Predicting Enhancer-Promoter Interaction from Genomic Sequence with Deep Learning

Shashank Singh
Statistics & Machine Learning Departments
Carnegie Mellon University
sss1@andrew.cmu.edu

Yang Yang
Computational Biology Department
Carnegie Mellon University
yy3@andrew.cmu.edu

Barnabás Póczos
Machine Learning Department
Carnegie Mellon University
bapoczos@cs.cmu.edu

Jian Ma
Computational Biology Department
Carnegie Mellon University
jianma@cs.cmu.edu

1 Introduction

Our understanding of how the human genome regulates complex cellular functions in a living organism is still primitive. One particular aspect that we know extremely little about is the high-order organization of the human genome in the cell nucleus. The genome in each human cell contains approximately 6 feet long DNA being tightly folded and packaged into a nucleus with 5μm diameter. Intriguingly, this packaging is highly organized and tightly controlled [1]. For example, distal regulatory enhancer elements in the genome can interact with proximal promoter regions to regulate the target gene expression, and the mutations that change such interactions will cause the target gene to be dysregulated [2–4]. However, the principles at the sequence level underlying such organization and chromatin interaction are poorly understood.

In this work, we focus on enhancer-promoter interactions (EPI) of the genome. Although certain sequence features (e.g., CTCF binding motifs [5]) are known to mediate chromatin loops, it remains largely elusive whether and what information encoded in the genome sequence contains instructions for forming EPI. In mammalian and vertebrate genomes, the promoter regions of the gene and their distal regulatory enhancers may be millions of base-pairs away from each other; and a promoter may not interact with its closest enhancer. There are prior works in predicting EPI based on epigenetic features in enhancers and promoters as well as target gene expressions [6, 7]. However, no algorithm exists to predict EPI using sequence-level signatures only. In the past year, there has been an explosion of deep learning approaches to the related problem of genome annotation [8–13]. However, no deep learning model currently exists to predict the high-order chromatin interactions of functional sequences. The main contribution of this work is to provide a deep learning based architecture for predicting EPI using sequence-based features only, which in turn demonstrates that the principles of regulating EPI may be largely encoded in the genome sequences.

2 Methods

2.1 Model

Our model, named SPEID (Sequence-based Promoter-Enhancer Interaction with Deep learning), is illustrated in Figure 1. The network consists of a pair of convolution/activation/max-pool layers, a recurrent long short-term memory (LSTM) layer, and a dense layer.

The first layers of the network are responsible for learning informative subsequence features of the inputs. Because informative subsequence features may differ between enhancers and promoters, we train separate branches for each. These features might include, for example, transcription factor (TF) protein binding motifs and other sequence-based signals. Characterizing these sequence features is an important problem for future study; for now, we are simply using the deep network to learn them,

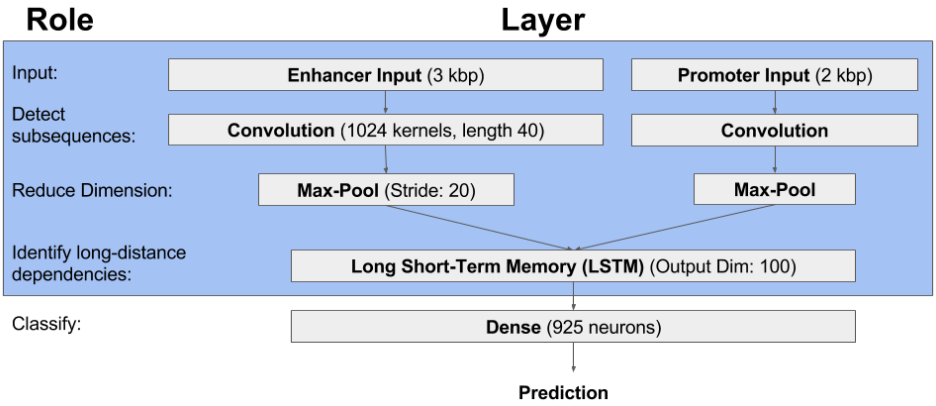

Figure 1: Diagram of our deep learning model SPEID.

supported by some prior knowledge of TF binding motifs. Specifically, as suggested by [11], we inject some prior knowledge by initializing about half of the convolutional kernels with known motifs from the JASPAR database [14]. Note that convolutional kernels are model parameters to be learned from the data, and the model is free to retain, modify, or discard these initial values based on whether they are useful for prediction.

Each branch consists of a *convolution layer* and a rectified linear unit (ReLU) *activation layer*, which together extract subsequence features from the input, and a *pooling layer*, which reduces dimensionality. Pooling is especially important in EPI prediction because of the long input sequence (5kbp, as compared to 1kbp used when predicting sequence variant function [8, 11]). (Parameters: Number of Kernels: 1024, Filter Length: 40, L_2 penalty weight: 10^{-5}, Pool Length: 20, Stride: 20)

Before feeding into the next layer, the enhancer and promoter branches are concatenated into a single output. The remaining layers of the network act jointly on this concatenation, rather than as disjoint pairs of layers, as in the previous layers.

The next layer is a *recurrent long short-term memory (LSTM) layer*, responsible for identifying informative combinations of the extracted features, across the extent of the sequence. To do this the LSTM sweeps across its input, choosing to remember or forget each feature. This layer is bidirectional, in that it sweeps from both left to right and right to left; the outputs of each direction are concatenated before feeding into the next layer. (Parameters: Output dimension: 100 left-to-right and 100 right-to-left)

The final *dense layer* is simply an array of hidden units with nonlinear (ReLU) activations feeding into a single sigmoid unit that predicts a class based on the output R. (Parameters: Number of units: 925, L_2 penalty weight: 10^{-5})

Similar (albeit simpler) architectures have been used for the related problem of predicting function of non-coding sequence variants [8, 11]. In fact, our use of a recurrent LSTM layer rather than a hierarchy of convolutional/max-pooling layers is inspired by the architecture of DanQ [11]. However, our method solves a fundamentally different problem – predicting interactions between sequences rather than predicting annotations from a single sequence. Hence, our model has a branched architecture, which takes two inputs and produces a single classification, rather than a sequential architecture. Because the data for this problem are far sparser, we also require a more careful training procedure, as detailed in the next section. There are also several finer distinctions between the models, such as our use of batch normalization to accelerate training and weight regularization to improve generalization.

Other training parameters. To accelerate training, we performed batch normalization at 4 points in the network: before and after the LSTM layer, between the dense layer and its ReLU activation, and between the dense layer and the final sigmoid classifier. The model was trained in minibatches of 100 samples by back-propagation, using binary cross-entropy loss, minimized by Adam [15] with a learning rate of 10^{-5}. The initial training phase lasted 32 epochs and the retraining phase lasted 80 epochs, taking, for instance, 11 and 6 hours for K562, respectively, on an NVIDIA GTX 1080 GPU.

2.2 Training Procedure

Recall that our data set is highly imbalanced – there are many more negative (noninteracting) pairs than positive (interacting) pairs. In each cell line, there are typically 20 times more negative samples than positive samples. To combat the difficulty of learning highly imbalanced classes, we utilize a two-stage training procedure that involves pre-training on a data set balanced with data augmentation, followed by training on the original data.

2.2.1 Pre-training with Data Augmentation

Data augmentation is commonly used as an alternative to re-weighting data when training deep learning models on highly imbalanced classes. For example, image data is often augmented with random translations, scalings, and rotations of the original data. In our case, because enhancers and promoters are typically smaller than the fixed window size we use as input, the labels are invariant to small shifts of the input sequence, as long as the enhancer or promoter remains within this window. By randomly shifting each positive promoter and enhancer within this window, we generated "new" positive samples. We did this 20 times with each positive sample, effectively balancing the positive and negative classes.

In addition to balancing class sizes, this data augmentation has the additional benefit of promoting translation invariance in our model. This is desirable because informative subsequences of an input enhancer or promoter need not consistently appear in the same position.

2.2.2 Imbalanced Training

Data augmentation results in a consistent training procedure for the network, allowing the convolutional layers to identify informative subsequence features and the recurrent layer to identify long-range dependencies between these features. However, in typical applications of predicting interactions, classes are, as in our original data, highly imbalanced. In these contexts, naively using the network trained on augmented data results in a very high false positive rate.

Fortunately, this has relatively little to do with the convolutional and recurrent layers of the network, which correctly learn features that distinguish positive and negative samples, and this issue is largely due to the dense layer, which performs prediction based on these features. Hence, to correct for this, we only to retrain the dense layer. We do this by "freezing" the lower layers of the network (i.e., setting the learning rate to 0), and then continuing to train the network as usual on the subset of the original imbalanced data that was used to generate the augmented data.

2.2.3 Summary of Training Procedure

1. Begin with an imbalanced data set A.
2. Split the data uniformly into a large training set B and a small testing set C.
3. Augment positive samples in B to produce a balanced data set D.
4. Train the model extensively on D, using a small subset for validation.
5. Freeze the convolution and recurrent layers of the model.
6. Continue training the dense layer of the model on B.
7. Evaluate on C.

3 Results

3.1 Data

We utilized the EPI data sets previously used in TargetFinder [7] for our model training and evaluation. The data include six cell lines (GM12878, HeLa-S3, HUVEC, IMR90, K562, and NHEK). Cell-line specific active enhancers and promoters were identified using annotations from the ENCODE Project [16] and Roadmap Epigenomics Project [17]. The data for each cell line consist of enhancer-promoter pairs which are annotated as positive (interacting) or negative (non-interacting) using high-resolution genome-wide measurements of chromatin contacts in each cell line based on Hi-C [5], as described in [7]. 20 negative pairs were sampled per positive pair, under constraints on the genomic distance between the paired enhancer and promoter as described in [7], such that positive and negative pairs had similar enhancer-promoter distance distributions. The resultant datasets are heavily imbalanced, in accordance with the fact that enhancer/promoter interactions are far fewer than non-interactions.

To address the problem induced by the data imbalance, we applied data augmentation accordingly to the positive pairs, which is described in detail in Section 2.2.1. The numbers of positive pairs, augmented positive pairs, and negative pairs on each cell line are listed in Table 1. The original annotated enhancers in the dataset of each cell line are mostly only a few hundred base pairs in length. We extended the enhancers to 3kbp by including adjustable flanking regions for more informative feature extraction with the use of the surrounding context. The enhancers are fitted to a uniform length with the extensions, as sequences of fixed sizes are needed as input to the proposed model. The original promoters are mostly 1-2 kbp in length, which are similarly fitted to a fixed window size of 2kbp. We convert the genomic sequence to a 4×3000 matrix (for enhancer) and 4×2000 matrix (for promoter) as inputs to our model with a one-hot encoding (e.g., 'A' is $(1, 0, 0, 0)^T$, 'T' is $(0, 1, 0, 0)^T$, etc.).

Cell Line	Positive Pairs	Augmented Positive Pairs	Negative Pairs
GM12878	2,113	42,260	42,200
HeLa-S3	1,740	34,800	34,800
HUVEC	1,524	30,480	30,400
IMR90	1,254	25,080	25,000
K562	1,977	39,540	39,500
NHEK	1,291	25,820	25,600

Table 1: Positive sample, augmented positive sample, and negative sample counts, for each cell line.

3.2 Evaluation results as compared to TargetFinder

We compared our prediction results to a state-of-the-art model, TargetFinder [7], which uses boosted trees to predict EPI based on a number of epigenetic feature annotations. TargetFinder has 3 variants, which use features from different regions: Enhancer/Promoter (E/P) uses only annotations within the enhancer and promoter, Extended Enhancer/Promoter (EE/P) additionally uses annotations within an extended 3kbp flanking region around each enhancer, and Enhancer/Promoter/Window (E/P/W) additionally uses annotations in the region between the enhancer and promoter.

Table 2 shows a comparison of results between our SPEID method and different TargetFinder models, on each of 6 different cell types. Due to high class imbalance, we report F_1 score (harmonic mean of precision and recall), rather than accuracy. As described in Section 2.2.3, reported results are on a held-out subset of 10% of the original data. We found that, although results vary across different cell lines, the performance of our SPEID model can achieve comparable results with TargetFinder's best model. Taken together, our results suggest that sequence-based features have important information that can determine EPI and our model can effectively predict EPI using sequence features only.

	Cell Type					
Model	GM12878	HeLa-S3	HUVEC	IMR90	K562	NHEK
SPEID	0.85	0.81	0.75	0.78	0.85	0.94
TargetFinder (E/P)	0.59	0.61	0.48	0.48	0.61	0.83
TargetFinder (EE/P)	0.84	0.83	0.71	0.83	0.81	0.83
TargetFinder (E/P/W)	0.81	0.87	0.77	0.78	0.85	0.90

Table 2: F_1 scores for different for different EPI prediction methods.

4 Conclusions and Future Work

The question we address in this work is: If we are given a pair of genomic sequences as putative enhancer and promoter, can we predict whether they interact using sequence-based features only? We have developed, to the best of our knowledge, the first deep learning model, SPEID, to tackle this problem. Our results demonstrate that it is possible to obtain state-of-the-art prediction of EPI using only sequence information, obtaining results comparable to those of TargetFinder, which utilizes epigenetic signals and gene expression. We also show that deep learning can effectively extract relevant sequence information. A natural next step is to improve our model to reveal specific sequence-based determinants and also to characterize the most important sequence-based and epigenetic factors combination in determining EPI.

Acknowledgments

This material is based upon work supported by a National Science Foundation Graduate Research Fellowship to the first author under Grant No. DGE-1252522.

References

[1] Tom Sexton and Giacomo Cavalli. The role of chromosome domains in shaping the functional genome. *Cell*, 160(6):1049–1059, 2015.

[2] Yubo Zhang, Chee-Hong Wong, Ramon Y Birnbaum, Guoliang Li, Rebecca Favaro, Chew Yee Ngan, Joanne Lim, Eunice Tai, Huay Mei Poh, Eleanor Wong, et al. Chromatin connectivity maps reveal dynamic promoter-enhancer long-range associations. *Nature*, 504(7479):306–310, 2013.

[3] Jesse R Dixon, Inkyung Jung, Siddarth Selvaraj, Yin Shen, Jessica E Antosiewicz-Bourget, Ah Young Lee, Zhen Ye, Audrey Kim, Nisha Rajagopal, Wei Xie, et al. Chromatin architecture reorganization during stem cell differentiation. *Nature*, 518(7539):331–336, 2015.

[4] Ya Guo, Quan Xu, Daniele Canzio, Jia Shou, Jinhuan Li, David U Gorkin, Inkyung Jung, Haiyang Wu, Yanan Zhai, Yuanxiao Tang, et al. Crispr inversion of ctcf sites alters genome topology and enhancer/promoter function. *Cell*, 162(4):900–910, 2015.

[5] Suhas SP 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, et al. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680, 2014.

[6] Sushmita Roy, Alireza Fotuhi Siahpirani, Deborah Chasman, Sara Knaack, Ferhat Ay, Ron Stewart, Michael Wilson, and Rupa Sridharan. A predictive modeling approach for cell line-specific long-range regulatory interactions. *Nucleic acids research*, 43(18):8694–8712, 2015.

[7] Sean Whalen, Rebecca M Truty, and Katherine S Pollard. Enhancer-promoter interactions are encoded by complex genomic signatures on looping chromatin. *Nature Genetics*, 2016.

[8] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature methods*, 12(10):931–934, 2015.

[9] Yongjin Park and Manolis Kellis. Deep learning for regulatory genomics. *Nature Biotechnology*, 33(8):825–826, 2015.

[10] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 2015.

[11] Daniel Quang and Xiaohui Xie. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *bioRxiv*, page 032821, 2015.

[12] Yifeng Li, Wenqiang Shi, and Wyeth W Wasserman. Genome-wide prediction of cis-regulatory regions using supervised deep learning methods. *bioRxiv*, page 041616, 2016.

[13] David R Kelley, Jasper Snoek, and John L Rinn. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 2016.

[14] Anthony Mathelier, Oriol Fornes, David J Arenillas, Chih-yu Chen, Grégoire Denay, Jessica Lee, Wenqiang Shi, Casper Shyr, Ge Tan, Rebecca Worsley-Hunt, et al. Jaspar 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, page gkv1176, 2015.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] ENCODE Project Consortium et al. The encode (encyclopedia of dna elements) project. *Science*, 306(5696):636–640, 2004.

[17] Bradley E Bernstein, John A Stamatoyannopoulos, Joseph F Costello, Bing Ren, Aleksandar Milosavljevic, Alexander Meissner, Manolis Kellis, Marco A Marra, Arthur L Beaudet, Joseph R Ecker, et al. The nih roadmap epigenomics mapping consortium. *Nature biotechnology*, 28(10):1045–1048, 2010.