Chromatin interaction–aware gene regulatory modeling with graph attention networks

Alireza Karbalayghareh, Merve Sahin, and Christina S. Leslie

Computational and Systems Biology Program, Memorial Sloan Kettering Cancer Center, New York, New York 10065, USA

Linking distal enhancers to genes and modeling their impact on target gene expression are longstanding unresolved problems in regulatory genomics and critical for interpreting noncoding genetic variation. Here, we present a new deep learning approach called GraphReg that exploits 3D interactions from chromosome conformation capture assays to predict gene expression from 1D epigenomic data or genomic DNA sequence. By using graph attention networks to exploit the connectivity of distal elements up to 2 Mb away in the genome, GraphReg more faithfully models gene regulation and more accurately predicts gene expression levels than the state-of-the-art deep learning methods for this task. Feature attribution used with GraphReg accurately identifies functional enhancers of genes, as validated by CRISPRi-FlowFISH and TAP-seq assays, outperforming both convolutional neural networks (CNNs) and the recently proposed activity-by-contact model. Sequence-based GraphReg also accurately predicts direct transcription factor (TF) targets as validated by CRISPRi TF knockout experiments via in silico ablation of TF binding motifs. GraphReg therefore represents an important advance in modeling the regulatory impact of epigenomic and sequence elements.

[Supplemental material is available for this article.]

Transcriptional gene regulation involves the binding of transcription factors (TFs) at both promoter and enhancer regions and the physical interaction of these bound complexes via DNA looping. Technological advances in chromosome conformation capture assays such as Hi-C (Lieberman-Aiden et al. 2009), HiChIP (Mumbach et al. 2016), Micro-C (Krietenstein et al. 2020), and HiCAR (Wei et al. 2022) provide high-resolution data on 3D chromatin interactions, including regulatory interactions between enhancers and promoters. Three-dimensional interaction data sets combined with traditional 1D epigenomic profiles, including chromatin accessibility (DNase-seq and ATAC-seq) (Buenrostro et al. 2015) and histone modifications (via ChIP-seq and CUT&RUN) (Skene and Henikoff 2017), together map the chromatin state and connectivity of regulatory elements and should provide rich training data for predictive gene regulatory models, including models that also incorporate the underlying genomic DNA sequence. Ultimately, such models could be used to infer the regulatory function of noncoding genetic variants in a cell type of interest.

Here, we propose a gene regulatory modeling approach called GraphReg that integrates 1D epigenomic data (Epi-GraphReg) or both epigenomic and DNA sequence data (Seq-GraphReg) with 3D chromatin interaction data from HiChIP, Hi-C, Micro-C, or HiCAR via a graph neural network to predict gene expression. The 1D input data can include any standard epigenomic assays such as histone modification ChIP-seq, transcription factor ChIP-seq, or chromatin accessibility from DNase-seq or ATAC-seq.

GraphReg models use convolutional neural network (CNN) layers to learn local representations from 1D inputs, followed by graph attention network (GAT) layers to propagate these representations over the 3D interaction graph, to predict gene expression (CAGE-seq) across genomic positions (bins). GraphReg is trained to predict CAGE-seq (Shiraki et al. 2003), a tag-based protocol for gene expression measurement and transcription start site (TSS) mapping, because it quantifies promoter output and does not depend on transcript length.

Our motivation for proposing GraphReg is twofold. First, we aim to improve the accuracy of predictive gene regulatory models by leveraging 3D genomic architecture to incorporate distal enhancer elements, and we show that GraphReg outperforms baseline CNN models for prediction of CAGE output. Second, we seek to interpret the model to assess the functional importance of distal enhancers for regulation of specific target genes. To this end, we use feature attribution methods on trained GraphReg models and show that we obtain a better ranking of functional enhancers as validated by CRISPRi-FlowFISH (Fulco et al. 2019) and TAP-seq (Schraivogel et al. 2020) than baseline CNN models and the recently proposed activity-by-contact (ABC) model (Fulco et al. 2019). Finally, we show that Seq-GraphReg models capture meaningful TF binding signals through in silico motif ablation experiments, which accurately predict direct TF targets as validated by CRISPRi-based TF perturbation experiments.

Results

GraphReg: a deep graph neural network model for interaction-aware gene expression prediction

In our experiments, we used a minimal set of 1D epigenomic data relevant to gene regulation: DNase-seq as a measure of chromatin accessibility, H3K4me3 ChIP-seq for promoter activity, and H3K27ac ChIP-seq for enhancer activity. For 3D interaction data, we have used a variety of chromatin conformation assays, such as Hi-C, H3K27ac HiChIP, Micro-C, and HiCAR. Because Epi-GraphReg is trained on 1D epigenomic and 3D interaction input data in a given cell type, to predict gene expression output in the

Corresponding author: cleslie@cbio.mskcc.org

Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.275870.121. Freely available online through the *Genome Research* Open Access option.

32:930–944 Published by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/22; www.genome.org

same cell type, it learns a regulatory model that can generalize to other contexts. That is, given cell type–specific input data in a new cell type, the trained Epi-GraphReg model can predict gene expression in that cell type. In this sense, the Epi-GraphReg model is cell type–agnostic. Seq-GraphReg uses DNA sequence as the input and performs multitask learning, predicting DNase, H3K4me3, and H3K27ac in the CNN block and predicting CAGE-seq after the GAT block. Seq-GraphReg is therefore a cell type–restricted model because it learns the TF binding motifs specific to the training cell type and consequently cannot generalize to another cell type. However, Seq-GraphReg can capture cell type–specific TF binding motifs in enhancer and promoter regions and potentially predict the impact of DNA sequence alterations on target gene expression.

We bin the epigenomic data (H3K4me3, K3K27ac, and DNase) at 100-bp resolution. We process the 5-kb resolution Hi-C/HiChIP/Micro-C/HiCAR contact matrices with the HiC-DC+ package (Carty et al. 2017; Sahin et al. 2021) to identify significant interactions of genomic distance up to 2 Mb, and we use the 3D interactions satisfying three different false discovery rates (FDR) of 0.1, 0.01, and 0.001 to define a graph on 5-kb genomic bins. We process 3D interaction data up to 2 Mb for two main reasons. First, the read coverage of 3D assays become sparser after 2 Mb, making it difficult to robustly find significant interactions. Second, the majority of known functional enhancers of genes reside within 2 Mb of the TSS. For consistency with the 3D interaction data, we also bin CAGE-seq at 5-kb resolution. In each batch of training, we extract 6-Mb genomic regions and use the corresponding DNA sequences, epigenomic data, interaction graphs from 3D data, and the corresponding CAGE-seq as training examples for the model. For the next batch, we shift the entire 6-Mb region by 2 Mb and repeat this to cover the training chromosomes.

For Epi-GraphReg (Fig. 1A; Supplemental Fig. S1), each epigenomic signal track is treated as a different channel and fed to 1D CNN layers followed by max-pooling and ReLU activation functions. This produces latent local representations at 5-kb resolution, which serve as the node features for the interaction graphs derived from 3D data. We then use several graph attention network (GAT) layers to propagate these local representations over the interaction graphs, so that promoter regions are influenced by the representations of their candidate distal enhancers. Finally, we predict the CAGE values (promoter activities) by a fully connected (FC) layer.

For Seq-GraphReg (Fig. 1B; Supplemental Fig. S1), we feed one-hot-coded genomic DNA sequences to a series of 1D CNN, max-pooling, and ReLU layers, similar to Epi-GraphReg. The resulting bottleneck representations are then supplied to two blocks corresponding to different prediction tasks: a GAT block, similar to the Epi-GraphReg model, uses several GAT layers followed by a FC layer to predict the CAGE values; and a dilated CNN block, containing several dilated CNN layers whose dilated rate is multiplied by two each layer, to predict the 1D epigenomic data. Dilated CNNs have been used previously to increase the receptive field of CNN layers in deep learning models that predict epigenomic or expression data from DNA sequence, such as Basenji (Kelley et al. 2018), BPNet (Avsec et al. 2021b), ExPecto (Zhou et al. 2018), and Xpresso (Agarwal and Shendure 2020). Therefore, Seq-GraphReg follows a multitask learning approach to find more meaningful bottleneck representations to provide to the GAT block.

GATs have an advantage over other graph neural networks (GNN) in that they learn to weight edges in the graph from node features. In this way, GATs weigh enhancer–enhancer (E–E) and enhancer–promoter (E–P) interactions, based on the features learned in the promoter and enhancer bins, to predict CAGE values more accurately. However, non-attention-based GNNs such as graph convolutional networks (GCN) (Kipf and Welling 2017; Bigness et al. 2022) fail to learn the importance of individual interactions. It has been shown that GATs outperform GCNs in other machine learning contexts as well (Veličković et al. 2018). We use different attention heads (Fig. 1A,B; Methods) in each GAT layer to enhance the flexibility of the model to learn distinct E–E and E–P interaction weights and to improve the prediction accuracy by integrating them together. Figure 1C shows an example of a 6-Mb region (11 Mb–17 Mb) of Chromosome 19 in K562 cells with the corresponding true CAGE output, 1D epigenomic inputs, H3K27ac HiChIP interaction graph (FDR = 0.1), and the predicted CAGE signals using Epi-GraphReg and Seq-GraphReg and their CNN counterparts, which we denote as Epi-CNN and Seq-CNN, respectively. The dashed middle 2-Mb region in Figure 1C indicates where we compute the loss in training and evaluate CAGE predictions. The inputs and outputs of Epi-CNN and Seq-CNN models are the same as those of Epi-GraphReg and Seq-GraphReg, respectively, with the exception that they use dilated CNN layers instead of GAT layers and do not use any 3D interaction data (Supplemental Fig. S1). We can train Seq-GraphReg in an end-to-end or separate manner. In separate training, instead of multitask learning, we first predict epigenomic data from DNA sequence and then feed the bottleneck representation to the GAT layers to predict CAGE expression levels. Because CNN models to predict epigenomic signals can use smaller window sizes (100 kb instead of 6 Mb), more filters can be used given the same GPU memory resources, yielding a better bottleneck representation that consequently leads to improvement in CAGE prediction. However, this prediction improvement comes at the cost of losing access to fast backpropagation-based saliency scores at base-pair resolution (Methods).

To assess the performance of GraphReg on different graphs, we ran it using four different 3D assays—Hi-C, HiChIP, Micro-C, and HiCAR—with three different FDR cutoffs (from HiC-DC+) 0.1, 0.01, and 0.001. Supplemental Figure S2 shows the degree distributions of the genes using different graphs in four cell lines: K562, GM12878, hESC, and mESC. Some example graphs corresponding to a subset of Chromosome 1 for these cell types are shown in Supplemental Figures S3 and S4, showing how various 3D assays and FDR cutoffs yield different graph topologies.

GraphReg accurately predicts gene expression by using 3D interaction data

We trained GraphReg models for three ENCODE human cell lines (GM12878, K562, hESC) and for mouse embryonic stem cells (mESC), for which complete 1D epigenomic data and 3D interaction data are available. To evaluate, we performed cross-validation experiments in which we held out two chromosomes for testing, used two chromosomes for validation, and trained on all remaining autosomal chromosomes (Methods). Although our model predicts the CAGE signal at all genomic bins, we focused first on predictions for GENCODE-annotated TSS bins of protein coding genes, where the CAGE signal can be non-zero. Figure 2, A and B, and Supplemental Figures S5–S11 show the CAGE prediction results for GM12878, K562, hESC, and mESC in which predictions for test chromosomes across runs are concatenated together to obtain global performance results (20 test chromosomes over 10 runs). We computed the negative log-likelihood values (*NLL*),

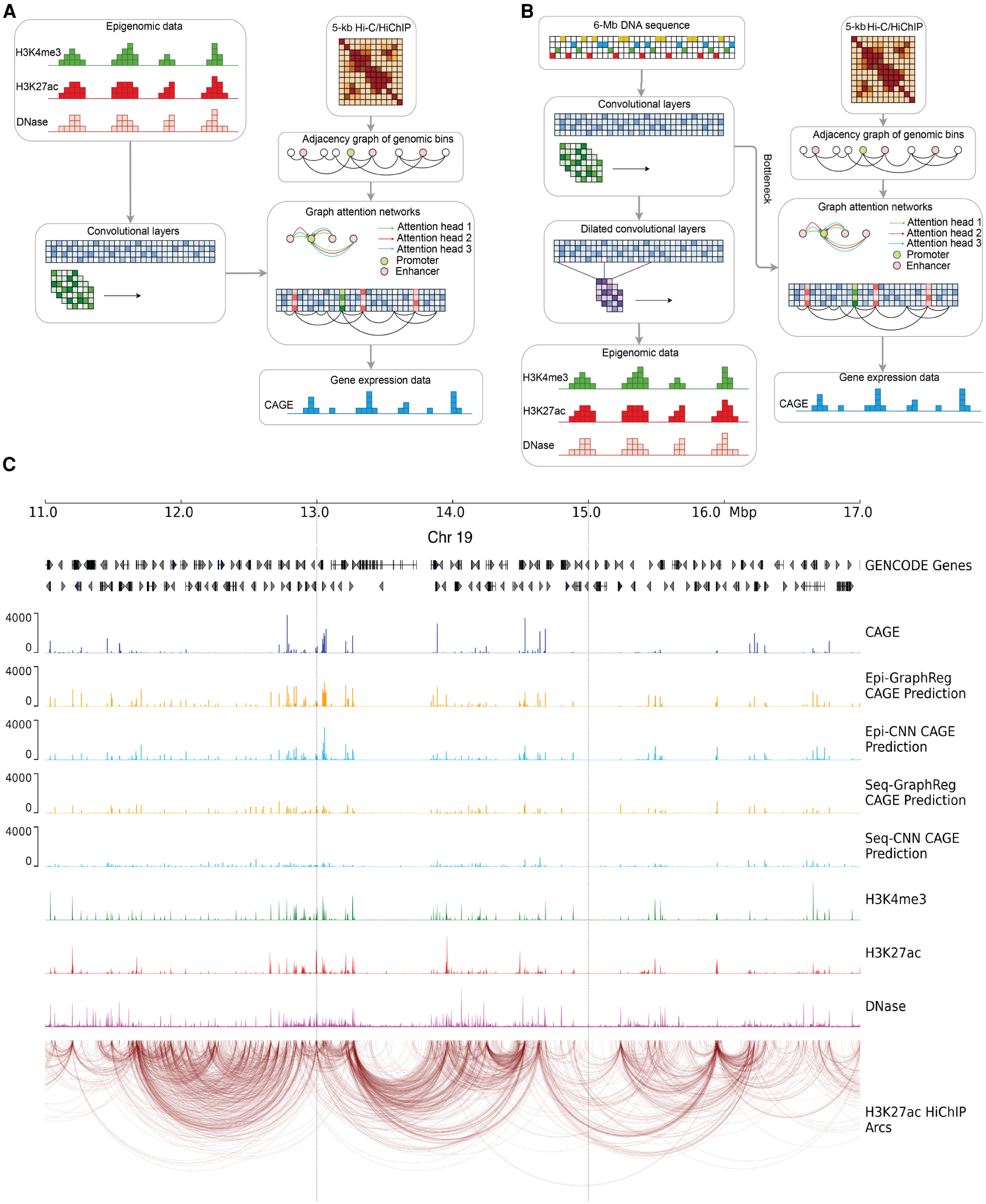

Figure 1. A schematic overview of GraphReg models. (*A*) The Epi-GraphReg model uses 1D epigenomic data, such as H3K4me3 and H3K27ac ChIP-seq and DNase-seq (or ATAC-seq) to learn local features of genomic bins via convolutional neural networks, and then propagates these features over adjacency graphs extracted from Hi-C/HiChIP contact matrices using graph attention networks to predict gene expression (CAGE-seq) across genomic bins. (*B*) The Seq-GraphReg model uses DNA sequence as input and, after some convolutional and dilated convolutional layers, predicts epigenomic data. This helps to learn useful latent representations of genomic DNA sequences that are then passed to the graph attention networks to be integrated over the adjacency graphs derived from Hi-C/HiChIP contact matrices and to predict gene expression values (CAGE-seq). (*C*) A 6-Mb genomic region (11 Mb–17 Mb) of Chr 19 showing input and output signals and predictions in K562 cells, including epigenomic data (H3K4me3, H3K27ac, DNase), CAGE, HiChIP interaction graph, and predicted CAGE values for GraphReg and CNN models. Training and evaluations of the models are performed in the dashed middle 2 Mb (here 13 Mb–15 Mb) region so that all genes can see the effects of their distal enhancers up to 2 Mb.

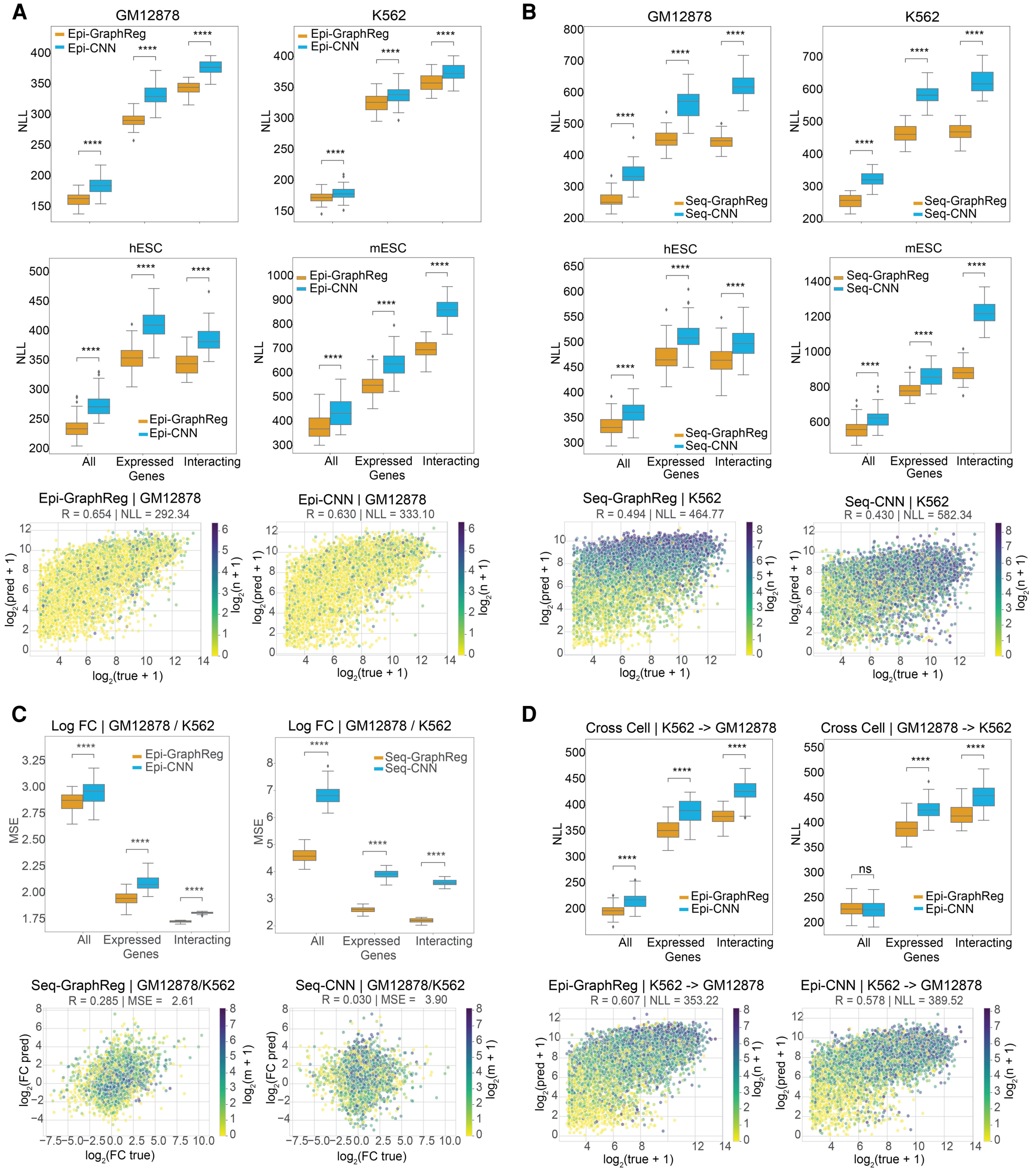

Figure 2. GraphReg models outperform their CNN counterparts for gene expression prediction. (*A*,*B*) Negative log-likelihood (*NLL*, lower is better) between true and predicted CAGE signals of epigenome-based (*A*) and sequence-based (*B*) GraphReg and CNN models over 50 random selections of 2000 predicted genes from test chromosomes concatenated from 10 cross-validation experiments with different training, test, and validation chromosomes. Box plots show the distributions of *NLL* in GM12878, K562, hESC, and mESC for three gene sets: all genes, expressed genes (CAGE signal ≥ 5), and expressed genes with at least one 3D interaction (interacting). The 3D data used in Epi-GraphReg (*A*) for each cell type is as follows: Hi-C (FDR = 0.001) for GM12878, H3K27ac HiChIP (FDR = 0.01) for K562, Micro-C (FDR = 0.1) for hESC, and H3K27ac HiChIP (FDR = 0.1) for mESC. The 3D data used for Seq-GraphReg (*B*) is H3K27ac HiChIP (FDR = 0.1) for GM12878, K562, and mESC, and Micro-C (FDR = 0.1) for hESC. Example scatterplots of all predicted test genes that are expressed (CAGE ≥ 5) are plotted for GM12878 in epigenome-based models (*A*) and K562 in sequence-based models (*B*), where the genes are color-coded by the number of 3D interactions *n*. The sequenced-based models have been trained separately (and not using dilated CNN) for K562 and end-to-end for GM12878, hESC, and mESC. (*C*) Box plots show mean squared error (*MSE*) of the true and predicted log-fold gene expression changes between GM12878 and K562 in 50 random selections of 2000 predicted genes from test chromosomes concatenated from 10 cross-validation experiments with different training, test, and validation chromosomes. The sets All, Expressed, and Interacting denote the intersections of such sets in GM12878 and K562. Both Epi- and Seq-GraphReg models have better prediction accuracy than their CNN counterparts. The scatterplots of the true log-fold gene expression changes and the log-fold changes derived from the predicted CAGE values by Seq-GraphReg and Seq-CNN, between GM12878 and K562, are shown for expressed genes (CAGE ≥ 5 in both K562 and GM12878). TSS bins are color-coded by the minimum number of 3D interactions in GM12878 and K562 (*m*). Seq-GraphReg has higher *R* and lower *MSE* than Seq-CNN. (*D*) Epi-GraphReg models show higher cell-to-cell generalization capability than Epi-CNN models. Box plots show the distributions of *NLL* on the test cell type (K562 or GM12878) when trained on the other cell type over 50 random selections of 2000 predicted genes from test chromosomes of the test cell concatenated from 10 cross-validation experiments with different training, test, and validation chromosomes in the training cell. The models are evaluated on the same test chromosomes in the unseen test cell. HiChIP (FDR = 0.1) is used for both cells. The generalization of Epi-GraphReg from K562 to GM12878 is significantly better ($P < 10^{-4}$, Wilcoxon signed-rank test) than Epi-CNN in all gene sets. The generalization of Epi-GraphReg from GM12878 to K562 is significantly better ($P < 10^{-4}$, Wilcoxon signed-rank test) than Epi-CNN in expressed and interacting genes. The scatterplots of all predicted test genes that are expressed (CAGE ≥ 5) are plotted when trained on K562 and tested on GM12878.

based on our loss function for the Poisson distribution, of the predicted CAGE signals for three gene sets: all genes (All); expressed genes, defined as CAGE signal ≥ 5 (Expressed); and expressed genes with at least one E–P interaction (Interacting). We also reported Pearson's correlation (R) of log-normalized predicted and true CAGE values (Fig. 2A,B; Supplemental Figs. S5–S11). We also compared Seq-GraphReg and Seq-CNN with Basenji on K562 and GM12878 (Supplemental Fig. S12). In all cases, GraphReg models outperform the corresponding CNN models, achieving higher R and lower NLL (Fig. 2A,B; Supplemental Figs. S5–S11). After restricting to expressed or interacting genes, the problem gets harder, and the prediction improvement of GraphReg over CNN models increases. Epigenome-based models (Epi-GraphReg and Epi-CNN) have higher prediction accuracy than the sequence-based models (Seq-GraphReg and Seq-CNN) (Fig. 2A,B; Supplemental Figs. S5–S11) because the epigenomic data are highly correlated with the CAGE output. To plot the box plots in Figure 2 and assess statistical significance, we randomly sampled 2000 predicted genes (in each category, All, Expressed, and Interacting) 50 times, computed the mean of 2000 NLLs for both GraphReg and CNN, and performed a Wilcoxon signed-rank test to see if the NLLs are significantly smaller for GraphReg than CNN. For both epigenome-based (Fig. 2A) and sequence-based (Fig. 2B) models, GraphReg has significantly smaller NLL than CNN (Wilcoxon signed-rank test), demonstrating more accurate predictions for GraphReg.

We also examined scatterplots of log-normalized [$\log_2(x+1)$] predicted versus true CAGE values of GraphReg and CNN models across the expressed genes in mESC, hESC, K562, and GM12878 (Fig. 2A,B; Supplemental Figs. S5–S11). Here, TSS bins are color-coded by their number of 3D interactions (n). An example of Epi-GraphReg and Epi-CNN on GM12878 using Hi-C (FDR = 0.001) is shown in Figure 2A, and an example of Seq-GraphReg and Seq-CNN on K562 using HiChIP (FDR = 0.1) with separate training and no dilated layers is shown in Figure 2B. For epigenome-based models in Figure 2A, Epi-GraphReg yields $R = 0.654$ and $NLL = 292.34$ on expressed genes, outperforming Epi-CNN with $R = 0.63$ and $NLL = 333.10$. For sequence-based models in Figure 2B, Seq-GraphReg yields $R = 0.494$ and $NLL = 464.77$ on expressed genes, outperforming Seq-CNN with $R = 0.43$ and $NLL = 582.34$. More scatterplots for different GraphReg models can be found in Supplemental Figures S6–S11. Analogous to our previous notion of gene regulatory complexity (González et al. 2015), we refer to genes with $n = 0$ as "simple genes" and those with $n > 0$ (at least one 3D interaction) as "complex genes." We hypothesized that by exploiting 3D interaction data, GraphReg would better model the regulation of complex genes. This fact can be easily seen in scatterplots shown in Figure 2B.

Notably, the extent of improvement of Seq-GraphReg over Seq-CNN is usually higher than that of the Epi-GraphReg over Epi-CNN (Fig. 2A,B; Supplemental Figs. S5–S11), suggesting the importance of 3D interaction data for integrating distal regulatory elements and improving prediction accuracy when predicting from DNA sequence, especially for more complex genes with many E–P interactions (Supplemental Figs. S10, S11). In particular, Seq-CNN models predict lower values for some highly expressed genes with many 3D interactions, whereas Seq-GraphReg can more accurately predict these higher expression values (Fig. 2B; Supplemental Figs. S10, S11). The effects of 3D assay (Hi-C, HiChIP, Micro-C, HiCAR) and different FDR levels (0.1, 0.01, 0.001, corresponding to different noise levels in the graphs) for Epi-GraphReg are shown in Supplemental Figures S5–S9.

GraphReg's improved performance over CNNs can also be witnessed by calculating the log-fold change between CAGE predictions for two different cell types (Fig. 2C) and by comparing to true log-fold changes using mean squared error (MSE). Evaluation of predicted log-fold changes in GM12878 versus K562 on held-out chromosomes in 10 cross-validation runs (Fig. 2C) confirmed the improvement of Epi-GraphReg and Seq-GraphReg over Epi-CNN and Seq-CNN, respectively. We see in Figure 2C that MSEs of GraphReg are significantly smaller than CNN (Wilcoxon signed-rank test), and the performance improvements are higher for Seq-GraphReg versus Seq-CNN. Figure 2C shows the scatterplots of predicted versus true log-fold change in TSS bins on held-out chromosomes for expressed genes (CAGE ≥ 5 in both GM12878 and K562), color-coded by the minimum number of 3D interactions over the two cell types (m). For Seq-GraphReg, HiChIP (FDR = 0.1) is used in both GM12878 and K562. For Epi-GraphReg, Hi-C (FDR = 0.001) and HiChIP (FDR = 0.01) are used in GM12878 and K562, respectively. As shown in scatterplots of Figure 2C, Seq-GraphReg achieves $R = 0.285$ and $MSE = 2.61$ in expressed genes, compared to very poor $R = 0.03$ and $MSE = 3.9$ for Seq-CNN. Scatterplots of true versus predicted log-fold change by Epi-GraphReg and Epi-CNN are shown in Supplemental Figure S13, with $R = 0.561$ and $MSE = 1.94$ for Epi-GraphReg and $R = 0.535$ and $MSE = 2.08$ for Epi-CNN. Overall, these results confirm that using 3D chromatin interactions in GraphReg leads to improved gene expression prediction.

We investigated how well the cell type–agnostic Epi-GraphReg and Epi-CNN models can generalize from one cell type to another. Figure 2D shows the box plots of loss (NLL) when we train on one cell type (K562 or GM12878) and test on the other, using 10 cross-validation runs with distinct sets of validation chromosomes (held out from training and used to assess performance in the test cell type). Epi-GraphReg significantly outperforms Epi-CNN ($P < 10^{-4}$, Wilcoxon signed-rank test) for generalization from K562 to GM12878 and vice versa in all categories except one: all genes from GM12878 to K562. Example scatterplots for these cross-cell-type and cross-chromosome generalization tasks show the improvement of Epi-GraphReg over Epi-CNN (Fig. 2D): when training on K562 with HiChIP (FDR = 0.1) and testing on GM12878 with HiChIP (FDR = 0.1), Epi-GraphReg attains $R = 0.607$ and $NLL = 353.22$, whereas Epi-CNN yields $R = 0.578$ and $NLL = 389.52$ for expressed genes. The effect of 3D assay choice and FDR values for cross-cell-type generalization is examined in Supplemental Figures S14–S16, where we see that the best and most robust generalization happens when we use H3K27ac HiChIP in both train and test cell types or HiChIP in train and Hi-C in test cell types, and the worst and least robust generalization performance occurs when we use Hi-C in train and HiChIP in test cell types. One reason for this phenomenon could be that HiChIP interactions, which are enriched for enhancer–promoter interactions, are more generalizable between cell types than Hi-C interactions, which depend strongly on the depth of coverage and library complexity and may be dominated by structural interactions in lower coverage data sets. Similarly to K562 → GM12878, when training on GM12878 with HiChIP (FDR = 0.001) and testing on K562 with HiChIP (FDR = 0.1), Epi-GraphReg attains $R = 0.563$ and $NLL = 389.71$ compared to $R = 0.585$ and $NLL = 425.91$ for Epi-CNN on expressed genes (Supplemental Fig. S16).

Normalization of epigenomic tracks is very critical in cross-cell-type experiments because sequencing depths can create undesired batch effects and prevent good generalization to the unseen cell types. For the results shown in Figure 2D and Supplemental

Figures S14–S16, we used DESeq-like normalization between K562 and GM12878 for each pair of epigenomic tracks. We also tried another normalization called reads per genomic coverage (RPGC) or 1× normalization from deepTools (Ramírez et al. 2016), and we evaluated the generalization performance across three cell lines: K562, GM12878, and hESC. Supplemental Figures S17–S22 show the generalization performance from cell type x to the cell type y ($x \rightarrow y$), where x and y ($x \neq y$) could be any of K562, GM12878, or hESC. We used Hi-C and HiChIP graphs in K562 and GM12878, and Micro-C and HiCAR graphs in hESC, all with the FDR of 0.1. Overall, the generalization of Epi-GraphReg is better than Epi-CNN in terms of having lower NLL and higher R. However, the generalization performance varies using different 3D assays in train and test cell types. In each $x \rightarrow y$ experiment, the scatterplots for one of the best performing pairs of 3D assays are shown in Supplemental Figures S17–S22.

We also wanted to determine the classes of genes for which GraphReg's predictions are better than those of CNN models and thus understand the genes for which 3D information gives the most benefit. To this end, we defined a simple metric called $Delta_{NLL} = NLL_{CNN} - NLL_{GraphReg}$. If $Delta_{NLL} > 0$ for a gene, it means that GraphReg's prediction is better than CNN for that gene, and if $Delta_{NLL} < 0$ for a gene, it means that CNN's prediction is better than GraphReg for that gene. We produced scatterplots of $Delta_{NLL}$ versus the number of interactions for all expressed genes in all four cell types—GM12878, K562, hESC, and mESC—and for both sequence-based (Seq-GraphReg and Seq-CNN) and epigenome-based (Epi-GraphReg and Epi-CNN) models (Supplemental Figs. S6–S11, S15–S22). The majority of genes have positive $Delta_{NLL}$, especially the ones with non-zero interactions. Furthermore, the number of genes with positive $Delta_{NLL}$ is higher in sequenced-based models, showing the greater advantage of using 3D information and GraphReg when predicting from DNA sequence. We have also listed in Supplemental Figures S23–S30 the top genes for which either GraphReg or CNN predicts better.

GraphReg accurately identifies functional enhancers of genes

Thanks to the feature attribution methods for machine learning models—for example, saliency maps (gradient-by-input), DeepLIFT (Shrikumar et al. 2017), DeepSHAP (Lundberg and Lee 2017), and Integrated Gradients (Sundararajan et al. 2017)—it is possible to derive the important input features for the prediction of a specific output. Because GraphReg allows each gene to be influenced by its potential enhancers through interactions in the 3D graph, we hypothesized that feature attribution analysis would allow the identification of functional distal enhancers, that is, those that contribute to the regulation of target genes. CRISPRi-FlowFISH (Fulco et al. 2019) is a recent enhancer screening approach that uses KRAB-dCas9 interference with candidate enhancer regions in pooled fashion and RNA FISH against a gene of interest as readout; this enables estimation of the effect size of perturbing each candidate enhancer on target gene expression. The developers of CRISPRi-FlowFISH introduced a score called activity-by-contact (ABC) for finding and ranking enhancers of a gene, which is considered the current state-of-the-art for this problem. The ABC score is the product of "activity," defined as the geometric mean of DNase and H3K27ac signal in each candidate enhancer region, and "contact," defined by the KR-normalized Hi-C signal between the target promoter and the candidate enhancer, normalized so that the ABC scores of all candidate enhancers up to 5 Mb from the gene sum to one.

To determine functional enhancers for genes, we used two feature attribution methods: the saliency map (gradient-by-input) and DeepSHAP (Lundberg and Lee 2017). For the epigenome-based models (Epi-GraphReg and Epi-CNN), as the inputs are binned at 100-bp resolution, both saliency and DeepSHAP scores are computed at 100-bp resolution as well. We used all-zero data as the reference signal for DeepSHAP so that it can evaluate the importance of peak regions in the input tracks. For the sequence-based models (Seq-GraphReg and Seq-CNN), the feature attribution methods can provide scores at nucleotide resolution. However, for enhancer scoring and validation with FlowFISH analysis, as suggested in Basenji (Kelley et al. 2018), we derived the saliency maps as the dot product of the 100-bp bin representations (bottleneck) and the gradient of the model prediction for the gene with respect to those bin representations. Figure 3A shows the distribution of area under the precision-recall curve (auPR) values for 19 genes from the K562 FlowFISH data set with more than 10 candidate enhancers for all models. The overall precision-recall curve across 2574 enhancer-gene (E-G) pairs for all 19 genes is shown in Figure 3B. HiChIP (FDR = 0.1) was used for GraphReg in these experiments. These results confirm that feature attribution applied to GraphReg models more accurately identifies the functional enhancers of the genes from a pool of candidate enhancers than the ABC score or feature attribution on the corresponding CNN models (Fig. 3A,B). The highest auPR among GraphReg models was 0.4236 (Epi-GraphReg with saliency), outperforming the best CNN model (auPR 0.366 for Epi-CNN with saliency) and ABC (auPR 0.364).

To assess the ability of models to find distal enhancers, we examined the example of *MYC* and considered only candidate enhancers more than 10 kb from TSS. There are 200 such candidate distal enhancer-gene (DE-G) pairs for *MYC*, including eight true functional enhancers. Precision-recall analysis for *MYC* (Supplemental Fig. S31A) found the highest auPR of GraphReg models to be 0.576 (Epi-GraphReg with DeepSHAP), strongly outperforming the best CNN model (auPR 0.0792, Epi-CNN with DeepSHAP) as well as ABC (auPR 0.3567). Visualization of the feature attribution scores gives insight into how 3D information allows GraphReg to access distal regulatory information that dilated CNNs cannot exploit, despite their large receptive field. Figure 3E shows a 2.5 Mb genomic region containing the *MYC* locus, with the corresponding epigenomic data, HiChIP graph edges, and true and predicted CAGE signals by GraphReg and CNN models in K562 cells, together with DeepSHAP and saliency scores of the four models for *MYC*, and ABC scores and FlowFISH results for *MYC*. Large negative scores in the FlowFISH track indicate true functional enhancers for *MYC*. Feature attribution tracks in Figure 3E show that GraphReg models are able to capture the most distal enhancers of *MYC* (green boxes), about 2 Mb away from its TSS, but the CNN models fail to capture these enhancers and produce false negatives (red boxes).

We further evaluated GraphReg using recent chromosome-wide enhancer screening data from targeted perturb-seq (TAP-seq) for Chromosomes 8 and 11 in K562 cells (Schraivogel et al. 2020). We restricted to the 35 genes with at least one functional enhancer, considering all screened enhancers whose distance from the target gene TSS is <2 Mb. Figure 3C shows the distribution of auPR for these 35 genes. Similar to CRISPRi-FlowFISH results (Fig. 3A), GraphReg models have better median auPR than ABC scores or the CNN models using either DeepSHAP or saliency as the feature attribution method. Pooling all the E-G pairs of these 35 genes (4423 E-G pairs in total), precision-recall analysis (Fig. 3D) shows that the highest auPR for the GraphReg models is

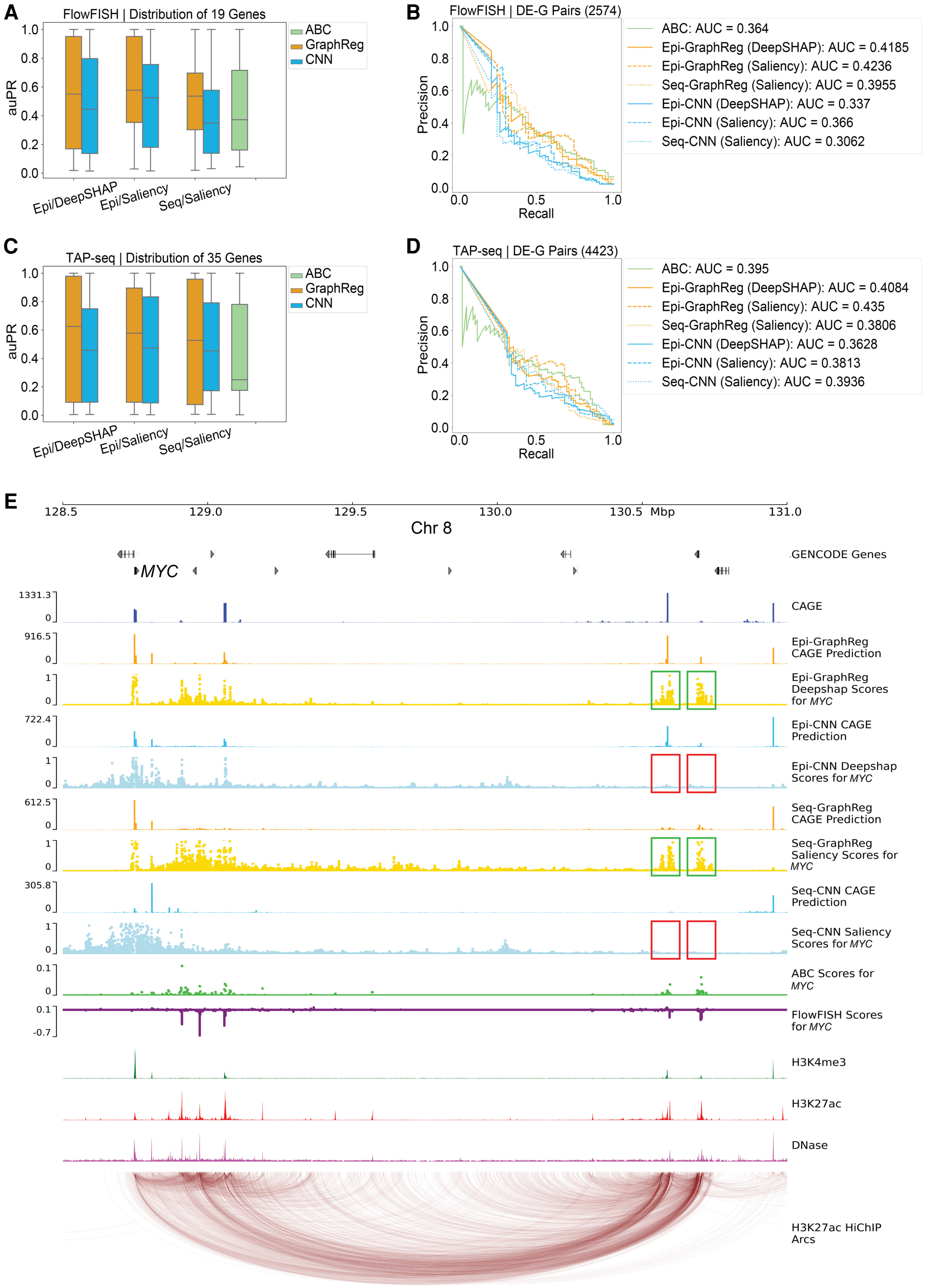

Figure 3. GraphReg models more accurately identify functional enhancers of genes. (*A*) Distribution of the area under the precision-recall curve (auPR) for 19 genes in K562 cells based on CRISPRi-FlowFISH data. GraphReg models have higher median of auPR than both CNN and activity-by-contact (ABC) models. (*B*) Precision-recall curves of the GraphReg, CNN, and ABC models for identifying enhancers of 19 genes screened by CRISPRi-FlowFISH. (*C*) Distribution of auPR for 35 genes in K562 cells based on TAP-seq data. GraphReg models have higher median of auPR than both CNN and ABC models. (*D*) Precision-recall curves for the GraphReg, CNN, and ABC models for identifying functional enhancers of 35 genes as determined by TAP-seq. (*E*) *MYC* locus (2.5 Mb) on Chr 8 with epigenomic data, true CAGE, predicted CAGE using GraphReg and CNN models, HiChIP interaction graph, and the saliency maps of the GraphReg and CNN models, all in K562 cells. Experimental CRISPRi-FlowFISH results and ABC values are also shown for *MYC*. Feature attribution shows that GraphReg models exploit HiChIP interaction graphs to find the distal enhancers, whereas CNN models find only promoter-proximal enhancers. Green and red boxes show true positives and false negatives, respectively. CNN models miss the distal enhancers and consequently lead to false negatives in very distal regions.

0.435 (Epi-GraphReg with saliency), outperforming the best CNN model (auPR 0.3936) and ABC (auPR 0.395).

As another example of distal enhancer discovery in the TAP-seq data set, we considered screened enhancers more than 20 kb from TSS of *IFITM1* (79 distal DE-G pairs, five true functional enhancers). Precision-recall analysis for *IFITM1* shows that the highest auPR among the GraphReg models is 0.894 (Epi-GraphReg with saliency), compared to 0.635 for the best CNN model (Epi-CNN with DeepSHAP) and 0.751 for ABC (Supplemental Fig. S31B). We also plotted a 250-kb genomic region containing the *IFITM1* locus, with the corresponding K562 epigenomic data, HiChIP graph edges, true and predicted CAGE signals by GraphReg and CNN models, feature attribution scores for *IFITM1*, ABC scores for *IFITM1*, and TAP-seq results for *IFITM1* (Supplemental Fig. S31C). Large negative scores in the TAP-seq track indicate true functional enhancers of *IFITM1*. Here, all GraphReg, CNN, and ABC models are able to capture the distal enhancers of *IFITM1* (green boxes), but the CNN models also capture many nonfunctional enhancers as false positives (red boxes).

Seq-GraphReg predicts the direct targets of transcription factor perturbations via motif ablation

To assess whether Seq-GraphReg captures meaningful transcription factor (TF) motif information, we asked whether we could predict differential gene expression in TF perturbation experiments by applying our model to genomic sequences where the TF's binding motif had been ablated. We downloaded RNA-seq data for 51 CRISPRi TF knockout (KO) experiments in K562 cells from ENCODE and retained 29 experiments for which at least 200 genes were significantly down-regulated. Using two trained models each for Seq-GraphReg and Seq-CNN, we predicted gene expression using wild-type genomic sequences and sequences in which hits of a given TF motif had been zeroed out, then performed differential expression analysis on these values to predict the 100 most down-regulated target genes for the corresponding TF KO experiment (Methods). To have confident true labels, we restricted this analysis to genes that have true significant differential expression (up- or down-regulated) in the real respective TF KO experiments and that also have a wild-type CAGE value of at least 20.

Figure 4A shows the distributions of true mean logFC of the 100 predicted target genes over all TFs for the Seq-GraphReg and Seq-CNN models. As a baseline, we report the distribution of true mean logFC over all significantly differential genes ($P_{adj} < 0.05$). This baseline actually shows the average performance of a random algorithm that chooses 100 genes randomly out of all significantly differential genes. We also show the corresponding distributions after restricting to predicted genes with $n \geq 5$, where n denotes the number of 3D interactions. In both settings, Seq-GraphReg's predicted target genes are significantly more down-regulated than those of Seq-CNN's or than baseline performance (Wilcoxon signed-rank test on distributions of mean logFC values). Figure 4B shows the heatmaps of the true mean logFC of the top 100 predicted genes by Seq-GraphReg and Seq-CNN for each TF, for $n \geq 0$ and $n \geq 5$. For the majority of TFs, the mean logFC of Seq-GraphReg's predicted targets is more negative than that of Seq-CNN's predicted targets, validating the improved performance of Seq-GraphReg for identifying true down-regulated genes. Figure 4C shows the distributions of the precision (fraction of true significantly down-regulated genes among the 100 predicted genes) for all TFs for Seq-GraphReg and Seq-CNN, for $n \geq 0$ and $n \geq 5$. As baseline, we report the fraction of significantly down-regulated genes ($P_{adj} < 0.05$ and logFC < 0) over all significantly differential genes ($P_{adj} < 0.05$), for each TF. This baseline shows the average performance of a random algorithm. Again, the precision values are always highest for Seq-GraphReg and significantly greater than for Seq-CNN and baseline (Wilcoxon signed-rank test). Figure 4D shows this data in heatmap form, confirming that for the majority of TFs, the precision for Seq-GraphReg is higher than for Seq-CNN.

Figure 4E depicts a visual example of predicting the effect of *JUND* KO on the gene *TCF3* in K562. *TCF3* is among the top 20 predicted genes of Seq-GraphReg for the JUND TF (Supplemental Fig. S32A), and CRISPRi KO of *JUND* leads to significant down-regulation of *TCF3*. Figure 4E shows JUND motif hits (blue bars) around *TCF3*, indicating the motifs ablated for in silico prediction, along with JUND ChIP-seq data in K562 cells, demonstrating that the motif hits indeed decrease in JUND ChIP-seq peaks. To see if Seq-GraphReg can identify the JUND motifs in distal enhancers, we did the following. First, we chose two distal enhancers (indicated in Fig. 4E by A and B) of the gene *TCF3* with direct (one-hop) enhancer–promoter interactions in the HiChIP graph. Enhancers with direct interactions will have a bigger effect size on gene expression upon mutation. We also visualize graph heatmaps (adjacency matrices) instead of loops to facilitate viewing the interactions. Second, we used in silico mutagenesis (ISM) for feature attribution. We performed ISM on Seq-GraphReg trained with fast Fourier transform (FFT) loss to get more meaningful and interpretable motifs. We have borrowed the idea of adding FFT loss from a recent study (Tseng et al. 2020). We used FFT in the separate training scheme of Seq-GraphReg instead of end-to-end training owing to GPU memory limitation. Figure 4E shows in silico mutagenesis results for the two distal enhancer regions A (1.08 Mb downstream) and B (1.29 Mb upstream) of the gene *TCF3*, whose interactions with the *TCF3* promoter are marked by blue circles in the HiChIP graph. The promoter of *TCF3* is specified by a green line, and the enhancers A and B are specified by red lines. ISM is performed in 100-bp regions of enhancers A and B, each centered at a JUND motif. The ISM heatmaps show the difference in predictions (mutated – reference) after applying a mutation at each nucleotide. The heatmaps around the JUND motif in both enhancers A and B are blueish, meaning that ISM identifies the importance of the JUND motif in these regions for target gene expression prediction. In other words, mutations in the JUND motif in distal enhancers A and B lead to a reduction in predicted expression, showing the positive effect of these distal JUND motifs on the expression of *TCF3*. The base-level representations of ISM scores are the negative summation of all four scores (only three are non-zero) at each nucleotide, which identify the JUND motifs.

Overall, these results show that Seq-GraphReg models can better capture the regulatory relationships of TFs and genes than Seq-CNN models. This validates the primary motivation for our model—namely, that TFs can regulate their target genes by binding distal enhancers that loop to target promoters, and that these regulatory effects cannot be effectively learned without 3D interaction data. GraphReg is therefore the model of choice to capture complex gene regulatory relationships. To further show the potential of Seq-GraphReg for decoding gene regulation, we looked at the feature attributions of the 100 best predicted complex genes (having high numbers of E–P interactions) in GM12878, K562, and hESC and identified distal TF motifs that play a role in regulating those genes. Supplemental Figures S33–S38 show the most enriched distal (at least 20 kb away from the TSS) TF motifs and their contributions in regulating their target genes in each cell type

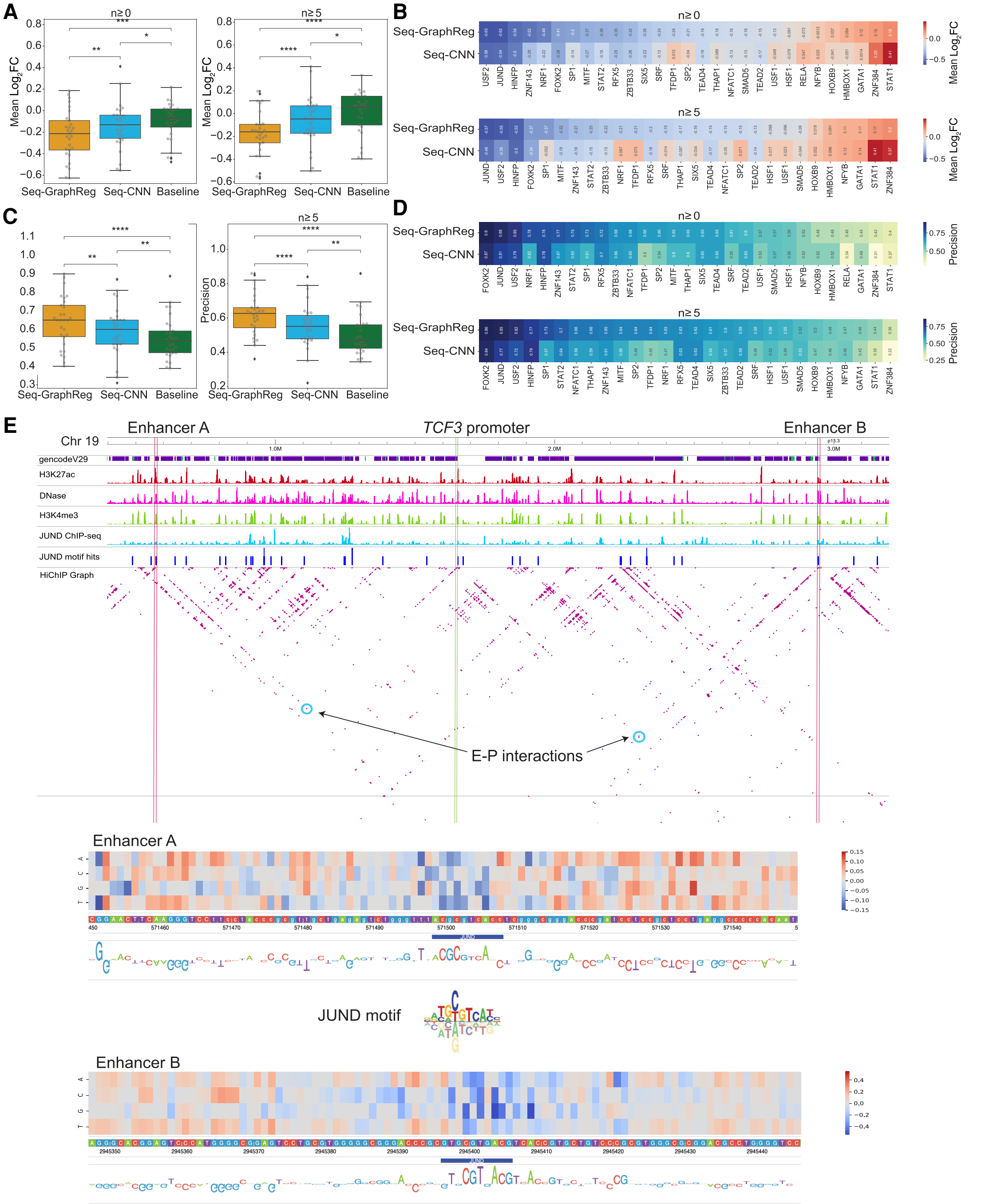

Figure 4. Seq-GraphReg accurately predicts the regulatory effects of transcription factor knockouts by in silico motif ablation. (*A*) Distributions of true mean logFC (over 100 predicted target genes) over all TFs for Seq-GraphReg, Seq-CNN, and baseline for $n \geq 0$ and $n \geq 5$, where n denotes the number of enhancer–promoter (E–P) interactions. The median of true mean logFCs for predicted genes by Seq-GraphReg is negative, and the distribution is significantly more down-regulated than that of Seq-CNN and baseline (Wilcoxon signed-rank test). (*B*) Heatmaps of the true mean logFC of the top 100 predicted genes by Seq-GraphReg and Seq-CNN for each TF, for $n \geq 0$ and $n \geq 5$. For the majority of TFs, the mean logFC of Seq-GraphReg's predicted targets is more negative than that of Seq-CNN's targets. (*C*) Distributions of precision values (fraction of true significantly down-regulated genes among 100 predicted genes) of all TFs for Seq-GraphReg, Seq-CNN, and baseline, for $n \geq 0$ and $n \geq 5$. The precision is always highest in Seq-GraphReg and significantly greater than for Seq-CNN and baseline (Wilcoxon signed-rank test). (*D*) Heatmaps of precision values (fraction of true significantly down-regulated genes among the top 100 predicted genes) for Seq-GraphReg and Seq-CNN for each TF and for $n \geq 0$ and $n \geq 5$. For the majority of TFs, the precision of Seq-GraphReg is higher than Seq-CNN. (*E*) A visual example of the effect of *JUND* KO on the gene *TCF3*. JUND motif hits around *TCF3* are plotted in blue bars. The promoter of *TCF3* is indicated by green lines, and two distal enhancers A (1.08 Mb downstream) and B (1.29 Mb upstream) of the gene *TCF3* by red lines. The interactions of enhancers A and B with the promoter of *TCF3* in HiChIP graph are marked by blue circles. In silico mutagenesis (ISM) is performed in 100-bp regions of enhancers A and B, each centered at a JUND motif, and the ISM heatmaps are shown. The heatmaps show the difference in predictions (mutated − reference) after applying a mutation at each nucleotide. The heatmaps around the JUND motif in both enhancers A and B are blueish, indicating the importance of JUND motif in these regions for *TCF3* expression prediction. The base-level representations of ISM scores are the negative summation of all four scores (only three are non-zero) at each nucleotide.

(Methods). Such motif analysis results are interesting in that they provide insights about the interplay between the distal TF motifs and their target genes. Although these results are at this stage just predictions by the Seq-GraphReg model, it would be interesting to pursue experimental validation by deletion of the identified distal TF motifs and assessing the resulting effect on target gene expression. We hope to perform such validation experiments in other biologically relevant systems in the near future.

Discussion

Until now, machine learning models for the genome-wide prediction of gene expression in a given cell state, or gene expression changes in a cell state transition, have largely relied on 1D epigenomic data and/or genomic DNA sequence without using 3D genomic architecture. Prior models include regularized linear regression approaches (González et al. 2015; Osmanbeyoglu et al. 2019) as well as more recent work using convolutional neural networks (Singh et al. 2016; Kelley et al. 2018; Zhou et al. 2018; Agarwal and Shendure 2020; Avsec et al. 2021a). However, previous linear models used a fixed assignment of regulatory elements to genes without incorporating 3D interaction data, whereas current deep learning models consider relatively local features (such as promoters and at most nearby enhancers) and therefore cannot capture the impact of distal regulatory elements, which can be 1 Mb or farther away from gene promoters. Two previous studies tried to use 3D data to predict gene expression (Zeng et al. 2019; Bigness et al. 2022) but did not address important aspects of modeling gene regulation. The most directly relevant one, GC-MERGE (Bigness et al. 2022), uses histone modifications and Hi-C data to predict gene expression (RNA-seq) using graph convolutional networks (GCN); however, this model does not provide any new insights about regulation of genes, such as finding functional enhancers or revealing the regulatory role of TF binding motifs. Our proposed method, GraphReg, is the most comprehensive and versatile model that is capable of revealing mechanistic information on gene regulation. We have provided the properties of previous deep learning models for gene expression prediction in Supplemental Table S1. The comparisons of GraphReg with these methods in terms of Pearson's correlation (R) of predicted versus true gene expression values are provided in Supplemental Tables S2, S3, and Supplemental Figure S12.

We have shown that GraphReg more effectively models the impact of distal enhancers on gene expression than 1D dilated CNNs. Although dilated CNNs have a large receptive field, our feature attribution analyses show that only relatively promoter-proximal information influences gene expression prediction in these models. In contrast, GraphReg exploits 3D chromatin interactions to access distal information up to 2 Mb from the gene promoter. In other words, GraphReg adds inductive bias from biology to the deep predictive model, which converts a black-box model attending to everywhere to a more focused model attending to relevant regions.

An alternative attention-based methodology for learning long-range interactions in sequential input is the transformer, a model that provides state-of-the-art performance in machine translation and other natural language processing tasks (Vaswani et al. 2017). A recent work, Enformer (Avsec et al. 2021a), introduced a transformer model for the prediction of CAGE as well as epigenomic tracks from genomic sequence. Currently, however, this transformer-based architecture can integrate information up to 100 kb away in the genome, an order of magnitude less than GraphReg, while requiring considerable computational resources to train. GraphReg's use of 3D interaction data through graph attention networks therefore provides an efficient and biologically well-motivated means to encode distal regulation.

We showed that Seq-GraphReg meaningfully encodes TF motif binding information by performing motif ablation experiments, in which predicted differential gene expression based on ablated versus wild-type genomic sequences recovered true regulatory effects as measured by CRISPRi TF knockout experiments. A longer-term goal of this work is to model the impact of distal regulatory variants on gene expression, extending efforts to predict the regulatory impact of promoter-proximal genetic variants (Zhou et al. 2018). We anticipate that it will be critical to train such models using true genetic sequence variation and expression in disease-relevant cellular contexts as these data become available.

Although we currently use GraphReg to predict bulk CAGE-seq counts with Poisson loss, other transcriptomic assays could be explored in future work. For example, suitably processed nascent transcription assays such as GRO-seq would yield a quantification of promoter activity as well as enhancer RNA expression and could potentially be used to train GraphReg models. We also envision extensions to single-cell multi-omic data sets—training at the pseudo-bulk level where signals are aggregated over cell clusters or metacells, or even at the single-cell level. Here, (aggregated) scRNA-seq gives a tag-based output signal similar to CAGE-seq. Therefore, we anticipate that GraphReg models will have broad applicability for interpreting the function of epigenomic and genomic variation on gene expression.

Methods

Epigenomic data processing

We use bigWig coverage tracks for epigenomic data. We acquired these tracks in three ways: (1) by downloading the processed bigWig files from the ENCODE portal; (2) by using the *bam_cov.py* script from Basenji (Kelley et al. 2018), which reads in the BAM alignment file and generates the bigWig files; and (3) by using the *bamCoverage* command from deepTools (Ramírez et al. 2016), which reads in a BAM alignment file and returns the bigWig file using different normalization schemes. We used 1× normalization or reads per genome coverage (RPGC), which normalizes the coverage tracks by sequencing depth. For cross-cell-type analyses in which we train on one cell type and test on another one, the normalization of epigenomic coverage tracks is critical to avoid any undesired batch effects. We used two different normalizations for this purpose: (1) RPGC, and (2) DESeq-like normalization. We bin the epigenomic tracks at 100 bp, get the coverage in each bin, and apply log-normalization using the function $\log_2(x+1)$. This log-normalization is performed for the epigenome-based models in which the epigenomic data are inputs to the models.

CNN layers for learning local representations of 1D data

The inputs to the first layer of the Epi-GraphReg (Epi-CNN) and Seq-GraphReg (Seq-CNN) models are 1D epigenomic data and genomic DNA sequence, respectively. Regardless of the 1D input type, we use several CNN layers followed by ReLU activation, BatchNorm, dropout, and max-pooling to learn local representations for 5-kb bins of the genome (Supplemental Fig. S1). We consider genomic regions of 6 Mb in length as our input; hence, we have vectors of size $N = 1200$, representing 5-kb bins over the 6-Mb region, for CAGE values. We bin the epigenomic data at 100-

bp resolution; therefore, the length of each epigenomic input vector would be 60,000. In the GraphReg models, we define the set of local representations by $H = \{h_1, h_2, \ldots, h_N\}$, where $h_i \in \mathbb{R}^F$ and $F = 128$ is the channel number of the last CNN layer just before the GAT block. Then H is given to the GAT block, which is a type of graph neural network (GNN) (Kipf and Welling 2017; Veličković et al. 2018; Xu et al. 2019), where h_i is the node feature of node i. To make the comparisons fair, in Epi-CNN and Seq-CNN models we use eight layers of dilated CNNs with residual connections (Kelley et al. 2018; Avsec et al. 2021b) to increase the receptive fields up to 2.5 Mb upstream and downstream from gene TSSs, where the dilation rate is multiplied by 2 at each layer. In the sequence-based models (Seq-GraphReg and Seq-CNN) we also predict epigenomic data to guide the models to learn informative local motif representations; after the first five CNN layers, where the resolution is 100 bp, we use six layers of dilated CNNs with residual connections to predict three epigenomic assays, H3K4me3, H3K27ac, and DNase, at 100-bp resolution (Supplemental Fig. S1). The learned local features just before six dilated CNN layers are called bottleneck representations that are passed through some CNN layers with max-pooling to bring the resolution to 5 kb and then are given to GAT block in Seq-GraphReg (and eight layers of dilated CNN in Seq-CNN) to predict the CAGE (Supplemental Fig. S1).

Graph attention networks for integration of local features via 3D interaction data

We use the graphs extracted from 3D data (Hi-C, HiChIP, Micro-C, HiCAR) to capture gene regulatory interactions and predict gene expression (CAGE-seq). We processed 3D data sets at 5 kb resolution using HiC-DC+ (Carty et al. 2017; Sahin et al. 2021) and kept only the significant interactions with three significance levels of FDR less than 0.1, 0.01, and 0.001. The nodes in the extracted graphs therefore represent 5-kb genomic bins, and the edges indicate the most significant interactions between bins. Each 6-Mb genomic input region thus corresponds to a graph with $N = 1200$ nodes whose features $H = \{h_1, h_2, \ldots, h_N\}$ have been learned in the previous CNN block.

The graph attention network (GAT) block receives a graph $G = (V, E)$ and a set of node features $H^t = \{h_1^t, h_2^t, \ldots, h_N^t\}$ from the previous layer t and outputs an updated set of node features $H^{t+1} = \{h_1^{t+1}, h_2^{t+1}, \ldots, h_N^{t+1}\}$. Each GAT layer uses two weight matrices: $W_p^t \in \mathrm{R}^{F' \times F}$ for promoters (or self-nodes) and $W_e^t \in \mathrm{R}^{F' \times F}$ for enhancers (or neighbor nodes), where F' is the number of output features in each GAT layer. The importance of distinct enhancers on a promoter might be different. In the case of GCNs, all enhancers are treated as of equal importance because all the E–P weights are the same. However, by using GAT, we give the model the ability to adjust the weights of different enhancers based on their importance for the task of predicting expression of a specific gene. Therefore, the GAT formulation makes more biological sense for our problem. Here, unlike previous graph neural networks (Veličković et al. 2018), we do not include self-loops in the graph G. We have decoupled self-loops and neighbor-loops because of the differing role of promoters and enhancers in the model. By only including the neighbor-loops in the graph G, the model focuses on distal enhancers that cannot be captured in local models such as CNNs.

We define the self-attention mechanism from node j to node i at layer t as

$$\beta_{ij}^t = \sigma((a_p^t)^T W_p^t h_i^t + (a_e^t)^T W_e^t h_j^t), \tag{1}$$

where $a_p^t \in \mathrm{R}^{F'}$ and $a_e^t \in \mathrm{R}^{F'}$ are two weight vectors, $\sigma(.)$ is the sigmoid function, $\beta_{ij}^t \in \mathrm{R}$ is the attention weight from node j to node i at layer t, and T denotes the transpose of a vector. By using a sigmoid function instead of a conventional softmax function, we give extra freedom to the model to discard irrelevant and nonenhancer interactions. We also account for the cardinality of the nodes by defining $\alpha_i^t = \sigma(a^t \sqrt{|\mathcal{N}_i|} + b^t)$, where $a^t \in \mathrm{R}^{F'}$ and $b^t \in \mathrm{R}^{F'}$ are two weight vectors. Finally, we define the updates of the node features at the next layer as

$$h_i^{t+1} = f\left(\alpha_i^t \circ \left(W_p^t h_i^t + \sum_{j \in \mathcal{N}_i} \beta_{ij}^t W_e^t h_j^t\right)\right), \tag{2}$$

where f is a nonlinearity function and ° is the element-wise product, $\mathcal{N}_i$ is the set of neighbors of node i (not including itself), and $|\mathcal{N}_i|$ is the number of neighbors of node i. We use an exponential linear unit (ELU) for f. As in Veličković et al. (2018), we use K heads and concatenate the features as

$$h_i^{t+1} = ||_{k=1}^K f\left(\alpha_i^{t,k} \circ \left(W_p^{t,k} h_i^t + \sum\nolimits_{j \in \mathcal{N}_i} \beta_{ij}^{t,k} W_e^{t,k} h_j^t\right)\right) \tag{3}$$

where $\|_{k=1}^K$ means concatenation of K independent heads. Therefore, h_i^{t+1} will have KF' features for $i \in \{1, \ldots, N\}$ and $t \in \{1, \ldots, L\}$, where L is the number of GAT layers.

In Equation (3), the updated feature h_i^{t+1} at bin i is also a function of the number of its neighbors, namely, $|\mathcal{N}_i|$. The impact of node cardinality on its feature is nontrivial by adding $\alpha_i^{t,k}$ in Equation (3). We mentioned that Epi-GraphReg models are cell type–agnostic, meaning that they can be generalized from one cell type to another one. When we want to train the Epi-GraphReg models on some cell types and test them on other unseen cell types, this might lead to performance degradation if the distributions of node cardinality of graphs in the train and test cells are different. This could happen for many reasons, including the library sizes or FDR levels of Hi-C, HiChIP, Micro-C, or HiCAR data from which we extract the graphs. For example, in our experiments, we noticed this discrepancy between the graphs of GM12878 and K562 cells. To solve this and improve the generalization performance, we normalize the attention weights of each node i such that the summation of all the attention weights becomes one and we remove the cardinality parameter $\alpha_i^{t,k}$. As such, the normalized updates of GAT layers can be written as

$$h_i^{t+1} = f\left(\eta_{ii}^t W_p^t h_i^t + \sum_{j \in \mathcal{N}_i} \eta_{ij}^t W_e^t h_j^t\right), \tag{4}$$

where $\eta_{ii}^t = \dfrac{1}{1 + \sum_{j \in \mathcal{N}_i} \beta_{ij}^t}$ is the attention coefficient of node i (or promoter coefficient) and $\eta_{ij}^t = \dfrac{\beta_{ij}^t}{1 + \sum_{j \in \mathcal{N}_i} \beta_{ij}^t}$ is the attention coefficient from node j to node i (or enhancer coefficient). We see that $\eta_{ij} \in [0, 1]$ for all $i, j \in \{1, \ldots, N\}$ and all the attention coefficients of each node i sum up to one: $\sum_{j \in \mathcal{N}_i \cup \{i\}} \eta_{ij}^t = 1$. Similar to Equation (3), when having K heads, the normalized updates would be

$$h_i^{t+1} = ||_{k=1}^K f\left(\eta_{ii}^{t,k} W_p^{t,k} h_i^t + \sum\nolimits_{j \in \mathcal{N}_i} \eta_{ij}^{t,k} W_e^{t,k} h_j^t\right), \tag{5}$$

Because the second formulation in Equation (5) is more robust, it is used as the default choice for the GraphReg models.

Poisson regression and loss function

After the GAT layers in GraphReg models, the last layer before the predictions is a width-1 CNN layer (equivalent to a fully connected layer) with exponential nonlinearity to predict the CAGE value at each bin. As the CAGE values are counts, we used Poisson regression, meaning that the expected value of each (CAGE) output given the inputs is the mean of a Poisson distribution.

For the Epi-GraphReg model, we let X denote the 1D epigenomic inputs for a 6-Mb region, G the corresponding graph, $Y = [y_1, \ldots, y_N]$ the observed CAGE signal across 5-kb bins, and $f_i^\theta(X, G)$ the predicted CAGE signal for the bin i, where θ represents parameters of the model. Now we assume that $y_i|X, G \sim Poisson(\lambda_i)$, where $\lambda_i = f_i^\theta(X, G)$ and $E(Y|X, G) = f^\theta(X, G)$. Hence, the loss function is the negative log-likelihood of the Poisson distribution,

$$L_E = \frac{3}{N}\sum_{i=N/3}^{2N/3}(\log\Gamma(y_i+1) + f_i^\theta(X, G) - y_i \log f_i^\theta(X, G)), \qquad (6)$$

where $\Gamma(.)$ is the gamma function. We train our model for the middle one-third region (2 Mb or middle $N/3$ bins) instead of the whole 6-Mb region (all N bins) in each batch. Because we include 3D interactions of genomic distance up to 2 Mb, promoters in the middle 2-Mb region can see the effects of distal enhancers in the full 6-Mb input region. In the test phase we also restrict to predictions on the middle 2-Mb regions. We shift input regions by 2 Mb across genome so there is no overlap between predictions in two different batches. Depending on the number of GAT layers used, we can see the effects of enhancers via multi-hop interactions up to 4 Mb away from the promoters. If we have L GAT layers, message passing is done L times over the graph, and the promoters will see the effects of enhancers up to L hops.

For the Seq-GraphReg model, we also predict the 1D epigenomic data with a resolution of 100 bp after the dilated CNN layers from DNA sequence. Here, we let X denote the DNA sequence of the 6-Mb input region, G the corresponding graph, $Y^e = [y_1^e, \ldots, y_{N'}^e]$ the observed 100-bp epigenomic inputs (over N' = 60, 000 bins), and $g_i^e(X)$ the predicted epigenomic signal e for the bin i. The loss for each epigenomic signal e is given by

$$L_S^e = \frac{1}{N'}\sum_{i=1}^{N'}(\log\Gamma(y_i^e+1) + g_i^e(X) - y_i^e \log g_i^e(X)), \qquad (7)$$

and the CAGE loss L_S^{cage}, similar to Equation (6), can be written as

$$L_S^{cage} = \frac{3}{N}\sum_{i=N/3}^{2N/3}(\log\Gamma(y_i+1) + f_i^\theta(H^\star, G) - y_i \log f_i^\theta(H^\star, G)), \qquad (8)$$

where $H^\star$ is the bottleneck representation just before dilated CNN layers (shown by a star [$\star$] in Supplemental Fig. S1) at 100-bp resolution. The final loss function to be minimized is the convex combination of the CAGE loss and the epigenomic losses,

$$L_S = \lambda L_S^{cage} + (1-\lambda)\sum_{e=1}^{3} L_S^e, \qquad (9)$$

where λ is a hyperparameter to control the weights of the CAGE loss versus the epigenomic losses. We use a sigmoid-like schedule for λ given by $\lambda = \max\left(0.01, \frac{1}{1+\exp(-(epoch-10))}\right)$, which starts with a small value $\lambda = 0.01$ and approaches a value close to one at higher epochs. This schedule lets the model first learn meaningful motif representations from DNA sequences to predict the epigenomic data and then puts higher weights on predicting the CAGE signals from those motif representations at 100-bp resolution. *NLL* in all figures refers to L_E and L_S^{cage} for epigenome-based and sequence-based models, respectively.

End-to-end versus separate training of Seq-GraphReg

Seq-GraphReg as shown in Supplemental Figure S1 is an end-to-end model with two main tasks: (1) epigenomic tracks are predicted from DNA sequences; and (2) CAGE is predicted from the bottleneck representation $H^\star$ learned in the first task. For end-to-end training, we need to load the entire 6-Mb DNA sequence to the GPU memory, and this can limit our capability to use more CNN filters in the first layers to learn a richer bottleneck representation. Therefore, the overall CAGE prediction might be sacrificed because of GPU memory limitations. To resolve this problem, we propose another scheme for training Seq-GraphReg, which we call separate training. The separate training works as follows. First, we perform task (1) to predict the epigenomic data from DNA sequences of length 100 kb. Then, we freeze the learned parameters of the CNN layers and get the bottleneck representations $H^\star$, which are at 100-bp resolution. Finally, we use $H^\star$ as the input to the second block (Supplemental Fig. S1) to predict the CAGE values. The second block has several CNN and max-pool layers to bring the resolution to 5 kb, and then it is given to three GAT layers to predict the CAGE values.

We have shown that separate training on K562 outperforms end-to-end training in terms of CAGE prediction performance. The reason for the improvement is that in separate training, we use 100-kb DNA sequences (instead of 6 Mb), so we do not exhaust GPU memory and thus can use more CNN filters to learn richer bottleneck representations that consequently lead to superior gene expression prediction. However, this prediction improvement comes at the cost of losing base-pair level saliency scores via backpropagation. The reason for this is that to get the gradient of the output CAGE with respect to the input DNA sequence via backpropagation, we need an end-to-end model from input to the output. Nevertheless, we can use perturbation-based feature attribution methods such as in silico mutagenesis (ISM) for the separately trained Seq-GraphReg models (as was performed and is shown in Fig. 4E). However, backpropagation-based feature attribution methods, such as gradient-by-input and DeepSHAP, are much faster than ISM. Overall, there is a trade-off between better prediction performance (achieved by separate training) and having fast backpropagation-based feature attribution capability (achieved by end-to-end training).

Dilated CNN layers and FFT loss in Seq-GraphReg

In separate training of Seq-GraphReg, we also studied the effects of using dilated CNN layers and also adding FFT loss to get better motif interpretations. In task (1) of separate training, where we predict epigenomic data from DNA sequence, we removed the residual dilated CNN layers and instead used three CNN layers after the bottleneck $H^\star$ for each task of predicting H3K4me3, H3K27ac, and DNase. We observed that not using dilated layers in fact helped prediction performance in K562 (Supplemental Fig. S11). This is partly because when we do not use dilated layers, the CNN filters can focus more on local peak regions and learn better motif representations.

Because our goal is not only prediction but also interpretation of base-pair feature attribution methods, we used the idea of adding FFT loss to the prediction loss (*NLL*) at training, as suggested in Tseng et al. (2020). Here, we explain how we used FFT loss in separate training of Seq-GraphReg models. In task (1), let $g(H^\star, X)$ be the gradient of bottleneck $H^\star$ with respect to the input DNA sequence X and summed over 4 bases as our attribution vector. Thus, the length of the vector $g(H^\star, X)$ is the same as that of X. Let a denote the magnitudes of the positive-frequency Fourier components of (a slightly smoothed version of) $g(H^\star, X)$, and $\hat{a}_i$ denote the ith component of $\frac{a}{\|a\|_1}$ (smaller i corresponds to lower frequencies). Similar to Tseng et al. (2020), we penalize the high-frequency components as

$$L_p(g(H^\star, X)) = 1 - \sum_i w_i \hat{a}_i, \qquad (10)$$

where $w_i = 1$ for $i \le T$, and $w_i = \frac{1}{1 + (i - T)^s}$ for $i > T$. The total loss to be trained is $L_{total} = \frac{1}{3}\sum_{e=1}^{3} L_S^e + \gamma L_p(g(H^\star, X))$, where L_S^e is the prediction loss (NLL) of the epigenomic data e, $L_p(g(H^\star, X))$ is attribution prior, and γ is a positive scalar to control the contribution of attribution loss. We have used $s = 0.2$ and $\gamma = 1$ as suggested by Tseng et al. (2020). We have applied ISM to the Seq-GraphReg models trained separately without dilation and with FFT loss and observed that relevant TF motifs can be identified in distal enhancer regions (Fig. 4E).

Model architecture and training

For Epi-GraphReg and Seq-GraphReg, we used respectively two and three GAT layers with four heads in each layer (Supplemental Fig. S1). For Epi-CNN and Seq-CNN we used eight dilated CNN layers to increase their receptive field up to 2.5 Mb. For human cell types GM12878, K562, and hESC, we held out Chromosomes i and $i + 10$ for validation and Chromosomes $i + 1$ and $i + 11$ for test, for $i = 1, \ldots, 10$, and trained on all remaining chromosomes except X and Y. For mouse cell type mESC, we held out Chromosomes i and $(i + 10 \bmod 20) + \text{sign}(i \ge 10)$ for validation and Chromosomes $i + 1$ and $(i + 11 \bmod 20) + \text{sign}(i \ge 9)$ for test, for $i = 1, \ldots, 10$, and trained on all remaining chromosomes except X and Y.

TF knockout analysis and motif ablation experiments

For RNA-seq data sets from CRISPRi TF KO experiments, we performed differential gene expression analyses using DESeq2 (Anders and Huber 2010) between control and TF KO K562 cells for each TF. We only considered TFs whose KO led to significant down-regulation (adjusted $P < 0.05$) of at least 200 genes to have robust statistics. To assess the ability of Seq-GraphReg and Seq-CNN to predict the down-regulation of genes as direct effects of TF KO, we performed an in silico KO of each TF by zeroing out all the motifs for the TF in 6-Mb DNA sequences encompassing candidate target genes. Then we used the models to predict the change in gene expression caused as direct effects of each TF KO. Similar to the true KO experiments, we used two trained models each for Seq-GraphReg and Seq-CNN as two replicates to predict control and in silico KO expression levels. We then performed differential expression analysis using DESeq2 to get the predicted logFC (log-fold change) and adjusted P-values. We ranked the significantly regulated genes (adjusted $P < 0.05$) with true CAGE values at least 20 based on predicted logFC and used the 100 genes with the largest negative predicted logFC as the predicted target genes for each model.

Seq-GraphReg predicts how distal TF motifs regulate their target genes

We first looked at our gene expression predictions using Seq-GraphReg in GM12878, K562, and hESC, and we selected the top 100 accurately predicted genes using the following criteria: $NLL < 300$, $Delta_{NLL} > 300$, CAGE ≥ 100, and $n \ge n_{EP}$, where n is the number of enhancer–promoter interactions. Because GM12878 and K562 have denser graphs, we used $n_{EP} = 10$ in GM12878 and K562. Because hESC has sparser graphs, we used $n_{EP} = 5$ in hESC. Then we performed feature attribution for all those genes using gradient-by-input, which gives us base-pair saliency scores for all the nucleotides in the enhancer regions of those genes. High saliency sequence regions correspond to our candidate motifs for each gene.

To find the candidate motifs we screened the enhancer regions of each gene using a window of length 20 bp (because the usual length of TF motifs is in the range 7–20 bp) and summed the saliency scores in each window. We shifted this window by steps of 10 bp to not miss motifs that might appear on the borders of the window. This process produces 500 windows (scores) in each enhancer bin of length 5 kb. If a gene has n enhancer–promoter interactions, there will be $500n$ windows for that gene. For each gene, we kept the windows that have the highest 5% absolute scores as the motif candidates for that gene, giving $25n$ windows with the highest absolute scores. To have a control set, we also retained for each gene $25n$ windows with the lowest non-zero absolute scores. To study the effects of distal motifs on gene regulation, we kept $25n$ sequences of length 20 bp in both high and low saliency regions that have 1D distance of at least 20 kb from the TSS of their target genes.

We used FIMO software (Grant et al. 2011) to scan these 20-bp sequences and find motif matches to the CIS-BP motif database. We kept all matches with P-values of at least 10^{-4} (default FIMO parameters) for both high and low saliency regions. We also restricted to motifs whose TF is expressed in the cell type of interest, meaning that CAGE ≥ 50 for each retained TF. Supplemental Figure S33A shows the number of motifs in high and low saliency regions in GM12878. The listed motifs in Supplemental Figure S33A are the ones that are significantly enriched in high saliency regions compared to low saliency ones, meaning that the adjusted P-values are less than 0.01. Here, we have used Fisher's exact test to derive the P-values and Benjamini–Hochberg (BH) to adjust for multiple hypothesis testing. The adjusted P-values for these motifs are shown in Supplemental Figure S34A. We also counted the occurrences of each motif for each gene individually in both high and low saliency regions, as shown in Supplemental Figure S33B. Then for each gene-motif pair, we performed Fisher's exact test to see if each motif is enriched in high saliency regions for the gene. The resulting P-values are shown in Supplemental Figure S34B, where we see the association of TF motifs with their target genes. Results in Supplemental Figure S34B suggest that combinations of distal TF motifs play a role in regulating the expression of target genes. Another important observation from Supplemental Figure S34B is that some of the TF motifs are universal and play a role in almost all the genes, but some of the motifs are more gene-specific and act only on the selected genes.

Supplemental Figures S35–S38 show similar motif analysis results for K562 and hESC. Three-dimensional data used by Seq-GraphReg for GM12878 and K562 is HiChIP (FDR = 0.1) and for hESC is Micro-C (FDR = 0.1). Supplemental Figures S36 and S38 show the enrichment of distal TF motifs for the best predicted genes in K562 and hESC, respectively.

Data sets

We downloaded the epigenomic and CAGE data of GM12878, K562, hESC, and mESC cells from ENCODE (https://www.encodeproject.org/) portal with the following accession numbers: K562 CAGE: ENCFF623BZZ and ENCFF902UHF; K562 DNase-seq: ENCFF826DJP; K562 H3K4me3: ENCFF689TMV, ENCFF915MJO (RPGC), and ENCFF367FNL (RPGC); K562 H3K27ac: ENCFF010PHG, ENCFF384ZZM (RPGC), and ENCFF070PWH (RPGC); K562 JUND TF ChIP-seq: ENCFF709JGL; GM12878 CAGE: ENCFF915EIJ and ENCFF990KLZ; GM12878 DNase-seq: ENCFF775ZJX and

ENCFF783ZLL; GM12878 H3K4me3: ENCFF818GNV, ENCFF342CXS (RPGC), and ENCFF818UQV (RPGC); GM12878 H3K27ac: ENCFF180LKW, ENCFF197QHX (RPGC), and ENCFF882PRP (RPGC); hESC (H1) CAGE: ENCFF525HJR and ENCFF521WGM; hESC (H1) DNase-seq: ENCFF131HMO and ENCFF546PJU (RPGC); hESC (H1) H3K4me3: ENCFF760NUN, ENCFF151CFM (RPGC), and ENCFF765FFD (RPGC); hESC (H1) H3K27ac: ENCFF919FBG, ENCFF860ABR (RPGC), and ENCFF693IFG (RPGC); mESC (ES-E14) DNase-seq: ENCFF754ILF and ENCFF785XJZ; mESC (ES-E14) H3K4me3: ENCFF240MDV; mESC (ES-E14) H3K27ac: ENCFF163HEV. We downloaded mESC CAGE data from the NCBI Gene Expression Omnibus (GEO; https://www.ncbi.nlm.nih.gov/geo/) with the accession numbers GSM3852792, GSM3852793, and GSM3852794. We got the H3K27ac HiChIP data of K562, GM12878, and mESC from GEO with the accession numbers GSE101498, GSE101498, and GSE113339, respectively. We obtained Hi-C data for K562 and GM12878 from GEO with accession number GSE63525. We obtained Micro-C data for hESC (H1) from the 4D Nucleome Data Portal (4DN; https://data.4dnucleome.org) with accession number of 4DNFI2TK7L2F. We obtained HiCAR data for hESC (H1) from GEO with accession number GSE162819. We got the CRISPRi-FlowFISH enhancer validation data set for K562 from Supplemental Table 6a of Fulco et al. (2019). We got the CRISPRi TAP-seq enhancer validation data set for K562 (Chromosomes 8 and 11) from Supplemental Tables 2 and 3 of Schraivogel et al. (2020). We acquired ABC enhancer predictions of the K562 (at Chromosomes 8 and 11) from https://osf.io/f2uvz/. We downloaded CRISPRi RNA-seq (TF knockout) experiments of 51 TFs (plus two controls) in K562 from ENCODE with the accession numbers ENCSR439NIR, ENCSR447WYJ, ENCSR490XKT, ENCSR895NMN, ENCSR747UPR, ENCSR045BVL, ENCSR658NCC, ENCSR685TOO, ENCSR157JAK, ENCSR290RLW, ENCSR849RWU, ENCSR177JPA, ENCSR434ZAM, ENCSR467RQJ, ENCSR179XMY, ENCSR231XYT, ENCSR569WHC, ENCSR612JVF, ENCSR844UQZ, ENCSR466OEJ, ENCSR156EPV, ENCSR991TCB, ENCSR559VHS, ENCSR109KMO, ENCSR539CHL, ENCSR153WDD, ENCSR046XJC, ENCSR692NVT, ENCSR932JIP, ENCSR315EHR, ENCSR612NHI, ENCSR143YTV, ENCSR997FLI, ENCSR588MYR, ENCSR366ONT, ENCSR708MPN, ENCSR747ZDR, ENCSR521VAG, ENCSR045LNG, ENCSR199FFQ, ENCSR208KNW, ENCSR004ZBD, ENCSR975MRK, ENCSR966USM, ENCSR496MYK, ENCSR866PAO, ENCSR205DTM, ENCSR212QDE, ENCSR349LHQ, ENCSR043UEE, ENCSR785ATE, ENCSR095PIC, ENCSR016WFQ.

Software availability

The code to preprocess data, train models, and perform the analyses in the paper, as well as all the trained models, are available at GitHub (https://github.com/karbalayghareh/GraphReg) and as Supplemental Code.

Competing interest statement

The authors declare no competing interests.

Acknowledgments

This work was supported by the National Institutes of Health/National Human Genome Research Institute (NIH/NHGRI) U01 awards HG009395 and HG012103 and National Institutes of Health/National Institute of Diabetes and Digestive and Kidney Diseases (NIH/NIDDK) U01 award DK128852.

Author contributions: A.K. developed the machine learning models, performed all computational experiments, and co-wrote the manuscript. M.S. performed analysis on HiChIP data sets. C.S.L. supervised the research and co-wrote the manuscript.

References

Agarwal V, Shendure J. 2020. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. *Cell Rep* **31:** 107663. doi:10.1016/j.celrep.2020.107663

Anders S, Huber W. 2010. Differential expression analysis for sequence count data. *Genome Biol* **11:** R106. doi:10.1186/gb-2010-11-10-r106

Avsec Ž, Agarwal V, Visentin D, Ledsam JR, Grabska-Barwinska A, Taylor KR, Assael Y, Jumper J, Kohli P, Kelley DR. 2021a. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat Methods* **18:** 1196–1203. doi:10.1038/s41592-021-01252-x

Avsec Ž, Weilert M, Shrikumar A, Krueger S, Alexandari A, Dalal K, Fropf R, McAnany C, Gagneur J, Kundaje A, et al. 2021b. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat Genet* **53:** 354–366. doi:10.1038/s41588-021-00782-6

Bigness J, Loinaz X, Patel S, Larschan E, Singh R. 2022. Integrating long-range regulatory interactions to predict gene expression using graph convolutional networks. *J Comput Biol* doi:10.1089/cmb.2021.0316

Buenrostro JD, Wu B, Chang HY, Greenleaf WJ. 2015. ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Curr Protoc Mol Biol* **109:** 21–29. doi:10.1002/0471142727.mb2129s109

Carty M, Zamparo L, Sahin M, González A, Pelossof R, Elemento O, Leslie CS. 2017. An integrated model for detecting significant chromatin interactions from high-resolution Hi-C data. *Nat Commun* **8:** 15454. doi:10.1038/ncomms15454

Fulco CP, Nasser J, Jones TR, Munson G, Bergman DT, Subramanian V, Grossman SR, Anyoha R, Doughty BR, Patwardhan TA, et al. 2019. Activity-by-contact model of enhancer–promoter regulation from thousands of CRISPR perturbations. *Nat Genet* **51:** 1664–1669. doi:10.1038/s41588-019-0538-0

González AJ, Setty M, Leslie CS. 2015. Early enhancer establishment and regulatory locus complexity shape transcriptional programs in hematopoietic differentiation. *Nat Genet* **47:** 1249–1259. doi:10.1038/ng.3402

Grant CE, Bailey TL, Noble WS. 2011. FIMO: scanning for occurrences of a given motif. *Bioinformatics* **27:** 1017–1018. doi:10.1093/bioinformatics/btr064

Kelley DR, Reshef YA, Bileschi M, Belanger D, McLean CY, Snoek J. 2018. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Res* **28:** 739–750. doi:10.1101/gr.227819.117

Kipf TN, Welling M. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France.

Krietenstein N, Abraham S, Venev SV, Abdennur N, Gibcus J, Hsieh THS, Parsi KM, Yang L, Maehr R, Mirny LA, et al. 2020. Ultrastructural details of mammalian chromosome architecture. *Mol Cell* **78:** 554–565.e7. doi:10.1016/j.molcel.2020.03.003

Lieberman-Aiden E, Van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie BR, Sabo PJ, Dorschner MO, et al. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326:** 289–293. doi:10.1126/science.1181369

Lundberg SM, Lee SI. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA, pp. 4765–4774.

Mumbach MR, Rubin AJ, Flynn RA, Dai C, Khavari PA, Greenleaf WJ, Chang HY. 2016. HiChIP: efficient and sensitive analysis of protein-directed genome architecture. *Nat Methods* **13:** 919–922. doi:10.1038/nmeth.3999

Osmanbeyoglu HU, Shimizu F, Rynne-Vidal A, Alonso-Curbelo D, Chen HA, Wen HY, Yeung TL, Jelinic P, Razavi P, Lowe SW, et al. 2019. Chromatin-informed inference of transcriptional programs in gynecologic and basal breast cancers. *Nat Commun* **10:** 4369. doi:10.1038/s41467-019-12291-6

Ramírez F, Ryan DP, Grüning B, Bhardwaj V, Kilpert F, Richter AS, Heyne S, Dündar F, Manke T. 2016. deepTools2: a next generation web server for deep-sequencing data analysis. *Nucleic Acids Res* **44:** W160–W165. doi:10.1093/nar/gkw257

Sahin M, Wong W, Zhan Y, Van Deynze K, Koche R, Leslie CS. 2021. HiC-DC+ enables systematic 3D interaction calls and differential analysis for Hi-C and HiChIP. *Nat Commun* **12:** 3366. doi:10.1038/s41467-021-23749-x

Schraivogel D, Gschwind AR, Milbank JH, Leonce DR, Jakob P, Mathur L, Korbel JO, Merten CA, Velten L, Steinmetz LM. 2020. Targeted perturb-seq enables genome-scale genetic screens in single cells. *Nat Methods* **17:** 629–635. doi:10.1038/s41592-020-0837-5

Shiraki T, Kondo S, Katayama S, Waki K, Kasukawa T, Kawaji H, Kodzius R, Watahiki A, Nakamura M, Arakawa T, et al. 2003. Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proc Natl Acad Sci* **100:** 15776–15781. doi:10.1073/pnas.2136655100

Shrikumar A, Greenside P, Kundaje A. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia. *PMLR* **70:** 3145–3153. https://proceedings.mlr.press/v70/shrikumar17a.html

Singh R, Lanchantin J, Robins G, Qi Y. 2016. DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics* **32:** i639–i648. doi:10.1093/bioinformatics/btw427

Skene PJ, Henikoff S. 2017. An efficient targeted nuclease strategy for high-resolution mapping of DNA binding sites. *eLife* **6:** e21856. doi:10.7554/eLife.21856

Sundararajan M, Taly A, Yan Q. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia. *PMLR* **70:** 3319–3328. https://proceedings.mlr.press/v70/sundararajan17a.html

Tseng A, Shrikumar A, Kundaje A. 2020. Fourier-transform-based attribution priors improve the interpretability and stability of deep learning models for genomics. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pp. 1913–1923.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA, pp. 5998–6008.

Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC.

Wei X, Xiang Y, Peters DT, Marius C, Sun T, Shan R, Ou J, Lin X, Yue F, Li W, et al. 2022. HiCAR is a robust and sensitive method to analyze open-chromatin-associated genome organization. *Mol Cell* **82:** 1225–1238.e6. doi:10.1016/j.molcel.2022.01.023

Xu K, Hu W, Leskovec J, Jegelka S. 2019. How powerful are graph neural networks? In *International Conference on Learning Representations, ICLR 2019*, New Orleans, LA.

Zeng W, Wang Y, Jiang R. 2019. Integrating distal and proximal information to predict gene expression via a densely connected convolutional neural network. *Bioinformatics* **36:** 496–503. doi:10.1093/bioinformatics/btz562

Zhou J, Theesfeld CL, Yao K, Chen KM, Wong AK, Troyanskaya OG. 2018. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nat Genet* **50:** 1171–1179. doi:10.1038/s41588-018-0160-6

Received June 7, 2021; accepted in revised form April 5, 2022.