Computational Biology

A Self-Attention Model for Inferring Cooperativity between Regulatory Features

Fahad Ullah [1] and Asa Ben-Hur [1,*]

[1] Computer Science Department, Colorado State University, Fort Collins, 80523, USA.

[*] To whom correspondence should be addressed.

Abstract

Motivation: Deep learning has demonstrated its predictive power in modeling complex biological phenomena such as gene expression. The value of these models hinges not only on their accuracy, but also on the ability to extract biologically relevant information from the trained models. While there has been much recent work on developing feature attribution methods that provide the most important features for a given sequence, inferring cooperativity between regulatory elements, which is the hallmark of phenomena such as gene expression, remains an open problem.

Results: We present SATORI, a **S**elf-**AT**tenti**O**n based model to predict **R**egulatory element **I**nteractions. Our approach combines convolutional and recurrent layers with a self-attention mechanism that helps us capture a global view of the landscape of interactions between regulatory elements in a sequence. We evaluate our method on simulated data and three complex datasets: human TAL1-GATA1 transcription factor ChIP-Seq, DNase I Hypersensitive Sites (DHSs) in human promoters across 164 cell lines, and genome-wide DNase I-Seq and ATAC-Seq peaks across 36 arabidopsis samples. In each of the three experiments SATORI identified numerous statistically significant TF interactions, many of which have been previously reported. Our approach not only provides a global, biologically relevant set of interactions but, unlike existing methods, it does not require a computationally expensive postprocessing step.

Availability: The source code for SATORI is available at https://github.com/fahadahaf/SATORI.

Contact: asa@colostate.edu

1 Introduction

High-throughput sequencing techniques are producing an abundance of transcriptome and epigenetic datasets that can be used to generate genome-wide maps of different aspects of regulatory activity. The complexity and magnitude of these data has made deep neural networks an appealing choice for a variety of prediction tasks, including transcription factor (TF) binding prediction [Alipanahi *et al.*, 2015, Hassanzadeh and Wang, 2016, Zhou and Troyanskaya, 2015, Quang and Xie, 2016, Qin and Feng, 2017], chromatin accessibility analysis [Kelley *et al.*, 2016, Banovich *et al.*, 2018, Angermueller *et al.*, 2017], prediction of chromatin structure and its modifications [Koh *et al.*, 2017, Schreiber *et al.*, 2018], and identification of RNA-binding sites [Pan and Shen, 2017, Zhang *et al.*, 2015]. Besides providing improvement in accuracy over traditional machine learning models, deep learning methods typically require less feature engineering, and can learn directly from sequence data. Moreover, because of their complexity and depth, these models can capture higher-order non-linear feature interactions that better explain the underlying regulatory phenomenon.

The discovery that TFs work in tandem to regulate the expression of their targets [Wasserman and Fickett, 1998] has sparked the development of a variety of computational methods for predicting cooperativity among TFs and other regulatory proteins by looking at regulatory element co-occurrences [Hannenhalli and Levy, 2002, GuhaThakurta and Stormo, 2001, Pilpel *et al.*, 2001, Sudarsanam *et al.*, 2002, Roca and Franceschi, 2008, Safe, 2015, Ceddia *et al.*, 2019]. Despite the demonstrated ability of deep neural networks to extract regulatory signals directly from sequence, there are very few studies that explore cooperativity between regulatory features in genomic data using these methods. Deep Feature Interaction Maps (*DFIM*) uses a network attribution method called DeepLIFT [Shrikumar *et al.*, 2017] to estimate interactions between regulatory elements, tested for one pair at a time [Greenside *et al.*, 2018]. The major drawback of DFIM is that it is computationally expensive: the interactions are inferred in a separate post-processing step and involves recalculation of network gradients. We note that the recent *DeepResolve* method infers feature importance and whether a feature participates in interactions with other features, but does not infer pairs of interacting features explicitly [Liu *et al.*, 2019].

Recently, neural networks that use the concepts of *attention* and *self-attention* [Parikh *et al.*, 2016, Lin *et al.*, 2017] have achieved remarkable success in natural language processing tasks, specifically in machine translation [Vaswani *et al.*, 2017]. One of the strengths of attention is that it can capture associations between features regardless of the distance between them, addressing a major shortcoming of convolutional and

© The Author 2020.

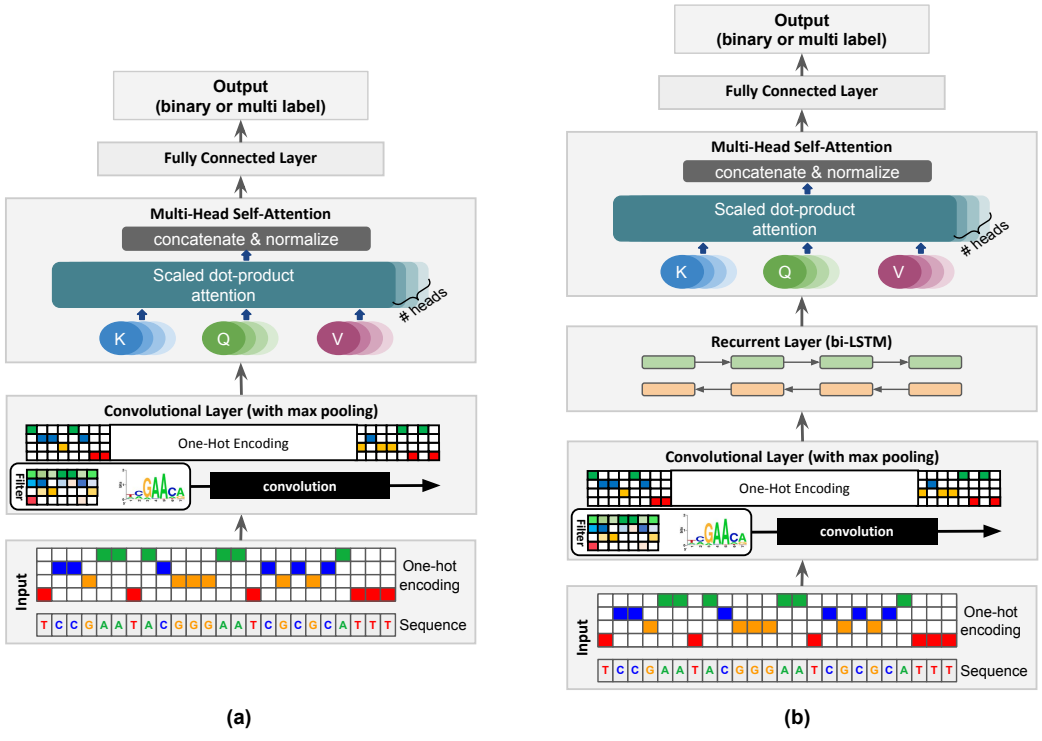

Fig. 1. Model architecture variants. We use a convolutional layer followed by a multi-head self-attention layer (a); optionally, we add a recurrent layer between the two (b). The input in both cases is a one-hot encoding of the DNA sequence. The output of the model is either be a binary or multi-label prediction.

recurrent networks. This is particularly useful for tasks in computational biology where our goal is to identify regulatory elements and their associations/interactions in DNA or RNA sequences. The value of attention for modeling transcription factor binding site prediction was recently demonstrated, and their work was motivated by the greater interpretability of the resulting networks [Chen *et al.*, 2019]. However, to the best of our knowledge, it has not been employed for inferring regulatory interactions between TFs and other regulatory elements. To this end, we propose **SATORI**, a self-attention based deep learning model to capture regulatory element interactions in genomic sequences. The primary components of the architecture of our model are a CNN layer and a multi-head self-attention layer. Optionally, we also incorporate an RNN layer between the two primary layers. The convolutional layer discovers features/motifs in the input sequences. The self-attention layer then captures potential interactions between those features without the need for explicitly testing all possible combinations of motifs. That enables us to infer a global landscape of interactions in a given genomic dataset, without a computationally-expensive post-processing step.

We test SATORI on several simulated and real datasets, including data on chromatin accessibility in 164 cell lines in all human promoters and genome-wide chromatin accessibility data across 36 samples in Arabidopsis. Our results demonstrate the ability of the method to infer the landscape of cooperative interactions between TFs genome-wide.

2 Methods

2.1 Model architecture

We present a self-attention based deep neural network to capture interactions between regulatory features in genomic sequences. Figure 1 depicts the two main architectures we explored in our work. Note that the input to the model—DNA/RNA sequences—was represented as a one-hot encoding: a sequence of length L is transformed into a matrix of size $4 \times L$ where each position in the sequence is represented by a column in the matrix as a non-zero value at location corresponding to one of the four DNA nucleotides.

The first component of our model is a CNN layer where a finite set of filters are scanned against the input sequence/matrix. Formally, we can write the convolution operation as:

$$X'_{i,j} = \sum_{a=0}^{A-1} \sum_{b=0}^{B-1} \omega^j_{a,b} X_{i+a,b} \tag{1}$$

Where X is the input matrix, i is the position at which convolution is performed, j is the index of the filter, and ω^j is the weight matrix of the filter with size $A \times B$ where A is the length of the filter (window size) and B is the number of input channels: four in the case of one-hot encoded input DNA sequences. Here the filters are equivalent to Positional Weight Matrices (PWMs) or sequence motifs. After the convolution operation, we use a non-linear function known as the Rectified Linear Unit (ReLU) which is given by:

$$f(x) = \max(0, x). \tag{2}$$

ReLU is a standard activation function in deep learning which fixes the problem of vanishing gradients. Next, we reduce the output size by max-pooling by taking the maximum value in a window of a pre-determined size. This reduces the input size for the next layer and also achieves invariance to small shifts in the input sequence.

Optionally, as shown in Figure 1(b), we use an RNN layer following the CNN layer. RNNs have an internal state that enables them to capture distant feature interactions in the input sequence. Specifically, we employ a bi-directional RNN with Long Short-Term Memory (LSTM) units [Hochreiter and Schmidhuber, 1997].

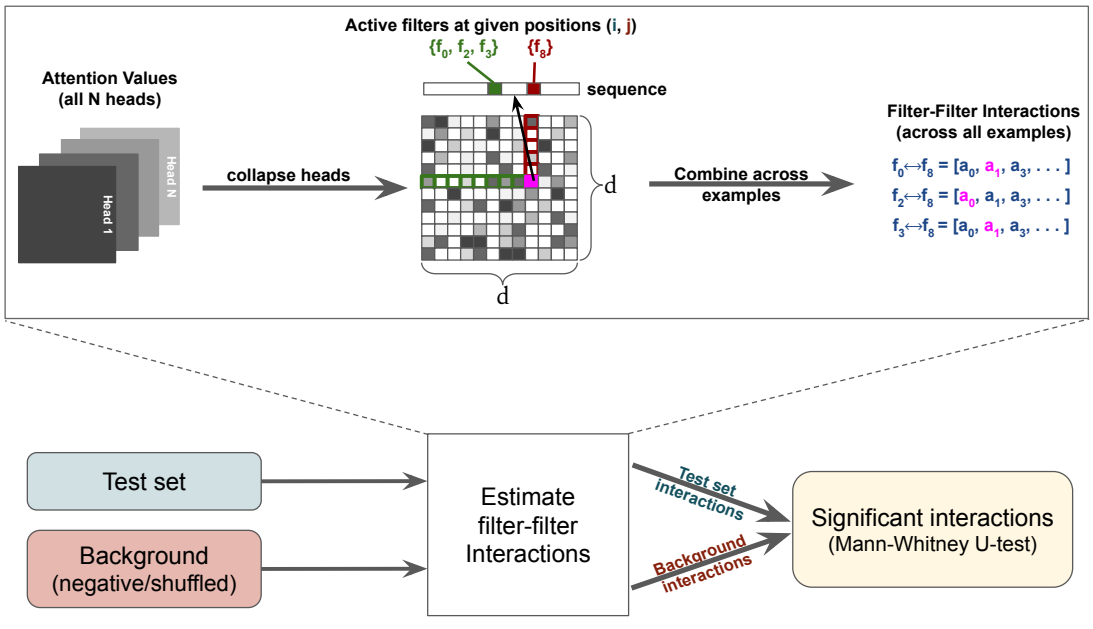

Fig. 2. Summary of the process of inferring interactions from self-attention layer values. For a given example, we collapse the attention heads into a single matrix. Next, at each pair of positions, the corresponding active CNN filters are identified and the attention value is assigned to the interacting pair. This is repeated for all examples to generate interaction profiles for all filter-pairs. Finally, we use a background set to test the significance of filter-filter interactions.

Table 1. Summary of the datasets used in the four experiments we designed to test and analyze SATORI. The first two datasets have binary labels whereas the last two experiments deal with a multi-label, multi-calss problem.

Dataset	Description	Size
Simulated	Simulated DNA sequences	120,000 (total)
	ELF1 and SIX5 embedded in positive examples	40,000 (positive), 80,000 (negative)
	ELF1 or SIX5 embedded in negative examples	
	Instances of AP3 or TAL1 are also randomly embedded	
TAL-GATA ChIP-Seq	DNA sequences for TAL1, GATA1, and GATA2	105,134 (total)
	ChIP-peaks in K562. Positive examples were peaks	25,134 (positive), 80,000 (negative)
	that overlapped DHSs. Negative examples were all	
	other DHSs.	
Human promoter open chromatin	DHSs overlapping the promoters of human genome	20,613 examples across 164 cell types
Arabidopsis open chromatin	DHSs/THSs across arabidopsis genome	88,245 examples across 36 samples

2.1.1 Multi-head self-attention

The core component of our network is a multi-head self-attention layer. Attention mechanism can model dependencies within the input sequence regardless of their distance [Vaswani *et al.*, 2017]. By doing so, it guides the network to focus on relevant features within the input and ignore irrelevant information. Pertinent to the problem at hand, a self-attention layer can help us identify interacting regions within the input sequences. Consequently, we can capture interactions between regulatory events.

To define self-attention, consider the input X to this layer, which is the output of the convolutional/recurrent layer and two linear transformations of it, called the Query Q, and Key K which are defined by:

$$Q = W_Q^\top X, \tag{3}$$

$$K = W_K^\top X, \tag{4}$$

where W_Q, and W_K are the corresponding weight matrices for the Query and Key, respectively. The attention matrix A is then computed using the following expression:

$$A(Q, K) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right), \tag{5}$$

where the softmax function is defined as:

$$\text{softmax}(\text{x}_\text{i}) = \frac{e^{x_i}}{\sum_j e^{x_j}}. \tag{6}$$

In Equation 5, d_k is the dimension of the Key K. The scaling by $\frac{1}{\sqrt{d_k}}$ ensures more stable gradients of the softmax function for large values of d_k [Vaswani *et al.*, 2017]. A is a $d \times d$ matrix where

$$d = \frac{L}{M_p}, \tag{7}$$

L is the length of the input sequence, and M_p is the size of the window used in the max-pooling operation. For every position in the max-pooled

output, the attention matrix summarizes the influence of all other locations on that position. To generate the output of the attention layer, we first define the Value matrix

$$V = W_V^\top X \tag{8}$$

using the associated weight matrix W_V. Finally, we define the output of the attention layer as:

$$Z = A \times V. \tag{9}$$

Intuitively, Equation (9) allows us to generate the output using the parts of the input that we want to focus on—those which exhibit strong interdependencies—and ignore irrelevant information.

For multi-head self-attention, we concatenate the N single-heads followed by a linear transformation. The final output is then collapsed along the hidden (attention) layer dimensions through addition and normalized by the mean and the standard deviation of the result. Empirically we find that this step is not only computationally efficient but also lead to better model accuracy in comparison to just flattening the attention layer output. The final fully connected read-out layer outputs the model's prediction: either a binary or multi-label classification, depending on the experiment. For binary classification, we use the standard cross entropy loss function. For multi-label classification, we use the binary cross entropy with logits loss function:

$$L(y, \hat{y}) = -\frac{1}{C}\sum_{i=1}^{C}\left[y_i \log \sigma(\hat{y}_i) + (1 - y_i)\log(1 - \sigma(\hat{y}_i))\right],$$

where y is the vector of ground truth labels, $\hat{y}$ are the network predictions, C is the number of classes, and σ is the sigmoid function.

2.2 Network training and evaluation

For model selection and optimization, we employ a random search based algorithm to tune the network's hyperparameters. For the convolutional layers we considered filter size, number of filters, and size of window over which pooling is performed. For the multi-head attention layer we tuned the dimensionality of the features generated, and the size of the output of the multi-head attention layer. Details are provided in Table S1 in the Supplementary material. To evaluate the model, we use a simple strategy of splitting the data into 80%, 10%, and 10% for train, test, and validation sets, respectively. To assess the model's performance, Area Under the ROC Curve (AUC) is used. We do train multiple models to avoid the effects of random initialization. The hyperparameters for the two architectures (see figure 1) are summarized in supplementary table S1. The package was implemented in PyTorch [Paszke *et al.*, 2019] and all the experiments were ran on a Ubuntu server with a 12 GB TITAN V GPU.

2.3 Motif extraction

To interpret the deep learning model, we extract sequence motifs from the weights matrices (filters) of the first convolutional layer, similarly to the methodology used in [Kelley *et al.*, 2016]. For binary classification problems, we use the positive test set examples that achieve a probability score greater than 0.70. This cutoff was chosen as a good trade-off between the number of qualifying examples and confidence in the prediction. We use all test set examples when dealing with a multi-class or multi-label problems. Next, for each filter we identify regions in the set of sequences that activate the filter with a value greater than half of the filter's maximum score over all sequences. The resulting substrings are stacked and for each filter, a PWM is calculated using the nucleotide frequency and background information. Sequence logos are generated using the WebLogo tool [Crooks *et al.*, 2004]. The PWMs are searched against appropriate TF databases using the TomTom tool [Gupta *et al.*, 2006] with distance metric set to Euclidean. For searching we use the human CISBP [Weirauch *et al.*, 2014] and arabiodpsis DAP [O'Malley *et al.*, 2016] databases. In the benchmark experiments, we use custom TF databases, details of which are provided in the supplementary material.

Recently, it has been reported that exponential activation functions such as Softplus in the convolutional layer improve motif information content [Koo and Ploenzke, 2019]. In our experiments this led to a slight improvement in the information content of the CNN filters when the Softplus function was used instead of ReLU activation (median information content = 4.12 with Softplus compared to 4.00 with ReLU).

2.4 Quantifying feature interactions

In this section we describe the process of inferring motif interactions from the self-attention layer. The attention matrix for each head is calculated using Equation (5). Next, we collapse the N heads to a single $d \times d$ matrix by taking the maximum at each position (see Figure 2). This step summarizes the self-attention values from multiple subspaces associated with the corresponding single-heads to a single, attention profile.

The attention matrix provides information about interactions between positions in the sequence. That is next converted into interactions between filters by retrieving the filters that are active in those positions. Finally, for each identified filter-filter interaction, we generate the attention profile across all testing examples. For a given pair, this profile consists of a vector of its attention values at positions where the corresponding filters were active. An interaction pair is discarded if its maximum attention value is below a certain threshold. By default we used the value 0.10; in the human promoter data we used 0.08 to increase sensitivity. We note that the distribution of attention values tends to be bi-modal, with most of the values close to 0 or 1 (see supplementary figure F1).

Filter-filter interactions are then translated to motif interactions by picking the most significant TomTom hits in the appropriate TF database. Note that we might not find significant matches for every CNN filter in the database; it can be expected that our model is capturing interactions of uncharacterized regulatory elements. However, in this paper we focus on the interactions between known TFs. To test the statistical significance of motif interactions, we first generate their attention profiles in the background data (described next). Then the non-parametric Mann-Whitney U Test is used to calculate their significance. All the p-values are adjusted for multiple hypothesis testing using Benjamini-Hochberg method [Benjamini and Hochberg, 1995].

2.4.1 Background selection

As mentioned above, to test the statistical significance of regulatory interactions, we need to compare them to a background. We use a biologically relevant background depending on the experiment:

- For binary classification problems, the negative test set is used as the background.
- For multi-label, multi-class or regression problems we generate a background set by shuffling the test set sequences while preserving their di-nucleotide frequencies. Next, in the shuffled sequences, we randomly embed motifs that are generated based on the CNN filters, interpreted as probability distributions, taking into account the number of times a filter is active in the original test sequences.

2.5 Data collection and processing

We use four datasets to test our model's ability to capture interactions between regulatory elements. The datasets are summarized in Table 1, and specific details are provided in the Methods section of the supplementary material.

Table 2. The most frequent interacting families of human transcription factors in the TAL-GATA ChIP-peaks in human K562 cell-line. All interactions are significant with adjusted p-value < 0.05.

TF Family Interaction	Frequency	Percent of total interactions
C2H2 ZF $\longleftrightarrow$ CxxC	170	14.36%
Homeodomain $\longleftrightarrow$ C2H2 ZF	121	10.22%
C2H2 ZF $\longleftrightarrow$ C2H2 ZF	90	7.60%
GATA $\longleftrightarrow$ C2H2 ZF	79	6.67%
Homeodomain $\longleftrightarrow$ CxxC	72	6.08%
C2H2 ZF $\longleftrightarrow$ Sox	62	5.24%
C2H2 ZF $\longleftrightarrow$ bHLH	53	4.48%
GATA $\longleftrightarrow$ CxxC	51	4.31%
Sox $\longleftrightarrow$ CxxC	39	3.29%
C2H2 ZF $\longleftrightarrow$ THAP finger	21	1.77%
CxxC $\longleftrightarrow$ bHLH	19	1.60%
Nuclear receptor $\longleftrightarrow$ C2H2 ZF	19	1.60%
GATA $\longleftrightarrow$ bHLH	18	1.27%
Homeodomain $\longleftrightarrow$ bHLH	17	1.44%
SAND $\longleftrightarrow$ C2H2 ZF	17	1.44%

3 Results and Discussion

3.1 Benchmark 1: embedded motif interactions in simulated genomic sequences

In this experiment we used SATORI to test if it can recover interactions embedded in simulated DNA sequences. This test served as a benchmark in order to compare our model to the recently published DFIM method [Greenside *et al.*, 2018]. We used a very similar approach to theirs when creating the simulated dataset: $120,000$ random DNA sequences were generated; in $40,000$ sequences we embedded motifs of the transcription factors SIX5 and ELF1. This simulated an interaction between the two TFs since we required both to be present in every sequence. We labelled these sequences as positive examples. In the rest sequences that serve as negative examples, we embedded only one of the two motifs in each sequence. In addition, motifs of TAL1 and AP3 were embedded at random across the whole dataset.

Not surprisingly, both variants of our model achieved perfect classification accuracy on the test set for this data. We then analyzed the attention layer weights and inferred statistically significant motif interactions and found that all the significant interactions returned by our model involve SIX5 and ELF1 as expected. These interactions are summarized in Supplementary Table S2.

3.2 Benchmark 2: Inferring TAL-GATA motif interactions from ChIP-Seq data

The TFs TAL1 and GATA1 have been reported to interact: GATA1 requires a prior or simultaneous binding of TAL1 before it can bind DNA [Kassouf *et al.*, 2010]. To investigate these interactions in this experiment, which follows a similar experiment performed by the authors of DFIM, we formulated a binary classification problem where the positive set consisted of sequences of the TAL1, GATA1, and GATA2 ChIP-Seq peaks that overlapped regions of open chromatin (DHSs) in the human K562 cell-line. For the negative set, sequences of all other K562 DHSs that didn't overlap any of the ChIP-Seq peaks were used. This experiment serves as another benchmark for our model, and was also used by Greenside et al. to test their model's ability uncover interactions between TAL1 and GATA1/GATA2 [Greenside *et al.*, 2018]. Further details regarding the dataset are provided in the supplementary material.

We trained both variants of our model on this dataset; in this harder dataset the variant with an RNN layer performed much better with an AUC of 0.94 on the test set compared to 0.85 for the model without an RNN. We note that the authors of DFIM used five layers of convolution to achieve comparable accuracy. We recovered multiple significant filter-filter interactions that mapped to TAL1 and GATA motifs with highly significant p-values (see supplementary table S3), demonstrating the ability of our model to recover in-*vivo* interactions between TFs.

Since the ChIP-Seq peaks of TAL and GATA transcription factors are occurring in regions of open chromatin, it is highly likely that in those regions, other regulatory elements/TFs are active and interacting with each other. Therefore, we also let SATORI search for interactions among all known human transcription factors and found numerous other interactions. Table 2 summarizes the 15 most frequent TF family-family interactions, consisting predominantly of interactions between members of the C2H2 ZF, Homeodomain, CxxC, and GATA families (see Supplementary Table S4 for the list of individual TF interactions). An interesting observation here is that the interactions between the GATA and bHLH (TAL1) families are not the most frequent, despite the fact that the model used their ChIP-Seq peaks. This is likely because of the differences in size of these TF families: the C2H2 ZF and Homeodomain families are the largest TF families in humans, whereas bHLH and particularly GATA, are much smaller in size.

Because of the similarity between some TF motifs, the matching between filters and motifs is not without errors. In this second experiment GATA was predicted to interact with TCF15, which has a motif that closely resembles that of TAL1. In fact, both of them belong to the same bHLH family. Figure F2 in the Supplement shows the similarity between these motifs.

3.3 The TF interaction landscape across human promoters

In this experiment we investigated regulatory interactions between TFs in all human promoter regions using DNase I hypersentivity data (DHSs) across 164 immortalized cell lines and tissues. This experiment was based on Kelley et al's work where they predicted chromatin accessibility from sequence information alone across the entire human genome [Kelley *et al.*, 2016]. The labels in this data represent presence/absence of a given DHS across each of the 164 cell lines, and is a multi-label classification problem. We trained both network variants (see figure 1) and observed that with the optional RNN layer, the network performed better in terms AUC scores. Supplementary figure F3 compares the accuracy for each of the 164 cell lines for both variants of the architecture.

The trained network yielded filters that matched 93 TFs with known motifs (counting only filters that had information content greater than 3.0). Among those 93 TFs, our model identified 285 pairs of motifs that interact, with a total of 1250 statistically significant interactions. The 20 most frequent interactions are shown in Figure 3(a). For the complete list, refer to Supplementary Table S5. We also looked at the distribution of the distances between interacting motifs, and observed that, as expected, interactions tend to occur in close proximity with a median distance of interaction of 168 bp (see Figure 3(b)). Overall, the Homeodomain, C2H2 ZF, and CxxC families were the most frequent families of interacting TFs (see Figure 4). Finally, it is worth mentioning that for twelve interactions out of the total of 285, we found evidence in the TRRUST database [Han *et al.*, 2017] which lists only 58 interactions among the 93 TFs (see Table S6 in the Supplementary Material for details). This overlap is statistically significant with a p-value of 4.66×10^{-7} using the hypergeometric test.

We note that in this work we only analyzed interactions between motifs of known TFs. Not all filters can be mapped to characterized regulatory proteins. In this dataset, TomTom returned no significant matches for 80 out of the CNN filters with motif information content greater than 3.0. The interactions of such filters require further investigation to discover the regulatory molecules associated with them.

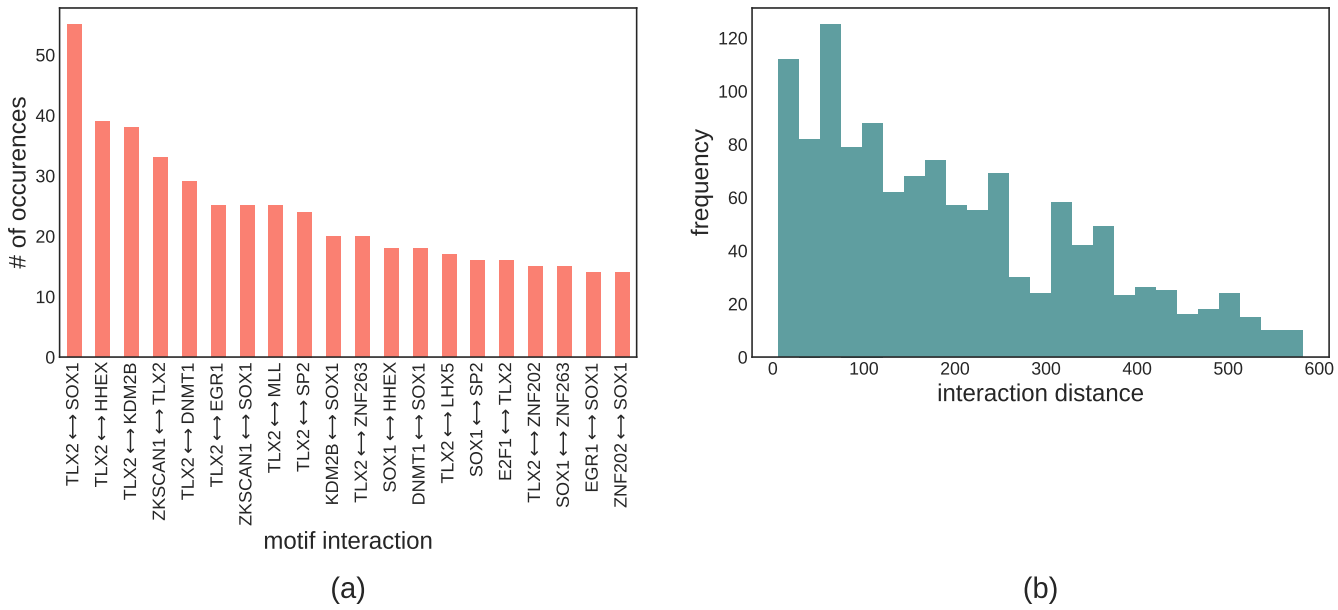

Fig. 3. The most frequent TF interactions in human promoters (a). The distribution of TF-TF interaction distances (b).

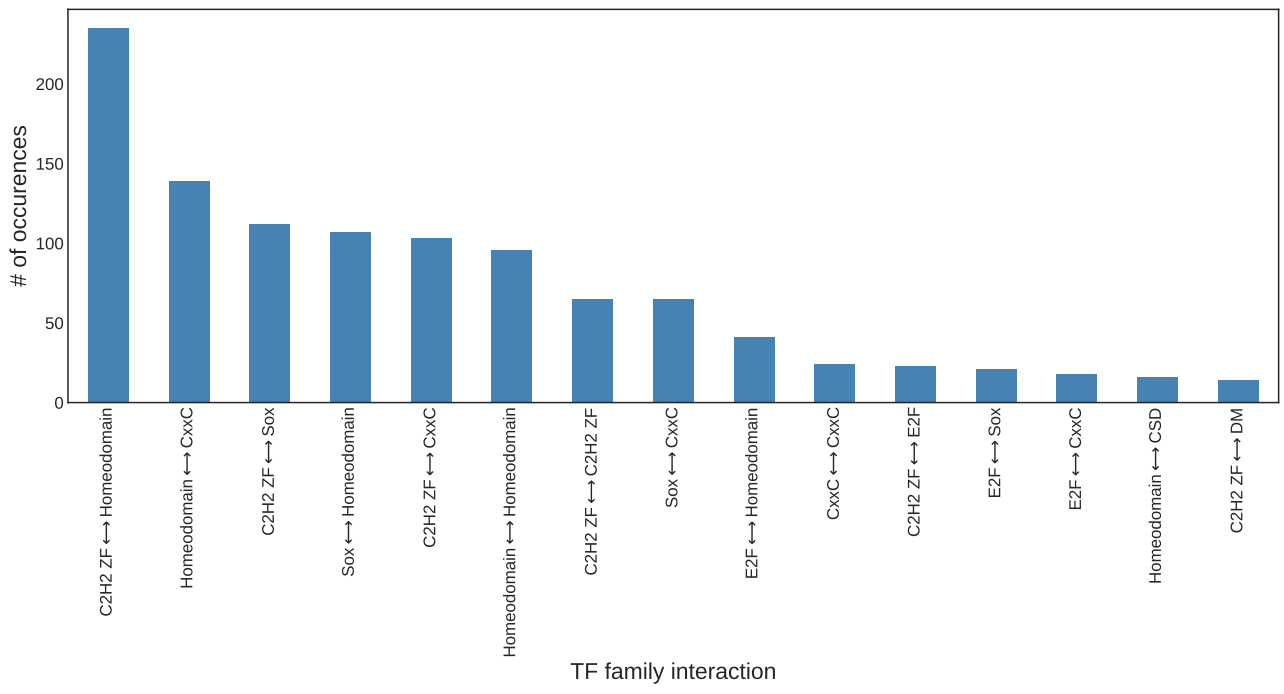

Fig. 4. The most frequent interacting transcription factor families in human promoter regions.

As mentioned above, the motif matching results returned by TomTom are noisy and imperfect. For example, some of the statistically significant matches are clearly incorrect, as shown in Supplementary Figure F4. However, this is not a shortcoming of our model, but rather a limitation of the interpretation of filter-filter interactions.

3.4 Genome-wide regulatory interactions in arabidopsis

We extended our analysis to plants by designing a similar experiment as described in the previous section. More specifically, we predict chromatin accessibility using sequence information alone across 36 arabidopsis samples from recently published arabidopsis DNase I-Seq and ATAC-Seq studies (GEO accession numbers provided in the Supplementary Methods section). Like the previous dataset, this too is a multi-label prediction problem, where the labels indicates whether a given region has a peak in each of the 36 samples of DNase I-Seq and ATAC-Seq. We trained both variants of our model and similarly to the other datasets, the network that included an RNN layer performed slightly better in terms of median AUC across samples (0.86 compared to 0.85).

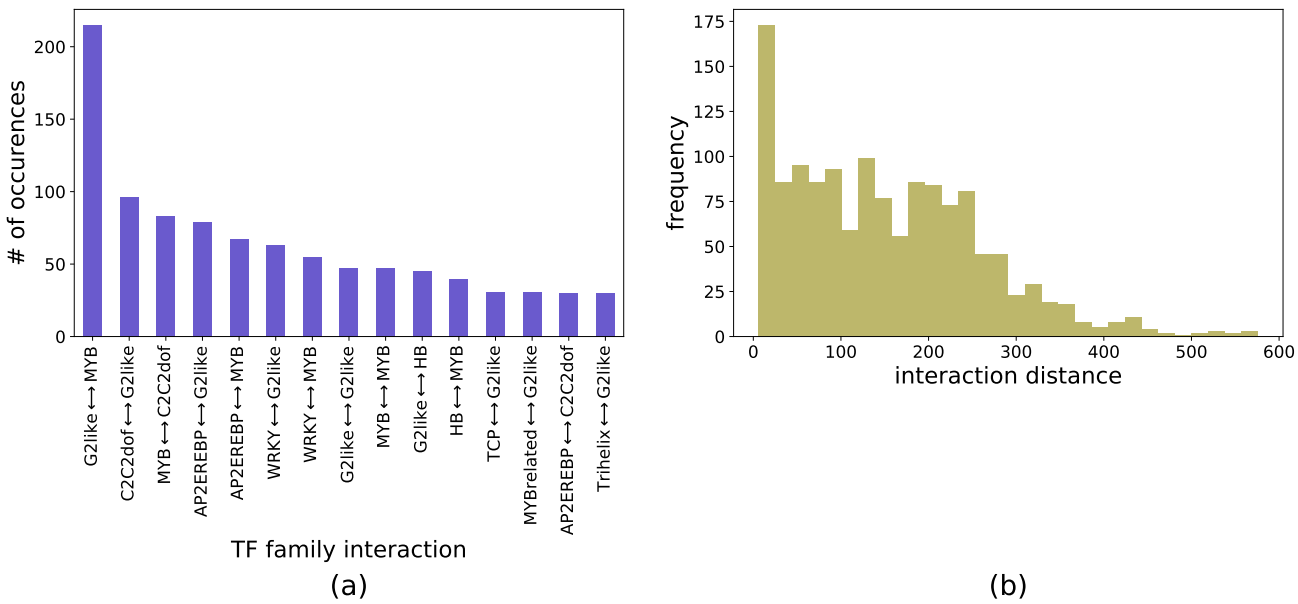

Fig. 5. The regulatory interaction landscape in accessible chromatin in the arabipdosis genome. The most frequently interacting families of plant transcription factors (a). The distribution of distances between inferred TF-TF interactions (b).

In the next step, we investigated genome-wide regulatory interactions in those regions of open-chromatin. The trained network yielded 189 filters with information content above 3.0, and we obtained 100 unique matches for those filters in the DAP-Seq arabidopsis TF database [O'Malley *et al.*, 2016]. Among these 100 TFs, our model identified interactions between 230 pairs of TFs with a total of approximately 1400 statistically significant interactions involving diverse plant transcription factors (see Figure 5(a)). G2like, MYB, C2C2dof, and AP2 were the most frequently represented TF families in those interactions. Similarly to our findings in human, plant TF interactions tend to occur in relative proximity (median distance = 138 bp) as shown in Figure 5(b). Arabidopsis does not have a database of known interactions between TFs, so our results could not be validated. Targeted experimental validation of these predictions can thus significantly enrich our knowledge of the combinatorial regulation of gene expression in plants.

4 Conclusions and Future Work

In this work we presented SATORI—a deep learning model and associated method for extracting interactions between its learned features. Unlike existing methods, it does not require any post-processing and uses the sparsity of the attention matrix to infer the most salient interactions. The proposed method can be extended in several ways. Here, we focused on globally scoring interactions between regulatory features. This is in contrast to feature attribution methods that score the contribution of features in genomic regions of interest. We believe that the sparsity of the attention matrix makes it useful as an attribution method as well, but further experiments are required in order to validate that. Finally, the proposed methodology is flexible enough to be applied to deep networks that integrate multiple data modalities, and in applications outside of computational biology. For example, it can allow discovery of interactions between different characteristics of chromatin structure to provide a better understanding of the relationship between epigenetic markers such as histone modifications, DNA methylation, and nucleosome positioning and their contribution to the regulation of gene expression.

References

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, **33**(8), 831.

Angermueller, C., Lee, H. J., Reik, W., and Stegle, O. (2017). Deepcpg: accurate prediction of single-cell dna methylation states using deep learning. *Genome biology*, **18**(1), 67.

Banovich, N. E., Li, Y. I., Raj, A., Ward, M. C., Greenside, P., Calderon, D., Tung, P. Y., Burnett, J. E., Myrthil, M., Thomas, S. M., *et al.* (2018). Impact of regulatory variation across human ipscs and differentiated cells. *Genome research*, **28**(1), 122–131.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, **57**(1), 289–300.

Ceddia, G., Martino, L. N., Parodi, A., Secchi, P., Campaner, S., and Masseroli, M. (2019). Association rule mining to identify transcription factor interactions in genomic regions. *Bioinformatics (Oxford, England)*.

Chen, C., Hou, J., Shi, X., Yang, H., Birchler, J. A., and Cheng, J. (2019). Interpretable attention model in transcription factor binding site prediction with deep neural networks. *bioRxiv*, page 648691.

Crooks, G. E., Hon, G., Chandonia, J. M., and Brenner, S. E. (2004). Weblogo: A sequence logo generator. *Genome Research*, **14**, 1188–90.

Greenside, P., Shimko, T., Fordyce, P., and Kundaje, A. (2018). Discovering epistatic feature interactions from neural network models of regulatory dna sequences. *Bioinformatics*, **34**(17), i629–i637.

GuhaThakurta, D. and Stormo, G. D. (2001). Identifying target sites for cooperatively binding factors. *Bioinformatics*, **17**(7), 608–621.

Gupta, S., Stamatoyannopoulos, J. A., Bailey, T. L., and Noble, W. S. (2006). Quantifying similarity between motifs. *Genome Biology*, **8**, R24 – R24.

Han, H., Cho, J.-W., Lee, S., Yun, A., Kim, H., Bae, D., Yang, S., Kim, C. Y., Lee, M., Kim, E., *et al.* (2017). Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, **46**(D1), D380–D386.

Hannenhalli, S. and Levy, S. (2002). Predicting transcription factor synergism. *Nucleic acids research*, **30**(19), 4278–4284.

Hassanzadeh, H. R. and Wang, M. D. (2016). Deeperbind: Enhancing prediction of sequence specificities of dna binding proteins. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 178–183. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Kassouf, M. T., Hughes, J. R., Taylor, S., McGowan, S. J., Soneji, S., Green, A. L., Vyas, P., and Porcher, C. (2010). Genome-wide identification of tal1's functional targets: insights into its mechanisms of action in primary erythroid cells. *Genome*

research, **20**(8), 1064–1083.

Kelley, D. R., Snoek, J., and Rinn, J. L. (2016). Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, **26**(7), 990–999.

Koh, P. W., Pierson, E., and Kundaje, A. (2017). Denoising genome-wide histone chip-seq with convolutional neural networks. *Bioinformatics*, **33**(14), i225–i233.

Koo, P. K. and Ploenzke, M. (2019). Improving convolutional network interpretability with exponential activations. *bioRxiv*, page 650804.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Liu, G., Zeng, H., and Gifford, D. K. (2019). Visualizing complex feature interactions and feature sharing in genomic deep neural networks. *BMC bioinformatics*, **20**(1), 1–14.

O'Malley, R. C., Huang, S.-s. C., Song, L., Lewsey, M. G., Bartlett, A., Nery, J. R., Galli, M., Gallavotti, A., and Ecker, J. R. (2016). Cistrome and epicistrome features shape the regulatory dna landscape. *Cell*, **165**(5), 1280–1292.

Pan, X. and Shen, H.-B. (2017). Rna-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. *BMC bioinformatics*, **18**(1), 136.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pilpel, Y., Sudarsanam, P., and Church, G. M. (2001). Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature genetics*, **29**(2), 153.

Qin, Q. and Feng, J. (2017). Imputation for transcription factor binding predictions based on deep learning. *PLoS computational biology*, **13**(2), e1005403.

Quang, D. and Xie, X. (2016). Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, **44**(11), e107–e107.

Roca, H. and Franceschi, R. T. (2008). Analysis of transcription factor interactions in osteoblasts using competitive chromatin immunoprecipitation. *Nucleic acids research*, **36**(5), 1723–1730.

Safe, S. (2015). Microrna-specificity protein (sp) transcription factor interactions and significance in carcinogenesis. *Current pharmacology reports*, **1**(2), 73–78.

Schreiber, J., Libbrecht, M., Bilmes, J., and Noble, W. (2018). Nucleotide sequence and dnasei sensitivity are predictive of 3d chromatin architecture. *bioRxiv*, page 103614.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org.

Sudarsanam, P., Pilpel, Y., and Church, G. M. (2002). Genome-wide co-occurrence of promoter elements reveals a cis-regulatory cassette of rrna transcription motifs in saccharomyces cerevisiae. *Genome research*, **12**(11), 1723–1731.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wasserman, W. W. and Fickett, J. W. (1998). Identification of regulatory regions which confer muscle-specific gene expression. *Journal of molecular biology*, **278**(1), 167–181.

Weirauch, M. T., Yang, A., Albu, M., Cote, A. G., Montenegro-Montero, A., Drewe, P., Najafabadi, H. S., Lambert, S. A., Mann, I., Cook, K. B., Zheng, H. Y., Goity, A., van Bakel, H., Lozano, J. F., Galli, M., Lewsey, M. G., Huang, E., Mukherjee, T., Chen, X., Reece-Hoyes, J. S., Govindarajan, S., Shaulsky, G., Walhout, A. J. M., Bouget, F.-Y., Ratsch, G., Larrondo, L. F., Ecker, J. R., and Hughes, T. R. (2014). Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, **158**, 1431–1443.

Zhang, S., Zhou, J., Hu, H., Gong, H., Chen, L., Cheng, C., and Zeng, J. (2015). A deep learning framework for modeling structural features of rna-binding protein targets. *Nucleic acids research*, **44**(4), e32–e32.

Zhou, J. and Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, **12**(10), 931.