitative and binary (Supplementary Fig. B.7). This demonstrates that DL models consider complex higher-order dependencies of flanking nucleotides to be an important feature of TF binding sites, a well-known phenomenon (Le et al., 2018; Levo et al., 2015). Moreover, both binary and quantitative models can capture this information *de novo* from being trained on just the sequences.

We then explored to what extent the distance between motifs plays a role in model predictions. Specifically, we performed GIA experiments where the AP-1 motif was fixed at the center of the sequences and the position of the other motif (i.e. AP-1 or ATAAA) was varied (Fig. 2.6b). Interestingly, we found that two AP-1 motifs yielded a symmetric 50 bp window where predictions are plateaued, beyond which, the global importance begins to drop off for both models. On the other hand, the ATAAA motif exhibits an asymmetric distance dependence with a favorable location on the 3' end flanking the AP-1 motif with a few nucleotide gap, beyond which there is a precipitous drop in global importance. This was also observed across other models, with variable magnitudes in effect size but a similar relative trend (Supplementary Fig. B.8). These results suggest that flanking nucleotides and distance dependence was consistently learned across quantitative and binary models.

From the attribution maps, it appears that multiple motifs are often present in combinations across accessible sites. To test whether ResidualBind-32 and ResidualBind binary have learned cooperativity between AP-1 and other motifs, we compared the global importance for the motifs embedded in sequences alone and in combinations with other motifs (at the optimal distance identified through the distance dependence GIA experiments). In ResidualBind-32, we observed the sum of individual effects were lower than when both motifs were present, indicating the model has indeed learned cooperative interactions (Fig. 2.6c). The effect size was varied across transcription factors, with a smaller effect observed for GATA::AP-1 compared to other motifs, such as ATAAA::AP-1 and AP-1::AP-1. This suggests that cooperative interactions are strongly associated with chromatin accessibility levels. Surprisingly, a discrepancy arose for binary models, including in ResidualBind binary, for which there was no strong evidence that cooperativity was learned. These trends were also observed across other binary and quantitative models (Supplementary Fig. B.9).

Instead of directly imposing patterns on background sequences, we also conducted occlusion-based interventional experiments where we identified exact instances of the core motif for AP-1 and replaced them with randomized sequences across the test set – a global importance of motif occlusion within its natural sequence context. We find that the number of AP-1 motifs indeed drives high functional activity for PC-3, while other cell lines depend on a different distribution of motifs for their functional activity (Supplementary Fig. B.10).

Together, the interpretations of quantitative models appear to be more consistent with each other than binary models. Despite under-performing on generalization tasks, well-trained binary models can largely capture similar biological interpretations as quantitative models, with the exception of cooperative interactions.

2.3 Discussion

The variety of deep learning models being proposed to predict regulatory genomic tasks has increased substantially in recent years. The variations of proposed models, how the prediction tasks are framed, the composition of the data sets, and the tricks used for training make it challenging to assess which innovations are driving performance gains. Moreover, while many methods provide software to deploy their methods, which only includes their specific pipeline, it is often challenging to mix-and-match modeling innovations across methods. To address this gap, we introduced GOPHER to provide an evaluation framework to compare various modeling choices and enable a comprehensive and fair evaluation of existing and emerging DL models in regulatory genomics. While previous software, such as Janggu (Kopp et al., 2020) and Selene (Chen et al., 2019), help to process biological data (mainly focused on peak-centered data) and high-level APIs to train neural network models in TensorFlow (Abadi et al., 2015) and Pytorch (Paszke et al., 2019), respectively, they do not focus on downstream evaluation across different prediction tasks. By contrast, GOPHER provides a comprehensive model evaluation framework that also supports data processing of peak-based binary classification and quantitative regression analysis of bigWig tracks, in addition to training custom deep learning models with various data augmentations. GOPHER also incorporates many popular model interpretability tools, such as first-layer filter visualization, global importance analysis, and attribution methods, including *in silico* mutagenesis (Nair et al., 2022; Schreiber et al., 2022), saliency maps (Simonyan, Vedaldi, and Zisserman, 2013), integrated gradients (Sundararajan, Taly, and Yan, 2017), and SmoothGrad (Smilkov et al., 2017).

Using GOPHER, we addressed several open questions: (1) how to fairly compare binary models and quantitative models; (2) how choice of loss function affects performance; (3) how dataset selection influences model performance; (4) how to compare quantitative models that making predictions at different resolutions; (5) how augmentation strategies influence model performance and robustness to translational perturbations; how modeling choices influence downstream (6) functional variant effect predictions and (7) model interpretability.

While the study here focuses on ATAC-seq data, the specific claims of optimal architectures and training procedures may be nuanced across other data types, such as ChIP-nexus (Avsec et al., 2021b) and CAGE-seq(Kodzius et al., 2006). In such cases, additional considerations may arise, such as GC-bias and signal normalization. These were not investigated in this study.

Moreover, many of the explored architectures in this study relied on pure convolutional networks. The emergence of transformers (Vaswani et al., 2017) could have better inductive bias to capture distal interactions, though the rationale for the benefits of convolution-based networks and transformer-based networks remains an ongoing research topic (Liu et al., 2022b). Due to the lack of established transformer-based models beyond Enformer (Avsec et al., 2021d), we elected to focus only on convolution-based models in this study.

In addition, BPNet and Basenji were initially developed for predictions on very different data types and dataset sizes. Thus modifications had to be made to each model architecture to adapt it to the ATAC-seq data used in this study (see Methods for differences). These choices may have affected their performance. The fact that both models performed well suggests that our hyperparameter optimization mitigated any substantial disparities.

In general, our work largely supports that quantitative modeling yields better generalization (on average), both on held-out data and OOD variant effect predictions. Of course, well-tuned binary models can perform comparable to (or even better than) a poorly designed quantitative model. It remains unclear whether binary models are fundamentally limited based on their treatment of functional activity or whether incorporating more inactive regions during training would boost performance. Moreover, it is not clear whether the performance gains of quantitative models are due to learning better biological signals or whether they are just better at learning noise sources within sequencing experiments. One major limitation arises as a consequence of focusing on performance – treating experimental measurements as ground truth, despite biological variability across replicates and technical noise (eg. Supplementary Table A.4). Thus, focusing on important downstream tasks, such as variant effect prediction and model interpretability, as was done here, provides a path to move beyond performance benchmarks to the beneficial use case of genomic DL models – biological discovery.

2.4 Methods

Training data

ATAC-seq (Assay for Transposase-Accessible Chromatin with high-throughput sequencing (Buenrostro et al., 2015)) data for human cell lines were acquired from the ENCODE database(Consortium et al., 2012) – fold change over control bigWig files for quantitative analysis and IDR peak bed files for binary analysis – using experimental accessions in Supplementary Table A.5. The bigWig tracks were log-fold-normalized for sequencing depth using control tracks as per the ENCODE data processing pipeline; no further processing was done. Each of the 15 cell lines were sub-selected based on a lower cross-correlation of coverage values at IDR peaks across cell lines below 0.75. Data from replicate 1 for each experiment was used to generate the train, validation, and test sets. Data from replicate 2 was used to assess the experimental ceiling of prediction performance.

Coverage-threshold data. Each chromosome is split into equal, non-overlapping input size chunks of 3 kb and each chunk is included in the dataset if the max coverage value for any of the targets is above the threshold. By default, coverage-threshold data employed a threshold of 2, unless specified otherwise. Each sequence that passed this threshold was included as part of the dataset and down-sampled to 2 kb with a strategy that depends on data augmentations (see below). The targets were then binned with non-overlapping windows according to the specified target resolution and was calculated online during training and testing. For any given coverage value array of length L and bin size B, it was reshaped into an array with shape (B, L/B) – down-sampling was achieved according to the mean within each bin.

Peak-centered data. For peak-centered datasets, we selected IDR bed-files from the ENCODE experiments corresponding to the same replicate as the coverage-threshold data. The bed files of each cell line were merged into a single bed file, in a manner similar to Kelley et al(Kelley, Snoek, and Rinn, 2016). The Basset data processing pipeline divides the genome into segments of length specified as the input size and merges peaks according to an overlap size parameter. Each sequence in the dataset contains at least one peak across all cell lines. Sequences containing an IDR peak for the cell line is given label ‘1’ otherwise label ‘0’.

Data splits. We split the dataset into training, validation, and test sets using chromosome 8 for test, chromosome 9 as validation and the rest as training (excluding Y chromosome). We also removed the unmappable regions across all data splits. The same split was applied to datasets to allow a direct comparisons across experiments.

Held-out test evaluation

Pearson correlation can be calculated using the concatenated whole chromosome per cell line, which is referred to as Pearson's r (whole), or per sequence correlation averaged across the test set when specified. The difference between these metrics manifests as a different mean in the correlation calculation; a global mean for whole chromosome versus a per sequence mean. Whole chromosome evaluation is calculated by concatenating the predictions for the entire chromosome 8 with the exception of unmappable regions. A per sequence Pearson correlation was calculated for peak-centered data, test selection analysis, and robustness analysis, unless specified otherwise. For a compilation of all model evaluations see Supplementary Data 1.

Scaling predictions. Predictions were scaled to address the large discrepancy between predictions and experimental values for shape-based loss functions (eg. Pearson's r). Though we found that applying it to other losses also yielded slightly better performance. This was accomplished by calculating a global scaling factor per cell line, which is computed as the ratio of the mean of experimental and predicted coverage values across the entire test chromosome, and multiplying the scaling factor to the predictions.

Models

Basenji. Basenji-based model is composed of a convolutional block, max-pooling with pool size of 2 (which shrinks the representations to 1024), 11 residual blocks with dilated convolutional layers, followed by a final convolutional layer. The convolutional block consists of: GELU activation(Hendrycks and Gimpel, 2016), convolutional layer with a kernel of width 15, batch normalization (Ioffe and Szegedy, 2015). The residual block is composed of: GELU activation, dilated convolutional layer with a kernel of width 3 and half the number of filters and a dilation rate that grows by a factor of 1.5 in each subsequent residual block, batch normalization, GELU activation, convolutional layer with width 1 and the original number of filters, batch normalization, and dropout with a rate 0.3. Each residual block has a skipped connection to the inputs to residual block. An average pooling layer is applied to the final output convolutional layer to shrink the representations to the corresponding target resolution. A dense layer with softplus activations following the last convolutional block then outputs the predictions target. In case of base resolution, the first max-pool size is set to 1.

The original Basenji model employs multiple convolutional blocks with a max pooling of size 2 to reduce the dimensions of the sequence to 1024 units, upon which 11 residual blocks are applied. Since our input size is 2048 bps, we employed a single convolutional block to achieve the same dimensions as the original Basenji model. We performed hyperparameter search of the number of convolutional filters in each layer to optimize for the ATAC-seq dataset used in this study. For additional details of specific hyperparameters, see Supplementary Data 2.

BPNet. BPNet consists of a convolutional layer, followed by 9 dilated convolutional layers with progressively increasing dilation rates (scaled by powers of 2) that each have a residual connection to the previous layer. Task-specific output-heads, each with a separate transpose convolution, is built upon the final residual layer. To adapt BPNet to lower resolutions, all predictions are initially made at base-resolution followed by an average pooling layer for each task-specific output-head, with a window size and stride that matches the target resolution.

A key difference with the original BPNet architecture is that the negative strand, bias track and read counts output-head was not used throughout this study. Moreover, the original loss function was not employed here as we found better success with the modified BPNet using a Poisson NLL loss. This may be attributed to the lower resolution in read coverage for bulk ATAC-seq, or due to original model targeting raw read count instead of fold change over control tracks, though further investigation is needed to understand the disparity. These modifications may have affected the performance of BPNet. We optimized hyperparameters of the model, focusing on the number of filters in each layer and the kernel size of the transpose convolution in the task-specific output heads (Supplementary Fig. 1). The specific choices of hyperparameters in BPNet can be found in Supplementary Data 2.

CNN-baseline. The CNN baseline model is composed of 3 convolutional blocks, which consist of a 1D convolution, batch normalization, activation, max pooling and dropout, followed by 2 fully-connected blocks, which includes a dense layer, batch normalization, activation, and dropout. The first fully connected block scales down the size of the representation, serving as a bottleneck layer. The second fully-connected block rescales the bottleneck to the target resolution. This is then reshaped to match the number of bins $\times$ 8. For instance, the number of hidden units for models at 32 bin target resolution are $2048/32 = 64 \times 8$, then reshaped to (64, 8). Base resolution models set the hidden units to 2048×8 then reshaped to (2048, 8). This is followed by another convolutional block. The representations from the outputs of the convolutional block is then input into task-specific output heads or is directly fed to a linear output layer with softplus activations. For task-specific output heads, each head consists of a convolutional block followed by a linear output layer with softplus activations. The activation of the first layer is either exponential or ReLU, while the rest of the hidden layer activations are ReLU. The specific hyperparameters of each layer, including the dropout rates, are specified in detail in Supplementary Data 2.

ResidualBind-base. ResidualBind-base builds upon the CNN-baseline models by adding a residual block after the first 3 convolutional layers. The first two residual blocks consist of 5 dilated convolutional layers and the third residual block consists of 4 dilated convolutional layers. Similar to CNN-baseline models, this is then followed by 2 fully connected blocks, which are reshaped to a shape (2048, 8), and a convolutional block. Here, another residual block that consists of 5 dilated convolutional layers was applied. This is then fed into an output head, which has the same composition as the CNN-baseline. The details of model architecture and hyperparameters can be found in Supplementary Data 2.

ResidualBind-32. ResidualBind-32 also builds upon the CNN-baseline models by adding a residual block after the first 3 convolutional layers, but with a few key differences from ResidualBind-base. The third residual block consists of 3 dilated convolutional layers instead of 4. Moreover, ResidualBind-32 does not go through a bottleneck layer that is prototypical of the CNN-baseline design. For task-specific output heads, the representations of the third residual block are input into a convolutional block followed by a task-specific output heads similar to the CNN-baseline models. For a single output head, the representations of the third residual block are input into a position-wise fully connected block followed by a linear output layer. The details of model architecture and hyperparameters can be found in Supplementary Data 2.

Binary models Four main model structures are used for binary models. One fine-tuned Basset(Kelley, Snoek, and Rinn, 2016) structure and three re-purposed quantitative models structures: Basenji, CNN-base, and ResidualBind-base. Basset is composed of three blocks of convolutional layer followed by batch normalization, activation and max-pooling. The output is then flattened and fed into 2 fully connected layers with dropout and an output layer with sigmoid activations. Basset hyperparameters were optimized for the binary version of the ATAC-seq dataset in a similar manner to Basenji. For Basenji-binary, CNN-binary and ResidualBind-binary, their structure highly resembles the quantitative model based on a single output-head. For CNN-binary and ResidualBind-binary, we apply a fully connected output layer with sigmoid activations to the bottleneck layer. For Basenji-binary, we take the penultimate representation and perform a global average pool, followed by a fully connected output layer. The details of model architecture and hyperparameters can be found in Supplementary Data 2.

Training. Each quantitative model was trained for a maximum of 100 epochs using ADAM(Kingma and Ba, 2014) with default parameters. Early stopping was employed with a patience set to 6 epochs (monitoring validation loss as a criterion). By default, models were trained with random reverse-complement and random shift data augmentations unless specified otherwise.

Quantitative CNN and ResidualBind (base and 32 bin-resolution), along with binary versions of these models, were trained for a maximum of 100 epochs using ADAM with default parameters. Early stopping with a patience of 10 was used. The initial learning rate was set to 0.001 and decayed by a factor of 0.2 when the loss function did not improve after a patience of 3 epochs.

Data augmentations

Random shift. Random shift is a data augmentation that randomly translates the input sequence (and corresponding targets) online during training. All datasets were generated with input size set to 3,072 bp. When random shift is used, for each mini-batch, a random sub-sequence of 2,048 bp and its corresponding target profile was selected separately for each sequence. When random shift is not used, the central 2,048 bp is selected for all sequences in the mini-batch.

Reverse-complement. Reverse-complement data augmentation is employed online during training. During each mini-batch, half of training sequences were randomly selected and replaced by their reverse-complement sequence. For those sequences that were selected, the training target was correspondingly replaced by the reverse of original coverage distribution.

Hyperparameter search

The ATAC-seq datasets in this study differ greatly in complexity, i.e. size and coverage distribution, from the original Basenji and BPNet studies. Therefore, we performed a hyperparameter search for each base architecture for our ATAC-seq dataset (Supplementary Fig.B.1). We used WandB(Biewald, 2020) to keep track of the model choices and for visualization. We fine-tuned Basenji and BPNet at 128 bp and base resolution, respectively, which represent the original resolutions for these models. We also kept their original training set selection strategy, that is, we trained Basenji on coverage-threshold data and BPNet on peak-centered data. For Basenji, the number filters across the convolutional layers were varied as well as the presence or absence of dropout layers (fixed rate for each layer). For BPNet, we performed a hyperparameter search over the number of convolutional filters in each layer and the kernel size in the task-specific output heads. We employed the original data augmentations (i.e. random reverse-complement and random shifts for Basenji and only random reverse-complement for BPNet). For each model, we used the Poisson NLL loss function. We originally used a MSE and multinomial NLL loss for BPNet, but found that optimization using Poisson NLL yielded better performance. The models were trained for maximum of 40 epochs with an Adam optimizer (Kingma and Ba, 2014) using default parameters. Initialization was given according to Ref.(He et al., 2016). The optimal set of hyperparameters for each model was selected based on the lowest validation loss and the final architectures are given in Supplementary Data 2.

Robustness test

To measure the robustness to translational perturbations, we analyzed the sequences within the held-out test chromosome that were identified to contain a statistically significant peak for the given cell-type under investigation. This ensures that the robust predictions are only considered for genomic regions that exhibit statistically significant coverage values. Specifically, we took each 3072 bp sequence in the dataset and generated 20 contiguous sub-sequences of length 2048 bp. Each sub-sequence was sent through the model to get a prediction, and all of the predictions were aligned based on the sub-sequence. All sub-sequences contain a center 1024 bp window that overlaps. Standard deviation is calculated for each position across these 20 sequences and averaged across the length of prediction. The average sequence coverage across 20 sequences was used to normalize the average standard deviation to make it invariant to scale. Therefore variation score for each sequence is calculated as average per position standard deviation divided by average sequence coverage. A higher variation score corresponds to a less robust model, while a lower variation score corresponds to more stable predictions, irrespective of translations to the inputs. Due to binning artifacts, we only compare this robustness test for models that share the same bin-resolution.

Variant effect prediction

Dataset. The CAGI5 challenge dataset was used to benchmark model performance on variant calling. Each regulatory element ranges from 187bp - 600bp in length. We extracted 2048 bp sequences from the reference genome centered on each regulatory region of interest and converted it into a one-hot representation.

Alternative alleles are then substituted correspondingly to construct the CAGI test sequences.

Standard predictions. For a given model, the prediction of 2 sequences, one with a centered reference allele and the other with an alternative allele in lieu of the reference allele, is made and the coverage values are summed separately for each cell-type. For each sequence, this provides a single value for each cell-type. The cell-type agnostic approach employed in this study then uses the mean across these values to calculate a single coverage value. The effect size is then calculated with the log-ratio of this single coverage value for the alternative allele and reference allele, according to: $\log$(alternative coverage / reference coverage).

Robust predictions. For a given model, robust predictions were made by: 1) sampling 20 randomly shifted sequences centered on a variant-of-interest, 2) sending them through the model to get coverage predictions for each cell-type, 3) align predictions based on the shifted sub-sequences, 4) calculating the mean coverage within overlapping 1024 bp region for each cell-type, and 5) averaging the mean coverage values across cell-type. This was done separately for the reference allele and the alternative allele, and the effect size was calculated similar to the standard predictions as the log-ratio.

Evaluation. To evaluate the variant effect prediction performance, Pearson correlation was calculated within each CAGI5 experiment between the experimentally measured and predicted effect size. The average of the Pearson correlation across all 15 experiments represents the overall performance of the model. A full list of variant effect prediction performances for models can be found in Supplementary Data 3.

Model interpretability

Tomtom. The motif comparison tool Tomtom (Gupta et al., 2007) was used to match the position probability matrix of the first convolutional layer filters (calculated via activation-based alignments(Alipanahi et al., 2015a)) to the 2022 JASPAR nonredundant vertebrates database (Castro-Mondragon et al., 2021). Matrix profiles MA1929.1 and MA0615.1 were excluded from filter matching to remove poor quality hits; low information content filters then to have a high hit rate with these two matrix profiles. Hit ratio is calculated by measuring how many filters were matched to at least one JASPAR motif. Average q-value is calculated by taking the average of the smallest q-value for each filter among its matches.

Attribution analysis. Attribution analysis was based on grad-times-input with saliency maps (Simonyan, Vedaldi, and Zisserman, 2013). For a given model, gradients of the prediction with respect to a given cell-type were calculated with respect to the input sequence to yield a $L \times A$ map, where L is the length of the sequence and A is 4 – one for each nucleotide. Each saliency map was multiplied by the input sequence, which is one-hot, to obtain just the sensitivity of the observed nucleotide at each position. A sequence logo was generated from this by scaling the heights of the observed nucleotide, using Logomaker (Tareen and Kinney, 2020).

Global importance analysis. For global importance analysis(Koo et al., 2021), we generated background sequences by performing a dinucleotide shuffle of 1,000 randomly sampled sequences from those within our coverage-threshold test set. The global importance is calculated via the average difference in predictions of background sequences with embedded patterns-under-investigation and without any embedded patterns. For quantitative models, the predictions represent the average coverage predictions for the cell-type under investigation. For binary models, the predictions represent the logits for the cell-type under investigation.

GIA for flanking nucleotides. We fixed the core motif at the center of all background sequences, i.e. starting at position 1024, and varied the 2 flanking nucleotides on each side (and the central nucleotide for only AP-1) by separately performing a GIA experiment for all possible combinations of flanking nucleotides.

GIA for distance-dependent motif interactions. To quantify the functional dependence of the distance between 2 motifs with optimized flanks, we fixed the position of 1 motif at the center of the sequence, i.e. starting at position 1024, and then systematically performed a GIA experiment with the second motif at different locations ensuring no overlap. This experiment provides a global importance score for the 2 motifs at different distances in both positive and negative directions.

GIA for motif cooperativity. To quantify whether motifs are cooperatively interacting, we inserted each motif (with optimized flanks) at the corresponding position (1024 for motif 1 and best position for interaction for motif 2 based on the distance-dependent GIA experiments) individually and in combinations. We then compared the global importance when both motifs are embedded in the same sequence versus the sum of the global importance when only one motif is embedded.

Occlusion-based experiments. We randomly sampled 10,000 sequences from those within our coverage-threshold test set. We performed a string search looking for exact matches to the core motif of AP-1, i.e. TGA-TCA, where the - can be any nucleotide. For each cell-type, we grouped the sequences according to the number of instances that the core AP-1 motif was observed – 1 observed motif, 2 observed motifs, and 3 or more observed motifs. For each group, we replaced the core motif with randomized sequences. Due to spurious patterns from randomized sequences, we performed a GIA experiment where 25 randomized sequences were embedded in lieu of the core binding site and the model predictions were averaged

– first across the coverage for the cell-type under investigation, then across the 25 randomized sequences. This effectively marginalizes out the impact of the motif for a given sequence. This occlusion-based (or conditional) GIA experiment was done for each sequence in each group.

Chapter 3

Evaluating the representational power of pre-trained DNA language models for regulatory genomics

The emergence of genomic language models (gLMs) offers an unsupervised approach to learn a wide diversity of *cis*-regulatory patterns in the non-coding genome without requiring labels of functional activity generated by wet-lab experiments. Previous evaluations have shown pre-trained gLMs can be leveraged to improve prediction performance across a broad range of regulatory genomics tasks, albeit using relatively simple benchmark datasets and baseline models. Since the gLMs in these studies were tested upon fine-tuning their weights for each downstream task, determining whether gLM representations embody a foundational understanding of *cis*-regulatory biology remains an open question. Here we evaluate the representational power of pre-trained gLMs to predict and interpret cell-type-specific functional genomics data that span DNA and RNA regulation. Our findings suggest that current gLMs do not offer substantial advantages over conventional machine learning approaches that use one-hot encoded sequences. This work highlights a major limitation with current gLMs, raising potential issues in conventional pre-training strategies for the non-coding genome.

3.1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in natural language processing (Devlin et al., 2018; OpenAI, 2023; Touvron et al., 2023; Wei et al., 2022) and protein sequence analysis (Rives et al., 2021; Elnaggar et al., 2021; Madani et al., 2020; Bepler and Berger, 2021). These LLMs, often termed "foundation models", are trained through self-supervised learning to encode input data as contextual embeddings (also known as representations). The strength of pre-trained LLMs lies in the versatility of their embeddings, which can be leveraged for a broad spectrum of downstream predictive tasks. For instance, representations from pre-trained protein language models have been used to predict protein structures (Lin et al., 2022b; Chowdhury et al., 2022; Wu et al., 2022), predict non-synonymous variant effects (Brandes et al., 2023; Meier et al., 2021), design novel protein sequences (Madani et al., 2023; Ferruz and Höcker, 2022; Hie et al., 2023), and study protein evolution (Hie, Yang, and Kim, 2022; Zhang et al., 2024).

LLMs pre-trained on DNA sequences offer a promising new paradigm to accelerate our understanding of functional elements in the non-coding genome (Consens et al., 2023). Genomic language models (gLMs) could, in principle, help to understand the complex coordination of transcription factors (TFs) to control the activity of *cis*-regulatory elements (CREs). They might also enable more accurate predictions of the functional consequences of non-coding mutations, which can help to prioritize diease-associated variants. Additionally, gLMs capable of learning *cis*-regulatory rules could become instrumental in designing novel regulatory sequences with desirable functional properties. They might also facilitate functional comparisons of non-coding sequences across different species, a task currently complicated due to substantial evolutionary drift in non-coding regions.

Recently, there has been a surge of pre-trained gLMs (Benegas, Batra, and Song, 2023; Nguyen et al., 2023; Lal, Biancalani, and Eraslan, 2023; Dalla-Torre et al., 2023; Ji et al., 2021; Zhang et al., 2023; Zhou et al., 2023; Sanabria, Hirsch, and Poetsch, 2023b; Karollus et al., 2023; Chu et al., 2023; Chen et al., 2023; Shen and Li, 2024; Zaheer et al., 2020; Fishman et al., 2023; Benegas et al., 2023; Hallee, Rafailidis, and Gleghorn, 2023; Li et al., 2023; Gündüz et al., 2023; Yang et al., 2022; Chen et al., 2022a; Zvyagin et al., 2023; Levy et al., 2022; Liang et al., 2023; Gündüz et al., 2023; Gu and Dao, 2023; Liu et al., 2024; Outeiral and Deane, 2024; Schiff et al., 2024). gLMs take as input DNA sequences that have undergone tokenization, an encoding scheme applied to either a single nucleotide or k-mer of nucleotides. Through self-supervised pre-training, the gLM learns a vector representation for each token in the DNA sequence via masked language modeling (MLM)(Devlin et al., 2018) or causal language modeling (CLM) (Radford et al., 2019). In a standard setting of MLM, a portion of the input tokens, typically 15%(Devlin et al., 2018), is randomly masked, and the task is to predict the masked tokens using the context provided by the rest of the unmasked tokens in the sequence. On the other hand, CLM is an autoregressive pre-training task where the goal is to predict the next token in a sequence given the previous tokens. These language modeling objectives result in learning self-supervised representations of the sequence that capture information about individual tokens and the complex interrelationships between other tokens in the sequence. The burden of learning biologically meaningful features is paid upfront during the pre-training. Afterward, the gLM's representations can be leveraged for a broad spectrum of downstream prediction tasks as inputs to simpler models, bypassing the need to learn essential features for each task from scratch. In contrast, the conventional one-hot representation of DNA sequences treats each element independently, assigning an identical representation for the same nucleotide characters irrespective of their positions in the sequence or what context is nearby. Consequently, the responsibility of extracting important features falls solely on the machine learning model being employed.

Current gLMs are composed of different choices for the tokenization, base architecture, language modeling objective, and pre-training data. *Tokenization* of DNA sequences is employed for either single nucleotide (Benegas, Batra, and Song, 2023; Nguyen et al., 2023; Lal, Biancalani, and Eraslan, 2023) or k-mer of fixed size (Dalla-Torre et al., 2023; Ji et al., 2021; Zhang et al., 2023) or a k-mer of variable sizes via byte-pair tokenization (Sennrich, Haddow, and Birch, 2015; Zhou et al., 2023; Sanabria, Hirsch, and Poetsch, 2023b), which aims to aggregate DNA in a manner that reduces the k-mer bias in the genome, a problem known as rare token imbalance. The *base architecture* is typically a stack of transformer layers (Vaswani et al., 2017), with a vanilla multi-head self-attention(Ji et al., 2021; Dalla-Torre et al., 2023; Karollus et al., 2023; Chu et al., 2023; Chen et al., 2023; Zhang et al., 2023; Shen and Li, 2024; Sanabria, Hirsch, and Poetsch, 2023b) or an exotic attention variant (e.g., flash attention (Dao et al., 2022; Zhou et al., 2023), sparse attention (Zaheer et al., 2020; Fishman et al., 2023), or axial attention (Ho et al., 2019; Benegas et al., 2023)). Alternatively, the base architecture has also been constructed with a stack of residual-connected convolution blocks, either with dilated convolutional layers (Benegas, Batra, and Song, 2023) or state-space models, such as a Hyena (Poli et al., 2023; Nguyen et al., 2023; Lal, Biancalani, and Eraslan, 2023) or Mamba (Gu and Dao, 2023; Schiff et al., 2024). The *pre-training data* can vary significantly, encompassing the whole genome of a single species (Benegas, Batra, and Song, 2023; Ji et al., 2021; Zaheer et al., 2020) or the whole genomes across multiple species(Dalla-Torre et al., 2023; Zhou et al., 2023;

Fishman et al., 2023; Karollus et al., 2023; Zhang et al., 2023) or focused only within specific regions of the genomes, such as the untranslated regions (UTRs) (Chu et al., 2023), pre-mRNA (Chen et al., 2023), promoters (Lal, Biancalani, and Eraslan, 2023), coding regions (Hallee, Rafailidis, and Gleghorn, 2023; Li et al., 2023; Outeiral and Deane, 2024), non-coding RNA (Chen et al., 2022a; Penić et al., 2024), or conserved sites (Benegas et al., 2023).

Notably, Nucleotide-Transformer (Dalla-Torre et al., 2023) is a collection of BERT(Devlin et al., 2018)-style models that consider non-overlapping k-mer tokenization and is pre-trained via MLM on either a single human genome, a collection of 3,202 human genomes from the 1000 Genomes Project(Consortium et al., 2015) alone or in combination with 850 genomes across diverse species. DNABERT2 (Zhou et al., 2023) is also a BERT-style architecture but uses flash attention, considers byte-pair tokenization, and is trained via MLM on the genomes of 850 species. Genomic Pre-trained Network (GPN) is a convolution-based model with a stack of residual-connected dilated convolutions, uses single-nucleotide tokenization, and is trained via MLM on *Arabidopsis thaliana* genome and seven related species within the Brassicales order (Benegas, Batra, and Song, 2023). Similarly, HyenaDNA (Nguyen et al., 2023) is a state-space model using Hyena layers, single-nucleotide tokenization, and is trained via CLM on the human reference genome. Supplementary Table A.6 summarizes the unique combination of components that comprise other gLMs.

The utility of gLMs pre-trained on whole genomes for studying the non-coding genome has been limited. Previous benchmarks have largely considered gLMs that have been fine-tuned (i.e., adjusting the weights of the gLM) on each downstream task (Ji et al., 2021; Dalla-Torre et al., 2023; Chen et al., 2022a; Chen et al., 2023; Zhou et al., 2023). In each benchmark, a fine-tuned gLM has demonstrated improved predictions on a host of downstream prediction tasks, often based on the classification of functional elements, such as histone marks or promoter annotations. However, the chosen benchmarks do not reflect the complexity of *cis*-regulatory mechanisms observed in gene regulation, and the baseline models used in the comparisons often do not represent the state-of-the-art. Hence, the capabilities of gLMs in understanding the regulatory genome have yet to be demonstrated in a fair assessment.

The reliance on fine-tuning poses challenges, as foundation models are typically large, and fine-tuning on individual tasks demands substantial GPU resources, which may not be readily available to academic labs. Although parameter-efficient fine-tuning methods have emerged, such as LoRA (Low Rank Adaptation)(Hu et al., 2021; Zhou et al., 2023; Zhan, Wu, and Zhang, 2024), (hard or soft) prompt tuning (Lester, Al-Rfou, and Constant, 2021; Nguyen et al., 2023), and $(IA)^3$ (Liu et al., 2022a; Dalla-Torre et al., 2023), fine-tuning makes it challenging to assess the contribution of the prior knowledge gained via pre-training on each downstream task. Moreover, benchmarks that do not fine-tune gLMs are limited in their downstream tasks (Marin et al., 2023; Robson and Ioannidis, 2023; Vilov and Heinig, 2024), relying on either binary classification of functional activity, which does not reflect the complexity of cis-regulatory biology (Toneyan, Tang, and Koo, 2022; Nair et al., 2023) or lack a more comprehensive set of benchmarking tasks. Thus, it remains unclear the extent to which existing gLMs pre-trained on whole genomes can genuinely serve as foundation models that can transfer their knowledge to predict and interpret functional genomics data, without necessitating additional fine-tuning of the gLM weights.

Here we perform a focused evaluation to assess the informativeness of

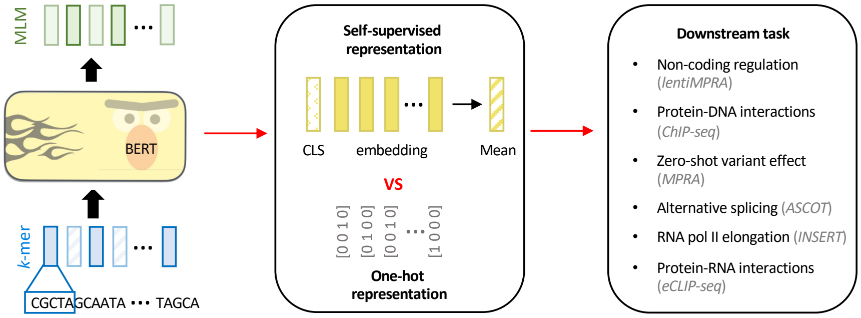

MLM
BERT
k-mer
CGCTAGCAATA ••• TAGCA
Self-supervised representation
CLS
embedding
Mean
VS
[0 0 1 0]
[0 1 0 0]
[0 0 1 0]
[1 0 0 0]
One-hot representation
Downstream task
Non-coding regulation (lentiMPRA)
Protein-DNA interactions (ChIP-seq)
Zero-shot variant effect (MPRA)
Alternative splicing (ASCOT)
RNA pol II elongation (INSERT)
Protein-RNA interactions (eCLIP-seq)

FIGURE 3.1: Experimental overview. Comparison of gLM embeddings versus one-hot representations for various functional genomics prediction tasks.

learned representations by various gLMs pre-trained on whole genomes (without fine-tuning) for six major functional genomics prediction tasks, which encompass different levels of *cis*-regulation complexity at DNA and RNA levels (see Fig. 3.1). In particular, we compared the predictive power of representations from pre-trained gLMs – namely Nucleotide-Transformer, DNABERT2, HyenaDNA, and a custom GPN pre-trained on the human reference genome – versus one-hot encoded DNA and representations acquired from a supervised "foundation model" pre-trained on a large corpus of functional genomics data. Our results suggest that current gLMs pre-trained on whole genomes do not provide noticeable advantages over conventional approaches to analyzing human functional genomics with deep learning using one-hot sequences. Moreover, supervised foundation models pre-trained on functional genomics data appear to encapsulate more relevant information and transfer better to other functional genomics data, albeit when the source pre-training tasks and the target tasks are closely aligned. Our results suggest that the standard pre-training schemes for current gLMs struggle to understand cell-type specific functional elements and, therefore, fall short of achieving a foundation model status for the non-coding genome of humans.

3.2 Results

3.2.1 Task 1: Predicting cell-type specific regulatory activity from lentiMPRA data

Understanding the mechanisms that drive CRE activity is a major goal in functional genomics; it is challenging due to complex rules of cell-type-specific TF binding (Shlyueva, Stampfel, and Stark, 2014; Zeitlinger, 2020). In the first

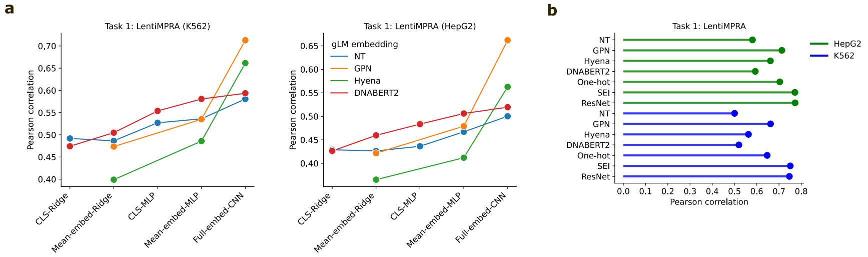


FIGURE 3.2: Test performance on predicting cell-type-specific regulatory activity from lentiMPRA data. **a**, Comparison of prediction performance across various downstream machine learning models, including ridge regression and MLP using either the gLM's CLS token or mean embedding, and a CNN trained using the full embedding of the penultimate layer of gLMs. **b**, Prediction performance using a baseline CNN trained using different gLM embedding inputs, one-hot sequences, or supervised embeddings from Sei. ResNet represents the performance of a more sophisticated model that is trained using one-hot sequences.

task, we compared the performance of various machine learning models that consider different input representations of DNA sequences at predicting experimentally measured enhancer activity via lentiMPRA (lentiviral Massively Parallel Reporter Assay) (Agarwal et al., 2023). Specifically, this task involves taking a 230 nucleotide (nt) DNA sequence as input, represented either as a gLM embedding or one-hot sequence, and predicting a scalar value that represents the CRE's functional activity measured in a given cellular context via lentiMPRA (see Methods). This task enables a direct comparison in performance across matched downstream models for each sequence encoding scheme. By considering two cell types, namely HepG2 and K562, we can assess whether pre-trained gLM representations capture cell-type-specific CRE activity.

For each gLM, we considered the embeddings from the penultimate layer using a linear model or multi-layer perceptron (MLP) based on the classification token (CLS) or the mean embedding, which is standard practice for harnessing sequence summarization of LLM embeddings. We also employed a baseline convolutional neural network (CNN) that analyzed the full embeddings of the penultimate layer as well as one-hot sequences for comparison (see Methods). We also considered embeddings from the penultimate layer of Sei (Chen et al., 2022b), a supervised foundation model pre-trained on 21,907 chromatin profiling datasets across over 1,300 cell lines and tissues. To assess the performance against a more sophisticated supervised model, we trained a ResidualBind(Koo et al., 2021)-like model (ResNet) using one-hot sequences. These choices provide a fair benchmark to assess whether embeddings from foundation models, acquired via unsupervised gLMs or supervised CNNs, are more informative for downstream models than naive one-hot sequences.

We found that CNNs trained on the whole sequence embedding led to improved performance over the linear or MLP models that analyzed CLS or mean embeddings (Fig. 3.2a). This suggests that summarized gLM representations lack sufficient information to predict cell-type specific regulatory activity, whereas CNNs can build upon the full embeddings to better discriminate cell-type specific features. Moreover, the performance gap between MLPs and linear models suggests that the mapping between the pre-trained representations and the functional readouts of lentiMPRA data is highly non-linear.

We also observed that CNNs trained using sequence embeddings from gLMs generally under-performed standard one-hot sequences, with the exception of our custom-trained GPN (Fig. 3.2b). Notably, the performance of all gLM-based representations were significantly lower than the supervised representations given by Sei. Due to differences in the data splits for Sei, it is unclear to what extent data leakage might lead to performance inflation. Nevertheless, the ResNet model trained using one-hot sequences on the LentiMPRA dataset also achieved high performance (Fig. 3.2b). These results suggest that gLM embeddings may not provide beneficial context for CREs that cannot already be learned from one-hot sequences for the lentiMPRA dataset.

To control for the possibility that gLM embeddings from the penultimate layer may not be optimal, we performed the same analysis using embeddings from other layers of Nucleotide-Transformer. While some layers yielded modest improvements, particularly layer 10, the overall trends held and thus did not change the conclusions (Supplementary Fig. C.1).

3.2.2 Task 2: Predicting TF binding sites from ChIP-seq data

Since TF binding is a cell-type-specific phenomenon, but standard language modeling objectives are not cell-type aware, we surmised that the low performance of gLMs on the lentiMPRA prediction task may be due to losing information about key motifs during the pre-training. To test this hypothesis, we evaluated whether the gLM embeddings can predict TF binding sites measured via ChIP-seq (Chromatin Immuno-Precipitation sequencing(Ren et al., 2000)). Briefly, this task is framed as a binary classification where a model takes a 200 nt DNA sequence, either as a gLM embedding or a one-hot sequence, as input and predicts whether the sequence corresponds to a ChIP-seq peak. We consider ten ChIP-seq datasets spanning different TFs in GM12878 cells; a separate single-task model was trained for each TF (see Methods).

Evidently, CNNs trained using one-hot sequences modestly outperformed the whole embeddings from DNABERT2, HyenaDNA, and Nucleotide-Transformer. On the other hand, the custom GPN occasionally led to improved performance (Fig. 3.3). Nevertheless, the performance differences across all sequence encoding schemes were modest, suggesting that gLMs do not appear to lose TF-related information in their embeddings. However, it is unclear whether the information provided by gLM embeddings actively encodes for TF motifs or whether the embeddings are simply not losing essential information about the input sequence from which a downstream CNN can learn TF binding information directly from the gLM embeddings, similar to one-hot sequences.

As a control experiment, we trained MLP or linear models using the CLS token of Nucleotide-Transformer. In this way, any information about motifs must be fully encoded in these summarized embeddings. We observed that CNNs trained on the whole embedding yielded substantially higher performance than an MLP

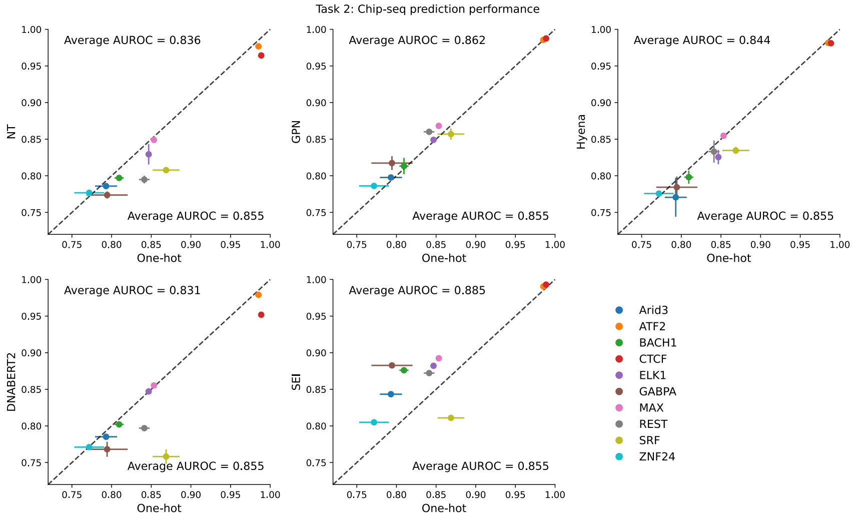


FIGURE 3.3: Performance comparison on TF binding prediction tasks from ChIP-seq data. Comparisons of CNNs trained using different gLM embeddings versus CNNs trained using one-hot sequences for 10 TF ChIP-seq datasets. Performance is measured by the average area-under the receiver-operating characteristic curve (AUROC) and error bars represent the standard deviation of the mean across 5 different random initializations. Average AUROC represents the average performance across all ChIP-seq datasets.

trained using the CLS token (Supplementary Fig. C.2a). Nevertheless, the MLP still demonstrated aptitude in predicting TF binding overall. To rule out the possibility that biases in the dataset create a trivial prediction task, where low-level sequence statistics can be used to discriminate class labels, we also trained an MLP model on bag-of-dinucleotide frequencies. Indeed, the MLP based on dinucleotide frequencies yielded comparable performance to the CLS token (Supplementary Fig. C.2a), except for CTCF, a protein that plays an important role in chromatin structure for all cell types. Together, these results suggest that gLMs do not appear to lose TF-related information in their embeddings, albeit only a slight information boost is gained regarding TF binding compared to low-level dinucleotide statistics. Nevertheless, downstream models that analyze conventional one-hot sequences can easily rectify any information deficiencies, leading to higher performances.

3.2.3 Task 3: Zero-shot variant effect prediction with MPRA data

A major use case of highly accurate sequence-function models is their ability to predict the functional consequences of non-coding mutations (Avsec et al., 2021c). In previous studies, Nucleotide-Transformer and GPN have demonstrated an ability to predict single-nucleotide variant effects, albeit as part of a binary classification task (Dalla-Torre et al., 2023; Benegas, Batra, and Song, 2023). However, it is not intuitive how gLMs pre-trained on whole genomes could yield good zero-shot predictions of cell-type-specific variant effects in the non-coding region of human genomes since they are trained without any cell-type information. Thus, we assessed the ability of gLMs, specifically Nucleotide-Transformer, GPN, and HyenaDNA, to quantitatively predict single-nucleotide variant effects within CREs using saturation mutagenesis data measured via MPRAs (Massively Parallel Reporter Assay)(Kircher et al., 2019). This task involves calculating the zero-shot

Table 3.1: Zero-shot variant effect generalization on CAGI5 dataset. The values represent the Pearson correlation between the variant effect predictions with experimental saturation mutagenesis values of a given CRE measured via MPRAs. Values are reported for a single CRE experiment for K562 and the average of three CRE experiments for HepG2.

Training task	Model	Variant effect prediction	HepG2	K562
self-supervised pre-training	NT (2B51000G)	cosine distance	0.125	0.007
	NT (2B5Species)	cosine distance	0.112	0.135
	NT (500MHuman)	cosine distance	0.020	0.088
	NT (500M1000G)	cosine distance	0.041	0.068
	GPN (human)	log2-ratio	0.002	0.037
	HyenaDNA	cosine distance	0.064	0.021
lentiMPRA-embedding	CNN-GPN	difference from wild type	0.377	0.457
	CNN-NT	difference from wild type	0.137	0.240
	CNN-SEI	difference from wild type	0.559	0.719
lentiMPRA-one-hot	CNN	difference from wild type	0.313	0.426
	ResidualBind	difference from wild type	0.486	0.551
	MPRAnn	difference from wild type	0.301	0.369
supervised one-hot	SEI	cosine distance	0.545	0.641
	Enformer (DNase)	difference from wild type	0.510	0.685

variant effect predictions of gLMs either by the cosine similarity of embedding vectors for the input sequence with mutant or wild-type allele (e.g. Nucleotide-Transformer and Hyena) or the log2-ratio of predicted variant and wild-type nucleotide via single-nucleotide masking (e.g. GPN). These variant effect scores are compared with experimentally measured variant effects according to the Pearson correlation coefficient (see Methods). This analysis includes MPRA measurements for three CREs in HepG2 cells and one CRE in K562 cells as part of the CAGI5 challenge(Kircher et al., 2019; Shigaki et al., 2019b).

We found that all tested gLMs (without fine-tuning) exhibited poor variant effect predictions in this quantitative zero-shot generalization task (Table 3.1). These results extended to all Nucleotide-Transformer models(Dalla-Torre et al., 2023), including a 2.5 billion parameter BERT-based gLM trained on 3,202 diverse human genomes and 850 genomes from various species. On the other hand, CNNs trained on lentiMPRA data using gLM embeddings yielded substantially better performance relative to their pre-trained counterparts (Table 3.1). Moreover, sophisticated models trained using one-hot sequences, such as Enformer(Avsec et al., 2021c), which is a state-of-the-art model trained with supervised learning on a wide variety of functional genomics data using one-hot sequences, and Sei yielded better performance than all CNNs trained using gLM representations. However, the CNN trained using Sei embeddings on the lentiMPRA dataset yielded the best overall performance. Together, these results highlight a major gap in the zero-shot variant effect performance of gLMs with the state-of-the-art.

3.2.4 Task 4: Predicting alternative splicing from RNA-seq data

Previous studies demonstrated that Nucleotide-Transformer and GPN has learned properties related to gene definition and splice sites (Benegas, Batra, and

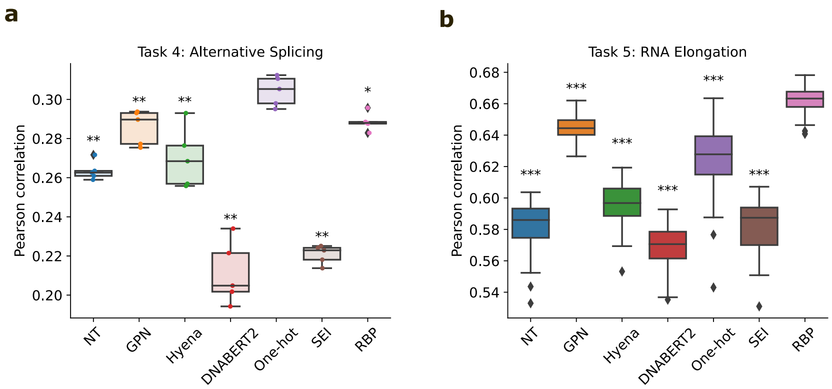


FIGURE 3.4: Performance on RNA regulation tasks. **a**, Box-plot shows average Pearson correlation across tissues on test data for various models trained with different encoding schemes on an alternative splicing prediction task using MTSplice data. **b**, Box-plot shows Pearson correlation for various models trained with different encoding schemes on a RNA poll II elongation potential prediction task using INSERT-seq data. Box-plots show the first and third quartiles, central line is the median, and the whiskers show the range of data. Box-plots represent 5 different random initializations for **a** and 50 different random initializations for **b**. Statistical significance represents the Mann-Whitney U test with a p value < 0.05 (*), < 0.01 (**), and < 0.001 (***).

Song, 2023; Dalla-Torre et al., 2023). Thus, we surmised that gLMs pretrained on whole genomes may be more beneficial for RNA regulation tasks. To investigate this, we tested the informativeness of gLM embeddings to predict mRNA alternative splicing quantified using RNA-seq (RNA-sequencing) from the ASCOT dataset(Ling et al., 2020). Specifically, the prediction task takes as input two sequences – a sequence with 300 nt upstream of the splice acceptor and 100 nt downstream of the acceptor and a sequence with 100 nt upstream of the splice donor and 300 nt downstream of the donor – with the goal of predicting the percentage-spliced-in (PSI) across 56 tissues as a multi-task regression; a task introduced by MTSplice (Cheng et al., 2021). Similar to the DNA analysis, a baseline CNN was trained to take as input the full embeddings from gLMs or the embeddings of a pre-trained supervised model (see Methods).

Our results mirrored those seen for regulatory DNA, with embedding-based models largely under-performing compared to one-hot-based models (Fig. 3.4a). In contrast, Sei's embeddings led to substantially lower performance than most gLM embeddings for this task. This is likely due to Sei's pre-training focus on DNA-based functional genomics data, which leads to learning a set of DNA regulatory features that do not transfer well to RNA regulation. To test whether a more relevant set of features acquired through supervised learning could transfer better for RNA regulation, we trained a multi-task ResidualBind-like model to classify RNA-protein binding (RBP) sites from a large trove of eCLIP-seq data (see Methods). The task is to take 1,000 nt sequences as input and predict binding for 120 RBPs in K562 cells as a multi-task classification. Indeed, the embeddings from this RBP-trained supervised model led to substantially better performance than the gLM embeddings, except GPN, which yielded comparable results (Fig. 3.4a).

3.2.5 Task 5: Predicting RNA pol II elongation potential from INSERT-seq data

Next, we performed a similar analysis for a prediction task that takes 173 nt RNA sequences as input and predicts RNA pol II elongation potential measured via INSERT-seq (INtegrated Sequences on Expression of RNA and Translation using high-throughput sequencing)(Vlaming et al., 2022). The INSERT-seq dataset is modest in size, containing only 10,774 sequences. This small data regime may not provide sufficient examples to learn all relevant patterns using one-hot sequences. Training a large deep learning model on this dataset can easily lead to over-fitting. Thus, this task can help evaluate a scenario (i.e., the low data regime) where a baseline CNN that uses gLM embeddings might have an advantage over one-hot sequences.

Similarly, we found that the baseline CNNs trained using gLM embeddings yielded lower performance than one-hot RNA sequences, except for the custom GPN, which performed slightly better (Fig. 3.4b). Again, the CNN performance based on Sei's supervised embeddings was worse, and the best-performing model was achieved using embeddings from the supervised multi-task model pre-trained to classify RBPs. These results highlight that generic pre-training strategies are not always beneficial; when carefully selecting pre-training tasks, one should consider which relevant features are needed to ensure more positive outcomes on downstream applications.

While the custom GPN was the only embedding that demonstrated improved performance over one-hot sequences, we hypothesized that further downsampling of the training data could lead to situations where gLM embeddings become more beneficial than one-hot sequences. We systematically down-sampled both the alternative splicing and INSERT-seq datasets and retrained the same baseline CNNs using different input encoding schemes. Interestingly, the GPN embeddings consistently outperformed other embeddings (Supplementary Fig. C.3). The improved performance by GPN suggests that gLMs may specialize more effectively in specific genomic regions. Specifically in this dataset, capturing 5' splice sites is a critical feature (Vlaming et al., 2022). Thus, understanding what features gLMs learn well can help to identify suitable downstream tasks for which they can thrive.

3.2.6 Task 6: Predicting RNA-binding protein binding with eCLIP-seq data

RBPs are essential for various RNA processing stages, so next, we examined the ability of gLMs to predict RBP binding sites using eCLIP-seq (enhanced chromatin immunoprecipitation sequencing) datasets (Van Nostrand et al., 2016). Briefly, the task involves taking 200 nt DNA sequences as input and predicting binary labels of whether the sequence corresponds to an eCLIP-seq peak or not (see Methods). Ten eCLIP-seq datasets spanning different RBPs were used in the evaluation. We trained a baseline CNN model using different sequence encoding schemes similar to previous tasks.

We found that CNNs trained using gLM embeddings performed slightly worse on average compared to the one-hot sequences (Fig. 3.5a), in agreement with the ChIP-seq results of Task 2. The narrow performance difference between models using gLM embeddings and one-hot sequences also indicates that RBP motif information is not lost in the gLM embeddings. In a similar control, we found that an MLP based on Nucleotide-Transformer's CLS token led to slightly better performance than an MLP based on dinucleotide frequencies (Supplementary Fig.

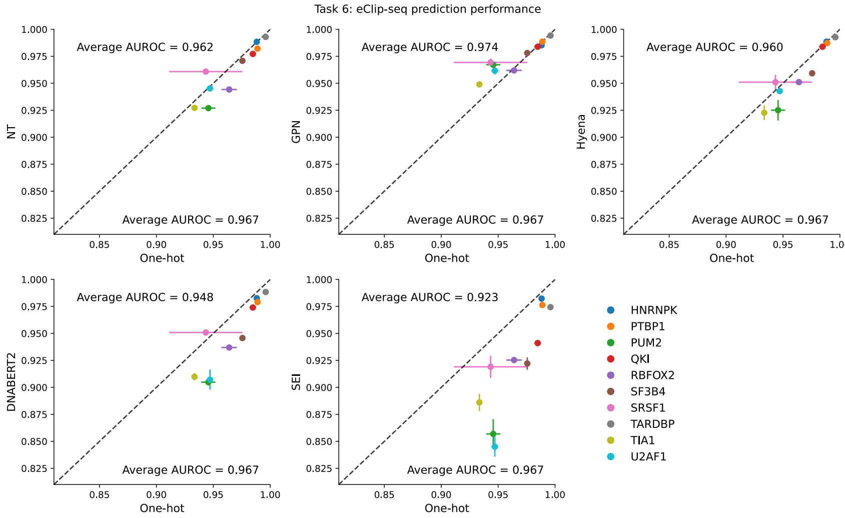


FIGURE 3.5: Performance comparison on RBP binding prediction tasks from eCLIP-seq data. Comparisons of CNNs trained using different gLM embeddings versus CNNs trained using one-hot sequences for 10 RBP eCLIP-seq datasets. Performance is measured by the average area-under the receiver-operating characteristic curve (AUROC) and error bars represent the standard deviation of the mean across 5 different random initializations. Average AUROC represents the average performance across all eCLIP-seq datasets.

C.2b). This supports that gLM embeddings encode beyond low-level sequence statistics in regulatory regions of RNA. Again, we found that Sei embeddings lead to a substantial decline in performance, further highlighting the importance of selecting appropriate pre-training tasks.

3.2.7 Uncovering cell-type-specific motifs learned by gLMs is challenging

As a follow up, we performed attribution analysis to identify motifs that are captured by gLMs. Attribution maps were generated for a given sequence by systematically masking one input token (i.e., a single nucleotide position for GPN and a non-overlapping k-mer for Nucleotide-Transformer) at a time and calculating the entropy over the predicted distribution of the masked token; ΔEntropy, which is the difference between the maximum entropy value across the whole sequence and the entropy values at each position, was used to identify positions that yielded informative nucleotides (see Methods). For comparison, we generated gradient-corrected Saliency Maps (Majdandzic et al., 2023) for a CNN trained using one-hot sequences. The analysis focused on lentiMPRA and CTCF ChIP-seq data to cover tasks from different systems with varying levels of complexity.

As expected, the attribution maps for pre-trained gLMs alone (i.e., not considering the downstream task) were difficult to interpret for both lentiMPRA (Fig. 3.6a) and ChIP-seq data (Supplementary Fig. C.4a). The attribution maps did not reflect any known motifs, nor did they match any of the patterns captured in the CNN's Saliency Maps. This disparity can arise if the probed locus is used for multiple purposes across different cell types. If cell-type-specific *cis*-regulatory patterns are projected onto a single DNA sequence, the overlapping set of motifs can lead to complex attribution maps that may not resemble distinct cell-type-specific

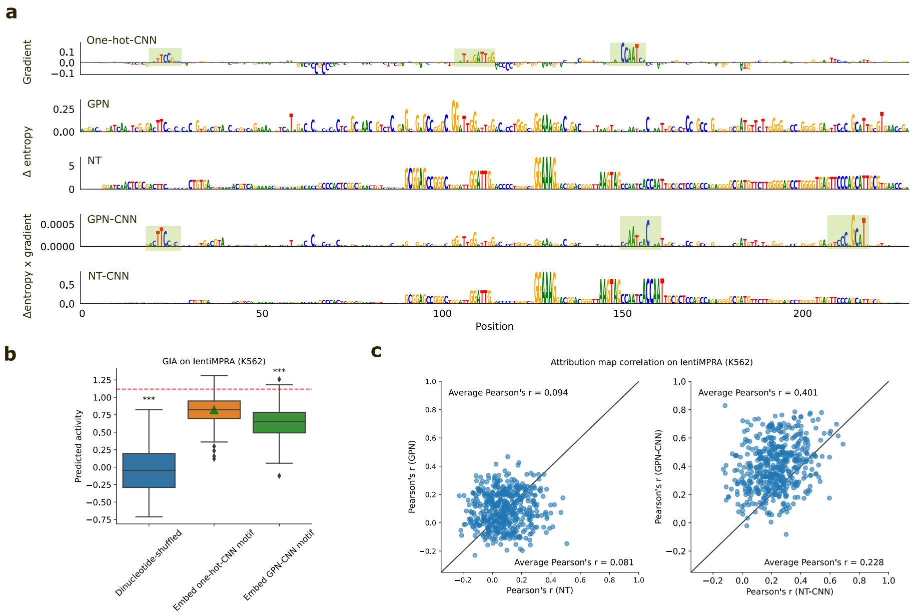


FIGURE 3.6: Attribution analysis comparison for sequences from the lentiMPRA dataset. **a**, Representative example of attribution maps for a regulatory sequence from the lentiMPRA dataset. Attribution maps include (top to bottom): the gradient-times-input of a one-hot-trained CNN; the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained GPN and Nucleotide-Transformer; the gradient of a CNN-trained using corresponding embeddings multiplied by the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained GPN adn Nucleotide-Transformer. **b**, Box-plot of the predicted activity for 300 dinucleotide-shuffled sequences from **a**, dinuc-shuffled sequences with the annotated patterns from the Saliency Map of the one-hot-trained CNN, and dinuc-shuffled sequences with the annotated patterns from the CNN trained using GPN embeddings (GPN-CNN). Green triangle represents the global importance analysis value. Red dashed line represents the prediction of the wild type sequence according to the one-hot-trained CNN. **c**, Scatter plot comparison of the attribution map correlations for different pre-trained gLMs (left) and CNNs trained using gLM embeddings (right). Attribution map correlations reflect the Pearson correlation coefficient between the attribution map generated by the gLM-based attribution method with the Saliency Map generated by a one-hot-trained CNN. Each dot represents a different sequence in the lentiMPRA dataset (N=500).

motifs. Alternatively, the complex patterns that seem to span the length of the sequence could also reflect low-level sequence statistics that are memorized. Without ground truth, interpreting attribution maps remains challenging.

Next, we evaluated attribution maps generated by the downstream CNN that used gLM embeddings as input. Specifically, we scaled the gLM's entropy-based attribution map with the maximum gradients at each position based on the downstream CNN (see Methods). Through a qualitative comparison, we noticed that the attribution maps generated by GPN appear to be visually aligned with Saliency Maps generated by the one-hot-trained CNN compared to Nucleotide-Transformer (Fig. 3.6a), even after accounting for the block-like structure which arises due to the k-mer tokenization. This trend was observed for other loci as well (Supplementary Fig. C.5).

To validate the importance of the putative binding sites identified via Saliency Maps for the one-hot-trained CNN, we employed global importance analysis (GIA) (Koo et al., 2021). Specifically, we embedded the 3 annotated patterns into different dinucleotide-shuffled sequences, which serve as background sequences with low CRE activities, and measured the effect of including the patterns on model predictions. Indeed, GIA shows that the motif patterns identified by Saliency Maps for the one-hot-trained CNN are more-or-less sufficient to explain model predictions (Fig. 3.6b).

We then quantified the correlation between the attribution maps generated by the one-hot-trained CNN and the gLM-based attribution maps. We found that attribution maps generated by pre-trained gLM are not well-aligned with each other, nor the attribution maps generated by the one-hot-trained CNN (Fig. 3.6c, Supplementary Fig. C.2b). By contrast, attribution maps generated by CNNs trained with gLM embeddings led to improved alignment between their attribution maps and with one-hot-trained CNNs. These results suggest that the gLMs learn non-overlapping features during pre-training, but a downstream model can still use them to build cell-type-specific motifs (that are better aligned with motifs learned by one-hot-trained CNNs).

Together, the attribution maps given by pre-trained gLMs seem to visually capture a more diffuse set of patterns, which speculatively reflect low-level statistics of genomic sequences. Downstream models, like CNNs, appear to use these seemingly uninformative gLM embeddings (especially from GPN) to build cell-type-specific regulatory features that are relevant for downstream prediction tasks.

3.3 Discussion

To assess the transferability of knowledge acquired during pre-training for current genome language models for regulatory genomics, we evaluated four gLMs pre-trained on whole genomes (without fine-tuning) across six functional genomics prediction tasks with appropriate baselines for comparison. We found that the gLM representations provide little to no advantage compared to standard one-hot sequences. On a relative basis, we found that GPN, a convolution-based LLM, yielded slightly more informative representations in the non-coding genome compared to highly parameterized BERT-style LLMs. This suggests that stronger inductive biases toward learning relevant features in the model architecture can improve gLMs, albeit modestly.

Notably, we elected to not fine-tune weights of the gLM on each downstream task, which is how gLMs have been previously benchmarked (Ji et al., 2021; Dalla-Torre et al., 2023; Chen et al., 2022a; Chen et al., 2023; Zhou et al., 2023). While gLM performance would likely improve with fine-tuning, the scope of this study was to strictly gauge the knowledge of *cis*-regulatory biology learned during pre-training. The poor performance observed in this study suggests that cell-type-specific *cis*-regulatory mechanisms are predominantly learned during fine-tuning. Our results suggest that the benefit of pre-training gLMs appears to be initializations that are pre-loaded with just a little more information than low-level statistical properties for non-coding genomic sequences.

In previous studies, pre-trained gLMs have found some success by focusing on specific regions of the genome during pre-training or working with simpler organisms with compact genomes (Benegas et al., 2023; Karollus et al., 2023; Nguyen et al., 2024b). For instance, a BERT-based LLM trained strictly in the coding genome can provide more context than only considering amino-acids with protein language modeling (e.g., codon usage) (Outeiral and Deane, 2024; Hallee, Rafailidis, and Gleghorn, 2023; Li et al., 2023). However, our evaluation shows that extending the pre-training task across the whole genome struggles to capture meaningful representations in the non-coding genome.

The performance gap may be due to differences in the structure of the coding regions versus the non-coding regions. To elaborate, protein sequences have a clear start and end with low-level grammars (i.e., secondary structures) and high-level grammars (i.e., protein domains) shared throughout most globular proteins, with structures that are conserved across species. On the other hand, the non-coding genome contains a variety of short sequence motifs that vary broadly in binding affinities and are sparsely located in seemingly random DNA, with usage and rules that vary across loci and cell types. Few non-coding elements exhibit deep conservation that is typical in proteins. The differing selection pressures in the non-coding regions lead to loss of syntenty, which makes it difficult to study sequence and functional conservation. Thus, treating each nucleotide position equally, whether informative or uninformative, makes this a challenging language modeling task. In the non-coding genome, this is tantamount to expecting the LLM to predict predominantly random nucleotides, which, by definition, can only be achieved via memorization. Hence, this may explain why gLMs have also found greater utility in learning *cis*-regulatory features in simpler organisms with compact genomes, such as bacteria (Nguyen et al., 2024b; Zvyagin et al., 2023; Shao, 2023), arabidopsis (Benegas, Batra, and Song, 2023), or yeast (Karollus et al., 2023), which have substantially reduced junk DNA (Eddy, 2012; Niu and Jiang, 2013; Graur et al., 2013).

By contrast, supervised deep learning models trained on large troves of functional genomics data in a multitask setting can learn discriminative features related to *cis*-regulatory mechanisms in the non-coding genome (Koo and Eddy, 2019; Koo and Ploenzke, 2021; Avsec et al., 2021a; Almeida et al., 2022; Nair et al., 2023). However, the representations learned by these models are biased towards the experiments they are trained on, which are predominantly generated within a few cell lines. Hence, their generalization capabilities to other cell types remain limited. A major benefit of gLMs is their lack of reliance on labels generated from wet-lab experiments during training, allowing them to learn a broader set of patterns. However, our results suggest that gLMs have yet to learn a foundational set of *cis*-regulatory features in the non-coding genome of humans that can be harnessed in prediction tasks across cell types.

Evaluating what gLMs have learned through predictive modeling remains an endless endeavor. A more efficient approach can be achieved through model interpretation of the gLMs, which should help to understand the alignment between gLMs and prior biological knowledge. Our preliminary analysis of attribution maps was inconclusive, highlighting the need for a more in-depth understanding of what gLMs are learning from pre-training. Further development can build upon the initial progress (Clauwaert, Menschaert, and Waegeman, 2021; Sanabria, Hirsch, and Poetsch, 2023a; Zhang, Bai, and Imoto, 2023) towards more meaningful domain-inspired model interpretation tools to bridge this gap.

Looking forward, it remains an open question whether LLMs will bring the same revolution in human genomics as seen in other fields. The current trends in scaling gLMs (via larger models and considering broader sequence contexts (Zaheer et al., 2020; Nguyen et al., 2023)) might only produce incremental gains, albeit achieved inefficiently according to Chinchilla scaling laws (Hoffmann et al., 2022), as the availability of diverse and informative genomics data is a major limiting factor. It remains unclear whether continued scaling of the gLMs pre-trained with standard language modeling objectives (i.e., MLM or CLM) will eventually lead to realizing emergent capabilities, such as learning cell-type-specific *cis*-regulatory biology in the non-coding genome. The amount of genetic variation required to capture the full complexity of the human genome may be simply too great, as a single genome encodes for the spatio-temporal regulation for all cell types. Incorporating additional information, such as functional genomics data, is likely needed during the pre-training for gLMs to become proficient in characterizing cell-type specific functional elements. Even protein language models trained solely on amino-acid sequences can learn elements of conservation and protein structure and yet struggle to generalize well to a wide diversity of functional tasks (Li et al., 2024). In the least, a separate language modeling objective for different regions in the genome to account for the high entropy in the non-coding regions is needed. Due to the high upfront costs to train gLMs with the lack of reciprocal performance gains on downstream tasks, gLMs will likely require a more focused, domain-inspired revelation in pre-training objectives to achieve the esteemed “foundation” status for the non-coding genome.

3.4 Methods

Pre-trained language models

Nucleotide-Transformer. Nucleotide-Transformer consists of multiple BERT-based language models with 2 different model sizes (i.e., 500 million and 2.5 billion parameters) and trained on various sets of genome sequences: human reference genome, 1000 genomes project, and 850 genomes from several species. Details of the tokenizer, model structure, and training procedure can be found in the original paper(Dalla-Torre et al., 2023). We acquired weights for each Nucleotide-Transformer model from the official GitHub repository. In this analysis we mostly used representations from NT2.5B-1000G, except for the zero-shot variant effect generalization analysis, which considered all Nucleotide-Transformer models. Since Nucleotide-Transformer models allow flexible input sizes, no padding was necessary for any evaluation tasks.

Custom GPN. The GPN model is a convolutional neural network that was originally trained on Arabidopsis genome sequences via masked language modeling with an input size of 512 nucleotides (Benegas, Batra, and Song, 2023). It consists of 25 convolutional blocks, where each convolutional block includes a dilated convolutional layer followed by a feed-forward layer, connected by intermediate residual connections and layer normalization. The dilation rate for each convolutional layer cycles with increasing exponentially by factors of 2, from 1 to 32. The embedding dimension was kept fixed at 512 throughout the layers. For our custom GPN (human) model, we created training datasets using the human reference genome (hg38(Schneider et al., 2017)). The genome was split into contigs and filtered for a minimum length of 512 nucleotides, with chromosome 8 held out as test set. During training, 15% of the nucleotide positions were masked and the model is tasked to predict the nucleotide probabilities for each masked location. The model was trained for 2 million steps with a constant learning rate of 0.001 using ADAM (Kingma and Ba, 2014).

HyenaDNA. The HyenaDNA model is a gLM pretrained on the human reference genome, with context lengths up to 1 million tokens at the single nucleotide-resolution (Nguyen et al., 2023). Architecturally, it adopts a decoder-only, sequence-to-sequence configuration, organized into a succession of blocks each encompassing a Hyena operator(Poli et al., 2023), followed by a feed-forward neural network. The model weights and representation extraction code was acquired through the Hugging Face repository (Wolf et al., 2019). For all experiments in this study, we used the “hyenadna-tiny-1k-seqlen-d256” model due to the sequence length limitation of the functional genomics datasets.

DNABERT2. DNABERT2, a second generation version of the original DNABERT model(Ji et al., 2021), is constructed on the BERT architecture, comprising 12 Transformer blocks. In this new iteration, the authors improved the model by replacing learned positional embeddings with Attention with Linear Biases (ALiBi) and utilizing Flash Attention to increase computation and memory efficiency(Zhou et al., 2023). In the context of this study, analyses were done with the representations generated by the last Transformer block. The model was acquired through the Hugging Face repository, using the “ DNABERT-2-117M” model.

Pre-trained supervised models

Sei. The Sei model is composed of three sequential modules: (1) a convolutional network with dual linear and nonlinear paths; (2) residual dilated convolution layers; (3) spatial basis function transformation and output layers. Sei was trained to take as input 4 kb length sequences and predict 21,907 TF binding, histone marks and DNA accessibility from peak data of *cis*-regulatory profiles. For this study, we extracted our representations after the spline basis function transformation, before inputting into fully connected layers. The pre-trained Sei model was acquired through zenodo from the original study (Chen et al., 2022b).

RBP. Our custom RBP model was trained using eCLIP-seq(Van Nostrand et al., 2016) data of 120 RBPs in K562 from ENCODE("An integrated encyclopedia of DNA elements in the human genome" 2012). The dataset was organized into a multi-task binary classification format. The model has a ResidualBind-like structure:

1. 1D convolution (96 filters, size 19, batch-norm, exponential)

   dropout (0.1)

2. Dilated residual block(Yu, Koltun, and Funkhouser, 2017)

   convolution (96 filters, size 3, batch-norm, ReLU)

   dropout (0.1)

   convolution (96 filters, size 3, batch-norm, dilation rate 2)

   dropout (0.1)

   convolution (196 filters, size 3, batch-norm, dilation rate 4)

   dropout (0.1)

skip connection to input

ReLU activation

max-pooling (size 10)

dropout(0.1)

3. 1D convolution (192 filters, size 7, batch-norm, ReLU)

   dropout (0.1)

   global average-pooling

4. flatten

5. fully-connected (512 units, batch-norm, ReLU)

   dropout (0.5)

6. output layer (120 units, sigmoid)

Data

lentiMPRA. The lentiMPRA dataset for K562 and HepG2 cell lines was acquired from the Supplementary Tables in Ref.(Agarwal et al., 2023). The HepG2 library consists of 139,984 sequences, each 230 nucleotides long, and the K562 library contains 226,253 sequences. Each sequence is paired with a target scalar value that represents the transcriptional activity. Each cell line was treated independently as a single-task regression. For each dataset, we randomly split the training, validation, and test sets according to the fractions 0.7, 0.1, 0.2, respectively. Unlike the original study, we treated reverse-complement sequences separately; they were not aggregated or augmented during test time. The results represent the performance over a single fold.

CAGI dataset. The CAGI5 challenge dataset(Kircher et al., 2019) was used to evaluate the performance of the models on zero-shot single-nucleotide variant effect generalization as following the same procedure as Ref. (Toneyan, Tang, and Koo, 2022). We only considered MPRA experiments in HepG2 (LDLT, SORT1, F9) and K562 (PKLR). We extracted 230 nucleotide sequences from the reference genome centered on each regulatory region of interest. Alternative alleles are then substituted correspondingly to construct the CAGI test sequences. Pearson correlation was calculated between the varient effect scores by the model and experimentally measured effect size per experiment. For HepG2 performances, we report the average Pearson's r across the three experiments.

ChIP-seq. Ten transcription factor (TF) chromatin immunoprecipitation sequencing (ChIP-seq) datasets were acquired from the zenodo repository of Ref.(Majdandzic et al., 2023). The prediction task is a binary classification of whether 200nt input DNA sequences are associated with a ChIP-seq peak (positive label) versus sequences from DNase I hypersensitive sites from the same cell type (i.e., GM12878) that do not overlap with any ChIP-seq peaks (negative label). The number of negative sequences were randomly down-sampled to exactly match the number of positive sequences to ensure balanced classes. The dataset was split randomly into training, validation, and test set according to the fractions 0.7, 0.1, and 0.2, respectively.

Alternative splicing data. Data was acquired from direct correspondence with the authors of Ref.(Cheng et al., 2021) Briefly, 61,823 cassette exons from ASCOT was split into a training, validation, and test set. The training set consisted of 38,028 exons from chromosome 4, 6, 8, 10-23, and the sex chromosomes. The 11,955 exons from chromosome 1, 7, and 9 were used as the validation set, and the remaining 11,840 exons were used as the test set (chromosomes 2, 3, and 5).

Models are evaluated based on their performance on the test set. The prediction task takes as input two sequences – a sequence with 300 nt upstream of the acceptor and 100 nt downstream of the acceptor and a sequence with 100 nt upstream of the donor and 300 nt downstream of the donor – and the goal is to predict PSI across 56 tissues as a multi-task regression.

INSERT-seq. INSERT-seq data was obtained from Ref.(Vlaming et al., 2022). INSERT-seq measures the impact of transcribed sequences on the RNA polymerase II elongation potential and expression in mouse embryonic stem cells. 11,417 insert sequences of length 173nt long were used as inputs and the goal is to predict the totalRNA output, which measures the relative abundance in RNA relative to genomic DNA, as a regression task. Training, validation, and test sets were split according to the fractions 0.8, 0.1, and 0.1, resulting in 9,131, 1,149, and 1,137 sequences, respectively.

eCLIP datasets. The *in vivo* eCLIP-based datasets were downloaded from the ENCODE. For each RBP experiment, the bed narrowPeaks (two replicates) and the bam file for the corresponding mock inputs experiment were downloaded. For each replicate, we removed peaks with a signal value less than 1 and a log-p-value greater than 3. Using bedtools, the remaining peaks that share at least one nucleotide across the two replicates were selected as positive peaks. A correlation filter across the replicates was applied: $(2(s_i^1 - s_i^2)/(s_i^1 + s_i^2))^2 < 1.0$, where s_i^j represent the signal value for the ith peak in replicate j. The median peak size was about 50 nt with a positive tail that exceeded 200 nt in some cases. Positive sequences were generated by extracting 200 nucleotide sequences about the center position of the peak coordinates. Sequences with undefined nucleotides were filtered out. Negative peaks were generated by employing Piranha peak caller on the bam file of the mock inputs with a bin size of 20 and a p-value threshold of 0.01.

We then removed negative peaks which overlap with any unfiltered peaks from each replicate. Negative peaks were generated by extracting 200 nt sequences about the center position of the remaining negative peak coordinates. Because the negative peaks usually had more entries compared to the positive peaks, we randomly selected a similar number of negative peaks as positive peaks. All sequences were given a corresponding label 1 for sequences which contain a positive peak and 0 for sequences which contain a negative peak. All sequences were then randomly split into a training set, validation set, and test set according to the fractions 0.7, 0.1, and 0.2, respectively.

Models for downstream tasks

Linear models. Linear models with L2 regularization (i.e., Ridge) serve as the baseline, representing a simple downstream model. The inputs of the model were based on the embeddings of the CLS token or the average embedding across sequences for Nucleotide-Transformer models. For regression and classification tasks, the linear model was a linear regression or logistic regression, respectively. The strength of the L2 regularization was set to 1e-3.

MLP. A multi-layer perceptron model was used to train on CLS token embeddings or the average embedding across sequences for Nucleotide-Transformer models. The model is constructed by two fully connected blocks. The first block includes a fully-connected layer with 512 units and ReLU ativation, followed by batch-normalization and a dropout rate of 0.5. The second block consists of a fully-connected layer with 256 units and the same activation, batch-normalization, and dropout layers. The model was trained on lentiMPRA dataset with Adam optimizer, learning rate of 0.0001, mean-squared error loss function, learning rate decay with a patience of 5 epochs and a decay factor of 0.2, and early stopping patience of 10 epochs.

MPRAnn for lentiMPRA. MPRAnn is a convolutional based model with a total of 4 convolutional and 3 dense layers trained on the lentiMPRA dataset. It takes 230 nt one-hot encoded sequences including the adapters as input to predict the mean log2(RNA/DNA) values from forward and reverse strands. We augmented the batches using the reverse-complement of the 200 nt target sequence, while keeping the two 15 bp adapters fixed. To fit the model, we used a learning rate of 0.001, an early stopping criterion with patience of 10 on 100 epochs, and the Adam optimizer with a mean square error loss function. Model structure and training parameters obtained from Github directory of original publication(Agarwal et al., 2023).

Baseline CNN for lentiMPRA. We designed a baseline CNN model with the following structure:

1. batch-norm (optional)
2. 1D convolution (196 filters, size 1) (optional)
3. 1D convolution (196 filters, size 7, batch-norm, exponential)

   dropout (0.2)

   max-pooling (size 5)
4. 1D convolution (256 filters, size 7, batch-norm, ReLU)

   dropout (0.2)

   max-pooling (size 4)
5. flatten

6. fully-connected (512 units, batch-norm, ReLU)

   dropout (0.5)

7. fully-connected (256 units, batch-norm, ReLU)

   dropout (0.5)

8. output layer (1 unit, linear)

CNN models were trained with Adam optimizer, mean-squared error loss function, learning rate of 0.0001 with a learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping with patience of 10 epochs for both one-hot sequence and language model embedding-based training on the lentiMPRA dataset. For one-hot sequences, batch-norm and the convolution with kernel 1 were not employed.

ResidualBind for lentiMPRA. We designed the ResidualBind model by adding a dilated residual block after the first convolutional layer of the baseline CNN model, according to:

1. 1D convolution (196 filters, size 15, batch-norm, exponential)

   dropout (0.2)

2. Dilated residual block

   convolution (196 filters, size 3, batch-norm, ReLU)

   dropout (0.1)

   convolution (196 filters, size 3, batch-norm, dilation rate 2)

   dropout (0.1)

   convolution (196 filters, size 3, batch-norm, dilation rate 4) dropout (0.1)

convolution (196 filters, size 3, batch-norm, dilation rate 8)

dropout (0.1)

convolution (196 filters, size 3, batch-norm, dilation rate 16)

dropout (0.1)

convolution (196 filters, size 3, batch-norm, dilation rate 32)

skip connection to input

ReLU activation

max-pooling (size 10)

dropout(0.2)

3. 1D convolution (256 filters, size 7, batch-norm, ReLU)

   dropout (0.2)

   max-pooling (size 5)

4. flatten

5. fully-connected (512 units, batch-norm, ReLU)

   dropout (0.5)

6. fully-connected (256 units, batch-norm, ReLU)

   dropout (0.5)

7. output layer (1 unit, linear)

ResidualBind was trained with Adam optimizer, mean-squared error loss function, learning rate of 0.001 with a learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping with patience of 10 epochs.

Baseline CNN for ChIP-seq and CLIP-seq. We designed a baseline CNN model with the following structure:

1. batch-norm (optional)

2. 1D convolution (512 filters, size 1) (optional)

3. 1D convolution (64 filters, size 7, batch-norm, ReLU)

   max-pooling (size 4)

   dropout (0.2)

4. 1D convolution (96 filters, size 5, batch-norm, ReLU)

   max-pooling (size 4)

   dropout (0.2)

4. 1D convolution (128 filters, size 5, batch-norm, ReLU)

   max-pooling (size 2)

   dropout (0.2)

5. flatten

6. fully-connected (256 units, batch-norm, ReLU)

   dropout (0.5)

8. output layer (1 unit, linear)

CNN models were trained with Adam optimizer, binary cross-entropy loss function, learning rate of 0.001 with a learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping with patience of 10 epochs for both one-hot sequence and language model embedding-based training on the lentiMPRA dataset. For one-hot sequences, batch-norm and the convolution with kernel 1 were not employed.

Insert-seq model. For the RNA pol II elongation potential dataset, we developed a residual convolutional network structure and used it for all embedding and one-hot-based models. The model was trained using mean square error loss function, Adam optimizer, learning rate of 0.0001, learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping patience of 10 epochs.

1. convolution(48 filters, size 1) (optional)
2. convolution (96 filters, size 19, batch-norm, exponential)

   dropout (0.1)
3. dilated residual block

   convolution (96 filters, size 3, batch-norm, ReLU)

   dropout (0.1)

   convolution (96 filters, size 3, batch-norm, dilation rate 2)

   dropout (0.1)

   convolution (96 filters, size 3, batch-norm, dilation rate 4)

   skip connection to block input

   ReLU activation

   max-pooling (size 10)

   dropout(0.1)

4. convolution (128 filters, size 7, batch-norm, ReLU)

   global average-pooling

   dropout (0.1)

5. fully-connected layer (128 units, ReLU)

   dropout (0.5)

6. output layer (1 unit, linear)

CNN models were trained with Adam optimizer, mean-squared error loss function, learning rate of 0.0001 with a learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping with patience of 10 epochs for both one-hot sequence and language model embedding-based training on the lentiMPRA dataset. For one-hot sequences, the convolution with kernel 1 was not employed.

Zero-shot variant effect prediction methods

For Nucleotide-Transformer, we derived the zero-shot predictions using cosine similarity as suggested in the original study (Dalla-Torre et al., 2023). For each variant, we passed the sequences with the centered reference allele and the alternative allele through the model to extract embeddings. The cosine similarity between the two complete sequence embeddings was calculated and used as the zero-shot score. A negative correlation is expected between the score and effect size. Since this distance-based zero-shot score only reflects the magnitude, not the direction, of function change, we calculated the Pearson correlation using the absolute value of the effect size.

For GPN, we followed a similar procedure as the original study (Benegas, Batra, and Song, 2023). First, we input sequences with the center variant loci masked and acquired the predicted allele probabilities for the masked loci. Then, we calculate the zero-shot prediction score as the log-likelihood ratio between the alternate and reference alleles. Again, since the likelihood ratio doesn't reflect the direction of function change associated with the variants, we calculated the correlation score using the absolute value of effect size.

Finally, for the embedding-based and one-hot based models, we used the difference in predictions between the alternative and reference allele sequence as the zero-shot prediction score. For Enformer, we use the cell-type agnostic approach of averaging the effect size across all DNase-seq tracks. To reduce predictions to scalars, we summed across the profile predictions.

Attribution methods

For CNN models, the attribution analysis was based on grad-times-input with saliency maps. The gradients of the prediction were calculated with respect to the input sequence to yield an L x A map, where L is the length of the sequence and A is 4 (one for each nucleotide). By subtracting the position-wise average saliency scores from this map and then multiplying by the one-hot encoded sequence, the method isolates the sensitivity of each observed nucleotide at every position, enhancing interpretability by pinpointing nucleotide-specific contributions to predictions.

For gLMs, the analysis involved sequentially masking each token of the input sequence and predicting the probability of the masked token by the model. The entropy of the probability distribution for each position was computed to quantify the information content represented by the gLM. Given that lower entropy signifies a higher information level, the saliency score was derived as the difference between the maximum entropy value and the entropy at each position, ensuring that a higher saliency score reflects greater information retention.

Sequence logos were visualized using Logomaker (Tareen and Kinney, 2020).

Global importance analysis

Global importance analysis was carried out according to Ref. (Koo et al., 2021). A example sequence was selected from the LentiMPRA (K562) dataset. We sampled 300 dinucletoide shuffled versions of the sequence to be used as background sequences. The shuffling aims to preserve the dinucleotide frequency while destroying any coherent patterns. The LentiMPRA trained One-Hot-CNN models' predictions for the shuffled sequences are considered to be the baseline for predicted CRE activity. The top three positive motif patters identified separately in the One-hot-CNN and GPN-CNN saliency maps (Fig. 3.6c) were inserted into the corresponding position of the shuffled sequences, creating two experiment sequences sets. The One-Hot-CNN model was used to make predictions for the motif embedded sequences. The difference in prediction for the three sets of sequences reflect the global importance of these motif patterns to the CNN model.

Chapter 4

Discussion and perspectives

The work in this thesis has been done as attempts to understand the utility of current DL methods for understanding the sequence-function landscape of genomic regulation. In this chapter I discuss the major limitations encountered during my research, as well as proposing paths forward that could overcome these challenges and expand our understanding.

4.1 Modeling considerations from the sequence-function landscape perspective

DNNs learn a sequence to molecular function mapping through training on functional genomic data. From a function approximation standpoint, this can be thought of as the DNN is simulating the functional genomics assay and the subsequent data interpretation step. This allows one to use the trained DNN as a surrogate for the experimental assay, albeit for the given condition that generated the functional genomics data. This enables making new *in silico* measurements for sequences that the model has not seen during training or as part of the original experiment. Thus, the trained genomic DNN can generate hypothetical or counterfactual measurements to sample new regions of sequence space, through the lens of the approximated functional genomics assay. Considering genomic DNNs according to the inferred sequence-function landscape is a powerful tool that facilitates their understanding. In this section, we elaborate how we view functional genomics problems through the lens of sequence-function landscapes and distribution shifts, explaining the behaviour of model generalization and interpretability.

4.1.1 Generalization under different levels of covariate shifts

Sequence space is high dimensional with 4^L possible sequences (i.e., $x \in \{A, C, G, T\}^L$), where L is the length of the sequence. This discrete combinatorial space is larger than the number of atoms in the universe for typical prediction tasks (with $L > 200$). While not exact, it can be helpful to gain intuition by visualizing sequence space according to a projection on a lower-dimensional, continuous manifold. In this space, a sequence is a point. Similar sequences are placed in close proximity and the surface, while portrayed smooth here, is rugged, with few mutations that can dramatically change the functional activity of the sequence.

Through this perspective, the challenge becomes quite apparent – experimental data samples sequence space only sparsely and in a biased manner, only including the natural genetic variation across a reference genome. Thus, its unclear whether the model would generalize well and produce reliable predictions for data far outside the narrowly sampled training data. However, we are not pursing generalization everywhere in the sequence space. Since trivial sequences such as a genome fully comprised of adenine are not of interest for understanding biological mechanisms.

Generalization becomes increasingly difficult as the model encounters data points that diverge from training data more; this is known as a covariate shift, which is a type of distribution shift of the input data. Thus, the evaluation for generalization capabilities of genomic DNNs should be set according to the level of covariate shift that is appropriate for the target application. Below we provide examples of different scales of covariate shifts that are relevant for genomics applications: individual-level variation, population-level variation, disease-level variation, and synthetic sequence variation.

Individual-level variation. Generalization across an individual's genome is the most typical form of evaluation, using held-out data from the same dataset. In practice, this is accomplished by holding out chromosomal data. This test ensures that the model can predict new sequences that sample similar genetic variation as the training data, under the same sequence biases that are under similar selection pressures from evolution. Good performance on held-out test chromosomes does not inform generalization across more distant regions of sequence space.

Population-level variation. Generalization across population-level variation is a test to see how well population-level variant effects can be predicted. This usually includes a single nucleotide variant, but can extend to numerous variants as part of a haplotype in a genomic locus. These variants are typically important for phenotypic variability, but their effect sizes are often weak. Thus, generalization across population-level variation can be subtle. Recently, a study led by Sasse et al. has found that Enformer, a state-of-the-art sequence-to-expression DNN, struggles to capture population variation. Moreover, Huang et al. have also found that Enformer strugges to predict eQTLs. Nevertheless, Enformer's prediction on held-out chromosomes remains SOTA and hence this gap can be attributed to poor generalization under population-level variation.

Disease-level variation. The level of genetic variation is higher in diseases compared to healthy populations. Certain diseases or predisposition to disease, are caused by mutations - the majority of which have been mapped to the non-coding genome. Other, such as cancer, can introduce numerous rearrangements in the genome, from single-nucleotide mutations to indels and structural variations. Currently, much effort is focused on single-nucleotide level variation to fine-map disease loci and provide mechanistic understanding through generalization of genomic DNNs. While models can predict variants, the evaluation of disease-associated variants remains difficult to comprehensively assess due to a lack of ground truth. Further progress on curating meaningful tasks that can assess generalization to disease-level variation is needed to be able to benchmark genomic DNNs at this important generalization task.

Synthetic sequence variation. A major application of genomic DNNs is design of synthetic sequences that can provide tunable control of regulatory function. This requires the ability to either mine existing sequences that provide the desired effects or to navigate along sequence space outside the training distribution to a mode with a desirable level of activity. Thus, evaluating the ability of genomic DNNs to generalize for synthetic sequence variation is required to ensure this objective.

All genomic DNNs produce a functional prediction for the entirety of sequence space. However, demonstrating generalization capabilities within individual-level variation does not guarantee their ability to predict disease-level variation or synthetic sequence variation, due to the varying levels of covariate shifts. Moreover, generalization on a finite set of values on average can appear good but there could still be some pathological regions of sequence space, where predictions are much less reliable.

There are not many experimental datasets that have been generated for the purposes of evaluating genomic DNNs. Not all evaluations are equally informative. The level of covariate shifts should be stated for a given evaluation to ensure that the expectation level of generalization is calibrated for the level of covariate shift expected according to the downstream application of the model.

4.1.2 Generalization under label shifts

Sequence-function landscapes are complex due to the dynamic nature of biological processes, such as the evolution or spatio-temporal changes in the molecular function of sequences. When training a sequence-based genomic DNN on functional genomics data, the state of the cell is assumed to be constant with a fixed concentration level of TFs. Since a single genome encodes for all cell types in a human body, it is possible, in principle, to predict specific functional activity from sequence alone. However, the complex biological processes are dynamic and can easily alter the state of the cell, which changes the sequence-function landscape. This distribution shift is known as a label shift and can manifest if the training and test data were measured under different conditions. Examples of different conditions could be in vitro versus in vivo, or different perturbations applied to the cell, or under different stress conditions.

Label shifts can also arise due to measurements using different experimental technologies. Although it is typically assumed that the experimental measurements are ground truth, this notion creates an arms race to make better predictions on experimental data, which may not necessarily learn better underlying biology but rather a better noise model of the experimental technology. Noise is intrinsic to biology as well as measurements. Thus, the inferred landscape by the genomic DNN may not be smoothing out the noise as well as it could be. Thus, comparisons of the prediction of models comparing to replicate variability has emerged and this is quite informative for the level of performance given the expected noise levels of the experiments.

By thinking about the label shifts across technologies and experimental conditions, one can gain a better appreciation for the challenges in assessing generalization of the underlying biology versus the ability to better predict noise. Recent progress by ChromBPNet has illuminated a powerful strategy to decompose technical biases of different sequencing assays. This is a powerful way to remove biases from data. While the intended purposes has been to regress out the bias for better modeling, this process can also help to remove bias from held-out test data to better show generalization capabilities across technologies.

Further, the expectation of generalization across cell types or generalization across species can be informative but the take aways of the lack of generalization is not constructive as this can be due to not learning shared biology or due to the divergence of the biological systems (or due to being measured under different conditions). One cannot draw any meaningful conclusions from these experiments without further control experiments that test alternative hypotheses.

4.1.3 Model explanations from a function approximation perspective

Attribution methods—such as Saliency Maps, DeepLIFT, DeepSHAP, Integrated Gradients, SmoothGrad, and more—are powerful tools to explain the elements within a sequence that are important for model predictions. Each of these popular attribution methods can be thought of as an additive approximation to a local region about the sequence of interest; each method's approximation window (also called a neighborhood size) varies.

For instance, a Saliency Map is a linear approximation of a point, tangent to the sequence-function landscape. SmoothGrad samples different regions nearby via input perturbations and averages the gradients at each sampled sequence. However, in practice, adding Gaussian noise as is done in the original SmoothGrad study means that the sampled region lies off of the simplex where the data resides; SmoothGrad explores the input space assuming its a continuous distribution while valid sequences exist only in discrete space. Even though a genomic DNN can produce predictions for all of 4D Euclidean space, its behavior off the simplex can exhibit arbitrary biases due to not having any data to ground its behavior. Hence, the gradients contain arbitrary components, albeit a simple statistical correction that projects the gradient along the simplex can mitigate the impacts of this.

Integrated gradients, on the other hand, interpolates the gradients from a reference point, such as a vector of all zeros or an inactive reference sequence, to the sequence of interest. Similar to SmoothGrad, this also suffers from sampling gradients in the function off of the simplex. Hence, importing attribution methods developed in computer vision into genomics has brought some peculiar behaviors that are not necessarily rigorous for categorical inputs. Some recent work has attempted to address these gaps by adapting these methods for genomics, including enhanced integrated gradients and a version of integrated gradients that walks along the sequence space.

To address this challenge a surrogate modeling approach called SQUID directly samples a local region of the sequence-function landscape similar to MAVEs and uses inherently interpretable quantitative models designed for MAVEs for the approximation. Unlike previous attribution maps, SQUID enables approximations that fit non-linear functions and more tuned levels of neighborhood sizes.

It can also make non-additive approximations, providing insights into pairwise interactions within motifs and between motifs. Further progress on thinking about sequence-function landscapes can help to identify weaknesses in exiting attribution methods and develop theoretically more rigorous adaptations for genomics.

4.2 Major limitations and paths forward

4.2.1 Framing regulatory genomics prediction tasks

A major challenge lies in framing a meaningful prediction task in regulatory genomics. The current paradigm that predicts regulatory functions from just DNA sequences assumes fixed confounding variables of cell state and environmental factors. However, features such as the concentration levels of TFs in the cell or signalling molecules in the cellular environment, change dynamically. Consequently, these models struggle to generalize across different cell states, cell types, and environmental conditions, as they are trained on static snapshots in time and couldn't learn the dynamics of regulatory processes. Moreover, the models are trained on observational profiling data instead of interventional perturbation data which can give more direct insights into causal relationships for cis-regulatory mechanisms.

Confounding in variant effect predictions. As the consequence of the sequence-focused approach, models are not aware of the potential changes and dynamics of the gene regulatory network (GRNs), i.e. the interactome of TFs that bind to CREs and regulate a target gene. Consequently, they overlook the cascade of interactions at the gene level by TFs, which can significantly influence expression levels. This could explain why eQTL predictions by such models are not always accurate. A comprehensive model would need to consider all cellular context features instead of solely base its prediction on sequences. Nevertheless, although sequence-based

DNNs cannot directly predict the effect on the state of the cell, their instantaneous variant effect predictions can still provide valuable insights for significant changes in the GRN.

Bias in peak-centered training data. In addition to the limitations of pure sequence-based models, artificial bias can be introduced during training. While convolutional layers are often considered to be translation equivariant – the same patterns in different locations would lead to the same activation. This is only valid within the receptive field of the layer and edge effects can limit its equivariance. If a CNN's receptive field covers the entire input seqeunce, the model would loss equivariance to the translational transformation of inputs. This applies to all models that employ a flatten layer followed by a dense layer, which is equivalent to a convolutional kernel that spans the length of the input sequence. Thus, the non-equivariant CNNs trained on peak-centered data where input sequences always centered at high read coverage regions, have a tendency to learn only important patterns within the center of the sequence. This artificial bias is introduced due to the sampling of training data, and makes DNNs susceptible to inaccurate predictions upon translational shifts of the sequence. Solutions to improve DNN robustness to such effects include data augmentations, random sampling for training data, or manually include translational shifts of the input data.

Limited training data. Given that training sequences cover only a small fraction of all possible sequences, sequence-function landscape inferred by DNNs may not fully reflect biological ground truth, especially in regions with little to no data support. This includes most of sequence space because it is incredibly high-dimensional, and training data only sparsely samples it in a biased manner (because of evolutionary constraints). For example, a DNN will struggle to accurately learn the biophysical mechanism for a TF binding from a dataset with only tens to a few hundred positive binding sites across the genome.

A possible solution is to integrate other biological data that share similar cis-regulatory grammar through pre-training or multi-tasking. Pre-training datasets may contain similar binding sites for different TFs or for different cell types. However, this strategy may not accurately learn binding contexts such as flanking nucleotide preferences. In case of multi-tasking, the model learns different sequence-function landscape for each task in parallel. The assumption is that each task is independent, with potential overlap in underlying features. This method aims to mitigate the limitations of small datasets by including more information during training. However, the current approaches in genomics multi-tasking use the same input and make predictions across different sequence assays, which can introduce class imbalances for each task.

Poorly framed prediction tasks can introduce biases. Lastly, when prediction performance is overly optimistic, this could be a red flag that the task itself is trivial. One possible source of ill-posed problems is the selection of background sequences. For example, the natural genetic variation in regulatory regions across the human genome contains distinct low-level sequence biases that have been well-recognized, such as GC bias and other low-level statistical biases. Hence careful consideration in background selection is important for discriminative learning. When poorly chosen, expressive models like DNNs can easily exploit low-level statistics to make predictions, producing to good classification performance without learning meaningful features. Another possible source is when the signal of interest is correlated with other signal that is not mechanistically important. For instance, when predicting TF binding, the background set should not be dinucleotide shuffled sequences as any accessible signals could be predictive of the TF binding in the given cell type. Rather, it should contain GC-matched accessible sites that do not have any TF binding in the cell type of interest. Other ways that framing can go awry is when the prediction task is dependent on understanding other features, such as when multitasking across different cell types for the same TF. The model cannot simply learn the correct TF motif in order to make the right predictions, rather, it has to learn other signals about accessibility for each cell type in order to make the right prediction of binding. This may be why model performance was found to improve when increasing the input sequence context, which provides more opportunities to learn cell type specific signals, not necessarily information specific to the TF of interest. Moreover, the interpretation of in silico mutagenesis for these models is also conflated due to the confounding factor of chromatin accessibility.

4.2.2 Models are not yet reliable oracles

Another major issue with DNNs is their lack of awareness about their own knowledge gaps. Their inferred sequence-function landscape is not likely to be fully reliable outside of the data support, in fact, there may be pathological regions not far from training data. For instance, adversarial examples have long been known to exist for computer vision models and their impact in genomics has not been fully explored. Even thought there may not be clearly defined adversay examples, we care about the robustness of these models to small perturbations. Since DNNs learn a 4D Euclidean space, learning off of the data simplex can still be beneficial in terms of improving robustness properties on the simplex. Accurate model predictions outside the training data distribution is important for variant effect prediction, model interpretation through in silico experiments, and designing novel sequences with desirable properties. Therefore whenever a prediction for a new sequence is required, it's important to assess the reliability of that prediction.

There are several paths forward that can help address this gap, including uncertainty quantification, measures of OOD, robustness measures of the local-sequence-function landscape, and sanity checks.

Uncertainty quantification. DNN predictions are known to be overconfident. For binary classifiers, the predictions can be thought of as probabilities. However, predictions tend to be more heavily skewed to extremes, 0 or 1. Thus, the confidence in the predictions is uncalibrated. Expected Calibration Error(ECE) is one approach to measure the prediction uncerntainty.

Genomic DNNs that are trained in a supervised manner typically learn point estimates of weights, not distributions like Bayesian machine learning approaches. So for a given input sequence, DNNs will produce a deterministic prediction. In a Bayesian approach, the prediction are represented by a distribution for which the uncertainty can be drawn. The simplest, but computationally expensive approach to estimate uncertainty is to train an ensemble of models, each with a different initialization. The models will learn slightly different sequence-function landscapes, providing a distribution of predictions for each input. This so-called DeepEnsembles is a powerful approach to estimate epistemic uncertainty, the model-based uncertainty that can become reduced with more data. Alternative approaches include less conventional methods like MCDropout, which keeps dropout on during inference time, leading to variations in the model predictions. MCDropout has been shown to approximate uncertainty by generating a distribution of outcomes based on the dropout-induced variations. Alternatively, Bayesian neural networks, with bayesian noise layers, can produce uncertainty estimations, but often do not achieve similar performance as conventional DNNs. However, uncertainties estimated by the model has no promise to be properly calibrated. Recently, there has been a resurgence of conformalize predictions, which ensure that quantities like uncertainties are calibrated and can make statistical guarantees of their estimates.

This often involves using a held-out calibration set to set the bounds, the resultant uncertainties can then ensure that they encompass 95 confidence interval of the true values.

However, uncertainty is not a panacea in itself as the modeling bias can still be misleading. Recently, a detailed study that explored found that there are regimes where epistemic uncertainty is low, but the model prediction remains incorrect. Moreover, epistemic uncertainty is only one aspect of the predictive uncertainty, aleatoric uncertainty is a different type of uncertainty that is based on the noise of the system. For genomics, this is the combined effect of the biological noise and measurement noise. One strategy for estimating aleatoric uncertainty is to consider replicate uncertainty within a given locus. When assessing the efficacy of sequence-function landscapes, its important to consider the combination of epistemic uncertainty, aleatoric uncertainty and the bias of the models. If the epistemic uncertainty is high, this may indicate an unstable region, perhaps pathological region of sequence space. A high aleatoric uncertainty may suggest that the biological variability or experimental noise across replicates was high and the predictions may not necessarily be as accurate in these regimes as a result.

OOD measures. OOD is a way to measure the distance of new examples from the training distribution. Descriptive statistics could be based on distribution measures of distance, such as KL divergence, JS distance and Wasserstein distance, between descriptive statistics like k-mer frequencies. Recently a model-based approach to predict OOD score between two gropus of sequences has been developed. It has shown promising performances, albeit the breadth of the demonstrated use cases has not been extensive. Alternatively, generative approaches that learn the data distribution offer another effective means for outlier detection. These methods are typically similar to homology search methods which consider log-odds scores. Further development of generative models for regulatory sequences and their comprehensive evaluation (which is currently lacking) is needed to properly test their abilities in OOD detection.

Robustness properties of local sequence space. An alternative approach to characterizing pathological regions in sequence space, where model predictions would not be reliable, could be to quantitatively analyze local properties of sequence space. For example, if a specific region exhibits a more rugged landscape compared to what the model encountered during training, it could signal a potentially pathological area. This approach could draw from existing research on understanding fitness landscapes, where techniques for assessing landscape ruggedness have been explored.

4.2.3 Issues with model explanation methods

Similar to robustness in model predictions, the interpretations of DNNs can be fragile. Interpretations may reside in unstable regions of sequence space where small and biologically unsignificant changes can lead to large changes in the attribution maps. This is problematic as because there are no established metrics to gauge the reliability of interpretations from attribution maps. This issue also extends to in silico experiments which may be operating in completely unreliable regions of sequence space.

Benign overfitting can influence reliability of attribution maps. Benign overfitting is the phenomenon where the model overfits on the training set but still generalizes well to the test set. This occurs when the model, rather than learning a smooth function, transitions into an interpolation mode. This occurs not for all training labels; but rather for those data points for which the initial learned patterns are not sufficient to drive model predictions to being 100% confident and accurate. For these incorrectly predicted sequences, an extent of memorization occurs. While the memorized labels does not generalize, the model still relies on learnt features for the majority of the data and hence generalize well out of the training set. Consequently, benign overfitting can introduce roughness in the sequence-function landscape. This is challenging for attribution methods since they rely on characterizing local properties of sequence-function landscapes and are therefore susceptible to overfitting. The significance of this problem in practice is still unclear. Empirical comparisons of models have shown that while test performance may be similar, their resulting attribution maps can vary significantly, often due to benign overfitting.

Regularization strategies can help models learn smoother function. Regularization is a powerful method to combat benign overfitting and to learn smoother functions, ensuring that local explanation methods work as intended. Standard approaches to regularizing genomic DNNs include dropout and weight regularization through L1 and L2 penalties. In practice, regularization does not necessarily improve model performance, so their application has been varied. However, they can have significant impact on attribution analysis outcomes.

More sophisticated regularization techniques such as mixup, manifold mixup, adversarial training, gaussian smoothing, and attribution priors have also demonstrated improved function smoothness properties. Mixup is a strategy that creates new input samples by linearly interpolating between two input data points according to a weighted average. The model is trained to predict the corresponding mix of labels for these synthetic inputs. This ensures linear smoothness in the model's behaviour. Manifold-mixup also mixes two data points but at the hidden layer representation level instead of raw data. It has demonstrated increased robustness over traditional mixup, offering advantages in adversarial robustness and improving the quality of attribution analysis. Adversarial training expose the model to adversarial examples during its training phase, aiming to make it learn robust input features rather than relying on low-level noise statistics for class discrimination. Adversarial training is typically employed by applying epsilon perturbations within an L-infinity ball, using projected gradient descent. Adversarial trained models tend to produce attribution maps that are more focused and human-interpretable, demonstrating favorable properties in attribution analysis. Randomized smoothing, on the other hand, adds Gaussian noise to the inputs to ensure local smoothness. Alternatively, regularization of the learned function through attribution priors can be achieved directly by regularizing the Jacobian or gradients, or indirectly via regularizing Fourier frequency components. The latter has demonstrated a powerful approach to yield more reliable attribution maps. Despite the promising results of these strategies, their practical application remains limited. The development of effective regularization techniques that can be easily to implemented and compatible across different models, without requiring extensive coding infrastructure, is crucial.

Evaluating ability to generate reliable attribution maps Since attribution methods are typically sensitive to local function properties, it is not clear whether a given attribution map is stable. Thus, averaging attribution maps across an ensemble of models typically leads to higher efficacy. This suggests that individual models are suffering from poor fits and could benefit from regularization. However, visualizing this effect can be challenging, highlighting the need for evaluating the robustness of attribution methods. Basic sanity checks can greatly improve our trust for patterns identified from attribution maps. For instance, small translational shifts should not affect attribution maps. By repeatedly randomly shifting a sequence, generating corresponding attribution map, then realigning them, one can quantify the variability of the attribution maps induced by the shifts. This method has proven effective for comparing the robustness of attribution maps across different models. Averaging these realigned attribution maps can also provide a more robust attribution map as well – a genomic-specific version of SmoothGrad.

Evaluating the quality of attribution maps across different models can also be approached by examining the consistency of these maps. One existing approach calculates the KL divergence between the k-mer frequencies within high attributed positions versus a null distribution (i.e., k-mer frequency of the whole sequence) for a population of attribution maps. Another approach compares the KL-divergence between an embedding space of a small window of attribution scores scaled according to their average attribution score compared to a null distribution. In this embedded space, clusters represent patterns that appear more consistently across the population of attribution maps. While this method provide a measure of consistency, it still fails to highlight the diversity of consistent patterns. Hence, further research to identify quantitative measures of desirable properties of attribution maps is needed. These metrics, such as the robustness scores and consistency scores, can be utilized along with the predictive performance as part of a multivariate criteria for model selection that yields a good balance between generalization performance and model interpretability with attribution methods.

4.2.4 Model evaluation remains tricky

Model evaluation is typically demonstrated on held-out test performance. This approach provides a basic evaluation, summarizing the model's performance through a single statistic and gauge its ability to generalize to held-out data within the same experiment. However, evaluating models remains the most difficult challenge for genomic deep learning, as we continue to explore the full extent of the models' capability and limitations. Comprehensive evaluations that provide deeper insights are required to advance genomic deep learning.

Evaluating multi-task models. For multi-task models, its crucial to move beyond summary statistics and focus on cell type specificity instead. The model can simply be predicting average functional activities across all cell types and produce misleadingly high performances that's difficult to beat. Thus, evaluations should be focused on performance at differential activities across cell types or conditions. Moreover, model interpretations for these differential regions should also be compared to constitutive regions. While anecdotal demonstrations provide informative qualitative insights, aiming for quantitative comparisons is essential for advancing studies and providing more definitive outcomes. In silico experiments that directly test the understanding of cell type specific regulatory activities can be an promising avenue forward.

Evaluating genomic language models. While gLMs have recently emerged as a promising innovation for self-supervised learning of genomic sequences, their utility has been plagued with poor evaluations, using benchmark datasets that are not insightful. Some of the most frequently used tasks include binary classification for histone marks and promoter sequences. These binary classification tasks often do not require learning meaningful features, and can sometimes be explained by basic statistic summary of sequences. Hence, quantitative predictions tasks are important for gLMs' evaluation. Also, current benchmarks continue to propagate existing tasks and only compare to existing gLMs as baseline for performances. While a meaningful comparsion would be against state-of-the-art supervised models or downstream models that utilize their representations.

Evaluating generative models. Generative modeling for genomic sequences is becoming increasingly popular, but evaluating these models effectively continues to be a challenge. Current approaches often consider homology search, or percent identity as metrics for assessing diversity. These methods have limitations since a single shift in a sequence can significantly reduce its percent identity with the original sequence, despite it is essentially the same sequence. The standards for evaluating generated sequences must go beyond low-level statistics like GC-bias, in the least, to k-mer frequencies with different ks. Recent progress to help advance the evaluation of generated regulatory sequences was proposed by PolyGraph, which also performs observational motif analysis. Alternatively, a trained model can be used as a scoring function for regulatory activity, but this is subject to the biases of the model. Its necessary to test the trained model's generalizability beyond the natural genome for it to be a reliable oracle. While these approaches provide a good start, more rigorous evaluations are needed to characterize learned regulatory mechanisms, diversity of regulatory mechanisms embedded in generated sequences, and diversity of the sequence context.

Evaluating supervised models using orthogonal datasets. A powerful method to evaluate the generalization capabilities of genomic DNNs is to leverage orthogonal measurements of biological phenomena. This is especially true if the orthogonal dataset incorporates interventional data that samples beyond the natural genome. Ideally, we would have different levels of perturbation data to evaluate the level of robustness to covariate shifts. Moreover, it would help advance the modeling community to come up with standards to process and share these datasets so that we could have a database of them and use them as other fields in ML have standardized benchmark datasets. Several established models have utilized this approach to demonstrate their model performance. For example, Enformer(Avsec et al., 2021c) was trained with gene expression profiles and evaluated with variant effect in regulatory elements; saluki(Agarwal and Kelley, 2022) trained to predict mRNA half-life was evaluated on 3' UTR regulatory function; APARENT2 trained with 3' cleavage and polyadenylation activites was evaluated on clinically relevant variants etc.

While standardized benchmarks are always flawed, the absence of any benchmark is even more problematic to the field. We can start with existing benchmarks and continue to evolve them as better, higher-quality data becomes available. A major limitation of orthogonal experiments is that its readout might not directly reflect the underlying biology due to technical biases and the intricate biological processes, potentially limiting the generalizability of trained models. Thus, careful considerations for the data generation process is needed. Recent work by Kundaje lab has decomposed Tn5 biases from ATAC-seq-based DNNs(Brennan et al., 2023). Their extension of this method to other datasets have shown that such biases are widespread issues. This suggest that it may be better to benchmark against data whose biases has been decomposed, instead of raw data, to help evaluate generalizability of biological knowledge.

4.3 Concerning directions

4.3.1 Unnecessary arms race

Building neural networks in genomics has become relatively simple due to the democratization efforts of deep learning frameworks like TensorFlow, PyTorch, and now Jax. While basic analysis of model fitting and held-out test performance is common, it remains difficult to demonstrate that a given model structure or specific hyperparameter choices is what has enabled the model. In fact, often times the model innovations are not demonstrated as the key innovation that has led to the better model. And a model that may perform slightly worse on held-out test performance could often yield comparable biological discovery as well. This gap is tricky as the arms race is based on model performance but the knowledge gained from these models is not reciprocated with the better performance.

There have been efforts to demonstrate that DNNs can learn biologically relevant features. A common method is by identifying known motifs from attribution maps or convolution filters. However, this can also be achieved by traditional PWM methods, and doesn't benefit from the better performance provided by DNNs. Also, since the search is still based on known sequence patterns, it remains difficult to probe new biological features and mechanisms. Therefore, despite DNNs have shown better performance across many genomic tasks, they have yet provide reciprocal gains in biological insights. Furthermore, better prediction performance can also be the result of learning experimental measurement noise. If the model has learnt to overly fit on experiment specific biases, its ability to generalize based on biological understanding would be impaired. Also if we overly trust the interpretation of these models, we might introduce experimental biases to our understanding of biology.

4.3.2 Blurry line between motivation, claims, and speculation

There is a gap between how model choices, such as architecture or training hyperparameters, are motivated from the application of models in practice. While motivation for using convolutional layers – first layer is used to learn motifs and deeper layers will learn motif interactions, is reasonable, deep learning models can easily spread the motif representations across different layers in practice, making the identification of motifs difficult. These claims cannot be justified without controlled experiments. Instead, the function approximation perspective is more helpful, with arguments that hyperparameter choices were arbitrarily chosen, attempting to find a better solution. If claims of beneficial modeling choice or biological discoveries are made based on observational and attribution analysis, control experiments need to be conducted to ensure they are valid. For example, claims that a model learnt regulatory code need to be substantiated by describing the identified regulation mechanisms, and demonstrating their consistency with experimental behaviors.

Moreover, DNNs are function approximators. Hence, their approximation may be good in some regions of sequence space and poor in others. It's easy to conflate good overall average performance with experitise in all regions of sequence space, even beyond the training distribution, which is typically the sequence space that samples from the human reference genome. Good generalization on sequence space about human reference genome, even if held out, does not gaurantee generalization for larger covariate shifts – a distribution shift of the input sequence.

4.3.3 Lack of FAIR practices

FAIR (Findable, Accessible, Interoperable and Reusable) practices for research software were proposed to facilitate reproducibility and use of packages (Wilkinson et al., 2016). This includes the standards that the software (and its components) is easily searchable, retrievable and free for the community use, and can easily interface with other packages. Importantly, reusability means that the software should be executable with minimal further debugging and complications. Moreover, users should be able to build and modify on top of the released version. All the components of FAIR principles are violated to some extent by many genomic DNN model publications. However, the lack of reusability is likely the most detrimental to the field. Many published models lack modular code to reproduce the data preparation or model training leading to training and processing overhead as other groups attempt to reuse the released models. This is fueled by the lack of standardized coding practices, poorly specified environments, lack of containerization and general lack of interoperability.

In addition to following FAIR principles and deposition of all the necessary components on platforms such as Zenodo, the field should also invest in creation and maintenance of frameworks, such as Kipoi (Liu, n.d.) for depositing datasets and models. In parallel to this, toolkits, such as Gopher (outlined in Chapter 2) and EUGENe (Klie et al., 2023) can facilitate FAIR practices. Following the example in protein sequence modeling by (Gruver et al., 2024), higher levels of abstraction from the specific task of interest (e.g. ATAC-seq specific modeling in Gopher) can improve such efforts and make such toolkits attractive for use by a larger audience.

4.3.4 Resource inequity: academia versus industry

Another major concern is that the rise of large-scale models relies on industry partners, who have the GPU resources, to provide the foundation models for the academic community. The typical cost of the H100 GPU is now at $40,000. Furthermore, industry priorities command the access to these scarce GPUs, which leaves academics with limited resources. There are multiple possibilities for how this can affect the future of genomic DL. A potential future is that there will be synergy between models developed by industry and exploration of model interpretability and applications by academia. This is likely and already takes place to some extent, as models such as Enformer or large language models in genomics (Mendoza-Revilla et al., 2023; Tang and Koo, 2024) become more mainstream and replace single-task or smaller-scale models. Another possibility is increased investment into unified and easy-to-use, GPU-powered servers for academia by government bodies such as NIH (National Institute of Health), NSF (National Science Foundation) or equivalents in other countries. Independent of the next phase, academic research in this field will likely need to change and adapt to survive this resource inequity.

Appendix A

Supplemental Tables

Supplementary Table A.1: Model design and training strategy differences between Basenji and BPNet.

Features	Basenji	BPNet
Epigenetic assays	Histone modifications, TF binding, DNA accessibility, 5’ cap levels	Binding of 4 TFs
Input sequence	131Kb	1Kb
Target resolution	128 bin resolution	Base resolution
Training set selection	Random selection	Center on IDR peaks
Loss function	Poisson NLL	Multinomial NLL + MSE
Evaluation Metrics	Pearson’s r	Jenson-Shannon divergence
Model architecture features	Residual connections, dilated convolutions, maxpool layers	Residual connections, dilated convolutions, task-specific heads
Augmentation	reverse complement, random shift (up to 3 bps)	reverse complement
Test set selection	Random held out 5% of all sequence data	IDR peaks on held-out test chromosomes 1, 8 and 9

SUPPLEMENTARY TABLE A.2: Comparison of models trained on peak-centered versus coverage-threshold data. Basenji-128 and BPNet-base with and without data augmentations (i.e. random RC and translational shift) were evaluated according to the mean-squared error (MSE) and Pearson's r on different held-out test sets, i.e. peak-centered or whole chromosome. Red values highlight the better metric when comparing different training sets for a given model and data augmentation.

Model	Data augmentation	Training data	Test set selection			
			MSE Peak	MSE Chrom	Pearson's r Peak	Pearson's r Chrom
Basenji-128	None	Peak-centered	2.06 ±0.020	0.664 ±0.020	0.582 ±0.003	0.413 ±0.003
		Thresholded	2.21 ±0.059	0.580 ±0.007	0.590 ±0.003	0.456 ±0.001
	RC and shift	Peak-centered	1.78 ±0.032	0.576 ±0.007	0.604 ±0.002	0.441 ±0.001
		Thresholded	1.89 ±0.018	0.545 ±0.002	0.610 ±0.001	0.464 ±0.000
BPNet-base	None	Peak-centered	2.22 ±0.003	0.882 ±0.004	0.550 ±0.001	0.374 ±0.00130
		Thresholded	2.48 ±0.019	0.683 ±0.002	0.545 ±0.000510	0.422 ±0.001
	RC and shift	Peak-centered	2.06 ±0.028	0.770 ±0.018	0.565 ±0.002	0.401 ±0.00225
		Thresholded	2.29 ±0.013	0.640 ±0.003	0.563 ±0.00104	0.429 ±0.002

Supplementary Table A.3: Performance comparison across prediction tasks with various models – trained on binary and coverage data, different output heads, and activation functions – were compared across prediction tasks within whole-chromosome data (whole) and peak-centered data (peak) and filter analysis – hit-ratio of the filters to the JASPAR database and the average of the best q-value for each filter.

Model	Resolution	Output Head	Activation	CHROM AUROC	CHROM AUPR	CHR PEARSON R	PEAK AUROC	PEAK AUPR	PEAK PEARSON R	HIT RATIO	Q VALUE
Binary Models											
Basenji	Binary	Single	GELU	0.819	0.307	0.442	0.802	0.576	0.518	0.527	0.204
			Exponential	0.851	0.292	0.445	0.817	0.607	0.577	0.676	0.153
Basset	Binary	Single	ReLU	0.812	0.314	0.394	0.802	0.592	0.557	0.503	0.199
			Exponential	0.821	0.314	0.404	0.812	0.603	0.551	0.833	0.104
CNN	Binary	Single	ReLU	0.808	0.305	0.392	0.813	0.608	0.565	0.682	0.106
			Exponential	0.826	0.334	0.386	0.850	0.657	0.595	0.859	0.064
Residual	Binary	Single	ReLU	0.843	0.361	0.381	0.863	0.677	0.610	0.448	0.178
			Exponential	0.832	0.346	0.358	0.869	0.688	0.609	0.833	0.074
Quantitative Models											
CNN	1	Task-Specific	ReLU	0.898	0.478	0.659	0.815	0.636	0.641	0.823	0.074
			Exponential	0.889	0.470	0.649	0.806	0.629	0.624	0.859	0.050
		Single	ReLU	0.894	0.472	0.652	0.806	0.625	0.625	0.797	0.070
			Exponential	0.893	0.469	0.650	0.806	0.627	0.628	0.844	0.062
	32	Task-Specific	ReLU	0.891	0.470	0.651	0.807	0.628	0.629	0.812	0.072
			Exponential	0.889	0.468	0.649	0.807	0.629	0.625	0.828	0.072
		Single	ReLU	0.896	0.474	0.654	0.810	0.629	0.630	0.812	0.081
			Exponential	0.890	0.469	0.649	0.806	0.626	0.623	0.839	0.050
Residual	1	Task-Specific	ReLU	0.908	0.520	0.679	0.834	0.667	0.659	0.443	0.173
			Exponential	0.921	0.538	0.694	0.849	0.680	0.684	0.823	0.070
		Single	ReLU	0.913	0.524	0.685	0.836	0.666	0.666	0.505	0.179
			Exponential	0.920	0.540	0.696	0.850	0.683	0.686	0.833	0.067
	32	Task-Specific	ReLU	0.926	0.552	0.696	0.863	0.699	0.684	0.496	0.175
			Exponential	0.931	0.561	0.704	0.869	0.707	0.699	0.867	0.063
		Single	ReLU	0.919	0.531	0.680	0.851	0.681	0.664	0.516	0.194
			Exponential	0.927	0.544	0.691	0.860	0.691	0.680	0.852	0.068
Basenji	128	Single	GELU	0.924	0.532	0.683	0.853	0.680	0.674	0.531	0.178
BpNet	1	Task-Specific	ReLU	0.914	0.507	0.674	0.838	0.660	0.663	0.395	0.080

Supplementary Table A.4: CAGI5 replicability. Each CAGI saturated mutagenesis experiment was conducted for three replicates. Pearson correlation across replication results were measured to estimate experiment consistency.

Name	Replicate correlation
F9	0.61
FP1BB	0.74
HBB	0.62
HBG1	0.78
HNF41	0.75
IRF4	0.98
IRF6	0.90
LDLD	0.99
MSMB	0.75
MYC	0.55
PKLR	0.79
SORT1	0.98
TERT(GBM)	0.90
TERT(HEK293T)	0.65
ZFAND3	0.72

Supplementary Table A.5: ATAC-seq experiments selected from ENCODE database. The IDR-peak bed file and log-fold over control BigWig files were taken based on the following ENCODE experimental accessions.

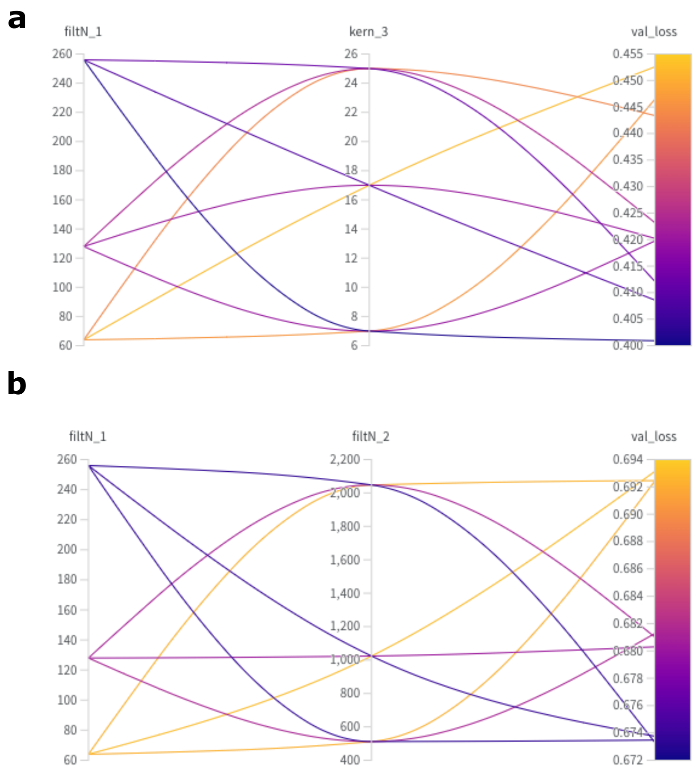
Cell line	Experiment accession
GM21381	ENCSR512YXO
GM23338	ENCSR485TLP
HepG2	ENCSR291GJU
RWPE2	ENCSR080SNF
HG03575	ENCSR331JFZ
K562	ENCSR868FGK
DND-41	ENCSR660WSB
GM12878	ENCSR637XSC
A549	ENCSR032RGS
HCT116	ENCSR872WGW
IMR-90	ENCSR200OML
NCI-H929	ENCSR382LBS
Panc1	ENCSR591PIX
PC-3	ENCSR499ASS
MCF-7	ENCSR422SUG

Supplementary Table A.6: Summary of existing gLM architecture and training choices.

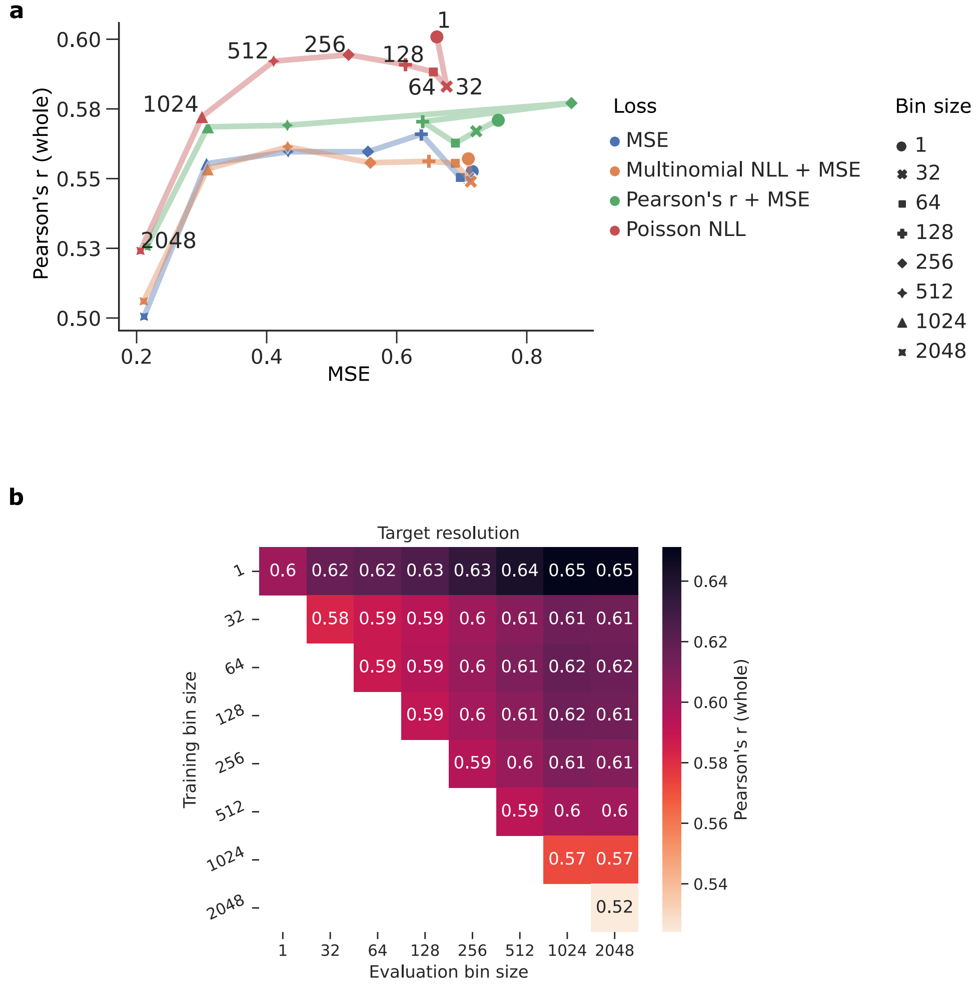
Name	Base Model	Training Data	Pre-training Task	Encoding Scheme	Finetuned for evaluations
GPN [20]	CNN	8 plant genomes	MLM	Base-res	No
GPN-MSA [34]	BERT	Conserved regions in 100 vertebrates MSA (PhastCons)	MLM	Base-res	No
Nucleotide Transformer [23]	BERT	1000 Genomes, Human genome and multi-species genomes	MLM	Non-overlapping k-mer	Yes
DNABERT [24]	BERT	Human genome	MLM	Overlapping k-mer	Yes
DNABERT2 [26]	BERT-Flash attention	Human genome and multi-species genomes	MLM	Byte-pair encoding	Yes
BigBird [32]	Sparse attention	Human genome	MLM+CLM	Byte-pair encoding	Yes
hyenaDNA [21]	Hyena	Human genome	CLM	Base-res	Yes
OmniNA [31]	LLaMA	Multi-species and text annotations	CLM (multimodal)	Byte-pair encoding	Yes
GENA-LM [33]	BERT	T2T, 1000 Genomes, and multi-species genomes	MLM	Byte-pair encoding	Yes
LOGO [39]	ALBERT	Human genome	MLM	Non-overlapping k-mer	Yes
DNA-GPT [25]	GPT	Genome from 9 species	CLM, GC-content, and sequence order	Non-overlapping k-mer	Yes
GROVER [27]	BERT	Human genome	MLM	Byte-pair encoding	Yes
UTR-LM [29]	BERT	5’ UTR sequences (Ensembl)	MLM and supervised (secondary structure and minimum free energy)	Base-res	Yes
SpliceBERT [30]	BERT	pre-mRNA from 72 vertebrates	MLM	Base-res	Yes
regLM [22]	Hyena	Yeast promoter sequences	CLM	Base-res	Yes
RNA-FM [39]	BERT	RNAcentral (promoters)	MLM	Base-res	Yes
Species-aware DNA LM [28]	BERT	UTR of 1500 yeast genomes	MLM	Base-res	No
FloraBERT [41]	BERT	Multi-species plant genome	MLM	Byte-pair encoding	Yes
RandomMask [42]	BERT	Human genome	MLM (expanding size)	Overlapping k-mer	Yes
GenSLMs[40]	BERT	BV-BRC dataset	Contrastive loss	Codon	Yes
Evo [78]	StripedHyena	prokaryotic genomes and plasmids	CLM	Base-res	Both
Mamba [43]	Mamba	human genome	CLM	Base-res	Both
Caduceus [46]	Bi-directional Mamba	prokaryotic genomes and plasmids	MLM	Base-res	Yes

Appendix B

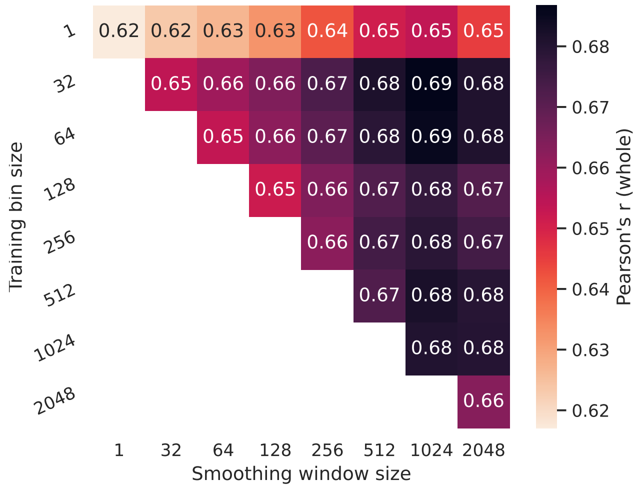
Chapter 2 Supplemental Figures



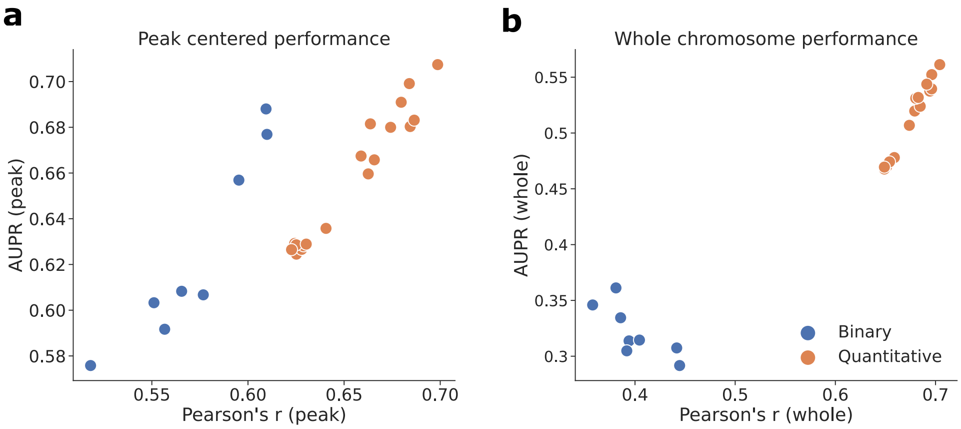
SUPPLEMENTARY FIGURE B.1: Hyperparamter search with WandB. Parallel plot for different hyperparameters used in a grid search for (**a**) BPNet-base and (**b**) Basenji. (**a**) BPNet-base optimized hyperparameters were the number of kernels in the first convolutional layer (filtN_1), which is then fixed throughout all subsequent convolutional layers, and the kernel size (kern_3) of the task-specific output heads. (**b**) Basenji-based optimized hyperparameters specifying the number of filters in the first convolutional layer (filtN_1) and the second convolutional layer (filtN_2), which correspondingly sets the number of filters in the subsequent residual blocks.



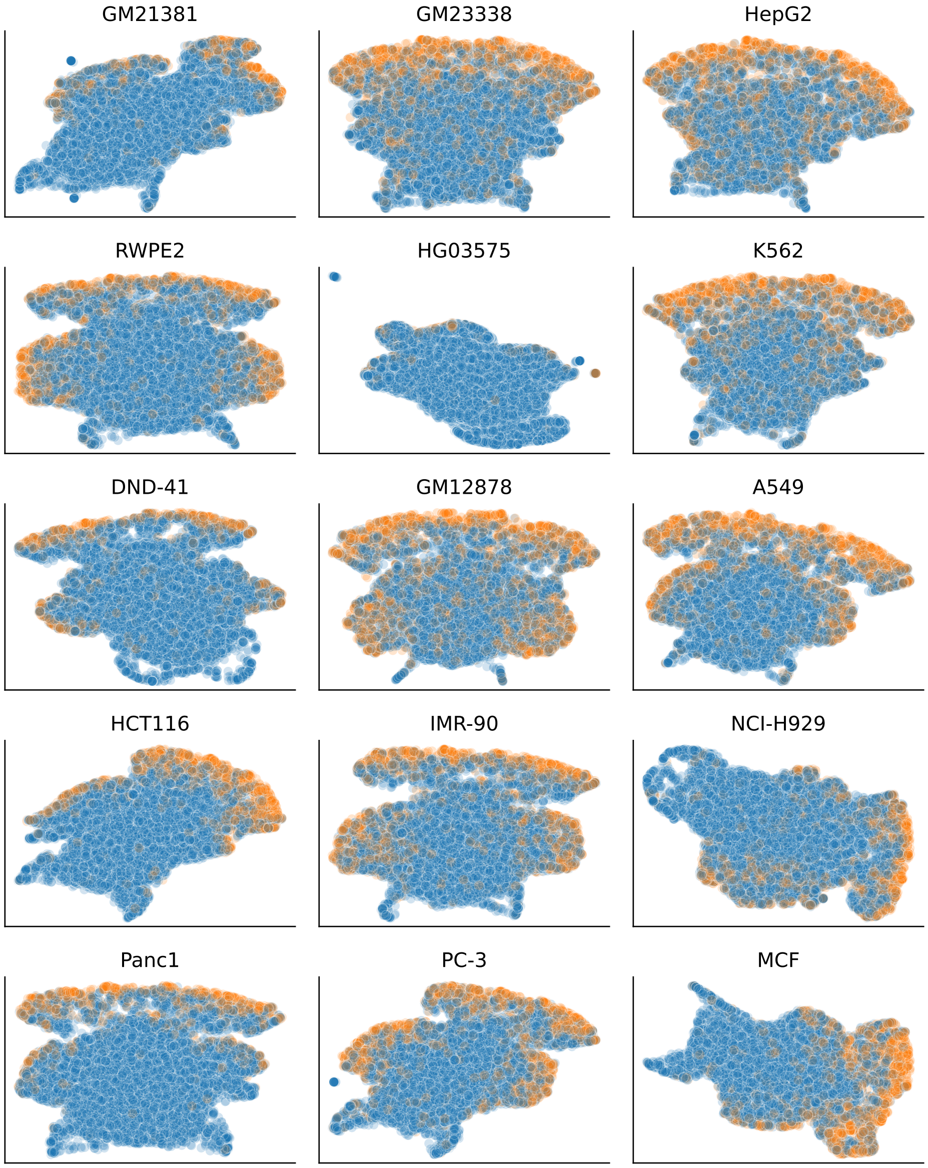
SUPPLEMENTARY FIGURE B.2: Evaluation of BPNet-based quantitative models. (**a**) *Loss function analysis.* Scatter plot of the whole-chromosome Pearson's r versus the MSE for different loss functions (shown in a different color) and different target resolutions (shown in a different marker). The results for the scaled Pearson's r loss function was removed due to poor training runs. (**b**) *Bin resolution analysis*. Plot of the whole-chromosome Pearson's r for models trained on a given bin size (y-axis) with predictions that were systematically down-sampled to a lower resolution for evaluation (x-axis). (**a**,**b**) Pearson's r represents the average across cell lines.



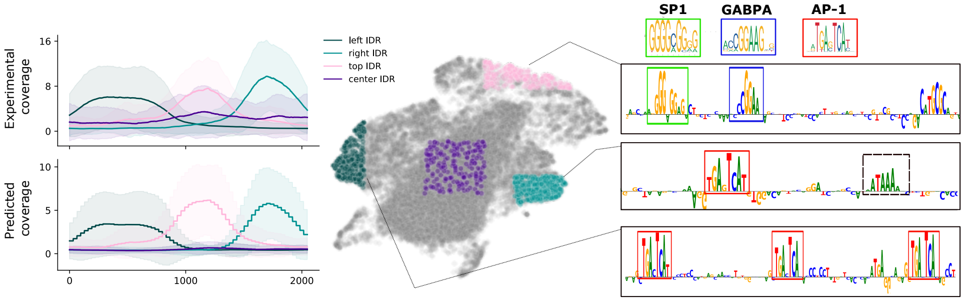
SUPPLEMENTARY FIGURE B.3: The effect of smoothing coverage on performance. Basenji-based models were trained on target resolutions (y-axis) and evaluated using different levels of smoothing with a box-car filter. For each higher resolution model, a box-car filter was applied to both predictions and experimental coverage values with various kernel sizes prior to calculating the average Pearson's r (x-axis). Pearson's r represents the average across cell lines.



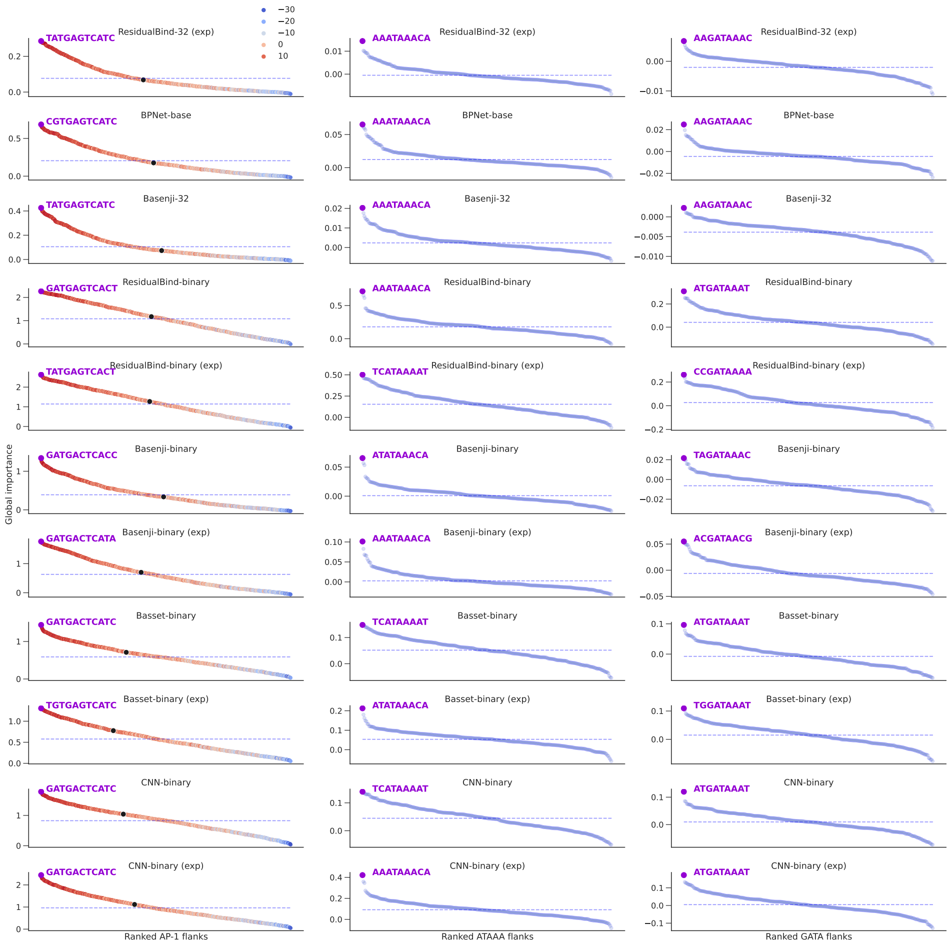
SUPPLEMENTARY FIGURE B.4: Performance comparison between quantitative and binary models. Scatter plot of the classification-based AUPR versus the regression-based Pearson's r for various binary models (blue) and quantitative models (orange) on peak-centered test data (left) and whole-chromosome test data (right). Metrics represent the average across cell lines.



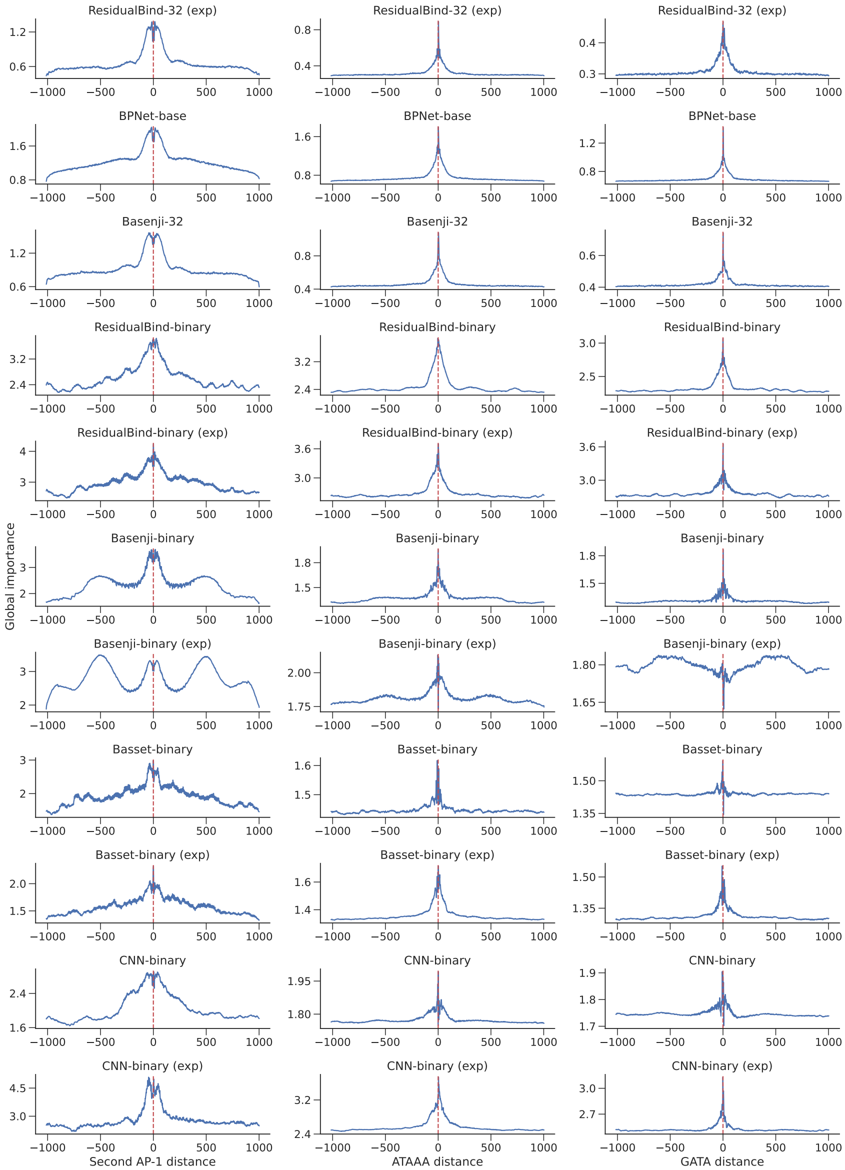
SUPPLEMENTARY FIGURE B.5: UMAP embeddings of the test set representations for Residualbind-32 with single-task outputs and exponential activations. For each cell line, embedded sequences were selected based on the coverage value above a threshold of 2. Orange dots indicate sequences that overlap with a statistically significant peak.



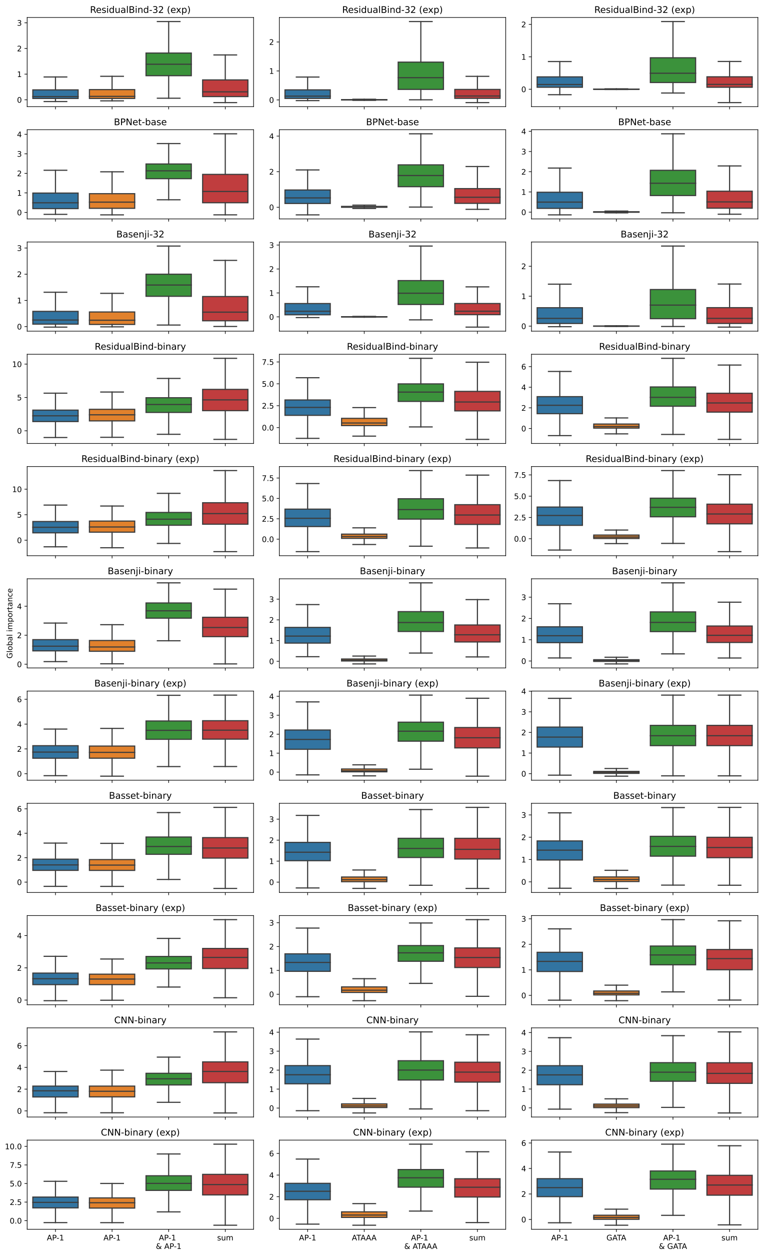
SUPPLEMENTARY FIGURE B.6: UMAP embedding of the penultimate layer representations across all test sequences. (Left) Shows the average experimental coverage and predicted coverage from sequences that map to specific locations in the embedding (shown in a different color). (Right) Representative saliency maps (zoomed in) for sequences within different embedding regions. The known motifs from the JASPAR database are shown at the top and an unknown 'ATAAA' motif is annotated with a box with black dashed lines.



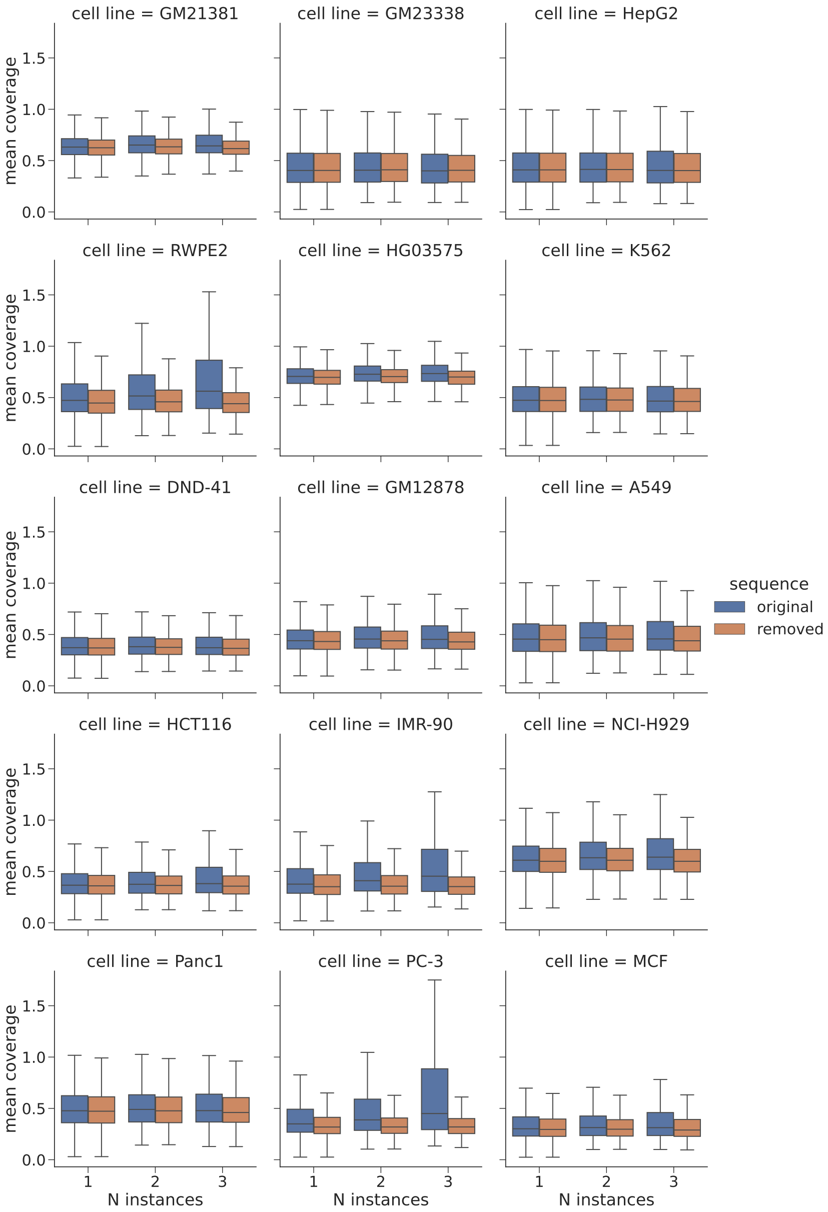
SUPPLEMENTARY FIGURE B.7: GIA for optimal flanking nucleotides of motifs in PC-3 cell line for various models. Ranked plot of the global importance for each tested flank for AP-1 motif (left column), ATAAA motif (middle column) and GATA (right column) for different models (shown in a different row). Dashed line represents the global importance of the core motif with random flanks. The hue in the first column represents the position-weight-matrix score for an AP-1 motif from the JASPAR database (ID: MA0491.1). The first 3 rows are quantitative models, the rest are binary models (with (exp) in the name indicating that the first layer ReLU activation has been replaced with an exponential function). For binary models, the results are based on the logits before the output sigmoid activation. The hue in the first column plots represents the PWM score for an AP-1 motif from the JASPAR database (ID: MA0491.1). The black dot in each plot (in the first column) indicates “TGTGATTCATG”, which has a high PWM score (12.800) but yields a global importance close to the core motif with randomized flanks.



SUPPLEMENTARY FIGURE B.8: GIA for distance dependence between AP-1 and other motifs for PC-3 cell line for various models. Global importance plot for sequences with an AP-1 motif fixed at the center of the sequence and another motif that is systematically placed in different non-overlapping locations. First column shows results where the second motif is an identical AP-1 motif, the center column shows results for ATAAA motif and right column for the GATA motif. All the motifs were embedded with optimized flanks. Red vertical dashed lines indicate the 1024bp position. Each row corresponds to a different trained model, the first 3 are quantitative models, the rest are binary models (with (exp) in the name indicating that the first layer ReLU activation has been replaced with an exponential function). For binary models, the results are based on the logits before the output sigmoid activation.



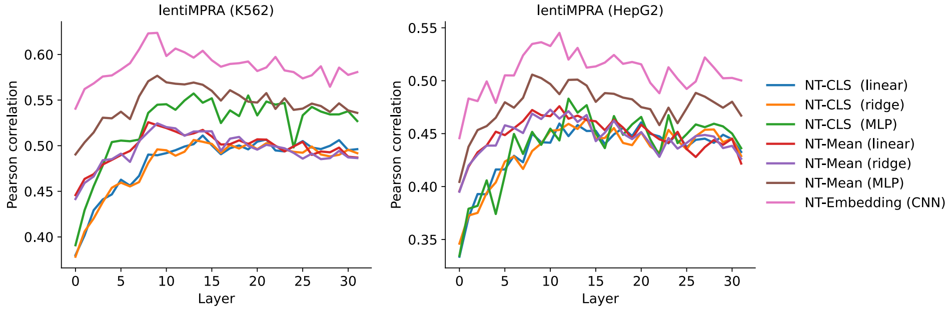
Supplementary Figure B.9: GIA for cooperative interactions between AP-1 and other motifs for PC-3 cell line for various models. Each column corresponds to a motif pair between two copies of AP-1, ATAAA and AP-1 and AP-1 and GATA. Each row corresponds to a different trained model, the first 3 are quantitative models, the rest are binary models ((exp)indicating that the first layer uses exponential activation). For binary models, the results are based on the logits before the output sigmoid activation. The pairs were embedded at the optimal distance specified from the distance dependence GIA experiments. For each motif pair experiment n=1000 independent samples were drawn from the test set sequences.



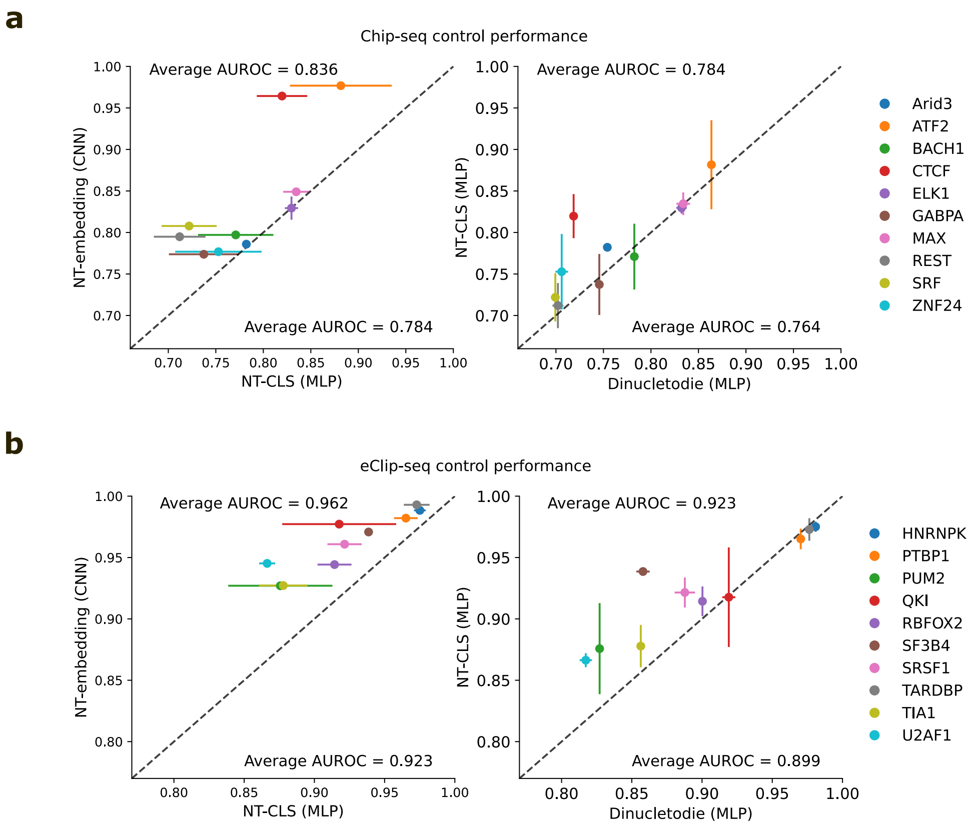
SUPPLEMENTARY FIGURE B.10: Occlusion analysis for AP-1 motifs using Residualbind-32 with single-task outputs and exponential activations. Randomly sampled sequences, n=10,000, from the test set were used. Box-plots of the mean coverage value of predictions for sequences from the sample that contained 1, 2, and 3 (or more) AP-1 motifs and the mean coverage values when the AP-1 motifs are replaced by randomized sequences (averaged across 20 random trials) for each cell line. n=3,381, n=1,310, and n=323 sequences contained 1, 2, and 3 instances of the AP-1 motif, respectively. Box plots show the first and third quartiles, central line is the median, and the whiskers show the range of data with outliers removed.

Appendix C

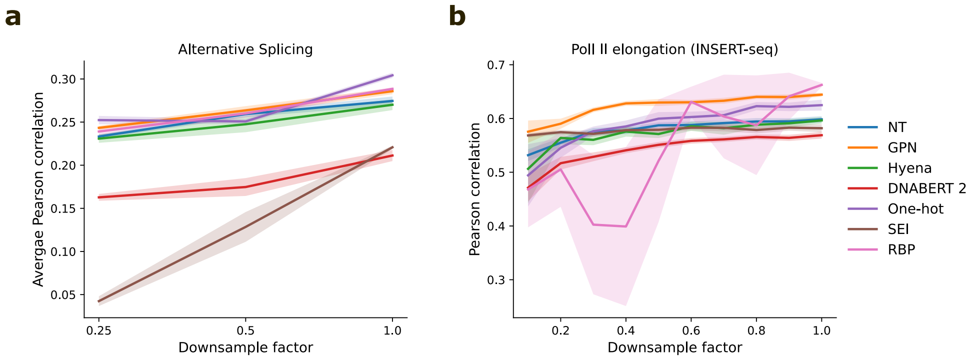
Chapter 3 Supplemental Figures



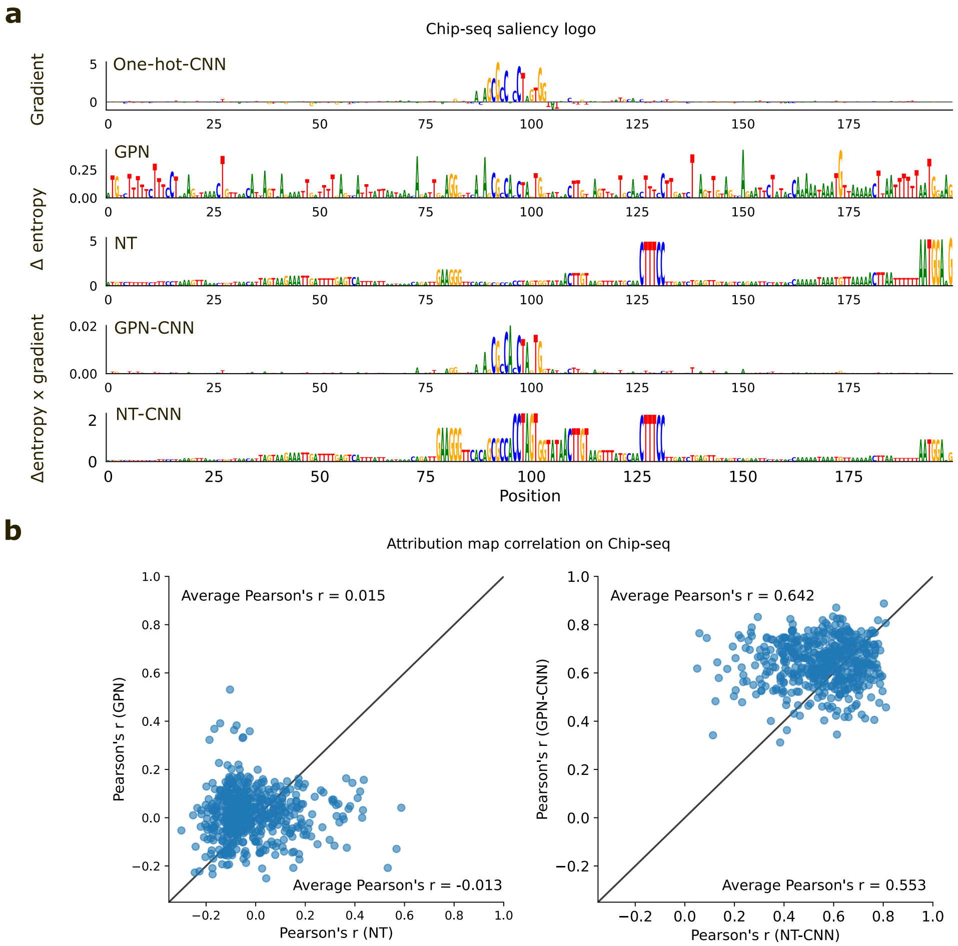
SUPPLEMENTARY FIGURE C.1: Layer-wise performance of Nucleotide-Transformer on the lentiMPRA dataset. Test performance of various machine learning models trained using embeddings from different layers of Nucleotide-Transformer. Embeddings include the CLS token, mean embedding (Mean), and the full embedding (Embedding). Machine learning models include linear regression (linear), ridge regression (ridge), multi-layer perceptron (MLP) and a convolutional neural network (CNN).



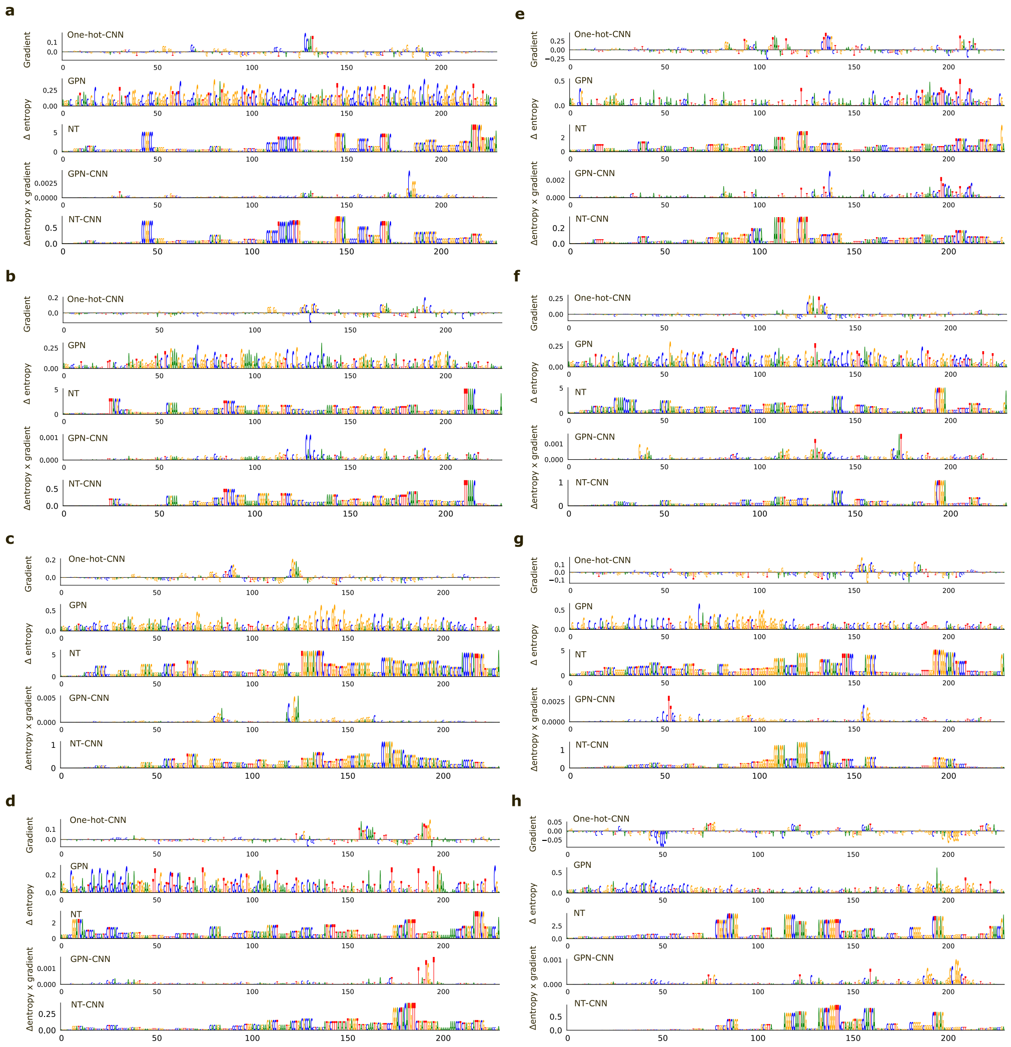
SUPPLEMENTARY FIGURE C.2: Control experiments with different embeddings. Performance comparison between a CNN trained using full embeddings of the penultimate layer from Nucleotide-Transformer, an MLP trained using Nucleotide-Transformer's CLS token, and an MLP trained using dinucleotide frequencies of the sequence on (**a**) ChIP-seq data and (**b**) eCLIP-seq data. Performance is measured by the average area-under the receiver-operating characteristic curve (AUROC) and error bars represent the standard deviation of the mean across 5 different random initializations. Text valeus represent the average AUROC across all ChIP-seq or CLIP-seq datasets.



SUPPLEMENTARY FIGURE C.3: Down-sampling performance on RNA regulation tasks. Average performance of machine learning models on (**a**) alternative splicing, task 4, and (**b**) RNA Pol II elongation potential, task 5, down-sampled by various factors. Shaded region represents standard deviation of the mean across 5 different random initializations.



Supplementary Figure C.4: Attribution analysis comparison for sequences from CTCF ChIP-seq data. **a**, Representative example of attribution maps for a CTCF binding sequence. Attribution maps include (top to bottom): the gradient-times-input of a one-hot-trained CNN; the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained GPN; the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained Nucleotide-Transformer; the gradient of a CNN-trained using GPN embeddings multiplied by the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained GPN; and the gradient of a CNN-trained using Nucleotide-Transformer embeddings multiplied by the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained Nucleotide-Transformer. **b**, Scatter plot comparison of the attribution map correlations for different pre-trained gLMs (left) and CNNs trained using gLM embeddings (right). Attribution map correlations reflect the Pearson correlation coefficient between the attribution map generated by the gLM-based attribution method with the Saliency Map generated by a one-hot-trained CNN. Each dot represents a different sequence in the CTCF ChIP-seq dataset (N=500).



SUPPLEMENTARY FIGURE C.5: Representative examples of attribution maps for sequences from the lentiMPRA dataset. In each panel, attribution maps are shown for different sequences in order of (top to bottom): the gradient-times-input of a one-hot-trained CNN; the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained GPN; the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained Nucleotide-Transformer; the gradient of a CNN-trained using GPN embeddings multiplied by the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained GPN; and the gradient of a CNN-trained using Nucleotide-Transformer embeddings multiplied by the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained Nucleotide-Transformer.

Bibliography

Abadi, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

Agarwal, Vikram and David R Kelley (2022). “The genetic and biochemical determinants of mRNA degradation rates in mammals”. In: *Genome Biology* 23.1, p. 245.

Agarwal, Vikram and Jay Shendure (2020). “Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks”. In: *Cell reports* 31.7.

Agarwal, Vikram et al. (2023). “Massively parallel characterization of transcriptional regulatory elements in three diverse human cell types”. In: *bioRxiv*.

Alexandari, Amr M et al. (2023). “De novo distillation of thermodynamic affinity from deep learning regulatory sequence models of in vivo protein-DNA binding”. In: *bioRxiv*.

Alipanahi, Babak et al. (July 2015a). “Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning”. In: *Nature Biotechnology* 33.8, pp. 831–838.

— (2015b). “Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning”. In: *Nature biotechnology* 33.8, pp. 831–838.

Almeida, Bernardo P de et al. (2022). “DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers”. In: *Nature Genetics* 54.5, pp. 613–624.

Almeida, Bernardo P de et al. (2024). “Targeted design of synthetic enhancers for selected tissues in the Drosophila embryo”. In: *Nature* 626.7997, pp. 207–211.

Ameen, Mohamed et al. (2022). “Integrative single-cell analysis of cardiogenesis identifies developmental trajectories and non-coding mutations in congenital heart disease”. In: *Cell* 185.26, pp. 4937–4953.

“An integrated encyclopedia of DNA elements in the human genome” (Sept. 2012). In: *Nature* 489.7414, pp. 57–74.

Angermueller, Christof et al. (2016). “Deep learning for computational biology”. In: *Molecular systems biology* 12.7, p. 878.

Angermueller, Christof et al. (2017). “DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning”. In: *Genome biology* 18, pp. 1–13.

Araujo, André, Wade Norris, and Jack Sim (2019). “Computing receptive fields of convolutional neural networks”. In: *Distill* 4.11, e21.

Arnold, Cosmas D et al. (2013). “Genome-wide quantitative enhancer activity maps identified by STARR-seq”. In: *Science* 339.6123, pp. 1074–1077.

Asif, Maor and Yaron Orenstein (2020). “DeepSELEX: inferring DNA-binding preferences from HT-SELEX data using multi-class CNNs”. In: *Bioinformatics* 36.Supplement_2, pp. i634–i642.

Atak, Zeynep Kalender et al. (2021). “Interpretation of allele-specific chromatin accessibility using cell state–aware deep learning”. In: *Genome Research* 31.6, pp. 1082–1096.

Avsec, Žiga et al. (2021a). “Base-resolution models of transcription-factor binding reveal soft motif syntax”. In: *Nature genetics* 53.3, pp. 354–366.

Avsec, Žiga et al. (Feb. 2021b). “Base-resolution models of transcription-factor binding reveal soft motif syntax”. In: *Nature Genetics* 53.3, pp. 354–366.

Avsec, Žiga et al. (2021c). “Effective gene expression prediction from sequence by integrating long-range interactions”. In: *Nature methods* 18.10, pp. 1196–1203.

Avsec, Žiga et al. (Oct. 2021d). “Effective gene expression prediction from sequence by integrating long-range interactions”. In: *Nature Methods* 18.10, pp. 1196–1203.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E Hinton (2016). “Layer normalization”. In: *arXiv preprint arXiv:1607.06450*.

Baeza-Centurion, Pablo et al. (2019). “Combinatorial genetics reveals a scaling law for the effects of mutations on splicing”. In: *Cell* 176.3, pp. 549–563.

Banerji, Julian, Sandro Rusconi, and Walter Schaffner (1981). “Expression of a β-globin gene is enhanced by remote SV40 DNA sequences”. In: *Cell* 27.2, pp. 299–308.

Barash, Yoseph et al. (2003). “Modeling dependencies in protein-DNA binding sites”. In: *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pp. 28–37.

Belton, Jon-Matthew et al. (2012). “Hi–C: a comprehensive technique to capture the conformation of genomes”. In: *Methods* 58.3, pp. 268–276.

Benegas, Gonzalo, Sanjit Singh Batra, and Yun S Song (2023). “DNA language models are powerful predictors of genome-wide variant effects”. In: *Proceedings of the National Academy of Sciences* 120.44, e2311219120.

Benegas, Gonzalo et al. (2023). “GPN-MSA: an alignment-based DNA language model for genome-wide variant effect prediction”. In: *bioRxiv*.

Bengio, Yoshua and Yannick Pouliot (1990). “Efficient recognition of immunoglobulin domains from amino acid sequences using a neural network”. In: *Bioinformatics* 6.4, pp. 319–324.

Bepler, Tristan and Bonnie Berger (2021). “Learning the protein language: Evolution, structure, and function”. In: *Cell systems* 12.6, pp. 654–669.

Berg, Otto G and Peter H von Hippel (1987). “Selection of DNA binding sites by regulatory proteins: Statistical-mechanical theory and application to operators and promoters”. In: *Journal of molecular biology* 193.4, pp. 723–743.

Berger, Michael F et al. (2006). “Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities”. In: *Nature biotechnology* 24.11, pp. 1429–1435.

Biewald, Lukas (2020). *Experiment Tracking with Weights and Biases*. Software available from wandb.com. URL: https://www.wandb.com/.

Boureau, Y-Lan, Jean Ponce, and Yann LeCun (2010). “A theoretical analysis of feature pooling in visual recognition”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 111–118.

Brandes, Nadav et al. (2023). “Genome-wide prediction of disease variant effects with a deep protein language model”. In: *Nature Genetics* 55.9, pp. 1512–1522.

Brennan, Kaelan J et al. (2023). “Chromatin accessibility in the Drosophila embryo is determined by transcription factor pioneering and enhancer activation”. In: *Developmental cell* 58.19, pp. 1898–1916.

Buenrostro, Jason D et al. (2015). “ATAC-seq: a method for assaying chromatin accessibility genome-wide”. In: *Current Protocols in Molecular Biology* 109.1, pp. 21–29.

Bulyk, Martha L, Philip LF Johnson, and George M Church (2002). “Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors”. In: *Nucleic acids research* 30.5, pp. 1255–1261.

Castro-Mondragon, Jaime A et al. (Nov. 2021). “JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles”. In: *Nucleic Acids Research* 50.D1, pp. D165–D173.

Cazares, Tareian A et al. (2023). “maxATAC: Genome-scale transcription-factor binding prediction from ATAC-seq with deep neural networks”. In: *PLOS Computational Biology* 19.1, e1010863.

Chen, Jiayang et al. (2022a). “Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions”. In: *arXiv preprint arXiv:2204.00300*.

Chen, Kathleen M et al. (2019). “Selene: a PyTorch-based deep learning library for sequence data”. In: *Nature Methods* 16.4, pp. 315–318.

Chen, Kathleen M et al. (2022b). “A sequence-based global map of regulatory activity for deciphering human genetics”. In: *Nature Genetics*, pp. 1–10.

Chen, Ken et al. (2023). “Self-supervised learning on millions of pre-mRNA sequences improves sequence-based RNA splicing prediction”. In: *bioRxiv*.

Chen, Lu et al. (2022c). “Real-time spatial registration for 3D human atlas”. In: *Proceedings of the 10th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, pp. 27–35.

Cheng, Jun et al. (2021). “MTSplice predicts effects of genetic variants on tissue-specific splicing”. In: *Genome Biology* 22.1, pp. 1–19.

Ching, Travers et al. (2018). “Opportunities and obstacles for deep learning in biology and medicine”. In: *Journal of the royal society interface* 15.141, p. 20170387.

Chowdhury, Ratul et al. (2022). “Single-sequence protein structure prediction using a language model and deep learning”. In: *Nature Biotechnology* 40.11, pp. 1617–1623.

Chu, Yanyi et al. (2023). “A 5’UTR Language Model for Decoding Untranslated Regions of mRNA and Function Predictions”. In: *bioRxiv*, pp. 2023–10.

Clauwaert, Jim, Gerben Menschaert, and Willem Waegeman (2021). “Explainability in transformer models for functional genomics”. In: *Briefings in bioinformatics* 22.5, bbab060.

Cohen, Jeremy, Elan Rosenfeld, and Zico Kolter (2019). “Certified adversarial robustness via randomized smoothing”. In: *International Conference on Machine Learning*. PMLR, pp. 1310–1320.

Consens, Micaela E et al. (2023). “To Transformers and Beyond: Large Language Models for the Genome”. In: *arXiv 2311.07621*.

Consortium, 1000 Genomes Project et al. (2015). “A global reference for human genetic variation”. In: *Nature* 526.7571, p. 68.

Consortium, ENCODE Project et al. (2012). “An integrated encyclopedia of DNA elements in the human genome”. In: *Nature* 489.7414, p. 57.

Crawshaw, Michael (2020). “Multi-task learning with deep neural networks: A survey”. In: *arXiv preprint arXiv:2009.09796*.

Cuperus, Josh T et al. (2017). “Deep learning of the regulatory grammar of yeast 5 untranslated regions from 500,000 random sequences”. In: *Genome research* 27.12, pp. 2015–2024.

Cybenko, George (1989). “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4, pp. 303–314.

Dalla-Torre, Hugo et al. (2023). “The nucleotide transformer: Building and evaluating robust foundation models for human genomics”. In: *bioRxiv*, pp. 2023–01.

Dao, Tri et al. (2022). “Flash attention: Fast and memory-efficient exact attention with io-awareness”. In: *Advances in Neural Information Processing Systems* 35, pp. 16344–16359.

Dekker, Job et al. (2002). “Capturing chromosome conformation”. In: *science* 295.5558, pp. 1306–1311.

Della Chiara, Giulia et al. (2023). “Enhancers dysfunction in the 3D genome of cancer cells”. In: *Frontiers in Cell and Developmental Biology* 11.

Devlin, Jacob et al. (2018). “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv 1810.04805*.

Dey, Kushal K et al. (2020). “Evaluating the informativeness of deep learning annotations for human complex diseases”. In: *Nature Communications* 11.1, pp. 1–9.

Diao, Yarui et al. (2016). “A new class of temporarily phenotypic enhancers identified by CRISPR/Cas9-mediated genetic screening”. In: *Genome research* 26.3, pp. 397–405.

Diederik, P Kingma (2014). “Adam: A method for stochastic optimization”. In: *(No Title)*.

Doni Jayavelu, N et al. (2020). *Candidate silencer elements for the human and mouse genomes. Nat. Commun. 11, 1061*.

Dudnyk, Kseniia, Chenlai Shi, and Jian Zhou (2023). “Sequence basis of transcription initiation in human genome”. In: *bioRxiv*.

Duncan, Andrew G, Jennifer A Mitchell, and Alan M Moses (2023). “Improving the performance of supervised deep learning for regulatory genomics using phylogenetic augmentation”. In: *bioRxiv*, pp. 2023–09.

Eddy, Sean R (2012). “The C-value paradox, junk DNA and ENCODE”. In: *Current biology* 22.21, R898–R899.

Elnaggar, Ahmed et al. (2021). “Prottrans: Toward understanding the language of life through self-supervised learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10.

Er, Meng Joo et al. (2016). “Attention pooling-based convolutional neural network for sentence modelling”. In: *Information Sciences* 373, pp. 388–403.

Eraslan, Gökcen et al. (2019). “Deep learning: new computational modelling techniques for genomics”. In: *Nature Reviews Genetics* 20.7, pp. 389–403.

Ferruz, Noelia and Birte Höcker (2022). “Controllable protein design with language models”. In: *Nature Machine Intelligence* 4.6, pp. 521–532.

Fishman, Veniamin et al. (2023). “GENA-LM: A Family of Open-Source Foundational Models for Long DNA Sequences”. In: *bioRxiv*, pp. 2023–06.

Foat, Barrett C, Alexandre V Morozov, and Harmen J Bussemaker (2006). “Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE”. In: *Bioinformatics* 22.14, e141–e149.

Fudenberg, Geoff, David R Kelley, and Katherine S Pollard (2020). “Predicting 3D genome folding from DNA sequence with Akita”. In: *Nature methods* 17.11, pp. 1111–1117.

Fulco, Charles P et al. (2016). “Systematic mapping of functional enhancer–promoter connections with CRISPR interference”. In: *Science* 354.6313, pp. 769–773.

Fulco, Charles P et al. (2019). “Activity-by-contact model of enhancer–promoter regulation from thousands of CRISPR perturbations”. In: *Nature genetics* 51.12, pp. 1664–1669.

Gasperini, Molly, Jacob M Tome, and Jay Shendure (2020). “Towards a comprehensive catalogue of validated and target-linked human enhancers”. In: *Nature Reviews Genetics* 21.5, pp. 292–310.

Gasperini, Molly et al. (2017). “CRISPR/Cas9-mediated scanning for regulatory elements required for HPRT1 expression via thousands of large, programmed genomic deletions”. In: *The American Journal of Human Genetics* 101.2, pp. 192–205.

Ge, Wanwan et al. (2021). “Bayesian Markov models improve the prediction of binding motifs beyond first order”. In: *NAR Genomics and Bioinformatics* 3.2, lqab026.

Ghotra, Rohan et al. (July 2021). “Designing Interpretable Convolution-Based Hybrid Networks for Genomics”. In: *bioRxiv*.

Glorot, Xavier and Yoshua Bengio (2010). “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 249–256.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy (2014). “Explaining and Harnessing Adversarial Examples”. In: *arXiv 1412.6572*.

Graur, Dan et al. (2013). “On the immortality of television sets:“function” in the human genome according to the evolution-free gospel of ENCODE”. In: *Genome Biology and Evolution* 5.3, pp. 578–590.

Gruver, Nate et al. (2024). “Protein design with guided discrete diffusion”. In: *Advances in Neural Information Processing Systems* 36.

Gu, Albert and Tri Dao (2023). “Mamba: Linear-time sequence modeling with selective state spaces”. In: *arXiv preprint arXiv:2312.00752*.

Gündüz, Hüseyin Anil et al. (2023). “A self-supervised deep learning method for data-efficient training in genomics”. In: *Communications Biology* 6.1, p. 928.

Gupta, Shobhit et al. (2007). “Quantifying similarity between motifs”. In: *Genome Biology* 8.2, pp. 1–9.

Hallee, Logan, Nikolaos Rafailidis, and Jason P Gleghorn (2023). "cdsBERT-Extending Protein Language Models with Codon Awareness". In: *bioRxiv*.

Hannenhalli, Sridhar and Li-San Wang (2005). "Enhanced position weight matrices using mixture models". In: *ISMB (Supplement of Bioinformatics)*. Citeseer, pp. 204–212.

He, Kaiming et al. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

— (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Heintzman, Nathaniel D et al. (2009). "Histone modifications at human enhancers reflect global cell-type-specific gene expression". In: *Nature* 459.7243, pp. 108–112.

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian error linear units (GeLUs)". In: *arXiv 1606.08415*.

Heumann, J, A Lapedes, and G Stormo (1995). "Alignment of regulatory sites using neural networks to maximize specificity". In: *Proceedings of the 1995 World Congress on Neural NetworksII*, pp. 771–775.

Hie, Brian L, Kevin K Yang, and Peter S Kim (2022). "Evolutionary velocity with protein language models predicts evolutionary dynamics of diverse proteins". In: *Cell Systems* 13.4, pp. 274–285.

Hie, Brian L et al. (2023). "Efficient evolution of human antibodies from general protein language models". In: *Nature Biotechnology*.

Ho, Jonathan et al. (2019). "Axial attention in multidimensional transformers". In: *arXiv 1912.12180*.

Hoffmann, Jordan et al. (2022). “Training compute-optimal large language models”. In: *arXiv 2203.15556*.

Hsieh, Tsung-Han S et al. (2015). “Mapping nucleosome resolution chromosome folding in yeast by micro-C”. In: *Cell* 162.1, pp. 108–119.

Hu, Edward J et al. (2021). “Lora: Low-rank adaptation of large language models”. In: *arXiv 2106.09685*.

Hu, Jie, Li Shen, and Gang Sun (2018). “Squeeze-and-excitation networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141.

Huang, Connie et al. (2023). “Personal transcriptome variation is poorly explained by current genomic deep learning models”. In: *Nature Genetics* 55.12, pp. 2056–2059.

Huang, Yong-Heng et al. (2015). “SOXE transcription factors form selective dimers on non-compact DNA motifs through multifaceted interactions between dimerization and high-mobility group domains”. In: *Scientific reports* 5.1, p. 10398.

Inukai, Sachi, Kian Hong Kock, and Martha L Bulyk (2017). “Transcription factor–DNA binding: beyond binding site motifs”. In: *Current opinion in genetics & development* 43, pp. 110–119.

Ioffe, Sergey and Christian Szegedy (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International Conference on Machine Learning*, pp. 448–456.

Izmailov, Pavel et al. (2018). “Averaging weights leads to wider optima and better generalization”. In: *arXiv preprint arXiv:1803.05407*.

Jaganathan, Kishore et al. (2019). “Predicting splicing from primary sequence with deep learning”. In: *Cell* 176.3, pp. 535–548.

Jain, Samyak et al. (2023). “Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks”. In: *arXiv preprint arXiv:2311.12786*.

Janssens, Jasper et al. (2022). “Decoding gene regulation in the fly brain”. In: *Nature*, pp. 1–7.

Ji, Yanrong et al. (2021). “DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome”. In: *Bioinformatics* 37.15.

Jin, Fulai et al. (2013). “A high-resolution map of the three-dimensional chromatin interactome in human cells”. In: *Nature* 503.7475, pp. 290–294.

Jindal, Granton A and Emma K Farley (2021). “Enhancer grammar in development, evolution, and disease: dependencies and interplay”. In: *Developmental cell* 56.5, pp. 575–587.

Karbalayghareh, Alireza, Merve Sahin, and Christina S Leslie (2022). “Chromatin interaction–aware gene regulatory modeling with graph attention networks”. In: *Genome Research* 32.5, pp. 930–944.

Karollus, Alexander et al. (2023). “Species-aware DNA language models capture regulatory elements and their evolution”. In: *bioRxiv*.

Keilwagen, Jens and Jan Grau (2015). “Varying levels of complexity in transcription factor binding motifs”. In: *Nucleic acids research* 43.18, e119–e119.

Kelley, David R (2020). “Cross-species regulatory sequence activity prediction”. In: *PLoS Computational Biology* 16.7, e1008050.

Kelley, David R, Jasper Snoek, and John L Rinn (2016). “Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks”. In: *Genome research* 26.7, pp. 990–999.

Kelley, David R et al. (2018a). “Sequential regulatory activity prediction across chromosomes with convolutional neural networks”. In: *Genome research* 28.5, pp. 739–750.

Kelley, David R. et al. (Mar. 2018b). "Sequential regulatory activity prediction across chromosomes with convolutional neural networks". In: *Genome Research* 28.5, pp. 739–750.

Kim, D et al. (2020). *The dynamic, combinatorial cis-regulatory lexicon of epidermal differentiation. bioRxiv 2020.10. 16.342857*.

Kim, Daniel S et al. (2021). "The dynamic, combinatorial cis-regulatory lexicon of epidermal differentiation". In: *Nature Genetics* 53.11, pp. 1564–1576.

Kim, Seungsoo and Joanna Wysocka (2023). "Deciphering the multi-scale, quantitative cis-regulatory code". In: *Molecular cell* 83.3, pp. 373–392.

Kingma, Diederik and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv*, p. 1412.6980.

Kinney, Justin B et al. (2010). "Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence". In: *Proceedings of the National Academy of Sciences* 107.20, pp. 9158–9163.

Kircher, Martin et al. (2019). "Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution". In: *Nature Communications* 10.1, pp. 1–15.

Klie, Adam et al. (2023). "Predictive analyses of regulatory sequences with EUGENe". In: *Nature Computational Science* 3.11, pp. 946–956.

Kodzius, Rimantas et al. (2006). "CAGE: cap analysis of gene expression". In: *Nature Methods* 3.3, pp. 211–222.

Koh, PW, E Pierson, and A Kundaje (2017). *Denoising genome-wide histone chipseq with convolutional neural networks. Bioinformatics 33, i225–i233*.

Koo, Peter K and Sean R Eddy (2019). "Representation learning of genomic sequence motifs with convolutional neural networks". In: *PLoS Computational Biology* 15.12, e1007560.

Koo, Peter K and Matt Ploenzke (2020). “Deep learning for inferring transcription factor binding sites”. In: *Current Opinion in Systems Biology* 19, pp. 16–23.

— (2021). “Improving representations of genomic sequence motifs in convolutional networks with exponential activations”. In: *Nature Machine Intelligence* 3.3, pp. 258–266.

Koo, Peter K et al. (2021). “Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks”. In: *PLoS Computational Biology* 17.5, e1008925.

Koohy, Hashem et al. (May 2014). “A Comparison of Peak Callers Used for DNase-Seq Data”. In: *PLoS ONE* 9.5. Ed. by Manuela Helmer-Citterich, e96303.

Kopp, Wolfgang et al. (2020). “Deep learning for genomics using Janggu”. In: *Nature Communications* 11.1, pp. 1–7.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25.

Kwasnieski, Jamie C et al. (2012). “Complex effects of nucleotide variants in a mammalian cis-regulatory element”. In: *Proceedings of the National Academy of Sciences* 109.47, pp. 19498–19503.

Lal, Avantika, Tommaso Biancalani, and Gökcen Eraslan (2023). “regLM: Designing realistic regulatory DNA with autoregressive language models”. In: *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*.

Le, Daniel D et al. (2018). “Comprehensive, high-resolution binding energy landscapes reveal context dependencies of transcription factor binding”. In: *Proceedings of the National Academy of Sciences* 115.16, E3702–E3711.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *nature* 521.7553, pp. 436–444.

Lee, Nicholas Keone et al. (2023). “EvoAug: improving generalization and interpretability of genomic deep neural networks with evolution-inspired data augmentations”. In: *Genome Biology* 24.1, p. 105.

Lester, Brian, Rami Al-Rfou, and Noah Constant (2021). “The power of scale for parameter-efficient prompt tuning”. In: *arXiv 2104.08691*.

Levo, Michal and Eran Segal (2014). “In pursuit of design principles of regulatory sequences”. In: *Nature Reviews Genetics* 15.7, pp. 453–468.

Levo, Michal et al. (2015). “Unraveling determinants of transcription factor binding outside the core binding site”. In: *Genome Research* 25.7, pp. 1018–1029.

Levy, Benjamin et al. (2022). “FloraBERT: cross-species transfer learning withattention-based neural networks for geneexpression prediction”. In.

Li, Francesca-Zhoufan et al. (2024). “Feature Reuse and Scaling: Understanding Transfer Learning with Protein Language Models”. In: *bioRxiv*, pp. 2024–02.

Li, Hongyang, Daniel Quang, and Yuanfang Guan (2019). “Anchor: trans-cell type prediction of transcription factor binding sites”. In: *Genome Research* 29.2, pp. 281–292.

Li, Jiawei et al. (2021). “DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences”. In: *Briefings in Bioinformatics* 22.3, bbaa159.

Li, Sizhen et al. (2023). “CodonBERT: Large Language Models for mRNA design and optimization”. In: *bioRxiv*.

Liang, Chaoqi et al. (2023). “Rethinking the BERT-like Pretraining for DNA Sequences”. In: *arXiv 2310.07644*.

Lin, Xueqiu et al. (2022a). “Nested epistasis enhancer networks for robust genome regulation”. In: *Science* 377.6610, pp. 1077–1085.

Lin, Zeming et al. (2022b). “Language models of protein sequences at the scale of evolution enable accurate structure prediction”. In: *BioRxiv* 2022, p. 500902.

Linder, Johannes et al. (2023). “Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation”. In: *bioRxiv*, pp. 2023–08.

Ling, Jonathan P et al. (2020). “ASCOT identifies key regulators of neuronal subtype-specific splicing”. In: *Nature Communications* 11.1.

Liu, Brandon (n.d.). “Kipoi: Accelerating the Community Exchange and Reuse of Predictive Models for Genomics”. In: ().

Liu, Haokun et al. (2022a). “Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning”. In: *Advances in Neural Information Processing Systems* 35, pp. 1950–1965.

Liu, Huaqing et al. (2024). “Exploring Genomic Large Language Models: Bridging the Gap between Natural Language and Gene Sequences”. In: *bioRxiv*, pp. 2024–02.

Liu, Zhuang et al. (2022b). “A convnet for the 2020s”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986.

Loshchilov, Ilya and Frank Hutter (2016). “Sgdr: Stochastic gradient descent with warm restarts”. In: *arXiv preprint arXiv:1608.03983*.

Lundberg, Scott M and Su-In Lee (2017). “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30.

Madani, Ali et al. (2020). “Progen: Language modeling for protein generation”. In: *arXiv preprint arXiv:2004.03497*.

Madani, Ali et al. (2023). “Large language models generate functional protein sequences across diverse families”. In: *Nature Biotechnology*, pp. 1–8.

Madry, Aleksander et al. (2017). “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: *arXiv 1706.06083*.

Majdandzic, Antonio et al. (2023). “Correcting gradient-based interpretations of deep neural networks for genomics”. In: *Genome Biology* 24.1.

Mallet, Vincent and Jean-Philippe Vert (2021). “Reverse-complement equivariant networks for DNA sequences”. In: *Advances in neural information processing systems* 34, pp. 13511–13523.

Marin, Frederikke Isa et al. (2023). “BEND: Benchmarking DNA Language Models on biologically meaningful tasks”. In: *arXiv 2311.12570*.

Maslova, Alexandra et al. (Sept. 2020). “Deep learning of immune cell differentiation”. In: *Proceedings of the National Academy of Sciences* 117.41, pp. 25655–25666.

Mathelier, Anthony and Wyeth W Wasserman (2013). “The next generation of transcription factor binding site prediction”. In: *PLoS computational biology* 9.9, e1003214.

Mauduit, David et al. (2021). “Analysis of long and short enhancers in melanoma cell states”. In: *Elife* 10, e71735.

McInnes, Leland, John Healy, and James Melville (2018). “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”. In: *arXiv 1802.03426*.

Meier, Joshua et al. (2021). “Language models enable zero-shot prediction of the effects of mutations on protein function”. In: *Advances in Neural Information Processing Systems* 34, pp. 29287–29303.

Mendoza-Revilla, Javier et al. (2023). “A Foundational Large Language Model for Edible Plant Genomes”. In: *bioRxiv*, pp. 2023–10.

Miesfeld, Joel B et al. (2020). “The Atoh7 remote enhancer provides transcriptional robustness during retinal ganglion cell development”. In: *Proceedings of the National Academy of Sciences* 117.35, pp. 21690–21700.

Minnoye, Liesbeth et al. (2020). “Cross-species analysis of enhancer logic using deep learning”. In: *Genome Research* 30.12, pp. 1815–1834.

Monahan, Kevin et al. (2017). “Cooperative interactions enable singular olfactory receptor expression in mouse olfactory neurons”. In: *Elife* 6, e28620.

Nair, Surag et al. (2022). “fastISM: performant in silico saturation mutagenesis for convolutional neural networks”. In: *Bioinformatics* 38.9, pp. 2397–2403.

Nair, Surag et al. (2023). “Transcription factor stoichiometry, motif affinity and syntax regulate single-cell chromatin dynamics during fibroblast reprogramming to pluripotency”. In: *bioRxiv*.

Nguyen, Eric et al. (2023). “Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution”. In: *arXiv preprint arXiv:2306.15794*.

Nguyen, Eric et al. (2024a). “Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution”. In: *Advances in neural information processing systems* 36.

Nguyen, Eric et al. (2024b). “Sequence modeling and design from molecular to genome scale with Evo”. In: *bioRxiv*.

Niu, Deng-Ke and Li Jiang (2013). “Can ENCODE tell us how much junk DNA we carry in our genome?” In: *Biochemical and biophysical research communications* 430.4, pp. 1340–1343.

Novakovsky, Gherman et al. (2021). “Biologically relevant transfer learning improves transcription factor binding prediction”. In: *Genome Biology* 22.1, pp. 1–25.

Novakovsky, Gherman et al. (2023). “ExplaiNN: interpretable and transparent neural networks for genomics”. In: *Genome Biology* 24.1, p. 154.

Ogden, Pierce J et al. (2019). “Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design”. In: *Science* 366.6469, pp. 1139–1143.

Omidi, Saeed et al. (2017). “Automated incorporation of pairwise dependency in transcription factor binding site prediction using dinucleotide weight tensors”. In: *PLoS computational biology* 13.7, e1005176.

Ong, Chin-Tong and Victor G Corces (2011). “Enhancer function: new insights into the regulation of tissue-specific gene expression”. In: *Nature Reviews Genetics* 12.4, pp. 283–293.

OpenAI (2023). “GPT-4 Technical Report”. In: *arXiv 2303.08774*.

Outeiral, Carlos and Charlotte M Deane (2024). “Codon language embeddings provide strong signals for use in protein engineering”. In: *Nature Machine Intelligence* 6.2, pp. 170–179.

Panigrahi, Anil and Bert W O’Malley (2021). “Mechanisms of enhancer action: the known and the unknown”. In: *Genome biology* 22.1, p. 108.

Park, Christopher Y et al. (2021). “Genome-wide landscape of RNA-binding protein target site dysregulation reveals a major impact on psychiatric disorder risk”. In: *Nature Genetics* 53.2, pp. 166–173.

Paszke, Adam et al. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*, pp. 8024–8035.

Penić, Rafael Josip et al. (2024). “RiNALMo: General-Purpose RNA Language Models Can Generalize Well on Structure Prediction Tasks”. In: *arXiv 2403.00043*.

Pennington, Jeffrey, Samuel Schoenholz, and Surya Ganguli (2017). “Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice”. In: *Advances in neural information processing systems* 30.

Poli, Michael et al. (2023). “Hyena hierarchy: Towards larger convolutional language models”. In: *arXiv 2302.10866*.

Pudimat, Rainer, Ernst-Günter Schukat-Talamazzini, and Rolf Backofen (2005). “A multiple-feature framework for modelling and predicting transcription factor binding sites”. In: *Bioinformatics* 21.14, pp. 3082–3088.

Quang, Daniel and Xiaohui Xie (2016). “DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences”. In: *Nucleic acids research* 44.11, e107–e107.

— (2019). “FactorNet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data”. In: *Methods* 166, pp. 40–47.

Radford, Alec et al. (2019). “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8, p. 9.

Raghu, Maithra et al. (2017). “On the expressive power of deep neural networks”. In: *international conference on machine learning*. PMLR, pp. 2847–2854.

Ren, Bing et al. (2000). “Genome-wide location and function of DNA binding proteins”. In: *Science* 290.5500, pp. 2306–2309.

Rives, A et al. (2021). “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences.” In: *Proceedings of the National Academy of Sciences* 118.15.

Robbins, Herbert and Sutton Monro (1951). “A stochastic approximation method”. In: *The annals of mathematical statistics*, pp. 400–407.

Robson, Eyes S and Nilah M Ioannidis (2023). “GUANinE v1. 0: Benchmark Datasets for Genomic AI Sequence-to-Function Models”. In: *bioRxiv*, pp. 2023–10.

Sanabria, Melissa, Jonas Hirsch, and Anna R Poetsch (2023a). “Distinguishing word identity and sequence context in DNA language models”. In: *bioRxiv*, pp. 2023–07.

— (2023b). “The human genome’s vocabulary as proposed by the DNA language model GROVER”. In: *bioRxiv*, pp. 2023–07.

Sasse, Alexander et al. (2023). “Benchmarking of deep neural networks for predicting personal gene expression from DNA sequence highlights shortcomings”. In: *Nature Genetics* 55.12, pp. 2060–2064.

Schiff, Yair et al. (2024). “Caduceus: Bi-directional equivariant long-range dna sequence modeling”. In: *arXiv 2403.03234*.

Schneider, Valerie A et al. (2017). “Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly”. In: *Genome Research* 27.5, pp. 849–864.

Schreiber, Jacob et al. (2022). “Accelerating in silico saturation mutagenesis using compressed sensing”. In: *Bioinformatics* 38.14, pp. 3557–3564.

Schwessinger, Ron et al. (2020). “DeepC: predicting 3D genome folding using megabase-scale transfer learning”. In: *Nature methods* 17.11, pp. 1118–1124.

Segal, Eran et al. (2006). “A genomic code for nucleosome positioning”. In: *Nature* 442.7104, pp. 772–778.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2015). “Neural machine translation of rare words with subword units”. In: *arXiv preprint arXiv:1508.07909*.

Shaham, Uri, Alexander Cloninger, and Ronald R Coifman (2018). “Provable approximation properties for deep neural networks”. In: *Applied and Computational Harmonic Analysis* 44.3, pp. 537–557.

Shao, Bin (2023). “A long-context language model for deciphering and generating bacteriophage genomes”. In: *bioRxiv*, pp. 2023–12.

Shen, Xilin and Xiangchun Li (2024). “OmniNA: A foundation model for nucleotide sequences”. In: *bioRxiv*, pp. 2024–01.

Shigaki, Dustin et al. (2019a). “Integration of multiple epigenomic marks improves prediction of variant impact in saturation mutagenesis reporter assay”. In: *Human Mutation* 40.9, pp. 1280–1291.

Shigaki, Dustin et al. (2019b). “Integration of multiple epigenomic marks improves prediction of variant impact in saturation mutagenesis reporter assay”. In: *Human mutation* 40.9, pp. 1280–1291.

Shlyueva, Daria, Gerald Stampfel, and Alexander Stark (2014). “Transcriptional enhancers: from properties to genome-wide predictions”. In: *Nature Reviews Genetics* 15.4, pp. 272–286.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2017). “Learning important features through propagating activation differences”. In: *International Conference on Machine Learning*. PMLR, pp. 3145–3153.

Shrikumar, Avanti et al. (2018). “Technical note on transcription factor motif discovery from importance scores (TF-MoDISco) version 0.5. 6.5”. In: *arXiv 1811.00416*.

Siddharthan, Rahul (2010). “Dinucleotide weight matrices for predicting transcription factor binding sites: generalizing the position weight matrix”. In: *PloS one* 5.3, e9722.

Siebert, Matthias and Johannes Söding (2016). “Bayesian Markov models consistently outperform PWMs at predicting motifs in nucleotide sequences”. In: *Nucleic acids research* 44.13, pp. 6055–6069.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). “Deep inside convolutional networks: Visualising image classification models and saliency maps”. In: *arXiv 1312.6034*.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2014). "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *Workshop at International Conference on Learning Representations*.

Slattery, Matthew et al. (2011). "Cofactor binding evokes latent differences in DNA binding specificity between Hox proteins". In: *Cell* 147.6, pp. 1270–1282.

Smilkov, Daniel et al. (2017). "Smoothgrad: removing noise by adding noise". In: *arXiv 1706.03825*.

Snyder, Eric E and Gary D Stormo (1993). "Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks". In: *Nucleic acids research* 21.3, pp. 607–613.

Song, Lingyun and Gregory E Crawford (2010). "DNase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells". In: *Cold Spring Harbor Protocols* 2010.2, pdb–prot5384.

Srivastava, Nitish (2013). "Improving neural networks with dropout". In: *University of Toronto* 182.566, p. 7.

Stormo, Gary D (2000). "DNA binding sites: representation and discovery". In: *Bioinformatics* 16.1, pp. 16–23.

Stormo, Gary D et al. (1982). "Use of the 'Perceptron'algorithm to distinguish translational initiation sites in E. coli". In: *Nucleic Acids Research* 10.9, pp. 2997–3011.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning*, pp. 3319–3328.

Tan, Jimin et al. (2023). “Cell-type-specific prediction of 3D chromatin organization enables high-throughput in silico genetic screening”. In: *Nature biotechnology* 41.8, pp. 1140–1150.

Tang, Ziqi and Peter K Koo (2024). “Evaluating the representational power of pre-trained DNA language models for regulatory genomics”. In: *bioRxiv*, pp. 2024–02.

Tang, Ziqi, Shushan Toneyan, and Peter K Koo (2023). “Current approaches to genomic deep learning struggle to fully capture human genetic variation”. In: *Nature Genetics* 55.12, pp. 2021–2022.

Tareen, Ammar and Justin B Kinney (2020). “Logomaker: beautiful sequence logos in Python”. In: *Bioinformatics* 36.7, pp. 2272–2274.

Tomovic, Andrija and Edward J Oakeley (2007). “Position dependencies in transcription factor binding sites”. In: *Bioinformatics* 23.8, pp. 933–941.

Toneyan, Shushan, Ziqi Tang, and Peter K Koo (2022). “Evaluating deep learning for predicting epigenomic profiles”. In: *Nature machine intelligence* 4.12, pp. 1088–1100.

Touvron, Hugo et al. (2023). “Llama: Open and efficient foundation language models”. In: *arXiv preprint arXiv:2302.13971*.

Vaishnav, Eeshit Dhaval et al. (2022). “The evolution, evolvability and engineering of gene regulatory DNA”. In: *Nature*, pp. 1–9.

Van Nostrand, Eric L et al. (2016). “Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP)”. In: *Nature methods* 13.6, pp. 508–514.

Vaswani, Ashish et al. (2017). “Attention is all you need”. In: *Advances in Neural Information Processing Systems* 30.

Venkatesh, Ishwariya et al. (2021). “Co-occupancy identifies transcription factor co-operation for axon growth”. In: *Nature communications* 12.1, p. 2555.

Vilov, Sergey and Matthias Heinig (2024). “Investigating the performance of foundation models on human 3’UTR sequences”. In: *bioRxiv*.

Vlaming, Hanneke et al. (2022). “Screening thousands of transcribed coding and non-coding regions reveals sequence determinants of RNA polymerase II elongation potential”. In: *Nature Structural & Molecular Biology* 29.6.

Wei, Jason et al. (2022). “Emergent abilities of large language models”. In: *arXiv 2206.07682*.

Wilkinson, Mark D et al. (2016). “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific data* 3.1, pp. 1–9.

Wittkopp, Patricia J, Belinda K Haerum, and Andrew G Clark (2004). “Evolutionary changes in cis and trans gene regulation”. In: *Nature* 430.6995, pp. 85–88.

Wolf, Thomas et al. (2019). “Huggingface’s transformers: State-of-the-art natural language processing”. In: *arXiv 1910.03771*.

Wong, Aaron K et al. (2021). “Decoding disease: from genomes to networks to phenotypes”. In: *Nature Reviews Genetics* 22.12, pp. 774–790.

Wu, Lei, Mingze Wang, and Weijie Su (2022). “The alignment property of SGD noise and how it helps select flat minima: A stability analysis”. In: *Advances in Neural Information Processing Systems* 35, pp. 4680–4693.

Wu, Ruidong et al. (2022). “High-resolution de novo structure prediction from primary sequence”. In: *BioRxiv*, pp. 2022–07.

Yang, Meng et al. (2022). “Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution”. In: *Nucleic Acids Research* 50.14, e81–e81.

Yang, Rui et al. (2023). “Epiphany: predicting Hi-C contact maps from 1D epigenomic signals”. In: *Genome Biology* 24.1, p. 134.

Yin, Qijin et al. (2019). “DeepHistone: a deep learning approach to predicting histone modifications”. In: *BMC genomics* 20, pp. 11–23.

Yu, Fisher and Vladlen Koltun (2015). “Multi-scale context aggregation by dilated convolutions”. In: *arXiv 1511.07122*.

Yu, Fisher, Vladlen Koltun, and Thomas Funkhouser (2017). “Dilated residual networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 472–480.

Zaheer, Manzil et al. (2020). “Big bird: Transformers for longer sequences”. In: *Advances in Neural Information Processing Systems* 33, pp. 17283–17297.

Zeitlinger, Julia (2020). “Seven myths of how transcription factors read the cis-regulatory code”. In: *Current opinion in systems biology* 23, pp. 22–31.

Zeng, Haoyang et al. (2016). “Convolutional neural network architectures for predicting DNA–protein binding”. In: *Bioinformatics* 32.12, pp. i121–i127.

Zhan, Huixin, Ying Nian Wu, and Zijun Zhang (2024). “Efficient and Scalable Fine-Tune of Language Models for Genome Understanding”. In: *arXiv preprint arXiv:2402.08075*.

Zhang, Daoan et al. (2023). “DNAGPT: A Generalized Pretrained Tool for Multiple DNA Sequence Analysis Tasks”. In: *bioRxiv*, pp. 2023–07.

Zhang, Yao-zhong, Zeheng Bai, and Seiya Imoto (2023). “Investigation of the BERT model on nucleotide sequences with non-standard pre-training and evaluation of different k-mer embeddings”. In: *Bioinformatics* 39.10, btad617.

Zhang, Yongqing et al. (2020). “DeepSite: bidirectional LSTM and CNN models for predicting DNA–protein binding”. In: *International Journal of Machine Learning and Cybernetics* 11, pp. 841–851.

Zhang, Zhidian et al. (2024). “Protein language models learn evolutionary statistics of interacting sequence motifs”. In: *bioRxiv*, pp. 2024–01.

Zhao, Yue et al. (2012). “Improved models for transcription factor binding site identification using nonindependent interactions”. In: *Genetics* 191.3, pp. 781–790.

Zheng, An et al. (2021). “Deep neural networks identify sequence context features predictive of transcription factor binding”. In: *Nature Machine Intelligence* 3.2, pp. 172–180.

Zhou, Jian (2022). “Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale”. In: *Nature Genetics* 54.5, pp. 725–734.

Zhou, Jian and Olga G Troyanskaya (2015a). “Predicting effects of noncoding variants with deep learning–based sequence model”. In: *Nature methods* 12.10, pp. 931–934.

— (Aug. 2015b). “Predicting effects of noncoding variants with deep learning–based sequence model”. In: *Nature Methods* 12.10, pp. 931–934.

Zhou, Jian et al. (2018a). “Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk”. In: *Nature genetics* 50.8, pp. 1171–1179.

Zhou, Jian et al. (July 2018b). “Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk”. In: *Nature Genetics* 50.8, pp. 1171–1179.

Zhou, Jian et al. (May 2019). “Whole-genome deep-learning analysis identifies contribution of noncoding mutations to autism risk”. In: *Nature Genetics* 51.6, pp. 973–980.

Zhou, Zhihan et al. (2023). “Dnabert-2: Efficient foundation model and benchmark for multi-species genome”. In: *arXiv preprint arXiv:2306.15006*.

Zhu, Danqing et al. (2024). “Optimal trade-off control in machine learning–based library design, with application to adeno-associated virus (AAV) for gene therapy”. In: *Science Advances* 10.4, eadj3786.

Zhu, Fangjie et al. (2018). “The interaction landscape between transcription factors and the nucleosome”. In: *Nature* 562.7725, pp. 76–81.

Zou, James et al. (2019). “A primer on deep learning in genomics”. In: *Nature genetics* 51.1, pp. 12–18.

Zvyagin, Maxim et al. (2023). “GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics”. In: *The International Journal of High Performance Computing Applications* 37.6, pp. 683–705.