Maximum Entropy Methods for Extracting the Learned Features of Deep Neural Networks

Alex Finnegan[1,2] and Jun S. Song[1,2,3,*]

Abstract

New architectures of multilayer artificial neural networks and new methods for training them are rapidly revolutionizing the application of machine learning in diverse fields, including business, social science, physical sciences, and biology. Interpreting deep neural networks, however, currently remains elusive, and a critical challenge lies in understanding which meaningful features a network is actually learning. We present a general method for interpreting deep neural networks and extracting network-learned features from input data. We describe our algorithm in the context of biological sequence analysis. Our approach, based on ideas from statistical physics, samples from the maximum entropy distribution over possible sequences, anchored at an input sequence and subject to constraints implied by the empirical function learned by a network. Using our framework, we demonstrate that local transcription factor binding motifs can be identified from a network trained on ChIP-seq data and that nucleosome positioning signals are indeed learned by a network trained on chemical cleavage nucleosome maps. Imposing a further constraint on the maximum entropy distribution, similar to the grand canonical ensemble in statistical physics, also allows us to probe whether a network is learning global sequence features, such as the high GC content in nucleosome-rich regions. This work thus provides valuable mathematical tools for interpreting and extracting learned features from feed-forward neural networks.

1 INTRODUCTION

Multilayer artificial neural networks are becoming increasingly important tools for predicting outcomes from complex patterns in images and diverse scientific data including biological sequences. For example, recent works have applied multilayer networks, also called deep learning models, to predicting the transcription factor (TF) binding (Alipanahi *et al.*, 2015; Zeng *et al.*, 2016) and chromatin states (Zhou and Troyanskaya, 2015; Kelley *et al.*, 2016) from DNA sequence, greatly advancing the state-of-the-art prediction rate in these fields. The success of these multilayer networks stems from their ability to learn complex, non-linear prediction functions over the set of input sequences. The main challenge in using deep learning currently resides in the fact that the complexity of the learned function coupled with the typically large dimension of the input and parameter spaces makes it very difficult to decipher which input features a network is actually learning.

In spite of this challenge, the rise of deep learning has spurred efforts to understand network prediction, not only in genomics but also in the fields of computer vision and natural language processing. In earlier works, genomics researchers inferred learned sequence motifs by examining the weight matrices convolved with sequence inputs at the first layer of convolutional networks, as well as by performing *in silico* mutagenesis studies. The first approach can yield results resembling known TF motifs (Alipanahi *et al.*, 2015), but is, in general, unsatisfactory since it ignores the non-linear function computed by higher network layers which take the results of first layer convolutions to ultimately determine network predictions. The second approach respects the complexity of the network's prediction function, but can be difficult to apply since the locations, types and combinations of *in silico* mutations necessary to produce significant changes in network predictions are not obvious *a priori*.

More recently, Shrikumar *et al.* (2016) introduced DeepLIFT, a variant of the layer-wise relevance propagation method of network interpretation proposed by Bach *et al.* (2015). Both approaches show great promise for interpreting network prediction in a general context. However, they are based on a set of axioms external to the actual operations performed by the network, and it has been argued that a different set of equally plausible axioms can lead to changes in DeepLIFT's interpretation rules (Lundberg and Lee, 2016)

In this paper, we use the rigorous formalism of statistical physics to develop a novel method for extracting and interpreting network-learned sequence features. The method makes direct reference to the nonlinear function learned by the network by sampling a maximum entropy distribution over all possible sequences, anchored at an input sequence and subject to constraints implied by the learned function and by the background nucleotide content of the genome from which the network's input sequences are derived.

To extract learned features from inputs, we study two complementary quantities derived from sequences sampled from the constrained maximum entropy distribution via Markov Chain Monte Carlo (MCMC): (1) a local profile of nucleotide contents for revealing sequence motifs, and (2) a feature importance score based on the sample variance of a summary statistic for focusing on a particular sequence characteristic of interest. The latter score directly measures the effect of a global sequence feature, such as GC content, on the non-linear function learned by the network, and it can be used to rank global features, thereby answering questions about the relative importance of such features in the context of network prediction.

We demonstrate the utility of our method by applying it to interpret deep neural networks trained on the motif discovery task of Zeng *et al.* (2016) and on the task of predicting nucleosome positions. In the motif discovery task, our method successfully localized learned CTCF motifs, and sequence logos generated from sample nucleotide frequencies demonstrated that the network learned both the canonical CTCF motif and its reverse complement. In the nucleosome positioning task, sample mean nu-

[1]Department of Physics, [2]Carl R. Woese Institute for Genomic Biology, [3]Department of Bioengineering, University of Illinois, Urbana-Champaign

*Correspondence: songj@illinois.edu

cleotide contents showed a learned preference for G/C and A/T nucleotides at positions separated by 10 base pairs (bps); however, we found that the network did not require full adherence to the canonical 10 bp periodic pattern in individual sequences to classify them correctly as nucleosomal. Furthermore, application of our global sequence feature score allowed estimation of the fraction of nucleosomes for which high GC content is an important network feature. Finally, we showed that using only a few low-variance principal components of our maximum entropy distributions provides an effective dimensional reduction scheme for encoding inputs and facilitates the extraction of important features *de novo* from a trained network.

Although this paper restricts attention to genomics applications, the described interpretation method is general and can be applied to any feed-forward neural network.

2 METHODS

2.1 Monte Carlo sampling

Given an element $\boldsymbol{x_o}$ in the network's test data set, we sampled sequences **x** from the maximum entropy distribution (3) using a Markov chain Monte Carlo (MCMC) method. We initialized the Markov random sequence **x** at $\boldsymbol{x_o}$, then repeatedly selected an index i of **x** with uniform probability and proposed a mutation of nucleotide x_i to nucleotide x_i^* sampled from the set $\{G, C, A, T\} - \{x_i\}$ with uniform probability. The proposed mutation was accepted with probability given by the metropolis criteria:

$$\mathbb{P}_{accept} = \min\left(1,\ e^{-\beta\left(d\left(\boldsymbol{\Phi}(\boldsymbol{x_o}),\boldsymbol{\Phi}(\boldsymbol{x}^*)\right) - d\left(\boldsymbol{\Phi}(\boldsymbol{x_o}),\boldsymbol{\Phi}(\boldsymbol{x})\right)\right)}\right)$$

where $\boldsymbol{x}^*$ denotes the random variable $\boldsymbol{x}$ with the proposed mutation at index i (Metropolis *et al*., 1953).

To generate the results of section 3.4 we sampled the distribution (3) associated with each $\boldsymbol{x_o}$ with 100 Markov chains running in parallel with each chain constructed from 3×10^4 proposed mutations. Chains were sampled every 100 proposals so that we collected a total of 3×10^4 samples from the distribution associated with each $\boldsymbol{x_o}$.

We sampled the distribution (9) using the same MCMC scheme except the probability of accepting a proposed mutation becomes

$$\mathbb{P}_{accept} = \min\left(1,\ e^{-\beta\left[d\left(\boldsymbol{\Phi}(\boldsymbol{x_o}),\boldsymbol{\Phi}(\boldsymbol{x}^*)\right) - d\left(\boldsymbol{\Phi}(\boldsymbol{x_o}),\boldsymbol{\Phi}(\boldsymbol{x})\right)\right] + \mu\left[N(\boldsymbol{x}^*) - N(\boldsymbol{x})\right]}\right).$$

To generate the results of section 3.5 we sampled the distribution (9) associated with each $\boldsymbol{x_o}$ with a single Markov chain constructed by proposing 1×10^6 mutations. We sampled the chain every 100 proposals.

2.2 Application 1: Localizing learned motifs

2.2.1 Data accession and network training

We downloaded the CTCF motif discovery data set from http://cnn.csail.mit.edu/motif_discovery/wgEncodeAwgTfbsHaibSknshraCtcfV0416102UniPk/ (Zeng *et al*., 2016) This data set was derived from a CTCF ChIP-seq experiment performed in human neuroblastoma cells treated with retinoic acid. Downloaded data were pre-partitioned into training and test sets. We set aside 1/8th of training data for validation and trained the network architecture described in section 3.4 for 20 epochs with a batch size of 80, employing the categorical cross-entropy loss function and the adaDelta optimizer from python package Keras (Chollet, 2015). We evaluated validation performance at the end of each epoch and selected the model with best validation accuracy.

2.2.2 MEME motif discovery

We used the program MEME, version 4.10, to discover a consensus motif in 1500 inputs chosen randomly from the set of inputs where we applied our interpretation method (Bailey *et al*., 2015). We instructed MEME to find zero or one motifs in each input with minimum and maximum motif lengths of 6 and 19, respectively. We required the consensus motif be present in at least 400 inputs and stopped the search when the E-value exceeded 0.01.

2.2.3 MUSCLE alignment of network-derived motifs

Nineteen bp network-derived motifs were aligned with the multiple sequence aligner MUSCLE version 3.8 using a gap penalty of -1000 to force the aligner to treat each extracted motif as a contiguous sequence (Edgar, 2004). The resulting alignment spanned 35 bps, but almost all aligned sequences lay in the center of this interval, so we selected the central 19 bps to produce the sequence logo in Fig. 3(B).

2.2.4 FIMO motif scan

We used the program FIMO, version 4.10, to scan the full set of 2500 sequences where we applied our interpretation (Grant *et al*., 2011). We instructed FIMO not to search for the reverse complement of the MEME motif, and FIMO identified motifs in 1652 inputs. To construct the plot of relative distance distribution in Fig. 3(C), whenever FIMO called more than one motif in a sequence, we measured the distance between the network-derived motif and the FIMO motif with the lowest p-value.

2.3 Application 2: Extracting nucleosome positioning signals

2.3.1 Data set construction

We downloaded chemical cleavage maps of redundant and unique nucleosome dyads from the supplementary material of Brogaard *et al.* (2012), and we used these dyad indices to construct data sets from the UCSC SAC2 version of the *S. cerevisiae* genome. Our positive validation and test data sets consisted of genomic intervals centered on unique nucleosome dyads of chromosomes 7 and 12, respectively. The positive training set consisted of genomic intervals centered on the set of redundant dyads from all other chromosomes (note that the set of redundant dyads contains the set of unique dyads). Each data set was balanced by negative elements centered on genomic indices sampled uniformly and without replacement from genomic indices at least 3 bps from all redundant dyads. This chromosome-based division of training, validation, and test data corresponded to roughly an 80%, 10%, 10% split of available data, and our test set contained 10500 elements.

We represent all sequence inputs with a 1 hot encoding scheme and then mean center and normalize each input by considering it to be a Bernoulli random variable with success probability given by the genome-wide nucleotide frequency. We found that this preprocessing gives a marginal improvement over simple 1-hot encoding for this classification task.

2.3.2 Network training

We trained the network architecture described in section 3.5 using stochastic gradient descent with batch size of 8 and the categorical cross entropy loss function. Our gradient descent used a learning rate of 0.1, momentum parameter of 0.5, and L2 weight penalty of 0.001. Training was done with the python package Keras (Chollet, 2015).

2.3.3 Calculation of normalized Fourier amplitudes

Normalized Fourier amplitudes were calculated by performing discrete Fourier transform with the python package numpy (van der Walt *et al.,* 2011), setting the zero frequency component to 0, then normalizing by the Euclidean norm of the Fourier components and then calculating the amplitude at each frequency. These normalized amplitudes are averaged to product the plots in Fig. 4(B).

2.3.4 Generating random orthogonal vectors for nearest neighbor classifier

We generated sets of 5 orthogonal basis vectors distributed uniformly over the unit sphere embedded in 201 dimensions by sampling the 201 components of each vector from standard normal distributions and then performing QR decomposition on the 201×5 matrix of column vectors.

3 RESULTS

3.1 Discovering learned sequence features with a constrained maximum entropy distribution

Consider a trained, multilayer feed-forward neural network that takes a length L genomic sequence as input and performs a classification task. For concreteness, let the vector $\boldsymbol{x_o}$ represent a specific input sequence in the 0th class, where the entries at each index in $\boldsymbol{x_o}$ are one of the nucleotides A,C,G,T. (In practice, sequence inputs are commonly represented with a one-hot encoding scheme; however, for the purpose of discussion we will use the conceptually simpler sequence representation, observing that the adaptation of our algorithm to the case of one-hot encoding is straightforward.) The input $\boldsymbol{x_o}$ yields at each intermediate layer a vector with real valued entries corresponding to the activations of that layer's units. Activations of units in the output layer determine the network's classification, typically via a one-to-one correspondence between classes and the output layer's units, with $\boldsymbol{x_o}$ assigned to the class whose output unit activation is largest.

The standard motivation behind multilayer networks is that intermediate layers may learn to recognize a hierarchy of features present in the set of inputs with features becoming more abstract with the depth of the intermediate layer. Since changes in the features detected by a layer are encoded in changes in the intermediate layer's vector of activations, we propose that it is possible to identify learned features by looking for commonalities among the set of all input sequences that approximately preserve that layer's vector of activations. In this work, we focus attention on features learned by the penultimate layer, since, because this layer makes direct connection to the output layer, its learned features are the most relevant to the network's classification. Therefore, let $\boldsymbol{\Phi}(\boldsymbol{x_o})$ denote the vector of penultimate layer activations associated with input $\boldsymbol{x_o}$ from the network's training or test sets. We formalize the search for generic sequences that approximately reproduce $\boldsymbol{\Phi}(\boldsymbol{x_o})$, i.e. roughly trigger the set of given penultimate neurons, by weighting the set X of all length L sequences with a maximum entropy distribution p subject to the constraint

$$\langle d\big(\boldsymbol{\Phi}(\boldsymbol{x_o}), \boldsymbol{\Phi}(\boldsymbol{x})\big)\rangle_{x\in X} = D, \qquad (1)$$

where $\langle\cdot\rangle_{x\in X}$ denotes expectation with respect to the distribution p, $d(\cdot,\cdot)$ denotes a distance metric on the space of penultimate activations, and D is a positive constant, with smaller values corresponding to a more exact reproduction of the activation vector $\boldsymbol{\Phi}(\boldsymbol{x_o})$.

We use the following weighted Euclidean metric for d:

$$d\big(\boldsymbol{\Phi}(\boldsymbol{x_0}), \boldsymbol{\Phi}(\boldsymbol{x})\big) = \big(\textstyle\sum_{i=1}^{k}\big(\Phi_i(\boldsymbol{x_o}) - \Phi_i(\boldsymbol{x})\big)\, w_{0,i}^2\big(\Phi_i(\boldsymbol{x_o}) - \Phi_i(\boldsymbol{x})\big)\ \big)^{\frac{1}{2}},$$

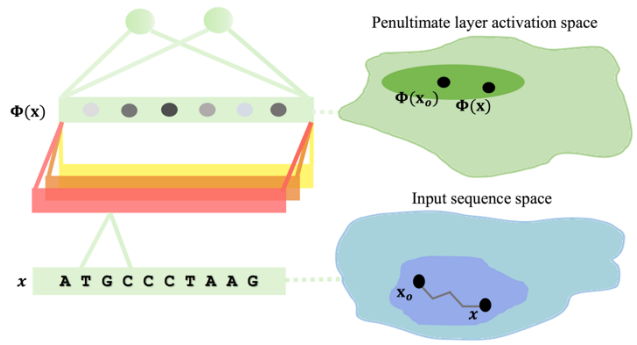

Fig. 1. Schematic representation of our maximum entropy-based interpretation method. Given a network input $\boldsymbol{x_o}$, we sample $\boldsymbol{x}$ from a maximum entropy distribution that is constrained by the requirement that the samples approximately reproduce the vector of penultimate activations $\boldsymbol{\Phi}(\boldsymbol{x_o})$ which encodes learned features.

where $w_{0,i}$ denotes the weight of connection of the ith unit in a penultimate layer with k units to the output unit indexed by 0. In general, the choice of output unit is task specific; but, for the two-class classification problems we consider, the learned weights connecting each penultimate layer unit to the two units of the softmax output layer are approximately equal in magnitude, so that the weights of either output unit can be used.

The explicit form of p is obtained by maximizing the Shannon entropy

$$-\textstyle\sum_{x\in X} p(\boldsymbol{x})\, log\big(p(\boldsymbol{x})\big) \qquad (2)$$

subject to the constrain (1). Applying the method of Lagrange multipliers, we find that

$$p(\boldsymbol{x}) = \frac{1}{Z}\, e^{-\beta d(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x}))} \qquad (3)$$

where Z is a normalization constant, and $\beta > 0$ is the Lagrange multiplier whose value is chosen to yield the desired value of D in (1).

The parameter β can be fixed to force the distribution of $\boldsymbol{\Phi}(\boldsymbol{x})$ stemming from $p(\boldsymbol{x})$ to have small width on the distance scale set by the empirical distribution of penultimate activations arising from the set of input sequences. Alternatively, because sufficiently large values of β effectively fix the nucleotide content at certain indices of sequences $\boldsymbol{x}$ sampled from p, one can examine the samples from constrained maximum entropy distributions at different values of β to uncover a hierarchy of important features in $\boldsymbol{x_o}$. We give examples of both methods in the following sections, where we sample the distribution (3) using MCMC (Methods). Figure 1, illustrates the sampling of sequences **x** according to their similarity to data set element $\boldsymbol{x_o}$ in the space of penultimate layer activations.

3.2 A constraint for genome nucleotide composition

In some cases, it is desirable for sampled sequences not only to satisfy the constraint (1), but also to reflect the background nucleotide frequencies of the genome from which the network's training/test data sets are derived. For example, in many genomes, the frequency of G or C nucleotides is significantly less than 50%. Unless average GC content of network inputs is important in determining the penultimate activation, samples from the distribution (3) will not preserve the GC content of the data set element $\boldsymbol{x_o}$. Since the number of sequences satisfying the constraint of a certain fraction of G or C nucleotides decays rapidly with departures from 50% GC content, many of the sequences sampled with the scheme developed above will correspond to sequences that may be undesirable, because they are too GC rich to be biologically relevant.

We control for GC content without diminishing the effect of constraint (1) on the maximum entropy distribution associated with an input

$\boldsymbol{x_o}$ by considering an arbitrary network input $\boldsymbol{x}$ to be embedded in a longer genomic sequence of length $M > L$, denoted $\boldsymbol{z}$. Let $\boldsymbol{y}$ denote the part of the length M sequence flanking $\boldsymbol{x}$ and let Y denote the set of all such flanking sequence. We seek a distribution over the set of length M sequences that maximizes (2) (with $\boldsymbol{x}$ replaced by $\boldsymbol{z}$) subject to the constraints (1) and

$$n = \langle N(\boldsymbol{x}) + N(\boldsymbol{y})\rangle_{x\in X\,,y\in Y} \tag{5}$$

where $N(\cdot)$ takes a sequence argument and returns the number of G and C nucleotides , $\langle\cdot\rangle_{x\in X\,,y\in Y}$ indicates expectation over the set of all length M sequences, and $n > 0$ is a constant. We apply the method of Lagrange multipliers again to obtain a new constrained maximum entropy distribution

$$p(\boldsymbol{z}) = \frac{1}{Z}\, e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)+\mu\left(N(\boldsymbol{x})+N(\boldsymbol{y})\right)} \tag{6}$$

where μ is the Lagrange multiplier associated with the constraint (5) and Z is a normalization constant. To determine the "chemical potential" μ, we write the expected GC content of sequence $\boldsymbol{z}$ as

$$\begin{aligned} n &= \frac{1}{Z}\sum_{x\in X\,,y\in Y}(N(\boldsymbol{x}) + N(\boldsymbol{y}))e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)+\,\mu\left(N(\boldsymbol{x})+N(\boldsymbol{y})\right)} \\ &= \frac{\partial}{\partial\mu}\log(Z), \end{aligned} \tag{7}$$

where Z can be rewritten as

$$\begin{aligned} Z &= \sum_{x\in X\;y\in Y} e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)+\,\mu\left(N(\boldsymbol{x})+N(\boldsymbol{y})\right)} \\ &= \left(\sum_{y\in Y} e^{\mu N(y)}\right)\left(\sum_{x\in X} e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)+\,\mu N(\boldsymbol{x})}\right) \\ &= \left(\sum_{N_y=0}^{M-L}\binom{M-L}{N_y} e^{\mu N_y} 2^{M-L}\right)\left(\sum_{x\in X} e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)+\,\mu N(\boldsymbol{x})}\right) \\ &= 2^{M-L}(1+e^{\mu})^{M-L}\left(\sum_{x\in X} e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)+\,\mu N(\boldsymbol{x})}\right). \end{aligned}$$

Using this expression for Z, equation (7) becomes

$$n = (M-L)\frac{e^{\mu}}{1+e^{\mu}} + \langle N(\boldsymbol{x})\rangle.$$

Diving by M and taking the limit $M \to \infty$, the second term on the right hand side tends to 0, and we obtain

$$\frac{n}{M} = \frac{e^{\mu}}{1+e^{\mu}} \tag{8}$$

where $\frac{n}{M}$ is the mean fraction of G and C nucleotides in sequence $\mathbf{z}$. This equation can be inverted to calculate μ for a given genome GC content.

With μ fixed as above, the distribution

$$p(\boldsymbol{x}) = \frac{1}{Z}\, e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)+\mu N(\boldsymbol{x})} \tag{9}$$

is sampled via MCMC (Methods). In section 3.5 we sample from (9) with the aim of extracting features from a network trained on input sequences from the *S. cerevisiae* genome. In that section, we show that our choice of $\mu = -0.49$, obtained from (8) using the 38% GC content of the *S. cerevisiae* genome, results in samples that on average possess appropriate GC contents for the genome, while also reflecting variations in the number of G and C nucleotides among the input sequences $\boldsymbol{x_o}$ satisfying the constraint (1).

3.3 Extracting features from samples

Given a collection of MCMC samples from distribution (3) or (9) associated with a network input $\boldsymbol{x_o}$, we extract the input features captured by the penultimate activation $\boldsymbol{\Phi}(\boldsymbol{x_o})$ by ploting the sample mean nucleotide frequency at each genomic index and also by examining the variance of linear combinations of nucleotide indicator variables that will serve to define sequence features of interest.

Sample mean nucleotide frequencies reveal important features due to the fact that input indices not important in determining $\boldsymbol{\Phi}(\boldsymbol{x_o})$ have mean nucleotide frequencies approximately equal to the probabilities implied by a max entropy distribution without the constraint (1). This averaging of unimportant input indices to "background" probabilities makes important input features apparent.

To assess the importance of a candidate input-wide feature in determining penultimate layer activations, we define a random variable V, measuring the concordance of sampled sequences $\boldsymbol{x}$ with the feature by:

$$V(\boldsymbol{x}) = \sum_{i=1}^{L} c_i\, I_i(x_i) \tag{10}$$

where x_i denotes the nucleotide at index i in $\boldsymbol{x}$, $I_i(\cdot)$, is the indicator variable for one of a set of nucleotides at index i and c_i is a real valued weight. For example, if we are interested in the importance of GC content $V(\boldsymbol{x})$ would be the sum in indicator variables for G or C nucleotides at each input index with unit weights. Intuitively, we expect the importance of the feature measured by V to decrease with increasing variance of this random variable. We can confirm our intuition and establish an equation relating the sample variance of V to the dependence of penultimate activations on the feature V as follows: the probability that the random variable V attains value v under the distribution (9) over the sequence space X is

$$\mathbb{P}_p(V = v) \propto \sum_{\boldsymbol{x}\in X_v} e^{\mu N(\boldsymbol{x})-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)} \tag{11}$$

where $X_v = \{\boldsymbol{x} \in X : V(\boldsymbol{x}) = v\}$. Two factors contribute to the probability mass at $V = v$:

- The factor $e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)}$ which is determined by the similarity of $\boldsymbol{x}$ to $\boldsymbol{x_o}$, as $\boldsymbol{x}$ varies over the set X_v
- The probability mass at $V = v$ implied by a maximum entropy distribution over X subject to the constraint on the mean value of $N(\boldsymbol{x})$, but not the constraint (1). We denote this distribution over X by $q(\boldsymbol{x})$.

Noting that $q(\boldsymbol{x}) \propto e^{\mu N(\boldsymbol{x})}$, we can control for the second factor listed above by multiplying and dividing by $q(\boldsymbol{x})$ in (11) to obtain

$$\mathbb{P}_p(V = v) \propto \left(\sum_{\boldsymbol{x}\in X_v} e^{\mu N(\boldsymbol{x})}\right) f(v) \tag{12}$$

where

$$f(v) = \left(\frac{\sum_{x\in X_v} e^{\mu N(x)} e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)}}{\sum_{x\in X_v} e^{\mu N(x)}}\right). \tag{13}$$

Note that $f(v)$ is the mean value of $e^{-\beta d\left(\boldsymbol{\Phi}(\boldsymbol{x_0}),\boldsymbol{\Phi}(\boldsymbol{x})\right)}$ under the distribution $q(\boldsymbol{x})$, conditioned on $x \in X_v$ and provides a good measure of the dependence of penultimate activations on the feature measured by V.

To obtain an approximate expression for $f(v)$ in terms of quantities measured by our MCMC sampling scheme, observe that the normalized probability mass $q(\boldsymbol{x})$ may be written as

$$q(\boldsymbol{x}) = \prod_{i=1}^{L} \frac{1}{2} \frac{e^{\mu I(x_i \in \{G,C\})}}{1+e^{\mu}}$$

where $I(x_i \in \{G,C\})$ is the indicator variable for G or C at index i. Each term in the product is the marginal distribution over nucleotides at index i implied by the distribution q. Thus, the x_i are mutually independent random variables under distribution q and so are the indicator variables $I_i(x_i)$ in (10). Moreover, since V is a linear combination of independent random variables with respect to distribution q, the Lindeberg version of the central limit theorem implies that if the number of indicators contributing to V is reasonably large and the weights c_i in (10) are of roughly the same magnitude then we can approximate

$$\mathbb{P}_q(V = v) = \sum_{\boldsymbol{x} \in X_v} q(\boldsymbol{x}) \approx \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - \langle V \rangle_q\right)^2} \tag{14}$$

where $\mathbb{P}_q$ denotes probability implied by the distribution q and $\langle V \rangle_q$ and σ^2 are the mean and variance of V under this distribution. Since

$$\sum_{\boldsymbol{x} \in X_v} q(\boldsymbol{x}) \propto \sum_{\boldsymbol{x} \in X_v} e^{\mu N(x)}$$

we can substitute the approximation (14) in (12) and take logs to obtain:

$$\log\big(f(v)\big) = \log(\mathbb{P}_p(V = v)) + \frac{1}{2\sigma^2}\left(v - \langle V \rangle_q\right)^2 + const.$$

In accordance with the approximation (14), we treat $\log(\mathbb{P}_p(V = v))$ as a smooth function of v and expand to second order in a Taylor series about the value $v = v^*$, which we assume maximizes $\log\big(f(v)\big)$:

$$\log\big(f(v)\big) \approx \log\big(f(v^*)\big) + \frac{1}{2}\left(\frac{1}{\sigma^2} + \frac{d^2}{dv^2} \log\left(\mathbb{P}_p(V = v)\right)\Big|_{v=v^*}\right)(v - v^2)$$

where the 1st order term is zero since we are expanding about a maximum and truncation of higher order terms is justified if the distribution $\mathbb{P}_p(V = v)$ is approximately normal, a condition which can be checked directly by estimating the distribution of V from MCMC samples. Moreover, in this case we can estimate

$$\frac{-1}{s^2} = \frac{d^2}{dv^2} \log\left(\mathbb{P}_p(V = v)\right)\Big|_{v=v^*}$$

where s^2 is the variance of V under the distribution p, estimated from MCMC samples. With this estimate and under the above approximations, we obtain:

$$f(v) \propto e^{-\frac{1}{2}\left(\frac{1}{s^2} - \frac{1}{\sigma^2}\right)(v - v^*)^2}$$

where the decay factor

$$\delta \equiv \frac{1}{s^2} - \frac{1}{\sigma^2} = \frac{1 - {s^2}/{\sigma^2}}{s^2} \tag{15}$$

measures the importance of the feature associated with V in determining the penultimate layer activations, with larger values of δ indicating greater importance.

In section 3.5, we apply this method to measure the importance of input sequences' GC content for predicting nucleosome positioning.

3.4 Application 1: Localizing learned motifs

We applied our interpretation method to a network trained on a benchmark motif discovery data set constructed by Zeng *et al.* (2016) from ENCODE CTCF ChIP-seq data (ENCODE Project Consortium, 2012). In this motif discovery task, the network distinguished elements of the positive class, consisting of 101 base-pair (bp) sequences centered on ChIP-seq peaks, from elements of the negative class consisting of positive class sequences

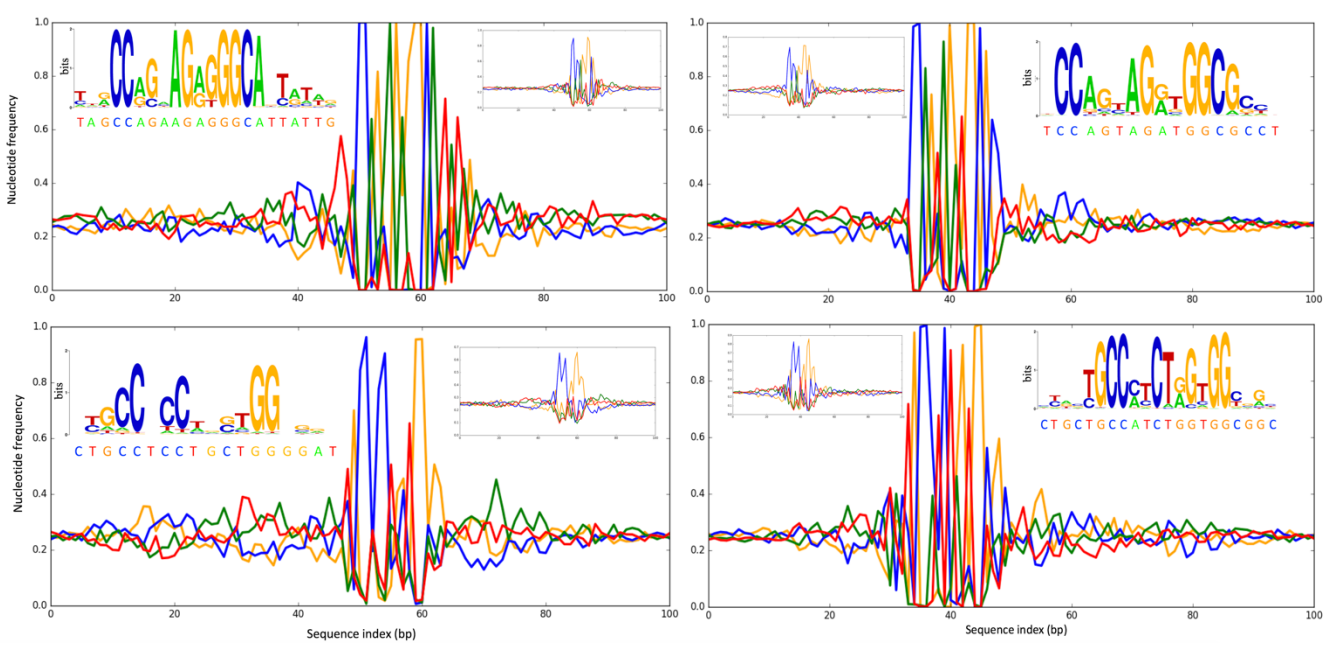

Fig. 2. Nucleotide frequencies for samples from network interpretation distributions associated with four randomly chosen input sequences containing a CTCF motif. Main plots indicate frequencies for samples collected with $\beta = 400$. Sequence logo insets illustrate network-derived importance of motif positions and nucleotide compositions, with the input sequence indicated below the x-axis. Nucleotide frequency plot insets are produced by sampling with $\beta = 100$, demonstrating the multiscale nature of our feature extraction method. Colors yellow, blue, green, red indicate G,C,A,T, respectively.

shuffled to maintain dinucleotide frequency. We represented network inputs with a one-hot encoding scheme and trained an architecture consisting of a convolutional layer of 64 convolutional filters each with a stride of 1 and taking 24 bps as input, followed by a layer of 100 units fully connected to the preceding layer and a two unit softmax output layer. During training, but not interpretation, the dropout method with probability 0.1 was applied to the output of the fully connect layer to reduce overfitting to the training data (Srivastava *et al.*, 2014). Rectified linear activation functions were used in all layers preceding the output. The trained network performed well achieving an area under the receiver operating characteristic (AUC) of 0.978 for the test set. (Methods).

We used the network interpretation method to analyze 2500 input sequences selected at random from the collection of test set elements in the positive class and correctly classified by the network. For each selected input $\boldsymbol{x_o}$, we sampled the distribution (3) with $\beta = 400$. This values of β was chosen so that for a given $\boldsymbol{x_o}$, the distance to the nearest test set element is, in general, 2 to 3 standard deviations away from the mean distance of MCMC samples associated with $\boldsymbol{x_o}$, where distances were measured in the space of penultimate layer activations according the weighted Euclidean metric d in defined in section 3.1.

Fig. 2 shows nucleotide frequencies as a function of sequence index for MCMC samples associated with four input elements chosen at random. In all four cases, the location of the motif identified by the network was indicated by an interval of single nucleotide frequencies that diverged dramatically from the uniform distribution over nucleotides implied by the distribution (3) for sequence locations with little effect on penultimate layer activations. Sequence logos were generated from the nucleotide frequencies on these intervals using WebLogo (Crooks *et al.*, 2004). Logos in the top row of Fig. 2 show the canonical CTCF motif, while logos in the bottom row are the reverse complements of the core canonical motif.

Fig. 3(A) shows the consensus motif discovered by the program MEME when it is run on a subset of the test inputs that we interpreted (Bailey *et al*., 2015). We produced a similar network-derived consensus motif by computing, for each analyzed input, the Kullback-Leibler (KL) divergence of sample nucleotide frequencies at each input index from a uniform distribution. The resulting arrays of KL divergences were scanned with 19 bp window and a network-derived motif was called for each analyzed input at the widow with largest average KL divergence. Fig. 3(B) shows the network-derived consensus motif produced by aligning these 19 bp windows with the multiple sequence aligner MUSCLE (Edgar, 2004) (Methods).

Finally, we scanned each of the 2500 analyzed inputs for the MEME consensus motif using the motif scanner FIMO (Grant *et al*., 2011). FIMO identified motifs in 1652 of these inputs. Fig. 3(C) shows the distribution of relative distances between the centers of FIMO motif calls and our network derived motif calls for these 1652 inputs (Methods).

3.5 Application 2: Extracting nucleosome positioning signals

We constructed a data set based on the chemical cleavage map of nucleosome dyads in *S. cerevisiae* (Brogaard *et al.*, *2012*). Each input was a 201 bp sequence with positive class elements centered on nucleosome dyads and negative class elements centered on indices sampled from genomic regions at least 3 bps from a dyad. We chose to allow sampling of negative sequences within the 73 bps of nucleosomal DNA flanking the dyad to encourage the network to learn features that direct precise nucleosome positioning as well as those determining nucleosome occupancy.

Our trained network consisted of a convolutional layer with 30 filters, each with a stride of 1 and taking 6 bp windows as input, followed by a 400-unit layer with full connections and a 2-unit output softmax layer. Sigmoid activation functions were used in all layers preceding the output. The trained network performed well, achieving an AUC of 0.956 on the test set (Methods).

We applied our network interpretation method to analyze 2500 input sequences randomly selected from validation set elements corresponding to nucleosomal sequences correctly classified by the network. For each selected input, we sampled the distribution (9) using $\mu = -0.49$ and $\beta = 40.0$. The values of μ is determined from (8) using the 38% GC content of the *S. cerevisiae* genome. The value of β was determined by examining the plots of nucleotide frequencies for a range of β values and selecting the largest value that permitted fluctuation in the nucleotide content at all of 201 bp locations among the MCMC samples collected.

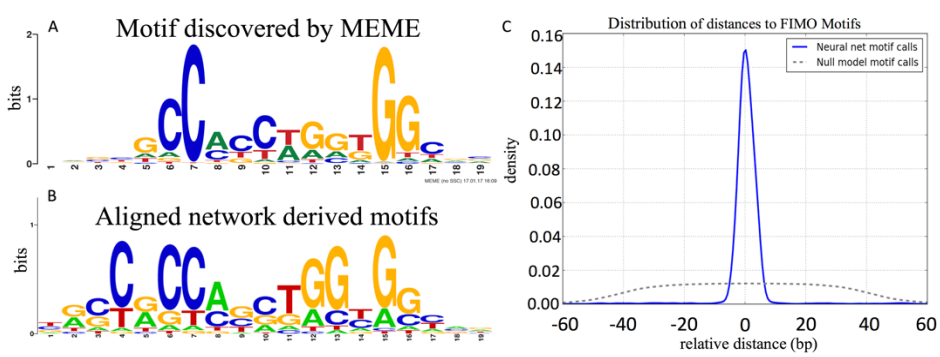

Fig 3. (A) CTCF motif discovered by MEME**. (B)** Network-derived consensus motif produced by aligning 2500, 19 bp motifs identified by network interpretation. **(C)** Distribution of relative distances (smoothed with kernel density estimation) between centers of network-derived motifs and 1652 motifs identified by the motif scanner FIMO.

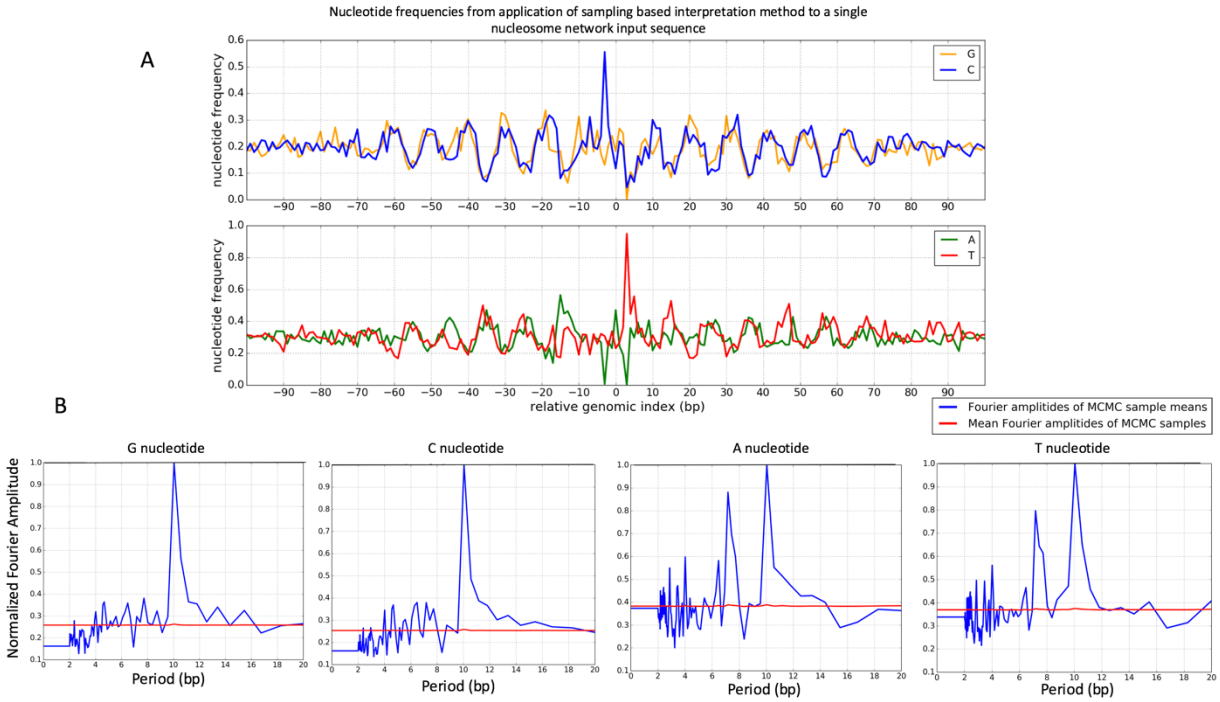

Fig 4. (A) Nucleotide frequencies for samples from the network interpretation distribution (9) associated with a single nucleosomal input sequence. **(B)** Fourier amplitudes of mean nucleotide content of MCMC samples associated with 2500 nucleosomal sequences (blue), and Fourier amplitudes of individual MCMC samples averaged over 2500 nucleosomal sequences (red). For display purpose, the y-axis was scaled by the largest amplitude.

Fig. 4(A) illustrates the sample nucleotide frequencies for one of the 2500 input analyzed. Several learned features can be inferred. For instance, the decay of nucleotide frequencies to background probabilities with distance from the dyad reflects the expected decay in the importance of nucleotide content outside the canonical 147 bp nucleosome sequence. Moreover, the apparent 10 bp periodicity of G/C and A/T signals demonstrates the importance of preferentially positioned G/C and A/T nucleotides to the classification rule learned by the network.

It is important to recognize that even though these plots of sample nucleotide frequencies illuminate learned features, they do not imply that the periodic features are present in all samples. Indeed, it has recently been shown that most nucleosomal sequences in *S. cerevisiae* do not contain significant 10 bp periodicity (Jin *et al*., 2016). We demonstrated that individual MCMC samples produced by our method were also not enriched for 10 bp periodicity by calculating the mean Fourier amplitudes of individual samples as well as the Fourier amplitudes of the sample mean nucleotide frequencies. Fig. 4(B) shows the two results averaged over the 2500 inputs analyzed. At 10bp period, the mean amplitude of individual samples was greatly suppressed relative to the amplitudes of sample nucleotide frequencies. This result implies that, like the *S. cerevisiae* input nucleosomal sequences themselves, individual MCMC sampled sequences do not need to possess true 10 bp periodic nucleotide content to be classified as nucleosomal by the network; rather, it is enough for each sequence to possess G/C and A/T nucleotides only at *some* of the locations separated by 10 bps potentially corresponding to histone-DNA contact points. Our deep learning study thus confirms the previous finding that periodicity arises mainly from aligning multiple nucleosomal sequences rather than being an intrinsic property of individual nucleosomal DNA (Jin *et al*., 2016).

It is also widely believed that nucleosomal DNA often possesses high GC content relative to the genomic background (Hughes and Rando, 2014); we thus explored the importance of GC content to our network's classification. Fig. 5(A) shows the mean GC content of collections of MCMC samples associated with 1000 inputs selected at random from the 2500 inputs analyzed. Sample mean GC contents generally agreed with the 38% GC content associated with our choice of μ; however, sample mean GC content also correlated with the GC content of the associated input. The correlation indicated that changes in GC content affected the penultimate layer activations to the extent that samples tended to preserve the GC enrichment/depletion of their associated input.

To rigorously measure the importance of the GC content feature, we defined a random variable V that sums the indicators for G or C nucleotides at each of the central 147 bp of the network input. For comparison, we also defined random variables measuring "dummy features" which also sum indicator variables at each of the central 147 bp of network input but where, at each position, the set of two nucleotides for which the indicator is 1 is uniformly sampled from the list {G,C}, {G,A}, {G,T}, {C,A}, {C,T}, {A,T}. For 1000 inputs chosen at random from the 2500 analyzed, we calculated the feature importance score δ, defined in (15), for the GC content random variable V and for 300 random variables measuring dummy features. We then computed the percentile of the importance score of the GC content variable in the distribution of importance scores of the dummy feature variables for each input. Fig. 5(B) shows the distribution of these percentiles, with enrichment of nucleosomal sequences near the 100th percentile; setting a threshold at the 90th percentile in the distribution of dummy feature importance scores, we estimate that GC content is a learned network feature of about 26% of the 1000 nucleosomal sequences analyzed

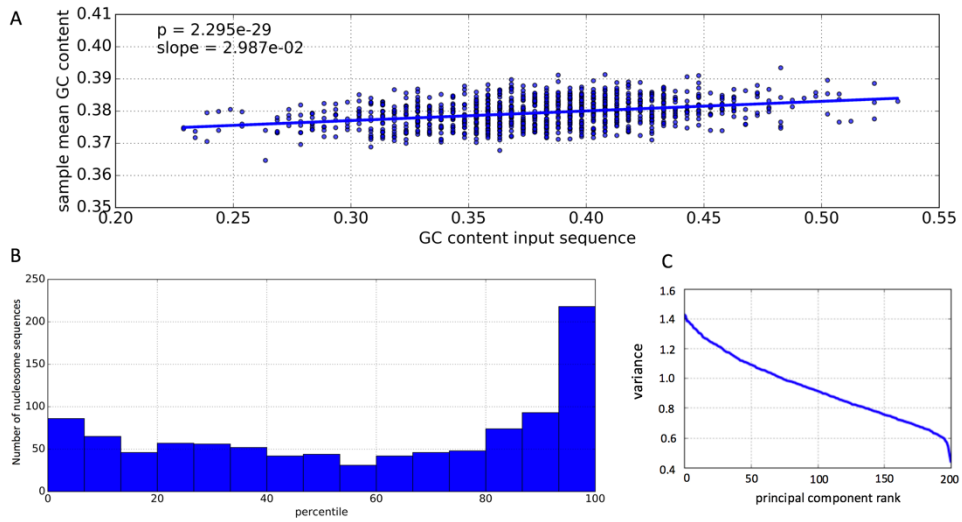

Fig. 5. (A) Distribution of GC content of 1000 network input sequences corresponding to nucleosomal DNA and the mean GC content of samples associated with these inputs. **(B)** Histogram of the percentiles of GC feature importance scores in the distribution of importance scores of 300 "dummy" sequence features. Histogram summarizes percentiles of GC feature importance scores for 1000 nucleosomal sequences. **(C)** Example of decay in variance associated with ranked principal component vectors in PCA analysis of samples from the network interpretation distribution (9).

Finally, we performed *de novo* extraction of the most important features learned by the network. Reasoning that the distribution (9) should decay vary rapidly with changes in sequence features most important in determining penultimate activation, we approximated the sequence feature importance measure (15) by

$$\delta \equiv \frac{1-{s^2}/{\sigma^2}}{s^2} \approx \frac{1}{s^2},$$

where s^2 is the variance of a sequence feature random variable estimated from MCMC samples (see section 3.3). Under this approximation the most important features are given by the lowest variance principal components (PCs) of our MCMC samples. We performed principal component analysis (PCA) by representing each sampled sequence as a length 201 vector with a binary encoding of components that maps G and C nucleotides to 1 and A and T nucleotides to -1. Fig. 5(C) shows an example where a sharp decrease is seen in the lowest variance of ranked PC vectors. We empirically observed that this sharp drop, signaling the prominent importance of features corresponding to the lowest variance PC vectors, is typical for our samples.

We demonstrated the utility of these features by constructing a simple 10-nearest neighbor classifier, where we used the lowest variance PCs to compute the inter-sequence distance. Briefly, we randomly selected 1200 correctly classified nucleosomal input sequences to which we applied our interpretation method, took the mean of binary encoded MCMC samples associated with each sequence as exemplars for the positive class in our nearest neighbor classifier, and calculated the 5 lowest variance PC vectors associated with each sequence's MCMC samples. We repeated this task for 1200 correctly classified non-nucleosomal sequences, sampling distribution (9) with the values of β and μ given above, and taking the mean of binary encoded MCMC samples as exemplars of the negative class. We performed classification on a balanced test set of 10500 sequences by projecting the vectors joining an exemplar and test set elements onto the space spanned by the exemplar's 5 lowest variance PC vectors, scaling the projected coordinates by the inverse standard deviation of the associated PC vectors, and then computing Euclidean distance. Test set elements were assigned to the majority class of the 10 nearest exemplars. This simple method yielded a classification accuracy of 76%. For comparison, we repeated this classification replacing the 5 lowest variance PC vectors of each exemplar with 5 mutually orthogonal vectors randomly sampled from the 201 dimensional binary space (Methods). Using this control, nearest neighbor classification accuracy dropped to 51%. This result thus demonstrates the ability of our interpretation method to extract *de novo* features used in the neural network's classification.

4 DISCUSSION

Deep neural networks provide researchers with powerful tools for making predictions based on complex patterns in biological sequence. Methods for extracting learned input features from these networks can provide valuable scientific insights, and several efforts in this direction have made deep learning an even more appealing approach for tackling complex problems in genomics and other scientific disciplines.

We have contributed to these efforts by introducing a novel feature extraction method based on sampling a maximum entropy distribution with a constraint imposed by the empirical non-linear function learned by the network. From a theoretical standpoint, this constraint allows the derivation of relationships between the statistics of the sampled distribution and the dependence of network classification on specific sequence features. In particular, we have developed a scheme for assessing feature importance that has been difficult to measure otherwise with currently available approaches to network interpretation. From a practical standpoint, we showed that our method identifies both canonical TF binding motifs as well as reverse complements learned by a network trained on a motif discovery task. We also demonstrated a method for extracting *de novo* features learned by a network trained to identify nucleosome positioning sequences and showed the utility of these extracted features by employing them in a successful nearest neighbor classifier.

The success of our maximum entropy approach signals that statistical physics may have much to contribute to the task of interpreting deep learning models. Indeed, a central goal of statistical mechanics is to understand constrained maximum entropy distributions of many degrees of freedom that interact according to known microscopic rules. While our formulation addresses an inverse problem of inferring unknown

characteristics of network inputs from observed statistics of a constrained maximum entropy distribution, statistical mechanics provides a wide range of tools that could be further explored in this new context. This theoretical paper provides an example of the growing synergy between machine learning and physics towards assessing the role of diffuse and subtle sequence features that direct important biological outcomes, such as the positing of nucleosomes.

Acknowledgements and Funding

We thank Miroslav Hejna , Hu Jin and Wooyoung Moon for helpful discussions.

This work has been supported by the National Science Foundation (DBI-1442504), the National Institutes of Health (R01CA163336), and the Founder Professorship from the Grainger Engineering Breakthroughs Initiative.

REFERENCES

Alipanahi, B., *et al.* Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol* 2015;33(8):831-838.

Bach, S., *et al.* On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS One* 2015;10(7):e0130140.

Bailey, T.L., *et al.* The MEME Suite. *Nucleic acids research* 2015;43(W1):W39-49.

Brogaard, K., *et al.* A map of nucleosome positions in yeast at base-pair resolution. *Nature* 2012;486(7404):496-501.

Chollet, F. Keras. GitHub, 2015; https://github.com/fchollet/keras.

Crooks, G.E., *et al.* WebLogo: a sequence logo generator. *Genome Res* 2004;14(6):1188-1190.

Edgar, R.C. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics* 2004;5:113.

ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012;489(7414):57-74.

Grant, C.E., Bailey, T.L. and Noble, W.S. FIMO: scanning for occurrences of a given motif. *Bioinformatics* 2011;27(7):1017-1018.

Hughes, A.L. and Rando, O.J. Mechanisms underlying nucleosome positioning in vivo. *Annu Rev Biophys* 2014;43:41-63.

Jin, H., Rube, H.T. and Song, J.S. Categorical spectral analysis of periodicity in nucleosomal DNA. *Nucleic acids research* 2016;44(5):2047-2057.

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

Lundberg, S. and Lee, S.-I. An unexpected unity among methods for interpreting model predictions. In, *NIPS 2016 Workshop on Interpretable Machine Learning in Complex Systems*. 2016.

Metropolis, N., *et al.* Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* 1953;21(6):1087-1092.

Shrikumar, A., *et al.* Not Just a Black Box: Learning Important Features Through Propagating Activation Differences. 2016(arXiv:1605.01713 [cs.LG]).

Srivastava, N., *et al.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 2014;15:1929-1958.

van der Walt, S., Colbert, C. and Varoquaux, G. The NumPy array: a structure for efficient numerical computation. *Computing in Science and Engineering* 2011;13(2):22-30.

Zeng, H., *et al.* Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics* 2016;32(12):i121-i127.

Zhou, J. and Troyanskaya, O.G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat Methods* 2015;12(10):931-934.