Fully Interpretable Deep Learning Model of Transcriptional Control

Yi Liu
Department of Statistics
University of Chicago
Chicago IL USA
yil@uchicago.edu

Kenneth Barr
Department of Human Genetics
University of Chicago
Chicago IL USA
barr@uchicago.edu

John Reinitz
Departments of Statistics,
Ecology and Evolution,
Molecular Genetics & Cell Biology
Institute of Genomics and Systems Biology
University of Chicago
Chicago IL USA
reinitz@uchicago.edu

Abstract

The universal expressibility assumption of Deep Neural Networks (DNNs) is the key motivation behind recent work in the system biology community to employ DNNs to solve important problems in functional genomics and molecular genetics. Because of the black box nature of DNNs, such assumptions, while useful in practice, are unsatisfactory for scientific analysis. In this paper, we give an example of a DNN in which every layer is interpretable. Moreover, this DNN is biologically validated and predictive. We derive our DNN from a systems biology model that was not previously recognized as having a DNN structure. This DNN is concerned with a key unsolved biological problem, which is to understand the DNA regulatory code which controls how genes in multicellular organisms are turned on and off. Although we apply our DNN to data from the early embryo of the fruit fly *Drosophila*, this system serves as a testbed for analysis of much larger data sets obtained by systems biology studies on a genomic scale.

1 Introduction

A central unsolved problem in biology is to understand how DNA specifies how genes turn on and off in multicellular organisms. The "universal expressibility" of deep neural nets suggests that they might be a valuable tool in this undertaking, but their applicability and acceptance in solving problems in natural science has been limited by the uninterpretability of their internal computations. We address both of these areas in this report by describing the reimplementation of a specific and highly predictive model of transcriptional control as a deep neural net (DNN). The model, chemical in nature, has a feedforward mathematical structure in which every layer has a specific scientific interpretation. When translated into DNN formalism, it provides an example of a DNN in which the internal structure is well understood and which may be of value to the machine learning field.

Deep learning has been widely deployed in genomics and systems biology over the last few years [11, 2, 27, 37, 9, 43, 39, 13, 35, 34]. Many of the developed tools have been highly successful in classification problems such as the identification of binding sites, open regions of chromatin, and the

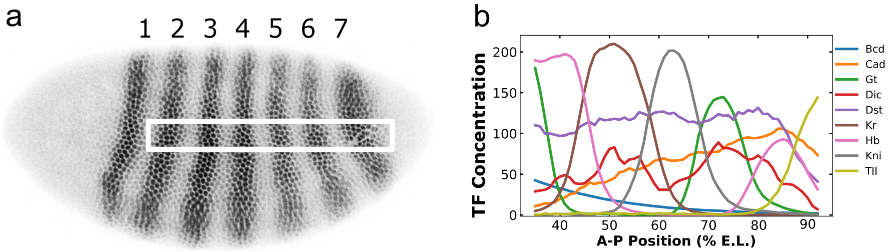

Figure 1: (a) shows a *Drosophila* embryo about 3 hours after fertilization which has been stained for Eve protein as described [48]. Anterior is to the left and dorsal is up. The dark shades indicate the concentration of Eve protein and the stripes are numbered. The white box is the one dimensional region of interest used to generate the data[20]. (b) The TF concentrations found across the embryo [48]. In the graph, 58 data points are shown, corresponding to 58 nuclei on the A-P axis. Each nucleus is 1 % E.L. in size. The identity of TFs is shown in the key; the horizontal axis shows position in percent egg length (E.L.) and the vertical axis shows protein concentration.

location of enhancers. Deeper understanding requires more quantitative studies. One recent example that goes beyond classification concerns a fully quantitative and highly predictive DNN model of the role of untranslated RNA leader sequences in gene expression in yeast [11], We believe that these studies have two sets of limitations. First, they take a universal expressibility approach without much understanding and interpretation of the underlying chemical and biological mechanisms giving rise to to the phenomena under study. This limits the contributions DNNs can make to increasing human understanding of fundamental biological processes. Here we consider a DNN in which each layer has a specific chemical or biological interpretation. Studies of regulatory DNA with DNNs have treated only the sequence itself, but in metazoa (multicellular animals) different cell types have very different gene expression states although they contain the same DNA. Here we consider a DNN in which state is described not only by the sequence, but also by the set of regulatory proteins present in each cell. We now describe the problem to be solved in both mathematical and biological terms.

Mathematically, the "expression" of a gene is the rate at which it synthesizes mRNA (the "transcript") from a complementary DNA template. This rate $d[\text{mRNA}]/dt = f(\boldsymbol{D}, \boldsymbol{v})$ where $\boldsymbol{D} = (D_i)$, where each $D_i \in \{A, C, G, T\}$ is a base in the sequence of regulatory DNA, and $\boldsymbol{v} = (v_1, \ldots, v_a, \ldots, v_n)$, where each v_a is the nuclear concentration of a regulatory DNA binding protein known as a "transcription factor" (TF). The machine learning task is to learn the function f from a series of observations $(\boldsymbol{D}_j, \boldsymbol{v}_{jk})$ of the expression of sequence j in cell type k, where $k \in \{1, \ldots, M\}$. The essence of the scientific problem is that each sequence $\boldsymbol{D}_j$ must express correctly in each cell type, reflecting the fact that in a multicellular organism, different sets of genes are expressed in different cell types, but each cell type contains the same DNA.

Biologically, regulatory DNA is noncoding DNA which can be upstream (5'), downstream (3'), or within (intronic) the complementary mRNA template, but it is distinct from (exonic) DNA that contains codons that specify the amino acid sequence of the gene's protein product. TFs bind to DNA in its double stranded form, in which each strand has a complementary base (A and T; G and C) at each position and the two strands have opposite 3'-5' orientations. In metazoans, regulatory DNA is frequently much larger than the coding portion of the gene. The regulatory DNA contains segments of 500 to 1000 base pairs (bp) called "enhancers", each of which directs expression in a particular domain or tissue type. In this study, we consider a gene called *eve* which acts in the early embryo of the fruit fly *Drosophila melanogaster*, at which time it forms a pattern of seven stripes as shown in Figure 1. The entire gene is 16.5 kilobases (kb) of DNA in length, but the mRNA transcript is only 1.5 kb (Figure 2). We consider the action of *eve* from 1 to 3 hours after the start of embryonic development. At this time the embryo is a hollow ellipsoid of cell nuclei that can be treated like a naturally grown gene chip in which $d[\text{mRNA}]/dt$, $\boldsymbol{D}_j$, and $\boldsymbol{v}_{jk}$ are fully observable at a quantitative level. The embryo contains two orthogonal axes in the anterior-poster (A-P) and dorsal-ventral (D-V) directions. In the central portion of the embryo, gene expression on the two axes is uncoupled, so cell type and gene expression can be visualized by plotting relevant state variables in one dimension.

In this paper, we show that DNNs can be used to generate a predictive model of gene expression. Our starting point is a previously published model of transcriptional control [23], which is one of a

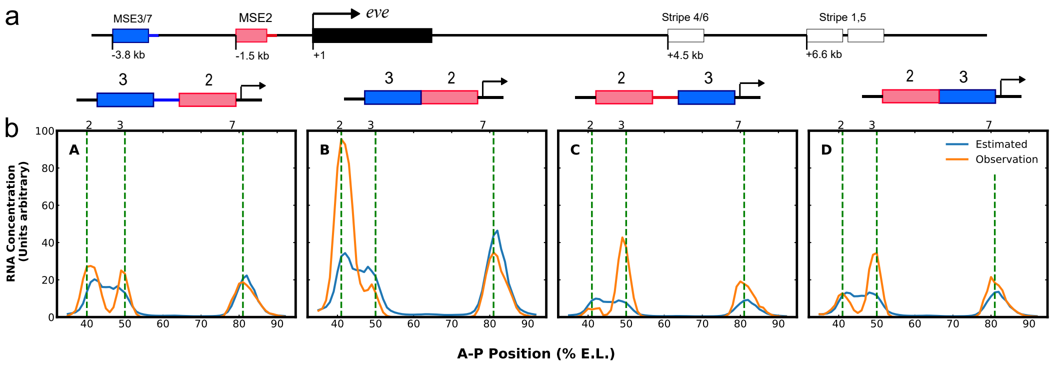

Figure 2: (a): The figure shows a diagram of the *eve* locus. The transcript is indicated by black box; enhancers are indicated by pink, blue or white boxes and are labeled with the *eve* stripe that they drive. (b): Fusion constructs that are used in the training process. The blue box represents the MSE3/7 enhancer and the red box represents the MSE2 enhancer. The four graphs in (b) shows the data used for training and outcome from the model after 500 epochs using Adam. In each graph, 58 data points are shown, corresponding to 58 nuclei on the A-P axis. Each nucleus is 1 % E.L. in size. The orange lines show the observed data from the experiments [23] and the blue lines show the model output. The overall RMS is of the training is calculated to be 10.4.

family of so-called thermodynamic transcription models [38, 21, 42, 40, 22, 16, 23, 32, 41, 4, 5]. In these models, occupancy of DNA by TFs is calculated using thermodynamics, and phenomenological rules are used to calculate the transcription rate from the configuration of bound factors. Like DNNs, thermodynamic transcription models have a feedforward structure that can be described in layers. The form of the resulting equations makes back propagation and hence SGD difficult or impossible because of the need to hand code complex partial derivatives, so these models are optimized by zero order methods such as Simulated Annealing or Genetic Algorithms. Despite this apparent mathematical distinction, we were able to translate the chemical model of transcription into a standard DNN form that could perform rapid learning by SGD.

The resulting model is, to our knowledge, one of the first fully interpretable DNNs with an exact interpretation for each of the unknown parameters. It is also an example of a biologically validated DNN that is not perceptron based. Some of the parameters can be and are extracted from independent experiments, resulting in a very small number of parameters for an extremely deep network. Although theories of metazoan transcriptional control have largely been developed in the fruit fly *Drosophila* because of its unique experimental advantages, they have also been applied to mouse [6]. Below, in Section 2, we will describe the function of each layer and the interpretation of parameters. In section 3, we focus on the training and evaluation of performance. Finally, in section 4, we discuss the scientific implications of our results.

2 Understanding each layer

The model's input is DNA sequence and TF concentration. The TFs have functional roles which, in the present application, are known from independent experiments. Activators activate transcription, quenchers suppress the action of activators over a limited range, and certain activators can also convert nearby quenchers into activators. In each of these regulatory mechanisms, multiple bound TFs are required to perform a regulatory action. We perform this calculation as follows. Binding site locations and affinities are determined from sequence. Together with TF concentrations, this information is used to calculate the occupancy of each binding site. We then calculate the effects of coactivation, followed by the effects of quenching. The total amount of remaining activation is then summed and passed through a thresholding function. We describe each step below.

2.1 Computing Fractional Occupancy

2.1.1 Identifying the binding sites

An indicator representation of DNA sequence is used as input. The column index is the base pair position number in the sequence. The row index indicates which of the 4 bases (A,C,G,T) is observed. In the case where the sequence cannot be identified, a fifth row (N) is used. For example, if we have a sequence of ACTNGTTA, the corresponding matrix is

$$\begin{pmatrix} \text{A} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \text{C} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{G} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \text{T} & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ \text{N} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The nine TFs of interest are Bicoid (Bcd), Caudal (Cad), Drosophila-STAT (Dst), Dichaete (Dic), Hunchback (Hb), Krupple (Kr), Knirps (Kni), Giant (Gt), and Tailless (Tll) (Figure 1b). The identification and affinity characterization of binding sites for TF a requires a convolution layer with a Position Weight Matrix (PWM$_a$) as its kernel. The PWM can be understood as a convolution kernel in which the number of columns is the number of nucleotides of DNA in physical contact with a bound TF. Padding is not required for the columns. In the work described here, we use the same high quality experimentally obtained PWMs as previously [23]. Unlike many convolution matrices used in deep learning, PWMs make direct experimental predictions about DNA properties, a fact used in a previous study to experimentally confirm PWMs obtained by deep learning techniques [11].

Chemically, the PWM represents an additive model of binding in which the Gibbs free energy ΔG of binding is the sum of the free energies of binding to each nucleotide. Statistically, the PWM score can be regarded as the likelihood of finding a binding site at a given position, calculated using a variational approximation of the likelihood using only the marginal likelihood of each base. The resulting score, denoted by $S_{i,i+m;a}$, is an affine transformation of the the free energy $\Delta G_{i,i+m;a}$ of TF a binding to a site extending from base i to base $i+m$. In many cases, including most of the TFs considered here, m can be read off directly from DNAse I footprints [44, for example]. TFs physically bind in the major groove of the DNA double helix but the PWM is convolved with only a single strand. We compensate for this fact by also scoring the complementary strand, in which bases are replaced by their complements (A → T; C → G; T → A; G → C), and orientation is reversed. Scoring each strand results in a $1 \times n$ array of scores S_a, where n is the sequence length. At each base position, we set the score to be the larger of the two scores at that position.

We next calculate the equilibrium affinity $K_{i,i+m;a}$ of each binding site. If $\Delta G_{i,i+m;a}$ were known in units of kcal/mole, then $K_{i,i+m;a} = \exp(\Delta G_{i,i+m;a}/RT)$, where R is Boltzmann's constant and T the absolute temperature. $S_{i,i+m;a}$ is related to $\Delta G_{i,i+m;a}$ by an affine transformation $ax+b$ in which b is found from experiment and a is learned by training as follows. Our convolution kernel is accompanied by a fixed bias $S_{\max;a}$ which is the maximum possible score for any particular TF a. We only consider binding to sites with scores greater than zero, so we pass the exponentiated free energy through a ReLU and apply an indicator function to assure that below threshold sites disappear, so that

$$K_{i,i+m;a} = \exp\left(-\frac{(\texttt{ReLU}(S_{i,i+m;a}) - S_{\max;a})}{\lambda_a}\right)\mathbb{I}(S_{i,i+m;a} > 0), \tag{1}$$

where λ_a is a learnable parameter for each TF a. The $K_{i,i+m;a}$, arranged according to positions on the DNA sequence, produce another $1 \times (n-m)$ vector K_a. Adjusting for the size of m, we further concatenate K_a for all TFs to produce K, which we use for the calculation of fractional occupancy.

2.1.2 Belief propagation and partition function computation

Moving from affinity $K_{i,i+m}$ to fractional occupancy requires the consideration of all possible states in which the TFs can bind on the DNA. The fractional occupancy $f_{i,i+m;a}$ denotes the average occupancy of the site at equilibrium or the probability of finding the specific protein a at the site $(i, i+m)$ at a given instant. The calculation requires consideration of interactions between sites. Two overlapping sites cannot be occupied at the same time, and in some cases a TF bound at one site increases the binding affinity at a nearby site by a factor of w_{ij}^{coop}. In the present application, cooperativity only occurs between pairs of bound Bcd less than 60 bp apart. This calculation is best performed by computing the partition function Z, which can be calculated by a fast, recently discovered algorithm [4, Appendix S1]. The algorithm is essentially a form of Belief propagation which can be represented as a bi-directional RNN across K, and is given in Algorithm 1.

Algorithm 1 The Algorithm for Fractional Occupancy

```
Initialize $Z_N^- = 1$ and $Z_0^+ = 1$
**for** $i \leftarrow 1$ to $N$ **do**
    $q_{i,a} = [TF]_a K_{i,a}$
    **for** $a \in \{\text{Transcription Factors}\}$ **do**
        $Z_{i,a}^{+nc} = q_{i,a} Z_{i-m-1}^+$
        $Z_{N-i,a}^{-nc} = q_{N-i,a} Z_{N-i+m+1}^-$
        $Z_{i,a}^{+c} = \sum_{j=m+1}^{c_d} w_{ij}^{\text{coop}} q_{i,a} q_{i-j,a} Z_{i-j-m-1}^+$
        $Z_{i,a}^{-c} = \sum_{j=m+1}^{c_d} w_{ij}^{\text{coop}} q_{N-i,a} q_{N-i+j,a} Z_{N-i+j+m+1}^-$
    **end for**
    $Z_i^+ = \sum_a Z_{i,a}^{+nc} + Z_{i,a}^{+c}$
    $Z_{N-i}^- = \sum_a Z_{i,a}^{-nc} + Z_{i,a}^{-c}$
**end for**
**return** $Z_a^{+nc}$,$Z_a^{+c}$,$Z_a^{-nc}$,$Z_a^{-c}$, $Z_0^-$
```

The fractional occupancy is then given by

$$f_{i,i+m;a} = \frac{Z_{i,i+m,a}^{+nc} Z_{i,i+m,a}^{-c} + Z_{i,i+ma}^{-nc} Z_{i,i+m,a}^{+c} + Z_{i,i+ma}^{-nc} Z_{i,i+m,a}^{+nc}}{Z_0^-}. \tag{2}$$

2.2 TF-TF Interactions

Interactions between bound TFs follow phenomenological rules. A central feature of the *cis*-regulatory DNA of metazoan genes is the fact that biological function is encoded in multiple binding sites [47, 44, 45]. This fact is expressed mathematically in the phenomenological equations below. In the present application, bound TFs have specific roles derived from specific experimental results, although this approach also works if the roles are not known a priori [6]. Here, Bcd, Cad, Dst and Dic are activators; Hb, Tll, Kni, Kr and Gt are quenchers; and both Bcd and Cad are coactivators of Hb. We now describe the actions of each class of TF in the order which we compute them. The ultimate goal of this computation is to obtain the summed action of activators after their contribution has been increased by coactivation and diminished by quenching.

2.2.1 Coactivators

Coactivators turn a nearby quencher into an activator. This action is described by the equation [23, 4]

$$\hat{f}_i^{Q_C} = f_i^{Q_C} \prod_{j=i-k}^{i+k} (1 - d_c(j) E_C^{Q_C} f_j^C), \tag{3}$$

where $\hat{f}_i^{Q_C}$ is the portion of activator fractional occupancy created from the total fractional occupancy $f_i^{Q_C}$. $d_c(j)$ is a convolutional kernel describing the distance dependence of coactivation. $d_c(j) = 1$ if $\mid j - i \mid$ less than 156 bp. In this application, $k = 206$. As $\mid j - i \mid$ increases to 206 bp, $d_c(j)$ decreases linearly to 0. $E_C^{Q_C} \in [0, 1]$ denotes the relative strength of the coactivators. f_j^C simply refers to fractional occupancy of Bcd and Cad since they are the only coactivators in this setting.

It is easy to observe that equation (3) is the first term of a Taylor expansion of

$$\hat{f}_i^{Q_C} \approx f_i^{Q_C} \prod_{j=i-k}^{i+k} (1 - E_C^{Q_C} f_j^C)^{d_c(j)} \Rightarrow \hat{f}_i^{Q_C} = \exp\left(\sum_{j=i-k}^{i+k} d_c(j) \log(1 - E_C^{Q_C} f_j^C) \right) f_i^{Q_C}. \tag{4}$$

This can be turned into three convolutional linear activation units, so that

$$y_j = \log(1 - E_C^{Q_C} f_j^C); \quad z_i = \exp\left(\sum_{j=i-k}^{i+k} d_c(j) y_j\right); \quad \hat{f}_i^{Q_C} = f_i^{Q_C} z_i. \tag{5}$$

2.2.2 Quenchers

As their name suggests, quenchers suppress activation. Their action is local and occurs only within around 100 to 150 base pairs [17]. The basic mathematical formulation of the effect of nearby quenchers on the activator at position i is given by [23]

$$\hat{f}_i^A = f_i^A \prod_{j=i-k}^{i+k} (1 - d_q(j) E_A^Q f_j^Q), \tag{6}$$

where f_i^A is the fractional occupancy of activators whether coactivated or not, $\hat{f}_i^A$ is the fractional occupancy of activators after quenching, f_j^Q is the fractional occupancy of a quencher bound at position j. $E_A^Q \in (0, 1)$ is the strength of quencher Q on activator A and $d_q(j)$ is a convolutional kernel representing the range of quenching on the DNA strand. $k = 150$ bp, and $d_q(j) = 1$ when $\mid j - i \mid \leq 100$ and goes linearly down to 0 from 100 to 150 bp. Using a mathematical argument similar to that of the previous section, we can write

$$\hat{f}_i^A \approx \exp\left(\sum_{j=i-k}^{i+k} d_q(j) \log(1 - E_A^Q f_j^Q)\right) f_i^Q. \tag{7}$$

This gives three convolutional linear activation units.

$$y_j = \log(1 - E_A^Q f_j^Q); \quad z_i = \exp\left(\sum_{j=i-k}^{i+k} d_q(j) y_j\right); \quad \hat{f}_i^Q = f_i^A z_i. \tag{8}$$

2.2.3 Activation

Last, we sum the fractional occupancies of all activators remaining after the previous two steps. We consider the activators to lower the energy barrier in a diffusion limited Arrhenius rate law [4], which has the mathematical form of a sigmoidal thresholding function. This results an fully connected layer which yields the mRNA synthesis rate. In the experimental system used, mRNA has a life time much shorter than the time required to change transcription rates, so that the mRNA concentration $[\mathtt{mRNA}]$, an experimentally observable quantity, is given by

$$[\text{mRNA}] \propto \frac{d[\text{mRNA}]}{dt} = R\left(\frac{\exp\left(\sum_{j\in\{A,Q_C\}} E_j \sum_i \hat{f}_i^j - \theta\right)}{1 + \exp\left(\sum_{j\in\{A,Q_C\}} E_j \sum_i \hat{f}_i^j - \theta\right)}\right). \tag{9}$$

Here $E_j > 0$ represents the activating strength of each activator and is obtained by training on the data. θ, also obtained by training, is the amount of activation in the absence of activator. R is the maximum synthesis rate, and we train on the target

$$[\text{mRNA}]_{\mathtt{train}} = \frac{[\text{mRNA}]_{\mathtt{cell}}}{\sqrt{(}\sum_{\mathtt{cells}} [\text{mRNA}]^2_{\mathtt{cell}})}. \tag{10}$$

3 Implementation, Training, and Results

We implemented and trained this model in Keras with a TensorFlow back end ([10, 1]).The resulting architecture is shown in Figure3. The resulting model is deemed by Keras to have 223 Layers and 52 unknown parameters. As this does not use any of the conventional architecture ([25, 19]), we frequently use the `Lambda` functions in Keras to represent some of the activation functions and PWM convolutions. Algorithm 1 was implemented as a special layer in Keras with two `rnn` functions.

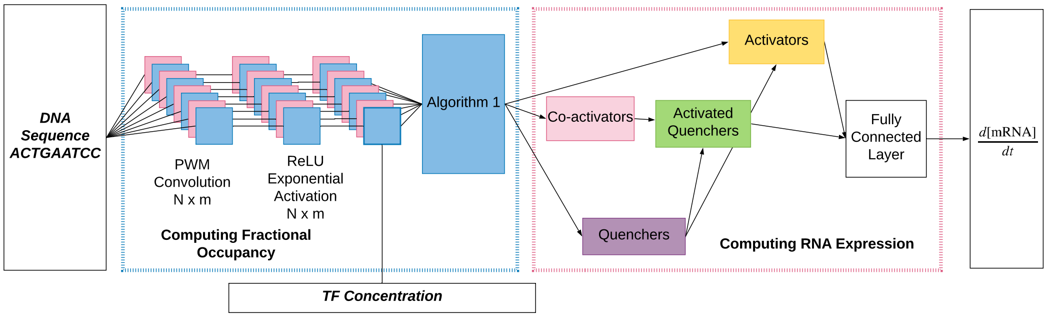

Figure 3: This is a graphical representation of the DNN. (Left) Chemical calculations. The computation can be seen as the DNA going through a convolution and then passing through a ReLU and then the the $(\exp(\cdot))$ activation function with bias. (Right) The graphical representation of interactions between bound TFs. Coactivators activate quenchers, quenchers quench activation, and activators combined together in a fully connected layer produce mRNA.

The training data and PWMs was as previously described [23], although training data was limited to the fusion constructs M32, M3_2, M23, and M2_3. The model was trained using a single Intel Core i7-8700K CPU. The training data contains 232 observations. The model was trained with 500 epochs using Adam ([24]) with Nesterov momentum as implemented in Keras. Training took approximately 4 hours. This compares with several days of serial simulated annealing before Algorithm 1 was devised [23], and is about equivalent to the time taken by code using Algorithm 1 running in parallel with the loss function for each construct computed on a separate core. However, optimization by simulated annealing requires several million evaluations of the loss function. while the implementation use here required about 10000. The earlier work used about 10000 lines of compiled C++ while this work uses less than 1000 lines of Python. However, the execution times indicate that there is considerable scope for improvement of the TensorFlow back end for this type of problem. Moreover, we see evidence of numerical issues in the Tensorflow back end which constrained us to make binding sites for all TFs the same size.

The results of the training are shown in Figure 2. We selected this training set, a pair of enhancers which produce greatly altered gene expression when fused, because they are known to produce a constrained model that is highly predictive [23]. It is important to note that this model tends not to over-fit the data since the number of parameters is only 52 compared to the 232 nuclei we used as the training set.

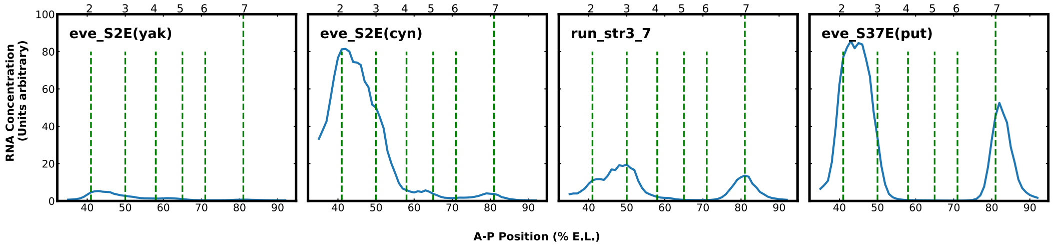

Figure 4: The figure shows four examples of predictions driven by enhancers not used for training. The location of eve stripes 2 through 7 are shown by vertical dashed lines. The vertical axis shows predicted mRNA concentration and horizontal axis shows A-P position in % E.L. The enhancers shown are described in the text.

In a practical setting, the model should generate biologically accurate relative mRNA concentration at the right A-P position in % E.L. given a sequence not in the training set. We tested the predictive power of our DNN by confronting it with set of enhancer sequences which the model has not previously seen. In Figure 4, we show examples of four types of predictions, each of which consists of 58 nuclei and hence 58 separate predictions. We used the following enhancers for predictions. *run_str3_7* is the enhancer of the *runt* gene of *D.melanogaster* that drives *runt* stripes 3 and 7, each of which is about 2 % E.L. anterior of the corresponding eve stripes. Biologically, the prediction is reasonably accurate except for the low level of expression in the anterior. *eve_S2E(yak)* is the stripe 2 enhancer of *Drosophila yakuba*, expressed in *D. melanogaster* embryos [29, 30]. *D. yakuba* is a species of *Drosophila* that is closely related to *D. melanogaster* but has altered sets of binding sites. The position of Stripe 2 in this case is accurate but the level is low. *eve_S2E(cyn)* and *eve_S37E(put)* are respectively the enhancers of *eve* stripe 2 from *Sepsis cynipsea* and the 3/7 enhancer from *Themira putris*, both of the Sepsidae family, driving expression in *D. melanogaster* embryos [15]. These predictions are quite accurate including the fact that they are expressed slightly posterior to those of the *D. melanogaster eve* gene. These enhancers have a DNA sequence completely diverged from those of *D. melanogaster* [15, 14]. These predictions demonstrate the generalization capabilities of the DNN implementation of the model.

4 Discussion

We discuss the the implication of the results presented here to both the biology and Deep Learning communities in turn. With respect to Deep Learning, our model constitutes an example of a fully interpretable DNN that is not merely biologically plausible but biologically validated [23]. It is our hope that this example will provide insights into the interpretabilities of DNNs in general, a problem that has received wide attention in the community [12, 8, 7, 26, 50, 31, 51].

More generally, neurobiology, together with the physics of spin-glasses, provided the initial inspiration for the mathematical structure of neural nets in general, and DNNs in particular were inspired by the structure of visual processing in the cerebral cortex. Perceptron based DNNs have an additive structure that arose from the additivity of voltages in neurons and spin glasses [33, 18]. In contrast, the mathematical structure of thermodynamic transcription models comes from multiplicative terms arising from the law of mass action, while the layered structure comes from the complex set of regulatory mechanisms that act in metazoan transcription. Perhaps the structure of the equations used here will suggest new applications and architectures for DNNs.

With respect to biology, the proof of concept presented here is much simpler to code than the original model, amounting to about 600 lines of Keras compared to about 10,000 lines of C++. Implementation in Keras also provides a numerical advantage by permitting the use of back propagation (BP) and stochastic gradient descent for optimization without the need to hand code partial derivatives. Although some issues connected to the speed of Keras implementation remain, these results suggest that models of this type could be scaled up to much larger datasets. Until recently such datasets, which must contain not only information about sequence but also data concerning the concentration of TFs, were scarce. Such datasets, on a genomic scale, have begun to be available for flies and mammals, including humans [3, 28, 46, 36, 49]. In these datasets, the functional role of TFs is typically unknown. In a study which used our model in mouse [6], all possible combinations of TF functional roles were considered, albeit with a considerable increase in computational expense. The data used in this study was not at a genomic scale, and an exhaustive search of TF functional roles at a genomic scale would be prohibitively expensive in computation. However, the introduction of reinforcement learning techniques may provide an avenue for surmounting this computational barrier. If the techniques presented here provide a way forward to precise characterization of gene regulation in metazoan organisms, it would be one further example of how investigation in the fruit fly *Drosophila*, with its special properties, can have global implications for biology.

Acknowledgement

This work was supported by grant R01 OD10936 from the US NIH

References

[1] Abadi, Martín, Agarwal, Ashish, Barham, Paul, Brevdo, Eugene, Chen, Zhifeng, Citro, Craig, Corrado, Greg S., Davis, Andy, Dean, Jeffrey, Devin, Matthieu, Ghemawat, Sanjay, Goodfellow, Ian, Harp, Andrew, Irving, Geoffrey, Isard, Michael, Jia, Yangqing, Jozefowicz, Rafal, Kaiser, Lukasz, Kudlur, Manjunath, Levenberg, Josh, Mané, Dan, Monga, Rajat, Moore, Sherry, Murray,

Derek, Olah, Chris, Schuster, Mike, Shlens, Jonathon, Steiner, Benoit, Sutskever, Ilya, Talwar, Kunal, Tucker, Paul, Vanhoucke, Vincent, Vasudevan, Vijay, Viégas, Fernanda, Vinyals, Oriol, Warden, Pete, Wattenberg, Martin, Wicke, Martin, Yu, Yuan, & Zheng, Xiaoqiang. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

[2] Alipanahi, Babak, Delong, Andrew, Weirauch, Matthew T., & Frey, Brendan J. 2015. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, **33**, 831–838.

[3] Arnold, C. D., Gerlach, D., Stelzer, C., Boryn, L. M., Rath, M., & Stark, A. 2013. Genome-wide quantitative enhancer activity maps identified by STARR-seq. *Science*, 1074–1077. PMID:23328393 doi:10.1126/science.1232542.

[4] Barr, K. A., & Reinitz, J. 2017. A sequence level model of an intact locus predicts the location and function of nonadditive enhancers. *PLoS One*, **12**, e0180861. doi:10.1371/journal.pone.0180861. PMCID:PMC5513433.

[5] Barr, K. A., Martinez, C., Moran, J. R., Kim, A. R., Ramos, A. F., & Reinitz, J. 2017. Synthetic enhancer design by *in silico* compensatory evolution reveals flexibility and constraint in *cis*-regulation. *BMC Systems Biology*, **11**, 116. PMID:29187214 PMCID:PMC5708098 doi:10.1186/s12918-017-0485-2.

[6] Bertolino, E., Reinitz, J., & Manu. 2016. The analysis of novel distal Cebpa enhancers and silencers using a transcriptional model reveals the complex regulatory logic of hematopoietic lineage specification. *Developmental Biology*, **413**, 128–144. doi:10.1016/j.ydbio.2016.02.030. PMCID:PMC4878123.

[7] Boger, Zvi, & Guterman, Hugo. 1997. Knowledge extraction from artificial neural network models. *Pages 3030–3035 of: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 4. IEEE.

[8] Castelvecchi, Davide. 2016. Can we open the black box of AI? *Nature News*, **538**(7623), 20.

[9] Celesti, F., Celesti, A., Carnevale, L., Galletta, A., Campo, S., Romano, A., Bramanti, P., & Villari, M. 2017. Big data analytics in genomics: The point on Deep Learning solutions. *Pages 306–309 of: 2017 IEEE Symposium on Computers and Communications (ISCC)*.

[10] Chollet, François, *et al.* 2015. *Keras*. `https://keras.io`.

[11] Cuperus, Josh T, Groves, Benjamin, Kuchina, Anna, Rosenberg, Alexander B, Jojic, Nebojsa, Fields, Stanley, & Seelig, Georg. 2017. Deep learning of the regulatory grammar of yeast 5 untranslated regions from 500,000 random sequences. *Genome research*, **27**(12), 2015–2024.

[12] Garson, G David. 1991. Interpreting neural-network connection weights. *AI expert*, **6**(4), 46–51.

[13] Greenside, Peyton, Shimko, Tyler, Fordyce, Polly, & Kundaje, Anshul. 2018. Discovering epistatic feature interactions from neural network models of regulatory DNA sequences. *Bioinformatics*, **34**(17), i629–i637.

[14] Hare, E. E., Peterson, B. K., & Eisen, M. B. 2008a. A careful look at binding site reorganization in the *even-skipped* enhancers of *Drosophila* and Sepsids. *PLoS Genetics*, **4**(11), e1000268. PMCID:PMC2582681.

[15] Hare, E. E., Peterson, B. K., Iyer, V. N., Meier, R., & Eisen, M. B. 2008b. Sepsid *even-skipped* Enhancers Are Functionally Conserved in *Drosopila* Despite Lack of Sequence Conservation. *PLoS Genetics*, **4**, e1000106. PMCID:PMC2430619.

[16] He, X., Samee, M. A. H., Blatti, C., & Sinha, S. 2010. Thermodynamics-Based Models of Transcriptional Regulation by Enhancers: The Roles of Synergistic Activation, Cooperative Binding and Short-Range Repression. *PLoS Computational Biology*, **6**, e1000935. PMCID:PMC2940721.

[17] Hewitt, G. F., Strunk, B., Margulies, C., Priputin, T., Wang, X. D., Amey, R., Pabst, B., Kosman, D., Reinitz, J., & Arnosti, D. N. 1999. Transcriptional repression by the *Drosophila* Giant protein: Cis element positioning provides an alternative means of interpreting an effector gradient. *Development*, **126**, 1201–1210.

[18] Hopfield, J. J. 1984. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences USA*, **81**, 3088–3092.

[19] Jaderberg, Max, Simonyan, Karen, Zisserman, Andrew, *et al.* 2015. Spatial transformer networks. *Pages 2017–2025 of: Advances in neural information processing systems*.

[20] Janssens, H., Kosman, D., Vanario-Alonso, C. E., Jaeger, J., Samsonova, M., & Reinitz, J. 2005. A high-throughput method for quantifying gene expression data from early *Drosophila* embryos. *Development, Genes and Evolution*, **215**, 374–381.

[21] Janssens, H., Hou, S., Jaeger, J., Kim, A. R., Myasnikova, E., Sharp, D., & Reinitz, J. 2006. Quantitative and predictive model of transcriptional control of the *Drosophila melanogaster even skipped* gene. *Nature Genetics*, **38**, 1159–1165.

[22] Kazemian, M., Blatti, C., Richards, A., McCutchan, M., Wakabayashi-Ito, N., Hammonds, A. S., Celniker, S. E., Kumar, S., Wolfe, S. A., Brodsky, M. H., & Sinha, S. 2010. Quantitative Analysis of the *Drosophila* Segmentation Regulatory Network Using Pattern Generating Potentials. *PLoS Biology*, **8**, e1000456. PMCID:PMC2923081.

[23] Kim, A. R., Martinez, C., Ionides, J., Ramos, A. F., Ludwig, M. Z., Ogawa, N., Sharp, D. H., & Reinitz, J. 2013. Rearrangements of 2.5 Kilobases of Noncoding DNA from the *Drosophila even-skipped* Locus Define Predictive Rules of Genomic *cis*-Regulatory Logic. *PLoS Genetics*, **9**, e1003243. PMCID:PMC3585115.

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

[25] Krizhevsky, Alex, Sutskever, Ilya, & Hinton, Geoffrey E. 2012. Imagenet classification with deep convolutional neural networks. *Pages 1097–1105 of: Advances in neural information processing systems*.

[26] Li, Yixuan, Yosinski, Jason, Clune, Jeff, Lipson, Hod, & Hopcroft, John E. 2015. Convergent Learning: Do different neural networks learn the same representations? *Pages 196–212 of: FE@ NIPS*.

[27] Libbrecht, Maxwell W., & Noble, William Stafford. 2015. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, **16**, 321–332.

[28] Liu, Yuwen, Yu, Shan, Dhiman, Vineet K, Brunetti, Tonya, Eckart, Heather, & White, Kevin P. 2017. Functional assessment of human enhancer activities using whole-genome STARR-sequencing. *Genome biology*, **18**(1), 219.

[29] Ludwig, M. Z., Bergman, C. M., Patel, N. H., & Kreitman, M. 2000. Evidence for stabilizing selection in a eukaryotic enhancer element. *Nature*, **403**, 564–567.

[30] Ludwig, M. Z., Palsson, A., Alekseeva, E., Bergman, C. M., Nathan, J., & Kreitman, M. 2005. Functional Evolution of a *cis*-Regulatory Module. *PLoS Biology*, **3**(4), e93. PMCID:PMC1064851.

[31] Maaten, Laurens van der, & Hinton, Geoffrey. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, **9**, 2579–2605.

[32] Martinez, Carlos, Kim, Ah-Ram, Rest, Joshua S., Ludwig, Michael, Kreitman, Martin, White, Kevin, & Reinitz, John. 2014. Ancestral resurrection of the *Drosophila* S2E enhancer reveals accessible evolutionary paths through compensatory change. *Molecular Biology and Evolution*, **31**, 903–916. PMCID:PMC3969564.

[33] McCulloch, Warren S, & Pitts, Walter. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, **5**(4), 115–133.

[34] Movva, Rajiv, Greenside, Peyton, Shrikumar, Avanti, & Kundaje, Anshul. 2018. Deciphering regulatory DNA sequences and noncoding genetic variants using neural network models of massively parallel reporter assays. *bioRxiv*, 393926.

[35] Nair, Surag, Kim, Daniel S, Perricone, Jacob, & Kundaje, Anshul. 2019. Integrating regulatory DNA sequence and gene expression to predict genome-wide chromatin accessibility across cellular contexts. *bioRxiv*, 605717.

[36] Patwardhan, Rupali P, Lee, Choli, Litvin, Oren, Young, David L, Pe'er, Dana, & Shendure, Jay. 2009. High-resolution analysis of DNA regulatory elements by synthetic saturation mutagenesis. *Nature biotechnology*, **27**(12), 1173.

[37] Pouladi, F., Salehinejad, H., & Gilani, A. M. 2015. Recurrent Neural Networks for Sequential Phenotype Prediction in Genomics. *Pages 225–230 of: 2015 International Conference on Developments of E-Systems Engineering (DeSE).*

[38] Reinitz, J., Hou, S., & Sharp, D. H. 2003. Transcriptional control in *Drosophila*. *ComPlexUs*, **1**, 54–64.

[39] Rui Xu, Wunsch, D.C., & Frank, R.L. 2007. Inference of Genetic Regulatory Networks with Recurrent Neural Network Models Using Particle Swarm Optimization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **4**, 681–692.

[40] Samee, M. A. H., & Sinha, S. 2014. Quantitative modeling of a gene's expression from its intergenic sequence. *PLoS Computational Biology*, **10**, 1–21.

[41] Sayal, R., Dresch, J. M., Pushel, I., Taylor, B. R., & Arnosti, D. 2016. Quantitative perturbation-based analysis of gene expression predicts enhancer activity in early *Drosophila* embryo. *eLife*, **5**, e08445. PMID:27152947 PMCID:PMC4859806 doi:10.7554/eLife.08445.

[42] Segal, E., Raveh-Sadka, T., Schroeder, M., Unnerstall, U., & Gaul, U. 2008. Predicting expression patterns from regulatory sequence in *Drosophila* segmentation. *Nature*, **451**, 535–540. PMID:18172436 doi:10.1038/nature06496.

[43] Shen, Jingxiang, Petkova, Mariela D., Liu, Feng, & Tang, Chao. 2018. Toward deciphering developmental patterning with deep neural network. *bioRxiv*, 374439.

[44] Small, S., Blair, A., & Levine, M. 1992. Regulation of *even-skipped* stripe 2 in the *Drosophila* embryo. *The EMBO Journal*, **11**, 4047–4057.

[45] Small, S., Blair, A., & Levine, M. 1996. Regulation of two pair-rule stripes by a single enhancer in the *Drosophila* embryo. *Developmental Biology*, **175**, 314–324.

[46] Smith, R. P., Taher, L., Patwardhan, R. P., Kim, M. J., Inoue, F., Shendure, J., Ovcharenko, I., & Ahituv, N. 2013. Massively parallel decoding of mammalian regulatory sequences supports a flexible organizational model. *Nature Genetics*, **45**, 1021–1028. PMID:23892608 PMCID:PMC3775494 doi:10.1038/ng.2713.

[47] Stanojevic, D., Small, S., & Levine, M. 1991. Regulation of a segmentation stripe by overlapping activators and repressors in the *Drosophila* embryo. *Science*, **254**, 1385–1387.

[48] Surkova, S., Kosman, D., Kozlov, K., Manu, Myasnikova, E., Samsonova, A., Spirov, A., Vanario-Alonso, C. E., Samsonova, M., & Reinitz, J. 2008. Characterization of the *Drosophila* Segment Determination Morphome. *Developmental Biology*, **313**(2), 844–862. PMCID:PMC2254320.

[49] Ulirsch, Jacob C, Nandakumar, Satish K, Wang, Li, Giani, Felix C, Zhang, Xiaolan, Rogov, Peter, Melnikov, Alexandre, McDonel, Patrick, Do, Ron, Mikkelsen, Tarjei S, *et al.* 2016. Systematic functional dissection of common genetic variation affecting red blood cell traits. *Cell*, **165**(6), 1530–1545.

[50] XuK, BaJ, KirosR, CourvilleA, *et al.* 2015. Show, attendandtell: Neural image caption generation with visual attention. *Proceedings of the International Conference on Machine Learning. Lille, France*, **2048**, 2057.

[51] Zeiler, Matthew D, & Fergus, Rob. 2014. Visualizing and understanding convolutional networks. *Pages 818–833 of: European conference on computer vision*. Springer.