Bioinformatics, 36(24), 2020, 5610–5617
doi: 10.1093/bioinformatics/btaa1075
Advance Access Publication Date: 26 December 2020
Original Paper

Sequence analysis

Prediction of histone post-translational modifications using deep learning

Dipankar Ranjan Baisya and Stefano Lonardi *

Department of Computer Science and Engineering, University of California, Riverside, CA, 92521, USA

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on April 23, 2020; revised on November 27, 2020; editorial decision on December 13, 2020; accepted on December 16, 2020

Abstract

Motivation: Histone post-translational modifications (PTMs) are involved in a variety of essential regulatory processes in the cell, including transcription control. Recent studies have shown that histone PTMs can be accurately predicted from the knowledge of transcription factor binding or DNase hypersensitivity data. Similarly, it has been shown that one can predict PTMs from the underlying DNA primary sequence.

Results: In this study, we introduce a deep learning architecture called DeepPTM for predicting histone PTMs from transcription factor binding data and the primary DNA sequence. Extensive experimental results show that our deep learning model outperforms the prediction accuracy of the model proposed in Benveniste *et al.* (PNAS 2014) and DeepHistone (BMC Genomics 2019). The competitive advantage of our framework lies in the synergistic use of deep learning combined with an effective pre-processing step. Our classification framework has also enabled the discovery that the knowledge of a small subset of transcription factors (which are histone-PTM and cell-type-specific) can provide almost the same prediction accuracy that can be obtained using all the transcription factors data.

Availabilityand implementation: https://github.com/dDipankar/DeepPTM.

Contact: stelo@cs.ucr.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Histones are a class of proteins that bind to DNA and help to condense the DNA into chromatin. They contain a large proportion of positively charged amino acids, while DNA is negatively charged. These opposite charges create a high-binding affinity structure between histones and DNA, called the *nucleosome*.

Histone proteins can be classified into *core* histones (H2A, H2B, H3, H4) and *linker* histone (H1). The eight histones in the core are arranged into a (H3)2(H4)2 tetramer and a pair of H2A-H2B dimers, called the *histone octamer*. Each core histone has an amino-terminal extension called *histone tail* which is the target for *post-translational modifications* (PTMs).

Research in epigenetics has shown that post-transcriptional modifications of core histones are involved in a variety of essential regulatory processes in the cell, including transcription control (see, e.g. Barski et al., 2007; VerMilyea et al., 2009; Zhang and Reinberg, 2001). Epigenetics factors (including PTMs and DNA methylation) and the complex interaction between nucleosomes and transcription factors are among the most critical factors influencing gene expression. Histone PTMs also have an indirect effect on gene expression by influencing the interactions between transcription factors and chromatin-modifying enzymes (Benveniste et al., 2014). In fact, some studies (e.g. Dong et al., 2012; Karlić et al., 2010) have shown that gene expression can be accurately predicted from the knowledge of histone tail PTMs.

In Benveniste *et al.* (2014), the authors investigated the opposite problem, that is, the prediction of histone PTMs from the knowledge of transcription factors (TF) binding and the underlying DNA sequence data. They reported that histone PTMs can be predicted accurately from the knowledge of TF binding either at the promoter or distal regulatory regions for three different human cell lines. They also showed that the prediction from TFs is more accurate than the prediction from the DNA sequence alone, and that the predictive power of TF binding data can be extended to predict histone modifications across cell lines. They suggested that interactions between TFs and histone-modifying enzymes might be important in driving the deposition of histone modifications.

Other works on the prediction of histone PTMs include DeepHistone (Yin et al., 2019), DanQ (Quang and Xie, 2016), DeepSEA (Zhou and Troyanskaya, 2015) and gkm-SVM (Lee et al., 2015), although the latter three are more general. DeepHistone is a convolutional neural network that can predict seven histone PTMs from the DNA sequence and DNase-Seq data for fifteen ENCODE cell lines. In Yin *et al.* (2019), the authors of DeepHistone show that their method outperformed DanQ, DeepSEA and gkm-SVM, but they did not compare to the regression model in (Benveniste *et al.*, 2014).

In this work, we propose a deep learning architecture called DeepPTM for predicting histone PTMs from TF binding and DNA sequence data. Extensive experimental results show that our neural network achieves a prediction accuracy substantially higher than the logistic regression model proposed in Benveniste *et al.* (2014) and the CNN used in DeepHistone (Yin *et al.*, 2019). The competitive advantage of our framework lies in synergistic use of deep learning combined with an effective data cleaning pre-processing step. Our framework has also enabled the discovery that the knowledge of a small subset of transcription factors (which are histone-PTM- and cell-type-specific) can provide essentially the same prediction accuracy that can be obtained using all the transcription factors data. Additional insights on which TFs are strongly positively (or negatively) correlated with the model's prediction were provided by the interpretability analysis of our deep learning model using the SHAP framework.

2 Materials and methods

2.1 Data collection

Human epigenetic and genetic data were obtained from ENCODE (ENCODE Project Consortium, 2004), as follows. Exactly 29 828 unique protein-coding transcription start sites (TSS) for the human reference genome were obtained from the ENSEMBL database. Histone tail modifications in the proximity of these TSS were assigned based on ChIP-Seq analysis for three ENCODE Tier 1 cell lines, namely H1 ES cells, K562 erythroleukemia cells and GM12878 lymphoblastoid cells. We considered three histone PTMs that are known to be relevant for transcription, namely H3K4me3, H3K9ac, H3K27ac for all three cell lines. For cell line H1, we also considered H3K27me3.

As it was done in Benveniste *et al.* (2014), we have focused on histone PTMs at gene promoters (rather than genome-wide) to obtain an homogeneous set of training samples. Given a histone PTM, each unique transcription start site (TSS) for a protein-coding gene was assigned a positive label if a ChIP-Seq peak for that PTM was detected within a 100 bp window center at the TSS (negative otherwise). In ChIP-Seq, (i) DNA-bound proteins are cross-linked to their DNA, which is then fragmented, (ii) DNA fragments those that are not cross-linked with proteins are removed by immuno-precipitation and (iii) short reads from either ends of cross-linked DNA fragments are produced. We aligned ChIP-Seq reads to the human genome using BWA, then used MACS2 (Zhang et al., 2008) using a false-discovery rate of 0.01 to detect peaks. As a result of this process, the proportion of positive/negative examples in the training set for cell line H1 ranges from 8.3% to 39.8% (see Supplementary Table S1).

The DNA sequence data were obtained following the protocol in Benveniste *et al.* (2014). We extracted the sequence ± 2000 bp upstream and downstream of each TSS. Then, we computed 6-mer counts for these 4000 bp-long sequences. We combined counts of 6-mers which are the reverse-complement of each other, ending up with 2080 6-mer counts. As result of this process, the DNA sequence dataset was represented by 29 828 vectors of dimension 2080. Each vector had a positive or negative label for a particular pair of (histone PTM, cell line) according to the presence of a corresponding ChIP-Seq peak around the TSS (as described above).

The transcription factor (TF) dataset was obtained following the protocol in Benveniste *et al.* (2014). First, ChIP-Seq data from the ENCODE project was filtered to remove proteins that lacked sequence-specific DNA-binding transcription factor activity. Overall, 30 TFs were assayed in H1 cells, 45 in K562 and 51 in GM12878. Among these, 17 TFs were assayed in all three cell lines. ChIP-Seq reads were aligned to the human genome, then the number of reads mapped within 2000 bp of the TSS were counted. Raw read counts were normalized by dividing the counts by the total number of reads from the control dataset. For the H1 cell line, we considered 29 828 TSS and 30 normalized read counts. For the K562 cell line, we had 45 normalized read counts. For the GM12878 cell line, we considered 51 normalized read counts. Again, each vector had a positive or negative label for a particular pair of (histone PTM, cell line) according to the presence of a corresponding ChIP-Seq peak around the TSS (as described above).

DNA hypersensitivity data, in the form of DNase-Seq normalized read counts, was obtained from ENCODE in bigwig format (ENCODE Project Consortium, 2004). Binary bigwig data were converted to textual BedGraph using bigWigToBedGraph.

2.2 Neighborhood cleaning rule

To gain insights into the structural properties of the training examples, we generated 2D projections for the feature vectors in the training set. For the transcription factor binding sites obtained from ChIP-Seq data, k-dimensional vectors and their respective labels for all histone PTMs ($k = 30$ for the H1 cell line, $k = 45$ for the K562 cell line and $k = 51$ for the GM12878 cell line) were processed using t-SNE (van der Maaten and Hinton, 2008) with default settings (n_components = 2, perplexity = 30.0, learning_rate = 200.0, n_iter = 1000, etc.). Figure 1 shows the results for cell line H1. Observe that positive examples (green) and negative examples (red) are not well separated. The same lack of separation can be observed on t-SNE plots for H3K9ac in cell line K562, H3K4me3 and H3K9ac in cell line GM12878 (Supplementary Fig. S1). Similarly, the t-SNE plots for the DNA sequence features for H3K4me3 in H1 cells, H3K9ac in K562 cells and H3K27ac in GM12878 cells, show that positive and negative examples are equally not-well separated (Supplementary Fig. S2). Based on these observations, we decided to discard a limited number training examples that could hamper the learning inference.

We employed the *neighborhood cleaning rule* (NCL) introduced in Laurikkala (2001). NCL is a method that is expected to remove noisy training samples from the majority class. For a two-class problem, the algorithm works as follows. For each example, E_i in the training set, identify its three nearest neighbors. If E_i belongs to the majority class and its three nearest neighbors contradicts the label of E_i, then E_i is removed. If E_i belongs to the minority class and its three nearest neighbors contradict the label of E_i, then the nearest neighbors that belong to the majority class are removed. Figure 2 shows a portion of the t-SNE projection of the data before and after applying NCL. Observe how the data is much better separated after NCL. The positive/negative ratio in the training set after NCL are reported in Supplementary Tables S2 (TF-data) and S3 (DNA sequence data).

2.3 Choice of the classifier and training

The authors of Benveniste *et al.* (2014) chose to use a logistic regression (LR) classifier because it produced interpretable weights that were used to verify that their method recapitulated known relationships between TF and histone PTMs.

Here, for modeling of the DNA sequence data, we have considered (i) a CNN-LSTM architecture and (ii) feed-forward fully connected neural network. The CNN-LSTM was fed the DNA sequence upstream and downstream of each TSS, as it was done (Quang and Xie, 2016). Instead, the input to the fully connected neural network was the 2080-dimensional vector composed of the 6-mer counts as explained in Section 2.1. We determined that the fully connected model performed better than the CNN when trained on the 'clean' data after Neighborhood Cleaning Rule. Henceforth, the feed-forward fully connected model was used in all the experiments below. An additional advantage of the feed-forward fully connected neural network is that it used exactly the same set of features that were used in Benveniste *et al.* (2014), which allows us to isolate the impact of the cleaning step and the type of classifier in our conclusions. The AUPR comparative analysis for the CNN-LSTM model and fully connected neural network is shown in Supplementary Figure S3.

The classifier for TF binding data is a feed-forward fully connected neural network because it vastly improved the prediction accuracy compared to LR and Gradient Boosted Classifier (GBC), as shown in the experimental results in Section 2.7. Instead, a GBC was used to determine the *minimal sets* of TFs (Section 3.1) because it provides interpretable weights (i.e. Gini importance).

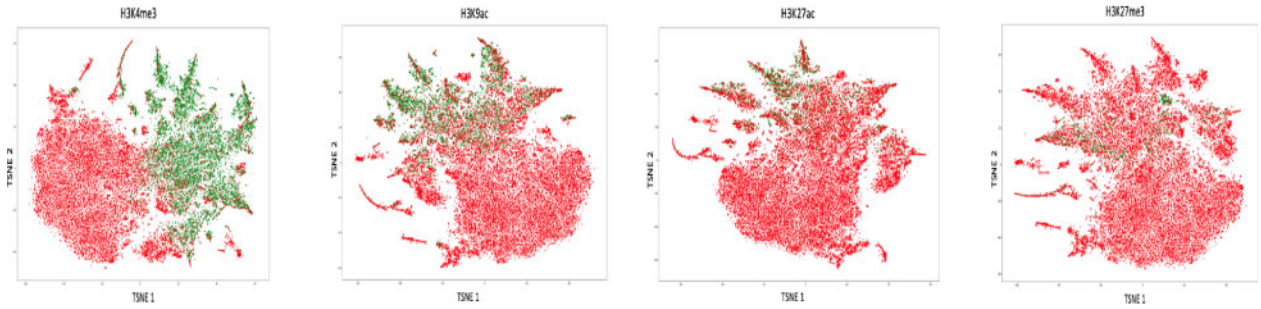

Fig. 1. Two-dimensional projection using t-SNE for H1-H3K4me3, H1-H3K9ac, H1-H3K27ac and H1-H3K27me3 (left to right) on TF-ChIP-Seq data for cell line H1; red points are negative examples; greens points are positive examples

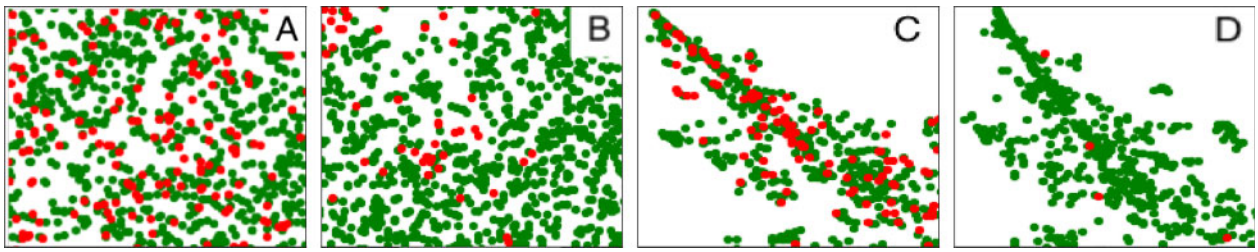

Fig. 2. Two-dimensional projection using t-SNE for H3K4me3 data before (**A**) and after NCL (**B**) and for H3K9ac data before (**C**) and after NCL (**D**) for cell line H1; red points are negative examples; greens points are positive examples

The training/testing procedure for both neural networks (TF ChIP-Seq data and DNA sequences) is as follows.

1. Let A be the input dataset (unbalanced).
2. Split A uniformly at random into a training set A' (80% of A), validation set B (10% of A) and a test set C (10% of A).
3. Let D the dataset after the Neighborhood Cleaning Rule on A'.
4. Train the model on D and validate on B.
5. Evaluate on C.

It is important to note that NCL can be applied only on the training set, and not on the test set because during testing one is not supposed to know the true labels (thus one cannot carry out NCL).

2.4 Model architecture for DNA sequences

To determine the optimal feed-forward fully connected architecture for DeepPTM to learn from the DNA sequence dataset we considered 1–5 hidden layers, each with a number of hidden units between 50 and 550. For dropout (Srivastava et al., 2014), we considered three values, namely, 0.3 corresponding to a weak dropout, 0.5 (medium dropout) and 0.7 (strong dropout). We evaluated batch sizes of $4, 8, 16, \ldots, 128$. The learning rate was chosen from the interval $[1e^{-5}, 1e^{-1}]$ evenly spaced in log scale. The optimal architecture for DNA sequences used three hidden layers (with 512, 180 and 70 hidden units, respectively), with a dropout of 0.5, a batch size of 128 and an initial learning rate of $1e^{-4}$.

2.5 Model architecture for TF data

To determine the optimal feed-forward fully connected architecture to learn from the TF ChIP-Seq dataset, we carried out the same procedure described in the previous subsection. For the TF dataset, the optimal architecture of DeepPTM has three hidden layers (with 256, 180 and 60 hidden units, respectively), with a dropout of 0.3, a batch size of 16 and an initial learning rate of $1e^{-4}$. The architecture DeepPTM for the TF dataset is illustrated in Figure 3.

In both architectures (DNA and TF), we used the procedure in Glorot and Bengio (2010) to initialize the weights, ReLU as the activation function for the hidden layers (LeCun et al., 2015), a sigmoid activation function in the output layer, binary cross entropy (Equation 1 below) as the loss function and Adam as the optimizer

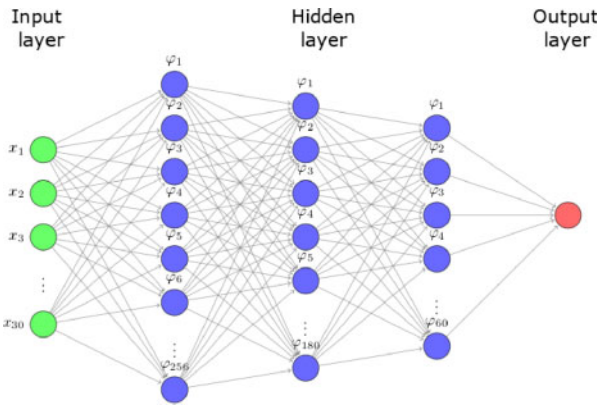

Fig. 3. The proposed architecture for DeepPTM for the TF ChIP-Seq dataset: one input layers, three hidden layers, one output layer

(Kingma and Ba, 2014). We allowed up to 90 epochs for training, with the possibility of early stopping when the value of loss function did not improve over ten iterations. For model architecture selection and hyper-parameter tuning, we followed (Alipanahi et al., 2015; Bergstra and Bengio, 2012; Angermueller, 2017; Quang and Xie, 2016; Singh et al., 2016). As said, the loss function used in DeepPTM is

$$L(y, y') = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(y'_i) + (1 - y_i)\log(1 - y'_i) \qquad (1)$$

where y is 1 for a positive example and 0 for a negative example, y' is classifier's predicted label/probability and N is the total number of examples in the training set.

2.6 Prediction of histone PTMs from DNA sequence

To evaluate the prediction quality of our classifier, we computed the Area Under Precision-Recall curve (AUPR), the Area under ROC curve (AUROC) and accuracy. To quantify the stability of these predictions, we repeated each training/testing experiment five times and recorded the mean of each performance metric.

Table 1. Histone PTM prediction performance (mean over five iterations) of DeepPTM (after NCL on the training set) versus the logistic regression classifier (no NCL on the training set) on DNA sequence data

Cell line	PTM	AUPR		AUROC		Accuracy	
		DeepPTM	LR	DeepPTM	LR	DeepPTM	LR
H1	H3K4me3	0.904	0.848	0.955	0.918	0.923	0.856
H1	H3K9ac	0.729	0.649	0.920	0.867	0.827	0.825
H1	H3K27ac	0.653	0.447	0.880	0.828	0.825	0.861
H1	H3K27me3	0.417	0.332	0.887	0.806	0.908	0.902
K562	H3K4me3	0.858	0.751	0.918	0.865	0.891	0.814
K562	H3K9ac	0.836	0.735	0.923	0.865	0.831	0.822
K562	H3K27ac	0.823	0.705	0.907	0.853	0.852	0.821
GM12878	H3K4me3	0.831	0.693	0.911	0.845	0.825	0.800
GM12878	H3K9ac	0.823	0.700	0.925	0.855	0.860	0.814
GM12878	H3K27ac	0.780	0.647	0.901	0.827	0.830	0.806

Table 2. Histone PTM prediction performance (mean over five iterations) of DeepPTM (after NCL on the training set) versus the logistic regression classifier (no NCL on the training set) on TF ChIP-Seq data

Cell line	PTM	AUPR		AUROC		Accuracy	
		DeepPTM	LR	DeepPTM	LR	DeepPTM	LR
H1	H3K4me3	0.913	0.885	0.962	0.950	0.909	0.888
H1	H3K9ac	0.795	0.729	0.952	0.921	0.867	0.844
H1	H3K27ac	0.685	0.545	0.944	0.909	0.898	0.878
H1	H3K27me3	0.668	0.447	0.902	0.877	0.914	0.922
K562	H3K4me3	0.922	0.901	0.969	0.961	0.915	0.913
K562	H3K9ac	0.921	0.906	0.972	0.969	0.919	0.919
K562	H3K27ac	0.905	0.887	0.967	0.965	0.904	0.912
GM12878	H3K4me3	0.898	0.829	0.958	0.942	0.902	0.883
GM12878	H3K9ac	0.915	0.873	0.988	0.962	0.930	0.906
GM12878	H3K27ac	0.889	0.855	0.971	0.956	0.906	0.901

We compared the performance of DeepPTM to the logistic regression classifier (LR) described in Benveniste *et al.* (2014).

First, we investigated the impact of the Neighborhood Cleaning Rule (NCL) on the performance of LR and DeepPTM. Supplementary Table S4 shows that LR performs better after NCL in predicting all histone PTMs except H3K9ac. DeepPTM also performs better after NCL for all histone PTM predictions (see Supplementary Table S5). In all experiments below (unless otherwise noted) NCL was used to clean the training set.

Table 1 shows AUPR, AUROC and accuracy for DeepPTM and LR when trained and tested on DNA sequence data (represented by the 2080 6-mer counts). Here, NCL was used as a pre-processing step for DeepPTM, but not for LR. Observe that the AUPR for DeepPTM ranges from 0.417 (for H3K27me3 on cell line H1) to 0.904 (for H3K4me3 on cell line H1). DeepPTM's predictions outperform LR on all histone PTMs. DeepPTM's AUPR improvement over LR ranges from 6.83% (for H3K4me3 on cell line H1) to 25.60% (for H3K27me3 on cell line H1).

To make a fair comparison we have also used NCL on the training set for the LR classifier. Supplementary Table S6 shows AUPR, AUROC and accuracy for DeepPTM and LR when trained and tested on DNA sequence data. Observe that AUPR for LR ranges from 0.412 (for H3K27me3 on cell line H1) to 0.850 (for H3K4me3 on cell line H1). DeepPTM's predictions outperform LR on all histone PTMs. DeepPTM's AUPR improvement over LR ranges from 7.75% (for H3K27ac on cell line H1) to 19.05% (for H3K4me3 on cell line GM12878).

2.7 Prediction of histone PTMs from TF binding data

As we did in the previous section, we recorded mean for AUPR, AUROC and accuracy over five iterations of each experiment. First, we investigated the impact of NCL on TF ChIP-Seq data. Both DeepPTM and LR performed better on cleaned data than on raw data for all histone PTM predictions (Supplementary Table S7 for LR, and Supplementary Table S8 for DeepPTM).

Table 2 reports these performance metrics for DeepPTM and LR when trained and tested on TF binding data, which is represented as normalized ChIP-Seq read counts. As we did in the previous section, we compared the performance of DeepPTM on the test set without NCL.

Observe that the AUPR for DeepPTM ranges from 0.668 (for H3K27me3 on cell line H1) to 0.913 (for H3K4me3 on cell line H1). DeepPTM's predictions outperform LR on all histone PTMs. DeepPTM's AUPR improvement over LR ranges from 1.56% (for H3K9ac on cell line K562) to 49.44% (for H3K27me3 on cell line H1).

As we did in the previous section, we also used NCL on the training set for LR. Supplementary Table S9 shows AUPR, AUROC and accuracy for DeepPTM and LR when trained and tested on TF-ChIP-Seq data. Observe that AUPR for LR ranges from 0.520 (for H3K27me3 on cell line H1) to 0.890 (for H3K4me3 on cell line H1). Here, DeepPTM's predictions outperform LR on all histone PTMs. DeepPTM's AUPR improvement over LR ranges from 1.54% (for H3K9ac on cell line K562) to 28.46% (for H3K9ac on cell line H1).

We tested the impact of the choice of the window size used to assign PTMs to a particular TSS (the default window size was 100 bp). Supplementary Table S10 shows very minor difference in DeepPTM's AUPR for other choices of the window size (500 bp, 1 kbp and 2 kbp).

We then compared the predictive performance of DeepPTM trained on TF ChIP-Seq binding data (Table 2) against its performance when trained on DNA sequence data (Table 1). Observe that

Table 3. Gradient Boosted classifier's prediction performance (mean AUPR, AUROC and accuracy over five iterations) using TF binding data for the H1 cell line (NCL on the traning set)

Cell line	PTM	AUPR	AUROC	Accuracy
H1	H3K4me3	0.905	0.951	0.912
H1	H3K9ac	0.781	0.929	0.848
H1	H3K27ac	0.646	0.921	0.887
H1	H3K27me3	0.585	0.892	0.896

Table 4. DeepPTM's prediction performance (mean AUPR, AUROC and accuracy over five iterations) using DNase-seq for cell line H1 (NCL on the traning set)

PTM	AUPR	AUROC	Accuracy
H3K4me3	0.903	0.974	0.938
H3K9ac	0.719	0.901	0.820
H3K27ac	0.554	0.860	0.810
H3K27me3	0.240	0.725	0.875

DeepPTM performed better when trained on TF ChIP-Seq data for all histone marks and on all cell lines.

As mentioned in Section 2.3, we justified the choice of a feed-forward fully connected network over either a Gradient Boosted classifier (GBC) or an LR model based on a comparative analysis of these classifiers' performance on TF ChIP-Seq data. To test the GBC, we used Python sklean library. We evaluated several choices for the two key GBC parameters, namely n_estimator $= 40, 60, 80, \ldots, 100$, max_depth$= 3, 4, 5, \ldots, 10$. The GBC learning rate was chosen in the interval $[1e^{-5}, 1e^{-1}]$ evenly spaced in log scale. We found that n_estimator$= 100$, max_depth$= 3$ and learning rate of $1e^{-1}$ were optimal for GBC on the H1 dataset. The performance of GBC on predicting PTMs on cell line H1 is shown in Table 3 and Supplementary Figure S6. Experimental results in Tables 2 and 3 show that DeepPTM performs better than the GBC.

2.8 Prediction of histone PTMs from DNA hypersensitivity data

As demonstrated by comparing Tables 1 and 2, transcription factor data enables DeepPTM to achieve a higher accuracy in predicting PTMs than the DNA sequence. An important question is whether DNA accessibility (i.e. DNase I hypersensitivity data) at the TF binding sites would allow a similar predictive performance, as it is done in DeepHistone (Yin *et al.*, 2019). To answer this question, we used the same feed-forward fully connected neural network model that we designed for TF binding data and trained it on DNase-Seq data for cell line H1. DNase-Seq normalized read counts were collected as explained in Section 2.1, then fed into DeepPTM for predicting histone PTMs. Experimental results are shown in Table 4 and Supplementary Figure S7. By comparing Tables 2 and 4, we concluded that TF binding data allows a more accurate prediction of PTMs than DNA hypersensitivity data.

2.9 Prediction of histone PTMs from the DNA sequence and TF binding data (combined)

Here, we combined the two neural networks for DNA sequence and TF binding data into one Ensemble classifier by feeding their prediction scores into a Logistic Regression classifier (see Fig. 4). The prediction provided by the Logistic Regression is the outcome of the combined model. For the Logistic Regression classifier, we have used the sklearn library with default settings (i.e. $C = 1$, Penalty $=$ l2, etc).

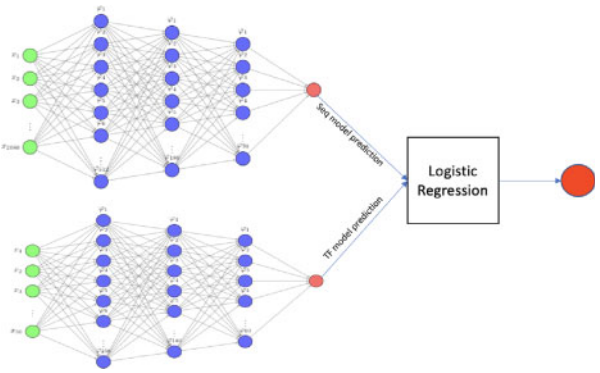

Fig. 4. The proposed architecture for the Ensemble classifier; the architecture on the top deals with the DNA sequence features; the architecture at the bottom deals with the TF binding data; the outputs of the two models are combined using a Logistic Regression classifier

Table 5. DeepPTM ensemble classifier's prediction performance (AUPR, AUROC and accuracy over five iterations) using both DNA sequence and TF binding data (NCL on the traning set) on cell line H1; DeepHistone's AUPR and AUROC were obtained from Tables 2 and 3 in Yin *et al.* (2019)

PTM	DeepPTM			DeepHistone	
	AUPR	AUROC	Accuracy	AUPR	AUROC
H3K4me3	0.961	0.974	0.938	0.843	0.9459
H3K9ac	0.865	0.963	0.892	0.727	0.9039
H3K27ac	0.822	0.955	0.907	0.771	0.9137
H3K27me3	0.765	0.925	0.922	0.666	0.8896

The prediction performance of the Ensemble classifier on cell line H1 using both DNA sequence and TF binding data is shown in Table 5. Supplementary Figure S8 shows the individual precision/recall curves of the Ensemble classifier for each PTMs. By comparing these experimental results to the results in Tables 1 and 2, we observed that DeepPTM's performance using Ensemble classifier using both features is higher than its performance using either DNA sequence or TF binding data. Also observe that DeepPTM outperforms DeepHistone, with the caveats that (i) DeepHistone uses DNase-Seq data while DeepPTM employs TF ChIP-Seq data (both use the DNA sequence) and (ii) DeepHistone employs one CNN to predict seven histone PTMs simultaneously, whereas DeepPTM trains one network for each histone PTM.

2.10 Prediction of histone PTMs across different cell lines using TF binding data

Given the excellent performance of DeepPTM, we tested its performance across cell lines, i.e. we trained DeepPTM on one cell line and tested on another. For this purpose, we selected as features the 17 TFs which are common to all cell lines. Observe in Table 6 that the prediction performance (AUPR) of DeepPTM for H3K4me3 across cell lines is lower than the prediction performance on the same cell line. The same is true for H3K9ac (Table 7) and for H3K27ac (Table 8). This suggests that cell type specific TFs are responsible for determining histone PTM profiles, which recapitulate the finding in Benveniste *et al.* (2014). Observe however, that the prediction performance of DeepPTM using the common 17 TFs is almost the same as using all the TFs. This is not the case for the logistic regression classifier. Finally, observe that DeepPTM significantly outperforms LR on all cross cell line predictions. Precision/recall curves for cross cell line prediction for H3K4me3 are shown in Supplementary Figure S9. Experimental results for H3K9ac and H3K27ac are shown in Supplementary Figures S10 and S11, respectively.

Table 6. DeepPTM's predictive performance (mean AUPR) across cell lines for H3K4me3 (NCL on the traning set)

	Test on H1		Test on K562		Test on GM12878	
	DeepPTM	LR	DeepPTM	LR	DeepPTM	LR
Train on H1	0.904	0.878	0.855	0.777	0.746	0.651
Train on K562	0.871	0.793	0.901	0.882	0.670	0.479
Train on GM12878	0.882	0.848	0.871	0.825	0.878	0.790

Table 7. DeepPTM's predictive performance (mean AUPR) across cell lines for H3K9ac (NCL on the traning set)

	Test on H1		Test on K562		Test on GM12878	
	DeepPTM	LR	DeepPTM	LR	DeepPTM	LR
Train on H1	0.851	0.681	0.850	0.7638	0.720	0.661
Train on K562	0.801	0.606	0.901	0.879	0.582	0.453
Train on GM12878	0.750	0.596	0.852	0.799	0.881	0.803

Table 8. DeepPTM's predictive performance (mean AUPR) across cell lines for H3K27ac (NCL on the traning set)

	Test on H1		Test on K562		Test on GM12878	
	DeepPTM	LR	DeepPTM	LR	DeepPTM	LR
Train on H1	0.680	0.480	0.810	0.719	0.670	0.604
Train on K562	0.587	0.416	0.901	0.850	0.530	0.423
Train on GM12878	0.480	0.376	0.823	0.758	0.863	0.735

3 Interpretability analysis of DeepPTM

Understanding the reasons behind a classifier's performance is as critically important as optimizing the classifier's accuracy. Unfortunately, deep learning models are notoriously challenging to interpret. In this section, we propose two methods that can provide some insights into DeepPTM's predictive performance. First, we used the concept of Gini importance to identify minimal sets of TFs (which are histone-PTM- and cell-type-specific) that can provide almost the same prediction accuracy of the full dataset of TFs. Second, we used the SHAP framework to determine which TFs are the most influential for the prediction, and which TF are directly (or inversely) correlated with the prediction of a particular histone PTM.

3.1 Minimal Sets of TFs using Gini importance

To obtain the minimal feature set on the TF binding data, we computed the *Gini importance* on the feature set. The *Gini importance* depends on the number of times a feature is used to split a node in a decision tree, weighted by the number of samples it splits (Breiman, 2001). We computed the *minimal feature set* of features as follows.

1. Sort features by Gini importance in descending order.
2. Let S be the top feature.
3. Train and test DeepPTM on the set S.
4. If the prediction accuracy of the classifier using S is within 1% of the prediction accuracy based on all features then STOP, else add the next feature to S and repeat from (3).

Supplementary Figure S12 illustrates the process of producing minimal sets. The Gini importance for each TF over all pairs of (PTM, cell line) are shown in Supplementary Tables S11–S20.

Table 9 shows the minimal sets for each pair of cell type and histone PTM. Recall that the TFs in each minimal set provides almost the same prediction performance of the entire available set of TFs.

Table 9. Minimal sets of TFs for cell specific histone PTM

Cell line	PTM	Minimal set
H1	H3K4me3	SIN3A, MAX
H1	H3K9ac	SIN3A, TCF12, NRSF, CREB1, YY1, SP4, SIX5
H1	H3K27ac	YY1, SIN3A, ATF2, CREB1, TCF12, SP4, NRSF, SP1, CTCF
H1	H3K27me3	TCF12, SIN3A, E2F6, ATF2, GABP, NRSF, SIX5, TEAD4, SP4, MAX, BACH1, SP1, CMYC, CREB1, YY1
K562	H3K4me3	E2F6, MAX, TEAD4, ATF3, GATA2, ZNF263, CREB1, E2F4
K562	H3K9ac	E2F6, MAX, CMYC, TEAD4, CTCF, ATF3
K562	H3K27ac	MAX, CMYC, TEAD4, E2F6, CTCF
GM12878	H3K4me3	ATF2, BATF, BCL3, BHLHE40, CFOS, E2F4, EBF1, ELF1, ELK1, ERRA, ETS1, FOXM1, IKZF1, IRF3, IRF4
GM12878	H3K9ac	SP1, NFATC1, CREB1, TCF3, STAT3, BATF, ELF1, RUNX3, POU2F2
GM12878	H3K27ac	NFATC1, SP1, CREB1, ELF1, CTCF, TCF3, BCL3

For example, SIN3A and MAX alone provide a prediction accuracy for H3K4me3 on the H1 cell line within 1% of the accuracy obtained using all thirty TFs.

Figure 5 shows Venn diagrams for the minimal sets of each histone PTM for a particular cell line. Observe that H3K4me3 can be predicted by only two TF for cell line H1; the other PTMs require significantly more TFs. In contrast, the minimal set for H3K4me3 is the largest for cell line GM12878. Observe that (i) SIN3A is shared by all the histone PTMs for cell line H1, (ii) TEAD4, E2F6, MAX are shared by all the histone PTMs for cell line K562 and (iii) ELF1 is shared by all the histone PTMs for cell line GM12878.

Figure 6 shows Venn diagrams for the minimal sets of each cell lines for a particular histone PTM. Observe that there are no TFs for H3K4me3 and H3K9ac that are shared by all three cell lines, i.e. the minimal set for H3K4me3 and H3K9ac are highly specific to that cell-line. Also observe that there is only one TF for H3K27ac shared by all three cell lines, namely CTCF.

3.2 Importance analysis of TFs using SHAP

We used the SHAP framework (Lundberg and Lee, 2017) to identify what TFs contribute the most in DeepPTM's prediction, and whether they are positively or negatively correlated with the prediction of a particular histone PTM. The idea proposed in Lundberg and Lee (2017) is to devise a simpler *explanation model* that is interpretable and that can approximate well the original (complex) model.

For the interpretability analysis of DeepPTM, we have used DeepSHAP which is an efficient approximation algorithm for SHAP for deep learning models that relies on DeepLIFT (Shrikumar et al., 2016), as described in Lundberg and Lee (2017). The implementation we used here differs from the original DeepLIFT because (i) it uses a distribution of background samples instead of a single reference value, and (ii) it uses Shapley's equations to linearize non-linear components such as max, softmax, products, divisions, etc. DeepLIFT approximates SHAP values under the assumption that the input features are independent of one another and the deep learning model is linear. The back-propagation rules that define how each component is linearized are heuristically chosen. Since DeepLIFT is an additive feature attribution method that satisfies local accuracy and missingness, the Shapley values represent the only attribution values that satisfy consistency (Lundberg and Lee, 2017). This motivates our choice in using DeepLIFT as a compositional approximation for SHAP values.

We have used the Python library shap that implements the method in Lundberg and Lee (2017). The method DeepExplainer

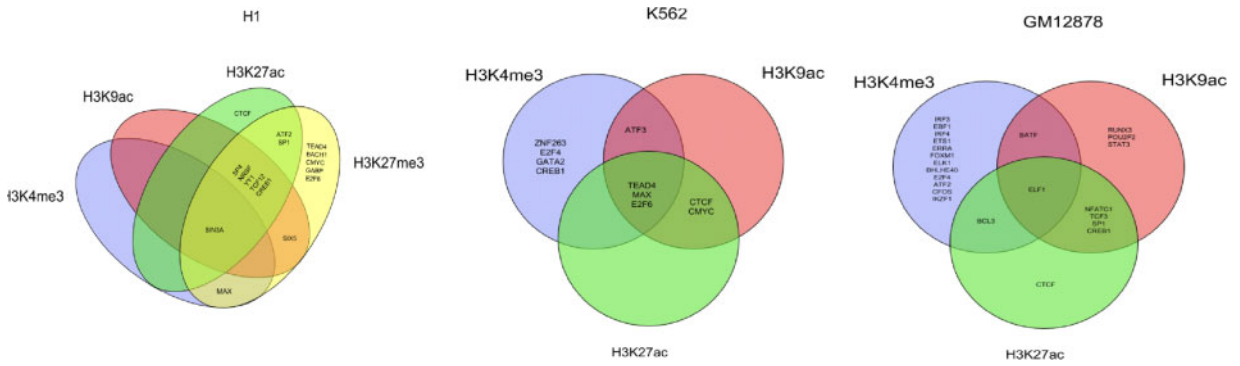

Fig. 5. Minimal sets of TFs of each histone PTM for three cell lines

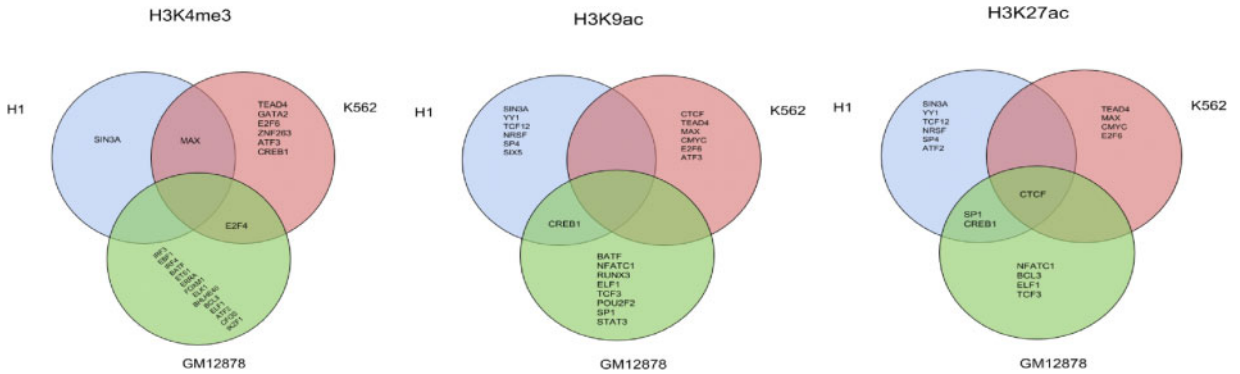

Fig. 6. Minimal sets of TFs of each cell line for the three PTMs

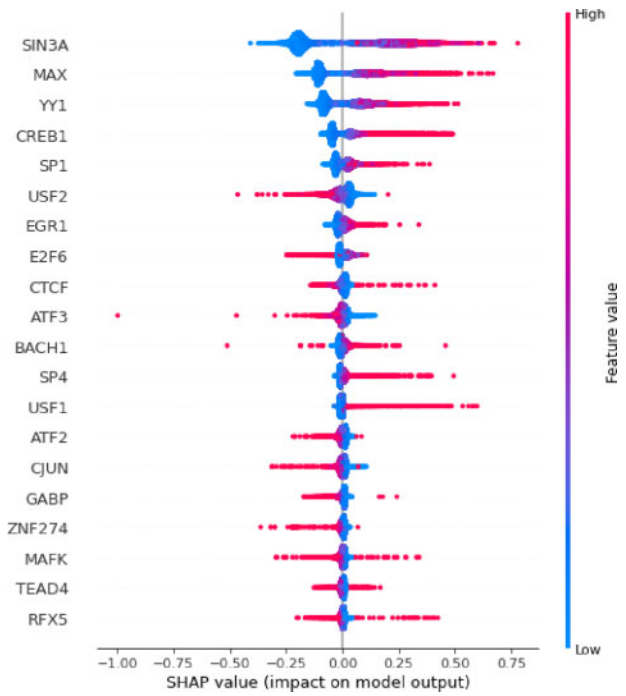

Fig. 7. Feature importance plot for TF binding data for the prediction H3K4me3 in cell line H1

was used to explain DeepPTM. The function summaryplot produces the feature importance plot that illustrates the features' contributions to DeepPTM's prediction.

The feature importance plot of DeepPTM for the prediction of H3K4me3 from TF binding data in cell line H1 is shown in Figure 7. Features (in this case, transcription factors) are ranked in descending order. Features at the top contribute more to the model prediction than the features at the bottom, while the scatter plot on the horizontal axis illustrates whether a feature is directly (red) or inversely (blue) correlated with the prediction. For example, a high normalized read count of the transcription factor SIN3A has a strong positive impact on the prediction of H3K4me3 for the H1 cell line. Similarly, USF2 has the 6th strongest impact on the prediction, but it is negatively correlated with the prediction of H3K4me3, that is, the higher is the binding, the less likely one should observe H3K4me3. Observe that SP1, MAX, CREB1, TR4, BHLHE40, NFE2, POU2F2, SIX5, IRF4, NFATC1, TCF12, NFIC and GABP are positively correlated with H3K4me3 PTM for H1 while STAT3, NFE2, TCF3, NFYA BCL3 are negatively correlated.

We carried out similar analysis for other cell lines and histone PTMs. Supplementary Figure S13–S21 show the corresponding feature importance plot obtained via SHAP. We also listed the top five positively and negatively correlated TFs for each PTM and each cell line in Table 10. Observe that transcription factor MAX is almost always positively correlated with these histone PTMs.

4 Conclusions

We proposed a deep learning architecture to predict histone PTMs from TF binding and DNA sequence data. We determined that (i) histone PTM can be predicted more accurately from TF ChIP-Seq binding data than from DNA sequence, (ii) histone PTMs can be predicted more accurately from TF ChIP-Seq binding data than from DNase-Seq data, (iii) histone PTMs can be predicted more accurately by combining DNA sequence and TF ChIP-seq binding data, (iv) DeepPTM's prediction accuracy is substantially higher than the logistic regression model proposed in Benveniste *et al.* (2014) and DeepHistone (with the caveats listed in Section 2.9). The

Table 10. Positively and Negatively Correlated TFs for cell specific histone PTM

Cell line	PTM	Positively correlated TFs	Negatively correlated TFs
H1	H3K4me3	SP1, MAX, CREB1, TR4, BHLHE40	STAT3, NFE2, TCF3, NFYA, BCL3
H1	H3K9ac	SIN3A, YY1, MAX, CREB1, SP1	TCF12, USF2, CJUN, ATF3
H1	H3K27ac	YY1, SIN3A, SP1, CREB1, ATF2	USF2, TCF12, ATF3, NANOG
H1	H3K27me3	TCF12, E2F6, GABP, MAX, BACH1	SIN3A, ATF2, SIX5, SP1, SP4
K562	H3K4me3	MAX, E2F4, HCFC1, CREB1, SP1	GATA2, GATA1, ETS1, ATF3, USF2
K562	H3K9ac	E2F6, MAX, E2F4, CMYC, GABP	GATA2, ATF3, GATA1, ETS1, ZNF274
K562	H3K27ac	MAX, E2F6, CMYC, GABP, E2F4	ATF3, GATA2, ETS1, CTCF, ZNF274
GM12878	H3K4me3	SP1, MAX, ELF1, CREB1, TR4	STAT3, NFE2, TCF3, NFYA, BCL3
GM12878	H3K9ac	ELF1, SP1, CREB1, MAX, POU2F2	STAT3, NFYA, NFYB, TCF3
GM12878	H3K27ac	ELF1, SP1, CREB1, POU2F2, NFIC	STAT3, NFYA, FOXM1, NFE2, ZEB1

competitive advantage of DeepPTM lies in synergistic use of deep learning combined with an effective data cleaning pre-processing step. Our framework has also enabled the discovery that the knowledge of a small subset of transcription factors (which are histone-PTM- and cell-type-specific) can provide almost the same prediction accuracy that can be obtained using all the transcription factors. The SHAP-based importance analysis showed which TF are most influential for the prediction, and which TFs are directly (or inversely) correlated with the prediction of a particular histone PTM.

Acknowledgements

The authors thank Prof. Duncan Sproul (University of Edinburgh) for providing help and for useful discussions.

Funding

This work was supported in part by the U.S. National Science Foundation [IOS-1543963, IIS-1814359] and by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, Genomic Science Program under Award Number DE-SC0019093.

Conflict of Interest

none declared.

References

Alipanahi,B. *et al.* (2015) Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, **33**, 831–838.

Barski,A. *et al.* (2007) High-resolution profiling of histone methylations in the human genome. *Cell*, **129**, 823–837.

Benveniste,D. *et al.* (2014) Transcription factor binding predicts histone modifications in human cell lines. *Proc. Natl. Acad. Sci. USA*, **111**, 13367–13372.

Bergstra,J. and Bengio,Y. (2012) Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, **13**, 281–305.

Breiman,L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.

Angermueller,C. *et al.* (2017) DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.*, **18**, 67.

Dong,X. *et al.* (2012) Modeling gene expression using chromatin features in various cellular contexts. *Genome Biol.*, **13**, R53.

ENCODE Project Consortium. *et al.* (2004) The ENCODE (ENCyclopedia of DNA elements) project. *Science*, **306**, 636–640.

Glorot,X. and Bengio,Y. (2010) Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS*, Vol. **9**, pp. 249–256

Karlić,R. *et al.* (2010) Histone modification levels are predictive for gene expression. *Proc. Natl. Acad. Sci. USA*, **107**, 2926–2931.

Kingma,D.P. and Ba,J. (2014) Adam: a method for stochastic optimization. *arXiv:1412.6980*.

Laurikkala,J. (2001) Improving identification of difficult small classes by balancing class distribution. *Technical Report, A-2001-2*, University of Tampere.

LeCun,Y. *et al.* (2015) Deep learning. *Nature*, **521**, 436–444.

Lee,D. *et al.* (2015) A method to predict the impact of regulatory variants from DNA sequence. *Nat. Genet.*, **47**, 955–961.

Lundberg,S.M. and Lee,S.-I. (2017) A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4768–4777.

Quang,D. and Xie,X. (2016) DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.*, **44**, e107.

Shrikumar,A. *et al.* (2016) Not Just a Black Box: Learning important features through propagating activation differences. *arXiv preprint arXiv: 1605.01713*.

Singh,S. *et al.* (2016) Predicting enhancer-promoter interaction from genomic sequence with deep neural networks. *bioRxiv* doi:10.1101/085241.

Srivastava,N. *et al.* (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**, 1929–1958.

van der Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

VerMilyea,M.D. *et al.* (2009) Transcription-independent heritability of induced histone modifications in the mouse preimplantation embryo. *PLoS One*, **4**, e6086.

Yin,Q. *et al.* (2019) Deephistone: a deep learning approach to predicting histone modifications. *BMC Genomics*, **20**, 11–23.

Zhang,Y. *et al.* (2008) Model-based analysis of ChIP-Seq (MACS). *Genome Biol.*, **9**, R137.

Zhang,Y. and Reinberg,D. (2001) Transcription regulation by histone methylation: interplay between different covalent modifications of the core histone tails. *Genes Dev.*, **15**, 2343–2360.

Zhou,J. and Troyanskaya,O.G. (2015) Predicting effects of noncoding variants with deep learning–based sequence model. *Nat. Methods*, **12**, 931–934.