NetTIME: Improving Multitask Transcription Factor Binding Site Prediction with Base-pair Resolution

Ren Yi[a], Kyunghyun Cho[a,b,*], Richard Bonneau[a,b,c,d,*]

[a]*Department of Computer Science, New York University, New York, NY 10011, USA*
[b]*Center for Data Science, New York University, New York, NY 10011, USA*
[c]*Department of Biology, New York University, New York, NY 10003, USA*
[d]*Center for Computational Biology, Flatiron Institute, Simons Foundation, New York, NY 10010, USA*

Abstract

Motivation: Machine learning models for predicting cell-type-specific transcription factor (TF) binding sites have become increasingly more accurate thanks to the increased availability of next-generation sequencing data and more standardized model evaluation criteria. However, knowledge transfer from data-rich to data-limited TFs and cell types remains crucial for improving TF binding prediction models because available binding labels are highly skewed towards a small collection of TFs and cell types. Transfer prediction of TF binding sites can potentially benefit from a multitask learning approach; however, existing methods typically use shallow single-task models to generate low-resolution predictions. Here we propose NetTIME, a multitask learning framework for predicting cell-type-specific transcription factor binding sites with base-pair resolution.

Results: We show that the multitask learning strategy for TF binding prediction is more efficient than the single-task approach due to the increased data availability. NetTIME trains high-dimensional embedding vectors to distinguish TF and cell-type identities. We show that this approach is critical for the success of the multitask learning strategy and allows our model to make accurate transfer predictions within and beyond the training panels of TFs and cell types. We additionally train a linear-chain conditional random field (CRF) to classify binding predictions and show that this CRF eliminates the need for setting a probability threshold and reduces classification noise. We compare our method's predictive performance with several state-of-the-art methods, including DeepBind, BindSpace, and Catchitt, and show that our method outperforms previous methods under both supervised and transfer learning settings.

Availability: NetTIME is freely available at https://github.com/ryi06/NetTIME

Contact: rb133@nyu.edu

1. Introduction

Genome-wide modeling of non-coding DNA sequence function is among the most fundamental and yet challenging tasks in biology. Transcriptional regulation is orchestrated by transcription factors (TFs), whose binding to DNA initiates a series of signaling cascades that ultimately determine both the rate of transcription of their target genes and the underlying DNA functions. Both the cell-type-specific chromatin state and the DNA sequence affect the interactions between TFs and DNA *in vivo* [1]. Experimentally determining cell-type-specific TF binding sites is made possible through high-throughput techniques such as chromatin immunoprecipitation followed by sequencing (ChIP-seq) [2]. Due to experimental limitations, however, it is infeasible to perform ChIP-seq (or related single-TF-focused experiments) on all TFs across all cell types and organisms [1]. Therefore, computational methods for accurately predicting *in vivo* TF binding sites are essential for studying TF functions, especially for less well-known TFs and cell types.

*To whom correspondence should be addressed.

Preprint submitted to Elsevier

Multiple community crowdsourcing challenges have been organized by the DREAM Consortium[1] to find the best computational methods for predicting TF binding sites in both *in vitro* and *in vivo* settings [3, 4]. These challenges also set the community standard for both processing data and evaluating methods. However, top-performing methods from these challenges have revealed key limitations in the current TF binding prediction community. Generalizing predictions beyond the training panels of cell types and TFs can potentially benefit from multitask learning and increased prediction resolution. However, many existing methods still use shallow single-task models. Predictions generated from these methods typically have low resolution, and they cannot achieve competitive performance for prediction of binding regions shorter than 50 base pairs (bp), although the actual TF binding sites are considerably shorter [5].

1.1. Related work

Early TF binding prediction methods such as MEME [6, 7] focused on deriving interpretable TF motif position weight matrices (PWMs) that characterize TF sequence specificity. Amid rapid advancement in machine learning, however, growing evidence has suggested that sequence specificity can be more accurately captured through more abstract feature extraction techniques. For example, a method called DeepBind [8] used a convolutional neural network to extract TF binding patterns from DNA sequences. Several modifications to DeepBind subsequently improved model architecture [9] as well as prediction resolution [10]. Yuan et al. developed BindSpace, which embeds TF-bound sequences into a common high-dimensional space. Embedding methods like BindSpace belong to a class of representation learning techniques commonly used in natural language processing [12] and genomics [13, 14] for mapping entities to vectors of real numbers. New methods also explicitly model protein binding sites with multiple binding mode predictors [15], and the effect of sequence variants on non-coding DNA functions at scale [16, 17, 18].

In general, the DNA sequence at a potential TF binding site is just the beginning of the full DNA-function picture, and the state of the surrounding chromosome, the TF and TF-cofactor expression, and other contextual factors play an equally large role. This multitude of factors changes substantially from cell type to cell type. *In vivo* TF binding site prediction, therefore, requires cell-type-specific data such as chromatin accessibility and histone modifications. Convolutional neural networks as well as TF- and cell-type-specific embedding vectors have both been used to learn cell-type-specific TF binding profiles from DNA sequences and DNase-seq data [19]. The DREAM Consortium also initiated the ENCODE-DREAM challenge to systematically evaluate methods for predicting *in vivo* TF binding sites [4]. Apart from carefully designed model architectures, top-ranking methods in this challenge also rely on extensively curated feature sets. One such method, called Catchitt [20], achieves top performance by leveraging a wide range of features including DNA sequences, genome annotations, TF motifs, DNase-seq, and RNA-seq.

1.2. Current limitations

Compendium databases such as ENCODE [21] and Remap [22] have collected ChIP-seq data for a large collection of TFs in a handful of well-studied cell types and organisms [1]. Within a single organism, however, the ENCODE TF ChIP-seq collection is highly skewed towards only a few TFs in a small collection of well-characterized cell lines and primary cell types (Figure. S1). Knowledge transfer from well-known cell types and TFs is crucial for understanding less-studied cell types and TFs.

Top-performing methods from the ENCODE-DREAM Challenge typically adopt the single-task learning approach. For example, Catchitt [20] trains one model per TF and cell type. Cross cell-type transfer predictions are achieved by providing a trained model with input features from a new cell type. This approach can be highly unreliable as the chromatin landscapes between the trained and predicted cell types can be drastically different [23] and these differences can be functionally important [24]. Alternatively, each model can be trained on one TF using cell-type-specific data across multiple cell types of interest [25]. However, such models tend to assign high binding probabilities to common binding sites among training cell types without proper mechanisms to distinguish cell types. Very few methods have adopted the multi-task learning approach in which data from multiple cell types and TFs are trained jointly in order to improve the overall model performance. One multitask solution [16, 17, 18] involves training a multi-

[1] http://dreamchallenges.org/about-dream/

class classifier on DNA sequences, where each class represents the occurrence of binding sites for one TF in one cell type. This solution is suboptimal as it cannot generalize predictions beyond the training TF and cell-type pairs.

Sequence context affects TF binding affinity [26], and increasing context size can improve TF binding site prediction [16]. TF binding sites are typically only 4-20 bp long [5]; increasing the prediction resolution – the number of predictions a model makes given an input sequence – is therefore beneficial for experimental validation as well as *de novo* motif discovery. However, instead of identifying precise TF binding locations, existing methods mainly focus on determining the presence of binding sites. Predictions from these models therefore suffer from either low resolution or low context size, depending on the input sequence length.

In this work, we address the above challenges by introducing NetTIME (Network for TF binding Inference with Multitask-based condition Embeddings), a multitask learning framework for base-pair resolution prediction of cell-type-specific TF binding sites. NetTIME jointly trains multiple cell types and TFs, and effectively distinguishes different conditions using cell type-specific and TF-specific embedding vectors. It achieves base-pair resolution and accepts input sequences up to 1kb.

2. Approach

2.1. Feature and label generation

The ENCODE Consortium has published a large collection of TF ChIP-seq data, all of which are generated and processed using the same standardized pipelines [21]. We therefore collect our TF binding target labels from ENCODE to minimize data heterogeneity. Each replicated ENCODE ChIP-seq experiment has two biological replicates, from which two sets of peaks – conserved and relaxed – are derived; peaks in both sets are highly reproducible between replicates [27].

Compared to the relaxed peak set, the conserved peak set is generated with a more stringent threshold, and is generally used to provide target labels. However, the conserved peak set usually contains too few peaks to train the model efficiently. Therefore, we use both conserved and relaxed peak sets to provide target labels for training, and the conserved peak set alone for evaluating model performance.

To collect target labels that cover a wide range of cellular conditions and binding patterns, we first select 7 cell types (including 3 cancer cell types, 3 normal cell types, and 1 stem cell type) and 22 TFs (including 17 TFs from 7 TF protein families as well as 5 functionally related TFs.). Conserved and relaxed peak sets are collected from 71 ENCODE replicated ChIP-seq experiments conducted on our cell types and TFs of interest. Each of these TF ChIP-seq experiments is henceforth referred to as a condition. All peaks from these conditions form a set of information-rich regions where at least one TF of interest is bound. We generate samples by selecting non-overlapping L-bp genomic windows from this information-rich set, where L is the context size. We set the context size $L = 1000$ as it was previously shown to improve TF binding prediction performance [16].

In vivo TF binding sites are affected by DNA sequences, the TF of interest, and the cell-type-specific chromatin landscapes. In addition to using DNase-seq, which maps chromatin accessibility, we collect ChIP-seq data for 3 types of histone modifications to form our cell-type-specific feature set. The histone modifications we include are H3K4me1, H3K4me3, and H3K27ac, which are often associated with enhancers, promoters, and active enhancers, respectively [28]. We obtain TF motif PWMs from the HOCOMOCO [29] motif database. The TF-specific feature set is generated by computing FIMO [30] motif enrichment scores using sample DNA sequences and TF motif PWMs.

2.2. Methods

NetTIME performs TF binding predictions in three steps: 1) generating the feature vector $\mathbf{w} = (w_1, \cdots, w_L)$ given a TF label p, a cell type label q, and a sample DNA sequence $\mathbf{x} = (x_1, \cdots, x_L)$ where each $x_l \in \{A, C, G, T\}$; 2) training a neural network to predict base pair resolution binding probabilities $\mathbf{z} = (z_1, \cdots, z_L)$; and 3) converting binding probabilities to binary binding decisions $\mathbf{y} = (y_1, \cdots, y_L)$ of p in q by either setting a probability threshold or additionally training a linear-chain conditional random field (CRF) classifier (Figure 1).

2.2.1. Feature representation

We construct the feature vector $\mathbf{w} \in \mathbb{R}^{K \times L}$ from $\mathbf{x} \in \mathbb{R}^{L}$, where K represents the number of features. Different types of features are independently

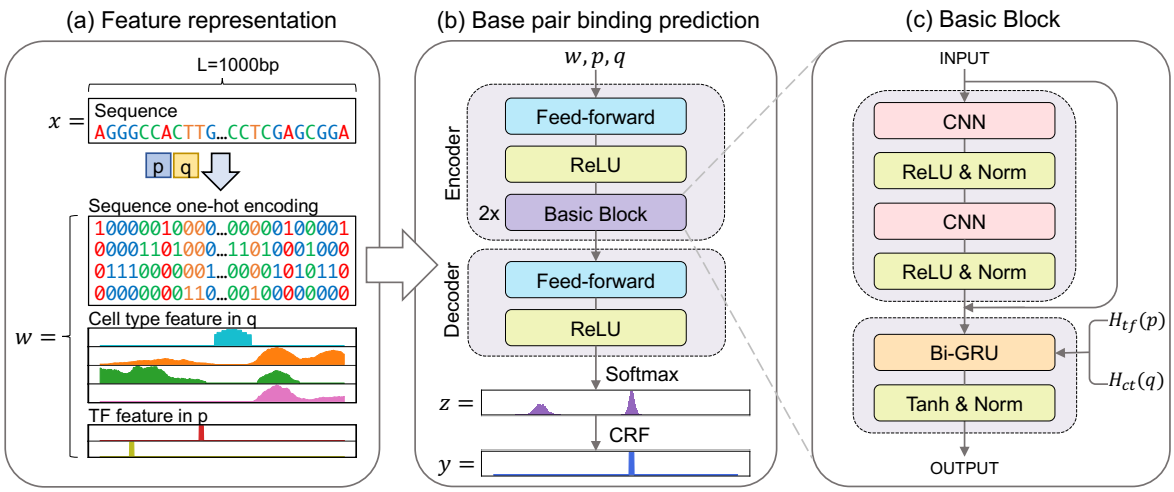

Figure 1: **Schematic method overview.** **(a)** Constructing feature vector w from input sequence x, TF label p and cell type label q. w consists of the sequence one-hot encoding, a set of cell-type-specific features – DNase-seq signals (cyan), and H3K4me1 (orange), H3K4me3 (green) and H3K4ac (pink) histone ChIP-seq signals – in cell type q, and a set of TF-specific features – TF motif enrichment in + (red) and - (olive) strands – for TF p. **(b)** Feature vector w, TF label p and cell type label q are provided to the NetTIME neural network to predict base-pair resolution binding probability z. An additional CRF classifier is trained to predict binary binding event y from z. **(c)** A detailed look at the Basic Block layer in NetTIME shown in (b).

stacked along the first dimension. For each element in $\mathbf{w}$, w_l is the concatenation of the one-hot encoding of the DNA sequence $O(x_l)$, the cell-type-specific feature $C(x_l)$, and the TF-specific feature $T(x_l)$ (Figure 1a).

$$\forall l \in [1, L],\ w_l = \begin{bmatrix} O(x_l) \\ C(x_l) \\ T(x_l) \end{bmatrix} \tag{1}$$

High-dimensional embedding vectors can be trained to distinguish different conditions as well as implicitly learning condition-specific features, and are therefore preferred by many machine learning models over one-dimensional condition labels [14, 19, 11]. Given TF label p and cell type label q, NetTIME learns the TF- and cell-type-specific embeddings $H_{tf}(p) \in \mathbb{R}^d$ and $H_{ct}(q) \in \mathbb{R}^{d'}$, where $d = d' = 50$.

2.2.2. Binding probability prediction

NetTIME (Figure 1b) adopts an encoder-decoder structure similar to that of neural machine translation models [31, 32, 33]:

Encoder:. the model encoder maps the input feature $\mathbf{w}$ to a hidden vector $\mathbf{h} \in \mathbb{R}^{2d \times L}$. The main structure of the encoder, called the Basic Block (Figure 1c), consists of a convolutional neural network (CNN) followed by a recurrent neural network (RNN). CNN uses multiple short convolution kernels to extract local binding motifs, whereas bi-directional RNN is effective at capturing long-range TF-DNA interactions. We choose the Res-Block structure introduced by ResNet [34] as our CNN, as it has become a standard approach for training deep neural networks [33, 35]. Traditional RNNs are challenging to train due to the vanishing gradient problem [36]. We therefore use the bi-directional gated recurrent unit (bi-GRU) [37], a variant of RNN proposed to address the above challenge. The hidden state of bi-GRU is initialized by concatenating the embedding vectors $H_{tf}(t)$ and $H_{ct}(c)$.

Decoder:. the model decoder converts the hidden vector $\mathbf{h}$ to binding probabilities $\mathbf{z}$. The conversion is achieved through a fully connected feed-forward network, as the relationship between $\mathbf{h}$ and $\mathbf{z}$ may not be trivial. The feed-forward network consists of two linear transformations with a ReLU activation in between. A softmax function subsequently transforms the decoder output to the binding probabilities.

2.2.3. Training

We train the model by minimizing the negative conditional log-likelihood of $\mathbf{z}$:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{L}\log p(z_l^n|x_l^n, p, q) \quad (2)$$

where N denotes the number of training samples. The loss function is optimized by the Adam optimizer [38].

2.2.4. Binding event classification

Binary binding events $\mathbf{y}$ can be directly derived from $\mathbf{z}$ by setting a probability threshold. Alternatively, a linear-chain CRF classifier can be trained to achieve the same goal. It computes the conditional probability of $\mathbf{y}$ given $\mathbf{z}$, defined as the following:

$$p(\mathbf{y}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})}exp(\sum_{l=1}^{L}(z_l)_{y_l} + \sum_{l=1}^{L}V_{y_l,y_{l+1}}) \quad (3)$$

where

1. $Z(\mathbf{z})$ is a normalization factor,
2. $V \in \mathbb{R}^{p\times p}$ is a transition matrix, where p denotes the number of classes of the classification problem and each $V_{i,j}$ represents the transition probability from class label i to j,
3. $\sum_{l=1}^{L}(z_l)_{y_l}$ calculates the likelihood of y_l given z_l, and
4. $\sum_{l=1}^{L}V_{y_l,y_{l+1}}$ measures the likelihood of y_{l+1} given y_l.

In linear-chain CRF, the class label at position l affects the classification at position $l+1$ [39]. This is potentially beneficial for TF binding site classification as positions adjacent to a putative binding site are also highly likely to be occupied by TFs. We train the CRF by minimizing $-\log p(\mathbf{y}|\mathbf{z})$ over all training samples. The Adam optimizer [38] is used to update the parameter V.

2.3. Model selection

We follow the guideline provided by the ENCODE-DREAM Challenge [4] to perform data split as well as model selection whenever possible. Training, validation, and test data are split according to chromosomes (Supplementary Table S1). We use the Area Under the Precision-Recall curve (AUPR) score to select the best neural network model checkpoint.

To assess how well our model predictions recover the positive binding sites in the truth target labels, we evaluate classifiers' performance according to Intersection Over Union (IOU) score. Suppose P and T are sets of predicted and target binding sites, respectively. Then

$$IOU = \frac{P \cap T}{P \cup T} \quad (4)$$

We test 300 random probability thresholds and select the best threshold, i.e., the threshold that achieves the highest IOU score in the validation set. We also train a CRF using predictions generated from the best neural network checkpoint. The best CRF checkpoint is selected according to the average loss on the validation set. Model performance reported here is evaluated using the test set.

3. Results

3.1. Multitask learning improves performance by increasing data availability.

NetTIME can be trained using data from a single condition (single-task learning) or multiple conditions (multitask learning). Jointly training multiple conditions potentially allows the model to use data more efficiently and improves model generalization [40]. We therefore evaluate the effectiveness of multitask learning when jointly training multiple related conditions. For this analysis, we choose three TFs from the JUN family that exhibit overlapping functions: JUN, JUNB, and JUND [41]. Combining multiple cell types of JUND allows the multitask learning model to significantly outperform the single-task learning models, each of which is trained with one JUND condition (Figure 2a). Jointly training multiple JUN family TFs further improves performance compared to training each JUN family TF separately (Figure 2b). However, we do not observe improved performance when subsampling the multitask models' training data to match the number of samples in the corresponding single-task models (Figure 2).

This indicates that the multitask learning strategy is more efficient due to the increased data available to multitask models rather than to the increased data diversity. Similar results are also observed when the same analysis is performed on three unrelated TFs (Figure S2).

Figure 2: **Performance comparison between multitask learning and single-task learning approaches using JUN family TFs.** Models are trained with datasets from **(a)** JUND across multiple cell types, and **(b)** multiple TFs in the JUN family across multiple cell types. MTL: multitask learning; MTL-sampled: multitask learning training data that has been subsampled to match the number of samples in the corresponding single-task models; STL: single-task learning. The right panels in (a) and (b) are the averaged AUPR of the models shown in the corresponding left panels. Error bars represent standard error of the mean across all training conditions.

3.2. Supervised predictions made by NetTIME achieves superior performance

Our complete feature set contains three types of features: DNA sequence, TF motif, and cell-type-specific features. In practice, however, data for these features are not always available for the conditions of interest. We therefore evaluate the quality of our model predictions when we reduce the types of input features available during training. We train separate models after removing TF- and/or cell-type-specific features using training data from all conditions mentioned in Section 2.1. Model prediction accuracy is evaluated in the supervised fashion using test data from the same set of conditions.

The addition of cell type features significantly improves NetTIME performance. However, adding TF motif enrichment features, either in addition to DNA sequence features or in addition to both sequence and cell type features, reduces prediction accuracy (Table 1). Despite exhibiting high sequence specificity *in vitro*, TF binding sites *in vivo* correlate poorly with TF motif enrichment [42]. Motif qualities in TF motif databases vary significantly depending on available binding data and motif search algorithms. Nevertheless, TF motifs have been the gold standard for TF binding site analyses due to their interpretability and scale. TF motif enrichment features are likely redundant when our model can effectively capture TF binding sequence specificity, though it's possible our protocol for generating TF motif enrichment features is suboptimal. As the TF motif features do not make positive contributions to our model performance, we exclude them for the remaining analyses in this manuscript.

	Seq	Seq + TF	Seq + CT	Seq + CT + TF
DeepBind	0.025±0.02	NA	NA	NA
BindSpace	0.035±0.02	NA	NA	NA
Catchitt	NA	NA	NA	0.260±0.14
NetTIME	0.384±0.15	0.378±0.15	**0.534±0.16**	0.525±0.15

Table 1: Comparing supervised prediction performance for DeepBind, BindSpace, Catchitt and NetTIME evaluated at 1 bp resolution. NetTIME models are trained separately under different feature settings. Seq: DNA sequence features; CT: cell-type-specific features including DNase-seq, and H3K4me1, H3K4me3 and H3K27ac histone ChIP-seq data; TF: TF-specific features containing the HOCOMOCO TF motif enrichment scores for plus and minus strands. We report here the mean±standard deviation of the AUPR scores across all training conditions.

We also compare the performance of NetTIME with that of DeepBind, BindSpace, and Catchitt. Given only DNA sequences as input, NetTIME significantly outperforms DeepBind and BindSpace. NetTIME also doubles the AUPR score achieved by Catchitt when all three types of features are used for training (Table 1).

The AUPR scores for DeepBind and BindSpace are significantly lower than those reported in the original manuscripts. One possible reason is that Table 1 reports model performance under 1 bp resolution (Supplementary section 1.2), although none of these competing methods truly achieves base-pair resolution. To avoid being biased towards NetTIME, we additionally compare method performance after reducing our prediction resolutions to match those of other methods. For each 1000 bp input sequence, we first divide NetTIME predictions into n-bp bins ($1 \leq n \leq 1000$). The binding probability of a particular bin is subsequently obtained by taking the maximum binding probabilities across all positions within that bin. We set the bin width $n = 20, 50, 100, 200, 500, 1000$. NetTIME maintains higher AUPR scores across all bin widths tested (Figure 3). We observe a consistent increase in prediction AUPR scores for DeepBind, BindSpace and NetTIME as we reduce the prediction resolutions. However, Catchitt achieves the highest AUPR score

at $n = 50$. This is the same bin width Catchitt used for the ENCODE-DREAM challenge, raising the possibility that Catchitt overfits parameters for a particular bin width of interest.

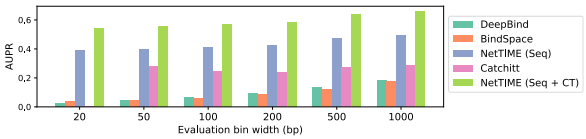

Figure 3: Supervised performance comparison of DeepBind, BindSpace, Catchitt and NetTIME evaluated at different bin widths. The Catchitt AUPR score at 20 bp bin width is set to 0 as the the program exited with error when given a 20bp-long input sequence.

3.3. TF- and cell-type-specific embeddings are crucial for an effective multitask learning strategy.

Although NetTIME outperforms several existing methods, we have yet to dissect the contributions of different components to our predictive accuracy. We use the TF and cell-type embedding vectors to learn condition-specific features and biases, and a combination of CNNs and RNNs to learn the non-condition-specific TF-DNA interaction patterns. TF and cell type embedding vectors can be replaced with random vectors at prediction time and training time to evaluate the contribution of each component individually.

To evaluate the model's sensitivity to different TF and cell type labels, TF and cell type embedding vectors are replaced with random vectors at prediction time (Table 2). Substituting both types of embeddings with random vectors reduces our model performance by 54.4% on average. Although replacing either TF or cell type embeddings with random vectors drastically reduces AUPR scores, the performance drop is more significant for cell type embeddings. This indicates that cell-type-specific chromatin landscape features are more important for defining *in vivo* TF binding sites, which explains the redundancy of TF motif features and the lack of correlation between TF ChIP-seq signals and TF motif enrichment mentioned in Section 3.2 and [42].

We additionally swap both types of embedding vectors for random vectors at training time to remove the condition-specific component, which results in a 2% drop in the mean AUPR score across all training conditions (Fig 4a). This suggests that, while embedding vectors are important for learning TF and cell-type specificity, the network components for learning common binding patterns among

	TF embeddings	
Cell type embeddings	Random	Trained
Random	0.244 ± 0.16	0.409 ± 0.16
Trained	0.310 ± 0.18	0.535 ± 0.15

Table 2: Evaluating the contribution of condition-specific network components. Trained TF and cell type embedding vectors (Trained) are replaced by random vectors (Random) at prediction time.

TFs and cell types are also crucial for maintaining high prediction accuracy.

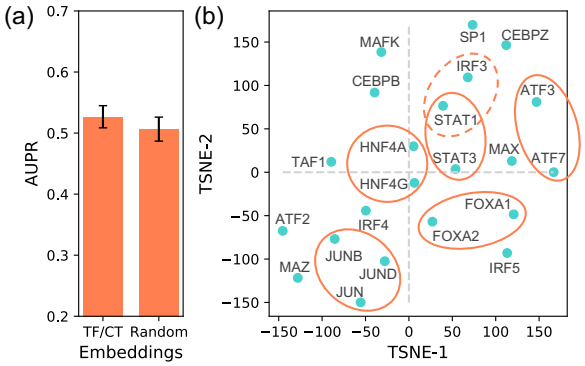

Figure 4: **Properties of trained embedding vectors.** **(a)** Evaluating the effect of replacing TF and cell type embeddings (TF/CT) with random vectors (Random) at training time. **(b)** t-SNE visualization of the TF embedding vectors. Orange circles indicate related TFs that are in close proximity in t-SNE projection space: solid circles illustrate TFs from the same protein family, and dashed circles illustrate TFs having similar functions.

Visualizing the trained TF embedding vectors in two dimensions using t-SNE [43] reveals that a subset of embedding vectors also reflects the TF functional similarities. Some TFs that are in close proximity in t-SNE space are from the same TF families, including FOXA1 and FOXA2, HNF4A and HNF4G, STAT1 and STAT3, ATF3 and ATF7, and JUN, JUNB, and JUND (Figure 4b, solid circles). Functionally related TFs including IRF3 and STAT1 [44] are also adjacent to each other in t-SNE space (Figure 4b, dashed circle). However, these TF embedding vectors are explicitly trained to learn the biases introduced by TF labels. Available data for TFs of the same protein family are not necessarily from the same set of cell types. As a result, not all functionally related TFs are close in t-SNE space, such as IRF (IRF3, IRF4, and IRF5) family proteins and TFs associated with c-Myc proteins (MAX and MAZ).

3.4. TF and cell type embeddings allow more reliable transfer predictions.

Transfer learning allows models to make cross-TF and cross-cell-type predictions beyond training conditions. Most existing methods achieve transfer learning by providing input features from a new cell type to a model trained on a different cell type. If multiple trained cell types are available for the same TF, the final cross-cell-type predictions are generated by averaging predictions from all trained cell types (Average Trained). Suppose we wish to make transfer predictions for TF p in cell type q, denoted $[p, q]$. This approach allows us to take advantage of all available data for TF p ($[p, *]$). However, the TF and cell type embedding approach (Embedding Transfer) allows our model to leverage all available data for both TF p and cell type q ($[p, *] \cup [*, q]$). Since the multitask learning paradigm benefits from having more training data (Section 3.1) and *in vivo* TF binding sites mostly correlate with cell-type-specific features (Section 3.3), the latter approach can potentially improve our model's transfer predictive performance.

To evaluate the prediction quality of these two approaches, we pretrain a NetTIME model by leaving out 10 conditions for transfer learning. For each transfer condition $[p, q]$, we use the pretrained model to derive both the Average Trained and the Embedding Transfer predictions. Transfer learning predictions are generally less accurate compared to supervised predictions (Supervised). However, transfer predictions generated by Embedding Transfer still significantly outperform those of the Average Trained (Figure 5a). Transfer predictions derived from NetTIME also achieve considerably higher accuracy compared to those from Catchitt (Figure S3). Average Trained and Embedding Transfer predictions can also be obtained after fine-tuning the pretrained model with data from $[p, *] \setminus [p, q]$ and $[p, *] \cup [*, q] \setminus [p, q]$, respectively. This fine-tuning step additionally introduces marginal improvement measured by average AUPR score across all leave-out conditions (Figure S4).

Using trained TF and cell type embeddings allows models to perform binding predictions beyond the training panels of TFs and cell types. We therefore test our model's robustness when making predictions on unknown conditions using 6 conditions from 6 new TFs in 3 new cell types. Starting from a NetTIME model pretrained on all original training conditions (Section 2.1), we fine-tune the pretrained model for each transfer condition $[p', q']$ by collecting available samples from $[p', *] \cup [*, q'] \setminus [p', q']$. Transfer predictions generated from models trained with TF and cell type embeddings significantly outperform those from models trained with random

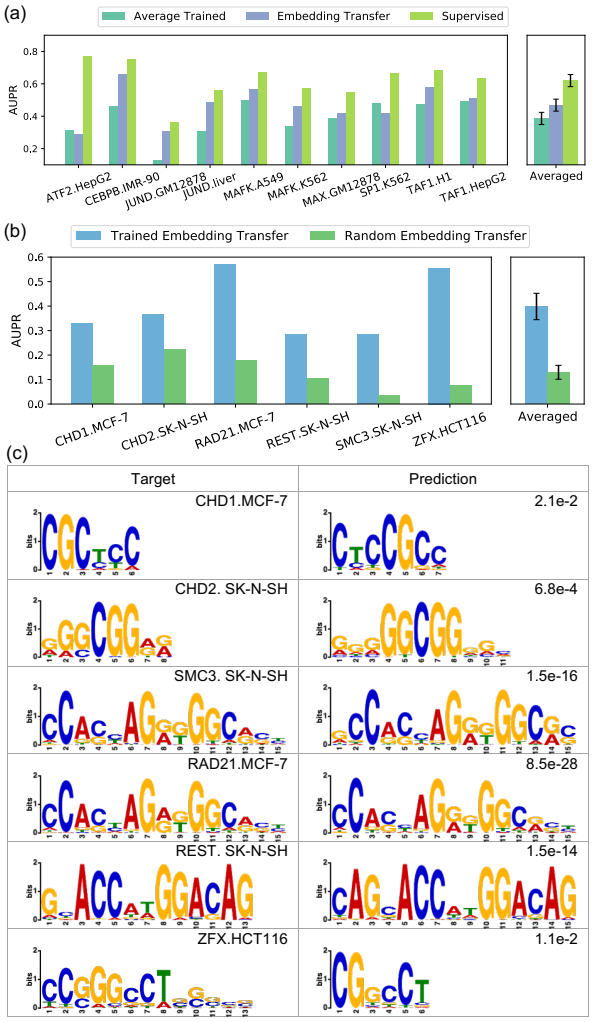

Figure 5: **Transfer predictions using NetTIME. (a)** Comparing the prediction efficiency of two transfer learning strategies using 10 leave-out conditions within the training set of conditions. **(b)** and **(c)** illustrate NetTIME transfer prediction performance using conditions beyond the training TF and cell type panels. **(b)** Transfer predictions using models trained with either TF and cell type embedding vectors (Trained Embedding Transfer) or random vectors (Random Embedding Transfer). **(c)** Comparison of predicted TF binding motifs to those derived from target ChIP-seq conserved peaks. Predicted motifs are derived from Trained Embedding Transfer predictions. *De novo* motif discovery is conducted using STREME [45] software. Motif similarity p-values shown in the top right corner of the Prediction column are derived by comparing predicted and target motifs using TOMTOM [46].

embeddings that cannot distinguish different TF and cell-type identities (Figure 5b). TF binding motifs derived from predicted binding sites also show a strong resemblance to those derived from conserved ChIP-seq peaks (Figure 5c).

3.5. A CRF classifier post-processing step effectively reduces prediction noise.

Summarizing the binding strength, or probability, along the chromosome at each discrete binding site is an important step for several downstream tasks ranging from visualization to validation. To make binary binding decisions from binding probability scores, we first test the predictive performance of 300 probability thresholds and find that at threshold 0.143, our model achieves the highest IOU score of 35.6% on the validation data (Figure S5). We alternatively train a CRF classifier, as a manually selected probability threshold is poorly generalizable to unknown datasets. These two approaches achieve similar predictive performance as evaluated by IOU scores (Figure 6a). However, prediction noises manifested as high probability spikes are likely to be classified as bound using the probability threshold approach. To evaluate the effectiveness of reducing prediction noises using the probability threshold and the CRF approaches, we calculate the percentage of class label transitions per sequence within the target labels and within each of the predicted labels generated by these two approaches. The transition percentage using CRF is comparable to that of the true target labels, and is also significantly lower than the percentage obtained using the probability threshold approach. This indicates that CRF is more effective at reducing prediction noise, and therefore CRF predictions exhibit a higher degree of resemblance to target labels.

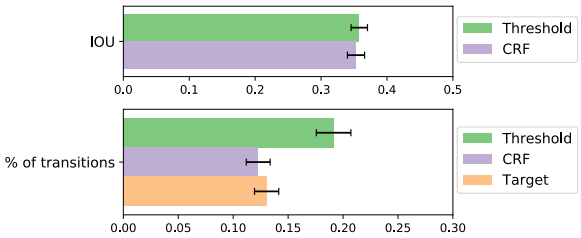

Figure 6: **Binary classification performance using the probability threshold and CRF.** Performance evaluated by mean IOU score (top) and the percentage of class label transitions per sequence (bottom), both calculated over all training conditions.

4. Conclusions

In this work, we address several challenges facing many existing methods for TF binding site predictions by introducing a multitask learning framework, called NetTIME, that learns base-pair resolution TF binding sites using embeddings. We first show that the multitask learning approach improves our prediction accuracy by increasing the amount of data available to the model. Both the condition-specific and non-condition-specific components in our multitask framework are important for making accurate condition-specific binding predictions. The use of TF and cell type embedding vectors additionally allows us to make accurate transfer learning predictions within and beyond the training panels of TFs and cell types. Our method also significantly outperforms previous methods under both supervised and transfer learning settings, including DeepBind, BindSpace, and Catchitt.

Although DNA sequencing currently can achieve base-pair resolution, the resolution of ChIP-seq data is still limited by the size of DNA fragments obtained through random clipping. A considerable fraction of the fragments is therefore false positives, whereas many transient and low-affinity binding sites are missed [47]. Additionally, ChIP-seq requires suitable antibodies for proteins of interest, which can be difficult to obtain for rare cell types and TFs. Alternative assays have been proposed to improve data resolution [48, 49, 50] as well as to eliminate the requirement for antibodies [51, 52]. However, datasets generated from these techniques are rare or missing in data consortiums such as ENCODE [21] and ReMap [22]. Base-pair resolution methods for predicting binding sites from these assays [10, 53] have largely been limited to characterizing TF sequence specificity. NetTIME can potentially provide base-pair resolution solutions to more complex DNA sequence problems as labels generated from these alternative assays become more widely available in the future.

ATAC-seq (Assay for Transposase-Accessible Chromatin using sequencing [54]) has overtaken DNase-seq as the preferred assay to profile chromatin accessibility, as it requires fewer steps and input materials. However, these two techniques each offer unique insights into the cell-type-specific chromatin states [55], and it is therefore potentially beneficial to incorporate both data types for TF binding predictions. In fact, extensive feature engineering has been the focus of many recent *in*

vivo TF binding prediction methods [42, 25, 20]. It is also important to note that, without strategies for handling missing features, increasing feature requirements significantly restricts models' scope of application (Figure S1). A comprehensive evaluation of data imputation methods [56, 57, 58, 59] can be difficult due to the lack of knowledge of the true underlying data distribution. We plan to extend our model's ability to learn from a more diverse set of features, and investigate more efficient ways to handle missing data. We also plan to explore other neural network architectures to improve model performance while reducing the model's feature requirement.

NetTIME is extensible, and can be adapted to improve solutions to other biology problems, such as transcriptional regulatory network (TRN) inference. TRN inference identifies genome-wide functional regulations of gene expressions by TFs. TFs control the expression patterns of target genes by first binding to regions containing promoters, distal enhancers and/or other regulatory elements. However, functional interactions between TFs and target genes are further complicated by TF concentrations and co-occurrence of other TFs. A series of methods have been proposed for inferring TRNs from gene expression data and prior knowledge of the network structure [60, 61, 62]. Prior knowledge can be obtained by identifying open chromatin regions close to gene bodies that are also enriched with TF motifs [63]. However, this method is problematic for identifying TF functional regulations towards distal enhancers and binding sites without motif enrichment. *In vivo* predictions of TF binding profiles, however, can serve as a more flexible approach to generating prior network structure as it bypasses the aforementioned unnecessary constraints. In future work, we hope to adapt the NetTIME framework to explore more efficient approaches for generating prior knowledge for more biophysically motivated TRN inference.

Acknowledgements

We thank members in the Bonneau lab and the Cho lab for providing helpful suggestions on the project and on the manuscript. We thank the NYU High Performance Computing team and the Flatiron Institute Scientific Computing team for their excellent high performance computing support.

Funding

RB and RY thank the following sources for research support: NSF IOS-1546218, NIH R35GM122515, NIH R01HD096770, NIH R01NS116350, Simons Foundation. KC is partly supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from pattern recognition to AI) and Samsung Research (Improving Deep Learning using Latent Structure). KC also thanks Naver, eBay, NVIDIA, and NSF Award 1922658 for support.

References

[1] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

[2] David S Johnson, Ali Mortazavi, Richard M Myers, and Barbara Wold. Genome-wide mapping of in vivo protein-dna interactions. *Science*, 316(5830):1497–1502, 2007.

[3] Matthew T Weirauch, Atina Cote, Raquel Norel, Matti Annala, Yue Zhao, Todd R Riley, Julio Saez-Rodriguez, Thomas Cokelaer, Anastasia Vedenko, Shaheynoor Talukder, et al. Evaluation of methods for modeling transcription factor sequence specificity. *Nature biotechnology*, 31(2):126–134, 2013.

[4] DREAM. Encode-dream in vivo transcription factor binding site prediction challenge, 2017.

[5] A. J. Stewart, S. Hannenhalli, and J. B. Plotkin. Why transcription factor binding sites are ten nucleotides long. *Genetics*, 192(3):973–985, Nov 2012.

[6] Timothy L. Bailey. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36. AAAI Press, 1994.

[7] T. L. Bailey, N. Williams, C. Misleh, and W. W. Li. MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res.*, 34(Web Server issue): W369–373, Jul 2006.

[8] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, 33 (8):831–838, Aug 2015.

[9] Hamid Reza Hassanzadeh and May D Wang. Deeperbind: Enhancing prediction of sequence specificities of dna binding proteins. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 178–183. IEEE, 2016.

[10] Sirajul Salekin, Jianqiu Michelle Zhang, and Yufei Huang. Base-pair resolution detection of transcription factor binding site by deep deconvolutional network. *Bioinformatics*, 34(20):3446–3453, 2018.

[11] Han Yuan, Meghana Kshirsagar, Lee Zamparo, Yuheng Lu, and Christina S Leslie. Bindspace decodes transcription factor binding signals by large-scale sequence embedding. *Nature methods*, 16(9):858–861, 2019.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[13] Ehsaneddin Asgari and Mohammad RK Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, 10(11):e0141287, 2015.

[14] Ren Yi, Pi-Chuan Chang, Gunjan Baid, and Andrew Carroll. Learning from data-rich problems: A case study on genetic variant calling. *arXiv preprint arXiv:1911.05151*, 2019.

[15] David Gfeller, Frank Butty, Marta Wierzbicka, Erik Verschueren, Peter Vanhee, Haiming Huang, Andreas Ernst, Nisa Dar, Igor Stagljar, Luis Serrano, et al. The multiple-specificity landscape of modular peptide recognition domains. *Molecular systems biology*, 7(1):484, 2011.

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

[17] Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50 (8):1171–1179, 2018.

[18] David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research*, 28(5): 739–750, 2018.

[19] Qian Qin and Jianxing Feng. Imputation for transcription factor binding predictions based on deep learning. *PLoS computational biology*, 13(2):e1005403, 2017.

[20] Jens Keilwagen, Stefan Posch, and Jan Grau. Accurate prediction of cell type-specific transcription factor binding. *Genome biology*, 20(1):9, 2019.

[21] Jill E Moore, Michael J Purcaro, Henry E Pratt, Charles B Epstein, Noam Shoresh, Jessika Adrian, Trupti Kawli, Carrie A Davis, Alexander Dobin, Rajinder Kaul, et al. Expanded encyclopaedias of dna elements in the human and mouse genomes. *Nature*, 583(7818):699–710, 2020.

[22] Jeanne Chèneby, Zacharie Ménétrier, Martin Mestdagh, Thomas Rosnet, Allyssa Douida, Wassim Rhalloussi, Aurélie Bergon, Fabrice Lopez, and Benoit Ballester. Remap 2020: a database of regulatory regions from an integrative analysis of human and arabidopsis dna-binding sequencing experiments. *Nucleic acids research*, 48(D1): D180–D188, 2020.

[23] Diego Calderon, Michelle LT Nguyen, Anja Mezger, Arwa Kathiria, Fabian Müller, Vinh Nguyen, Ninnia Lescano, Beijing Wu, John Trombetta, Jessica V Ribado, et al. Landscape of stimulation-responsive chromatin across diverse human immune cells. *Nature genetics*, pages 1–12, 2019.

[24] Paja Sijacic, Marko Bajic, Elizabeth C McKinney, Richard B Meagher, and Roger B Deal. Changes in chromatin accessibility between arabidopsis stem cells and mesophyll cells illuminate cell type-specific transcription factor networks. *The Plant Journal*, 94(2):215–231, 2018.

[25] Daniel Quang and Xiaohui Xie. Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods*, 166:40–47, 2019.

[26] Trevor Siggers and Raluca Gordân. Protein–dna binding: complexities and multi-protein codes. *Nucleic acids research*, 42(4):2099–2111, 2014.

[27] ENCODE. Encode experiment guidelines, 2020. URL https://www.encodeproject.org/about/experiment-guidelines

[28] Vicky W Zhou, Alon Goren, and Bradley E Bernstein. Charting histone modifications and the functional organization of mammalian genomes. *Nature Reviews Genetics*, 12(1):7–18, 2011.

[29] Ivan V Kulakovskiy, Ilya E Vorontsov, Ivan S Yevshin, Ruslan N Sharipov, Alla D Fedorova, Eugene I Rumynskiy, Yulia A Medvedeva, Arturo Magana-Mora, Vladimir B Bajic, Dmitry A Papatsenko, et al. Hocomoco: towards a complete collection of transcription factor binding models for human and mouse via large-scale chip-seq analysis. *Nucleic acids research*, 46(D1):D252–D259, 2018.

[30] Charles E Grant, Timothy L Bailey, and William Stafford Noble. Fimo: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[32] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[35] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[36] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[37] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

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

[39] Charles Sutton and Andrew McCallum. An introduction to conditional random fields, 2010.

[40] Rich Caruana. Multitask learning. *Machine learning*, 28 (1):41–75, 1997.

[41] Fatima Mechta-Grigoriou, Damien Gerald, and Moshe Yaniv. The mammalian jun proteins: redundancy and specificity. *Oncogene*, 20(19):2378–2389, 2001.

[42] Xi Chen, Bowen Yu, Nicholas Carriero, Claudio Silva, and Richard Bonneau. Mocap: large-scale inference of transcription factor binding sites from chromatin accessibility. *Nucleic acids research*, 45(8):4315–4329, 2017.

[43] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008.

[44] Trine H Mogensen. Irf and stat transcription factors-from basic biology to roles in infection, protective immunity, and primary immunodeficiencies. *Frontiers in immunology*, 9: 3047, 2019.

[45] Timothy L Bailey. Streme: Accurate and versatile sequence motif discovery. *bioRxiv*, 2020.

[46] Shobhit Gupta, John A Stamatoyannopoulos, Timothy L Bailey, and William Stafford Noble. Quantifying similarity between motifs. *Genome biology*, 8(2):R24, 2007.

[47] Peter J Park. Chip–seq: advantages and challenges of a maturing technology. *Nature reviews genetics*, 10(10):669–680, 2009.

[48] Ho Sung Rhee and B Franklin Pugh. Comprehensive genome-wide protein-dna interactions detected at single-nucleotide resolution. *Cell*, 147(6):1408–1419, 2011.

[49] Qiye He, Jeff Johnston, and Julia Zeitlinger. Chip-nexus enables improved detection of in vivo transcription factor binding footprints. *Nature biotechnology*, 33(4):395–401, 2015.

[50] Matthew J Rossi, William KM Lai, and B Franklin Pugh. Simplified chip-exo assays. *Nature communications*, 9(1): 1–13, 2018.

[51] Bas van Steensel and Steven Henikoff. Identification of in vivo dna targets of chromatin proteins using tethered dam methyltransferase. *Nature biotechnology*, 18(4):424–428, 2000.

[52] Tony D Southall, Katrina S Gold, Boris Egger, Catherine M Davidson, Elizabeth E Caygill, Owen J Marshall, and Andrea H Brand. Cell-type-specific profiling of gene expression and chromatin binding without cell isolation: assaying rna pol ii occupancy in neural stem cells. *Developmental cell*, 26(1):101–112, 2013.

[53] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, pages 1–13, 2021.

[54] Jason D Buenrostro, Paul G Giresi, Lisa C Zaba, Howard Y Chang, and William J Greenleaf. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature methods*, 10(12):1213, 2013.

[55] Aslıhan Karabacak Calviello, Antje Hirsekorn, Ricardo Wurmus, Dilmurat Yusuf, and Uwe Ohler. Reproducible inference of transcription factor footprints in atac-seq and dnase-seq datasets using protocol-specific bias modeling. *Genome biology*, 20(1):1–13, 2019.

[56] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[57] Bryan N Howie, Peter Donnelly, and Jonathan Marchini. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genet*, 5(6):e1000529, 2009.

[58] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

[59] Matthew Amodio, David Van Dijk, Krishnan Srinivasan, William S Chen, Hussein Mohsen, Kevin R Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, et al. Exploring single-cell data with deep multitasking neural networks. *Nature methods*, 16(11): 1139–1145, 2019.

[60] Alexandre Irrthum, Louis Wehenkel, Pierre Geurts, et al. Inferring regulatory networks from expression data using tree-based methods. *PloS one*, 5(9):e12776, 2010.

[61] Alex Greenfield, Christoph Hafemeister, and Richard Bonneau. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, 29(8):1060–1067, 2013.

[62] Ye Yuan and Ziv Bar-Joseph. Deep learning for inferring gene relationships from single-cell expression data. *Proceedings of the National Academy of Sciences*, 116(52): 27151–27158, 2019.

[63] Emily R Miraldi, Maria Pokrovskii, Aaron Watters, Dayanne M Castro, Nicholas De Veaux, Jason A Hall, June-Yong Lee, Maria Ciofani, Aviv Madar, Nick Carriero, et al. Leveraging chromatin accessibility for transcriptional regulatory network inference in t helper 17 cells. *Genome research*, 29(3):449–463, 2019.