Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Manuscript Category

Subject Section

An exact transformation of convolutional kernels enables accurate identification of sequence motifs

Yang Ding[1,2], Jing-Yi Li[1,2], Meng Wang[1], and Ge Gao[1,2*]

[1] State Key Laboratory of Protein and Plant Gene Research, School of Life Sciences, Center for Bioinformatics, Peking University, Beijing 100871, People's Republic of China.

[2] Beijing Advanced Innovation Center for Genomics (ICG), Peking University, Beijing 100871, People's Republic of China.

*To whom correspondence should be addressed.

Abstract

Motivation: The powerful learning ability of a convolutional neural network (CNN) to perform functional classification of DNA/RNA sequences could provide valuable clues for the discovery of underlying biological mechanisms. Currently, however, the only way to interpret the direct application of a convolutional kernel to DNA/RNA sequences is the heuristic construction of a position weight matrix (PWM) from fragments scored highly by that kernel; whether the resulting PWM still performs the sequence classification well is unclear.

Results: We developed a novel kernel-to-PWM transformation whose result is theoretically provable. Specifically, we proved that the log-likelihood of the resulting PWM of any DNA/RNA sequence is exactly the sum of a constant and the convolution of the original kernel on the same sequence. Importantly, we further proved that the resulting PWM demonstrates the same performance, in theory, as the original kernel under popular CNN frameworks. As expected, the transformation rivaled the performance of the heuristic PWMs in terms of sequence classification, whether the discriminative motif was sequence- or structure-conserved. The transformation also faithfully reproduced the output of trained CNN models where the heuristic one failed. These results compelled us to further develop a maximum likelihood estimation of the optimal PWM for each kernel and a back-transformation of predefined PWMs into kernels. These tools can benefit the biological interpretation of kernel signals.

Availability: Python scripts for the transformation from kernel to PWM, the inverted transformation from PWM to kernel, and the maximum likelihood estimation of optimal PWM are available through ftp://ftp.cbi.pku.edu.cn/pub/software/CBI/k2p or https://github.com/gao-lab/kernel-to-PWM .

Contact: gaog@mail.cbi.pku.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Introduction

Effective *ab inito* mining of the sequence motif from large number of sequences is critical to mechanistic delineation of how the sequences exert their functions (Mathelier *et al.*, 2016; Ray *et al.*, 2013; Nawrocki *et al.*, 2015). While classical methods are either computationally expensive and thus limited to low number of sequences only (e.g., MEME (Bailey *et al.*, 2009)), or make strong assumptions on input sequences (e.g. MEME-ChIP (Machanick and Bailey, 2011) assumes that all input sequences must be centered and has equal length, which will greatly bias the result when it handles sequences other than those coming from ChIP-Seq), researchers have discovered recently that convolutional neural networks (CNNs) are exceptionally good at handle such problem fast and accurately (Alipanahi *et al.*, 2015; Angermueller *et al.*, 2017; Quang and Xie, 2016; Zhou and Troyanskaya, 2015). Basically, a CNN uses a series of convolutional kernels as motif detectors to identify potential sequence patterns. Each of these kernels is a matrix which are used to scan the input sequences, being "align"ed to each position. For each such "alignment", a score is computed by convoluting the kernel over the aligned sequence fragment (LeCun *et al.*, 1989; Cotter *et al.*, 2011).The larger the score, the more "similar" the

motif is to the specific sequence fragment it aligns to. Therefore, it is increasingly important to interpret the particular sequence patterns learned by these CNN models.

Most known sequence patterns are described by either position weight matrices (PWMs), which are essentially probabilistically restricted matrices (e.g., JASPAR (Mathelier *et al.*, 2016), CISBP-RNA (Ray *et al.*, 2013)), or by hidden Markov models (e.g., Dfam (Hubley *et al.*, 2016)), or by covariance models (e.g., Rfam (Nawrocki *et al.*, 2015)). Convolutional kernels, which are essentially unrestricted matrices, are thus not directly comparable to these sequence profiles.

Deepbind (Alipanahi *et al.*, 2015) proposed a heuristic transformation to transform kernel to PWM by first stacking the kernel's highly scoring sequence fragments from input sequences and then normalizing nucleotide counts to obtain the final PWM. The exact transformation is biologically intuitive when the underlying signal resembles a fixed sequence. However, we found that such transformation, which tries to maximize the conditional probability of observing highly scoring fragments, fails to preserve the relative importance of kernel elements (see Discussion for more details). Therefore, we proposed here a novel exact kernel-to-PWM transformation to address this problem. We demonstrated, theoretically and empirically, that the transformed PWM is capable of classifying/regressing sequences in exactly the same way as the original kernel. In both simulations and real-world cases, the exact PWMs outperform Deepbind's heuristic PWMs at detecting both sequence- and structure-conserved motifs. The exact transformation, to our best knowledge, also provides the first mathematically sound interpretation of kernels that convolve DNA/RNA sequences directly. A Python implementation of the transformation is freely available online at ftp://ftp.cbi.pku.edu.cn/pub/software/CBI/k2p or https://github.com/gao-lab/kernel-to-PWM .

An exact kernel-to-PWM transformation

Below we will first describe the transformation briefly, then presented the interpretation. Detailed proofs of theorems and corollaries can be found in the Supplementary Notes.

The transformation

The transformation, as illustrated in Figure 1, is described below (with all coordinates one-based; also see Algorithm 1):

(1) Assume that the kernel to be transformed is a 4-by-L matrix W, where L is the length of this kernel. The element of W at the ith row and jth column is denoted as $w_{i,j}$.

(2) Choose an arbitrary base of logarithm, b (b > 1), for the log-likelihood calculation. There's no further restriction on the choice of b.

(3) Flip W along the second (position) axis to obtain the flipped kernel W': $w'_{i,j} = w_{i,L-j+1}$ for all i from {1, 2, 3, 4} and all j from {1, 2, …, L}.

(4) Replace each $w'_{i,j}$ with $b^{w'_{i,j}}$ to obtain the exponentially transformed kernel C; in other words, $c_{i,j} = b^{w'_{i,j}}$ for all I from {1, 2, 3, 4} and all j from {1, 2, …, L}.

(5) Normalize C in a column-wise manner by dividing each column by its sum, resulting in the PWM P(W, b): $(P(W, b))_{i,j} = c_{i,j} / (c_{1,j} + c_{2,j} + c_{3,j} + c_{4,j})$ for all i from {1, 2, 3, 4} and all j from {1, 2, …, L}.

Then by Theorem 1, for any b > 1 and any given sequence X no shorter than W, we have:

$$(\mathbf{X} * \mathbf{W})_j = constant + \log_b \mathrm{Prob}(\mathbf{X}[1:4, j:(j+L-1)] | \mathbf{P}(\mathbf{W}, b)) \quad (1)$$

where * denotes the convolution operator that does not consider border-crossing cases (i.e., W must fall completely within X; see Supplementary Notes for details). In other words, the sequence's convolution by W is exactly the sum of a constant and the log-likelihood of the PWM transformed on X. We also noted that the time complexity of the exact transformation (Algorithm 1) is O(L) and does not depend on the sequence datasets at all, making it very fast and easy to use.

The exact transformation has two immediate usages: kernel interpretation and PWM reuse.

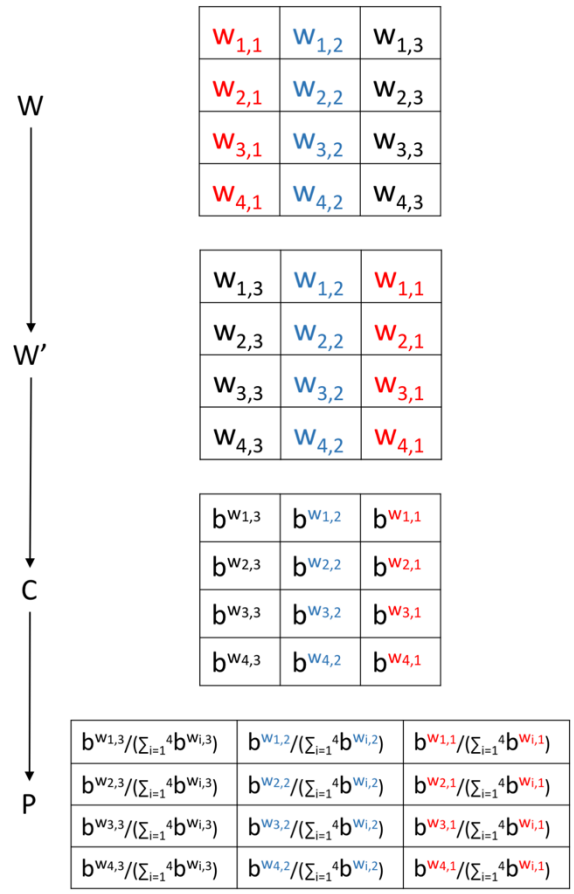

Fig. 1 The transformation from kernel to PWM. Elements were colored to signify the flipping step (W -> W').

Algorithm 1: the transformation from kernel to PWM

Input : the kernel W and an arbitrary base of logarithm b (b>1)
Output: the transformed PWM P(W, b)
begin
for i in {1, 2, 3, 4}
  for j in {1, 2, …, L}
    $w'_{i,j} = w_{i,L-j+1}$
    $c_{i,j} = b^{w'_{i,j}}$
for i in {1, 2, 3, 4}
  for j in {1, 2, …, L}
    $(P(W, b))_{i,j} = c_{i,j} / (c_{1,j} + c_{2,j} + c_{3,j} + c_{4,j})$

Interpreting a kernel using the exact transformation

The exact transformation maps a kernel to infinite PWMs parameterized by b only, because Theorem 1 always holds regardless of the specific b chosen. As demonstrated by Corollary 1 (See Supplementary Notes), these PWMs must be different as long as the original kernel prefers certain nucleotide(s) for at least one position, yet any of them is capable of regressing/classifying the input sequences in exactly the same manner as the original kernel does under popular CNN frameworks (e.g. those in (Alipanahi *et al.*, 2015; Quang and Xie, 2016; Zhou and Troyanskaya, 2015)). Therefore, the user may choose a specific PWM of interest based on prior biological knowledge.

If no such prior knowledge is available, which is common, an alternative method is to find the optimal b using maximum likelihood estimation (MLE). The exact likelihood depends on the model structure and underlying assumptions, and should be treated on a case-by-case basis. Nevertheless, to make the MLE directly accessible to a wide range of preexisting models, we deduced and implemented the following log-likelihood $q(b|X^{(1)},\ldots,X^{(n)},W^{(1)},\ldots,W^{(k)})$ for a popular CNN framework in which an input sequence is fed to convolution, linear or ReLU activation, global max-pooling, linear function, and finally arbitrary functions (see Supplementary Notes for the technical details):

$$q(b|\mathbf{X}^{(1)},\cdots,\mathbf{X}^{(n)},\mathbf{W}^{(1)},\cdots,\mathbf{W}^{(k)})$$
$$=(\ln b)\sum_{s=1}^{n}\sum_{t=1}^{k}(\mathbf{X}^{(s)}*\mathbf{W}^{(t)})_{j^*(s,t)}-n\sum_{t=1}^{k}\sum_{j'=1}^{L}\ln\left(\sum_{i=1}^{4}e^{(\ln b)w_{i,j'}^{(t)}}\right) \quad (2)$$

where $X^{(s)}$ are the n indexed input sequences, $W^{(t)}$ the k indexed kernels, L the length of each kernel, $j^*(s, t)$ the starting coordinate of the max-scored fragment by $W^{(t)}$ on $X^{(s)}$, and e the base of the natural logarithm.

Because evaluating the derivative of b is cumbersome, we used a derivative-free optimizer, Nelder-Mead (Nelder and Mead, 1965), from the Python package SciPy (Jones *et al.*, 2001). Of course, it is also feasible to use other optimizers by numerically estimating the derivative.

Reusing a PWM in CNN models

Corollary 2 guarantees that, from each PWM (with the restriction that no 0's or 1's are present), a kernel can be generated by the following steps (Figure 2) that is capable of classifying/regressing sequences in exactly the same manner as the PWM:

(1) Similar to the transformation above, assume that the PWM to be transformed is a 4-by-L matrix P, where L is the length of this PWM. The element of P at the ith row and jth column is denoted as $p_{i,j}$.
(2) Choose an arbitrary base of logarithm, b (b>1);
(3) Skip step 5 (normalization) by taking C = P;
(4) Invert step 4 to obtain W' (i.e., set $w'_{i,j} = \log_b c_{i,j}$);
(5) Invert step 3 to obtain the transformed kernel W (i.e., set $w_{i,j} = w'_{i,L-j+1}$);
(6) If ReLU activation is to be used, add to the transformed kernel a positive shift sufficiently large to make all elements nonnegative.

Algorithm 2: the (back)-transformation from PWM to kernel

Input : the PWM P and an arbitrary base of logarithm b (b>1)
Output: the (back-)transformed kernel W
begin
C = P
for i in {1, 2, 3, 4}
  for j in {1, 2, …, L}
    $w'_{i,j} = \log_b c_{i,j}$
    $w_{i,j} = w'_{i,L-j+1}$
if ReLU activation is to be used **then**
  $w_{i,j} = w_{i,j}$ + a positive constant that is no smaller than $|\min_{i,j}(w_{i,j})|$

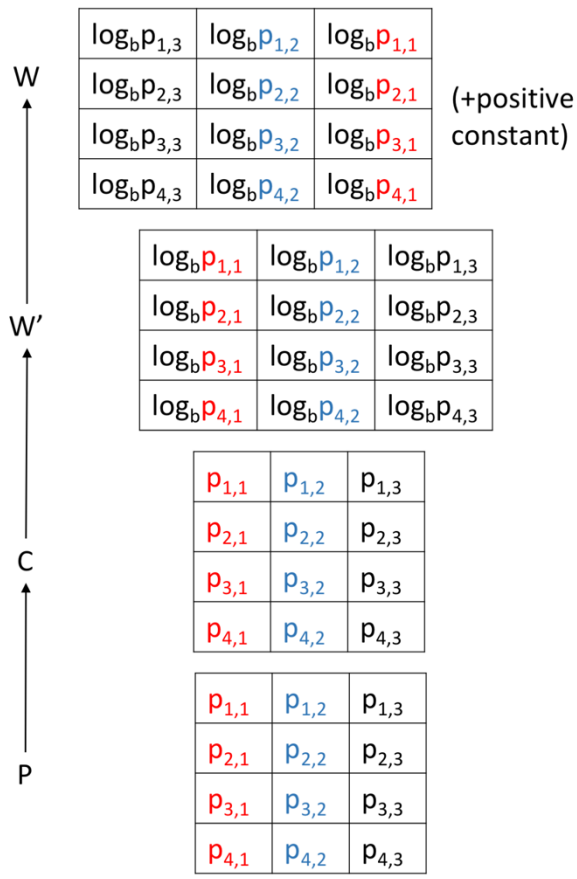

Fig. 2 The (inverted) transformation from PWM to kernel. Elements were colored to signify the flipping step (W'->W). Note that the normalization step is skipped, and thus the C matrix is identical to P.

Empirical results corroborate the outstanding performance of the exact transformation over the heuristic one

The exact transformation outperformed the heuristic one at classifying sequences with known sequence- and structure-conserved motifs

Although the exact transformation can, in theory, generate PWMs that classify/regress sequences in exactly the same way as the original kernel, it is still unknown whether such PWMs classify sequences with relevant motifs more accurately than PWMs transformed by previous method. To quantitatively compare the exact transformation and the heuristic transformation proposed by Deepbind (Alipanahi *et al.*, 2015), we first extracted all motifs from known sequence (JASPAR (Mathelier *et al.*, 2016)) and structure (Rfam (Nawrocki *et al.*, 2015)) motif databases. For each motif, we performed the following analysis (with the technical details available in Supplementary Table 1):

1. We simulated positive and negative sequences, and trained a CNN model to classify these sequences;
2. We extracted the kernels and transformed each kernel into a PWM by either (1) the exact transformation or (2) the Deepbind transformation;
3. We evaluated which transformation would better classify sequences by comparing their area under ROC curves (AUROC) and area under PR curves (AUPRC) across the entire dataset. We performed a Wilcoxon signed-rank test for both AUPRC and AUROC, with the null hypothesis being that the mean AUC of the exact transformation is equal to or less than the mean AUC of the Deepbind's transformation.

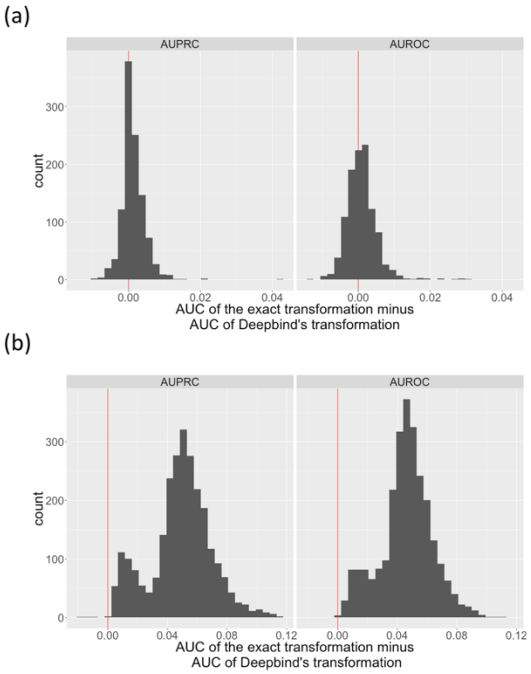

Fig. 3 The exact transformation rivaled Deepbind's in most cases of random sequences with (a) sequence-conserved motifs, and even outperformed Deepbind's in cases with (b) structure-conserved motifs.The histograms summarize the distribution of the AUC difference between the exact transformation and the Deepbind's transformation; the red line is x=0, on the right of which the exact transformation performed better than Deepbind's. Codes for reproducing this plot are available on the Github repository.

As being expected, we found that the exact transformation almost always rivaled (Figure 3(a); Wilcoxon signed-rank test p-value < 2.2e-16 for AUPRC and =1.053e-09 for AUROC, though the difference is generally no more than 0.01) or even outperformed (Figure 3(b); Wilcoxon signed-rank test p-value < 2.2e-16 for both AUPRC and AUROC) the heuristic one for both sets of motifs. Surprisingly, for some sequence-conserved JASPAR motifs, Deepbind's AUROC and AUPRC dropped to values slightly greater than 0.5, while ours are close to 1. We also noted that the outperformance was independent of the quality of the training of the original model, suggesting the exact transformation is able to capture the correct signal by the resulting PWM.

The exact transformation is able to reproduce the motif detection performance for real-world Deepbind CNN model faithfully.

We further demonstrated the powerfulness of the exact transformation by testing it on all 461 real-world ChIP-Seq and SELEX motifs that are not deprecated from Deepbind's paper (Alipanahi *et al.*, 2015). Because Deepbind models produce continuous outputs instead of discrete labels, AUROC and AUPRC are no longer appropriate here, and we switched to the other two types of losses, the mean absolute percentage error (MAPE; which is defined as the mean value (across all samples) of the absolute value of "(predicted value - real value) / real value" for each sample) and mean squared error (MSE).

For each of these motifs, we then performed the following analysis (with the technical details available in Supplementary Table 2):

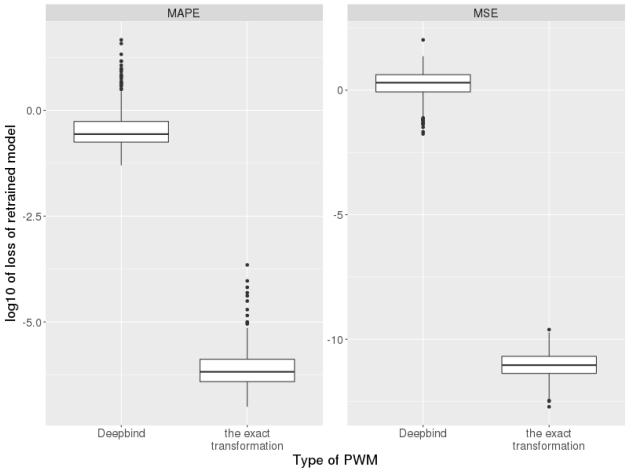

Fig. 4 The exact transformation outperformed Deepbind's in exemplar motifs of Deepbind's original work. For each of these motifs, the log-likelihoods from the two transformations were tested on how close they could approximate the output of original CNN model trained by Deepbind itself (by both mean absolute percentage error (MAPE) and mean squared error (MSE)). Codes for reproducing this plot are available on the Github repository.

1. We reconstructed the motif's Deepbind CNN model with structure and parameters provided by Deepbind itself. We note here that two motifs of the same protein carried out by ChIP-Seq and SELEX will be considered as different ones, because their models were trained using different datasets.
2. We used this model to generate the benchmarking dataset, where the input dataset is the very input sequence dataset for the motif, and the output dataset is the model's prediction made for the input dataset (note that Deepbind models output for each input sequence not a classification label, but a continuous unbounded real value);

3. We extracted the kernels and transformed each kernel into a PWM by either (1) the exact transformation or (2) the Deepbind transformation. Only sequences samples passing the ReLU activation were used for the Deepbind transformation and, if there's no such sample for a certain kernel of the motif, this kernel cannot be transformed into a Deepbind PWM, and we did not proceed further on the motif in question, which excluded 93 models from the follow-up analysis.
4. We evaluated which transformation would better approximate the benchmarking dataset by comparing their MAPE and MSE values on re-trained linear or neural network models. To avoid randomness brought by stochastic optimization, for PWMs from the exact transformation we used Theorems 1 and 2 to derive the parameters directly without any training, which is guaranteed to have the optimal performance (up to computer numerical error).

Consistent with the previous subsection, the exact transformation outperformed Deepbind's transformation on real datasets, further suggesting the superiority of the exact transformation (Figure 4).

Discussion

We have found for the first time that the kernel, together with a logarithm base, can be transformed into a PWM with a log-likelihood that is the sum of a constant and the kernel's convolution. The mathematical solidity of this method connects intimately the computational field of CNNs and the biological field of functional elements, and both empirical simulations and testing on real-world cases have further demonstrated the superiority of the exact transformation to the heuristic one.

The failure of Deepbind's PWM to recover the performance could be partially due to its inherent setup of optimization. From a perspective of maximum likelihood estimation (MLE), the resulting Deepbind PWM maximizes the probability of observing highly scoring fragments given the PWM itself, but does not take into account the kernel itself; i.e., for the t-th kernel $W^{(t)}$ it tries to find the PWM $P^{(t)}$ that maximizes the following conditional probability (using notations from the MLE above), but without considering the specific value of the kernel to transform:

$$\prod_{s=1}^{n} \mathrm{Prob}(\mathbf{X}^{(s)}[1:4, j^*(s,t):(j^*(s,t)+L-1)]|\mathbf{P}^{(t)}) \quad (3)$$

If we further impose the condition that the specific value of the kernel is also observed and should not be biased, we will end up with a combination of such MLE with the exact transformation; that is, it will try to find the PWM $P^{(t)}$ that maximizes the following conditional probability and that is subject to the following restriction:

$$\prod_{s=1}^{n} \mathrm{Prob}(\mathbf{X}^{(s)}[1:4, j^*(s,t):(j^*(s,t)+L-1)]|\mathbf{P}^{(t)})$$
$$s.t.\ \mathbf{P}^{(t)} = \mathbf{P}^{(t)}(\mathbf{W}^{(t)}, b) \quad (4)$$

This could explain partially why the exact transformation outperformed Deepbind's in some cases, especially when we'd like to recover the performance of a CNN model whose output is continuous.

Acknowledgements

Funding: This work was supported by funds from the National Key R&D Program of China (2016YFC0901603), China 863 Program (2015AA020108), as well as the State Key Laboratory of Protein and Plant Gene Research. The research of G.G. was supported in part by the National Program for Support of Top-notch Young Professionals. Part of the analysis was performed on the Computing Platform of the Center for Life Sciences of Peking University, and we thank Dr. Fangjin Chen for his help.

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.* (2017) DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.*, **18**.

Bailey,T.L. *et al.* (2009) MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Res.*, **37**, W202–W208.

Cotter,A. *et al.* (2011) Better Mini-batch Algorithms via Accelerated Gradient Methods. In, *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11. Curran Associates Inc., USA, pp. 1647–1655.

Hubley,R. *et al.* (2016) The Dfam database of repetitive DNA families. *Nucleic Acids Res.*, **44**, D81–D89.

Jones,E. *et al.* (2001) SciPy: Open source scientific tools for Python.

LeCun,Y. *et al.* (1989) Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.*, **1**, 541–551.

Machanick,P. and Bailey,T.L. (2011) MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics*, **27**, 1696–1697.

Mathelier,A. *et al.* (2016) JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Res.*, **44**, D110–D115.

Nawrocki,E.P. *et al.* (2015) Rfam 12.0: updates to the RNA families database. *Nucleic Acids Res.*, **43**, D130-137.

Nelder,J.A. and Mead,R. (1965) A Simplex Method for Function Minimization. *Comput. J.*, **7**, 308–313.

Quang,D. and Xie,X. (2016) DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.*, **44**, e107.

Ray,D. *et al.* (2013) A compendium of RNA-binding motifs for decoding gene regulation. *Nature*, **499**, 172–177.

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