Identifying complex sequence patterns in massive omics data with a variable-convolutional layer in deep neural network

Jing-Yi Li[1,#], Shen Jin[2,#], Xin-Ming Tu[1], Yang Ding[1,3,*], and Ge Gao[1,*]

[1] Biomedical Pioneering Innovation Center (BIOPIC) & Beijing Advanced Innovation Center for Genomics (ICG), Center for Bioinformatics (CBI), and State Key Laboratory of Protein and Plant Gene Research at School of Life Sciences, Peking University, Beijing, 100871, China, [2] Computational Biology Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, 15213,[3] Beijing Institute of Radiation Medicine, Beijing, 100850, China

[#] These authors contributed equally to this work.

* To whom correspondence should be addressed. Email: gaog@mail.cbi.pku.edu.cn, and mail could also be sent to dingy@mail.cbi.pku.edu.cn

ABSTRACT

Motif identification is among the most canonical and essential computational tasks for bioinformatics and genomics. Here we proposed a novel convolutional layer for deep neural network, named Variable Convolutional (vConv) layer, for effective motif identification in high-throughput omics data by learning kernel length from data adaptively. Empirical evaluations on DNA-protein binding and DNase footprinting cases well demonstrated that vConv-based networks have superior performance to their convolutional counterparts regardless of model complexity. All source codes are freely available at GitHub (https://github.com/gao-lab/vConv) for academic usage.

INTRODUCTION

Recurring sequence motifs (Achar and Sætrom, 2015; Kulakovskiy and Makeev, 2013) have been well demonstrated to exert or regulate important biological functions, such as protein binding (Stormo, 2015), transcription initiation (Kadonaga, 2012), alternative splicing (Blencowe, 2000), subcellular localization (Zhang, et al., 2014), translation control (Zucchelli, et al., 2015), and microRNA targeting (Thomson and Dinger, 2016). Effectively and efficiently identifying these motifs in massive omics data is a critical step for follow-up investigations.

Various computational tools have been developed to identify sequence motifs via word-based and profile-based models (Das and Dai, 2007; Lihu and Holban, 2015; Liu, et al., 2018; Tran and Huang, 2014; Zambelli, et al., 2013). Word-based tools start with a fixed-length and conservative segment and then perform a global scanning; such tools include DREME (Bailey, 2011), Fmotif (Jia, et al., 2014), RSAT peak-motifs (Thomas-Chollier, et al., 2012), SIOMICS (Ding, et al., 2015; Ding, et al., 2014), and Discover (Maaskola and Rajewsky, 2014). While these tools can theoretically obtain the globally optimal solution, they suffer from high computational complexity when applied to data with complex motifs or large-scale datasets (Das and Dai, 2007). Profile-based tools attempt to find representative motifs by heuristically fine-tuning a series of possible motifs, either generated from a subset of input data or randomly chosen (Ikebata and Yoshida, 2015; Kulakovskiy, et al., 2010; Sharov and Ko, 2009) (Bailey, et al., 2006; Machanick and Bailey, 2011), leading to a faster (but probably sub-optimal) motif calling (Lihu and Holban, 2015).

Several convolutional neural network (CNN)-based tools have been proposed recently as a more scalable approach for identifying motifs. Alipanahi et al. developed DeepBind to identify protein binding motifs from large-scale ChIP-Seq datasets (Alipanahi, et al., 2015) by treating each convolutional kernel as an individual motif scanner and discriminating motif-containing sequences from others based on the output of all kernels. Along with this line, several convolution-based networks have been proposed to effectively handle large amount of data in various settings (Angermueller, et al., 2017; Kelley, et al., 2018; Koo and Eddy, 2019; Wang, et al., 2018; Zhang, et al., 2018; Zhou, et al., 2018). Meanwhile, however, the inherent fixed-kernel design of canonical convolutional networks could hinder effective identification(Han, et al., 2018; Yin and Schütze, 2016; Zhang, et al., 2018) of *bona fide* sequence patterns, which are usually of various lengths and unknown *a priori* and often function combinatorically(Lambert, et al., 2018; Reiter, et al., 2017). One possible workaround is to build these representations in a hierarchical manner (Kelley, et al., 2018; Koo and Eddy, 2019), yet the deeper structure itself would demand additional time/space cost for network training and hyper-parameter search (also see Supplementary Notes 1).

Here, we proposed a novel convolutional layer for deep neural network called Variable

Convolutional neural layer (vConv), which learns the kernel length directly from the data. Evaluations based on both simulations and real-world datasets showed that vConv-based networks outperformed canonical convolution-based networks, making it an ideal option for the *ab initio* discovery of motifs from high-throughput datasets. Further inspection also suggested the vConv layer being robust for various hyper-parameter setups. A Keras-based implementation of vConv is available at GitHub (https://github.com/gao-lab/vConv).

MATERIAL AND METHODS

Design and implementation of vConv

Without loss of generality, we focus on nucleotide sequences where each nucleotide can take only one of four alphabets (A, C, G, and T (for DNA) or U (for RNA)). vConv (Fig. 1) is designed to be a convolutional layer equipped with a trainable "mask" to adaptively tune the effective length of the kernel during training. Specifically, for the *z*-th kernel w^z of length $L_k, z \in \{1, \dots, n\}$, the "mask" $f^*(w_m^z) := f^*((w_m^{z,0}, w_m^{z,1}))$ is a matrix with the same shape as this kernel (i.e., $L_k \times 4$) and most elements close to either 0 or 1 (Fig. 1A). Multiplying this mask and the original kernel by the Hadamard product $\circ$ $(f^*(w_m^z) \circ w^z)$ thus effectively masks the original kernel with the corresponding close-to-0 elements and gives the masked kernel (Fig. 1B), which can then be used as an ordinary kernel in a canonical convolutional layer (Fig. 1C). Mathematically, this mask is parameterized by two sigmoid functions of opposite orientation with two scalars $w_m^{z,0}$ and $w_m^{z,1}$ representing the left- and right-boundaries, respectively:

$$f^*\left(\left(w_m^{z,0}, w_m^{z,1}\right)\right)[i,j] := \frac{1}{1+e^{-(i-1)+w_m^{z,0}}} + \frac{1}{1+e^{(i-1)-w_m^{z,1}}} - 1 \quad (5)$$

As the above equation implies, all *i*'s falling outside the boundaries (i.e., $i < w_m^{z,0}$ or $i > w_m^{z,1}$) have their (masking) elements $f^*\left(\left(w_m^{z,0}, w_m^{z,1}\right)\right)[i,j]$ close to zero, and all *i*'s within the boundaries have their elements close to 1; those *i*'s around the boundaries have their elements around 0.5, thus 'soft'-masking boundary kernel elements.

To make $w_m^{z,0}$ and $w_m^{z,1}$ converge faster during training, we combine the binary cross-entropy (BCE) loss with a sum of masked Shannon losses (MSLs) from each kernel mask. Mathematically, we have the following total loss L:

$$L = \mathrm{BCE} + \lambda \cdot \sum_z MSL(w^z, w_m^z)$$
$$= \sum_{c=1}^{N}\left(y_c \log \hat{y}_c + (1-y_c)\log\left(1-\hat{y}_c\right)\right) + \lambda \cdot \sum(f^*(w_m^z) \cdot H(P_z) - threshold) \quad (6)$$

Where $H(P_z) := \sum_{i=1}^{L_k}(-\sum_{j=1}^{4}((P_z)[i,j])log_2((P_z)[i,j])$ is the sum of the Shannon entropy across all nucleotide positions of P_z, the position weight matrix (PWM) learned by the *z*-th kernel. For precision, we set $P_z = P(w^z, b = 2)$ following the exact kernel-to-PWM transformation specified by Ding et al. (Ding, et al., 2018). One can then immediately deduce the formula for updating the left boundary $w_m^{z,0}[\mathrm{t}-1]$ at time step *t* via gradient descent with learning rate *r* (*r*>0):

$$w_m^{z,0}[t] = w_m^{z,0}[t-1] - r \cdot \frac{\partial L(D)}{\partial w_m^{z,0}[t-1]}$$
$$= w_m^{z,0}[t-1] - r \cdot \frac{\partial BCE}{\partial w_m^{z,0}[t-1]} - r \cdot \frac{\partial MSL(w^z, w_m^z[t-1])}{\partial w_m^{z,0}[t-1]}$$

$$= w_m^{z,0}[t-1] - r \cdot \frac{\partial BCE}{\partial w_m^{z,0}[t-1]} + r \sum_{i=0}^{L_k-1} \frac{e^{-(i-1)+w_m^{z,0}[t-1]}}{(1+e^{-(i-1)+w_m^{z,0}[t-1]})^2} 2(H\big(P(w^z,2)\big)_i - threshold)$$

Similarly, for updating the right boundary $w_m^{z,1}[t-1]$, we have:

$$w_m^{z,1}[t] = w_m^{z,1}[t-1] - r \cdot \frac{\partial L(D)}{\partial w_m^{z,1}[t-1]}$$

$$= w_m^{z,1}[t-1] - r \cdot \frac{\partial BCE}{\partial w_m^{z,1}[t-1]} - r \sum_{i=0}^{L_k-1} \frac{e^{(i-1)-w_m^{z,1}[t-1]}}{(1+e^{(i-1)-w_m^{z,1}[t-1]})^2} 2(H\big(P(w^z,2)\big)_i - threshold)$$

The Shannon entropies of all kernel positions far from the mask boundaries $(w_m^{z,0}, w_m^{z,1})$ contribute little to the final derivatives. Therefore, if most boundary-flanking kernel positions have a low Shannon entropy (i.e., a high information content; defined by $H\big(P(w^z,2)\big)_i - threshold < 0$), then the masked Shannon loss (MSL) will help push the boundaries outwards (i.e., $\frac{\partial MSL(w^z, w_m^z[t-1])}{\partial w_m^{z,0}[t-1]} < 0$ and $\frac{\partial SL(w^z, w_m^z[t-1])}{\partial w_m^{z,1}[t-1]} > 0$) during gradient descent, just as if the current mask is too narrow to span all informative positions. Likewise, if most boundary-flanking kernel positions have a high Shannon entropy (i.e. a low information content; defined by $H\big(P(w^z,2)\big)_i - threshold > 0$), then the MSL will help push the boundaries inwards (i.e., $\frac{\partial MSL(w^z, w_m^z[t-1])}{\partial w_m^{z,0}[t-1]} > 0$ and $\frac{\partial SL(w^z, w_m^z[t-1])}{\partial w_m^{z,1}[t-1]} < 0$), just as if the current mask is too broad to exclude some positions with low information content.

To speed up computation in practice, we approximate the sum by ignoring most small, outside-mask kernel positions and retaining only those terms with $i \in \left[\lfloor w_m^{z,0} \rfloor - 2, \lceil w_m^{z,0} \rceil + 2\right]$ or $[\lfloor w_m^{z,1} \rfloor - 2, \lceil w_m^{z,1} \rceil + 2]$.

Finally, when introducing a vConv layer into a model, values of $(w_m^{z,0}[t=0], w_m^{z,1}[t=0])$, λ, and $threshold$ in MSL are of user's choice. In all subsequent benchmarking on vConv-based networks, unless otherwise specified, we set for each vConv layer the $(w_m^{z,0}[t=0], w_m^{z,1}[t=0])$ as $(\frac{1-l}{2}, \frac{1+l}{2})$ for an unmasked kernel length of *l*, the λ as 0.0025, and the $threshold$ in MSL as $-\frac{7}{10}\log_2\frac{7}{10} + \sum_{i=1}^{c-1} -\frac{3}{10*(c-1)}\log_2\frac{3}{10*(c-1)}$ for a kernel with *c* channels. This value of $threshold$ is the Shannon entropy for the (discrete) distribution whose highest probability is 0.7, and all others *c*-1's are equally divided. When processing the nucleic acid sequence, there are totally *c*=4 different probability values, so the probability of the remaining 3 items equally divided is (1-0.7)/3; on the other hand, in Basenji-based Basset networks, the *c* for deeper convolutional layers is not necessarily 4.

Benchmark vConv-based networks on simulated sequence classification

We simulated for each motif case 6,000 sequences (with 3,000 positive and 3,000 negative) of length 1,000, picked 600 as the test dataset randomly, and split the rest into 4,860 training and 540 validation sequences by setting "validation_split=0.1" in model.fit of Keras Model API (version 2.2.4)(Chollet, 2015); in this way, for each motif case, the same collection of training, validation, and test datasets was used for all hyper-parameter settings tested (see below). The ratio of counts of positive to negative sequences in the test dataset was within [0.8, 1.2] for all 7 motif cases. Each positive sequence is a random sequence with a "signal sequence" inserted at a random location, and each negative sequence is a fully random sequence. The "signal sequence" is a sequence fragment generated from one of the motifs associated with the case in question (Table 2). For the cases of 2, 4, 6 and 8 motifs, new motifs were introduced incrementally (i.e., all motifs in "2 motifs" were also included in "4 motifs", all motifs in "4 motifs" were also included in "6 motifs", and all motifs in "6 motifs" were also included in "8 motifs").
In each iteration of benchmarking a certain case, we (1) picked a hyper-parameter setting from Table 1; (2) used the training and validation datasets to train first a vConv-based network (see Supplementary Fig. 1 for its structure) with this hyper-parameter setting and then a canonical convolution-based network with a model structure identical to that of the vConv-based network, except that the vConv layer was replaced with a canonical convolutional layer with the same kernel number and (unmasked) kernel length; and (3) obtained the AUROC values of these two networks on the test dataset. Iterating over all possible hyper-parameter settings yielded a series of AUROC values for both the vConv-based network and the convolution-based network for the case at hand.

We used AdaDelta (Zeiler, 2012) with learning rate 1 as the optimizer. All parameters except for vConv-exclusive ones were initialized by the Glorot uniform initializer (Glorot and Bengio, 2010); vConv-exclusive parameters were initialized as specified in the subsection "Design and implementation of vConv". Both convolution-based and vConv-based networks were trained with early stopping on validation loss with patience set to 50. vConv-based networks with *n* kernels were first trained without updating $\{w_m^z \mid \forall\, z \in \{1, \ldots, n\}\}$for 10 epochs and then trained with updating $\{w_m^z \mid \forall\, z \in \{1, \ldots, n\}\}$.

Benchmark vConv-based networks on real-world sequence classification

To further demonstrate the performance of vConv-based networks in the real world, we examined whether replacing convolutional layers with vConv helps to improve the performance for motif identification in real-world ChIP-Seq data, against three published convolution-based networks: DeepBind (Alipanahi, et al., 2015), Zeng et al.'s convolution-based networks (Zeng, et al., 2016), and Basset (Kelley, et al., 2016).

For the first comparison, we downloaded DeepBind's original training (with positively

sequences only) and test datasets from https://github.com/jisraeli/DeepBind/tree/master/data/encode, and generated negative training dataset by shuffling positive sequences with matching dinucleotide composition as specified by DeepBind (Alipanahi, et al., 2015). We then trained vConv-based networks (see below for the model structure and training details) and compared their AUROC on test datasets with those of three sets of DeepBind performances from Supplementary Table 5 of DeepBind's original paper (Alipanahi, et al., 2015): (1) the DeepBind* networks, which were trained by using all the training dataset; (2) the DeepBind networks, whose positive training sequences consist of those from the top 500 odd-numbered peaks only; and (3) the DeepBindbest networks, the one of DeepBind and DeepBind* with larger AUROC for each test dataset. During comparison, we found that there are two datasets with different AUROC values yet sharing the same model name in their original dataset (REST_HepG2_NRSF_HudsonAlpha and REST_SK-N-SH_NRSF_HudsonAlpha), which makes us unable to determine their true AUROC values, so we excluded them from the comparison.

For the second comparison between vConv-based networks and Zeng et al.'s (Zeng, et al., 2016) convolution-based networks, we downloaded 690 ENCODE ChIP-Seq-based training and test datasets representing the DNA binding profile of various transcription factors and other DNA-binding proteins from http://cnn.csail.mit.edu/motif_discovery/. We then trained vConv-based networks (see below for the model structure and training details) and compared its AUROC value on test datasets with the AUROC values of Zeng et al.'s networks from http://cnn.csail.mit.edu/motif_discovery_pred/.

The first two comparisons used the same model structure (Supplementary Fig. 1) and hyper-parameter space (Supplementary Table 1). For direct comparison between vConv-based networks and convolution-based networks for DeepBind and Zeng et al.'s cases (Zeng, et al., 2016), the vConv-based network was implemented by replacing the convolutional layer of each network considered for comparison with a vConv layer (with exactly the same hyper-parameters for the number of kernels and the initial (unmasked) kernel length). The hyper-parameter space is listed in the Supplementary Table 1. The remaining details of parameter initialization followed those of the corresponding convolution-based networks (Alipanahi, et al., 2015; Zeng, et al., 2016), and all these networks used the training strategy described in the simulation case.

Finally, for the third comparison with the Basset network (Kelley, et al., 2016), we used the Basenji reimplementation of Basset (Kelley, et al., 2018) (https://github.com/calico/basenji/tree/master/manuscripts/basset) made and recommended by the original author of Basset (D.Kelley, personal communications); this reimplementation is a deep CNN with nine convolutional layers (see Supplementary Fig. 2 for the full details). We (1) generated the datasets by running "make_data.sh" under "manuscripts/basset/" (of this

repository), (2) re-trained the original Basset networks by running “python basenji_train.py -k -o train_basset params_basset.json data_basset” under “manuscripts/basset/” (of this repository) to obtain their AUROC values, (3) replaced either the first (‘Single vConv-based’; left figure in Supplementary Fig. 3) or all nine (‘Completed vConv-based’; right figure in Supplementary Fig. 3) convolutional layers in the Basset network with vConv layer(s) and retrained it to obtain the AUROC value of the vConv-based Basset network, and finally (4) compared the AUROC values between the two types of networks. The details of parameter initialization (except for the vConv-exclusive parameters) and stochastic gradient descent-based optimization (with learning rate 0.005) were the same between the vConv-based and the original Basset networks (Supplementary Fig. 2 and 3).

Benchmark vConv-based networks on real-world motif discovery

Finally, we compared vConv-based networks with the canonical methods on the motif discovery problem. We followed the protocol proposed by Zhang et al. (Zhang, et al., 2015) to assess the accuracy of the extracted representative motifs based on the ENCODE CTCF ChIP-Seq datasets (ENCODE Project Consortium, 2012). In brief, for each motif discovery tool and each ChIP-Seq dataset, we located the motif-containing sequence fragments for candidate motifs discovered by this tool, checked whether they overlapped with the ChIP-Seq peaks with CisFinder, and reported the largest per-motif ratio of overlapping fragments to all fragments across all candidate motifs as the final accuracy of this tool on this dataset (see Supplementary Fig. 4 for more details). For the sake of demonstrating the advantage of vConv-based networks, here the model structure of the vConv-based network was chosen to be the one used in simulation (Supplementary Fig. 1).

RESULTS

vConv-based networks identify motifs more effectively than convolution-based networks

We first present direct comparisons between vConv-based and convolution-based networks based on multiple simulated datasets (see the Methods for more details). vConv-based networks performed statistically significantly better (with all Wilcoxon rank sum test's p-values < 1e-18 for the null hypothesis that the AUROC of the vConv-based network is equal to or smaller than that of the convolution-based network), and this improvement became larger when more motifs are introduced (2 motifs to 8 motifs in Fig. 2A) and with increased heterogeneity of the motif length (2 motifs v.s. TwoDiffMotif1/2/3 in Fig. 2A); notably, this superior performance gain of vConv-based networks applies to most datasets (Supplementary Fig. 5). In addition, vConv-based networks showed a smaller mean standard error of the AUROC than convolution-based networks across different hyper-parameter settings (Levene's test, p-value < 0.001 for all datasets; see Supplementary Table 2), suggesting that vConv-based networks also have a better robustness to hyper-parameters than convolution-based networks (Fig. 2B).

We further visualized the first layer of trained vConv-based and convolution-based networks on the most complex dataset (8 motifs), and found that vConv-based networks accurately recovered the underlying real motif from sequences and converge to the real length well (Supplementary Fig. 10). Of note, we found that the vConv-based networks successfully learned a motif (MA0234.1) which was missed by canonical convolution-based networks (Supplementary Fig. 10, second row).

These findings further compelled us to suspect that vConv-based networks will perform better than convolution-based networks on real-world cases possibly with combinatorial regulation. To test this hypothesis, we examined whether replacing convolutional layer(s) with vConv would improve performance of the following convolution-based networks: (1) DeepBind (Alipanahi, et al., 2015), a series of DNA-protein-binding classifiers trained for each of 504 ENCODE ChIP-Seq datasets; (2) Zeng et al.'s convolution-based networks (Zeng, et al., 2016), a series of structurally optimized variants of DeepBind models for each of 690 ChIP-Seq datasets; and (3) Basset (Kelley, et al., 2016), a complex DNase footprint predictor with nine convolutional layers (see Methods for more details). As expected, vConv-based networks showed a statistically significantly improved performance compared to their convolution-based network counterparts for all these three cases (Fig. 3 and Supplementary Fig. 6), with a few exceptions some of which can be attributed to small sample size of the input dataset (Supplementary Fig. 7A-B) and disappeared upon additional grid-search on kernel length (Supplementary Fig. 7C). Of note, stacking vConv layers for a complex convolution-based network like Basset can statistically significantly improve the performance with respect to both the original Basset network and the

single vConv-based network (Fig. 3C), well demonstrating its combined power with the popular hierarchical representation learning strategy in CNN application.

vConv-based networks discover motifs from real-world sequences more accurately and faster than canonical tools

We further evaluated vConv-based networks' performance for *ab initio* motif discovery. In brief, for a particular trained vConv-based network, we first selected kernels with corresponding dense layer weights higher than a predetermined baseline (defined as mean (all dense layer weights) - standard deviation (all dense layer weights)) and then extracted and aligned these kernels' corresponding segments to compute the representative PWM (Fig. 4A). We then compared the accuracy of recovering ChIP-Seq peaks by the vConv-based motif discovery and other motif discovery tools across all these ChIP-Seq datasets (ENCODE Project Consortium, 2012) (see the Methods for details).

Out of all 100 CTCF datasets, the vConv-based network outperformed DREME (Bailey, 2011) on all datasets, CisFinder (Sharov and Ko, 2009) on 95 datasets and MEME-ChIP (Machanick and Bailey, 2011) on 87 datasets (Fig. 4B). In addition, the vConv-based network ran much faster than DREME and MEME-ChIP on large datasets (more than 17 million base pairs finished in less than 30 minutes compared to hours by DREME and MEME-ChIP's; see Fig. 4C).

DISCUSSION

Inspired by the theoretical model (Supplementary Notes 1) for analyzing the influence of the kernel size in the convolutional layer, we designed and implemented a novel convolutional layer, vConv, which adaptively tunes the kernel lengths at run time. vConv could be readily integrated into multi-layer neural networks, as an "in-place replacement" of canonical convolutional layer. To facilitate its application in various fields, we have implemented vConv as a new type of convolutional layer in Keras (https://github.com/gao-lab/vConv).

In theory, a regular fixed-length convolution kernel with size larger than that of the *bona fide* motif, may be able to converge to a solution where the kernel captures the motif faithfully with zero padding everywhere else, just as what the vConv kernel does. Nevertheless, preliminary empirical evaluations suggest that such "perfect" solution could be very rare in reality with a general training strategy: none of the convolution-based network kernels similar to the ground truth motifs in the 8 motifs case (as determined by Tomtom; see Supplementary Fig. 10) contains more than 1 position whose sum of absolute elements are one order of magnitude more closer to 0 than other positions.

vConv is scalable considering the fact that its current design only adds a *O(ncl)* per *n*-kernel-convolutional layer (with kernel length *l* and number of channels *c*) replaced to the overall time complexity than of the canonical fixed-kernel convolutional layer, and that preliminary empirical evaluations suggest no evidence of convergence slowdown (as benchmarked in Supplementary Fig. 8 for the simulation case). While we notice that the current variable-length kernel strategy implemented by vConv could, in theory, be approximated by combing multiple fixed-length-kernel networks, we'd argue that such approximation would lead to serious inflation for the number of trainable parameters, and higher time cost for training and optimization. For example, to approximate a simple network with one vConv layer of *n* unmasked kernels with average length *l* (which needs $n*(4*l+2)+n = n*(4*l+3)$ parameters), the ensemble of *m* convolution-based networks would take $m*(n*(4*l)+n) = m*(n*(4*l+1))$ parameters, far larger than $n*(4*l+3)$ when *m* is of tens or hundreds - a popular choice in ensemble learning.

vConv-based networks show robustness with various hyper-parameters and initialization setups (Supplementary Table 2). We believe that the major vConv-exclusive components, Shannon loss (MSL) and boundary parameters, contribute to the increased robustness. Intuitively, such components could regularize the parameters to learn, making the overall objective function simpler (i.e., having fewer suboptima) and thus more robust. Consistently, we found out that the vConv-based network -- when equipped with MSL -- had a smaller variance of AUROC that is statistically significantly different from that of the vConv-based network without MSL in 3 out of all 7 simulation cases (Levene's test, $p<0.001$; Supplementary Table 3), suggesting that MSL does contribute to this increased robustness in certain conditions. Of interest, equipping MSL also improved the AUROC of the vConv-based network (Supplementary Fig. 9).

We note that the current theoretical analysis (see details in Supplementary Notes 1) reported above relies on prior knowledge of the real motif $\mathcal{M}$, which may not be feasible to obtain for real-world datasets. A possible workaround is to estimate the empirical null distribution of P_{real} over a set of randomly initialized PWMs and kernels and then derive the expectation of P_{real} over the given dataset. Such theoretical distribution, once established, could serve as a general background distribution for further development of motif scanners.

Moreover, although current motifs in convolution-based networks are automatically represented by PWMs, this might be an oversimplified representation of the genuine motifs, which may allow insertions and deletions within motifs (e.g., the HMM motifs from Pfam (El-Gebali, et al., 2019) and Rfam (Kalvari, et al., 2018)). While recurrent neural networks have been expected to be able to learn such motifs (Liu, 2017; Liza and Grzes, 2019; Min, et al., 2019; Vazhayil and KP, 2018), the interpretation of such models is still challenging. A promising alternative would be to use the CNN framework to learn complex motifs directly, as demonstrated by a recently developed CNN model HOCNNLB (Zhang, et al., 2019) which can learn first-order Markov

models by re-encoding input sequences. Combining the variable-length nature of vConv and such complex-motif-capturing models might be much more useful for mining biological motifs.

ACKNOWLEDGEMENT

The authors would like to thank Drs. Cheng Li, Letian Tao, Minghua Deng, Zemin Zhang, Jian Lu and Liping Wei at Peking University for their helpful comments and suggestions during the study. The analysis was supported by the High-performance Computing Platform of Peking University, and we thank Dr. Chun Fan and Yin-Ping Ma for their assistance during the analysis.

DATA AVAILABILITY

A Keras-based implementation of vConv is available at GitHub (https://github.com/gao-lab/vConv) for academic usage, with data and script required to reproduce figures and tables in this study available at https://github.com/gao-lab/vConv-Figures_and_Tables.

FUNDING

This work was supported by funds from the National Key Research and Development Program [grant number 2016YFC0901603]; the China 863 Program [grant number 2015AA020108]; the State Key Laboratory of Protein and Plant Gene Research; the Beijing Advanced Innovation Center for Genomics (ICG) at Peking University; and in part by the National Program for the Support of Top-notch Young Professionals [to G.G.].

Conflict of interest statement. None declared.

REFERENCES

Achar, A. and Sætrom, P. RNA motif discovery: a computational overview. *Biology direct* 2015;10(1):61.

Alipanahi, B., Delong, A., Weirauch, M.T. and Frey, B.J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature biotechnology* 2015;33(8):831.

Angermueller, C., Lee, H.J., Reik, W. and Stegle, O. DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome biology* 2017;18(1):67.

Bailey, T.L. DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics* 2011;27(12):1653-1659.

Bailey, T.L., Williams, N., Misleh, C. and Li, W.W. MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res* 2006;34(Web Server issue):W369-373.

Blencowe, B.J. Exonic splicing enhancers: mechanism of action, diversity and role in human genetic diseases. *Trends in biochemical sciences* 2000;25(3):106-110.

Chollet, F., others. Keras. 2015.

Das, M.K. and Dai, H.-K. A survey of DNA motif finding algorithms. In, *BMC bioinformatics*. Springer; 2007. p. S21.

Ding, J., Dhillon, V., Li, X. and Hu, H. Systematic discovery of cofactor motifs from ChIP-seq data by SIOMICS. *Methods* 2015;79-80:47-51.

Ding, J., Hu, H. and Li, X. SIOMICS: a novel approach for systematic identification of motifs in ChIP-seq data. *Nucleic Acids Res* 2014;42(5):e35.

Ding, Y., Li, J.-Y., Wang, M. and Gao, G. An exact transformation of convolutional kernels enables accurate identification of sequence motifs. *bioRxiv* 2018:163220.

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S.R., Luciani, A., Potter, S.C., Qureshi, M., Richardson, L.J., Salazar, G.A., Smart, A., Sonnhammer, E.L.L., Hirsh, L., Paladin, L., Piovesan, D., Tosatto, S.C.E. and Finn, R.D. The Pfam protein families database in 2019. *Nucleic Acids Res* 2019;47(D1):D427-D432.
ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012;489(7414):57.
Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In: Yee Whye, T. and Mike, T., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Proceedings of Machine Learning Research: PMLR; 2010. p. 249--256.
Han, S., Meng, Z., Li, Z., O'Reilly, J., Cai, J., Wang, X. and Tong, Y. Optimizing filter size in convolutional neural networks for facial action unit recognition. In, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018. p. 5070-5078.
Ikebata, H. and Yoshida, R. Repulsive parallel MCMC algorithm for discovering diverse motifs from large sequence sets. *Bioinformatics* 2015;31(10):1561-1568.
Jia, C., Carson, M.B., Wang, Y., Lin, Y. and Lu, H. A new exhaustive method and strategy for finding motifs in ChIP-enriched regions. *PLoS One* 2014;9(1):e86044.
Kadonaga, J.T. Perspectives on the RNA polymerase II core promoter. *Wiley Interdisciplinary Reviews: Developmental Biology* 2012;1(1):40-51.
Kalvari, I., Argasinska, J., Quinones-Olvera, N., Nawrocki, E.P., Rivas, E., Eddy, S.R., Bateman, A., Finn, R.D. and Petrov, A.I. Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Res* 2018;46(D1):D335-D342.
Kelley, D.R., Reshef, Y.A., Bileschi, M., Belanger, D., McLean, C.Y. and Snoek, J. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research* 2018;28(5):739-750.
Kelley, D.R., Snoek, J. and Rinn, J.L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research* 2016;26(7):990-999.
Khan, A., Fornes, O., Stigliani, A., Gheorghe, M., Castro-Mondragon, J.A., van der Lee, R., Bessy, A., Cheneby, J., Kulkarni, S.R. and Tan, G. JASPAR 2018: update of the open-access database of transcription factor binding profiles and its web framework. *Nucleic acids research* 2017;46(D1):D260-D266.
Koo, P.K. and Eddy, S.R. Representation learning of genomic sequence motifs with convolutional neural networks. *PLoS computational biology* 2019;15(12):e1007560.
Kulakovskiy, I.V., Boeva, V.A., Favorov, A.V. and Makeev, V.J. Deep and wide digging for binding motifs in ChIP-Seq data. *Bioinformatics* 2010;26(20):2622-2623.
Kulakovskiy, I.V. and Makeev, V.J. DNA sequence motif: a jack of all trades for ChIP-Seq data. In, *Advances in protein chemistry and structural biology*. Elsevier; 2013. p. 135-171.
Lambert, S.A., Jolma, A., Campitelli, L.F., Das, P.K., Yin, Y., Albu, M., Chen, X., Taipale, J., Hughes, T.R. and Weirauch, M.T. The Human Transcription Factors. *Cell* 2018;175(2):598-599.
Lihu, A. and Holban, S. A review of ensemble methods for de novo motif discovery in ChIP-Seq data. *Brief Bioinform* 2015;16(6):964-973.
Liu, B., Yang, J., Li, Y., McDermaid, A. and Ma, Q. An algorithmic perspective of de novo cis-regulatory motif finding based on ChIP-seq data. *Brief Bioinform* 2018;19(5):1069-1081.
Liu, X. Deep recurrent neural network for protein function prediction from sequence. *arXiv preprint arXiv:1701.08318* 2017.
Liza, F.F. and Grzes, M. Relating RNN layers with the spectral WFA ranks in sequence modelling. *Association for Computational Linguistics* 2019:24-33.
Maaskola, J. and Rajewsky, N. Binding site discovery from nucleic acid sequences by discriminative learning of hidden Markov models. *Nucleic Acids Res* 2014;42(21):12995-13011.
Machanick, P. and Bailey, T.L. MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics* 2011;27(12):1696-1697.
Min, S., Park, S., Kim, S., Choi, H.-S. and Yoon, S. Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information. *arXiv preprint arXiv:1912.05625* 2019.
Reiter, F., Wienerroither, S. and Stark, A. Combinatorial function of transcription factors and cofactors. *Curr Opin Genet Dev* 2017;43:73-81.
Sharov, A.A. and Ko, M.S. Exhaustive search for over-represented DNA sequence motifs with CisFinder. *DNA Res* 2009;16(5):261-273.
Stormo, G.D. DNA motif databases and their uses. *Current protocols in bioinformatics* 2015;51(1):2.15. 11-12.15. 16.
Thomas-Chollier, M., Herrmann, C., Defrance, M., Sand, O., Thieffry, D. and van Helden, J. RSAT peak-motifs: motif analysis in full-size ChIP-seq datasets. *Nucleic Acids Res* 2012;40(4):e31.
Thomson, D.W. and Dinger, M.E. Endogenous microRNA sponges: evidence and controversy. *Nature*

Reviews Genetics 2016;17(5):272.
Tran, N.T. and Huang, C.H. A survey of motif finding Web tools for detecting binding site motifs in ChIP-Seq data. *Biol Direct* 2014;9(1):4.
Vazhayil, A. and KP, S. DeepProteomics: Protein family classification using Shallow and Deep Networks. *arXiv preprint arXiv:1809.04461* 2018.
Wang, M., Tai, C., E, W. and Wei, L. DeFine: deep convolutional neural networks accurately quantify intensities of transcription factor-DNA binding and facilitate evaluation of functional non-coding variants. *Nucleic acids research* 2018;46(11):e69-e69.
Yin, W. and Schütze, H. Multichannel variable-size convolution for sentence classification. *arXiv preprint arXiv:1603.04513* 2016.
Zambelli, F., Pesole, G. and Pavesi, G. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. *Briefings in Bioinformatics* 2013;14(2):225-237.
Zeiler, M.D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* 2012.
Zeng, H., Edwards, M.D., Liu, G. and Gifford, D.K. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics* 2016;32(12):i121-i127.
Zhang, B., Gunawardane, L., Niazi, F., Jahanbani, F., Chen, X. and Valadkhan, S. A novel RNA motif mediates the strict nuclear localization of a long noncoding RNA. *Molecular and cellular biology* 2014;34(12):2318-2329.
Zhang, J., Peng, W. and Wang, L. LeNup: learning nucleosome positioning from DNA sequences with improved convolutional neural networks. *Bioinformatics* 2018;34(10):1705-1712.
Zhang, Q., Zhu, L. and Huang, D.S. High-Order Convolutional Neural Network Architecture for Predicting DNA-Protein Binding Sites. *IEEE/ACM Trans Comput Biol Bioinform* 2019;16(4):1184-1192.
Zhou, J., Theesfeld, C.L., Yao, K., Chen, K.M., Wong, A.K. and Troyanskaya, O.G. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics* 2018;50(8):1171.
Zucchelli, S., Cotella, D., Takahashi, H., Carrieri, C., Cimatti, L., Fasolo, F., Jones, M., Sblattero, D., Sanges, R. and Santoro, C. SINEUPs: A new class of natural and synthetic antisense long non-coding RNAs that activate translation. *RNA biology* 2015;12(8):771-779.

Table and Figure Legends

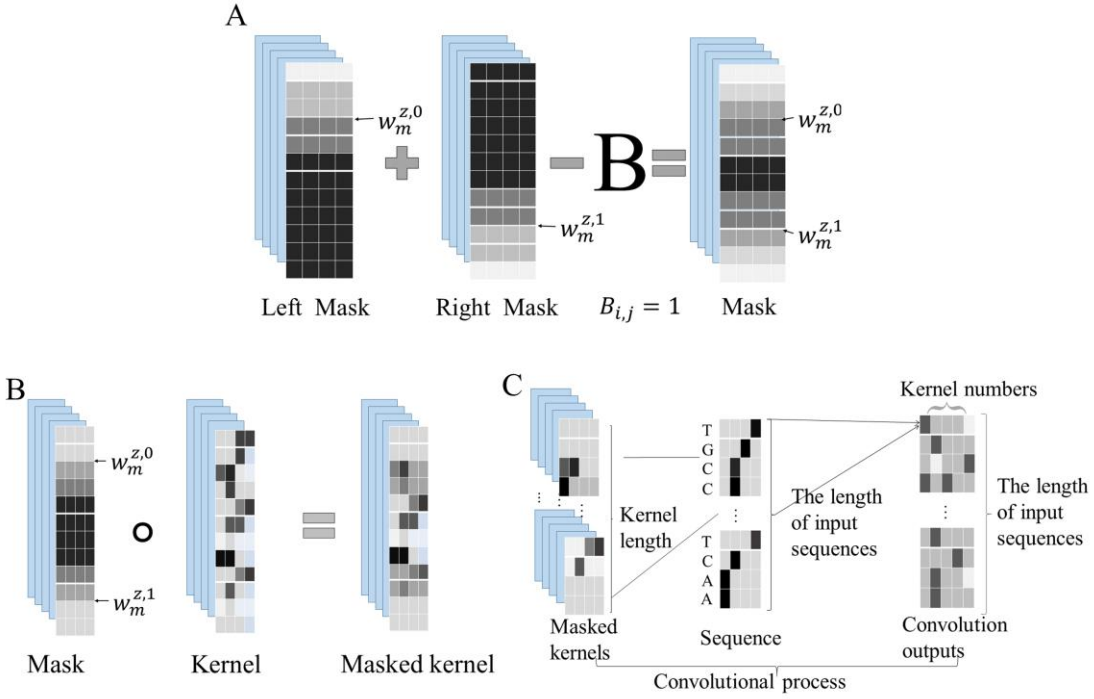

Fig. 1. The design of vConv. Firstly, a mask matrix is constructed from two sigmoid functions (A). Then vConv "masks" this mask matrix onto the kernel using the Hadamard product (B), and treats the masked kernel as an ordinary kernel to convolve the input sequence (C).

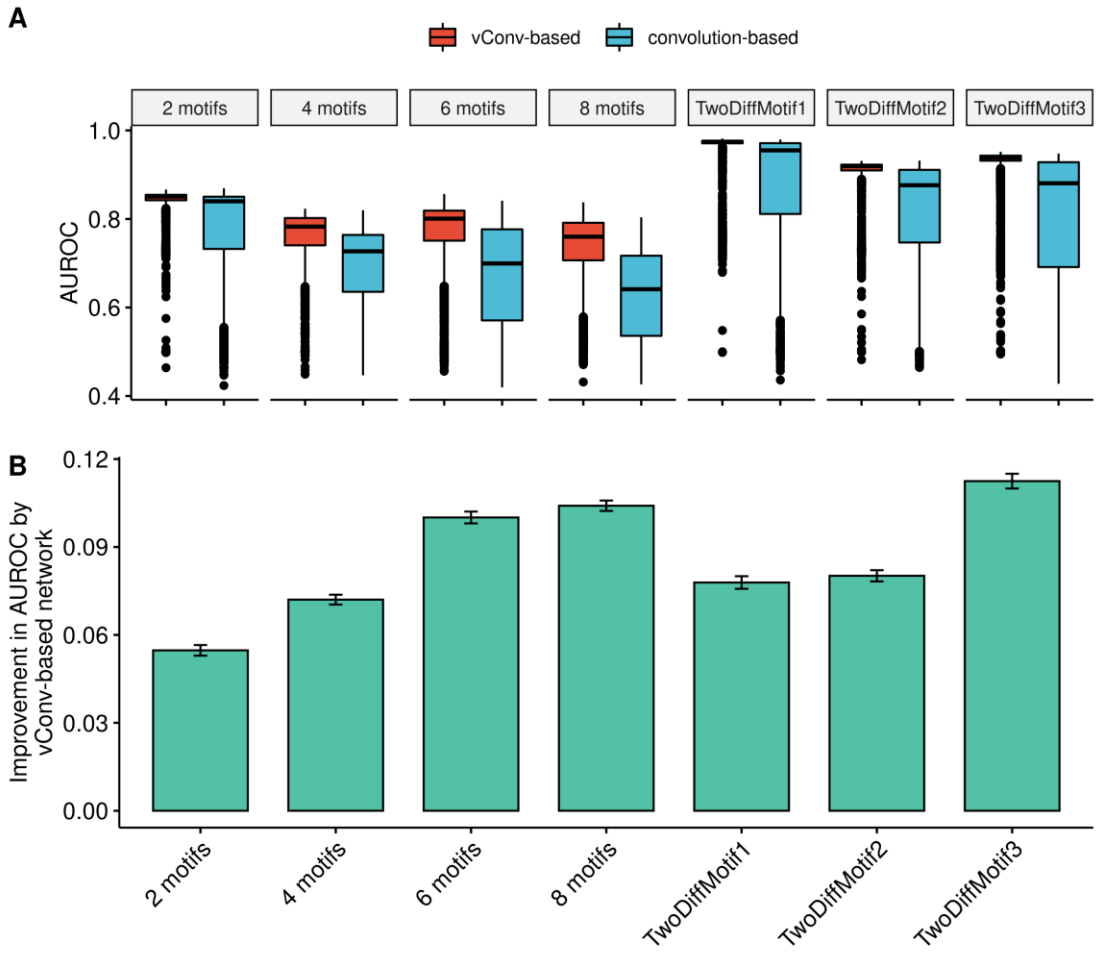

Fig. 2. vConv-based networks outperformed canonical convolution-based networks for motifs of different lengths on simulation datasets. (A) shows the comparison of overall AUROC distribution for each case between the vConv-based and the convolution-based networks. (B) shows the mean and standard error of per-hyper-parameter-setting AUROC difference between the vConv-based network and the convolution-based network.

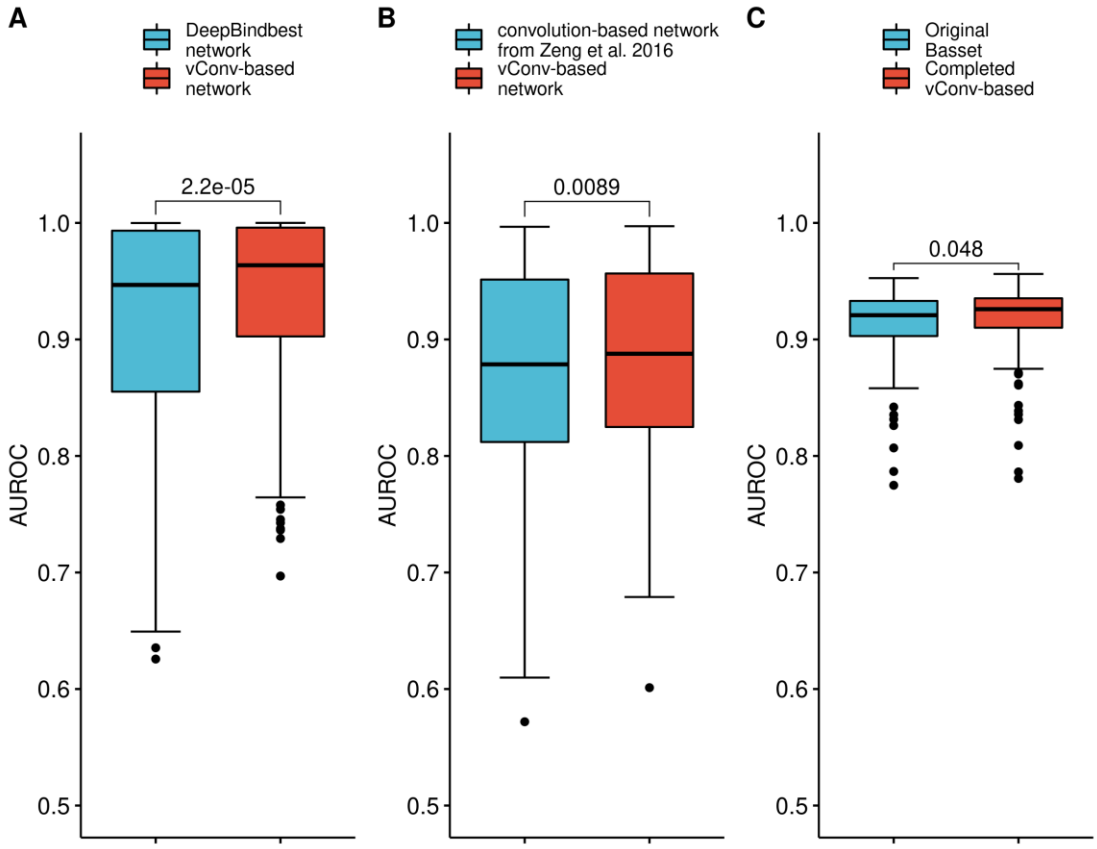

Fig. 3. vConv-based networks outperformed convolution-based networks on real-world cases. (A), comparison of AUROC between vConv-based networks and DeepBindbest networks for DeepBind (see also Supplementary Fig. 6 for comparison with other DeepBind networks). (B), comparison of AUROC between vConv-based networks and networks from Zeng et al. . (C), comparison of AUROC between vConv-based networks with 'Completed vConv-based', the vConv-based network with all convolutional layers of Basset network replaced by vConv (right figure in Supplementary Fig. 3; see Methods for more details) and networks from Basset. All p-values shown are from Wilcoxon rank sum test, single-tailed, with the null hypothesis that the AUROC of the vConv-based network is equal to or smaller than that of the convolution-based network. See Methods for more details of each specific network.

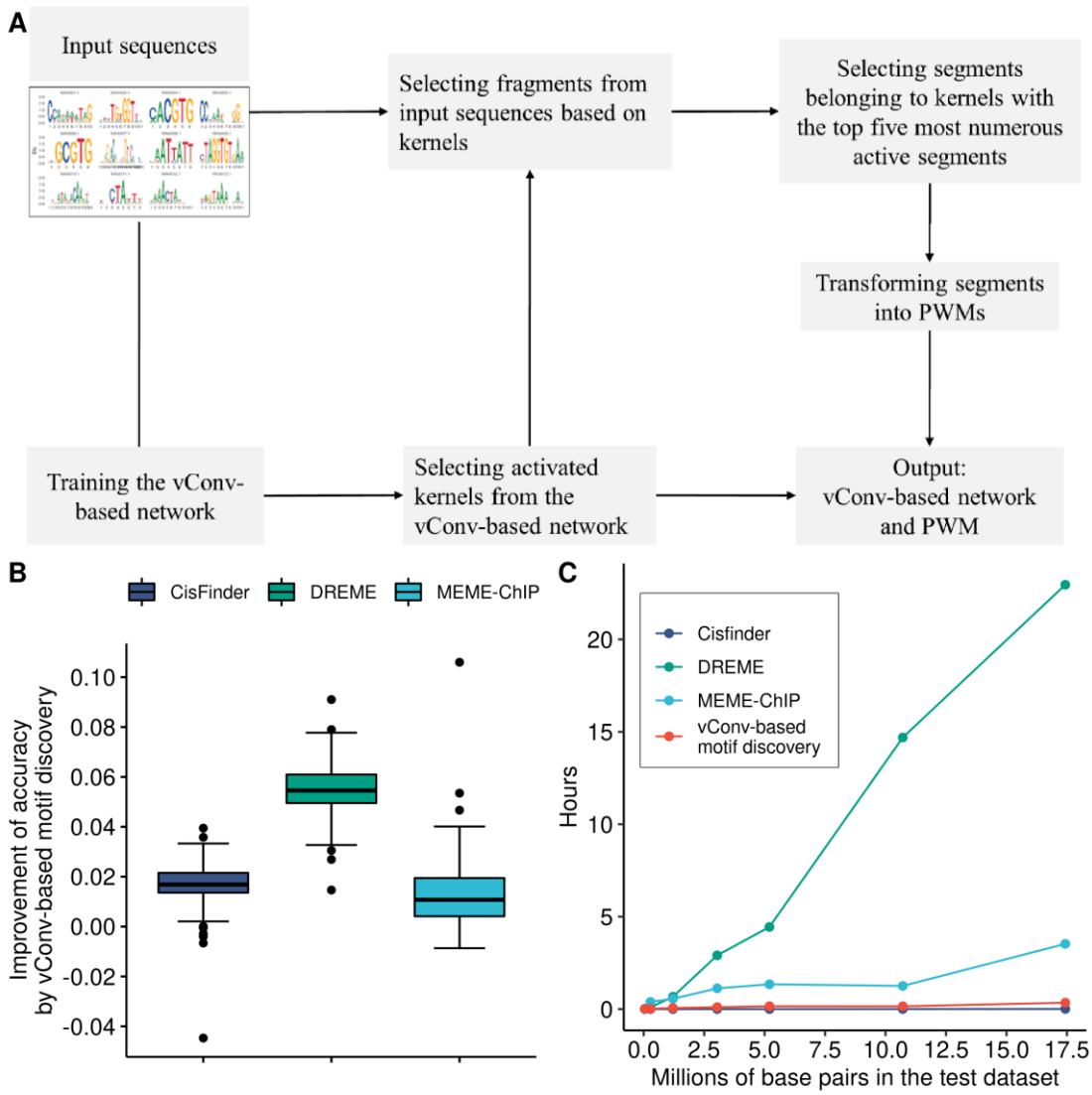

Fig. 4. vConv-based networks discover motifs more accurately and faster. (A) shows the process of calling a representative motif from a trained vConv-based network. (B) shows the difference in accuracy, defined as the accuracy of vConv-based motif discovery minus the accuracy of the motif discovery algorithm shown on the x-axis on the same dataset. MEME (Bailey, et al., 2006) failed to complete the discovery within a reasonable time (~50% datasets remained unfinished even after running for 1.5 weeks with 2,000 cores, amounting to 504,000 CPU hours) for these datasets, and its results were thus not listed here. (C) shows the time cost of each motif discovery algorithm as a function of millions of base pairs in the dataset tested.

Hyper-parameter	Value range space
Kernel number	{64, 96, 128}
Kernel length	{6, 8, 10, 12, 14, 16, 18, 20}

Random Seeds	16 different seeds
Momentum in AdaDelta optimizer	{0.9, 0.99, 0.999}
Delta in AdaDelta optimizer	{1e-4, 1e-6, 1e-8}

Table 1. Hyper-parameter space for the comparison between vConv-based networks and convolution-based networks on simulation datasets.

Dataset	Motif composition
2 motifs	MA0234.1 (length=6), MA0963.1 (length=8)
4 motifs	MA0234.1 (length=6), MA0963.1 (length=8), MA0626.1 (length=10), MA0667.1 (length=10)
6 motifs	MA0234.1 (length=6), MA0963.1 (length=8), MA0626.1 (length=10), MA0667.1 (length=10), MA1146.1 (length=15), MA1147.1 (length=15)
8 motifs	MA0234.1 (length=6), MA0963.1 (length=8), MA0626.1 (length=10), MA0667.1 (length=10), MA1146.1 (length=15), MA1147.1 (length=15), MA0009.2 (length=16), MA0470.1 (length=11)
TwoDiffMotif1	MA0138.2 (length=21), MA0157.2 (length=8)
TwoDiffMotif2	MA1046.1 (length=9), MA1148.1 (length=18)
TwoDiffMotif3	MA0326.1 (length=8), MA0556.1 (length=15)

Table 2. Motifs used to generate each dataset. All motifs were derived from JASPAR (Khan, et al., 2017).