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 applied directly to DNA/RNA sequences

Yang Ding[1*], Jingyi Li[1], Meng Wang[1], and Ge Gao[1*]

[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.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

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. Surprisingly, our PWMs almost always outperformed heuristic ones at sequence classification, whether the discriminative motif was sequence- or structure-conserved. 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.

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

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

Introduction

Convolutional neural networks (CNNs) are widely and successfully used as a deep learning framework in various fields (Girshick *et al.*, 2014; Kim, 2014; Sainath *et al.*, 2013). Recently, CNNs have also been used in the study of functional DNA/RNA elements (Alipanahi *et al.*, 2015; Angermueller *et al.*, 2017; Quang and Xie, 2016; Zhou and Troyanskaya, 2015). Because of their powerful learning abilities, it is tempting to interpret the biological implication of signals captured by CNN. Specifically, convolutional kernels directly scanning input DNA/RNA sequences might reflect certain functional DNA/RNA elements. Therefore, throughout this study we focus on these kernels and their convolution.

An immediate solution is to relate a kernel to a "similar" sequence profile. However, this is not directly applicable because most known sequence profiles 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.

An alternative method is to identify a reasonable transformation of kernels into these models. Deepbind (Alipanahi *et al.*, 2015) has provided such a transformation, by first stacking the kernel's highly scored fragments from input sequences and then normalizing nucleotide counts to obtain the final PWM. This transformation is heuristic and biologically reasonable when the underlying signal resembles a fixed sequence. However, this transformation has a critical weakness: the resulting PWM is not guaranteed to perform sequence classification as well as the original kernel; an obvious drop in performance after this transformation would thus be possible.

In this study, we addressed this problem using a novel, deductive kernel-to-PWM transformation. We proved that the log-likelihood of the resulting PWM on any DNA/RNA sequence is exactly the sum of a constant and the convolution of the original kernel on the same sequence. We further showed that, under popular CNN frameworks, the transformed PWM is capable of regressing/classifying sequences in exactly the same manner as the kernel. In simulations, our PWMs outperform Deepbind's heuristic PWMs at detecting both sequence- and structure-conserved motifs. To our knowledge, this is the first biological interpretation of kernels that convolve DNA/RNA sequences directly. The mathematical solidity of this method connects intimately the computational field of CNNs and the biological field of functional elements. To provide this transformation to a wide audience, we implemented the transformation, its back-transformation, and a maximum likelihood estimation of its optimal PWM in Python scripts.

Theoretical results

Below we describe the main results of this transformation and its consequences. Proofs of theorems and corollaries can be found in the Supplementary Information.

The transformation

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

(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 b (b > 1) for the logarithm used for the log-likelihood calculation.

(3) Flip W along the second 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 given sequence X no shorter than W, we have:

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

where * denotes the convolution operator that does not consider border-crossing cases (i.e., W must fall completely within X; see Supplementary Information 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.

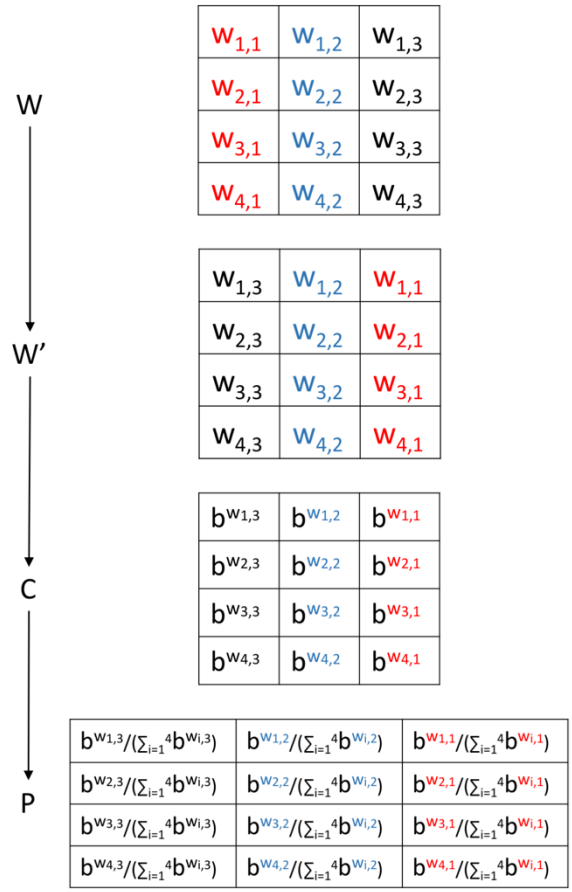

Fig. 1 The transformation from kernel to PWM. Elements were colored to signify the flipping step (W -> W').

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

Interpreting a kernel using this transformation

This 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 Information), 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 $q\big(b|X^{(1)},\dots,X^{(n)},W^{(1)},\dots,W^{(k)}\big)$ 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 Information for the technical details):

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

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).

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 regressing/classifying sequences in exactly the same manner as the PWM:

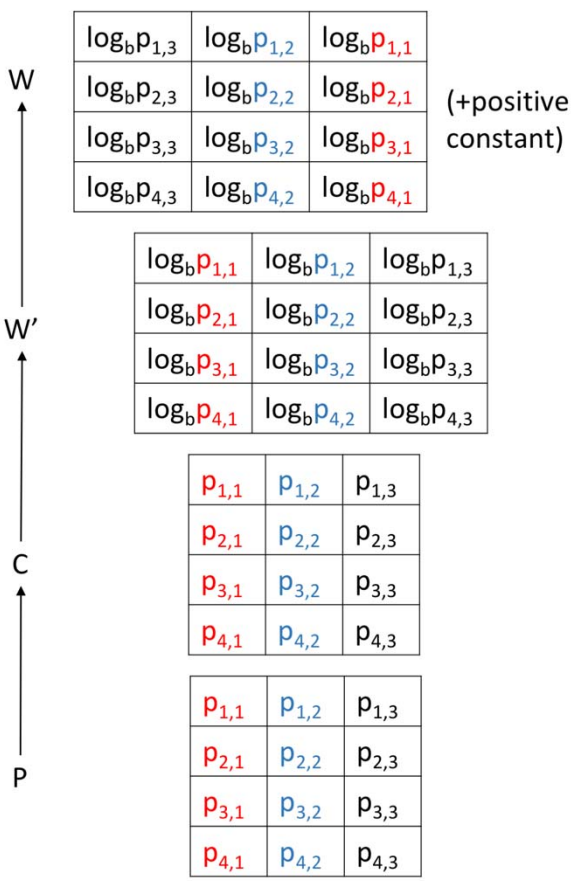

Fig. 3 The (inverted) transformation from PWM to kernel. Elements were colored to signify the flipping step (W'->W). Note that the normalization step is skipped, and thus the C matrix is identical to P.

(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 b (b>1);

(3) Skip step 5 (normalization) by taking C = P;

(4) Invert step 4 to obtain W' (i.e., set $w'_{i,j} = \log_b p_{i,j}$);

(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.

Simulation results

Since no theoretical results were obtained to quantitatively compare our transformation and the Deepbind transformation (Alipanahi *et al.*, 2015), we explored a comparison using numerical simulation. Specifically, 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

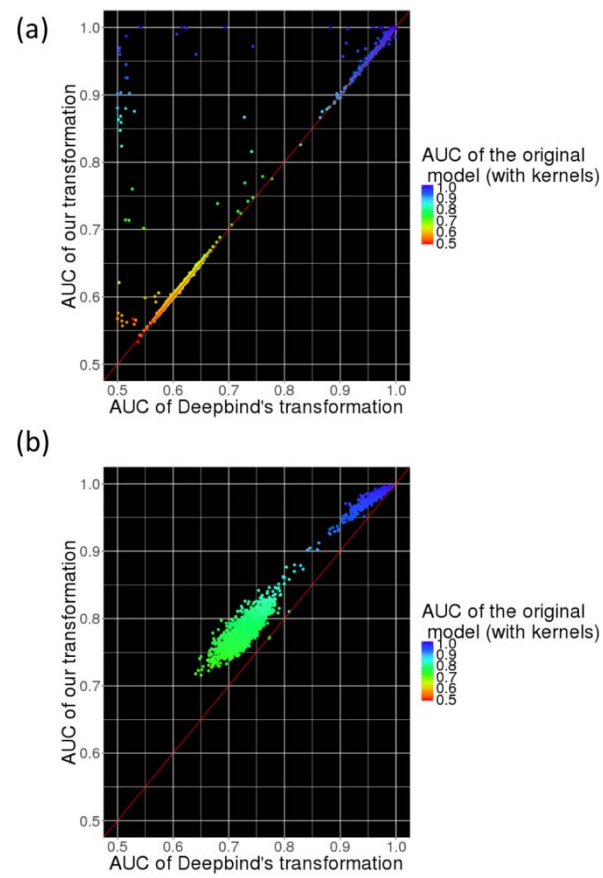

Fig. 2 Our transformation outperformed Deepbind's in most cases for both (a) sequence-conserved motifs and (b) structure-conserved motifs. Each point describes the two AUC values for a certain motif; the red line is y=x, above which our transformation performed better than Deepbind's. Points are colored with the AUC of the original kernel(s) as an indicator of how well the original model was trained; the outperformance, however, was independent of model training.

CNN model to classify these sequences;

2. We extracted the kernels and transformed each kernel into a PWM by either (1) our transformation or (2) the Deepbind transformation;
3. We evaluated which transformation would better classify sequences better by comparing their AUC values across the entire dataset.

We simulated all motifs from JASPAR and Rfam and found that our transformation almost always outperformed the Deepbind for both sets of motifs (Figure 3). Surprisingly, for some sequence-conserved JASPAR motifs, Deepbind's AUC dropped to values slightly greater than 0.5, although both the original kernel and our PWM have very high AUCs. We also noted that the outperformance was independent of the quality of the training of the original model, suggesting that the outperformance could be a general phenomenon. Therefore, one should use our transformation to ensure the capture of the correct signal by the resulting PWM.

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. Because the PWM transformed is capable of regressing/classifying sequences in exactly the same manner as the kernel, and because numerical simulations have demonstrated the superiority of this transformation to the heuristic Deepbind's transformation, we believe the interpretation of CNN kernels is more accurate when using our transformation. The failure of Deepbind's PWM to recover the performance could be due to its inherent optimizer setup; the resulting PWM maximizes the probability of observing highly scoring fragments given the PWM itself but does not take into account the kernel itself.

The connection between CNN kernels and the PWM enables the representation of a complex motif, e.g., RNA secondary structure, by a series of PWMs, provided that the model was trained with high accuracy. This conclusion is supported by our simulation on Rfam motifs that are described by covariance models, which generally exceed the descriptive capabilities of a single PWM. Because PWM log-likelihoods can be calculated more rapidly than the statistics of covariance models, the representation of a complex motif with its PWM surrogate could be used to speed up RNA structure predictions and comparisons.

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**.

Girshick,R. *et al.* (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. IEEE, pp. 580–587.

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.

Kim,Y. (2014) Convolutional Neural Networks for Sentence Classification. Association for Computational Linguistics, pp. 1746–1751.

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.

Sainath,T.N. *et al.* (2013) Deep convolutional neural networks for LVCSR. IEEE, pp. 8614–8618.

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