bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Learning distance-dependent motif interactions: an interpretable CNN model of genomic events

Thomas P. Quinn[1*], Dang Nguyen[1], Phuoc Nguyen[1], Sunil Gupta[1], and Svetha Venkatesh[1]

[1]Applied Artificial Intelligence Institute (A2I2), Deakin University, Geelong, Australia
* *contacttomquinn@gmail.com*

Abstract

In most biological studies, prediction is used primarily to validate the model; the real quest is to understand the underlying phenomenon. Therefore, interpretable deep models for biological studies are required. Here, we propose **HyperXPair** (the **Hyper**-parameter e**X**plainable Motif **Pair** framework), a new architecture that learns biological motifs and their distance-dependent context through explicitly interpretable parameters that are immediately understood by a biologist. This makes **HyperXPair** more than a decision- support tool; it is also a hypothesis-generating tool designed to advance knowledge in the field. We demonstrate the utility of our model by learning distance-dependent motif interactions for two biological problems: transcription initiation and RNA splicing.

1 Introduction

Living cells store information in large repeating chains of molecules called polymers. Example polymers include DNA, RNA, and proteins, each of which can be described as a sequence composed from a finite set of bases. The function of any molecule ultimately depends on the chemical properties of its atomic constituents. Yet, it is possible to make accurate predictions directly from 1D sequence representations by modelling the presence of biologically meaningful sub-sequences called *motifs* that cooperate together in a distance-dependent fashion. Identifying these motifs, and the dependencies between them, offers a key to understanding the mechanistic basis of Protein-DNA and Protein-RNA interactions.

We hypothesize that two key factors determine whether a motif will execute a genomic event. The first is *motif identity*: certain sub-sequences within the DNA are necessary in order to execute an event. They are short (1s-10s of bases long) and fuzzy. The second is *motif context*: motifs are necessary, but not always sufficient. Events often require multiple motifs separated along the DNA by a specific number of positions. This mechanism appears throughout genome biology. For example, consider transcription initiation. The Shine-Dalgarno sequence needs to be ~5-13 bases upstream from the Kozak consensus sequence to initiate transcription [Ma et al., 2002]. Another example is RNA splicing, a biological process in which sequences have sub-sequences removed and concatenated [Ule and Blencowe, 2019] (e.g., to convert the word "machine" to "m...ine" to "mine"). Here, fuzzy motifs also cooperate in a distance-dependent fashion, potentially involving very long-range interactions [Wong et al., 2016].

Interest has gathered recently around how machine learning, especially deep convolutional neural networks (CNNs), could be used to predict genomic events like RNA splicing [Ching et al., 2018, Jaganathan et al., 2019, Wang et al., 2019, Albaradei et al., 2020]. *Although deep CNNs achieve good performance, they do not learn an explicit model of motif pairs or their distance-dependent interactions. In most biological studies, prediction itself is used primarily to validate the model; the real quest is to understand the underlying phenomenon. Therefore, interpretable deep models for biological studies are required.*

Transcription initiation and RNA splicing both illustrate the challenges associated with learning an interpretable model of genomic events:

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

- **Motif degeneration:** Motifs act as "docking signals" for binding proteins. Several subsequences can allow the same protein to dock. As such, biological motifs are ill-defined constructs, and usually have fuzzy definitions. For example, GGAG and AGGA are both valid Shine-Dalgarno sequences [Ma et al., 2002]. Thus, it is common to define motifs according to a "consensus", for example by using the observed frequency over all DNA instances. An ideal model would learn to represent this consensus.

- **Distance-dependence:** A motif may be necessary, but not sufficient. Whether a motif is sufficient depends on its context, such as its distance from another motif. In other words, motif pairs act synergistically, not additively [Ke and Chasin, 2010], as a function of their proximity. Like motifs, distance-dependence has a fuzzy definition. In the case of transcription initiation, the observed frequencies of inter-motif distances follow a Gaussian distribution [Ma et al., 2002]. As such, one could also think about distance-dependence as a kind of "consensus". An ideal model would learn to represent this consensus.

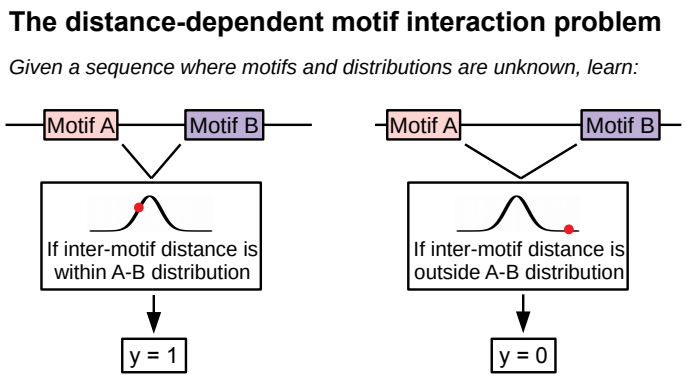

Figure 1: This figure provides a visual summary of the distance-dependent motif interaction problem that we intend to solve with an explicitly interpretable neural network model.

Figure 1 provides a visual summary of the *distance-dependent motif interaction problem*, which requires us to learn both motifs and inter-motif distance distributions simultaneously. Our solution to this problem takes inspiration from *self-explanation*, in which interpretability is built-in architecturally [Alvarez-Melis and Jaakkola, 2018]. Here, we propose **HyperXPair** (the **Hyper**-parameter e**X**plainable Motif **Pair** framework), a new architecture that models motifs and their distance-dependence through simple and understandable parameters. Its two modules solve the challenges above by enforcing strong biological priors about how motifs interact, yielding parameters that explicitly represent the aspects we want to interpret:

- **Motif identity module:** This is simply a 1-layer CNN without any non-linear activation or max-pooling, where the kernel parameters represent the consensus motifs. This module solves the challenge of motif degeneration because the CNN kernel can capture several overlapping motifs. In the case that motif degeneration is too extreme, separate filters could capture alternate versions of the motif.

- **Motif co-occurrence module:** This is another 1-layer CNN that uses a custom kernel. The kernel parameters depend on *learnable hyper-parameters that represent inter-motif distances* as a "consensus" Gaussian distribution, which we can efficiently search for using Bayesian Optimization [Shahriari et al., 2016]. This module solves the challenge of distance-dependence because the CNN kernel can score motif co-occurrences as a function of the distance between them, giving more weight to optimally spaced motif pairs.

Our model offers the first explicitly interpretable solution to the distance-dependent motif interaction problem, making it applicable to a wide range of important problems related to the post-transcriptional control of RNA. Unlike other models, **HyperXPair** learns to relate biological sequences to genomic events through parameters that are immediately understood by a biologist. This makes **HyperXPair** more than a decision-support tool; it is also a hypothesis-generating tool designed to advance knowledge in the field. We demonstrate the utility of our model by learning

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

the distance-dependent motif interactions for two biological problems: transcription initiation and RNA splicing. In the latter case, we show how an interpretation of the model's parameters can be comined with relevant literature to generate hypotheses that extend existing knowledge with a potentially novel mechanisms.

2 Related work

Interpretable AI: Researchers have proposed new frameworks to make "black-box" neural networks more interpretable. There are two general approaches: (1) to train an arbitrary neural network and infer predictions secondarily via post-hoc analysis, or (2) to train a neural network whose parameters can be interpreted directly. Among post-hoc methods, Zhou and Troyanskaya [2015] have used a perturbation method called *in silico mutagenesis* to identify motifs by "mutating" DNA bases and measuring the change in prediction. Intuitively, important bases will cause bigger prediction changes when perturbed. Koo et al. [2018] used a similar procedure to understand how motif number and spacing impacts prediction. Meanwhile, a back-propagation method called DeepLIFT [Shrikumar et al., 2019] identifies motifs by comparing the gradient for a sample-of-interest against a reference [Wang et al., 2019]. In contrast, our model takes inspiration from *self-explanation*, in which interpretability is built-in architecturally [Alvarez-Melis and Jaakkola, 2018]. In **HyperXPair**, the parameters explicitly represent the aspects we want to interpret.

Perceptrons for motif discovery A landmark paper from 1982 used a perceptron algorithm to classify whether DNA sequences will make a valid gene product. Their model learned a weights matrix, called a *position weights matrix* (PWM), such that its product with the embedded sequence would classify transcription initiation (a binary outcome) [Stormo et al., 1982]. During training, valid genes were aligned so that the PWM would have some semblance of translation invariance. During testing, the PWM was deployed at every position in the sequence, and sites were sorted based on the total activation. Although their model is accurate, the learned PWM cannot explain the Shine-Dalgarno sequence, or its distance-dependence, because all contexts are super-imposed onto a single weights matrix. We build upon this work by using CNNs to learn PWMs.

CNNs for biological sequences: Encoding the DNA as a matrix makes it image-like. If we conceptualize a motif as a kind of visual stimulus, then we can understand why CNNs would be a natural choice for learning translation-invariant motifs [Fukushima, 1980, Waibel et al., 1989]. In practice, convolutional filters from the lowest layers learn motif fragments which get assembled in deeper layers to capture higher-order interactions and distance-dependence. Deep CNNs have enabled many sequence prediction tasks, from splice prediction [Jaganathan et al., 2019] to drug-target affinity prediction [Nguyen et al., 2020a]. One potential advantage of deep CNNs is that they do not require prior knowledge about how an event, such as splicing, actually happens [Bao et al., 2019]. We include shallow and deep CNNs in our baseline.

RNNs for biological sequences: The 1-dimensional spatial arrangement of a biological sequence lends itself to analysis by recurrent neural networks (RNNs), which can detect sequential patterns with some tolerance to changes in the position and identity of the motifs [Hawkins and Bodén, 2005]. Long short-term memory (LSTM) is most popular today, owing to its ability to learn patterns with long time lags [Hochreiter and Schmidhuber, 1997]. LSTM has been applied to biological sequences, often in combination with CNNs. For example, Hassanzadeh and Wang [2016] and Pan et al. [2018] both propose a two-tier model where an initial CNN layer learns to represent DNA motifs, while a second LSTM layer learns long-range interactions between motifs. We include LSTMs in our baseline.

CNNs for motif discovery: Biological sequences are different than images. For images, lower-layer features like edges may not mean much to the investigator; for biological sequences, lower-layer features are the motifs themselves, and thus should be preserved accurately [Koo and Eddy, 2019]. Unfortunately, deep CNNs can cause the lower layers to fragment into partial motifs that misrepresent the underlying biological reality [Koo and Eddy, 2019]. Several studies have explored how architectural variants could encourage the lower layers to learn coherent motifs, such as the use of exponential activation [Koo and Ploenzke, 2019], Gaussian noise injection [Koo et al., 2019], or kernel regularization and constraints [Ploenzke and Irizarry, 2018]. However, even if a network can learn complete motifs, the interactions between them may be obscured in the depths of the CNN [Koo and Eddy, 2019]. Our architecture instead learns the interactions explicitly.

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

3 Model Overview

Figure 2 and **Supplemental Figure 1** illustrate the **HyperXPair** architecture.

HyperXPair works in 3 stages. First, a CNN learns motif identities from a 1-hot embedding of DNA, producing a **motif activity map** that describes the presence of each motif along the sequence. Second, another CNN uses a motif co-occurrence kernel to compute co-occurrence scores for each motif pair along the sequence. Third, a global max pool layer selects the largest co-occurrence score for each pair, which are then used to predict the outcome via a simple regression.

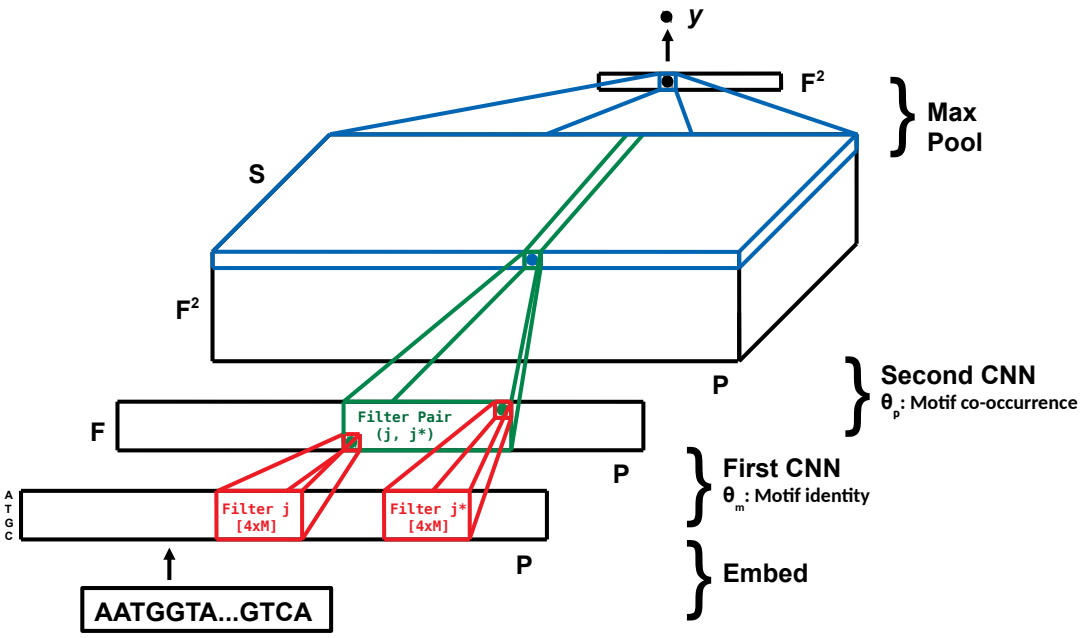

Figure 2: The **HyperXPair** architecture has 3 stages: (1) a first CNN learns motifs from an embedded sequence, (2) a second CNN computes distance-dependent co-occurrence scores, and (3) a global max pool layer predicts the output.

The 3 stages correspond to 3 sets of parameters that describe: (a) the consensus identity of the motifs, (b) the consensus distance-dependence for each motif pair, and (c) the contribution of each motif pair to the final outcome.

- **Motif identity parameters:** These are the filter weights from the first CNN layer. They are interpretable because they define a motif. Large positive weights imply that a DNA base is often present. Large negative weights imply that a DNA base is rarely present.
- **Motif co-occurrence parameters:** These are the filter weights from the second CNN layer (as generated from learnable hyper-parameters). These are interpretable because they define a **receptive field** that describes how the importance of motif co-occurrence relates to the distance between them.
- **Outgoing parameters:** These are the weights of a simple regression that predicts the outcome from the max co-occurrence scores. These are interpretable because large model coefficients imply that a pair is important for the outcome.

3.1 Stage 1: First CNN Discovers Consensus Motifs

Before we can make predictions based on the distances between motifs, we first need to learn the motifs themselves. For this, we learn a function that converts the input $\mathbf{x}$ into a **motif activity map** describing the presence of each motif along the sequence:

$$\mathbf{a} = f_m(\mathbf{x}; \theta_m) \tag{1}$$

where f_m is a CNN. The parameters θ_m contain F filters, each being a consensus motif. These filters are sized M-by-4, **where M is the maximum length of the consensus motif**.

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

The input $\mathbf{x}$ contains N genomic sequences, having up to P bases with 4 possible states. We 1-hot encode the 4 states–adenine, cytosine, guanine, and thymine–then add P bases of "padding" to both sides of the sequence. This results in a tensor of size $[N \times 3P \times 4]$. The padding allows the model to capture co-occurrences at the ends of the sequences.

3.2 Stage 2: Second CNN Applies Receptive Field

Now that we have a **motif activity map**, we can score motif co-occurrence based on the distance between the pairs. For F motifs, there are F^2 motif pairs. Our goal is to compute a distance-dependent co-occurrence score for each pair j and j^*,. The score should be large when the motifs are present *and* their distance falls within an assumed inter-motif distance distribution. The score should be small when a motif is absent *or* their distance falls outside the distribution.

Section 4 describes how we efficiently search for the parameters of the distance distribution using Bayesian Optimization. For now, let us assume we already know the mean $\mu^{(j,j^*)}$ and standard deviation $\sigma^{(j,j^*)}$ that define the Gaussian inter-motif distance distribution. This distribution forms the basis of a **receptive field** that assigns weight to motif j^* as a function of its distance from motif j. The receptive field scores each motif gap $\{0...S-1\}$, **where S is one more than the maximum motif gap considered**.

Figure 3 shows how we can encode the **receptive field** as a custom CNN kernel that scores co-occurrence as a function of distance. The associated CNN kernel will have the size $[F \times S \times S]$, where the first slice of the kernel captures a motif gap of 0 (i.e., when motif j and $j*$ overlap), the second slice of the kernel captures a motif gap of 1 (i.e., when motif j and $j*$ are separated by one position), and so on. One could think of the CNN kernel as a kind of "2-hot" encoding of the receptive field. The kernel $\theta_p^{(j,j^*)}$ thus follows deterministically from the hyper-parameters:

$$\theta_p^{(j,j^*)} = f_e(\mathcal{N}(\mu^{(j,j^*)}, \sigma^{(j,j^*)}), S) \tag{2}$$

where f_e is the encoding. We do not apply the receptive field directly because it would weigh multiples of one filter the same as a filter pair. Empirically, this approach does not work well.

Note that S gives $\mu^{(j,j^*)}$ contextual meaning. When $\mu^{(j,j^*)} = 0$, a motif gap of $S/2$ receives the highest weight. When $\mu^{(j,j^*)} \approx -3$, a motif gap of 0 receives the highest weight. When $\mu^{(j,j^*)} \approx 3$, a motif gap of $S-1$ receives the highest weight. Meanwhile, $\sigma^{(j,j^*)}$ determines how the weights decrease as the observed motif gap gets further from the optimal motif gap.

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

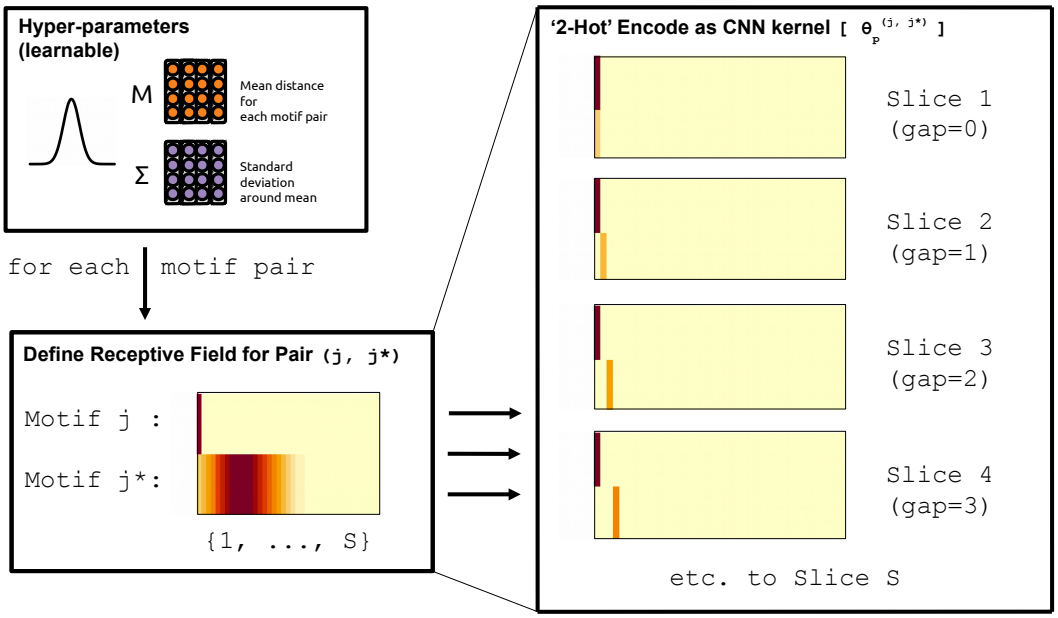

Figure 3: This figure shows how we can encode the hyper-parameters $\mu^{(j,j^*)}$ and $\sigma^{(j,j^*)}$ as a **receptive field** and associated CNN kernel. The receptive field assigns weight to motif j^* as a function of its distance from motif j. When a motif pair falls within the red bands, the pair will receive a high co-occurrence score. When a motif pair falls within the yellow-orange bands, the pair will receive a lower score. We "2-hot" encode the receptive field as a CNN kernel having the size $[F \times S \times S]$, where the first slice of the kernel captures a motif gap of 0, the second slice of the kernel captures a motif gap of 1, and so on. Note this figure shows a kernel for a single motif pair. The encoding is repeated F^2 times to cover all filter pairs.

We can now apply the kernel directly:

$$\mathbf{d}^{(j,j^*)} = f_p(\mathbf{a}; \theta_p^{(j,j^*)}) \tag{3}$$

where f_p is a CNN. The resultant tensor $\mathbf{d}^{(j,j^*)}$ represents the distance-dependent co-occurrence score for each motif gap in $\{1...S-1\}$, at each position along the DNA sequence (see Figure 2).

Note that because $\theta_p^{(j,j^*)}$ is applied to the **motif activity map**, large departures from the motif consensus will also penalize the co-occurrence score. As such, a high co-occurrence score implies that both motifs are present *and* their distance falls within the assumed distance distribution, solving the distance-dependent motif interaction problem presented in Figure 1.

3.3 Stage 3: Global Max Pool Regressed to Output

We calculate the max co-occurrence score for each motif pair as the global maximum of $\mathbf{d}^{(j,j^*)}$:

$$\gamma^{(j,j^*)} = \max(\mathbf{d}^{(j,j^*)}) \tag{4}$$

This results in a single number for each motif pair. When repeated for all F^2 motif pairs, we get a tensor of size $[N \times F \times F]$, or, equivalently, $[N \times F^2]$. Thus,

$$\Gamma = f_c(\gamma^{(1,1)}, \ldots, \gamma^{(F,F)}) \tag{5}$$

where f_c is a simple concatenation layer. From this, we can predict the final outcome:

$$\hat{y} = f_o(\Gamma; \theta_o) \tag{6}$$

We choose f_o to be a generalized linear model where $\theta_o = \boldsymbol{\beta}$, giving us:

$$\hat{y} = \Phi\Big(\sum_{j=1}^{F}\sum_{j^*=1}^{F} \beta^{(j,j^*)}\gamma^{(j,j^*)} + \beta_0\Big) \tag{7}$$

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

where Φ transforms the model output according to the nature of observed outcome. In our case, the outcome is binary and so Φ is the sigmoid transform.

4 Model Learning

Learning occurs in two loops. In the inner-loop, we use stochastic gradient descent to search for the motifs whose assumed distance-dependent interactions minimize the training loss. In the outer-loop, we use Bayesian Optimization (**BO**) to search for the hyper-parameters $\{M, \Sigma\}$ that give the best neural network, where the network weights are learned anew at each iteration of BO. Algorithm 1 describes our implementation.

Input: $\mathcal{D}_{tr} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$: training set, $\mathcal{D}_{va}$: validation set, T: the number of iterations
Output: **HyperXPair** model

```
begin
    for t = 1, 2, ..., T do
        Fit a Gaussian process using {(M_i, Σ_i, s_i), acc_i}_{i=1}^t,
        where M_i, Σ_i are means and standard deviations of the distance distributions, s_i is
          initialization seed, and acc_i is AUC of HyperXPair on D_va;
        Suggest next point (M_{t+1}, Σ_{t+1}, s_{t+1}) by maximizing UCB acquisition function;
        Train HyperXPair model:
        Initialize HyperXPair model with seed s_{t+1};
        for number of training epochs do
            Embed each sequence x_i into a 1-hot encoding;
            Compute motif activity map a = f_m(x_i; θ_m) (Eq. (1));
            Construct F^2 motif pairs from a;
            foreach j, j* pair in F^2 do
                Select μ_{t+1}^{(j,j*)} ∈ M_{t+1} and σ_{t+1}^{(j,j*)} ∈ Σ_{t+1};
                Compute co-occurrence score d^{(j,j*)} = f_p(a; θ_p^{(j,j*)}) (Eq. (3)),
                where θ_p^{(j,j*)} encodes the receptive field N(μ_{t+1}^{(j,j*)}, σ_{t+1}^{(j,j*)});
                Apply global max pool γ^{(j,j*)} = max(d^{(j,j*)}) (Eq. (4));
            end
            Concatenate scores Γ = [γ^{(1,1)}, γ^{(1,2)}, ..., γ^{(F,F)}] (Eq. (5));
            Compute prediction ŷ_i = Φ(Σ_{j=1}^F Σ_{j*=1}^F β^{(j,j*)} γ^{(j,j*)} + β_0) (Eq. (7));
            Update θ_m by minimizing L = H(ŷ_i, y_i),
            where H is binary cross-entropy loss;
        end
        Compute accuracy acc_{t+1} of HyperXPair on D_va;
        Augment new observed point {(M_{t+1}, Σ_{t+1}, s_{t+1}), acc_{t+1}} to {(M_i, Σ_i, s_i), acc_i}_{i=1}^t;
    end
end
```

Algorithm 1: The proposed **HyperXPair** algorithm.

4.1 Inner-loop: motif discovery

For a given set of hyper-parameters $M = [\mu^{(1,1)}, \ldots, \mu^{(F,F)}]$ and $\Sigma = [\sigma^{(1,1)}, \ldots, \sigma^{(F,F)}]$, we can train our neural network end-to-end. This will learn the motifs that minimize the loss

$$\mathcal{L} = \mathcal{L}_y + \lambda\Big(\sum_{j=1}^{F}\sum_{j^*=1}^{F} |\beta^{(j,j^*)}|\Big) \tag{8}$$

where $\mathcal{L}_y$ is binary cross-entropy, and λ penalizes the magnitude of the weights in the final layer. This regularization allows the analyst to over-specify the number of filter pairs needed for the model, the optimal number of which is usually unknown.

During early experiments, we noticed two drawbacks with the vanilla architecture. First, it quickly overfitted to the training data by learning overly elaborate consensus motifs that are not

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

found in the validation set. We address this by adding an aggressive dropout layer of 30% to the embedded sequence input. Second, it would sometimes fail to find a global optimum. We address this by initializing the model across several random seeds, and choosing the best performer.

4.2 Outer-loop: inter-motif distance discovery

Our model depends on 3 important hyper-parameters: the means and standard deviations of the distance distribution, and the random seed used for initialization. To find the optimal values for these hyper-parameters, we use Bayesian Optimization (BO) [Snoek et al., 2012, Nguyen et al., 2020b], as implemented in the rBayesianOptimization package. Here, we treat the classification process of our model as a black-box function, where the inputs are the means, standard deviations, and random seed, and the output is validation set AUC. We then use a Gaussian process (GP) [Rasmussen, 2003] to model the black-box function, using an Upper Bound Confidence (UCB) acquisition function [Srinivas et al., 2012, Snoek et al., 2012] to suggest the next point.

To speed up optimization, we only consider a half-matrix of $\mu^{(j,j^*)}$ where $j < j^*$, and further assume that all $\sigma^{(j,j^*)}$ have the same value. This greatly reduces the number of hyper-parameters we need to learn. The decision to treat random initialization as a hyper-parameter is based on early experiments where some random seeds led to better performances regardless of $\{M, \Sigma\}$.

5 Experiments

5.1 Data and baselines

5.1.1 Synthetic data

We simulated data based on the Stormo et al. [1982] transcription initiation case study. Using a custom script, we simulated DNA sequences belonging to 2 classes. In the first class, the DNA sequence contained the GGAGG motif ahead of the ATG motif by ~5-13 positions (with the actual distance sampled from a Gaussian distribution where $\mu = 8$ and $\sigma = 2$, as approximated from [Ma et al., 2002]). In the second class, the DNA sequence either (a) contained the GGAGG motif *after* the ATG motif, or (b) did not contain the ATG motif.

We simulated $N = [128, 256, 512, 1024, 2048, 4096]$ total samples so that we could test our model over a range of sample sizes. We repeated the simulation procedure 5 times, withholding 33% of samples as a test set (yielding 30 test sets overall). Test sets were used only once to evaluate the performance of the final model. Note that because of nested validation, $N = 1024$ would imply that **HyperXPair** is trained on just 439 samples, with 109 withheld to monitor neural network validation loss, 137 for BO validation, and the rest for testing. The validation sets are sampled randomly from the training set at each step in the BO search.

5.1.2 Real data

Albaradei et al. [2020] benchmarked a deep CNN model for splice prediction, and made their data publicly available. We downloaded the *Drosophila melanogaster* data[1], chosen because it contained the fewest samples, and because fruit fly splicing is well-studied [Mount et al., 1992].

The data include two binary classification problems. The first is to predict the presence of a "donor" splice site (i.e., spliced content is rightward of retained content). The second is to predict the presence of an "acceptor" splice site (i.e., spliced content is leftward of retained content). Each sequence contains 602 bases, centered on the splice site itself. We performed 5 separate random splits of the data, withholding 33% of the data each time (yielding 10 test sets overall). Again, test sets were used only once to evaluate the performance of the final model.

5.1.3 Baselines

We compare **HyperXPair** against several baselines which include:

- **Natural Language Processing (NLP) type methods**: By treating a DNA sequence as a document, and individual bases as words, we can represent the DNA sequence via conventional NLP methods. We consider two well-known methods: bag-of-words (**BOW**) and

[1] https://github.com/SomayahAlbaradei/Splice_Deep

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

term frequency-inverse document frequency (**TF-IDF**) [Arora et al., 2018]. After constructing feature vectors of the DNA sequences, we train a linear SVM to classify the outcome, tuning the hyper-parameter $C \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ by a validation set. These baselines are chosen because they have good performance for standard NLP tasks.

- **Sequence embedding methods**: We also compare with **Sqn2Vec**, a neural embedding method for learning sequence representations [Nguyen et al., 2018]. This method learns a sequence embedding by predicting frequent sub-sequences (e.g., motifs) that belong to a sequence. Sqn2Vec has two models: Sqn2Vec-SEP and Sqn2Vec-SIM. These baselines are chosen because they perform well for sequential data that have a small vocabulary, like DNA.

- **2-filter CNNs:** We implement a shallow CNN with a 1D convolutional layer containing 2 filters of size 4x8, followed by a global max pooling layer connected to the output (with or without an intermediate 64-node dense hidden layer). This CNN uses the same training specifications as **HyperXPair**, and is chosen because it allows us to isolate the contribution of the motif co-occurrence module.

- **Shallow CNNs:** We also implement a wider shallow CNN having a 1D convolutional layer with ReLu activation. As above, a global max pool layer is used to predict the output. For these CNNs, F is set to $\{15, 20, 25, 30, 35, 40\}$ for data set size N equal to 128, 256, 512, 1024, 2048, and ≥ 4096. See the "Deep CNNs" baseline below for training specifications. This baseline is chosen because it resembles the more interpretable models used for genome biology prediction tasks (c.f., Koo and Eddy [2019]).

- **Deep CNNs:** The deep CNN begins the same as the shallow CNN, except that the global max pool layer is replaced by a $[3 \times 1]$ max pool layer. It then has two more sets of convolutional and max pool layers (using $2F$ and $4F$ filters, respectively), followed by a global max pool layer. It also has a 100-node dense hidden layer before the final output layer. Here, F is set to $\{5, 10, 15, 20, 25, 30\}$ for data set size N equal to 128, 256, 512, 1024, 2048, and ≥ 4096. All hidden layers have ReLu activations and the output layer has sigmoid activation. For training, the cross-entropy loss and ADAM optimizer are used with a learning rate of 0.001 and a batch size of 16, with 100 epochs. This baseline is chosen because it resembles the high-performing models used for genome biology prediction tasks.

- **LSTMs:** We implement an LSTM with the following design: the dimension of symbol embedding is 128, the number of LSTM hidden units is 100, and the drop-out rate after each layer is 0.2. For training, the ADAM optimizer is used with a learning rate of 0.001 and batch size of 64, with 50 epochs (following suggestions from Keras[2]). This baseline is chosen because it resembles the high-performing models used for genome biology prediction tasks.

5.1.4 HyperXPair training

Unless otherwise noted, we used $F \geq 2$ total filters (chosen to simplify interpretation and improve run-time), $M = 8$ motif length (chosen because we are interested in short motifs), and $S = 40$ motif gap (chosen because we are interested in proximal motifs). If $F > 2$, $\lambda = 0.01$, else $\lambda = 0.001$. For training, the ADAM optimizer is used with a learning rate of 0.001 and a batch size of $\min(N/64, 320)$ (this makes each epoch have a similar number of batches). For hyper-parameter tuning, we run 80 iterations of BO after a random initialization of 8 trials (based on a "rule-of-thumb" of 2 initializations and 20 iterations per variable being optimized for a single pair). In addition to tuning all M (bounded $[-3, 3]$) and one Σ (bounded $[.1, 3]$), we tune the random seed over 1...10 (if $F = 2$) or 1...15 (if $F > 2$). Training occurs for 50 epochs during hyper-parameter tuning, or 150 epochs otherwise (chosen based on early experiments with toy data). In all cases, early stopping occurs if inner-fold validation loss does not improve for 15 epochs.

5.1.5 HyperXPair weights visualization

We can visualize filter weights via a conventional "seqLogo" plot by applying a row-wise softmax transform that scales the per-position kernel weights to a unit-sum; a subsequent perturbation of the data, having the form $f(z) = z^{\alpha} / \sum z^{\alpha}$, acts like a "gain" knob to magnify the signal-to-noise ratio of the associated information content.

[2] https://keras.io

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

5.2 Study 1: HyperXPair has interpretable parameters

We are interested in a problem setting where neither the consensus motifs nor the consensus inter-motif distances are known, but where we have available thousands of raw sequences and an empirically measured outcome. The simulated data allow us to test whether our model can correctly learn the relevant motifs and inter-motif distance distributions. Learning more accurate motifs and distance distributions should yield more accurate classifiers. Figure 4 shows excellent performance for **HyperXPair** as compared with several baselines.

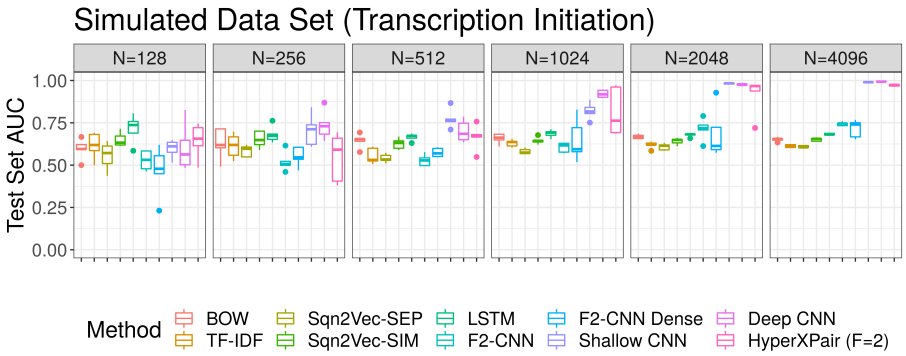

Figure 4: This figure shows the test set AUC (y-axis) for several methods (x-axis) as measured on the simulated transcription initiation data (facet). Acronyms: bag-of-words (BOW), term frequency-inverse document frequency (TF-IDF), 2-filter CNN (F2-CNN), 2-filter CNN with intermediate hidden layer (F2-CNN Dense).

We can also examine the motifs and distance distributions directly. The top panels of Figure 5 show the learned filter weights and corresponding seqLogo plots. The bottom panel shows the learned inter-motif distance distribution. Clearly, we see that **HyperXPair** has learned that having the motif GAGG ~5-13 bases ahead of ATG will predict transcription initiation, an insight provided by the model through explicitly interpretable parameters. **Supplemental Figures 2-6** show all filters for the 5 separate $N = 4096$ training sets, which all agree with the one below.

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

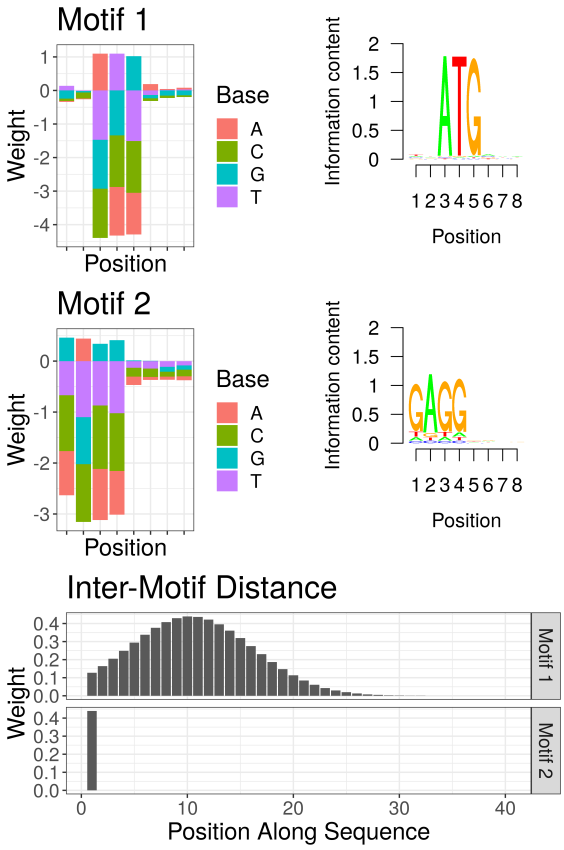

Figure 5: This figure shows the raw filter weights, corresponding seqLogo plots, and inter-motif distances for a single filter pair, used to predict **transcription initiation**. The **HyperXPair** model has correctly learned that the Shine-Dalgarno sequence should occur ~5-13 bases upstream from the Kozak consensus sequence to initiate transcription.

We note that Koo and Eddy [2019] observed how CNN depth could cause earlier layers to learn motif fragments instead of complete motifs. Our architecture does not appear to suffer from this problem. On one hand, our model only contains 2 layers, and so may not have the depth required to obstruct motif interpretability. On the other hand, our second layer is a co-occurrence module that models distance-dependence explicitly, and so may act to add depth without any compromise in the coherency of the first layer.

5.3 Study 2: HyperXPair works for real data

We also apply **HyperXPair** to two real data sets, for which we hypothesize motif pair interactions could play a role. In both cases, we want to predict the presence of a splice event in a segment of DNA. The first data set contains labelled donor splice sites (occurring upstream), while the other contains acceptor splice sites (occurring downstream). *Although we hypothesize that motif pairing contributes to splice prediction, we do not necessarily believe motif pairing defines splice prediction.* Nevertheless, these data let us evaluate the real-world utility of **HyperXPair** along two lines of evidence: (1) model performance in terms of classification accuracy, and (2) biological plausibility in terms of discovered motifs or motif pairs.

Using same biological priors as before ($M = 8$ motif length and $S = 40$ motif gap), we model the donor and acceptor splice sites separately. Figure 6 shows the performance of our model and baselines on the donor and acceptor data. Here, we see that running **HyperXPair** with 1 filter pair can achieve an impressive ~70% AUC, matching the NLP and LSTM baselines, as well as the 2-filter CNN baselines[3]. We also observe, unsurprisingly, that both deep CNNs and wider shallow CNNs can outperform the $F = 2$ **HyperXPair** model. This is because our model

[3]A comment on the AUC reported by Albaradei et al. [2020], and how it relates to both global pooling and the AUC reported here, is provided in the **Supplement**.

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

implies the hypothesis that a single motif pair determines RNA splicing, which is not a complete description of RNA splicing. However, our primary motivation is not accuracy, but rather to obtain an interpretable model that yields meaningful biological insights.

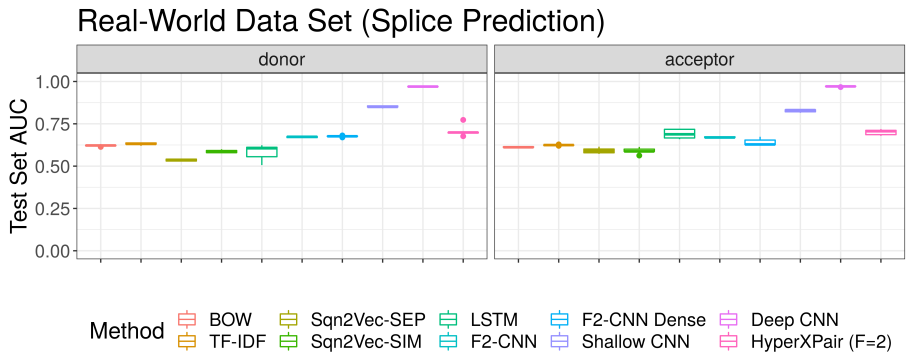

Figure 6: This figure shows the test set AUC (y-axis) for several methods (x-axis) as measured on the splice prediction data (facet). Acronyms: bag-of-words (BOW), term frequency-inverse document frequency (TF-IDF), 2-filter CNN (F2-CNN), 2-filter CNN with intermediate hidden layer (F2-CNN Dense).

HyperXPair can use real data to generate detailed hypotheses about genomic biology. Figure 7A shows the motifs and distances learned for one of the donor training sets. Here, the model learned to associate two motifs, located ~4 bases apart, with the splice event. In fact, the incredibly low variance of the distance distribution suggests that the model may have learned to stitch together two adjacent motifs into one larger motif. Indeed, we see how (C|A)AGGT(G|A)AG from Filter 2 overlaps with T(G|A)AGTACC on Filter 1. This finding is important for two reasons: (1) the complete motif almost perfectly resembles the known donor consensus motif '(C|A)AGGT(G|A)AG' to which the U1 small nuclear ribonucleoprotein particle (U1 snRNP) protein binds [Mount et al., 1992], and (2) it shows that **HyperXPair** can still produce meaningful results even when the biological hypothesis about distance-dependent interactions does not apply.

Figure 7B shows the motifs and distances learned for one of the acceptor training sets. Here, the model learned to associate two motifs, located ~13-21 bases apart, with the splice event. Filter 1 is the downstream motif, which looks like CGACGAGG. This exact 8-base motif shows up in 5/5 training sets, leading us to believe it represents the canonical AGG acceptor splice motif described in the literature [Mount et al., 1992]. Filter 2 is the upstream motif, which looks like CCGCCCTG. This motif shows up in 2/5 training sets, along with the same consensus distance. A cursory look through the literature did not reveal an obvious binding protein for this motif (c.f., Ray et al. [2013]). However, upstream the acceptor site is known to be pyrimidine-rich [Mount et al., 1992], meaning that we should expect Ts or Cs to occur to the *left* of our first motif, as we see here. Thus, the discovered C-rich motif seems biologically plausible. (In the other 3/5 training sets, we see something that resembles a second AGG motif, which could possibly indicate an alternate splice site, known to occur nearby primary splice sites [Hiller et al., 2004]).

Figure 7C proposes a simple schematic of *Drosophila melanogaster* splicing based on a synthesis of the **HyperXPair** results with the relevant literature (c.f., Mount et al. [1992], Rosenberg et al. [2015]). **Supplemental Figures 10-19** show all filters for the 5 donor and acceptor training sets. **Supplemental Figures 20-29** show all filters for a replication of this study using the biological prior $S = 160$ instead of $S = 40$.

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

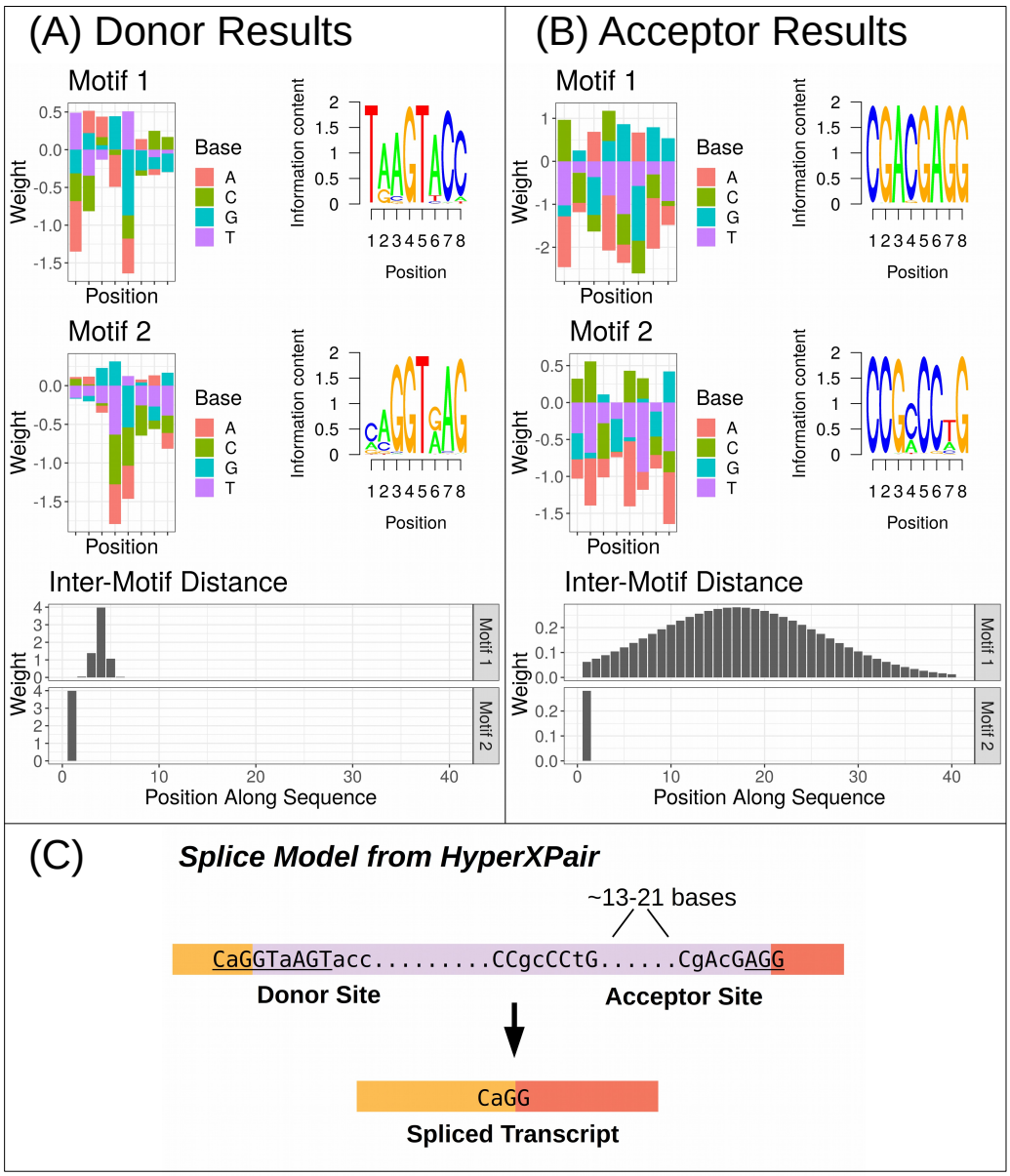

Figure 7: **Panel A** shows the raw filter weights, corresponding seqLogo plots, and inter-motif distances for a single filter pair, used to predict a **donor splice event**. The model appears to have concatenated two adjacent small motifs into a single larger motif, as evidenced by the small motif gap, its low variance, and the overlapping signal. The concatenated motif resembles the known donor consensus motif '(C|A)AGGT(G|A)AG'. **Panel B** shows the raw filter weights, corresponding seqLogo plots, and inter-motif distances for a single filter pair, used to predict an **acceptor splice event**. One motif in the pair appears to contain the canonical AGG acceptor splice motif. **Panel C** proposes a simple schematic of *Drosophila melanogaster* splicing based on a synthesis of the **HyperXPair** results with relevant literature. Lowercase letters symbolize degenerate motif bases; underlined letters symbolize agreement with the literature.

5.4 Study 3: HyperXPair is robust to filter number

In the real-world setting, we may not know the number of motifs involved. However, **HyperXPair** can still perform well even if we over-specify the number of motifs needed. Using the first $N = 4096$ replicate data set, we repeated the Bayesian Optimization scheme for $F = [3, 4, 5]$ filters

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

(corresponding to $[3, 6, 10]$ filter pairs). In all cases, the model successfully learned the GGAGG consensus motif, and achieved an AUC greater than 90%. The model also successfully learned ATG, and the correct GGAGG-ATG distance, when $F = 4$. Interestingly, when $F = 3$ and $F = 5$, the model instead learned the importance of placing GGAGG *after a stop codon* (instead of *before a start codon*). Although this is not the exact rule we expected **HyperXPair** to learn, it is a valid rule given the simulated data. (It also highlights how filters can learn degenerate motifs: TAG, TAA, and TGA are all stop codons, 2 of which get captured by the "T(A|G)A" filter.)

Note that the appropriate motifs emerge without any regularization on the CNN filters themselves. The only regularization in our model occurs at the final layer, which is a simple regression. We expect that aggressively regularizing the CNN filters, for example as proposed by [Koo et al., 2019] and [Ploenzke and Irizarry, 2018], could further improve motif coherence and interpretability when needed. **Supplemental Figures 7-9** show all filters for the $F = [3, 4, 5]$ runs.

6 Conclusion

Although a deep CNN can implicitly learn motif interactions, the identity of the motifs, and the relevant distances between them, are not readily interpretable from the model. In contrast, our proposed neural network architecture can answer two questions: (1) What are the relevant consensus motifs? and (2) What are the inter-motif distance distributions?

By learning the motifs and inter-motif distances explicitly, **HyperXPair** offers insights into the mechaninistic basis of the genomic event under study. In the case of transcription initiation, we were able to recover knowledge that the Shine-Dalgarno sequence should occur ~5-13 bases upstream from the Kozak consensus sequence. In the case of *Drosophila melanogaster* splicing, we were able to verify established knowledge about the donor and acceptor splice sites, and also discover a novel motif pair: a C-rich motif occurring ~13-21 bases upstream from the canonical acceptor splice site. These findings demonstrate how **HyperXPair** can generate detailed hypotheses about genomic biology, enabling biologists to adapt neural networks to advance knowledge in their field.

7 Data Availability

We will make all code publicly available after peer review of the article.

References

Jiong Ma, Allan Campbell, and Samuel Karlin. Correlations between Shine-Dalgarno Sequences and Gene Features Such as Predicted Expression Levels and Operon Structures. *Journal of Bacteriology*, 184(20):5733–5745, October 2002. ISSN 0021-9193. doi: 10.1128/JB.184.20.5733-5745. 2002. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC139613/.

Jernej Ule and Benjamin J. Blencowe. Alternative Splicing Regulatory Networks: Functions, Mechanisms, and Evolution. *Molecular Cell*, 76(2):329–345, October 2019. ISSN 1097-2765. doi: 10.1016/j.molcel.2019.09.017. URL http://www.sciencedirect.com/science/article/pii/S1097276519307026.

Ka-Chun Wong, Yue Li, and Chengbin Peng. Identification of coupling DNA motif pairs on long-range chromatin interactions in human K562 cells. *Bioinformatics*, 32(3):321–324, February 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv555. URL https://academic.oup.com/bioinformatics/article/32/3/321/1743368.

Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, Wei Xie, Gail L. Rosen, Benjamin J. Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne E. Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M. Cofer, Christopher A. Lavender, Srinivas C. Turaga, Amr M. Alexandari, Zhiyong Lu, David J. Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura K. Wiley, Marwin H. S. Segler, Simina M. Boca, S. Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey S. Greene. Opportunities and obstacles for deep learning in biology and medicine. *Journal of the Royal Society, Interface*, 15 (141), 2018. ISSN 1742-5662. doi: 10.1098/rsif.2017.0387.

Kishore Jaganathan, Sofia Kyriazopoulou Panagiotopoulou, Jeremy F. McRae, Siavash Fazel Darbandi, David Knowles, Yang I. Li, Jack A. Kosmicki, Juan Arbelaez, Wenwu Cui,

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Grace B. Schwartz, Eric D. Chow, Efstathios Kanterakis, Hong Gao, Amirali Kia, Serafim Batzoglou, Stephan J. Sanders, and Kyle Kai-How Farh. Predicting Splicing from Primary Sequence with Deep Learning. *Cell*, 176(3):535–548.e24, January 2019. ISSN 0092-8674. doi: 10.1016/j.cell.2018.12.015. URL http://www.sciencedirect.com/science/article/pii/S0092867418316295.

Ruohan Wang, Zishuai Wang, Jianping Wang, and Shuaicheng Li. SpliceFinder: ab initio prediction of splice sites using convolutional neural network. *BMC Bioinformatics*, 20(23):652, December 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-3306-3. URL https://doi.org/10.1186/s12859-019-3306-3.

Somayah Albaradei, Arturo Magana-Mora, Maha Thafar, Mahmut Uludag, Vladimir B. Bajic, Takashi Gojobori, Magbubah Essack, and Boris R. Jankovic. Splice2Deep: An ensemble of deep convolutional neural networks for improved splice site prediction in genomic DNA. *Gene: X*, 5:100035, December 2020. ISSN 2590-1583. doi: 10.1016/j.gene.2020.100035. URL http://www.sciencedirect.com/science/article/pii/S2590158320300097.

Shengdong Ke and Lawrence Allen Chasin. Intronic motif pairs cooperate across exons to promote pre-mRNA splicing. 11(R84), 2010. doi: 10.7916/D84F1P46. URL https://doi.org/10.7916/D84F1P46.

David Alvarez-Melis and Tommi S. Jaakkola. Towards Robust Interpretability with Self-Explaining Neural Networks. June 2018. URL https://arxiv.org/abs/1806.07538v2.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1): 148–175, January 2016. ISSN 1558-2256. doi: 10.1109/JPROC.2015.2494218.

Jian Zhou and Olga G. Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12(10):931–934, October 2015. ISSN 1548-7105. doi: 10.1038/nmeth.3547. URL https://www.nature.com/articles/nmeth.3547.

Peter K. Koo, Praveen Anand, Steffan B. Paul, and Sean R. Eddy. Inferring Sequence-Structure Preferences of RNA-Binding Proteins with Convolutional Residual Networks. *bioRxiv*, page 418459, September 2018. doi: 10.1101/418459. URL https://www.biorxiv.org/content/10.1101/418459v1.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. *arXiv:1704.02685 [cs]*, October 2019. URL http://arxiv.org/abs/1704.02685. arXiv: 1704.02685.

G D Stormo, T D Schneider, L Gold, and A Ehrenfeucht. Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Research*, 10(9):2997–3011, May 1982. ISSN 0305-1048. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC320670/.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, April 1980. ISSN 1432-0770. doi: 10.1007/BF00344251. URL https://doi.org/10.1007/BF00344251.

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3): 328–339, March 1989. ISSN 0096-3518. doi: 10.1109/29.21701.

Thin Nguyen, Hang Le, Thomas P. Quinn, Thuc Le, and Svetha Venkatesh. GraphDTA: Predicting drug–target binding affinity with graph neural networks. *bioRxiv*, page 684662, April 2020a. doi: 10.1101/684662. URL https://www.biorxiv.org/content/10.1101/684662v7.

Suying Bao, Daniel F. Moakley, and Chaolin Zhang. The Splicing Code Goes Deep. *Cell*, 176(3): 414–416, January 2019. ISSN 0092-8674. doi: 10.1016/j.cell.2019.01.013. URL http://www.sciencedirect.com/science/article/pii/S0092867419300467.

John Hawkins and Mikael Bodén. The Applicability of Recurrent Neural Networks for Biological Sequence Analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2: 243–253, 2005.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Hamid Reza Hassanzadeh and May D. Wang. DeeperBind: Enhancing Prediction of Sequence Specificities of DNA Binding Proteins. November 2016. URL https://arxiv.org/abs/1611.05777v1.

Xiaoyong Pan, Peter Rijnbeek, Junchi Yan, and Hong-Bin Shen. Prediction of RNA-protein se-

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.27.270967; this version posted August 29, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

quence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC Genomics*, 19(1):511, July 2018. ISSN 1471-2164. doi: 10.1186/s12864-018-4889-1. URL https://doi.org/10.1186/s12864-018-4889-1.

Peter K. Koo and Sean R. Eddy. Representation learning of genomic sequence motifs with convolutional neural networks. *PLOS Computational Biology*, 15(12):e1007560, December 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007560. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007560.

Peter K. Koo and Matt Ploenzke. Improving Convolutional Network Interpretability with Exponential Activations. *bioRxiv*, page 650804, May 2019. doi: 10.1101/650804. URL https://www.biorxiv.org/content/10.1101/650804v1.

Peter K. Koo, Sharon Qian, Gal Kaplun, Verena Volf, and Dimitris Kalimeris. Robust Neural Networks are More Interpretable for Genomics. *bioRxiv*, page 657437, June 2019. doi: 10.1101/657437. URL https://www.biorxiv.org/content/10.1101/657437v1.

M. S. Ploenzke and R. A. Irizarry. Interpretable Convolution Methods for Learning Genomic Sequence Motifs. *bioRxiv*, page 411934, September 2018. doi: 10.1101/411934. URL https://www.biorxiv.org/content/10.1101/411934v1.

Jasper Snoek, Hugo Larochelle, and Ryan Adams. Practical bayesian optimization of machine learning algorithms. In *NIPS*, pages 2951–2959, 2012.

Dang Nguyen, Sunil Gupta, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Bayesian optimization for categorical and category-specific continuous inputs. In *AAAI*, 2020b.

Carl Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

S M Mount, C Burks, G Hertz, G D Stormo, O White, and C Fields. Splicing signals in Drosophila: intron size, information content, and consensus sequences. *Nucleic Acids Research*, 20(16):4255–4262, August 1992. ISSN 0305-1048. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC334133/.

Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A Compressed Sensing View of Unsupervised Text Embeddings, Bag-of-n-Grams, and LSTMs. February 2018. URL https://openreview.net/forum?id=B1e5ef-C-.

Dang Nguyen, Wei Luo, Tu Dinh Nguyen, Svetha Venkatesh, and Dinh Phung. Sqn2vec: Learning sequence representation via sequential patterns with a gap constraint. In *ECML-PKDD*, pages 569–584. Springer, 2018.

Debashish Ray, Hilal Kazan, Kate B. Cook, Matthew T. Weirauch, Hamed S. Najafabadi, Xiao Li, Serge Gueroussov, Mihai Albu, Hong Zheng, Ally Yang, Hong Na, Manuel Irimia, Leah H. Matzat, Ryan K. Dale, Sarah A. Smith, Christopher A. Yarosh, Seth M. Kelly, Behnam Nabet, Desirea Mecenas, Weimin Li, Rakesh S. Laishram, Mei Qiao, Howard D. Lipshitz, Fabio Piano, Anita H. Corbett, Russ P. Carstens, Brendan J. Frey, Richard A. Anderson, Kristen W. Lynch, Luiz O. F. Penalva, Elissa P. Lei, Andrew G. Fraser, Benjamin J. Blencowe, Quaid D. Morris, and Timothy R. Hughes. A compendium of RNA-binding motifs for decoding gene regulation. *Nature*, 499(7457):172–177, July 2013. ISSN 0028-0836. doi: 10.1038/nature12311. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3929597/.

Michael Hiller, Klaus Huse, Karol Szafranski, Niels Jahn, Jochen Hampe, Stefan Schreiber, Rolf Backofen, and Matthias Platzer. Widespread occurrence of alternative splicing at NAGNAG acceptors contributes to proteome plasticity. *Nature Genetics*, 36(12):1255–1257, December 2004. ISSN 1061-4036. doi: 10.1038/ng1469.

Alexander B. Rosenberg, Rupali P. Patwardhan, Jay Shendure, and Georg Seelig. Learning the sequence determinants of alternative splicing from millions of random sequences. *Cell*, 163(3):698–711, October 2015. ISSN 1097-4172. doi: 10.1016/j.cell.2015.09.054.