ANNUAL REVIEWS

Annual Review of Biomedical Data Science

Identifying Regulatory Elements via Deep Learning

Mira Barshai,[1,*] Eitamar Tripto,[2,*] and Yaron Orenstein[1]

[1]School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva 8410501, Israel; email: yaronore@bgu.ac.il

[2]Department of Biomedical Engineering, Ben-Gurion University of the Negev, Beer-Sheva 8410501, Israel

Annu. Rev. Biomed. Data Sci. 2020. 3:315–38

The *Annual Review of Biomedical Data Science* is online at biodatasci.annualreviews.org

https://doi.org/10.1146/annurev-biodatasci-022020-021940

*These authors contributed equally to this article

Keywords

deep learning, regulatory genomics, gene regulation, motif finding

Abstract

Deep neural networks have been revolutionizing the field of machine learning for the past several years. They have been applied with great success in many domains of the biomedical data sciences and are outperforming extant methods by a large margin. The ability of deep neural networks to pick up local image features and model the interactions between them makes them highly applicable to regulatory genomics. Instead of an image, the networks analyze DNA and RNA sequences and additional epigenomic data. In this review, we survey the successes of deep learning in the field of regulatory genomics. We first describe the fundamental building blocks of deep neural networks, popular architectures used in regulatory genomics, and their training process on molecular sequence data. We then review several key methods in different gene regulation domains. We start with the pioneering method DeepBind and its successors, which were developed to predict protein–DNA binding. We then review methods developed to predict and model epigenetic information, such as histone marks and nucleosome occupancy. Following epigenomics, we review methods to predict protein–RNA binding with its unique challenge of incorporating RNA structure information. Finally, we provide our overall view of the strengths and weaknesses of deep neural networks and prospects for future developments.

1. INTRODUCTION

1.1. Biological Introduction

Gene regulation is one of the most critical processes taking place in every cell all the time (1). The central dogma of molecular biology states that genes encode for proteins (2). Genes reside on several double-stranded DNA molecules, known as chromosomes, which together make up the genome. The DNA segments containing genes are transcribed to RNA molecules known as messenger RNA. These in turn are translated into proteins that perform a specific function in the cell. The accurate and timely regulation of gene transcription and translation is essential for the correct function of individual cells and common function as tissues.

Gene regulation takes place at several layers in the genome (3). First, gene transcription is initiated by transcription factor (TF) binding. TFs are DNA-binding proteins; each protein has a specific DNA binding preference, which dictates its binding to genomic binding sites (BSs). Since genomic DNA is a long molecule wrapped around nucleosomes, only nucleosome-depleted regions are available for binding, thus providing another layer of regulation (4). Other epigenomic signals affect gene transcription, such as histone modifications and DNA methylation. Histone modifications mark regulatory regions with different functions and activity levels, such as enhancers and promoters (5). DNA methylation has a role in gene silencing (6). Once a gene is transcribed to RNA, posttranscriptional regulation occurs at the RNA level. This regulation is mediated through both the RNA sequence and structure, as an RNA molecule, as opposed to double-stranded DNA, may fold on itself (7). RNA-binding proteins (RBPs) bind RNAs with sequence and structure specificity. By RBP binding, RNA stability, splicing, and localization, among other RNA processes, are regulated. Therefore, RBPs are important players in posttranscriptional regulation.

The main mechanism by which control of gene expression is achieved is transcriptional regulation. Although a promoter is necessary to initiate gene transcription, a significant part of eukaryotic transcriptional regulation is mediated by distal *cis*-regulatory modules (8). The most common forms are known as enhancers: clusters of TF BSs that act without regard to orientation, distance, or location (up- or downstream) relative to the transcribed gene (9). Regulation of gene expression is also achieved by additional distal *cis*-acting regulatory elements that include silencers, insulators, and locus control regions (10).

Most of the functional DNA in the genome is likely regulatory (11), with TFs playing a central role in its recognition and utilization. There is a clear role for TFs and RBPs in many human diseases, highlighting the importance of continued efforts for understanding TF- and RBP-mediated gene regulatory mechanisms (12, 13). Multiple regulatory elements may work in a synergistic or redundant manner in regulating the same gene. In addition, interactions on a larger scale affect gene expression, such as enhancer–promoter contacts and the large-scale arrangement of regulatory features along chromosomes and in three dimensions (14). Much effort has been made to characterize and annotate the regulatory genome. For example, the Open Regulatory Annotation database (ORegAnno) is a resource for curated regulatory annotation. It contains information about regulatory regions, TF and RBP BSs, and other regulatory elements (15). The current version of ORegAnno has a total of almost two million unique records. These records cover more than 300 million bp (base pairs) across 18 species. The vast majority of these records are mapped to human and mouse genomes, with slightly less than 1.5 million records in human and slightly more than 400,000 records in mouse.

1.2. Technological Introduction

Due to their importance in almost any cellular process, several technologies have been developed in past years to measure DNA and RNA binding or epigenetic marks on a genome-wide

Table 1 Experimental protocols to measure regulatory elements in high throughput

Technology	Target	Domain	Number of sequences	Resolution	Label
ChIP-seq (16)	TF/histone	In vivo	Thousands	~100 bp	Bound
PBM (23)	TF	In vitro	Thousands	36 bp	Intensity
HT-SELEX (24)	TF	In vitro	Millions	~20 bp	Multiclass
ATAC-seq (25)	Nucleosome	In vivo	Thousands	~1,000 bp	Open
DNase-seq (26)	Nucleosome	In vivo	Thousands	~1,000 bp	Open
CLIP-seq (22)	RBP	In vivo	Thousands	~40 bp	Bound
RNAcompete (27)	RBP	In vitro	Thousands	~40 bp	Intensity
RNA Bind-n-Seq (28)	RBP	In vitro	Millions	~20 bp	Multiclass
scBS-seq (19)	Methylation	In vivo	Millions	1 bp	Intensity
scRRBS-seq (20)	Methylation	In vivo	Millions	1 bp	Intensity
WGBS (21)	Methylation	In vivo	Millions	1 bp	Intensity

scale (**Table 1**). Chromatin immunoprecipitation (ChIP) captures DNA binding in vivo. When combined with high-throughput sequencing, this protocol, known as ChIP-seq, can measure the genome-wide binding of a specific protein (16). Similar protocols can measure histone modifications and their genome-wide occupancy. Nucleosome depletion, which reveals genomic regions available for binding, can be measured on a genome-wide scale using the assay for transposase-accessible chromatin using sequencing (ATAC-seq) and DNase I–hypersensitive (DHS) sites sequencing (DNase-seq) protocols (17). Several methods have been developed to measure DNA methylation on a genome-wide scale using both microarrays and high-throughput sequencing (18), such as single-cell bisulfite sequencing (scBS-seq) (19), reduced-representation scBS-seq (scRRBS-seq) (20), and whole-genome bisulfite sequencing (WGBS) (21). Protein–RNA binding is measured on a transcriptome-wide scale using protocols based on cross-linking immunoprecipitation (CLIP) (22).

Unfortunately, in many cases the in vivo measurements are prone to experimental noise and technological artifacts and are subject to the complexity of the cellular environment. For example, measuring DNA binding in vivo may not reveal the full picture of TF-DNA interactions. First, the nucleosome-depleted regions may not cover the full spectrum of possible DNA BSs. Second, in vivo binding is affected by additional factors, such as chromatin structure, nucleosome positioning, and cofactors.

As alternatives to the noisy in vivo data, the technological advancements made by in vivo experimental protocols have been accompanied by advancements in the in vitro domain (**Table 1**). As opposed to in vivo binding, in vitro binding is purely due to direct TF-DNA or RBP-RNA interactions (or cooperative binding of specific factors) and allows sampling of the full spectrum of DNA or RNA BSs. Protein-binding microarrays (PBMs) were developed to measure protein–DNA binding in a high-throughput, unbiased, and universal manner (23). The binding of a specific protein is reported to around 40,000 synthetic DNA sequences. High-throughput systematic evolution of ligands by exponential enrichment (HT-SELEX) is based on high-throughput sequencing of TF-bound DNA sequences (24). On the RNA front, RNAcompete tests RNA binding by measuring hybridization of bound RNAs on a microarray (27). RNA Bind-n-Seq is based on sequencing of bound RNAs under different protein concentrations to identify RNA binding preferences (28).

1.3. Computational Introduction

The abundance of high-throughput data accumulated by these high-throughput experimental techniques has given rise to many computational challenges. The main computational challenge is known as the motif-finding problem. The goal is to find a repeating substring, with variants, in the set of bound sequences compared to an unbound set. Formally, a motif is a summarization of a set of BSs that share a common pattern. The most popular motif representation is the position weight matrix (PWM), where each column represents the affinity of the protein to different nucleotides in the corresponding position in the BS. These motifs can then be used both for finding novel regulatory elements in new DNA or RNA sequences and for understanding the regulatory mechanism of different elements.

A plethora of methods were developed for this classic bioinformatics problem for more than three decades (29). The common pattern, i.e., motif, represents a putative regulatory element, such as a TF BS. Variants of the motif-finding problem in bound and unbound sequences include identifying motifs in a ranked list of sequences (30), learning models to predict binding intensity (regression) (31), and taking into account RNA structure or epigenetic marks in addition to the sequence information (32).

The various datasets used to find regulatory elements have been deposited in public databases through the years. Hundreds of ChIP-seq, ATAC-seq, DNAse-seq, and other epigenomic experiments are publicly available through the ENCODE (Encyclopedia of DNA Elements) project (33) and Roadmap Epigenomics project (34), both in their raw read format and as called peaks representing bound or nucleosome-depleted genomic regions. PBM data have been deposited and curated in the UniPROBE database (35), while HT-SELEX data are available through the European Nucleotide Archive (36). Both PBM- and HT-SELEX-inferred motifs were compiled in the CIS-BP (Catalog of Inferred Sequence Binding Preferences) database (37). CLIP-seq data and other CLIP experiments have been deposited and curated in doRiNA (38), starBase (39), and CLIPdb (40) databases. A compendium of 244 RNAcompete experiments was published in 2013 (41), and RNA Bind-n-Seq data can be downloaded from the Sequence Read Archive database (42) and ENCODE. The GEO (Gene Expression Omnibus) repository hosts data of many high-throughput experiments, including many measuring regulatory elements (43). Methylation data can be downloaded from many databases, such as MethDB (44). All of these data were pivotal in the development of bioinformatics methods.

A major breakthrough in the machine learning field, termed deep learning, has been revolutionizing the data science world (45). Formally, deep learning refers to neural networks composed of at least two layers. In almost any common task, such as identifying objects in images or playing traditional games by computers, deep neural networks are outperforming previous methods. This revolution has not skipped the bioinformatics field (46). Many classic biomedical data challenges are now solved using deep learning, including problems in the gene regulation domain (31). Prediction accuracy has been improving tremendously for image and text processing tasks (45). Applications of deep neural networks methods to DNA, RNA, and epigenetic data have seen similar boosts in prediction accuracy (46).

In this review, we cover the great success stories of deep learning in regulatory genomics. We first go over the preliminaries of deep neural networks, including convolutional and recurrent neural networks (CNNs and RNNs), autoencoders, deep belief networks (DBNs), and the attention mechanism. Then, we review the pioneering method DeepBind, its successors, and interpretability challenges in predicting protein–DNA binding. In addition, we review methods developed to handle epigenetic and nucleosome occupancy data. Then, we cover the incorporation of RNA structure in the methods for posttranscriptional regulation. Finally, we provide our outlook for future advancements in the field.

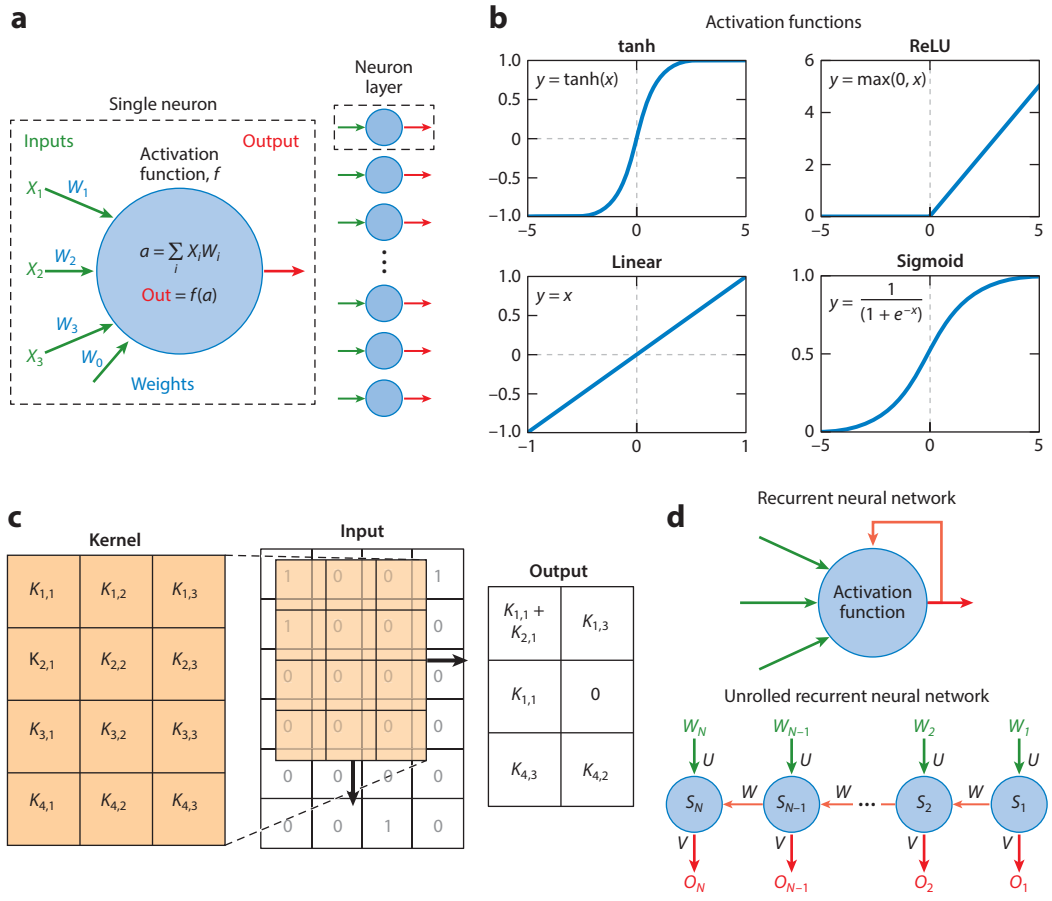

Figure 1

Fundamental building blocks of deep neural networks. (*a*) A basic neuron receives several inputs, and it outputs a linear combination of them, followed by an activation function. Multiple parallel neurons serve as a fully connected layer. (*b*) Activation function examples. The common property of all activation functions is that they pass along the input when it passes a certain threshold. (*c*) Convolutional operation. A weight matrix, termed kernel, traverses the input, and in each stride an inner product is calculated and passed to the output. (*d*) A recurrent layer. The current state S_i and output O_i is a function of the current input W_i and previous state S_{i-1}. Abbreviation: ReLU, rectified linear unit.

2. PRELIMINARIES

2.1. Fundamental Building Blocks of Deep Neural Networks

The basic building block of a neural network is an artificial neuron, which takes as input a vector of real values and computes the weighted sum of these values, followed by an activation function. There are three common families of architectures for connecting neurons into a network: fully connected, convolutional, and recurrent.

2.1.1. Basic neural networks. The basic unit of neural networks is the neuron (**Figure 1*a***). A neuron is formally defined as a node receiving several inputs $\{X_i\}_1^n$ and generating a single output. The output is a linear combination of the inputs followed by an application of an activation function. The linear combination depends on a set of weights $\{W_i\}_1^n$. Each input has a corresponding

weight. These weights are part of the model parameters. In addition, a neuron has a bias term W_0, i.e., a constant added to the weighted sum. For input vector $\mathbf{X}$ of length n and weight vector $\mathbf{W}$ of length $n + 1$, the net output of neuron a is

$$a(\mathbf{W}, \mathbf{X}) = W_0 + \sum_{i=1}^{n} W_i \cdot X_i \quad \text{(neuron)}. \qquad 1.$$

The linear combination is the net output, which is then passed to an activation function (**Figure 1*b***). Popular functions include sigmoid and rectified linear unit (ReLU) (47):

$$f(x) = \frac{1}{1 + e^{-x}} \quad \text{(sigmoid)}, \qquad 2.$$

$$f(x) = \max(x, 0) \quad \text{(ReLU)}. \qquad 3.$$

All functions share the common theme of passing along the signal if enough input was received (analog to a neuron in the brain), or restricting the range of output values. A set of nodes in one layer constitute a hidden layer. A hidden layer connected to all preceding and succeeding nodes is referred to as a fully connected layer.

Most commonly, a network's final output is dictated by a layer of neurons (**Figure 1*a***). Their activation function depends on the problem definition. For regression, where a numerical value, such as DNA binding intensity, is predicted, a linear or a ReLU function fits best. For classification, where a probability to belong to a class, such as bound or unbound, is predicted, a sigmoid function fits best, as it outputs a value between 0 and 1. For multiclassification problems, where a distribution is predicted (e.g., strongly bound, weakly bound, unbound), the softmax function, a generalization of the sigmoid function to output a distribution over classes, fits best.

Fully connected layers can model feature interactions, such as the dependence between different DNA sequence features. For example, there is a requirement for a BS to be composed of two half sites with a variable gap. It can also merge different feature sets, such as sequence features and RNA expression levels. A cascade of several fully connected layers can approximate nonlinear dependencies.

2.1.2. Convolutional neural networks. CNNs extend basic neural networks by convolution layers (**Figure 2*a***). In a convolutional layer a kernel is run through the input to detect local features (**Figure 1*c***). The weights of the kernel are learned in the training process. DNA sequences are often one-hot encoded before convolution, i.e., each nucleotide is transformed to a binary vector of length 4, with 1 set at the position corresponding to that nucleotide. A kernel on a genomic sequence is also known as a motif detector, i.e., a kernel running on one dimension to detect local sequence features. This is equivalent to a PWM with the difference that kernel weights are unconstrained, i.e., they do not have to form a distribution in each position and may be negative. A convolutional layer may include multiple kernels. In some cases, there are parallel convolutional layers, where the kernel width and the number of kernels vary between the layers.

Formally, for a DNA sequence a one-dimensional (1D) convolutional kernel is a matrix of dimension $4 \times k$. The kernel moves in strides over the sequence. For a sequence of length L, a kernel with stride one produces $L - k + 1$ outputs. Formally, for sequence $\mathbf{S}$ and kernel $\mathbf{K}$, the output c in position $i + 1$ of $\mathbf{S}$ is

$$c(\mathbf{K}, \mathbf{S}, i + 1) = \sum_{j=i+1}^{i+k} \sum_{\ell=1}^{4} K_{j,\ell} \cdot \delta(\ell, S_j) \quad \text{(convolution)}, \qquad 4.$$

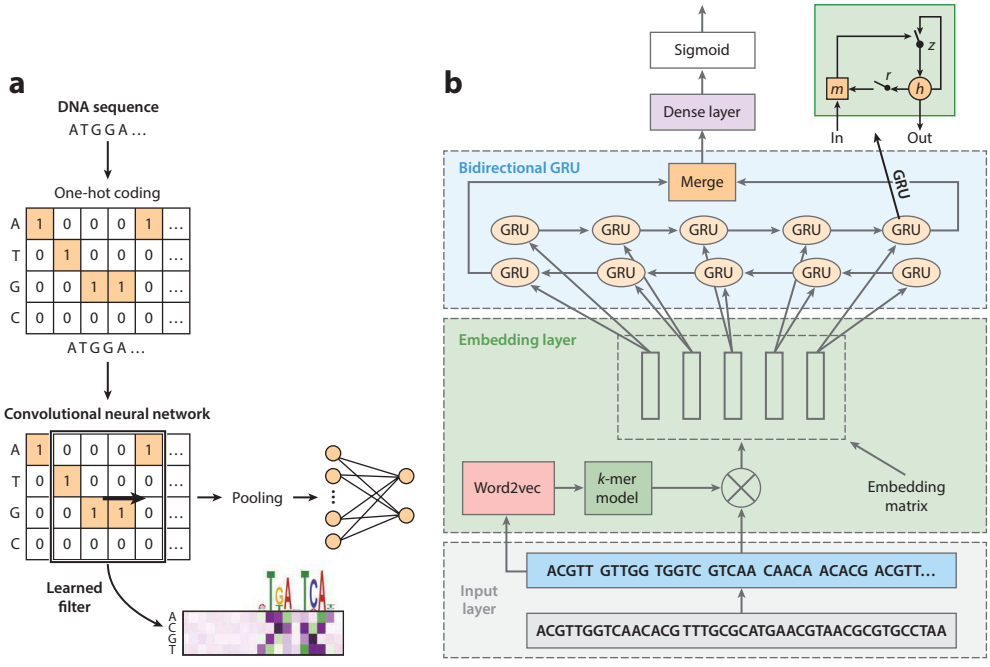

Figure 2

Popular deep neural network architectures in regulatory genomics. (*a*) A convolutional neural network. A one-hot-encoded representation of a sequence is fed to a one-dimensional convolutional layer, which provides a latent representation of the sequence using learned weights. The output is then down-sampled using a pooling operation. A fully connected layer captures interactions between different sequence features. (*b*) A recurrent neural network (RNN). The sequence is split into *k*-mers, which are embedded to a lower-dimension space using Word2vec. The embedded sequence flows through an RNN, which consists of gated recurrent units (GRUs). The output of each unit depends both on the current input and on the previous layer. GRU controls the output by reset (r) and update (z) gates, and h and m denote the current and candidate unit states, respectively. Panels adapted with permission from Reference 48 (*a*) and Reference 49 (*b*), both under a CC BY 4.0 license (**https://creativecommons.org/licenses/by/4.0/**).

where $\delta(\ell, S_j) = 1$ if and only if nucleotide S_j is ℓ. The values produced for $0 \leq i \leq L - k$ are then pooled by maximum or average pooling in configurable window size and step size between pooling windows. For example, the global max pooling value g of vector Y of size $L - k + 1$ is given by

$$g(Y) = \max_{0 \leq i \leq L-k} Y_i \quad \text{(max pooling)}. \qquad 5.$$

2.1.3. Recurrent neural networks. RNNs were developed to model a context of a word in a sentence (50). Similarly, on a DNA or RNA sequence, we can train them to learn the context of a regulatory element, such as a protein BS (51) (**Figure 2*b***). RNNs have been extremely effective in learning and predicting the so-called sentiment of a sentence (50, 52). For the DNA binding scenario, the sentiment may be protein bound or unbound. In this case, a DNA sequence serves as a text in a regulatory language we are training the model to learn.

RNNs have been shown to outperform CNNs and other deep neural networks on sequential data (53). They are capable of modeling the ordering dependence in sequences by memorizing long-range information through network recurrent loops. In each iteration, the input is composed of both the previous layer output and the current input segment (e.g., a DNA word). The output is

calculated by integrating both current and previous sequence information (**Figure 1*d***). Formally, for current word w_i and previous state s_{i-1}, the calculation of the next state is given by

$$s_i = f(\mathbf{U} \cdot w_i + \mathbf{W} \cdot s_{i-1}), \quad 6.$$

$$o_i = g(\mathbf{V} \cdot s_i). \quad 7.$$

where **U**, **V**, and **W** are weight matrices, as in **Figure 1*d***.

As the DNA vocabulary of all k-mers may be too large to convert in one-hot encoding (as its size will be 4^k), an embedding layer may be used to reduce dimensions, i.e., convert the one-hot-encoded vector to a real vector of a few hundred elements. Common embedding techniques include Word2vec, which has to be trained on a large corpora of text (in our case, DNA sequences) (54).

When using RNNs one has to choose their direction and implementation. Bidirectional RNNs are useful for other scenarios where both past and future inputs matter (55). In DNA or RNA sequences, this means that both the 5′ and 3′ flanks of an element are important to determine its activity and function. The cyclic structure makes a seemingly shallow RNN over long-time prediction actually very deep if unrolled in time. To resolve the problem of parameter updates getting increasingly smaller in the training process rendered by this, Hochreiter & Schmidhuber (56) substituted the hidden units in RNNs with long short-term memory (LSTM) units. Gated recurrent units (GRUs) have also been introduced for a similar purpose (57).

2.2. Standard Architectures

Popular deep neural network architectures include autoencoders, DBNs, and the attention mechanism.

2.2.1. Autoencoders. Autoencoders are neural networks that learn efficient data representations in an unsupervised manner (58). The aim of an autoencoder is to learn a representation for a set of data, typically for dimensionality reduction, by training the network to ignore noise. Along with the reduction, a reconstructing side is learned, where the autoencoder generates from the reduced encoding a representation as close as possible to its original input. Several variants to the basic model exist, with the aim of forcing the learned representations of the input to assume useful properties (59).

Autoencoders can embed sequence data into a low-dimensional space with a hidden layer, called the bottleneck layer, and reconstruct the original input sequence data (60). This approach forces the network to extract useful features in the sequence, as the bottleneck layer makes it infeasible to learn the perfect reconstruction. Reconstructing the data is often interpreted as denoising because the unimportant variations are automatically left out. Multiple nonlinear layers generalize linear autoencoders to a nonlinear dimensionality reduction method.

2.2.2. Attention mechanism. An attention mechanism enables a model to select which features are important given any input context. It provides a form of conditional importance to all input features (61). Each attention vector **w** is multiplied element-wise by an input vector **x** to produce a vector of the form $\mathbf{w} \odot \mathbf{x}$. An attention mechanism is essentially a vector of probabilities usually obtained by employing the softmax function on the final output layer of a neural network.

The attention mechanism was first proposed for machine translation and automatic image captioning (62, 63). For sequence data it allows salient features to come dynamically to the forefront for each regulatory element as needed. As a result, the global knowledge of the model is enhanced

by the local knowledge that each element provides. In other words, the attention mechanism offers an insight into the model's decision-making process by revealing a set of individualized importance scores that describe how important each feature is for the specific prediction task. Further analysis of these importance scores reveals valuable insights that are not directly apparent in the unattended data.

2.2.3. Deep belief networks. DBNs are composed of multiple layers of latent variables (also known as hidden units) with connections between the layers, but not between units within each layer (64). Each subnetwork's hidden layer serves as the input layer for the next. This architecture leads to a fast, layer-by-layer unsupervised training procedure, where contrastive divergence is applied to each subnetwork in turn, starting from the lowest pair of layers (where the lowest visible layer comprises the training data).

DBNs are obtained by stacking restricted Boltzmann machines (RBMs) (65). An RBM is a generative stochastic model that learns a probability distribution over the input space. RBMs are a variant of Boltzmann machines, with the restriction that their neurons must form a bipartite graph. This restriction allows for more efficient training algorithms than the general class of Boltzmann machines, which allow for connections between hidden units. RBMs have had success in dimensionality reduction for various genomic applications.

2.3. Training Neural Networks

The process of training neural networks is the main bottleneck in their utilization. The training procedure is based on optimizing a predefined loss function. This function reflects the attempt to learn model parameters that will bring predictions on training data closest to the true labels. Common loss functions include cross-entropy for classification (e.g., bound or unbound) and minimum squared error (MSE) for regression (e.g., binding intensity). Formally, the loss functions for real labels $\mathbf{y}$, predicted labels $\tilde{\mathbf{y}}$, and C categories for classification are

$$\text{MSE}(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2, \qquad 8.$$

$$\text{cross-entropy}(\mathbf{y}, \tilde{\mathbf{y}}) = -\sum_{i=1}^{n}\sum_{j\in C} y_{ij}\log(\tilde{y}_{ij}). \qquad 9.$$

The gradient, a vector of the derivatives of the loss function with respect to the parameters of the model, points in the direction of the biggest increase in the function. Thus, a step in the opposite direction, i.e., an update of the parameters by negation of the derivative, will incur a decrease in the loss function. The parameters are learned by taking a step in the opposite of the gradient of the loss function. Many variants of this basic gradient descent optimization algorithm exist (66), varying by the step size of parameter updates and the contribution of previous steps to the current step (i.e., momentum).

Neural networks are very sensitive to the choice of hyperparameters. These include hyperparameters of the optimization procedure, such as the learning step size, the contribution of previous steps, the number of epochs (training iterations over the whole data), and batch size (the number of data points in each parameter update). Hyperparameters also include architecture details, such as the number of layers, the number of neurons, and their activation function. In convolutional layers these include kernel size (corresponding to an element, such as a BS, in a sequence), kernel stride (kernel step size), and the number of kernels. In RNNs one has to decide about their directionality,

memory size, and specific implementation (GRU or LSTM). The common method to find optimal parameters is by random search, specifically, by testing various random combinations of hyperparameters and choosing the best one based on a validation set (a subset of the training dataset aside).

3. PROTEIN–DNA BINDING

3.1. Convolutional Neural Networks for Protein–DNA Binding

Computational models for protein–DNA binding have been constantly evolving for over three decades. They originated from simple consensus sequences and have evolved to more complex *k*-mer-based models as high-throughput quantitative data became available for many TFs. The most popular model to date is still the PWM, where each column represents the affinity of the protein to the different nucleotides in the corresponding position in the BS. Model parameters can be learned using different learning techniques. Trained models were used for both predicting binding to new sequences and investigating the binding mechanism. The continued improvements in models and data were outperformed with the emergence of deep learning and its application in genomics.

DeepBind pioneered the use of deep learning for predicting protein–DNA binding (31). The network architecture of DeepBind is based on a convolutional layer, a pooling layer, and a fully connected layer (**Figure 3*a***). This architecture proved to be very useful on different types of data: PBMs, ChIP-seq, HT-SELEX, RNAcompete, and CLIP-seq. For each dataset measuring the binding of a specific protein, a model is learned to enable binding predictions to new DNA or RNA sequences. For the case of PBM, for example, the fluorescence binding intensity serves

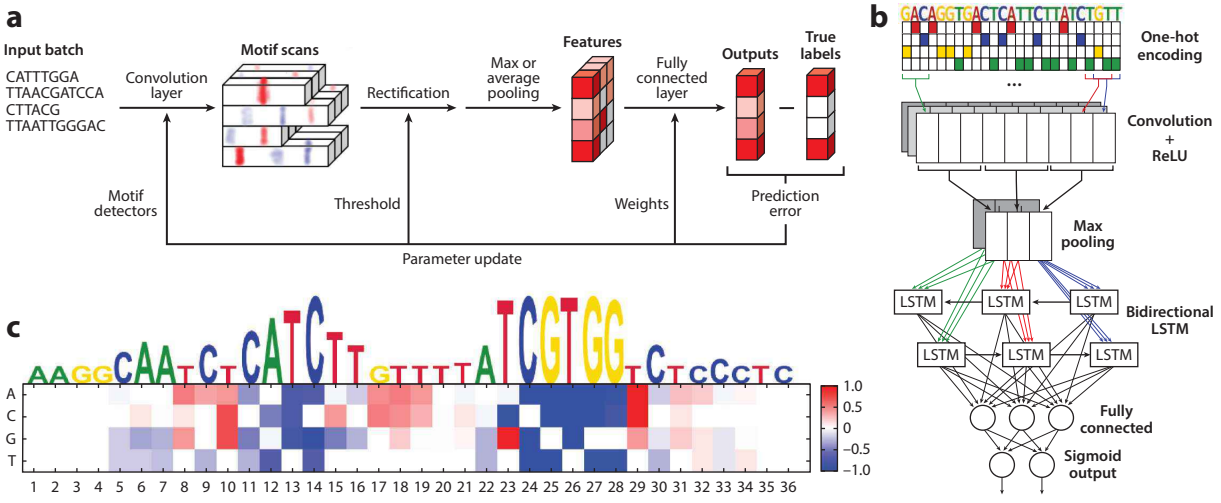

Figure 3

Deep neural networks for protein–DNA binding. (*a*) DeepBind CNN, composed of a one-dimensional convolutional layer, ReLU activation, global maximum or average pooling, and a fully connected layer. The network eventually outputs a single binding intensity score. (*b*) DanQ combines a CNN and a bidirectional RNN for predicting protein–DNA binding genome-wide based on DNA sequence. (*c*) A mutation map visualization to highlight important positions in a DNA sequence of a regulatory element. The height of each letter represents the ability of a mutation at that position to damage the binding and consequently decrease the binding score. Each cell contains the sensitivity of the model to a mutation at the corresponding position. Panels adapted with permission from (*a*) Reference 31, copyright 2015 Springer Nature, and (*b*) Reference 69, copyright 2016 Oxford University Press. Abbreviations: CNN, convolutional neural network; LSTM, long short-term memory; ReLU, rectified linear unit; RNN, recurrent neural network.

as the label. For ChIP-seq the label is binary (bound or unbound), where called peaks serve as the bound regions and nearby genomic regions or shuffled peaks serve as unbound sequences.

The method predicts binding affinity of a protein to a DNA or RNA sequence in two steps, consisting of a convolution layer for detecting local sequence features and a fully connected layer for modeling interactions and dependencies between these features. DeepBind uses 16 kernels that are 14–32 nt long [however, subsequent studies showed that many shorter kernels are more effective (67)]. The fully connected layer combines local features detected by the convolution layer into higher-level structures. Each neuron considers the local features in different combinations and orientations. This allows longer motifs, motif pairs and combinations, and more complex patterns to be picked up.

The performance of DeepBind was evaluated using more than 1,000 publicly available datasets, encompassing DNA binding in vivo (e.g., ChIP-seq) and in vitro (e.g., PBM). DeepBind outperformed previous methods, some of which are based on extensive biological knowledge or are customized to specific technological platforms or biological systems. Although the ranking of competing methods varied widely depending on the types of experiments and TFs, DeepBind consistently outperformed all of them, even when the training and testing datasets were of different types, which means that the knowledge encapsulated in the model is biologically relevant and transferable.

3.2. Advanced Architectures for Protein–DNA Binding

Although multiple studies have demonstrated the superiority of CNNs over other existing methods, inappropriate structure design would still result in even poorer performance than conventional models (68). Zeng et al. (68) developed a parameterized CNN to conduct a systematic exploration of CNNs on two classification tasks, motif discovery and motif occupancy. They examined the performance of nine variants of CNNs and observed that CNNs do not gain from deepness for the motif discovery task as long as the structure is appropriately designed. In addition, they concluded that researchers should pay more attention to particular hyperparameters that can be tuned in CNNs (such as the kernel size, the number of kernels, the pooling window or convolution strides, and the choice of the window size of input DNA sequences) or include prior genomic information if possible.

Following DeepBind, several more advanced architectures have been applied to protein–DNA binding, such as RNNs, CNN–RNN combinations, and residual networks. DanQ (69) uses an architecture that utilizes the strengths of both CNNs and RNNs to predict the function of DNA sequences (**Figure 3*b***). The architecture is based on a convolutional layer, followed by a max pooling layer and a bidirectional recurrent layer, which processes the sequence from left to right and from right to left, allowing for both upstream and downstream biological contexts to be learned. The convolution layer captures regulatory motifs, while the recurrent layer captures long-term dependencies between the motifs in order to learn a regulatory grammar to improve predictions. The final layers of DanQ comprise a fully connected layer and a sigmoid output. DanQ improved considerably upon other models in predicting regulatory elements across several metrics. For some regulatory markers, DanQ achieved over a 50% relative improvement in the area under the precision-recall curve metric compared to related models.

A more recent development used a weakly supervised framework (i.e., using overlapping subsequences with an overall sequence label), which combined multiple-instance learning with a hybrid deep neural network and used *k*-mer encoding for DNA sequences, for modeling protein–DNA binding (70), i.e., predicting binding of a specific protein given a DNA sequence. This framework segments sequences into multiple overlapping instances using a sliding window, and then encodes

all instances into inputs of high-order dependencies using *k*-mer encoding. Then, it separately computes a score for all instances in the same bag using a hybrid deep neural network that integrates CNNs and RNNs. Finally, it aggregates the predicted values of all instances as the final prediction of this bag. The experimental results on in vivo datasets demonstrated the superior performance of the proposed framework.

3.3. Interpretability of Deep Neural Networks for Protein Binding

In many biological applications, researchers are more interested in the molecular mechanisms revealed by the predictive model rather than the predictions themselves. Although deep neural networks can achieve state-of-the-art accuracy, it is more challenging to interpret them than more standard statistical models. For example, the high accuracy of DeepBind models in predicting protein–DNA binding indicated that meaningful representations were learned, but the authors only made the lowest-level representations explicit, while higher levels remained a black box (31). Opening this black box and interpreting higher levels would present an opportunity to gain insights about the language of gene regulation beyond the word level, which is a long-standing challenge in the field.

The simplest method to interpret a neural network is analogous to in silico mutagenesis (71). Given a particular data point *X*, each feature of *X* can be systematically varied while the rest of the features are fixed (e.g., mutating a single nucleotide), and how the network's output changes can be tracked. The changes can be visualized as a heatmap with a corresponding sequence logo on top, where letter height indicates the sensitivity of the corresponding position to mutations (**Figure 3*c***). This approach is easy to implement but can be computationally expensive, as the network recomputes the output for each mutation of *X*. A computationally tractable approximation to mutagenesis is to take the derivative of the network output with respect to each feature of *X*. This derivative can be computed in one pass, and it conveys the sensitivity of the output to small perturbations in input features. Features with large positive or negative derivatives may be more influential to the outcome.

In strict terms, the derivative is a valid measure of influence for only infinitesimally small perturbations to the input, whereas in practice, researchers are interested in larger changes (e.g., a mutation of A to C). Several variations of the derivative-based interpretation methods, such as integrated gradients (72) and DeepLIFT (73), have been developed to partially address this limitation. Other interpretation methods, such as LIME (74), select a small number of features to explain why a prediction is made.

For CNNs, it is also possible to visualize each convolution filter as a heatmap or PWM-style logo image. These visualizations are useful to obtain a sense of what local features the network might be learning. A caveat is that multiple convolution filters might be learning partially redundant features, and how the local features interact is less clear, because such interaction depends on the higher layers of the network.

4. EPIGENETIC MARKS

4.1. The Pioneers: DeepSEA and Basset

As with predicting protein–DNA binding, many methods have been developed in the past to predict epigenetic marks from DNA sequence on a genome-wide scale. Most methods are based on local sequence features combined together through different learning methods, such as hidden Markov models, random forests, or support vector machines. While those methods have shown

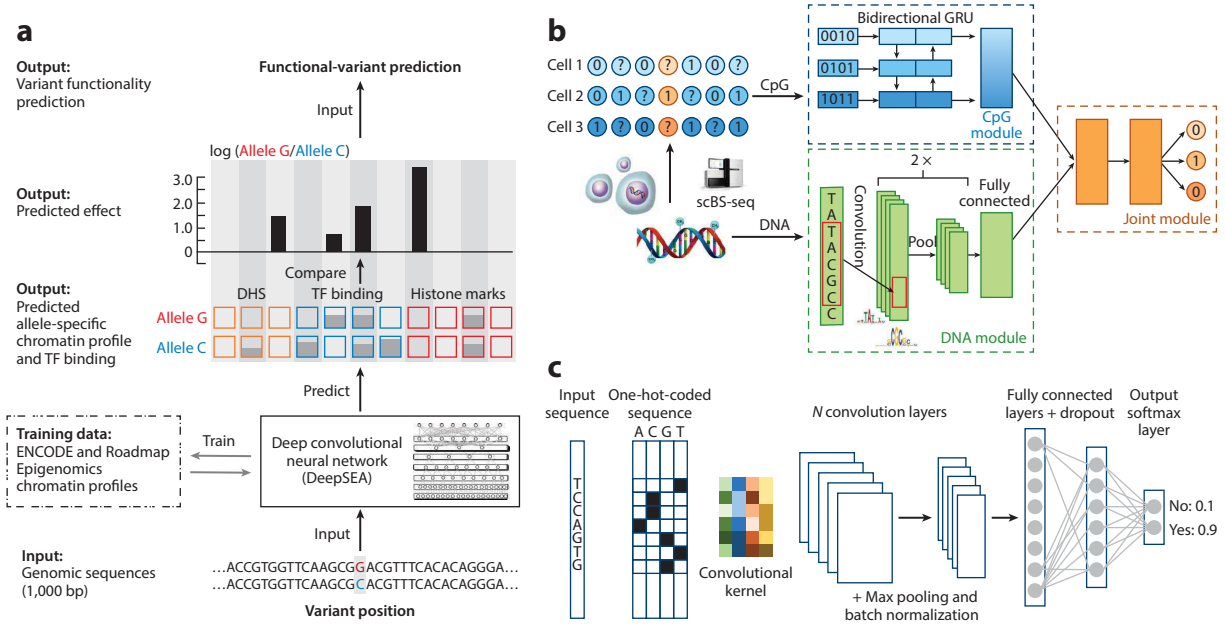

Figure 4

Examples of deep learning–based approaches to predict epigenetic marks. (*a*) DeepSEA pipeline and algorithm. DeepSEA is one of the pioneer algorithms based on deep learning to predict TF, histone, and DHS profiles. (*b*) DeepCpG pipeline and algorithm. DeepCpG predicts DNA methylation in single cells at a single-nucleotide resolution. (*c*) DeepEnhancer architecture. DeepEnhancer is a CNN model designed to detect enhancers genome-wide. Abbreviations: CNN, convolutional neural network; DHS, DNase I–hypersensitivity; GRU, gated recurrent unit; scBS-seq, single-cell bisulfite sequencing; TF, transcription factor. Panels adapted with permission from (*a*) Reference 75, copyright 2015 Springer Nature; (*b*) Reference 78, under a CC BY 4.0 license (**https://creativecommons.org/licenses/by/4.0/**); and (*c*) Reference 79, copyright 2016 IEEE.

some success, they were significantly outperformed in the same tasks with the emergence of deep learning methods.

The first method to predict epigenetic marks using deep learning is DeepSEA (deep learning–based sequence analyzer) (75). DeepSEA was developed as a fully sequence-based algorithmic framework for noncoding-variant effect prediction (**Figure 4*a***). It learns regulatory sequence code from genomic sequences by learning to simultaneously predict large-scale chromatin-profiling data, including TF binding, DHS, and histone mark profiles. DeepSEA includes three major features in its model: integrating sequence information from a wide sequence context, learning sequence code at multiple spatial scales with a hierarchical architecture, and multitask joint learning of diverse chromatin factors sharing predictive features. To train the model, the developers of DeepSEA compiled a diverse compendium of genome-wide chromatin profiles from the ENCODE (33) and Roadmap Epigenomics projects (77), including 690 TF binding profiles for 160 different TFs, 125 DHS profiles, and 104 histone mark profiles. In total, 521.6 Mbp of the genome (17%) were used as a regulatory information-rich set for training the DeepSEA regulatory code model.

Around the same time as the development of DeepSEA, Basset was developed to predict nucleosome-depleted regions genome-wide. Basset applies CNNs to learn functional activities of DNA sequences. Basset simultaneously predicts the accessibility of DNA sequences in 164 cell types mapped by DNase-seq from the ENCODE (33) and the Roadmap Epigenomics projects (77). From these datasets, Basset learns the relevant sequence motifs and the regulatory

logic with which they are combined to determine cell-specific DNA accessibility. Basset achieves a level of accuracy that provides meaningful, nucleotide-precision measurements.

4.2. Predicting Methylation in Single-Nucleotide Resolution

DNA methylation measurements pose a more challenging problem—predicting molecular phenotypes at single-nucleotide resolution (80). While protein–DNA binding, histone modifications, and nucleosome occupancy are measured in vivo at a resolution that provides peaks at least 100 bp in length, DNA methylation is measured at much higher resolution, and in some cases even single-nucleotide resolution. This brings new challenges in both prediction resolution and training time on much larger datasets.

Several methods have been developed to predict methylation at different levels of resolution from DNA sequence alone. DeepCpG is a computational approach based on deep neural networks to predict methylation states in single cells (78). DeepCpG was evaluated on single-cell methylation data from five cell types generated using alternative sequencing protocols and outperformed extant methods at the time (**Figure 4*b***). Interpretation of model parameters provided insights into how sequence composition affects methylation variability. MRCNN (methylation regression by CNN) was later developed as a deep learning method to predict genome-wide DNA methylation from DNA sequence (81). Experiments showed that the MRCNN model is more precise than DeepCpG. MRCNN was also used to discover motifs associated with DNA methylation.

To further improve prediction, several methods incorporated additional information such as gene expression or genome 3D architecture. DeepMethyl is a deep learning–based software to predict the methylation state of DNA CpG dinucleotides using features inferred from 3D genome topology (based on Hi-C) and DNA sequence patterns (82). Various stacked denoising autoencoder architectures with different configurations of hidden layers and amounts of pretraining data were tested. Using the methylation states of sequentially neighboring regions as one of the learning features, the model achieved an accuracy of almost 90%. When the methylation states of sequentially neighboring regions are unknown, the accuracy was almost 85%. Levy-Jurgenson et al. (83) developed a general model to predict DNA methylation for a given sample in any CpG position based solely on the sample's gene expression profile and the sequence surrounding the CpG. Depending on gene–CpG proximity, the model attained a Spearman correlation of up to 0.84 for thousands of CpG sites on two separate test sets of CpG positions and subjects (cancer and healthy samples). Using attention in their deep learning architecture offered a novel framework with which to extract valuable insights from gene expression data when combined with sequence information, as demonstrated by linking several motifs and genes to methylation activity.

4.3. Annotating Enhancers Genome-Wide

A long-standing challenge in gene regulation research is the annotation of cell type–specific enhancers in a genome-wide manner. Available histone marks and nucleosome occupancy data can help to categorize genomic regions (5), but these are not always available for all cell types. Thus, researchers turn to computational methods to predict enhancers from sequence and, in the last couple of years, to deep learning–based methods.

BiRen uses a deep learning hybrid architecture to predict enhancers based on DNA sequence alone (84). BiRen exhibited superior accuracy, robustness, and generalizability in enhancer prediction relative to other state-of-the-art enhancer predictors based on sequence characteristics. A follow-up, DeepEnhancer, distinguishes enhancers from background genomic sequences by

using a deep CNN on DNA sequence alone (85). The DeepEnhancer model was trained on permissive enhancers and then adopted a transfer learning strategy to fine-tune the model on cell type–specific enhancers (**Figure 4*c***). Results demonstrated the effectiveness and efficiency of DeepEnhancer in the classification of enhancers against random sequences, exhibiting advantages of deep learning over traditional sequence-based classifiers. The use of max pooling and batch normalization layers in DeepEnhancer was especially effective, as demonstrated by a comparison of different architectures (85).

Newer methods use additional data sources on top of DNA sequences. PEDLA learns to identify enhancers from massively heterogeneous data and generalizes in ways that are mostly consistent across various cell types (86). PEDLA was trained on 1,114-dimensional heterogeneous features in H1 cells and outperformed five extant methods at the time by integrating different data sources, including histone modifications (ChIP-seq), TFs and cofactors (ChIP-seq), chromatin accessibility (DNase-seq), transcription (RNA-seq), DNA methylation (RRBS), CpG islands, evolutionary conservation, sequence signatures, and occupancy of TFs. PEDLA was further extended to iteratively learn from 22 training cell types and showed superior performance in independent test sets, achieving 95% accuracy. EnhancerDBN is a DBN-based computational method for enhancer prediction (87). EnhancerDBN combines diverse features, composed of DNA sequence compositional features, DNA methylation, and histone modifications. EnhancerDBN outperformed 13 methods in prediction and demonstrated that GC content and DNA methylation can serve as relevant features for enhancer prediction.

5. PROTEIN–RNA BINDING

5.1. Sequence-Based Neural Networks for RNA Binding

Protein–RNA binding measurements have been accumulating at a rapid pace (88), leading to the development of many computational methods to infer RNA binding preferences from high-throughput data (89). The challenges raised by computational modeling of protein–RNA binding share many similarities with the challenges tackled in the protein–DNA binding domain. Thus, it is not surprising that many methods and models used for DNA binding have been used to solve RNA binding modeling. As with protein–DNA binding and epigenetic marks, deep neural networks' ability to predict accurate protein–RNA binding has been unsurpassed.

The first methods using deep learning considered only protein–RNA binding sequence preferences, without taking into account the role of RNA structure in the binding models. DeepBind, the method that pioneered the application of deep learning to protein–DNA binding, as reviewed in Section 3, was also applied to protein–RNA binding. DeepBind models were trained to learn a binding model from RNAcompete data and outperformed competing methods in both in vitro binding prediction, as measured using RNAcompete data in cross-validation, and in vivo data, as measured by CLIP-seq experiments (31).

While predictions based only on sequence provide some power, they are limited by the fact that they do not consider other sources of information to improve predictions. iDeep is a hybrid CNN and DBN to predict the RBP interaction sites and motifs on RNAs (**Figure 5*a***). It combines different sources of features, including sequences, structures, region type, and CLIP cobinding information (90). iDeep outperformed the state-of-the-art methods on predicting CLIP binding. It was also used to infer binding sequence motifs. The results of iDeep show that region type and CLIP cobinding contribute to predicting RBP BSs on RNAs, and complement the CNN models relying solely on RNA sequence. CONCISE is a neural network–based approach to model distances to predict protein–RNA binding by using spline transformations. It was used to extend iDeep to model distances of various genomic marks, such as 5′ and 3′ exons and AG dinucleotides (91).

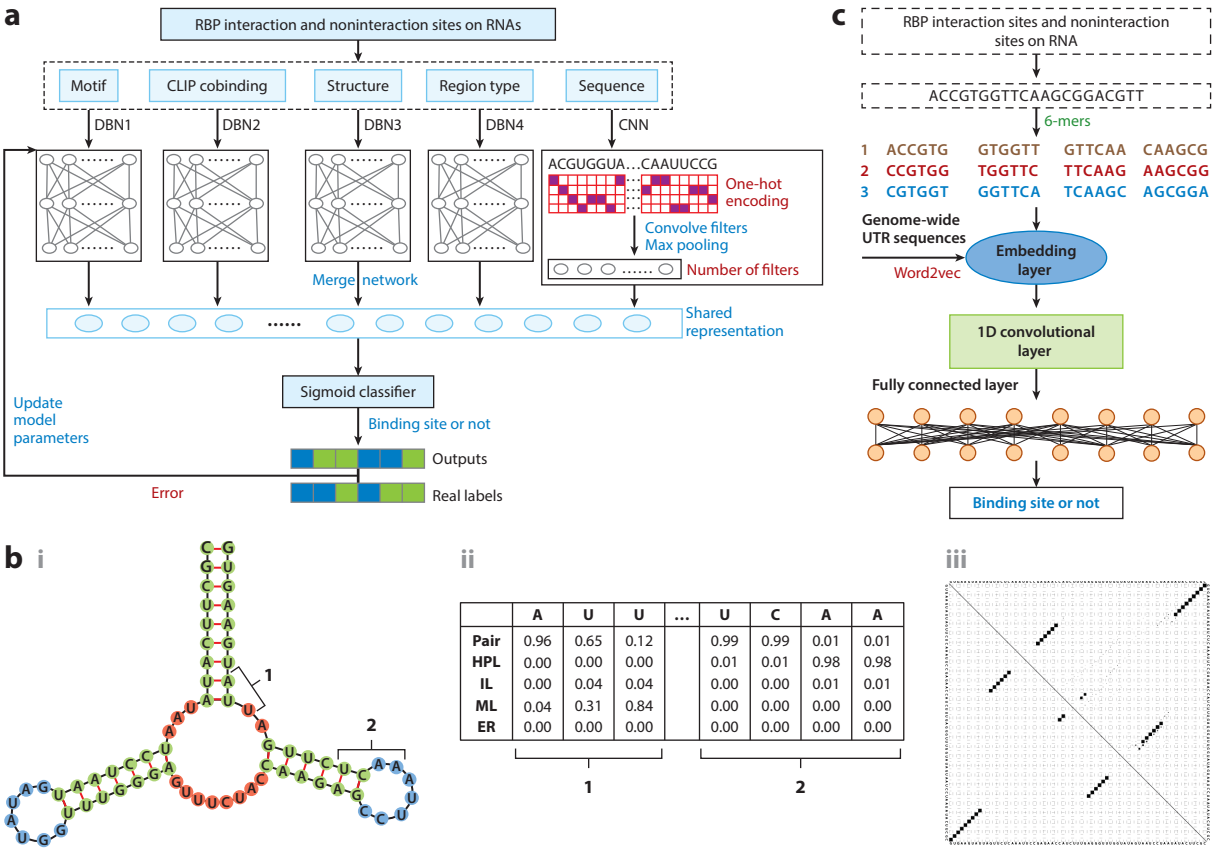

Figure 5

Deep neural networks for protein–RNA binding and RNA structure representation. (*a*) The iDeep algorithm for predicting binding of a protein to an RNA sequence. (*b*) Computational representations of RNA secondary structure. In a graph representation (*i*), each nucleotide is a node and edges connect adjacent and based-paired nucleotides. In a probability matrix representation (*ii*), each column is a distribution over structural contexts in which the nucleotide may reside in the ensemble of all RNA folds. In a matrix representation (*iii*), each cell (x, y) is the base-pairing probability of nucleotides in positions x and y. (*c*) In iDeepV architecture, a CNN for predicting protein–RNA binding receives k-mers following an embedding layer. Abbreviations: CNN, convolutional neural network; DBN, deep belief network; RBP, RNA-binding protein; UTR, untranslated region. Panels adapted with permission from (*a*) Reference 90, under a CC BY 4.0 license (**https://creativecommons.org/licenses/by/4.0/**), and (*c*) Reference 93, copyright 2018 Elsevier.

This extension outperformed iDeep on various CLIP datasets. Extending iDeep, iDeepA applies a hybrid model of a CNN and an attention mechanism to learn discriminant high-level features for predicting RBP BSs (92). iDeepA improves the prediction accuracy mostly for proteins with a small number of known RNA BSs.

5.2. Incorporating RNA Structure into Deep Neural Networks

It was shown that RNA binding model performance can be increased by adding information regarding the spatial structure of the RNA molecule to the data that are fed to the model (94). To do so, one would need to properly represent the structure in a way that the network can learn from. There are various approaches to represent RNA structure computationally (**Figure 5*b***). Here, we cover two different approaches for representing RNA structure information.

1. RNA graph structure: Each nucleotide is represented as a node in a graph, with edges connecting base-paired nucleotides and adjacent nucleotides on the sequence level. Several methods were developed to predict RNA structure from sequence. RNAfold calculates the minimum free energy structure (95), while RNAshapes outputs a set of representative structures (96). These graph structures may be given as input to a graph convolution layer, and it will find useful local features on the RNA level.
2. RNA structure probabilities: An RNA molecule may fold in many ways, varying in free energy and corresponding probability to reside in a specific conformation. Thus, structure probabilities may be assigned to each nucleotide representing the probability of that nucleotide being in a specific structural context, or to a pair of nucleotides being paired. On a single-nucleotide level, the probability of being unpaired or, in a more refined categorization, of being in a hairpin, multiloop, inner loop, or external region can be computed based on the RNA sequence using RNAplfold (95). For a sequence of length L, structure probabilities may be represented by a $2 \times L$ or $5 \times L$ matrix and given as input to a neural network. Similarly, base-pairing probabilities between each pair of nucleotides can be represented by an $L \times L$ probability matrix.

Much of the computational challenge in modeling protein–RNA binding resides in efficient incorporation of RNA structure information. The unique challenge of adding the RNA structure information in the models was tackled by many methods in the past, usually by augmenting known models such as PWMs or *k*-mer-based models to include this information. Deep learning–based methods followed similar strategies. iDeepS, an extension to iDeep and iDeepA, simultaneously identifies the binding sequence and structure motifs from RNA sequences using CNNs and a bidirectional RNN implemented by LSTM units (94). However, iDeepS performs worse on some RBPs compared to iDeep. This is due to the fact that iDeep uses additional sources of information, for example, genomic context, whereas iDeepS instead uses only sequences and predicted RNA structures. However, iDeepS outperforms GraphProt, a sequence- and structure-based method that uses support vectors to learn feature weights (97). Another algorithm, pysster, detects both sequence and structure motifs using CNNs, where the sequence and structure are encoded in an extended alphabet by combining the sequence and structure alphabets (98).

As opposed to the aforementioned methods developed to train a model on CLIP in vivo data, which suffers from experimental biases and noise (99), other methods were developed to learn a model from RNAcompete in vitro data. DLPRB (deep learning for protein–RNA binding) uses CNNs and RNNs to jointly analyze RNA sequences and structures from high-throughput in vitro data (67). Similarly, cDeepBind (100) introduces predicted RNA structures as contexts of RNA–protein interactions into the CNNs to enhance the prediction performance. Both methods, developed independently and simultaneously, outperformed the state-of-the-art method RCK by a large margin (101). A more recent approach, RDense (102), further improved the prediction of models learned from RNAcompete data by a unique combination of a neural network architecture and RNA sequence and structure representation in the network.

5.3. *k*-mer Encoding for Modeling RNA Binding Preferences

An often used approach to analyze long sequences is to divide them into *k*-mers (103, 104), which refer to subsequences of length *k* (e.g., 4-mers of RNA sequences are *AAAA*, *AAAC*, . . . , *UUUU*). The *k*-mer representation is widely used for predicting RNA–protein BSs. They mainly have two forms of representation: (*a*) a one-hot vector of length 4^k of all zeroes except for the corresponding position of the *k*-mer, and (*b*) a *k*-mer frequency vector consisting of *k*-mer frequencies of all *k*-mers (similar to bag of words in natural language processing).

Deepnet-RBP encodes the sequences, secondary structures, and tertiary structural information into a unified feature representation. This representation is further fed into a multimodal DBN to predict RBP BSs and motifs (105). However, *k*-mer frequencies cannot model the distance difference of individual *k*-mers, as some *k*-mers are semantically correlated, considering the polymorphic status of nucleic acids. Taking this into account, some methods first learn the distributed representations using word-embedding methods, which treat *k*-mers as words and sequences as sentences. The learned representations can reveal the similarity between *k*-mers. iDeepV first learns distributed vectors of *k*-mers from genome-wide sequences (93). Then, these learned vectors are further fed into a CNN to classify bound sites from unbound sites (**Figure 5*c***). iDeepV performs similarly to DeepBind, while for RBPs with a small number of training samples, iDeepV performs better. In addition, the learned distributed representations can be used for other downstream classification tasks.

Similarly, a *k*-mer-embedding method was recently used to achieve state-of-the-art performence on models learned from RNAcompete data. ThermoNet, a thermodynamic prediction model, integrates a new sequence-embedding CNN model over a thermodynamic ensemble of RNA secondary structures (106). First, the sequence-embedding CNN generalizes the existing *k*-mer-based methods by jointly learning convolutional filters and *k*-mer embeddings to represent RNA sequence contexts. Second, the thermodynamic average of deep learning predictions explores structural variability and improves the prediction, especially for the structured RNAs. Extensive experiments have demonstrated that ThermoNet significantly outperforms existing approaches on both in vitro and in vivo data.

6. DISCUSSION

The sweeping success of deep learning in various artificial intelligence fields has been followed closely with applications in genomics (107). The regulatory genomics field, which requires methods to identify local patterns in massive datasets, is particularly suitable for deep neural network applications. Computational challenges, such as predicting genetic variant function, epigenetic marks, and protein–DNA and protein–RNA binding, have been addressed by different solutions in the last five years using deep learning approaches (108). These approaches have been outperforming the state of the art by a large margin.

The success of deep neural networks has mostly been achieved by the timely convergence of both large genomic datasets, generated by either microarrays or high-throughput sequencing, and the computational advances of neural networks and the hardware used to train them. For the past decade large genomic data have begun accumulating at an ever-accelerating pace. While classic computational methods were successful to some extent, they were always limited due to their linear nature and other simplifying assumptions, as well as due to the requirement of classic machine learning approaches for feature engineering. Deep neural networks are achieving state-of-the-art performance thanks to their ability to learn and approximate complex functions, the kind of functions that may underlie the mechanism of gene regulation. They are particularly adept at analyzing raw data, without the need to define or formalize specific features. Thus, they succeed in two tasks: They learn the features by themselves, and they learn complex functions without making any assumptions.

One major limitation of deep learning–based methods is their dependence on accurate labels. The methods presented in this paper all share the need for high-throughput datasets with precise labeling. A network will never be able to achieve higher accuracy than in its given labels. This upper bound on accuracy is commonly measured by the concordance of two replicate experiments: If a replicate experiment achieves some level of accuracy, we cannot expect an algorithm to outperform

it, as there is inherent noise, either biological or experimental, that cannot be modeled by an algorithm.

Another important drawback is the generalizability of these methods. As the underlying model is a complex mathematical function, rather than a simple generative model defined analytically, it may learn an experiment rather than biological phenomena. Regulating the learning process of network parameters by techniques such as early stopping and dropout or loss penalties may provide some remedy, but there is no guarantee for success. One way to gauge the ability of a method to generalize is by testing it on independent datasets that measured the same biological phenomena but with a different experimental protocol. For example, protein–DNA binding models learned from HT-SELEX data should be tested on PBM data to ensure that they are not overfitted to SELEX data.

This new paradigm of deep learning is also criticized by many biologists for lacking interpretability and for being driven by the data rather than a hypothesized model (109). While there is consensus that deep learning–based methods work great as prediction tools, and may replace experiments in many scenarios, it is still an open challenge how to use them to test hypotheses and models. Several methods, such as integrated gradient, saliency maps, and DeepLIFT, have overcome some of the interpretability obstacle, but they are based on highlighting feature weights for a single input. There is still no single method that can interpret model parameters independent of any input.

Following these achievements and drawbacks, deep learning–based approaches are currently used most successfully as experiment simulators (110). As the number of DNA sequences of length L is 4^L, it is infeasible to test all sequences of $L \geq 20$, and many regulatory elements or regions occupy a hundred or more base pairs. Moreover, they can occur in different epigenomic contexts. Thus, the power of these networks lies mostly in their ability to recapitulate an experiment and generalize over an experimental dataset. As they can approximate a huge space of complex functions, provided there are enough variable data and given an appropriate architecture and learning hyperparameters, these networks can predict an accurate response to any unseen data point (111).

Using these networks as experiment simulators, the research community may find new ways to learn biology based on these simulations (112). As any synthetic sequence can be given as input for predicting a response, these sequences can be designed to learn governing underlying molecular mechanisms. For example, positional effects in a sequence can be tested using the same sequence context, but with different positions of a regulatory element in it. The number of elements and their combination and orientation are just a few more examples of underlying principles that can be tested by synthetic sequences. Thus, we may still yet see the promise of these networks, not only in predictability but also in discovering new biology by deciphering the gene regulatory grammar.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

LITERATURE CITED

1. Lodish H, Berk A, Kaiser CA, Krieger M, Bretscher A, et al. 2016. *Molecular Cell Biology*. New York: Macmillan. 8th ed.
2. Crick F. 1970. Central dogma of molecular biology. *Nature* 227:561–63
3. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, et al. 2002. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* 298:799–804

4. Jiang C, Pugh BF. 2009. Nucleosome positioning and gene regulation: advances through genomics. *Nat. Rev. Genet.* 10:161–72
5. Ernst J, Kellis M. 2012. ChromHMM: automating chromatin-state discovery and characterization. *Nat. Methods* 9:215–16
6. Attwood J, Yung R, Richardson B. 2002. DNA methylation and the regulation of gene transcription. *Cell. Mol. Life Sci.* 59:241–57
7. Glisovic T, Bachorik JL, Yong J, Dreyfuss G. 2008. RNA-binding proteins and post-transcriptional gene regulation. *FEBS Lett.* 582:1977–86
8. Hardison RC, Taylor J. 2012. Genomic approaches towards finding *cis*-regulatory modules in animals. *Nat. Rev. Genet.* 13:469–83
9. Andersson R, Sandelin A. 2019. Determinants of enhancer and promoter activities of regulatory elements. *Nat. Rev. Genet.* 21:71–87
10. Gaszner M, Felsenfeld G. 2006. Insulators: exploiting transcriptional and epigenetic mechanisms. *Nat. Rev. Genet.* 7:703–13
11. Yue F, Cheng Y, Breschi A, Vierstra J, Wu W, et al. 2014. A comparative encyclopedia of DNA elements in the mouse genome. *Nature* 515:355–64
12. Lambert SA, Jolma A, Campitelli LF, Das PK, Yin Y, et al. 2018. The human transcription factors. *Cell* 172:650–65
13. Neelamraju Y, Gonzalez-Perez A, Bhat-Nakshatri P, Nakshatri H, Janga SC. 2018. Mutational landscape of RNA-binding proteins in human cancers. *RNA Biol.* 15:115–29
14. Li Y, Hu M, Shen Y. 2018. Gene regulation in the 3D genome. *Hum. Mol. Genet.* 27:R228–33
15. Lesurf R, Cotto KC, Wang G, Griffith M, Kasaian K, et al. 2016. ORegAnno 3.0: a community-driven resource for curated regulatory annotation. *Nucleic Acids Res.* 44:D126–32
16. Furey TS. 2012. ChIP–seq and beyond: new and improved methodologies to detect and characterize protein–DNA interactions. *Nat. Rev. Genet.* 13:840–52
17. Tsompana M, Buck MJ. 2014. Chromatin accessibility: a window into the genome. *Epigenet. Chromatin* 7:33
18. Bibikova M, Fan JB. 2010. Genome-wide DNA methylation profiling. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 2:210–23
19. Clark SJ, Smallwood SA, Lee HJ, Krueger F, Reik W, Kelsey G. 2017. Genome-wide base-resolution mapping of DNA methylation in single cells using single-cell bisulfite sequencing (scBS-seq). *Nat. Protoc.* 12:534–47
20. Smallwood SA, Lee HJ, Angermueller C, Krueger F, Saadeh H, et al. 2014. Single-cell genome-wide bisulfite sequencing for assessing epigenetic heterogeneity. *Nat. Methods* 11:817–20
21. Li N, Ye M, Li Y, Yan Z, Butcher LM, et al. 2010. Whole genome DNA methylation analysis based on high throughput sequencing technology. *Methods* 52:203–12
22. Marchese D, de Groot NS, Lorenzo Gotor N, Livi CM, Tartaglia GG. 2016. Advances in the characterization of RNA-binding proteins. *Wiley Interdiscip. Rev. RNA* 7:793–810
23. Berger MF, Philippakis AA, Qureshi AM, He FS, Estep PW 3rd, Bulyk ML. 2006. Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nat. Biotechnol.* 24:1429–35
24. Jolma A, Kivioja T, Toivonen J, Cheng L, Wei G, et al. 2010. Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. *Genome Res.* 20:861–73
25. Buenrostro JD, Wu B, Chang HY, Greenleaf WJ. 2015. ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Curr. Protoc. Mol. Biol.* 109:21–29
26. Song L, Crawford GE. 2010. DNase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells. *Cold Spring Harb. Protoc.* **https://www.doi.org/10.1101/pdb.prot5384**
27. Ray D, Kazan H, Chan ET, Castillo LP, Chaudhry S, et al. 2009. Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nat. Biotechnol.* 27:667–70
28. Lambert N, Robertson A, Jangi M, McGeary S, Sharp PA, Burge CB. 2014. RNA Bind-n-Seq: quantitative assessment of the sequence and structural binding specificity of RNA binding proteins. *Mol. Cell* 54:887–900

29. Hashim FA, Mabrouk MS, Al-Atabany W. 2019. Review of different sequence motif finding algorithms. *Avicenna J. Med. Biotechnol.* 11:130–48
30. Leibovich L, Yakhini Z. 2014. Mutual enrichment in ranked lists and the statistical assessment of position weight matrix motifs. *Algorithms Mol. Biol.* 9:11
31. Alipanahi B, Delong A, Weirauch MT, Frey BJ. 2015. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* 33:831–38
32. Narlikar L, Gordân R, Hartemink AJ. 2007. Nucleosome occupancy information improves *de novo* motif discovery. In *Proceedings of the 11th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2007)*, ed. T Speed, H Huang, pp. 107–21. Cham, Switz.: Springer
33. Sloan CA, Chan ET, Davidson JM, Malladi VS, Strattan JS, et al. 2016. ENCODE data at the ENCODE portal. *Nucleic Acids Res.* 44:D726–32
34. Bernstein BE, Stamatoyannopoulos JA, Costello JF, Ren B, Milosavljevic A, et al. 2010. The NIH Roadmap Epigenomics Mapping Consortium. *Nat. Biotechnol.* 28:1045–48
35. Hume MA, Barrera LA, Gisselbrecht SS, Bulyk ML. 2015. UniPROBE, update 2015: new tools and content for the online database of protein-binding microarray data on protein–DNA interactions. *Nucleic Acids Res.* 43:D117–22
36. Leinonen R, Akhtar R, Birney E, Bower L, Cerdeno-Tárraga A, et al. 2010. The European Nucleotide Archive. *Nucleic Acids Res.* 39:D28–31
37. Weirauch MT, Yang A, Albu M, Cote AG, Montenegro-Montero A, et al. 2014. Determination and inference of eukaryotic transcription factor sequence specificity. *Cell* 158:1431–43
38. Blin K, Dieterich C, Wurmus R, Rajewsky N, Landthaler M, Akalin A. 2015. DoRiNA 2.0—upgrading the doRiNA database of RNA interactions in post-transcriptional regulation. *Nucleic Acids Res.* 43:D160–67
39. Li JH, Liu S, Zhou H, Qu LH, Yang JH. 2014. starBase v2.0: decoding miRNA–ceRNA, miRNA–ncRNA and protein–RNA interaction networks from large-scale CLIP-seq data. *Nucleic Acids Res.* 42:D92–97
40. Yang YCT, Di C, Hu B, Zhou M, Liu Y, et al. 2015. CLIPdb: a CLIP-seq database for protein-RNA interactions. *BMC Genom.* 16:51
41. Ray D, Kazan H, Cook KB, Weirauch MT, Najafabadi HS, et al. 2013. A compendium of RNA-binding motifs for decoding gene regulation. *Nature* 499:172–77
42. Kodama Y, Shumway M, Leinonen R. 2012. The Sequence Read Archive: explosive growth of sequencing data. *Nucleic Acids Res.* 40:D54–56
43. Barrett T, Troup DB, Wilhite SE, Ledoux P, Rudnev D, et al. 2007. NCBI GEO: mining tens of millions of expression profiles—database and tools update. *Nucleic Acids Res.* 35:D760–65
44. Grunau C, Renault E, Rosenthal A, Roizes G. 2001. MethDB—a public database for DNA methylation data. *Nucleic Acids Res.* 29:270–74
45. LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature* 521:436–44
46. Min S, Lee B, Yoon S. 2017. Deep learning in bioinformatics. *Brief. Bioinform.* 18:851–69
47. Nair V, Hinton GE. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, ed. J Fürnkranz, T Joachims, pp. 807–14. Madison, WI: Omnipress
48. Hirohara M, Saito Y, Koda Y, Sato K, Sakakibara Y. 2018. Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. *BMC Bioinform.* 19:526
49. Shen Z, Bao W, Huang DS. 2018. Recurrent neural network for predicting transcription factor binding sites. *Sci. Rep.* 8:15270
50. Graves A, Mohamed A-R, Hinton G. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–49. New York: IEEE
51. Hassanzadeh HR, Wang MD. 2016. DeeperBind: enhancing prediction of sequence specificities of DNA binding proteins. In *Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 178–83. New York: IEEE

52. Tang D, Qin B, Liu T. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, ed. L Màrquez, C Callison-Burch, J Su, pp. 1422–32. Stroudsburg, PA: Assoc. Comput. Linguist.
53. Schmidhuber J. 2015. Deep learning in neural networks: an overview. *Neural Netw.* 61:85–117
54. Ling W, Dyer C, Black AW, Trancoso I. 2015. Two/too simple adaptations of Word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ed. R Mihalcea, J Chai, A Sarkar, pp. 1299–304. Stroudsburg, PA: Assoc. Comput. Linguist.
55. Schuster M, Paliwal KK. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Proc.* 45:2673–81
56. Hochreiter S, Schmidhuber J. 1997. Long short-term memory. *Neural Comput.* 9:1735–80
57. Chung J, Gulcehre C, Cho K, Bengio Y. 2015. Gated feedback recurrent neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, ed. F Bach, D Blei, pp. 2067–75. New York: Assoc. Comput. Mach.
58. Liou CY, Cheng WC, Liou JW, Liou DR. 2014. Autoencoder for words. *Neurocomputing* 139:84–96
59. Goodfellow I, Bengio Y, Courville A. 2016. *Deep Learning*. Cambridge, MA: MIT Press
60. Vincent P, Larochelle H, Bengio Y, Manzagol PA. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–103. New York: Assoc. Comput. Mach.
61. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, et al. 2017. Attention is all you need. In *Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NIPS 2017)*, ed. I Guyon, UV Luxburg, S Bengio, H Wallach, R Fergus, et al. **https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf**
62. Bahdanau D, Cho K, Bengio Y. 2014. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 [cs.CL]
63. Xu K, Ba J, Kiros R, Cho K, Courville A, et al. 2015. Show, attend and tell: neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, ed. F Bach, D Blei, pp. 2048–57. New York: Assoc. Comput. Mach.
64. Hinton GE. 2009. Deep belief networks. *Scholarpedia* 4:5947
65. Sutskever I, Hinton GE, Taylor GW. 2009. The recurrent temporal restricted Boltzmann machine. In *Proceedings of the 21st International Conference on Advances in Neural Information Processing Systems (NIPS 2008)*, ed. D Koller, D Schuurmans, Y Bengio, L Bottou. **https://papers.nips.cc/paper/3567-the-recurrent-temporal-restricted-boltzmann-machine**
66. Ruder S. 2016. An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs.LG]
67. Ben-Bassat I, Chor B, Orenstein Y. 2018. A deep neural network approach for learning intrinsic protein-RNA binding preferences. *Bioinformatics* 34:i638–46
68. Zeng H, Edwards MD, Liu G, Gifford DK. 2016. Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics* 32:i121–27
69. Quang D, Xie X. 2016. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* 44:e107
70. Zhang Q, Shen Z, Huang DS. 2019. Modeling *in-vivo* protein-DNA binding by combining multiple-instance learning with a hybrid deep neural network. *Sci. Rep.* 9:8484
71. Zhou J, Theesfeld CL, Yao K, Chen KM, Wong AK, Troyanskaya OG. 2018. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nat. Genet.* 50:1171–79
72. Sundararajan M, Taly A, Yan Q. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, ed. D Precup, YW Teh, pp. 3319–28. New York: Assoc. Comput. Mach.
73. Shrikumar A, Greenside P, Kundaje A. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, ed. D Precup, YW Teh, pp. 3145–53. New York: Assoc. Comput. Mach.
74. Ribeiro MT, Singh S, Guestrin C. 2016. Why should I trust you? Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–44. New York: Assoc. Comput. Mach.

75. Zhou J, Troyanskaya OG. 2015. Predicting effects of noncoding variants with deep learning–based sequence model. *Nat. Methods* 12:931–34
76. Deleted in proof
77. Kundaje A, Meuleman W, Ernst J, Bilenky M, Yen A, et al. 2015. Integrative analysis of 111 reference human epigenomes. *Nature* 518:317–30
78. Angermueller C, Lee HJ, Reik W, Stegle O. 2017. DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.* 18:67
79. Min X, Chen N, Chen T, Jiang R. 2016. DeepEnhancer: predicting enhancers by convolutional neural networks. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, ed. T Tian, Y Wang, Q Jiang, X Hu, Y Liu, et al., pp. 637–44. New York: IEEE
80. Bock C. 2012. Analysing and interpreting DNA methylation data. *Nat. Rev. Genet.* 13:705–19
81. Tian Q, Zou J, Tang J, Fang Y, Yu Z, Fan S. 2019. MRCNN: a deep learning model for regression of genome-wide DNA methylation. *BMC Genom.* 20:192
82. Wang Y, Liu T, Xu D, Shi H, Zhang C, et al. 2016. Predicting DNA methylation state of CpG dinucleotide using genome topological features and deep networks. *Sci. Rep.* 6:19598
83. Levy-Jurgenson A, Tekpli X, Kristensen VN, Yakhini Z. 2019. Predicting methylation from sequence and gene expression using deep learning with attention. In *Proceedings of the 6th International Conference on Algorithms for Computational Biology*, ed. I Holmes, C Martín-Vide, MA Vega-Rodríguez, pp. 179–90. Cham, Switz.: Springer
84. Yang B, Liu F, Ren C, Ouyang Z, Xie Z, et al. 2017. BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics* 33:1930–36
85. Min X, Zeng W, Chen S, Chen N, Chen T, Jiang R. 2017. Predicting enhancers with deep convolutional neural networks. *BMC Bioinform.* 18:478
86. Liu F, Li H, Ren C, Bo X, Shu W. 2016. PEDLA: predicting enhancers with a deep learning-based algorithmic framework. *Sci. Rep.* 6:28517
87. Bu H, Gan Y, Wang Y, Zhou S, Guan J. 2017. A new method for enhancer prediction based on deep belief network. *BMC Bioinform.* 18:418
88. Wheeler EC, Van Nostrand EL, Yeo GW. 2018. Advances and challenges in the detection of transcriptome-wide protein–RNA interactions. *Wiley Interdiscip. Rev. RNA* 9:e1436
89. Pan X, Yang Y, Xia CQ, Mirza AH, Shen HB. 2019. Recent methodology progress of deep learning for RNA–protein interaction prediction. *Wiley Interdiscip. Rev. RNA* 10(6):e1544
90. Pan X, Shen HB. 2017. RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. *BMC Bioinform.* 18:136
91. Avsec Ž, Barekatain M, Cheng J, Gagneur J. 2017. Modeling positional effects of regulatory sequences with spline transformations increases prediction accuracy of deep neural networks. *Bioinformatics* 34:1261–69
92. Pan X, Yan J. 2017. Attention based convolutional neural network for predicting RNA-protein binding sites. arXiv:1712.02270 [q-bio.GN]
93. Pan X, Shen HB. 2018. Learning distributed representations of RNA sequences and its application for predicting RNA-protein binding sites with a convolutional neural network. *Neurocomputing* 305:51–58
94. Pan X, Rijnbeek P, Yan J, Shen HB. 2018. Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC Genom.* 19:511
95. Lorenz R, Bernhart SH, zu Siederdissen CH, Tafer H, Flamm C, et al. 2011. ViennaRNA package 2.0. *Algorithms Mol. Biol.* 6:26
96. Steffen P, Voß B, Rehmsmeier M, Reeder J, Giegerich R. 2005. RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics* 22:500–3
97. Maticzka D, Lange SJ, Costa F, Backofen R. 2014. GraphProt: modeling binding preferences of RNA-binding proteins. *Genome Biol.* 15:R17
98. Budach S, Marsico A. 2018. pysster: classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. *Bioinformatics* 34:3035–37
99. Chakrabarti AM, Haberman N, Praznik A, Luscombe NM, Ule J. 2018. Data science issues in studying protein–RNA interactions with CLIP technologies. *Annu. Rev. Biomed. Data Sci.* 1:235–61

100. Gandhi S, Lee LJ, Delong A, Duvenaud D, Frey B. 2018. cDeepbind: a context sensitive deep learning model of RNA-protein binding. bioRxiv 345140. **https://doi.org/10.1101/345140**
101. Orenstein Y, Wang Y, Berger B. 2016. RCK: accurate and efficient inference of sequence- and structure-based protein-RNA binding models from RNAcompete data. *Bioinformatics* 32:i351–59
102. Li Z, Zhu J, Xu X, Yao Y. 2019. RDense: a protein-RNA binding prediction model based on bidirectional recurrent neural network and densely connected convolutional networks. *IEEE Access* 8:14588–605
103. Livi CM, Blanzieri E. 2014. Protein-specific prediction of mRNA binding using RNA sequences, binding motifs and predicted secondary structures. *BMC Bioinform*. 15:123
104. Pan X, Fan YX, Yan J, Shen HB. 2016. IPMiner: hidden ncRNA-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction. *BMC Genom*. 17:582
105. Zhang S, Zhou J, Hu H, Gong H, Chen L, et al. 2015. A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Res*. 44:e32
106. Su Y, Luo Y, Zhao X, Liu Y, Peng J. 2019. Integrating thermodynamic and sequence contexts improves protein-RNA binding prediction. *PLOS Comput. Biol.* 15:e1007283
107. Eraslan G, Avsec Ž, Gagneur J, Theis FJ. 2019. Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* 20:389–403
108. Wainberg M, Merico D, Delong A, Frey BJ. 2018. Deep learning in biomedicine. *Nat. Biotechnol.* 36:829–38
109. Zou J, Huss M, Abid A, Mohammadi P, Torkamani A, Telenti A. 2018. A primer on deep learning in genomics. *Nat. Genet.* 51:12–18
110. Lan K, Wang D-T, Fong S, Liu L-S, Wong KK, Dey N. 2018. A survey of data mining and deep learning in bioinformatics. *J. Med. Syst.* 42:139
111. Cao C, Liu F, Tan H, Song D, Shu W, et al. 2018. Deep learning and its applications in biomedicine. *Genom. Proteom. Bioinform.* 16:17–32
112. Peng L, Peng M, Liao B, Huang G, Li W, Xie D. 2018. The advances and challenges of deep learning application in biological big data processing. *Curr. Bioinform.* 13:352–59

Annual Review of Biomedical Data Science

Volume 3, 2020

Contents

Errata

An online log of corrections to *Annual Review of Biomedical Data Science* articles may be found at http://www.annualreviews.org/errata/biodatasci