Bioinformatics, 34, 2018, i638–i646
doi: 10.1093/bioinformatics/bty600
ECCB 2018

A deep neural network approach for learning intrinsic protein-RNA binding preferences

Ilan Ben-Bassat[1], Benny Chor[1] and Yaron Orenstein[2,*]

[1]Blavatnik School of Computer Science, Tel-Aviv University, Tel-Aviv 6997801, Israel and [2]Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva POB 653, Israel

*To whom correspondence should be addressed.

Abstract

Motivation: The complexes formed by binding of proteins to RNAs play key roles in many biological processes, such as splicing, gene expression regulation, translation and viral replication. Understanding protein-RNA binding may thus provide important insights to the functionality and dynamics of many cellular processes. This has sparked substantial interest in exploring protein-RNA binding experimentally, and predicting it computationally. The key computational challenge is to efficiently and accurately infer protein-RNA binding models that will enable prediction of novel protein-RNA interactions to additional transcripts of interest.
Results: We developed DLPRB (Deep Learning for Protein-RNA Binding), a new deep neural network (DNN) approach for learning intrinsic protein-RNA binding preferences and predicting novel interactions. We present two different network architectures: a convolutional neural network (CNN), and a recurrent neural network (RNN). The novelty of our network hinges upon two key aspects: (i) the joint analysis of both RNA sequence and structure, which is represented as a probability vector of different RNA structural contexts; (ii) novel features in the architecture of the networks, such as the application of RNNs to RNA-binding prediction, and the combination of hundreds of variable-length filters in the CNN. Our results in inferring accurate RNA-binding models from high-throughput *in vitro* data exhibit substantial improvements, compared to all previous approaches for protein-RNA binding prediction (both DNN and non-DNN based). A more modest, yet statistically significant, improvement is achieved for *in vivo* binding prediction. When incorporating experimentally-measured RNA structure, compared to predicted one, the improvement on *in vivo* data increases. By visualizing the binding specificities, we can gain biological insights underlying the mechanism of protein RNA-binding.
Availability and implementation: The source code is publicly available at https://github.com/ilanbb/dlprb.
Contact: yaronore@bgu.ac.il
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The application of neural networks for machine learning (ML) purposes dates back to Rosenblatt's perceptron (Rosenblatt, 1958) and to Minsky and Papert's celebrated book on the topic (Minsky and Papert, 2017). Classical learning methods often face difficulties in processing raw data due to the need for manually designed features. In contrast, deep learning approaches can discover effective features directly from the data, and circumvent the labor-intensive phase of feature engineering. However, for over five decades, neural networks were not the method of choice in ML, as they were outperformed by a number of alternative approaches. The availability of powerful computer hardware with a large number of fast processors (such as GPUs), combined with abundant training data, has recently made deep neural networks (DNNs) the top performer in numerous ML applications. Notable areas of success include computer vision (Karayev *et al.*, 2013; Krizhevsky *et al.*, 2012; Szegedy *et al.*, 2015), natural language processing (Bowman *et al.*, 2015; Sutskever *et al.*, 2014), complex board games, such as GO (Silver *et al.*, 2016) and more.

In most areas mentioned above, hundreds of research projects utilizing DNNs were carried out and published. In computational biology the numbers are lower, yet are catching up (Angermueller *et al.*, 2016). Known applications include gene and splicing regulation (Kelley *et al.*, 2016; Leung *et al.*, 2014; Zhou and Troyanskaya, 2015), DNA methylation (Vidaki *et al.*, 2017), protein classification (Asgari and Mofrad, 2015), and various tasks in biological image analysis (Bar *et al.*, 2015; de Brebisson and Montana, 2015). Of specific relavance to our work, applications to protein-RNA binding prediction were also developed, e.g., DeepBind and iDeep (Alipanahi *et al.*, 2015; Pan and Shen, 2017).

The central role of protein-RNA binding in numerous biological contexts (König *et al.*, 2012) makes it an important area of study to both experimentalists and machine learning researchers. On the experimental side, high-throughput measurement techniques were developed, both for *in vivo* experiments, and for *in vitro* ones. The CLIP method and its derivatives measure protein-RNA binding *in vivo*, on a transcriptome-wide scale (Darnell, 2010; Hafner *et al.*, 2010; Konig *et al.*, 2011; Van Nostrand *et al.*, 2016). These measurements are adversely effected by a variety of orthogonal cellular events, resulting in a non-negligible noise-to-signal ratio. As a consequence, these experiments are not accurate enough to produce reliable quantitative outcomes. Instead, they produce a binary outcome: yes (existence of a binding) or no (lack thereof). In every CLIP experiment, the bindings of one protein to each of its occupied transcripts *in vivo* is determined at a resolution of around 100 nucleotides. The complexity of the *in vivo* environment in the context of protein binding, as well as technological artifacts and experimental noise, make the learning of intrinsic protein-RNA binding preferences from such data a difficult challenge (Kishore *et al.*, 2011; Orenstein *et al.*, 2016b).

A different experimental technique, RNAcompete, works *in vitro* (Cook *et al.*, 2017; Lambert *et al.*, 2014; Ray *et al.*, 2017). In each RNAcompete experiment, the bindings of one protein to around 240 000 short synthetic RNAs (30-40 nucleotides long) are measured. Lacking interfering cellular processes, these experiments exhibit a low noise-to-signal ratio, and are accurate enough to produce good measurements of the bindings specificities, or strengths. The most comprehensive *in vitro* dataset measured using the RNAcompete technology (Ray *et al.*, 2013) contains 244 such experiments (each one on a single protein).

The computational challenge that arises from these experimental data is to infer protein-specific RNA-binding models that will enable prediction of the binding between the given protein and a new RNA transcript. Several methods have been developed to tackle this challenge. All the computational methods receive as input the RNA sequence. Some also receive the secondary structure of the RNA. We remark that the secondary structure is typically predicted by computational means, based on the sequence itself. For short RNA sequences, computational structure prediction is known to be quite accurate (Doshi *et al.*, 2004). Datasets that include both RNA-binding and RNA structure measurements on the same cells are currently available for only two proteins (Spitale *et al.*, 2015).

The first computational method, MEMERIS, used expectation-maximization algorithm to look for sequence motifs in RNA regions that are more likely to be unpaired, and thus available for binding (Hiller *et al.*, 2006). RNAcontext, developed with the RNAcompete technology, learns a simple model for sequence and structure binding preferences (Kazan *et al.*, 2010). The sequence preferences are represented as a position weight matrix, namely how each position in the binding site contributes to the binding, independently of others. The structure preferences are represented as a vector of the preferences to each structural context. A more recent approach, GraphProt, uses a graph representation of RNA structure to find enriched local sub-graphs to model the sequence and structure binding preferences (Maticzka *et al.*, 2014). However, GraphProt takes more than seven days to run on a single RNAcompete experiment (Orenstein *et al.*, 2016a). DeepBind, a new method based on deep learning, uses a convolutional neural network (CNN) to learn and predict protein-DNA and protein-RNA binding from many datasets, including RNAcompete and CLIP. It is based on the RNA sequence alone, i.e., without considering RNA structure (Alipanahi *et al.*, 2015). The most recent development and the state of the art, RCK, extends RNAcontext by using a *k*-mer based model, on both the sequence- and the structure-level (Orenstein *et al.*, 2016a). It assigns a binding score to each RNA word of length *k* under each structural context, and thus can capture position-dependence inside a binding site. iDeep tackles the problem of predicting *in vivo* binding based on several data sources representing the complexity of the cellular environment. It receives protein binding preferences as part of the input (Pan and Shen, 2017), and thus solves a different problem. Deepnet-RBP learns RNA-binding preferences based on deep learning and using both RNA secondary and tertiary structures, but was designed to learn it from *in vivo* data only (Zhang *et al.*, 2016). The latest method based on deep learning, pysster, considers only one RNA structure per sequence (Budach and Marsico, 2018). Moreover, it solves the sequence classification problem, and does not predict binding intensities when possible. Today, no study exploited the most advanced machine learning technique to learn intrinsic protein-RNA sequence and structure binding preferences from quantitative high-throughput *in vitro* data.

In this work, we introduce DLPRB, a Deep neural network approach for Learning Protein-RNA Binding preferences. DLPRB employs two DNN architectures: a convolutional neural network, and a recurrent neural network (RNN). CNNs (LeCun *et al.*, 1998) are known to have good performance in analyzing spatial information. RNNs process input data in a sequential manner. They exploit temporal dependencies in the data, mostly by using either long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997), or gated recurrent units (GRU) (Cho *et al.*, 2014). For both the *in vitro* case (predicting binding intensity) and the *in vivo* one (classification), our results exhibit a significant improvement over all previous predictors on the same set of benchmark experiments. This comparison includes DeepBind, which employed a CNN as well, but with substantially fewer filters and without taking RNA secondary structure into account. For *in vitro* data, our RNN achieved an average Pearson correlation of 0.628 (predicted vs. actual intensities), as compared to 0.46 by the state of the art RCK, and the runner up DeepBind with 0.41. For two different *in vivo* datasets, our CNN achieved a median AUC of 0.657 and 0.809. These results are better than the top performer DeepBind, with 0.648 and 0.803, respectively, and the improvement is statistically significant since our approach performs better in almost all the experiments. When using experimentally-measured RNA structure as opposed to predicted one, *in vivo* binding prediction improves even further.

The remainder of this paper is organized as follows: Section 2 describes the datasets used and the methods employed, including a description of the new RNN and CNN architectures we constructed. Section 3 presents the results of running our DNNs, and of visualizing specificities of binding sites. Finally, Section 4 contains concluding remarks and open problems.

2 Materials and methods

2.1 RNA secondary structure prediction

RNA secondary structural context profiles were predicted using a variant of RNAplfold (Lorenz *et al.*, 2011). In this variant, probabilities for four structural contexts are calculated per position: hairpin loop, inner loop, multi loop and external region (Kazan *et al.*, 2010). The probability for a position being paired is assigned, so that the total sum is 1. The probabilities were represented as vectors of length five, one for each position in the sequence. They were provided with the sequences as input to RCK, RNAcontext and our DNNs (DeepBind does not have such optional input).

2.2 *In vitro* binding prediction evaluation

To evaluate the performance of the algorithms for *in vitro* binding prediction, we used the RNAcompete dataset (Ray *et al.*, 2013). The dataset includes 244 experiments, each containing the binding intensities between a single protein and more than 240 000 RNA sequences. The set of sequences was designed as a union of two sets, A and B, such that each has similar 9-mer coverage. For each experiment, we trained a model on sequences from set A and predicted the intensities on set B. Performance was determined by the Pearson correlation of predicted and measured intensities of set B. Outlier intensities were clamped as done in the DeepBind study (Alipanahi *et al.*, 2015): all intensities above the 0.5 percentile were clamped to the value of the 0.5 percentile. Three methods were compared in this evaluation: RNAcontext, RCK and DeepBind, using results taken from (Orenstein *et al.*, 2016a).

To test how well DLPRB approach performs, we computed the Pearson correlation for all pairs of replicate experiments. For every pair of experiments measuring the binding of the same protein, i.e, identical amino acid sequence, the Pearson correlation of the results is an upper bound for any binding prediction algorithm. In the RNAcompete dataset there are 46 such pairs. We then compared these correlation scores with the results achieved by our algorithm.

2.3 *In vivo* binding prediction evaluation

For *in vivo* binding prediction, we used eCLIP experiments (Van Nostrand *et al.*, 2016), whose proteins overlap the RNAcompete dataset. There are 21 proteins in the overlap between these two datasets. These proteins were covered by 36 RNAcompete experiments and by 54 eCLIP experiments, forming a set of 94 experimental pairs covering 21 different proteins. For each eCLIP experiment, the bound peaks were used as positive sequences, and regions 300 nt downstream were used as controls, resulting in an equal-sized control set. The nearby regions were selected to test how well the binding model distinguishes between different regions on the same RNA transcript that are available for binding, while only one of them is bound. Structure prediction was performed using RNAplfold, together with 150 nucleotides flanking regions (as in previous studies, Maticzka *et al.*, 2014; Orenstein *et al.*, 2016a), and only the original sequence peaks were used for prediction. Performance was gauged by area under the ROC curve, which is appropriate for balanced positive and negative sets, as in our case. Each binding model is trained on a complete RNAcompete experiment, and tested on its paired eCLIP experiment.

Similarly, we gauged the performance of our networks in predicting *in vivo* binding using an older dataset taken from the GraphProt study (Maticzka *et al.*, 2014). This dataset includes 23 CLIP experiments, where the intersection with RNAcompete data covers 10 proteins.

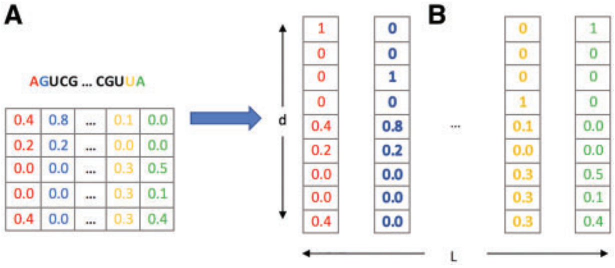

Fig. 1. Data representation. (**A**) The raw input data. Each sample is an RNA sequence and a structure probabilities matrix. (**B**) Data transformation. Each position in the sample is represented as a concatenation of a one-hot encoding vector representing the nucleotide, and a structure probabilities vector

2.4 Data representation

Our deep learning networks receive two types of data as input, instead of a single one (Fig. 1A). An RNA sequence of length ℓ is a string of ℓ nucleotides over the alphabet $\Sigma = \{A, G, C, U\}$. We encode every nucleotide as a one-hot vector of dimension $d_1 = 4$. RNA structural information is encoded in a matrix $S \in \mathbb{R}^{d_2 \times \ell}$, where d_2 denotes the number of possible structural contexts. In this paper we consider $d_2 = 5$ possible structural contexts.

The information on each position in the sequence is encoded in one vector of dimension $d = d_1 + d_2$ (Fig. 1B). Namely, every position is encoded by a concatenation of the one-hot encoding vector of the current nucleotide, and the vector of structural contexts probabilities. The method handles variable-length sequences by encoding every sequence using L vectors, where L denotes the maximum possible length of a sequence. Shorter sequences are zero-padded so that all sequences have the same length.

2.5 DLPRB: convolutional and recurrent neural networks for RNA-binding prediction

The proposed CNN for predicting binding intensities receives L vectors for each RNA sequence, as described in Section 2.4. It constructs an input matrix, $M \in \mathbb{R}^{L \times d}$, whose rows are the vectors representing the nucleotides and structure probabilities. The input matrix, M, is then fed into the convolutional neural network.

Figure 2 illustrates the architecture of our CNN. The first layer of the network is a convolutional layer, which applies a series of filters on the input matrices. A filter is a weight matrix $F \in \mathbb{R}^{m \times d}$, where d is the dimension of the vectors representing the nucleotides and structure probabilities, and m is the filter length. As it is sliding, or convolving, over the input, the network computes an elementwise multiplication of the filter with all possible consecutive submatrices $W \in \mathbb{R}^{m \times d}$ of the input data, with an addition of a bias b. We use a rectifier $f(x) = max(0, x)$ as a non-linear activation function on the convolution output. The network utilizes multiple filters, with several possible values of m. The max-pooling layer scans the output vector of each filter and chooses the maximum value in it. A fully-connected layer computes a weighted sum of the maximum values found in the previous layer. Its output is a hidden layer of size 128. A second fully connected layer computes the final outcome of the network. The filters, which capture local patterns on the sequence- and structure-level, are equivalent to position weight matrices, which are very popular in modeling protein binding preferences (Stormo, 2000). We use 256 filters, 128 of length 5 and 128 of length 11. The number of filters and their lengths are different than the ones used in DeepBind.

Given the actual binding intensities from an RNAcompete experiment, the network is trained with a mini-batch Adam optimization algorithm (Kingma and Ba, 2014), using 128 samples in each batch, a

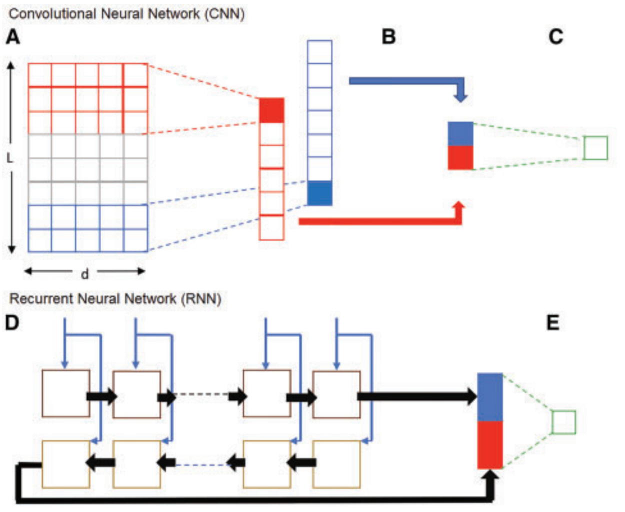
Convolutional Neural Network (CNN)
A
B
C
L
d
Recurrent Neural Network (RNN)
D
E

Fig. 2. Our deep neural network architectures. (**A**) Convolutional layer, including a non-linear activation function. The filters are applied on a matrix whose rows are the input vectors. The output vectors of two filters are shown: one red of length three, and one blue of length two. Two specific applications of the filters are marked using dotted lines. (**B**) Max-pooling layer. The colored rectangles contain the max values in each vector. (**C**) Fully-connected layer that computes an intensity prediction. A second fully-connected layer is used in the actual implementation of the network. (**D**) Bidirectional recurrent neural network, composed of LSTM or GRU cells (brown rectangles). In each time stamp *t*, the network receives a *d*-dimensional vector representing the nucleotide at position *t* in the sequence. (**E**) Fully-connected layer that computes an intensity score. A second fully-connected layer is used in the actual implementation of the network

constant learning rate of 0.0001, and a mean squared error (MSE) as a loss function. The filters were initialized with small random weights. L2-regularization term is added to the loss function as well, by adding the sum of the squares of all the weights in the network.

Some of the hyper-parameters of the network were chosen via a grid search on a small subset of the data. We randomly chose 10 out of the 244 RNAcompete experiments, and used them for hyper-parameter tuning. For each of these training sets we marked one third of the samples as a validation set, and trained several models on the rest. We performed a grid search for every experiment, and chose the model which performed best on the corresponding validation set. The value of every hyper-parameter was then determined using a majority voting. The hyper-parameters that were optimized during the grid search process were the number of filters (16, 64, 128 and 256), the filter lengths (combinations of the lengths 5, 8, 11 and 16), the initial learning rate (0.01, 0.001 and 0.0001), and the learning decay rate (0.96, 0.98 and 1).

To reduce overfitting, the number of times the training data is processed during training (also known as the number of epochs) was tuned specifically for every dataset. This was done by splitting each training set into three parts. A three-fold cross validation and an early-stopping procedure were used to derive three candidate values for the number of epochs. The number of epochs for every dataset was then set to be the average of the three candidate values, and the final model was trained using the entire training set.

In order to assess the impact of using structure information in addition to sequence data, we predicted binding intensities with a modified CNN architecture that takes as input the one-hot encoding of the nucleotides without concatenating structure probabilities to it. This modified network was trained and tested on data that contained no structure information at all.

We also tested a bidirectional RNN, which is visually described in Figure 2. RNNs can capture position dependencies, which are known to occur in protein binding sites (Barash *et al.*, 2003). The *L* input vectors are fed into the network in a sequential manner, both forward and backward. We used GRU cells to detect possible long-term dependencies, and set the cell size to 64. The rest of the network layers, as well as the loss function and hyper-parameters tuning method are the same as in the CNN version.

2.6 Evaluating the weight of RNA structure in the sequence and structure binding models

To evaluate the weight of RNA structural information in our DNNs, we ran the prediction of binding while assigning uniform structural probabilities to all positions of the test sequences. This removes any structural information from the test data.

2.7 Comparing experimentally-measured and computationally-predicted RNA structure

To evaluate the effect of using experimentally-measured RNA structure instead of predicted one, we used CLIP and icSHAPE data (Spitale *et al.*, 2015). Probability vectors of experimentally-measured RNA structure and CLIP-seq data were downloaded from the GEO database (accession numbers GSE60034 and GSE64168, respectively). Binding site peaks were extracted as in the original study (Spitale *et al.*, 2015) using a 40 nt window size. We used the same set of peaks and control sequences as in the RCK study (Orenstein *et al.*, 2016a), summing up to 4102 sequences in each category. For computational structure prediction, we flanked binding sites and control sequences by 150 nt on each end, which were only used for structure prediction by RNAplfold (Lorenz *et al.*, 2011). The flanking regions were discarded when testing the trained models on CLIP data.

2.8 Visualizing RNA-binding sequence and structure preferences

One of the main drawbacks of neural networks is their lack of interpretability. However, in some cases we can still infer how the network works, and what important features are extracted from the raw data. Here, we can gain an understanding of how the CNN works by analyzing its filters, similarly to what was done for DeepBind (Alipanahi *et al.*, 2015). A convolution filter works like a motif detector, taking both RNA sequence and structure information into account. After training the model, we ran it to predict binding over all the test data and analyzed the output of the max-pooling layer. Given a filter *F*, we extracted from each test sample the subsequence to which *F* had assigned the highest activation value, along with its structure information. We aligned all the subsequences that passed a certain threshold, and computed a modified position frequency matrix (PFM) that also captures the structure information for every position in the sequence. From this matrix we then generated the sequence and structure logos (Wagih, 2017).

3 Results

3.1 Predicting *in vitro* binding

To gauge the performance of DLPRB, our deep neural networks, compared to extant methods, we used the comprehensive dataset of RNAcompete, which includes 244 experiments (Ray *et al.*, 2013). For each protein, we trained a model on half of the RNA sequences, and tested it on the other half. Performance was gauged by Pearson

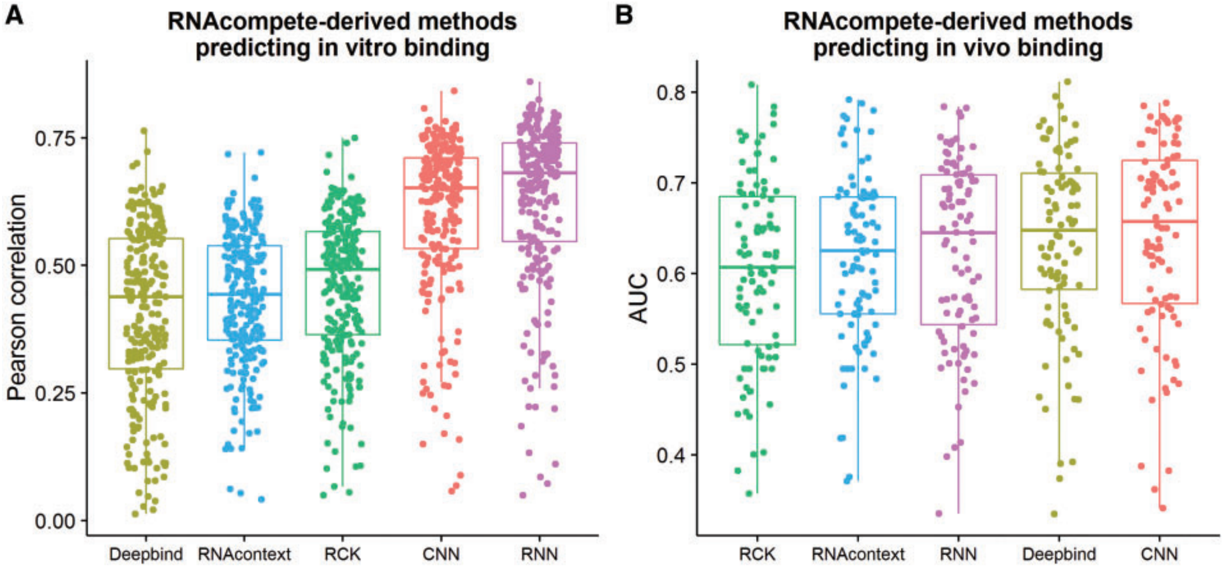

Fig. 3. Performance of RNAcompete-derived methods in binding predictions shown as boxplots for different methods. RNAcontext, RCK, CNN and RNN utilize RNA secondary structure. (**A**) Performance in predicting *in vitro* binding. For each RNAcompete experiment, a model was trained on set A of sequences and tested on set B. Results are based on 244 experiments. Performance gauged by Pearson correlation of predicted and measured intensities. (**B**) Performance in predicting *in vivo* binding. For each RNAcompete experiment, a model was trained on the whole dataset and tested in predicting bound and unbound transcripts as measured by eCLIP experiment on the same protein. Results are based on 94 experiment pairs. Performance gauged by area under the ROC curve

correlation of predicted and measured intensities. For complete details see Section 2.2.

Both our deep neural networks significantly outperformed all methods in *in vitro* binding prediction (Fig. 3A). When comparing them to extant methods, they both outperformed the state of the art RCK, which achieved an average Pearson correlation of 0.46 as compared to 0.606 for our CNN (P-value $= 9.16 \times 10^{-42}$, Wilcoxon rank-sum test). Our RNN outperformed the CNN architecture, achieving an average Pearson correlation of 0.628 (P-value $= 9.53 \times 10^{-26}$). The average Pearson correlation computed on 46 pairs of RNAcompete replicate experiments was 0.581. The result achieved by our RNN on the same set of experiments was 0.578 (P-value = 0.84). This means that our RNN is almost optimal on these experiments, since it captures all non-stochastic information in them.

When training and testing our CNN on RNA sequence alone, we saw a statistically significant difference in prediction accuracy, compared to training and testing on both RNA sequence and structure (average Pearson correlation dropped from 0.606 to 0.592, P-value $= 1.26 \times 10^{-24}$). As was the case in (Orenstein *et al.*, 2016a), this shows that structure information can improve binding prediction. We speculate that the relatively small difference is due to the fact that RNA structure was predicted from sequence, and DNNs are capable of capturing long-range nucleotide interactions, which are the basis of RNA secondary structure. Moreover, the library design of RNAcompete technology was designed to be unstructured, and as a consequence, contains very few structures (Ray *et al.*, 2013, 2009). Thus, it is no surprise that the sequence features alone can capture most of the structural information. Still, the statistically significant difference shows the importance of RNA structure in protein-RNA binding prediction.

As the gap between our CNN and the network used by DeepBind was not fully explained by removing RNA structural information, we ran a variant of our approach with much fewer convolution filters. Instead of using 256 filters of variable lengths, we used only 16 filters, each of length 16, imitating the configuration used in (Alipanahi *et al.*, 2015). The results dropped profoundly to an average Pearson correlation of 0.523 compared to 0.592, just by decreasing the number of filters and changing their lengths (P-value$=4.69 \times 10^{-41}$). We deduce that our advantage over DeepBind can be explained mainly by either the configuration of the convolution filters, or by the RNN architecture. In addition, RNA structural information also improves the accuracy of our prediction. For complete results see Supplementary Table S1.

3.2 Predicting *in vivo* binding

To gauge the performance of DLPRB, our neural networks, compared to extant methods on *in vivo* binding prediction, we used the eCLIP dataset (Van Nostrand *et al.*, 2016). The overlap with RNAcompete dataset covers 21 proteins by 94 experimental pairs involving 36 RNAcompete and 54 eCLIP experiments (Ray *et al.*, 2013). Each binding model was trained on a complete RNAcompete experiment, and tested on its paired eCLIP experiment. We report the performance in predicting *in vivo* binding by AUC, an appropriate metric for balanced bound and unbound sets as in our case. Similarly, we tested our method on an older CLIP dataset taken from the GraphProt study (Maticzka *et al.*, 2014). The overlap with RNAcompete covers 10 proteins. For complete details see Section 2.3.

The results of predicting *in vivo* binding show that our CNN performs the best, achieving a median AUC of 0.657, compared to 0.648 and 0.645 for DeepBind and RNN, respectively (Fig. 3B). In a pairwise comparison CNN is significantly better than DeepBind and RNN (P-values < 0.0001, Wilcoxon rank-sum test). When tested on an older dataset, our CNN network outperformed all other methods achieving a median AUC of 0.809, compared to 0.803 and 0.782 for DeepBind and RNN, respectively (Fig. 4A). This improvement is not statistically significant. However, there were only 10 proteins in the overlap with RNAcompete for this dataset.

Two reasons may hamper the accuracy of *in vitro* models in predicting *in vivo* binding. First, *in vivo* data is known to be noisy and suffer from experimental biases (Kishore *et al.*, 2011). Moreover, RNA

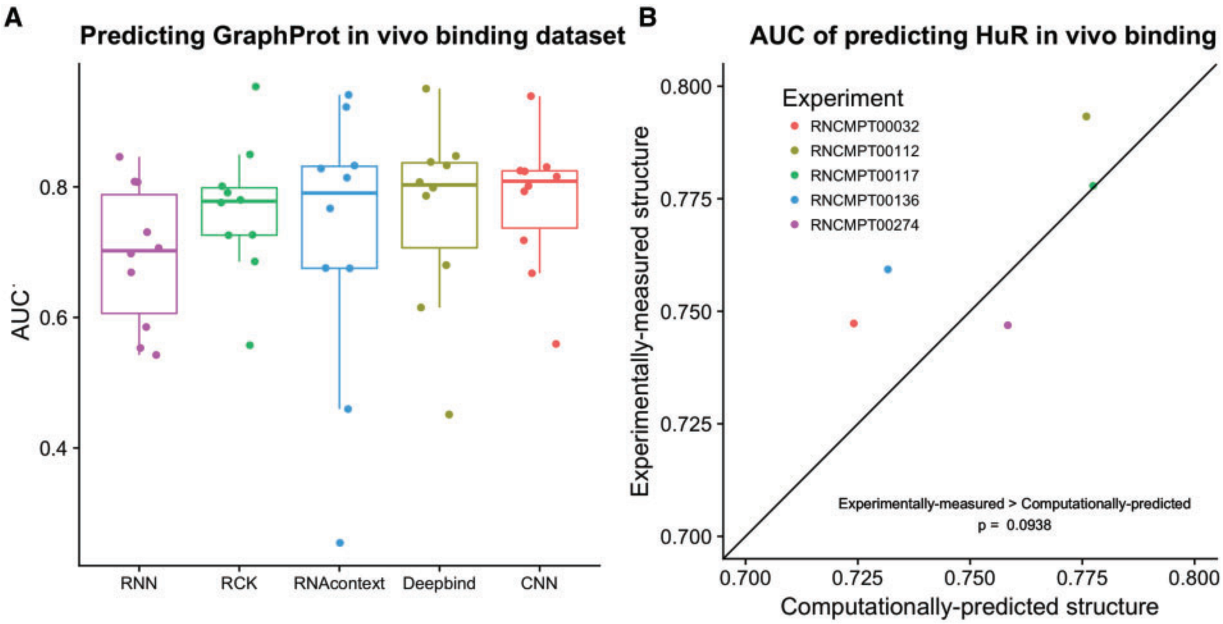

Fig. 4. Improved *in vivo* binding prediction. (**A**) We used GraphProt dataset of *in vivo* binding to gauge prediction accuracy. For each pair of RNAcompete and CLIP experiments on the same protein, a model was trained on the former and tested on the latter. Twenty-three pairs overlap with the GraphProt study and RNAcompete dataset, covering 10 proteins in 21 RNAcompete and 12 CLIP experiments. Performance per protein is gauged by average AUC. (**B**) Comparison of experimentally-measured and computationally-predicted RNA structure. We used available CLIP and iCSHAPE datasets that had an RNAcompete experiment available for the same protein. The only protein tested was HuR, which had five corresponding RNAcompete experiments. A model trained on RNAcompete data had better binding prediction with experimentally-measured RNA structure than with computationally-predicted structure

structure prediction is less accurate *in vivo* than *in vitro* (Rouskin *et al.*, 2014), so learned structural preferences may not improve binding prediction. At this stage, more datasets with higher quality are needed in the overlap between CLIP and RNAcompete to derive more definitive conclusions. For complete results see Supplementary Table S2.

3.3 Experimentally-measured structure may improve *in vivo* binding prediction

Following the results in the previous subsection, we examined the reason why RNA structural information did not improve *in vivo* binding prediction. We speculated that RNA structure prediction of long RNA transcripts *in vivo* is inaccurate. To test this hypothesis, we compared computationally-predicted RNA structure (Lorenz *et al.*, 2011) with experimentally-measured one (Spitale *et al.*, 2015) in the task of binding prediction. To demonstrate the effect of using experimental probabilities, we used available CLIP and icSHAPE experiments, performed on the same cells that also had an RNAcompete experiment on the same protein. Unfortunately, only the HuR protein, which had five RNAcompete experiments, was found to overlap. Since icSHAPE reports only unpaired probabilities, we trained a model based on two structural contexts: paired and unpaired. Performance was measured by AUC in predicting HuR binding sites. For complete details see Section 2.7.

Results show that our neural networks benefit from experimentally-measured RNA structure in predicting *in vivo* binding (Fig. 4B). Predictions using icSHAPE measurements are more accurate than predictions using predicted structure in four out of five experiments. We note that additional experimental measurements of RNA structure and protein-RNA binding on the same cells are needed to evaluate the benefit of experimentally-measured RNA structure for the task of *in vivo* binding prediction. For complete results see Supplementary Table S4.

3.4 The weight of RNA structure in the sequence and structure binding models

We gauged the weight of RNA structural preferences in binding prediction using our CNN architecture. We employed a comprehensive dataset of both *in vitro* and *in vivo* data as in previous sections. For each test set we used the same models, trained on both RNA sequence and structure data, but we now predicted binding using uniform structure probabilities, and compared them to predictions using predicted structure probabilities. As we already noted in Section 3.1, when training and testing on RNA sequence alone, the prediction accuracy is slightly, albeit significantly, lower compared to the one achieved by using RNA structure on top of sequence. Despite the fact that RNAcompete sequence set was designed to be unstructured, previous studies have shown that some structure exists and that RNA structural binding preferences can still be inferred from the data (Orenstein *et al.*, 2016a; Ray *et al.*, 2013). For complete details see Section 2.6.

We found that the use of predicted structure probabilities in test time significantly improves prediction performance *in vitro*, but not *in vivo*. In terms of *in vitro* binding, the improvement is across the board: for every single dataset, the performance improved by using predicted structure probabilities as compared to uniform ones (Fig. 5A). The improvement was striking, from an average Pearson correlation of 0.54 for sequence-only mode, to 0.608 when using predicted structure probabilities (P-value $= 4.53 \times 10^{-42}$, Wilcoxon rank-sum test). The results of the *in vivo* data, on the other hand, did not show significant improvement (P-value $= 0.96$) (Fig. 5B). We can see several reasons for this dichotomy. The *in vivo* dataset is smaller. It contains only 94 pairs

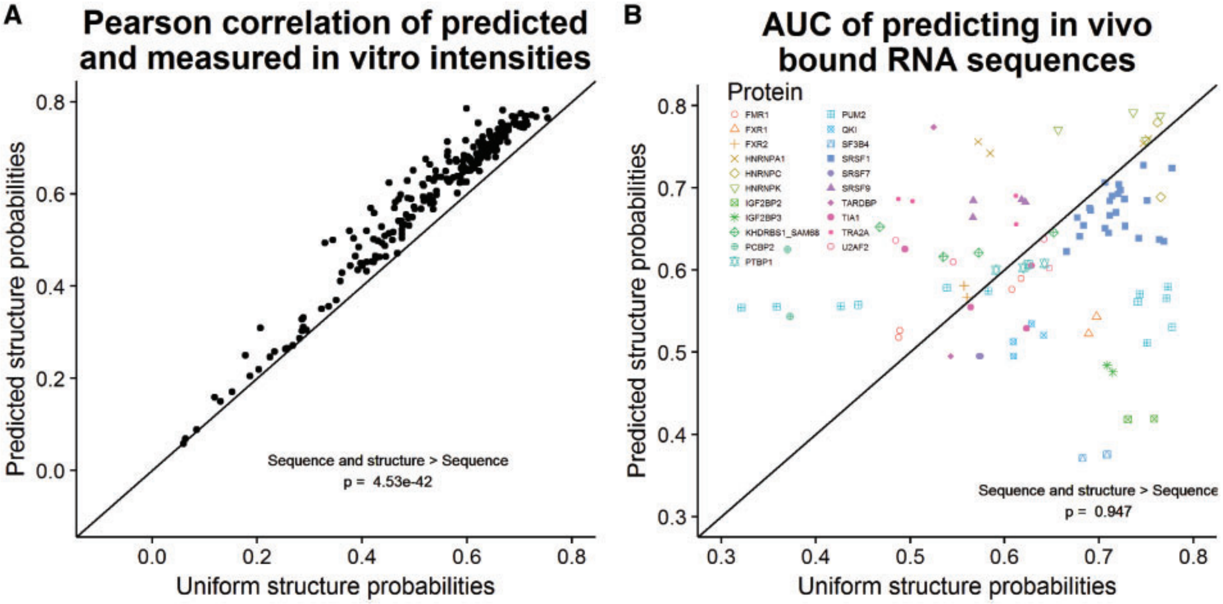

Fig. 5. The weight of RNA structure on top of sequence in binding prediction. (**A**) Predicted RNA structure probabilities improve *in vitro* binding prediction compared to uniform ones. Correlation results over 244 paired experiments uncover that RNA structure plays a significant role in protein-RNA interactions. (**B**) Predicted RNA structure probabilities do not improve *in vivo* binding prediction compared to uniform ones. AUC results of 96 paired eCLIP and RNAcompete experiments over 21 joint proteins demonstrate that RNA structure is not accurately predicted for *in vivo* transcripts, and that protein-intrinsic binding preferences do not capture the full complexity of the cellular environment

compared to 244, covering only 21 proteins compared to 205. The *in vivo* experiments are noisier and prone to technological artifacts (Kishore *et al.*, 2011). The *in vivo* environment contains many confounding factors, which are not part of the binding model, and thus may decrease prediction accuracy. Lastly, RNA secondary structure is less accurate for long sequences *in vivo* than for short sequences and *in vitro* (Rouskin *et al.*, 2014). For complete results see Supplementary Table S3.

3.5 Visualizing RNA-binding specificities

Finally, we wanted to learn new biological insights on the RNA sequence and structure binding preferences of the proteins in the RNAcompete dataset. Interpreting deep convolutional neural networks is a long-standing challenge which we do not solve in this study. Instead, we developed a heuristic. We looked for hits of the motif detector, i.e., binding sites that passed a certain threshold, and used their alignment to generate a position frequency matrix for both the RNA sequence and RNA structure probabilities. We drew these as sequence logos (Wagih, 2017). For complete details see Section 2.8.

Figure 6 shows a comparison of sequence logos generated by different methods on different datasets for three proteins: Pum2, Vts1 and HuR. Unfortunately, we could not use available motif databases, such as CIS-BP (Weirauch *et al.*, 2014), and comparison tools, such as TOMTOM (Tanaka *et al.*, 2011), since they hold and handle sequence motifs only. In our comparison, we see a high concordance in the sequence and structure preferences as discovered by GraphProt (Maticzka *et al.*, 2014) for Pum2 protein trained on an independent PAR-CLIP dataset (Hafner *et al.*, 2010), and for HuR and Vts1 as discovered by RNAcontext (Kazan *et al.*, 2010) trained on an old version of RNAcompete (Ray *et al.*, 2009). This demonstrates the ability of our CNN to learn true RNA sequence and structure binding preferences and to generate interpretable visualization of them.

4 Discussion

We have shown that carefully designed deep neural networks are capable of significantly improving the predictive power in protein-RNA binding experiments. By using different network architectures, and by incorporating structure information in the learning process, we outperformed the state-of-the-art results for this task. In particular, we have demonstrated the power of recurrent neural networks for the task of RNA-binding prediction. Regarding convolutional neural networks, our architecture benefits from a higher number of convolution filters, as well as from a mixture of different filter lengths. While adding convolution filters improved the prediction accuracy of the network, we did not experience such an improvement when adding more convolutional layers. This coincides with the results of (Zeng *et al.*, 2016), who explored CNNs for predicting protein-DNA binding.

The improvement was substantially noticeable for *in vitro* experiments. This is possibly due to the fact that *in vitro* experiments are designed to quantitatively measure protein-RNA binding for hundreds of thousands of synthetic RNA sequences. We believe these results demonstrate the usefulness of deep neural networks in the area of protein-RNA binding, and more generally in the field of computational biology, where they are starting to be used on a large scale.

A long-standing goal in the field of protein-RNA interaction is accurate prediction of *in vivo* binding. As demonstrated in this study, current computational methods perform rather poorly in predicting bound and unbound RNA transcripts (average AUCs around 0.65, and some predictions are even below 0.5, which corresponds to random guessing). We believe that learning the intrinsic binding preferences of an RNA-binding protein would not suffice in this case, as the *in vivo* environment is much more complex. Not only do proteins compete over the same binding sites or co-bind together, RNA structure also differs between *in silico*, *in vitro* and *in vivo* environments. On top of that, RNAcompete and other *in vitro*

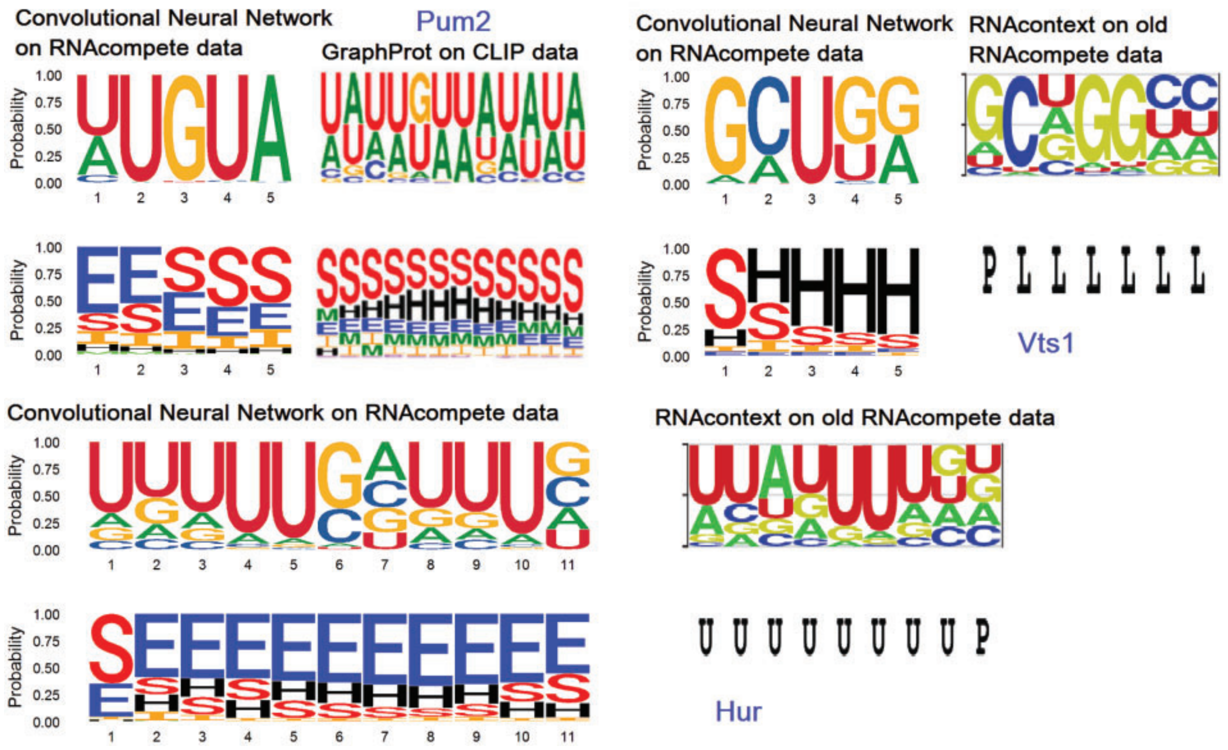

Fig. 6. Sequence and structure binding preference visualization. We visualize the binding preferences, represented as two frequency matrices, sequence and structure, by the sequence logo format. Compared to previous methods, GraphProt (Maticzka *et al.*, 2014) and RNAcontext (Kazan *et al.*, 2010), which were trained on different datasets than those used in our study (Hafner *et al.*, 2010; Ray *et al.*, 2009), we see high concordance in both the sequence and structure preference. U, P, L stand for unpaired, paired and loop. S, H, I, M, E stand for stem (paired), hairpin loop, inner loop, multi loop and external region

experiments measure binding to short RNA sequences (30–40 nt) (Lambert *et al.*, 2014; Ray *et al.*, 2009), which cannot fold to complex RNA structures that are found *in vivo*, where transcripts span thousands of nucleotides. This alone already inhibits *in vitro* trained models from learning binding preferences to complex structures.

There are a number of questions worth pursuing following our work: Why were RNNs better than CNNs for *in vitro* data, but worse than them for *in vivo* data? Our training and test data were based on experiments where the binding between a single protein and numerous RNAs was measured. Can we design a DNN (or another ML mechanism) to train on many proteins and RNAs, and then to predict the binding of *different* proteins and RNAs? Another future line of research is to further improve the interpretability of the suggested networks. In particular, a better understanding of how the structure information is incorporated in the learning and prediction processes, and what filters are more dominant and why, may yield interesting biological insights.

Acknowledgement

The work of Ilan Ben-Bassat is part of Ph.D. thesis research conducted at Tel-Aviv University.

Funding

This work was supported by fellowships from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University, the Blavatnik Research Fund, and the Blavatnik Interdisciplinary Cyber Research Center in Tel-Aviv University.

Conflict of Interest: none declared.

References

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

Angermueller,C. *et al.* (2016) Deep learning for computational biology. *Mol. Syst. Biol.*, **12**, 878.

Asgari,E. and Mofrad,M.R. (2015) Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS One*, **10**, e0141287.

Bar,Y. *et al.* (2015). Deep learning with non-medical training used for chest pathology identification. *Proc. SPIE*, **941**, 94140V.

Barash,Y. *et al.* (2003). Modeling dependencies in protein-DNA binding sites. In *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology*, ACM, Berlin, Germany, pp. 28–37.

Bowman,S.R. *et al.* (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv: 1508.05326*.

Budach,S. and Marsico,A. (2018) pysster: classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. *Bioinformatics*, **1**, 3.

Cho,K., V. *et al.* (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv Preprint arXiv: 1406.1078*,

Cook,K.B. *et al.* (2017) RNAcompete-S: combined RNA sequence/structure preferences for RNA binding proteins derived from a single-step in vitro selection. *Methods*, **126**, 18–28.

Darnell,R.B. (2010) HITS-CLIP: panoramic views of protein-RNA regulation in living cells. *WIREs RNA*, **1**, 266–286.

de Brebisson,A. and Montana,G. (2015). Deep neural networks for anatomical brain segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28. Boston, MA.

Doshi,K.J. *et al.* (2004) Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. *BMC Bioinformatics*, **5**, 105.

Hafner,M. *et al.* (2010) Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, **141**, 129–141.

Hiller,M. *et al.* (2006) Using RNA secondary structures to guide sequence motif finding towards single-stranded regions. *Nucleic Acids Res.*, **34**, e117–e117.

Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.

Karayev,S. *et al.* (2013) Recognizing image style. *arXiv Preprint arXiv: 1311.3715*,

Kazan,H. *et al.* (2010) RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins. *PLoS Comput. Biol.*, **6**, e1000832.

Kelley,D.R. *et al.* (2016) Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.*, **26**, 990–999.

Kingma,D., and Ba,J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv: 1412.6980.*

Kishore,S. *et al.* (2011) A quantitative analysis of CLIP methods for identifying binding sites of RNA-binding proteins. *Nature Methods*, **8**, 559–564.

Konig,J. *et al.* (2011) iCLIP-transcriptome-wide mapping of protein-RNA interactions with individual nucleotide resolution. *J. Vis. Exp.*, **50**, 2638.

König,J. *et al.* (2012) Protein-RNA interactions: new genomic technologies and perspectives. *Nat. Rev. Genet.*, **13**, 77.

Krizhevsky,A. *et al.* (2012) Imagenet classification with deep convolutional neural networks. In: *Advnaces in Neural Information Processing Systems*, Lake Tahoe, Nevada, pp. 1097–1105.

Lambert,N. *et al.* (2014) RNA Bind-n-Seq: quantitative assessment of the sequence and structural binding specificity of RNA binding proteins. *Mol. Cell*, **54**, 887–900.

LeCun,Y. *et al.* (1998) Gradient-based learning applied to document recognition. *Proc. IEEE*, **86**, 2278–2324.

Leung,M.K. *et al.* (2014) Deep learning of the tissue-regulated splicing code. *Bioinformatics*, **30**, i121–i129.

Lorenz,R. *et al.* (2011) ViennaRNA package 2.0. *Algorithm. Mol. Biol.*, **6**, 26.

Maticzka,D. *et al.* (2014) GraphProt: modeling binding preferences of RNA-binding proteins. *Genome Biol.*, **15**, R17.

Minsky,M. and Papert,S. (2017) *Perceptrons: An Introduction to Computational Geometry*. MIT press.

Orenstein,Y. *et al.* (2016) RCK: accurate and efficient inference of sequence-and structure-based protein-RNA binding models from RNAcompete data. *Bioinformatics*, **32**, i351–i359.

Orenstein,Y. *et al.* (2016b) Sequence biases in CLIP experimental data are incorporated in protein RNA-binding models. *bioRxiv*, 075259.

Pan,X., and Shen,H.-B. (2017) RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. *BMC Bioinformatics*, **18**, 136.

Ray,D. *et al.* (2009) Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nat. Biotechnol.*, **27**, 667–670.

Ray,D. *et al.* (2013) A compendium of RNA-binding motifs for decoding gene regulation. *Nature*, **499**, 172.

Ray,D. *et al.* (2017) RNAcompete methodology and application to determine sequence preferences of unconventional RNA-binding proteins. *Methods*, **118-119**, 3–15.

Rosenblatt,F. (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, **65**, 386.

Rouskin,S. *et al.* (2014) Genome-wide probing of RNA structure reveals active unfolding of mRNA structures in vivo. *Nature*, **505**, 701–705.

Silver,D. *et al.* (2016) Mastering the game of Go with deep neural networks and tree search. *Nature*, **529**, 484–489.

Spitale,R.C. *et al.* (2015) Structural imprints in vivo decode RNA regulatory mechanisms. *Nature*, **519**, 486.

Stormo,G.D. (2000) DNA binding sites: representation and discovery. *Bioinformatics*, **16**, 16–23.

Sutskever,I. *et al.* (2014) Sequence to sequence learning with neural networks. *In Advances in Neural Information Processing Systems*, Montreal, Canada, pp. 3104–3112.

Szegedy,C. *et al.* (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9. Boston, MA.

Tanaka,E. *et al.* (2011) Improved similarity scores for comparing motifs. *Bioinformatics*, **27**, 1603–1609.

Van Nostrand,E.L. *et al.* (2016) Robust transcriptome-wide discovery of RNA binding protein binding sites with enhanced CLIP (eCLIP). *Nature Methods*, **13**, 508.

Vidaki,A. *et al.* (2017) DNA methylation-based forensic age prediction using artificial neural networks and next generation sequencing. *Forensic Sci. Int. Genet.*, **28**, 225–236.

Wagih,O. (2017) ggseqlogo: a versatile R package for drawing sequence logos. *Bioinformatics*, **33**, 3645–3647.

Weirauch,M.T. *et al.* (2014) Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, **158**, 1431–1443.

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

Zhang,S. *et al.* (2016) A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Res.*, **44**, e32–e32.

Zhou,J. and Troyanskaya,O.G. (2015) Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, **12**, 931.