Integrating long-range regulatory interactions to predict gene expression using graph convolutional neural networks

Jeremy Bigness[1,2,4], Xavier Loinaz[2], Shalin Patel[3], Erica Larschan[1,4], and Ritambhara Singh[1,2]

[1]*Center for Computational Molecular Biology, Brown University*
[2]*Department of Computer Science, Brown University*
[3]*Division of Applied Mathematics, Brown University*
[4]*Department of Molecular Biology, Cell Biology and Biochemistry, Brown University*

Abstract

Long-range spatial interactions among genomic regions are critical for regulating gene expression and their disruption has been associated with a host of diseases. However, when modeling the effects of regulatory factors on gene expression, most deep learning models either neglect long-range interactions or fail to capture the inherent 3D structure of the underlying biological system. This prevents the field from obtaining a more comprehensive understanding of gene regulation and from fully leveraging the structural information present in the data sets. Here, we propose a graph convolutional neural network (GCNN) framework to integrate measurements probing spatial genomic organization and measurements of local regulatory factors, specifically histone modifications, to predict gene expression. This formulation enables the model to incorporate crucial information about long-range interactions via a natural encoding of spatial interaction relationships into a graph representation. Furthermore, we show that our model is interpretable in terms of the observed biological regulatory factors, highlighting both the histone modifications and the interacting genomic regions that contribute to a gene's predicted expression. We apply our GCNN model to datasets for GM12878 (lymphoblastoid) and K562 (myelogenous leukemia) cell lines and demonstrate its state-of-the-art prediction performance. We also obtain importance scores corresponding to the histone mark features and interacting regions for some exemplar genes and validate them with evidence from the literature. Our model presents a novel setup for predicting gene expression by integrating multimodal datasets.

1 Introduction

Gene regulation determines the fate of every cell and its disruption leads to diverse diseases ranging from cancer to neurodegeneration [1–3]. Although specialized cell types – from neurons to cardiac cells – exhibit different gene expression patterns, the information encoded by the linear sequence of DNA remains virtually the same in all non-reproductive cells of the body [4, 5]. Therefore, the observed differences in cell type must be encoded by elements extrinsic to sequence, commonly referred to as epigenetic factors [6]. Epigenetic factors found in the local neighborhood of a gene typically include histone marks (also known as histone modifications). These marks are naturally occurring chemical additions to histone proteins that control how tightly the DNA strands are wound around the proteins, and the recruitment or occlusion of transcription factors [7]. However, in recent years, the focus of attention in genomics has shifted increasingly to the study of long-range epigenetic regulatory interactions that result from the three-dimensional organization of the genome [8, 9]. Although sometimes separated by hundreds of kilobases along the DNA strand, DNA sequences known as enhancers increase the expression of their target genes by being brought into close spatial proximity by proteins such as CCCTC binding factor (CTCF) and cohesin [2, 9]. These enhancer sequences have been estimated to account for about 16% of the human genome [10].

Many diseases have been attributed to the disruption of the long-range transcriptional regulation of genes. For example, some early studies showed that chromosomal rearrangements disrupted the region downstream of the PAX6 transcription unit causing Aniridia (absence of the iris) and related eye anomalies [11–14]. These rearrangements were located as far as 125 kilo-basepairs (kbp) beyond the final exon of the gene [11, 13, 15]. Thus, chromosomal rearrangement can not only affect a gene directly but can indirectly disrupt a gene located far away by disrupting its regulatory (e.g., enhancer-promoter) interactions. This observation indicates that while local regulation of genes is informative, studying long-range gene regulation is

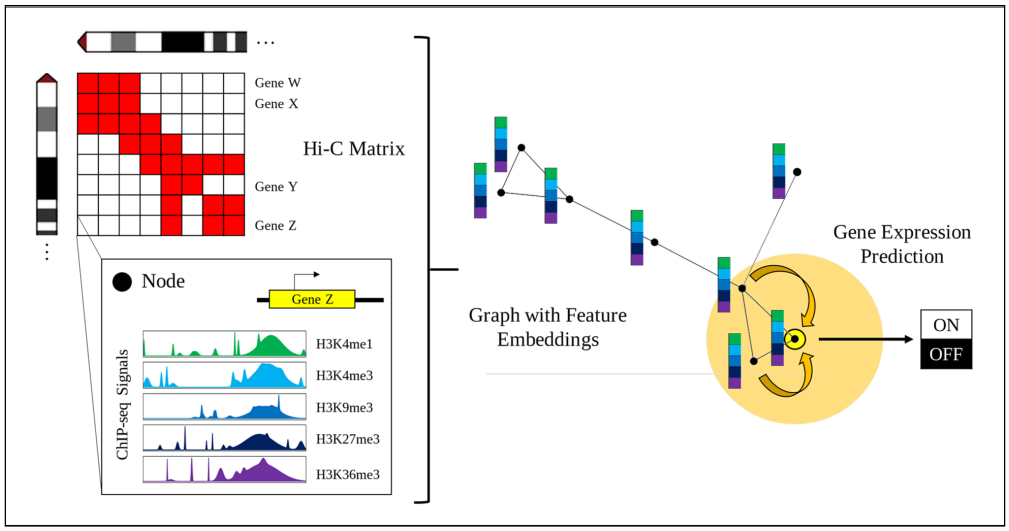

Figure 1: **Overview of the proposed GCNN model.** Our framework integrates information from both histone modification signals and long-range spatial interactions to predict and understand gene expression. Inputs to the model include Hi-C maps for each chromosome as well as ChIP-seq levels for five core histone marks (H3K4me1, H3K4me3, H3K9me3, H3K36me3, and H3K27me3). We subdivide the Hi-C maps uniformly into genomic regions of 10 kilo-basepair (kb) intervals. The row (or equivalently, column) corresponding to each genomic region is represented as a node in the graph. Each node's initial feature embedding is a vector of length five, as determined by the average ChIP-seq signals for the histone marks in each region. Edges between nodes are added based on the Hi-C interaction frequency of the corresponding genomic regions. For nodes with regions containing a gene, the model performs repeated graph convolutions over the neighboring nodes to yield a binarized class prediction of whether the gene is expressed or not.

critical to understanding cell development and disease. Experimentally testing for all possible combinations of long-range and short-range regulatory interactions that could control gene expression for $\sim 20,000$ genes is infeasible, given the vast size of the search space. Therefore, computational and data-driven approaches are necessary to efficiently search this space and reduce the number of testable hypotheses due to the sheer scope of the problem.

Various machine learning models investigating gene regulation have been introduced to attempt to address this need. Some examples include support vector machines [16], random forests [17], and elaborations on classical regression techniques [18–20]). However, in all these cases, the researcher must directly select the relevant features that lead to the target predictions. One way to overcome this limitation is to use a deep learning framework, which refers to a multi-layered neural network. Among their many advantages, deep neural networks perform automatic feature extraction by efficiently exploring feature space and then finding nonlinear transformations of the weighted averages of those features. This formulation is especially relevant to complex biological systems since they are inherently nonlinear. Hence, nonlinear feature transformations are often more strongly predictive of the output than the raw feature values [21].

Recently, deep learning frameworks have been applied to predict gene expression from histone modifications, and their empirical performance has often exceeded the machine learning methods referenced above. Singh *et al.* [22] introduced DeepChrome, which used a convolutional neural network (CNN) to aggregate five types of histone mark ChIP-seq signals in a $10,000$ bp region around the transcription start site (TSS) of each gene. This same data setup was then used in AttentiveChrome [23], in which the authors implemented a long short-term memory neural network (LSTM) for each histone mark. They then applied an attention-layer to focus on the most relevant features selectively. Agarwal *et al.* [24] employed a CNN

Computational Study	Local Epigenetic Factors	Feature-level Interpretation	Long-Range Interactions	Underlying Spatial Structure	Edge-level Interpretation
CNN [22, 24]	X	X			
Attention + LSTM [23]	X	X			
Attention + CNN + LSTM [25]	X	X			
Densely connected CNN [28]	X	X	X		
GCNN (Our model)	**X**	**X**	**X**	**X**	**X**

Table 1: **Comparison of the properties of previous deep learning models predicting gene expression from histone modifications with our GCNN framework.** The proposed method incorporates 3D DNA information, capturing the underlying spatial structure, and highlights both the important node-level (histone modifications) and edge-level (long-range interactions) features.

framework that operated on the promoter sequences of each gene and other annotated features associated with mRNA decay to predict steady-state mRNA levels. Kang *et al.* [25] used histone marks, DNA methylation state, and transcription factor levels as inputs to a model comprised of several integrated networks, including CNNs, bi-directional LSTMs, and attentive layers. The studies listed above incorporate nonlinear combinatorial interactions among features at the local level. However, they do not use spatial information about long-range regulatory interactions known to play a critical role in differentiation and disease [1–3].

Modeling these long-range interactions from regulatory signals is a challenging task due to two major reasons. First, we cannot confidently pick an input size for the genomic regions as regulatory elements can control gene expression from various distances. Second, inputting a large region will introduce sparsity and noise into the data, making the learning task difficult. A potential solution to this problem is to incorporate information from long-range interaction networks captured from experiments like Hi-ChIP [26] and Hi-C [27]. These assays use high-throughput sequencing to measure 3D genomic structure, where each read pair corresponds to an observed 3D contact between two genomic loci. While Hi-ChIP focuses only on spatial interactions mediated by a specific protein, Hi-C captures the genome-wide global interactions of the genomic regions. Recently, Zeng *et al.* [28] combined a CNN encoding promoter sequences with a dense multi-layer perceptron network using Hi-ChIP datasets to predict gene expression. The authors evaluated the relative contributions of the promoter sequence and promoter-enhancer submodules with respect to the model's overall performance. While this method incorporates the long-range interaction information, its use of HiChIP experiments narrows this information to spatial interactions mediated by H3K27ac and YY1. Furthermore, CNN models only capture the local topological patterns instead of modeling the underlying spatial structure of the data.

We propose integrating 3D genome organization data with the existing histone mark signals using a graph-based deep neural network to predict gene expression. Figure 1 summarizes our overall approach, which is guided by the following desiderata. First, unlike previous methods, our model incorporates the genome-wide global interaction information using the Hi-C data. Second, we use a graph convolutional neural network (GCNN) to capture the underlying spatial structure. GCNNs are particularly well-suited to representing spatial relationships, as the Hi-C map can be represented as an adjacency matrix of an undirected graph $G \in \{V, E\}$. Here, V nodes represent the genomic regions and E edges represent their interactions. Finally, we perform an interpretation of the GCNN model that quantifies the relative importance of the underlying biological regulatory factors driving the model's predictions for each gene. We use the GNNExplainer method [29] that highlights not only the important node features (histone modifications) but also the important edges (long-range interactions) that contribute to determining a particular gene's predicted expression. In this paper, we apply our method to two cell lines – GM12878 (lymphoblastoid) and K562 (myelogenous leukemia) – and demonstrate that our model outperforms state-of-the-art deep learning models. More importantly, we show that our framework allows biologists to tease apart the cumulative effects of different regulatory mechanisms at the genic level. Table 1 places the proposed framework among

the state-of-the-art deep learning models and lists the properties of each model. To summarize: (1) Our GCNN model performs gene expression predictions using the histone modification signals and provides feature-level interpretation on histone modifications similar to previous methods; (2) it innovates by incorporating 3D genomic information, capturing the underlying spatial structure, and highlighting the important edge-level features representing the long-range interactions.

2 Methods

Graph convolutional neural networks (GCNNs) Graph convolutional neural networks (GCNNs) are a generalization of convolutional neural networks (CNNs) to graph-based relational data that is not natively structured in Euclidean space [30]. Due to the expressive power of graphs, GCNNs have been applied across a wide variety of domains, including traffic flow prediction [31], recommender systems [32], and social networks [33]. The prevalence of graph-based datasets in biology has made these models a popular choice for tasks like modeling protein-protein interactions [34], stem cell differentiation [35], and chemical reactivity for drug discovery [36].

Although there are many variations of GCNNs [37, 38], we use the GraphSAGE formulation [39]. We use this method for its relative simplicity and its capacity to learn generalizable, inductive representations not limited to a specific graph. The input to the model is represented as a graph $G \in \{V, E\}$, with nodes V and edges E, and a corresponding adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ [30], where N is the number of nodes. For each node v, there is also an associated feature vector $\boldsymbol{x_v}$. The goal of the network is to learn a state embedding $\boldsymbol{h}_v^K \in \mathbb{R}^d$ for v, which is obtained by aggregating information over v's neighborhood K times. Here, d is the dimension of the embedding vector. This state embedding is then fed through a fully-connected network to produce an output $\hat{y_v}$, which can then be applied to downstream classification or regression tasks.

Within this framework, the first step is to initialize each node with its input features. In our case, the feature vector $\boldsymbol{x}_v \in \mathbb{R}^m$ is obtained from the ChIP-seq signals corresponding to the five ($m = 5$) core histone marks (H3K4me1, H3K4me3, H3K9me3, H3K36me3, and H3K27me3) in our dataset:

$$\boldsymbol{h}_v^0 = \boldsymbol{x}_v \tag{1}$$

Next, to transition from the $(k-1)^{th}$ layer to the k^{th} hidden layer in the network for node v, we apply an aggregation function to the neighborhood of each node. This aggregation function is analogous to a convolution operation over regularly structured Euclidean data such as images. A standard convolution function operates over a grid and represents a pixel as a weighted aggregation of its neighboring pixels. Similarly, a graph convolution performs this operation over the neighbors of a node in a graph. In our case, the aggregation function calculates the mean of the neighboring node features:

$$\boldsymbol{h}_{\mathcal{N}(v)}^k = \sum_{u \in \mathcal{N}(v)} \frac{\boldsymbol{h}_u^{k-1}}{|\mathcal{N}(v)|} \tag{2}$$

Here, $\mathcal{N}(v)$ represents the adjacency set of node v. To retain information from the original embedding, we update the node's embedding by concatenating the aggregation with the previous layer's representation. Next, as done in regular convolution, we take the matrix product of this concatenated representation with a learnable weight matrix to complete the weighted aggregation step. Finally, we apply a non-linear activation function, such as ReLU, to capture the higher-order non-linear interactions among the features:

$$\boldsymbol{h}_v^k = \sigma\left(\mathbf{W}_k \left[\boldsymbol{h}_{\mathcal{N}(v)}^k || \boldsymbol{h}_v^{k-1}\right]\right), \forall k \in \{1, ..., K\} \tag{3}$$

Here, $||$ represents concatenation, σ is a non-linear activation function, and $\mathbf{W}_k$ is a learnable weight parameter. After this step, each node is assigned a new embedding. After K iterations, the node embedding encodes information from the neighbors that are K-hops away from that node:

$$\boldsymbol{z}_v = \boldsymbol{h}_v^K \tag{4}$$

Here, $\boldsymbol{z}_v$ is the final node embedding after K iterations. For our model, we formulate gene expression prediction as a binary classification task with classes $c \in \{0, 1\}$, corresponding to whether the gene is either off/inactive ($c = 0$) or on/active ($c = 1$). Thereafter, we feed the learned embedding $\boldsymbol{z}_v$ into a fully connected network and output a prediction $\hat{y_v}$ for each target node using a $Softmax$ layer to compute probabilities for each class c. Finally, we use the true binarized gene expression value $y_v \in \mathbb{R}^{\{0,1\}}$ associated with the gene in each node to train the model using the negative log likelihood (NLL) loss. The overall model architecture is summarized in Figure 2.

Interpretation of GCNN model Neural networks have sometimes been criticized for being "black box" models, such that no insight is provided into how the model arrives at its predictions. Most graph-based interpretability approaches either approximate models with simpler models whose decisions can be used for explanations [40], or they use attention mechanism to identify relevant features in the input that guide a particular prediction [41]. In general, these methods, along with gradient-based approaches [42, 43] or DeepLift [44], focus on the explanation of important node features and do not incorporate structural information of the graph. However, a recent method called *Graph Neural Net Explainer* (or GNNExplainer) [29], given a trained GCN, can identify a small subgraph as well as a small subset of features that are crucial for a particular prediction. The authors demonstrate its interpretation capabilities on simulated graphs, MUTAG [45], and graphs obtained from Reddit discussion forums.

We apply GNNExplainer to our GCNN model to ensure that our model makes interpretable predictions based on the underlying biological features. GNNExplainer maximizes the mutual information between the probability distribution of the model's class predictions over all nodes and the probability distribution of the class predictions for a particular node conditioned on some fractional masked subgraph of neighboring nodes and features. Subject to regularization constraints, GNNExplainer jointly optimizes the fractional node and feature masks, determining the extent to which each element informs the prediction for a particular node.

Specifically, given a node v, the goal is to learn a subgraph $G_s \subseteq G$ and a feature mask $X_s = \{x_j \mid v_j \in G_s\}$ that are the most important in driving the full model's prediction of $\hat{y}_v$. To achieve this objective, the algorithm learns a mask that maximizes the mutual information (MI) between the original model and the masked model. Mathematically, this objective function is as follows:

$$\max_{G_s} MI(Y, (G_s, X_s)) = H(Y) - H(Y \mid G_s, X_s) \tag{5}$$

where H is the entropy of a distribution. Since this is computationally intractable with an exponential number of graph masks, GNNExplainer optimizes the following quantity using gradient descent:

$$\min_{M,N} -\sum_{c=1}^{C} \mathbb{1}_{\{y=c\}} \log(P_\phi(Y = y | G = A_c \odot \sigma(M), X = X_c \odot \sigma(N)) \tag{6}$$

where c represents the class, A_c represents the adjacency matrix, M represents the subgraph mask, and N represents the feature mask. The importance scores of the nodes and features are obtained by applying the sigmoid function to the subgraph and feature masks, respectively. Finally, the element-wise entropies of the masks are calculated and inserted as regularization terms into the loss function. Therefore, in the context of our model, GNNExplainer learns which genomic regions (via the subgraph mask) and which features (via the feature mask) are most important in driving the model's predictions.

3 Experimental Setup

Overview of model inputs Our GCNN model requires the following information: (1) Interactions between the genomic regions (Hi-C contact maps); (2) Histone mark signals representing the regulatory signals

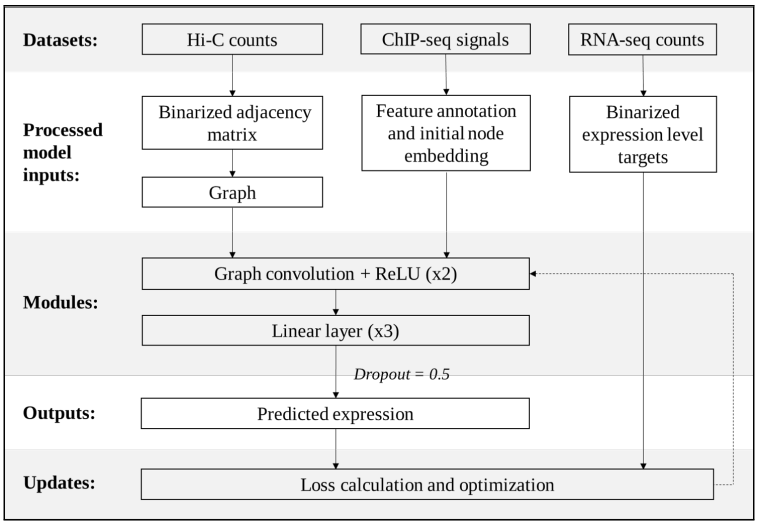

Figure 2: **Overview of the GCNN model architecture.** The datasets used in our model are Hi-C counts, ChIP-seq signals, and RNA-seq counts. A binarized adjacency matrix is produced from the Hi-C maps and converted into a graph. The nodes in the graph are annotated with features from the ChIP-seq datasets. Two graph convolutions, each followed by ReLU, are performed. The output from this module is fed into three dense layers (with a dropout of 0.5) to yield a binarized predicted expression level.

(ChIP-seq measurements); (3) Expression levels for each gene (binarized RNA-seq counts). For each gene in a particular region, the first two datasets are the inputs into our proposed model, whereas gene expression is the predicted target. We model the task as a classification problem by binarizing gene expression as 0 (off) or 1 (on) using median gene expression values. Our reasons for adopting this approach are as follows: (1) To demonstrate proof of principle that the task can be modeled as a graph convolutional neural network with a built-in interpretive mechanism; (2) To ensure consistency with previous studies that also binarize gene expression so that our results can be easily compared [22, 23, 28].

Data preprocessing We focused on two Tier 1 human cell lines as designated by ENCODE: (1) GM12878, a lymphoblastoid cell line with a normal karyotype, and (2) K562, a myelogenous leukemia cell line. For each of these cell lines, we accessed RNA-seq expression datasets as well as ChIP-Seq signal datasets for five uniformly profiled histone marks from the REMC repositories [46]. These histone marks include the following: (1) H3K4me1, associated with enhancer regions; (2) H3K4me3, associated with promoter regions; (3) H3K9me3, associated with heterochromatin; (4) H3K36me3, associated with actively transcribed regions; and (5) H3K27me3, associated with polycomb repression. We chose these marks because of the wide availability of the relevant data as well as for ease of comparison with previous studies [22, 23, 28].

For chromosome capture data, we used previously published Hi-C maps at 10 kilobase (kb) resolution for all 22 autosomal chromosomes [47]. We obtained an N x N symmetric matrix, where each row or column corresponds to a 10 kb chromosomal region. Therefore, each bin coordinate (*row, column*) corresponds to the interaction frequency between two respective genomic regions. We applied VC-normalization on the Hi-C maps. In addition, because chromosomal regions located closer together will contact each other more frequently than regions located farther away simply due to chance (rather than due to biologically significant effects), we made an additional adjustment for this background effect. Following Sobhy *et al.* [48], we took the medians of the Hi-C counts for all pairs of interacting regions located the same distance away and used this as a proxy for the background. We subtracted the appropriate median from each Hi-C bin and discarded negative values.

Graph construction and data integration We represented each genomic region with a node and connected edges between it and the nodes corresponding to its neighbors (bins with non-zero entries in the adjacency matrix) to construct the graph. Due to the large size of the Hi-C graph, we subsampled neighbors to form a subgraph for each node that we fed into the GCNN model. While there are methods to perform subsampling on large graphs using a random node selection approach (e.g. [49]), we used a simple strategy of selecting the top j neighbors with the highest Hi-C interaction frequency values. We empirically selected the value $j = 10$ for the number of neighbors. Smaller number of neighbors (i.e., $j = 5$) resulted in decreased performance, while selecting more neighbors proved prohibitive due to memory constraints.

To integrate the Hi-C datasets together with the RNA-seq and ChIP-seq datasets, we obtained the average ChIP-seq signal for each of the five core histone marks over the chromosomal region corresponding to each node. In this way, a feature vector of length five was associated with each node. For the RNA-seq data, we took each gene's transcriptional start site (TSS) and assigned it to the node corresponding to the chromosomal region in which the TSS is located. Since we formulated the task as a classification problem, we binarized the gene expression by taking the median as a cutoff, consistent with previous studies [22, 23, 28]. If multiple genes were assigned to the same node, we took the mode of the binarized expression level. If no genes were assigned to a node, we set the expression level to a value > 1. During the training phase, we applied a mask so that the model only made predictions on nodes with expression values of zero or one. Finally, we assigned 70% of the nodes to the training set, 15% to the the validation set, and 15% to the testing set.

Model architecture and training The final model architecture is represented in Figure 2. Here, the first layer of the model performs a graph convolution on the initial feature embeddings with an output embedding size of 256, followed by application of ReLU, a non-linear activation function. The second layer of the model performs another graph convolution with the same embedding size of 256 on the transformed representations, again followed by application of ReLU. Next, the output is fed into three successive linear layers of sizes 256, 256, and 2, respectively. A regularization step is performed by using a dropout layer after the first linear layer with probability 0.5. Applying $Softmax$ function to the final output yields the probabilities assigned to each of the two classes. These probabilities are fed into a negative log likelihood (NLL) loss function, which is then minimized via ADAM, a stochastic gradient descent algorithm [50]. We used the PyTorch Geometric package [51] to implement our GCNN framework.

Hyperparameter tuning We recorded the loss curves for the training and validation sets over 1000 epochs, by which time the model began to overfit. We performed hyperparameter tuning over the following grid of values and selected the optimal hyperparameters that gave the best AUC score performance on the validation set: number of graph convolution layers: $\{1, 2\}$, size of embedding layers: $\{16, 128, 256, 384\}$, and number of fully connected layers: $\{1, 2, 3\}$.

Baselines models We compared our GCNN model with the following deep learning baselines:

- **Multi-layer perceptron (MLP)**: A simple MLP comprised of three fully-connected layers. In this framework, the model predictions for each node do not incorporate feature information from the node's neighbors.
- **DeepChrome:** A convolutional neural network developed by Singh *et al.* [22]. This model takes a region of +/- 5000 bp about the TSS of each gene and divides it into 100 bins. Each bin is associated with five channels, which correspond to the ChIP-seq signals of the same five core histone marks in the present study. A standard convolution is applied to the channels, followed by linear layers.
- **AttentiveChrome:** A long-short term memory (LSTM) network with attention layers achieving state-of-the-art performance and developed by Singh *et al.* [23]. This model takes a region of +/- 5000 bp about the TSS of each gene and divides it into 100 bins. Each of these genes is feature annotated with the ChIP-seq signals of the five core histone marks used in the present study. An LSTM is then used to encode the bin features. From there, an attention layer is applied to the bins for each region, a

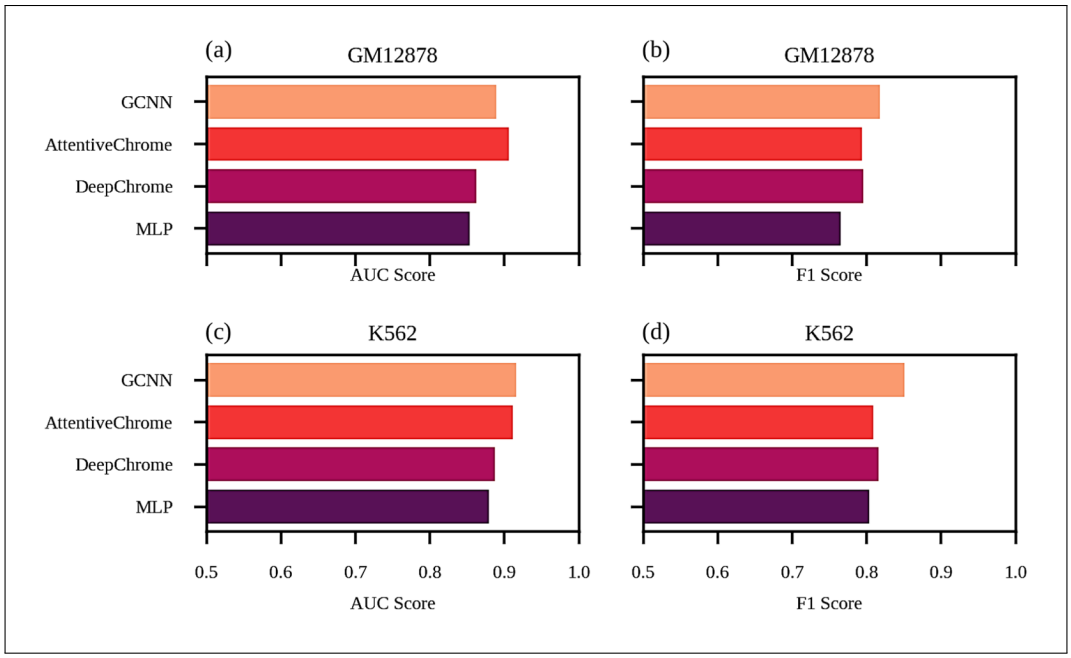

Figure 3: **Comparison of AUC and F1 scores for all models.** GCNN model gives state-of-the-art performance for classifying genes as on/active or off/inactive. (a) For GM12878, the AUC score of our model is higher than MLP and DeepChrome baselines and comparable to AttentiveChrome. (c) For K562, the GCNN model achieves the highest AUC score outperforming all the baselines. (b,d) When using F1 scores as the evaluation metric, the GCNN model outperforms all the baseline models for both cell lines.

second attention layer is applied to the histone mark features, and finally, the output is fed through a linear layer.

Note that while the DeepChrome and AttentiveChrome baselines divide the genomic regions into smaller 100-bp bins, our GCNN framework and MLP baseline average the histone modification signals over the entire 10 kb region. Therefore, the difference between the MLP and the GCNN performances can be viewed as a proxy for the importance of including information from long-range, regulatory interactions for similarly processed inputs. We report the classification performance of the GCNN model as well as the baseline models by using the area under the ROC curve (AUC) and F1 scores.

4 Results

GCNN model gives state-of-the-art performance for gene classification We compare the classification performance of the GCNN with the baseline models for two cell lines: GM12878 (lymphoblastoid cells) and K562 (myelogenous leukemia cells) in Figure 3. For GM12878, the AUC score of our model outperforms that of the multi-layer perceptron (MLP) and DeepChrome baselines and is comparable to AttentiveChrome. For K562, the GCNN model achieves the highest AUC score, outperforming all baselines. Additionally, our AUC score for K562 (0.916) is comparable to that reported by Zeng *et al.* [28] (0.91). Their model integrated histone modification signals with spatial information from Hi-ChIP data. We could not compare scores for GM12878 as they do not provide Hi-ChIP data for the cell line to run their model. Finally, when using F1 scores as the evaluation metric, we observe that our GCNN model outperforms all the baseline models for both cell lines. These results suggest that including spatial information can improve gene expression prediction performance over methods solely using histone modifications as input.

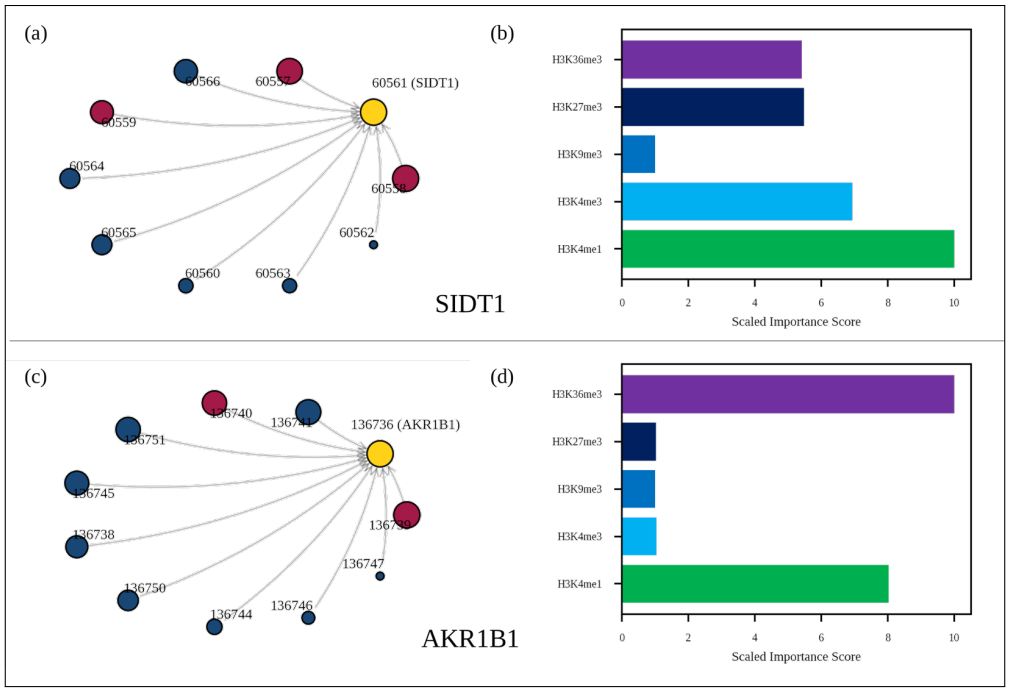

Figure 4: **Model explanations for genes SIDT1 and AKR1B1.** For SIDT1: (a) The subgraph of neighbor nodes for SIDT1, designated as node 136736 (yellow circle), is displayed. The size of each neighbor node correlates with its predictive importance as determined by GNNExplainer. Nodes in red denote regions corresponding to known enhancer regions regulating SIDT1 [52](note that multiple interacting fragments can be assigned to each node, see Supplementary Table S1). All other nodes are displayed in blue. Nodes with importance scores corresponding to outliers have been removed for clarity. (b) The scaled feature importance scores for each of the five core histone marks used in this study are displayed in the graph. For AKR1B1 (displayed in a manner similar to SIDT1): (c) The subgraph of neighbor nodes and their relative importance scores, and (d) the relative feature importance scores for each of the five core histone marks are displayed.

Interpretation of GCNN model highlights long-range interactions and histone modification profiles

To determine if our model makes predictions based on reasonable biological principles, we apply GNNExplainer to our model predictions. For GM12878, a lymphoblastoid cell line, we selected two genes, SIDT1 and AKR1B1, two of the most highly expressed genes in our dataset. These genes have also been shown to be controlled by several long-range promoter-enhancer interactions [52].

SIDT1 gene encodes a transmembrane dsRNA-gated channel protein and is part of a larger family of proteins necessary for systemic RNA interference [53, 54]. This gene has also been implicated in chemoresistance to the drug gemcitabine in adenocarcinoma cells [53]. It is located on chromosome 3: 113532296-113629579 bp and is known to be regulated by at least three chromosomal regions [52, 54]. In Figure 4(a), we show that for SIDT1, the model makes use of all three genomic regions known to have regulatory effects by assigning high importance scores to those nodes (indicated by the size of the node). The regions associated with each node are provided in Supplementary Table S1. In addition, in Figure 4(b), we plot the importance scores assigned to the histone marks (node features) that are most important in driving the model's predictions. From the graph, it is apparent that H3K4me1 and H3K4me3 are the two most important features in determining the model's prediction. This histone mark profile has been associated with flanking TSS sites in expressed genes [46, 55].

AKR1B1 gene encodes an enzyme that belongs to the aldo-keto reductase family. It catalyzes the reduction of aldehydes and ketones and is involved in glucose metabolism [54]. In addition, it has also been identified as a key player in complications associated with diabetes [54, 56]. It is located on chromosome 7: 134442350-134459239 bp and is known to be regulated by at least two chromosomal regions [52]. As seen in Figure 4(c),the model strongly bases its predictions for AKRB1 on both of the regions known to have regulatory effects (location information in Supplementary Table S1). In Figure 4(d), we show that H3K36me3 and H3K4me1 are the two histone marks with the highest scaled importance scores. This chromatin state signature is correlated with genic enhancers of highly expressed genes [46].

To confirm that the node importance scores obtained from GNNExplainer do not merely reflect the relative magnitudes of the Hi-C counts or the distances between genomic regions, we investigated the relationships among the Hi-C counts, genomic distances, and scaled importance scores for both genes (Supplementary Figures S1 and S2). We observe that the scaled importance scores do not correspond to the Hi-C counts or the pairwise genomic distances. For example, for SIDT1, the three experimentally validated interacting nodes achieve the highest importance scores ($10, 9.55$, and 7.73). However, they do not correspond to the regions with the highest Hi-C counts ($154.78, 412.53$, and 170.55 for each of the three known regulatory regions while the highest count is 602.84). In addition, although they are close to the SIDT1 gene region (40, 20, and 30 kbp away), there are other nodes at the same or closer distances that do not have promoter-enhancer interactions. Therefore, we show that by modeling not only the histone modifications but also the spatial configuration of the genome, the GCNN model infers connections that could serve as important hypothesis driving observations for gene regulatory experiments.

5 Discussion

We present a graph-based deep learning model that integrates both local and long-range epigenetic data using a graph convolutional neural network framework to predict gene expression. We demonstrate its state-of-the-art performance for the gene expression prediction task, outperforming the baselines on the GM12878 and K562 cell lines. We also determine the relative contributions of histone modifications and long-range interactions for two genes, showing that our model recapitulates known experimental results in a biologically interpretable manner.

With respect to future work, we plan on applying our framework on additional cell lines as high-quality Hi-C data sets become available. Incorporating other features, such as promoter sequence, would also be natural extensions. Another useful modification would be to incorporate weights corresponding to each node's Hi-C interaction frequency. One avenue of particular importance would be to develop more robust methods for interpreting GCNNs. For example, while the GNNExplainer model is a theoretically sound framework and yields an unbiased estimator for the importance scores of the subgraph nodes and features, there is variation in the interpretation scores generated over multiple runs. Furthermore, with larger GCNNs, the optimization function utilized in GNNExplainer is challenging to minimize in practice. For some iterations, the importance scores converge with little differentiation and the method fails to arrive at a compact representation. This may be due to the relatively small penalties the method applies with respect to constraining the optimal size of the mask and the entropy of the distribution. We plan to address this issue in the future by implementing more robust forms of regularization.

In summary, our model demonstrates proof-of-principle for using GCNNs to predict gene expression using both local epigenetic features and long-range spatial interactions. Interpretation of this model allows us to pose plausible biological explanations of the key regulatory factors driving gene expression. Understanding how gene expression is regulated through various mechanisms is critical to advancing fundamental science and discovering therapeutic interventions for diseases associated with gene misregulation.

References

[1] Peter Hugo Lodewijk Krijger and Wouter de Laat. Regulation of disease-associated gene expression in the 3d genome. 17(12):771–782.

[2] Stefan Schoenfelder and Peter Fraser. Long-range enhancer–promoter contacts in gene expression control. 20(8):437–455.

[3] Hui Zheng and Wei Xie. The role of 3d genome organization in development and cell differentiation. 20(9):535–550.

[4] Eric J. Richards. Inherited epigenetic variation — revisiting soft inheritance. 7(5):395–401.

[5] Hiroyuki Sasaki and Yasuhisa Matsui. Epigenetic events in mammalian germ-cell development: reprogramming and beyond. 9(2):129–140.

[6] Giacomo Cavalli and Edith Heard. Advances in epigenetics link genetics to the environment and disease. 571(7766):489–499.

[7] Tony Kouzarides. Chromatin modifications and their function. 128(4):693–705.

[8] Rieke Kempfer and Ana Pombo. Methods for mapping 3d chromosome architecture. 21(4):207–226.

[9] M. Jordan Rowley and Victor G. Corces. Organizational principles of 3d genome architecture. 19(12):789–800.

[10] Molly Gasperini, Jacob M. Tome, and Jay Shendure. Towards a comprehensive catalogue of validated and target-linked human enhancers. 21(5):292–310.

[11] Judy Fantes, Bert Redeker, Matthew Breen, Shelagh Boyle, John Brown, Judy Fletcher, Sinead Jones, Wendy Bickmore, Yoshimitsu Fukushima, Marcel Mannens, et al. Aniridia-associated cytogenetic rearrangements suggest that a position effect may cause the mutant phenotype. *Human molecular genetics*, 4(3):415–422, 1995.

[12] James D Lauderdale, Jonathan S Wilensky, Edward R Oliver, David S Walton, and Tom Glaser. 3 deletions cause aniridia by preventing pax6 gene expression. *Proceedings of the National Academy of Sciences*, 97(25):13755–13759, 2000.

[13] Dirk A Kleinjan, Anne Seawright, Andreas Schedl, Roy A Quinlan, Sarah Danes, and Veronica van Heyningen. Aniridia-associated translocations, dnase hypersensitivity, sequence comparison and transgenic analysis redefine the functional domain of pax6. *Human molecular genetics*, 10(19):2049–2059, 2001.

[14] John A Crolla and Veronica van Heyningen. Frequent chromosome aberrations revealed by molecular cytogenetic studies in patients with aniridia. *The American Journal of Human Genetics*, 71(5):1138–1149, 2002.

[15] Dirk A Kleinjan, Anne Seawright, Greg Elgar, and Veronica van Heyningen. Characterization of a novel gene adjacent to pax6, revealing synteny conservation with functional significance. *Mammalian genome*, 13(2):102–107, 2002.

[16] Chao Cheng, Koon-Kiu Yan, Kevin Y Yip, Joel Rozowsky, Roger Alexander, Chong Shou, and Mark Gerstein. A statistical framework for modeling gene expression using chromatin features and application to modENCODE datasets. 12(2):R15.

[17] Xianjun Dong, Melissa C Greven, Anshul Kundaje, Sarah Djebali, James B Brown, Chao Cheng, Thomas R Gingeras, Mark Gerstein, Roderic Guigó, Ewan Birney, and Zhiping Weng. Modeling gene expression using chromatin features in various cellular contexts. 13(9):R53.

[18] R. Karlic, H.-R. Chung, J. Lasserre, K. Vlahovicek, and M. Vingron. Histone modification levels are predictive for gene expression. 107(7):2926–2931.

[19] Yue Li, Minggao Liang, and Zhaolei Zhang. Regression analysis of combined gene expression regulation in acute myeloid leukemia. 10(10):e1003908.

[20] Z. Ouyang, Q. Zhou, and W. H. Wong. ChIP-seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. 106(51):21521–21526.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. Adaptive computation and machine learning. The MIT Press.

[22] Ritambhara Singh, Jack Lanchantin, Gabriel Robins, and Yanjun Qi. Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648, 2016.

[23] Ritambhara Singh, Jack Lanchantin, Arshdeep Sekhon, and Yanjun Qi. Attend and predict: Understanding gene regulation by selective attention on chromatin. 30:6785–6795.

[24] Vikram Agarwal and Jay Shendure. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. 31(7):107663.

[25] Minji Kang, Sangseon Lee, Dohoon Lee, and Sun Kim. Learning cell-type-specific gene regulation mechanisms by multi-attention based deep learning with regulatory latent space. 11:869.

[26] Maxwell R Mumbach, Adam J Rubin, Ryan A Flynn, Chao Dai, Paul A Khavari, William J Greenleaf, and Howard Y Chang. Hichip: efficient and sensitive analysis of protein-directed genome architecture. *Nature methods*, 13(11):919–922, 2016.

[27] Nynke L Van Berkum, Erez Lieberman-Aiden, Louise Williams, Maxim Imakaev, Andreas Gnirke, Leonid A Mirny, Job Dekker, and Eric S Lander. Hi-c: a method to study the three-dimensional architecture of genomes. *JoVE (Journal of Visualized Experiments)*, (39):e1869, 2010.

[28] Wanwen Zeng, Yong Wang, and Rui Jiang. Integrating distal and proximal information to predict gene expression via a densely connected convolutional neural network. page btz562.

[29] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. 2019.

[30] Zhiyuan Liu and Jie Zhou. Introduction to graph neural networks. 14(2):1–127.

[31] Mingqi Lv, Zhaoxiong Hong, Ling Chen, Tieming Chen, Tiantian Zhu, and Shouling Ji. Temporal multi-graph convolutional network for traffic flow prediction. pages 1–12.

[32] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–668. ACM.

[33] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. DeepInf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2110–2119. ACM.

[34] Fang Yang, Kunjie Fan, Dandan Song, and Huakang Lin. Graph-based prediction of protein-protein interactions with attributed signed graph embedding. 21(1):323.

[35] Ioana Bica, Helena Andrés-Terré, Ana Cvejic, and Pietro Liò. Unsupervised generative and graph representation learning for modelling cell differentiation. 10(1):9790.

[36] Mengying Sun, Sendong Zhao, Coryandar Gilvary, Olivier Elemento, Jiayu Zhou, and Fei Wang. Graph convolutional networks for computational drug development and discovery. 21(3):919–935.

[37] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[38] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[39] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc.

[40] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[41] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[42] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[43] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

[44] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

[45] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797, 1991.

[46] Roadmap Epigenomics Consortium. Integrative analysis of 111 reference human epigenomes. 518(7539):317–330.

[47] Suhas S.P. Rao, Miriam H. Huntley, Neva C. Durand, Elena K. Stamenova, Ivan D. Bochkov, James T. Robinson, Adrian L. Sanborn, Ido Machol, Arina D. Omer, Eric S. Lander, and Erez Lieberman Aiden. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. 159(7):1665–1680.

[48] Haitham Sobhy, Rajendra Kumar, Jacob Lewerentz, Ludvig Lizana, and Per Stenberg. Highly interacting regions of the human genome are enriched with enhancers and bound by DNA repair proteins. 9(1):4577.

[49] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*, 2019.

[50] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[51] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[52] Inkyung Jung, Anthony Schmitt, Yarui Diao, Andrew J. Lee, Tristin Liu, Dongchan Yang, Catherine Tan, Junghyun Eom, Marilynn Chan, Sora Chee, Zachary Chiang, Changyoun Kim, Eliezer Masliah, Cathy L. Barr, Bin Li, Samantha Kuan, Dongsup Kim, and Bing Ren. A compendium of promoter-centered long-range chromatin interactions in the human genome. 51(10):1442–1449.

[53] Mohamed O. Elhassan, Jennifer Christie, and Mark S. Duxbury. *Homo sapiens* systemic RNA interference-defective-1 transmembrane family member 1 (SIDT1) protein mediates contact-dependent small RNA transfer and MicroRNA-21-driven chemoresistance. 287(8):5267–5277.

[54] National Center for Biotechnology Information National Library of Medicine (US). Entrez gene. `https://www.ncbi.nlm.nih.gov/gene/`, 1988-. Accessed: 2020-10-22.

[55] Jason Ernst and Manolis Kellis. Chromatin-state discovery and genome annotation with ChromHMM. 12(12):2478–2492.

[56] K. C. Donaghue, S. H. Margan, A. K. F. Chan, B. Holloway, M. Silink, T. Rangel, and B. Bennetts. The association of aldose reductase gene (AKR1b1) polymorphisms with diabetic neuropathy in adolescents. 22(10):1315–1320.

Supplementary Information

Gene	Interacting Fragment Coordinates	Node Identifier	Node Coordinates
SIDT1	chr3:113212739-113215893	60557	chr3:113209241-113219241
	chr3:113228501-113232053	60558	chr3:113219241-113229241
	chr3:113228501-113232053	60559	chr3:113229241-113239241
		60560	chr3:113239241-113249241
		60562	chr3:113259241-113269241
		60563	chr3:113269241-113279241
		60564	chr3:113279241-113289241
		60565	chr3:113289241-113299241
		60566	chr3:113299241-113309241
AKR1B1		136738	chr7:134273323-134283323
	chr7:134293046-134298798	136739	chr7:134283323-134293323
	chr7:134293046-134298798	136740	chr7:134293323-134303323
		136741	chr7:134303323-134313323
		136744	chr7:134333323-134343323
		136745	chr7:134343323-134353323
		136746	chr7:134353323-134363323
		136747	chr7:134363323-134373323
		136750	chr7:134393323-134403323
		136751	chr7:134403323-134413323

Table S1: Node coordinates for SIDT1 and AKR1B1. The second column lists the regulatory fragments that interact with each gene as detailed in Jung *et al.* [52]. The third and fourth columns are the node identifiers and chromosome coordinates for all of the gene's neighbor nodes, including both nodes that contain interacting fragments and those that do not.

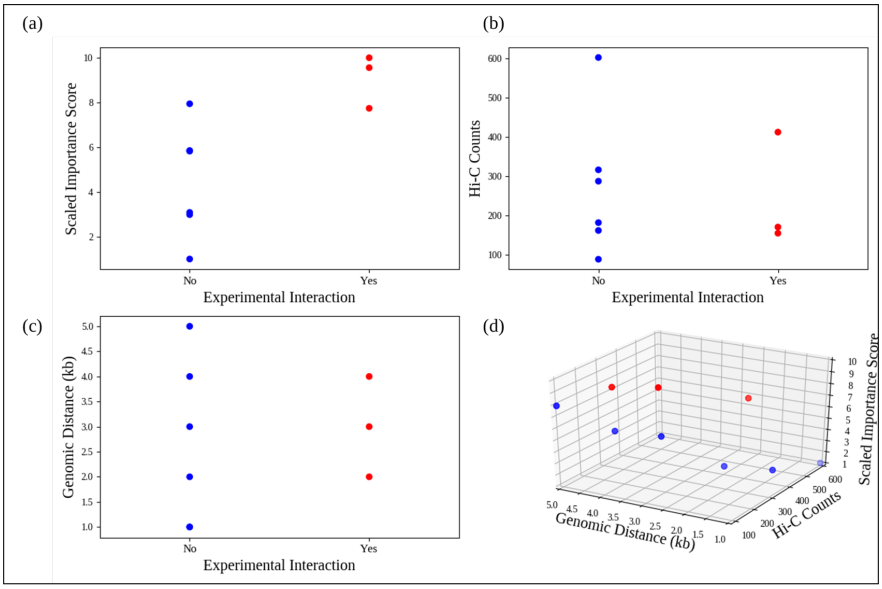

Figure S1: Relationships among scaled importance scores, genomic distances, and Hi-C counts for all SIDT1 neighbors. Nodes corresponding to experimentally validated interacting fragments are denoted in red and all others are denoted in blue. (a) Scaled importance score versus experimental interaction. Experimentally validated interacting fragments are ranked higher on average than non-interacting fragments. (b) Hi-C counts versus experimental interaction. Hi-C counts by themselves are not sufficient to explain the presence of experimentally validated interactions. (c) Genomic distance versus experimental interaction. Genomic distance does not correlate with experimentally validated interactions. (d) 3D plot displaying the relationships among scaled importance scores, genomic distances, and Hi-C counts.

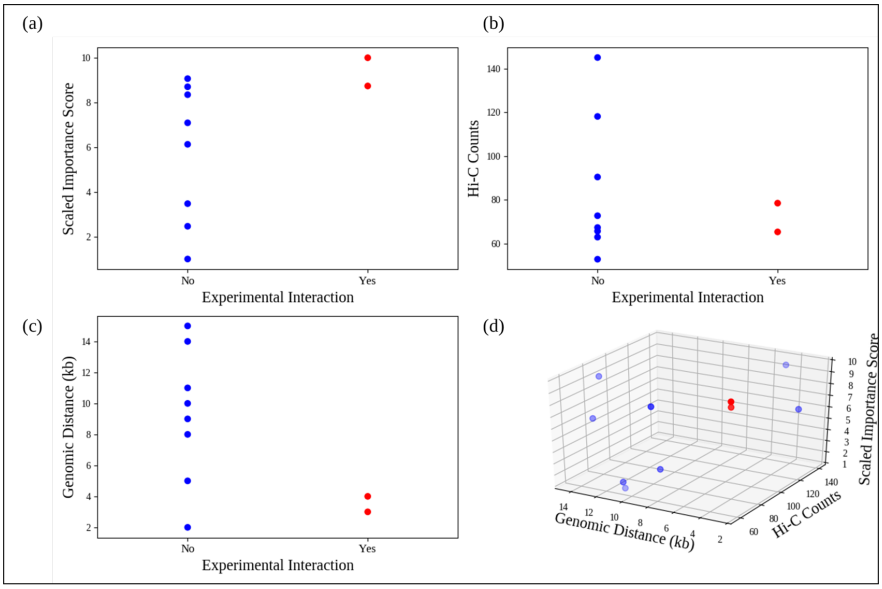

Figure S2: Relationships among scaled importance scores, genomic distances, and Hi-C counts for all AKR1B1 neighbors. Nodes corresponding to experimentally validated interacting fragments are denoted in red and all others are denoted in blue. (a) Scaled importance score versus experimental interaction. Experimentally validated interacting fragments are ranked higher on average than non-interacting fragments. (b) Hi-C counts versus experimental interaction. Hi-C counts by themselves are not sufficient to explain the presence of experimentally validated interactions. (c) Genomic distance versus experimental interaction. Genomic distance does not correlate with experimentally validated interactions. (d) 3D plot displaying the relationships among scaled importance scores, genomic distances, and Hi-C counts.