Bioinformatics, 35(20), 2019, 3937–3943
doi: 10.1093/bioinformatics/btz194
Advance Access Publication Date: 27 March 2019
Original Paper

Sequence analysis

Neural networks with circular filters enable data efficient inference of sequence motifs

Christopher F. Blum* and **Markus Kollmann**

Institute for Mathematical Modeling of Biological Systems, Heinrich-Heine University of Düsseldorf, Düsseldorf 40225, Germany

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on June 13, 2018; revised on November 15, 2018; editorial decision on February 11, 2019; accepted on March 26, 2019

Abstract

Motivation: Nucleic acids and proteins often have localized sequence motifs that enable highly specific interactions. Due to the biological relevance of sequence motifs, numerous inference methods have been developed. Recently, convolutional neural networks (CNNs) have achieved state of the art performance. These methods were able to learn transcription factor binding sites from ChIP-seq data, resulting in accurate predictions on test data. However, CNNs typically distribute learned motifs across multiple filters, making them difficult to interpret. Furthermore, networks trained on small datasets often do not generalize well to new sequences.

Results: Here we present circular filters, a novel convolutional architecture, that convolves sequences with circularly permutated variants of the same filter. We motivate circular filters by the observation that CNNs frequently learn filters that correspond to shifted and truncated variants of the true motif. Circular filters enable learning of full-length motifs and allow easy interpretation of the learned filters. We show that circular filters improve motif inference performance over a wide range of hyperparameters as well as sequence length. Furthermore, we show that CNNs with circular filters in most cases outperform conventional CNNs at inferring DNA binding sites from ChIP-seq data.

Availability and implementation: Code is available at https://github.com/christopherblum.

Contact: christopher.blum@hhu.de

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

1 Introduction

A fundamental property of biological macromolecules such as DNA, RNA and proteins is their ability to generate highly specific interactions. These high specificities are often associated with localized motifs in the primary structure of these macromolecules. Intricate motif variations, indirect effects on binding specificity and noisy data make motif inference from high-throughput data a challenging task (Berger and Bulyk, 2009; Kidder *et al.*, 2011; Man and Stormo, 2001; Rohs *et al.*, 2010; Weirauch *et al.*, 2013).

As interactions among biomolecules participate in almost all processes that have biotechnological or biomedical relevance, numerous methods have been developed to infer motifs and binding specificities from high-throughput data sources (Weirauch *et al.*, 2013). These methods range from derivatives of nucleotide frequency counting procedures such as *k*-mer and position weight matrix (PWM) methods to, most recently, convolutional neural networks (CNNs) (Alipanahi *et al.*, 2015; Zeng *et al.*, 2016). Unlike PWM methods, CNNs do not require aligned input sequences because they convolve input sequences with sliding weight matrices called filters. Although this sliding operation resembles *k*-mer methods, CNNs do not rely on predefined *k*-mers. Instead, they learn the weight matrices from data through an optimization process that involves predicting associated sequence features, such as ChIP-seq read counts.

CNNs have achieved state of the art performance at predicting fluorescence intensities derived from protein binding microarray data (Alipanahi *et al.*, 2015). It has been reported, however, that training these models with gradient descent methods is sensitive to weight initialization, which can impair generalization ability (Zeng *et al.*, 2016). A common work-around is to use multiple filters, but this has a draw-back: using CNNs with multiple filters results in motif information being distributed across the filters, preventing immediate interpretation. Moreover, deep neural networks typically require large amounts of training data. Due to the noisiness of protein binding microarray and ChIP-seq derived data, the number of positive training examples is often limited (positive examples are sequences that can be assumed to contain a certain motif). A CNN-based motif inference method, DeepBind, uses data augmentation to artificially increase the number of training examples, but it still relies on a sufficiently large number of positive training examples (Alipanahi *et al.*, 2015; Simard *et al.*, 2003).

Here, we present a novel CNN architecture called circular filters that enables efficient data utilization and easy interpretation of the learned filters. We show that filters of conventional CNNs often contain shifted and truncated motifs and argue that these correspond to local optima: during gradient descent optimization, the motif ‘develops’ in the CNN filter from a random position, and it can happen that only a truncated motif can be learned because a filter edge is reached. We then introduce circular filters as a natural solution to this problem. Circular filters are composed of all circularly permutated and weighted variants of the same underlying filter (Fig. 1). If one filter variant has learned a truncated motif, there is another filter variant in which the full motif can be learned (given that the filter is as long as the motif). We emphasize that the algorithm learns to ignore non-optimal filter variants so that after training, only one filter mainly contributes to the objective. This means that there are no performance drawbacks in using circular filters compared to normal convolution, but circular filters help to escape local optima.

We show that circular filters improve motif inference over a wide range of hyperparameter settings, allowing better inference from long sequences and when data are scarce. We demonstrate that both CNNs with and without circular filters can infer diverse motifs such as ‘AACCGT’ easier than less diverse motifs such as ‘AAAAAC’. Finally, we show that a CNN with circular filters yields accurate predictions for ChIP-seq derived data, performing at least as well as current state of the art algorithms for motif inference.

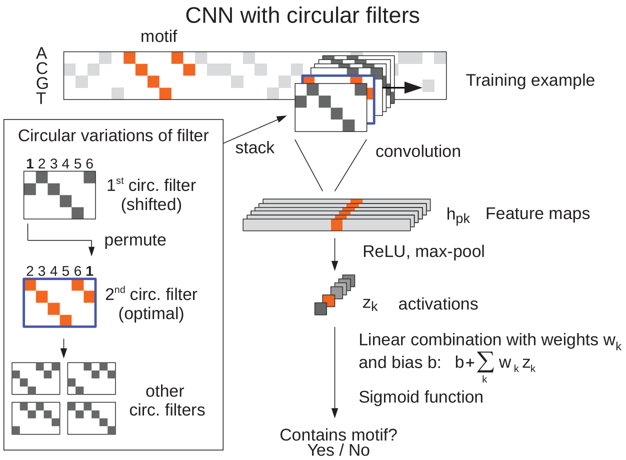

Fig. 1. Illustration of a CNN with circular filters. Circular filters consist of all circularly permutated variants of the same underlying filter. Convolution with one circular filter of length *N* yields *N* feature maps that are linearly combined in a subsequent layer. During training, the algorithm learns to select only one filter variant through the linear combination

2 Materials and Methods

2.1 Models

One-hot coding. To enable processing by CNNs, sequences were converted to image-like representations by one-hot coding. Specifically, a sequence $X = X_1, \ldots, X_L$ of length L with elements X_j coming from an ordered set with finite cardinality $X_j \in D, |D| = N$ can be represented as a $N \times L$ one-hot coding matrix S with elements S_{ij} that are equal to 1 if X_j is the i-th element in D and 0 otherwise.

Network architectures. We modeled the class-conditional probability $p(C_{\text{motif}}|S)$ that a sequence with one-hot coding S belongs to class C_{motif} (‘contains motif’) or not with different neural networks. These were: a conventional CNN, three different CNNs with circular filters, and a network with a fully connected layer. Input to the networks were $N \times L$ one-hot codings S, where L was the length of the sequence and $N = 4$ was the number of features (4 nt). In the following, the $N \times L_F$ matrix F is a convolutional filter of length L_F, w_k and w are weights and b is a bias, and $\sigma(x) = 1/(1+e^{-x})$ denotes the sigmoid function.

The three CNN architectures with circular filters that were investigated all convolved the input S with circular filters without padding, followed by a Rectifying linear unit (ReLU) and max-pooling to yield activations z_k (Fig. 1). That is, they shared the following chain of functions:

$$h_{pk} = \sum_{i=1}^{L_f}\sum_{j=1}^{N} S_{j,(i+p-1)}F_{jm}, \quad \text{with} \quad m = (i+k-1)\,(\mathrm{mod}\, L_F),$$

$$a_{pk} = \max(0, h_{pk}) \;\; (\text{ReLU}),$$

$$z_k = \max\{a_{p,1}, \ldots, a_{p,L-L_F+1}\} \;\; (\text{max-pooling}).$$

The three CNN architectures with circular filters then differed in how the activations z_k were mapped onto the class-conditional probabilities. Specifically, the *CNN with circular filters* used a weighted sum of the activations

$$p(C_{\text{motif}}|S) = \sigma\left(b + \sum_{k=1}^{L_F} w_k z_k\right).$$

Whereas the *CNN with circular filters and sum of the activations* simply summed up all activations

$$p(C_{\text{motif}}|S) = \sigma\left(b + w\sum_{k=1}^{L_F} z_k\right).$$

Finally, the *CNN with circular filters and max-out* (Goodfellow *et al.*, 2013) only used the largest of all activations

$$p(C_{\text{motif}}|S) = \sigma(b + w\,\max\{z_1, \ldots, z_{L_F}\}).$$

The CNN without circular filters (Fig. 2) convolved the input sequence S with one ordinary filter, then applied ReLU and max-pooling and mapped the activations z to the class-conditional probabilities

$$h_p = \sum_{i=1}^{L_f}\sum_{j=1}^{N} S_{j,(i+p-1)}F_{ji},$$

$$a_p = \max(0, h_p) \;\; (\text{ReLU}),$$

$$z = \max\{a_1, \ldots, a_{L-L_F+1}\} \;\; (\text{max-pooling}),$$

$$p(C_{\text{motif}}|S) = \sigma(wz + b).$$

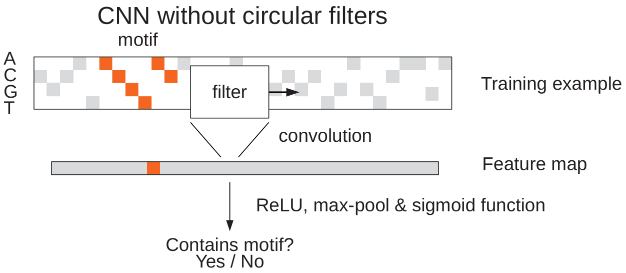

Fig. 2. Illustration of a CNN without circular filters. The network is trained to discriminate between sequences with and without motifs in a supervised manner

The fully connected network had the following chain of functions. Here, M is a $(N \cdot L) \times (L - L_F + 1)$ matrix, and S was reshaped into a vector of length $N \cdot L$

$$h_p = \sum_{i=1}^{N \cdot L} S_i M_{ip},$$

$$a_p = \max(0, h_p),$$

$$z = \max\{a_1, \ldots, a_{L-L_F+1}\},$$

$$p(C_{\text{motif}}|S) = \sigma(wz + b).$$

Extracting learned motifs from circular filters. A circular filter of length N consists of N circularly permutated variants of the same filter. After model training, each of these variants can in principle contain the correct motif, but the one with the largest associated weight w_k for linearly combining the activations is the most likely to contain the correct motif. We used this rationale to extract predicted motifs from trained circular filter variants.

Model training. Models were trained by minimizing the cross-entropy between network outputs and sequence labels using either mini-batch stochastic gradient descent or Stochastic Gradient Langevin Dynamics (SGLD) (Welling and Teh, 2011), depending on the experiment. When SGLD was used, the magnitude of the noise injected into the gradients was scaled by a factor γ, resulting in $\gamma\sqrt{\epsilon}\mathcal{N}(0,1)$ as injected noise, with ϵ as learning rate. Mini-batches were balanced, that is, they contained roughly equal numbers of positive and negative examples, with negative examples created either by random shuffling or dinucleotide shuffling of the positive sequences (described in Section 2.2).

DeepBind settings. The default DeepBind training parameter settings were used, except for an override of the filter length and the number of filters. Furthermore, the original routine for generating negative examples *de novo* was replaced by a routine to load the negative examples from the hard drive. To obtain a platform-independent instance of DeepBind, DeepBind was run in a Docker container using Nvidia Docker (Merkel, 2014). This was necessary because former attempts towards a platform-independent implementation were still depending on a certain CUDA-version, which in turn must be compatible with the graphic's card's chip architecture (Nickolls *et al.*, 2008; Zeng *et al.*, 2016).

Performance measures All algorithms assigned a score to each sequence that indicated the predicted likelihood that a sequence contained a motif. Consequently, we used either AUROC values or accuracies calculated based on test datasets as performance measures.

2.2 Data

Synthetic data. Synthetic data consisted of labeled positive and negative training examples (sequences with and without the desired motif, respectively). Sequences had a length of 40 nt if not mentioned otherwise explicitly. Positive training examples were created by first creating a random nucleotide background and then placing the desired motif at a random position within the sequence. Negative training examples were created by randomly drawing from the positive sequences (with replacement) and then randomly shuffling the nucleotide order. This procedure ensures that models cannot discriminate positive and negative examples based on their nucleotide composition.

ENCODE data. The pre-processed ENCODE datasets published by the DeepBind authors (http://tools.genes.toronto.edu/deepbind/nbtcode/nbt3300-supplementary-software.zip) were used for training and testing (Alipanahi *et al.*, 2015). In short, their datasets are derived from transcription factor ChIP-seq experiment data (http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeAwgTfbsUniform/) published by the Encyclopedia of DNA Elements (ENCODE) Consortium (Dunham *et al.*, 2012). These ChIP-seq datasets comprise of mapped sequence reads and read numbers for the most significant peaks. Regions around read peaks (starting 50 nt before and ending 50 nt after each peak) were extracted and sorted according to the read number in descending order. Since sequences with high read numbers are more likely to contain a binding site for the respective transcription factor, these sequences were used as positive examples. The DeepBind authors then used the first 500 even-numbered sequences as positive test data and all remaining sequences as positive training data. Then, dinucleotide shuffling was used to create negative examples based on the positive examples for both training and test datasets (Altschul and Erickson, 1985). Dinucleotide shuffling maintains the dinucleotide frequency, which can be necessary to prevent models from learning to discriminate positive and negative examples solely based on dinucleotide frequency (for example, the number of CG dinucleotides in naturally occurring sequences can be significantly lower than the frequency of other dinucleotides).

Here, we used the original test datasets published by the DeepBind authors ('*_B.seq.gz' files) for model testing (Alipanahi *et al.*, 2015). To create positive training examples, we used the top 500 sequences from the training datasets ('*_AC.seq.gz' files) (Alipanahi *et al.*, 2015). Negative training examples were created by randomly drawing from these 500 positive sequences (with replacement) and then shuffling the nucleotide order via dinucleotide shuffling. Negative training data were saved to a hard drive to be able to provide all models with exactly the same training data.

2.3 Experiments

How frequently do CNN architectures learn shifted motifs? Synthetic datasets were generated for all 4096 6-mers. Each dataset consisted of either 5 or 100 positive and 10^4 negative examples. Two network architectures were investigated: the CNNs with and without circular filters (Figs 1 and 2, respectively). Models were trained for 10^4 steps at a learning rate of 0.01.

The learned filters were then used to predict the most likely motif based on the nucleotides indicated by largest weight at each filter position. These predictions were then compared to the original 6-mer motifs. In this comparison, it was checked if the learned filters corresponded to shifted and truncated versions of the original motif. Specifically,

- shifts –3, –2 and –1: it was checked if the l rightmost nucleotides of the filter were equal to the l leftmost nucleotides of the motif hidden in the sequences, with $l \in 3, 4, 5$,
- shift 0: it was checked if all 6 nt of the filter were equal to the complete motif hidden in the sequences

- shifts 1, 2, 3: it was checked if the l leftmost nucleotides of the filter were equal to the l rightmost nucleotides of the motif hidden in the sequences, with $l \in 3, 4, 5$.

To test whether the number of times k with which shifted motifs were learned from the $n = 4096$ datasets, we used the Binomial distribution $B_{n,p}(k)$, where $p = 1/4^l$ is the probability to learn a motif of length l by chance. Since some of the resulting p-values were too small to be calculated with standard statistical programs, the smallest k necessary for a p-value less than or equal to 10^{-5}, k_{signif}, was calculated instead. For example, the value of the leftmost bar in the barplot in Figure 3b at $0.033 \approx 135/4096$ is significant because a shift of –3 was observed 135 times and k_{signif} was 100. Values for all k_{signif} can be obtained from Supplementary Table S1.

What is the effect of sequence length on motif inference? To investigate the effect of sequence length on motif inference, synthetic datasets were generated for 100 random 6-mer motifs (Fig. 3d; a list of the 100 motifs is given in Supplementary Table S2). For each motif, eight training and test datasets were generated, corresponding to sequence lengths between 10 and 80. Datasets contained either 10, 100 or 10^5 positive and 10^5 negative training examples. CNNs with and without circular filters as well as a neural network with a fully connected layer and were trained on all datasets. Models were trained for 10^4 steps at learning rate 0.1, and prediction accuracies on test datasets (1000 positive and negative examples) were calculated, and the median accuracies over all 100 random 6-mers were used to compare the architectures.

Does motif composition affect motif inference using CNNs? To assess whether the composition of a motif affects its inference with CNNs (Fig. 3e and f), we compared how well each of all 4096 6-mers could be inferred with both the CNN with and without circular filter. Synthetic datasets were generated for all 4096 6-mers. Then, models were trained on these datasets ten times (‘single run’), and it was checked how often the motifs had been inferred correctly. In addition to this, ten models were trained ten times each, and it was checked how often the best-performing among the ten models had inferred the motif correctly (‘best out of ten runs’). Taking the best out of several models essentially corresponds to a re-initialization and re-training of the model weights. Models were trained for 5000 steps at learning rate 0.01, and datasets had 100 positive and 10^4 negative examples. We then used the frequency of the most frequent letter as a measure for motif composition to correlate motif composition with the rate of correct inference of that motif. For example, in the motif ‘AACCGT’, the frequency of the most frequent letter is 2 (both ‘A’ and ‘C’ occur twice in ‘AACCGT’), whereas in the motif ‘AAAAAC’, it is 5. Differences in the relative frequency with which motifs were inferred correctly were tested using Mann-Whitney-U tests.

How do hyperparameters affect motif inference with CNNs? To investigate the effect of circular filters on motif inference for different hyperparameter settings (Supplementary Table S3), we trained and tested CNNs with and without circular filters on simulated data for a variety (grid) of hyperparameter combinations. We then tested for which hyperparameter settings utilization of circular filters significantly improved performance. Values of 0, 0.1, 1.0, 10 or 100, multiplied by $\sqrt{N_{\text{train}}}$, were used to scale the variance of the noise injected by SGLD. Here, N_{train} is the total number of training examples. Models were trained at a learning rate of $0.01/N_{\text{train}}$, and values of 0, 0.1, 1.0, 10 or 100, multiplied by N_{train}, were used as L_2-regularization strength. Filter lengths were either 3, 4, 6 or 12 nt, and either 5, 50 or 500 positive training examples were used for training. To investigate the effect of the linear combination layer of

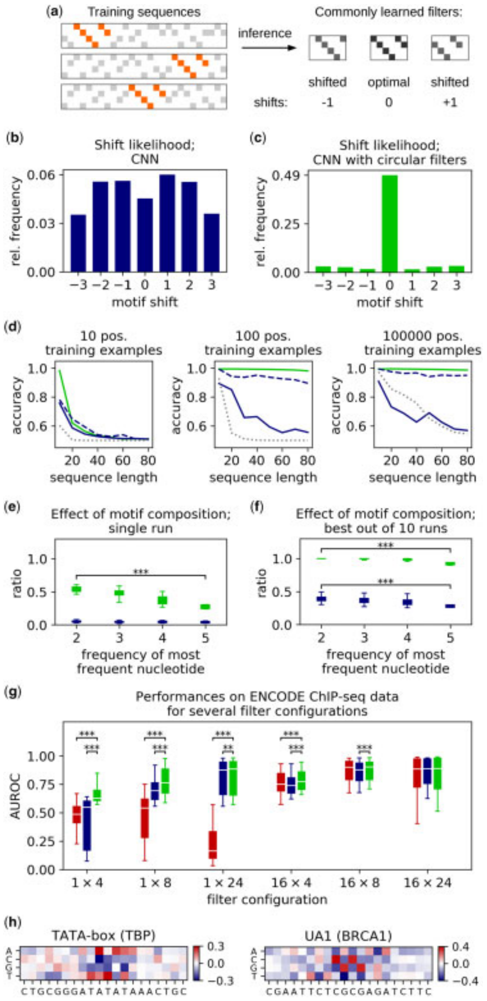

Fig. 3. (**a**) Illustration of truncated motifs learned from training sequences. (**b**, **c**) Frequencies how often a CNN without and with circular filters learned shifted and truncated motifs. The true motifs had length 6 and were embedded in sequences of length 40 nt. A shift of 0 indicates that the full motif was learned correctly. (**d**) Relationship between sequence length and accuracy of distinguishing sequences with and without particular 6-mer motifs. Four architectures were investigated: a CNN with circular filters (green line), a CNN with one filter (blue line), a CNN with six filters (blue dashed line), and a neural network with a fully connected layer (dotted gray line). Plots show median accuracies over 100 different, randomly selected motifs. (**e**) The ratio of times a motif is inferred correctly depends on its composition. Diverse motifs are more likely to be inferred correctly on the first try for a CNN with circular filters (green boxes), whereas no difference is detectable for a CNN without circular filters (blue boxes). Diversity is measured in terms of the frequency of the most frequent motif letter. For example, in the motif ‘AACCGT’, the frequency of the most frequent letter is 2, whereas in ‘AAAAAC’, it is 5. Differences were tested using Mann-Whitney U tests. (**f**) When ten attempts are made at inferring a motif, both CNN (blue boxes) and CNN with circular filters (green boxes) can infer diverse motifs more easily. (**g**) Performance comparison between the CNN with circular filters (green), CNN without circular filters (blue) and DeepBind (red) on ENCODE ChIP-seq datasets for several filter configurations. For example, a filter configuration of 1 × 4 indicates that one filter of length 4 was used. Original training datasets contained 500 positive and 10^4 negative examples. Differences were tested using a binomial test, and p-values were corrected for multiple hypothesis testing with the Bonferroni method. We suspect that DeepBind might have been optimized for the use of multiple, long filters. (**h**) Examples of filter weights learned from ENCODE ChIP-seq data with a CNN with one circular filter of length 20, represented as heat-maps. The shown filter weights can be interpreted as the TATA-box (recognized by TATA-binding protein, TBP, left) and the UA1 motif (recognized by Breast cancer type 1 susceptibility protein, BRCA1, right). Letters on the x-axis indicate the most likely nucleotide at each position

the CNN with circular filters (Fig. 1), we also investigated two alternative architectures with circular filters. These networks used a simple sum or max-out of the activations (Goodfellow *et al.*, 2013). We generated synthetic datasets for both training and testing based on 32 random 6-mer motifs (a list of the motifs is given in Supplementary Table S4). For the training data, three datasets corresponding to the number of positive training sequences (5, 50 and 500) were generated for each motif, and each training dataset contained 10^4 negative examples. The test datasets consisted of 1000 positive and 1000 negative examples. Networks were trained using SGLD for 40 000 training steps, with mini-batches containing 20 training examples. AUROC values were calculated based on predictions on the test datasets. We then calculated the ratio with which the CNNs with circular filters achieved higher AUROC values than the CNN without circular filters for a particular hyperparameter. For example, to determine whether circular filters at an L_2 regularization strength of 0.1 overall improved performance, we kept the L_2 regularization strength fixed at 0.1 and then counted how often the CNNs with circular filters led to larger AUROC values than the CNN without circular filters for all remaining hyperparameter combinations of SGLD noise variance scaling factor, number of positive training examples, filter length and all 32 motifs. To test the obtained ratio, r, for statistical significance ($H_0 : r \leq 0.5$), we bootstrapped over the 32 motifs (10^5 bootstrap samples) and used a significance level of $\alpha = 10^{-4}$.

How does the CNN with circular filters perform compared to the state of the art on ChIP-seq data? We compared how well the CNN with circular filters can infer motifs from ENCODE ChIP-seq datasets compared to a CNN without circular filters and the state of the art algorithm, DeepBind (Fig. 3g). Since our computational resources were limited, we restricted our analysis to 48 randomly selected out the 506 ENCODE ChIP-seq datasets that were used by the DeepBind authors (a list of the investigated datasets is given in Supplementary Table S5). We investigated the performances for several filter configurations, using either 1 or 16 filters with filter lengths of 4, 8 or 24 nt. All three algorithms were provided with exactly the same training and test data. Training datasets contained 500 positive and 10^4 negative examples and were then split into training and development datasets in a way such that training datasets contained 400 positive and 9900 negative examples, and development datasets contained 100 positive and negative examples, respectively. The CNNs with and without circular filters were trained for 5000 steps at a learning rate decreasing from 0.1 to 0.01, with batch size 100. These networks were trained 10 times and the best performing model was identified based on the development datasets (we found this to be fair since DeepBind uses 30 trials in its calibration phase).

Model performances were then compared based on AUROC values on test datasets. All ChIP-seq experiments for which one architecture yielded AUROC values equal to 0 were removed from further analyses. We found this to be necessary because for some runs, DeepBind returned AUROC values of 0, a behaviour that has been reported before (Zeng *et al.*, 2016).

For each filter configuration, we then tested if the number of times k with which the CNN with circular filters had produced AUROC values larger than the corresponding ones produced by the CNN and DeepBind was significant. The null hypothesis was modeled using a binomial distribution $B_{n,p}(k)$ with n the number experiments and $p = 0.5$. Resulting p-values were adjusted for multiple hypothesis testing with the Bonferroni method, since a total of 12 hypothesis tests were conducted.

Motifs inferred from ENCODE ChIP-seq data. We used the CNN with circular filters to infer motifs for all 506 ENCODE ChIP-seq datasets published by the DeepBind authors. A filter length of 20 was used for all datasets; models were trained for 5000 steps at learning rate 0.01 with balanced mini-batches of size 100. The filter weights shown in Figure 3h were learned based on datasets BRCA1_H1-hESC_BRCA1_(A300-000A)_Stanford and TBP_HeLa-S3_TBP_Stanford by the DeepBind authors, which are derived from ENCODE datasets wgEncodeAwgTfbsSydhH1hescBrca1IggrabUniPk and wgEncodeAwgTfbsSydhHelas3TbpIggrabUniPk, respectively (Alipanahi *et al.*, 2015; Dunham *et al.*, 2012).

3 Results

3.1 CNNs frequently learn truncated motifs

While studying motif inference with CNNs, we observed that the learned filters frequently did not correspond to the complete motif. Instead, the filters often contained truncated and shifted versions of the motif (Fig. 3a). To quantify this behaviour, we conducted simulations in which a known motif had to be inferred from a set of short sequences. Then, it was counted how frequently the trained filters contained a shifted version of the motif. We found that CNNs learned shifted and truncated motifs more frequently than the true motifs (Fig. 3b).

3.2 Circular filters improve robustness of sequence motif inference for simulated data

This observation motivated us to develop a novel convolutional architecture that already contained all circularly permutated variants of the same underlying filter, which we refer to as circular filters (Fig. 1). If one of the filter variants learns a shifted motif, there is another filter variant that is able to learn the full motif, provided the filter variants have at least the size of the motif. We found that CNNs with circular filters rarely learn shifted motifs (Fig. 3c). Specifically, when trained on 100 positive examples to infer motifs of 6 nt length from sequences of 40 nucleotide long sequence, the CNN with circular filters learned the correct motifs 11.2 times more often than a CNN without circular filters (Supplementary Table S1). Even when the CNN with circular filters was trained with only five positive examples, the correct motif was found 4.5 times more often.

Moreover, with increasing sequence length, CNNs with circular filters inferred motifs more easily than CNNs without circular filters and fully connected networks (Fig. 3d). In fact, a CNN with circular filters of length 6 nt inferred 6-mer motifs about as well a CNN with 6 filters of length 6 each when 100 positive training examples were provided, even though it only required approximately $\frac{1}{6}$th of the number of parameters. We quantified the effect of circular filters on motif recognition further by comparing network architectures with and without circular filters for a variety of hyperparameter combinations. These included the number of positive training examples, L_2-regularization strength and the amount of noise injected into parameter updates via SGLD (Welling and Teh, 2011). To investigate the effect of the weighted sum of activations that appears in the CNN with circular filters (Fig. 1), we also investigated two alternative architectures with circular filters, which used a simple sum or max-out of the activations (Goodfellow *et al.*, 2013). Overall, both the CNN with circular filters and the CNN with circular filters and sum of activations performed significantly better than the CNN without circular filters in 74% and 62% of all cases, respectively ($p < 10^{-4}$). The CNN with circular filters and max-out however performed significantly worse than the CNN without circular filters in 56% of cases ($p < 10^{-4}$). A detailed comparison is given in Supplementary Table S3.

We furthermore investigated the effect of motif composition on motif inference. For example, the motif 'AACCGT' is more diverse in its composition than 'AAAAAC'. We found that the CNN with circular filters inferred motifs more easily the more diverse they were (Fig. 3e). When models were trained ten times on the same dataset and the best performing model was used, this effect also appeared for the CNN without circular filters (Fig. 3f).

3.3 Circular filters improve motif inference from ChIP-seq data

To compare the performance of the CNN with circular filters to the state of the art algorithm for motif inference, DeepBind, as well as a CNN without circular filters, we compared these methods on ChIP-seq datasets for a variety of filter configurations (Fig. 3g) (Alipanahi *et al.*, 2015; Dunham *et al.*, 2012). The filter configurations included different filter numbers and lengths. We found that the CNN with circular filters performed better than the CNN without circular filters for most filter configurations. The CNN also performed better than DeepBind for most filter configurations. We suspect that DeepBind might have been optimized for use of multiple, long filters. When 16 filters of length 24 were used, no significant difference between the models could be detected.

A CNN with one circular filter correctly inferred sequence motifs such as the TATA-box, which is recognized by TATA-binding protein (TBP), and the UA1 motif, which is recognized by BRCA1 (Fig. 3h) (Wang *et al.*, 2012). Images containing inferred motifs for all 506 ENCODE ChIP-seq datasets can be obtained from: http://www.mathmodeling.hhu.de/datasets/.

4 Discussion

4.1 Motif inference differs from deep learning disciplines

We showed in a simple simulation that a shallow CNN architecture frequently learns non-optimal filters that correspond to shifted and truncated versions of the underlying inferable pattern. Furthermore, when networks were trained multiple times and the best performing models were used, substantially more motifs were inferred correctly. This indicates that the non-optimal filters corresponded to local optima. This is in agreement with the well-studied behavior of gradient descent optimization, which is highly sensitive to initial conditions and prone to local optima. The local optima observed here seem to be more difficult to escape compared to other machine learning disciplines where CNNs are applied.

4.2 Circular filters enable robust motif inference

The filters of conventional CNNs are developed during training by starting from randomly initialized weights ('seeds'). For motif inference, this seed influences at which position in the filter the inferred motif will appear. This means that it can happen that only one side of a motif can be learned because an edge of the filter has been reached. A common work-around is to use multiple and longer filters that give the seeds more space to develop; however, this comes at the cost of more parameters, and learned motifs are distributed across filters. With circular filters, the position from where a motif is developed does not matter anymore because in one of the filter variants, it will be at the correct position (assuming a filter that is at least as long as the motif). Thus, circular filters greatly improve the chance to capture motifs with a single filter.

It has been observed before that slight modifications in neural network structures alongside negligibly more parameters can substantially improve inference performance (He *et al.*, 2015; Ioffe and Szegedy, 2015). These findings have demonstrated the relevance of hard-wired prior knowledge about the underlying problem and optimization techniques. Circular filters can be regarded as hard-wired prior knowledge as well: sequence motifs are locally correlated data features that require convolution to be learned, and optimization with gradient descent develops the motif from initial weights. It hence is unsurprising that a neural network with a fully connected layer, for which the first condition is not fulfilled, performed poorly at learning sequence motifs (Fig. 3d). Moreover, a CNN required six filters to become as good as a CNN with a single circular filter (Fig. 3d).

Although the three investigated architectures with circular filters differed only in the last layer, they showed significant performance differences (Supplementary Table S3). For the CNN with circular filters and max-out, backpropagation of the classification error can only occur to the circular filter variant that led to the maximum activation. After applying the parameter updates, it can happen that another circular filter variant leads to the maximum activation in the next training step. This may complicate parameter optimization, explaining the lower performance compared to the architecture without circular filters. For the other architectures, the classification error can backpropagate to all filter variants, allowing the motif to be learned in any of the circular filters. However, also filter variants that do not contain the inferable pattern can contribute to the classification error, which may be harmful if some training sequences also randomly contain other patterns. For the CNN with circular filters, it can be adjusted by a linear combination how much the filter variants contribute to the classification, which is not the case for the CNN with circular filters and sum of activations.

4.3 Deep models may not be necessary for modeling TF-DNA specificities

The good performance of (Alipanahi *et al.*, 2015) suggests that deep neural networks are necessary to accurately predict molecular binding interactions. However, although neural networks are complex function approximators and a deep CNN with 152 layers achieved state of the art performance at classifying images, it was demonstrated that deeper architectures actually perform worse at learning sequence motifs than simpler architectures (He *et al.*, 2015; Hornik, 1991; Zeng *et al.*, 2016). A likely reason is that biological sequences are not composed of complex hierarchies of patterns such as those in images. In fact, mutually exclusive sequence patterns or spatial relationships can already be modeled with two layers, and to the best of our knowledge, no protein has yet been found which binds to mutually exclusive motifs. This likely makes truly deep architectures unnecessary and possibly deleterious because more parameters need to be trained. Also, most transcription factors bind to motifs of 30 nt length, a size that can be captured with simple convolutional filters (Stewart *et al.*, 2012). Because of the aforementioned reasons, it is unsurprising that the discriminative implementation of DeepBind has only one convolution layer but many filters to overcome local optima.

4.4 Summary

When applied to biological data, a CNN with circular filters performed at least as good as the current state of the art algorithm for several combinations of filter number and length. In simulations, CNNs with circular filters performed as good as or better than the corresponding CNNs without circular filters. Even for small dataset sizes, CNNs with circular filters were able to infer motifs more easily than CNNs without circular filters trained with 20 times more training examples. Although motif composition affects motif inference, this does not seem to be a side-effect of circular filters as it also occurred for CNNs without circular filters. Overall, our findings show that circular filters enable more efficient use of data for sequence motif inference.

Acknowledgements

Computational services were provided by the High Performance Computing Platform of the Heinrich-Heine University in Düsseldorf, Germany.

Funding

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2048/1 – Projektnummer 390686111. C.F.B. was supported for this research through the International Max Planck Research School (IMPRS) on Understanding Complex Plant Traits using Computational and Evolutionary Approaches at the Max Planck Institute for Plant Breeding Research and the Universities of Düsseldorf and Cologne.

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.

Altschul,S.F. and Erickson,B.W. (1985) Significance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleotide and codon usage. *Mol. Biol. Evol.*, **2**, 526–538.

Berger,M.F. and Bulyk,M.L. (2009) Universal protein-binding microarrays for the comprehensive characterization of the DNA-binding specificities of transcription factors. *Nat. Protoc.*, **4**, 393–411.

Dunham,I. *et al.* (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**, 57–74.

Goodfellow,I. *et al.* (2013) Maxout networks. In: Dasgupta,S. and McAllester,D. (eds) *Proceedings of the 30th International Conference on Machine Learning*, Volume 28 of Proceedings of Machine Learning Research, pp. 1319–1327, Atlanta, Georgia, USA.

He,K. *et al.* (2015) Deep residual learning for image recognition. *CoRR*, arXiv preprint arXiv:1512.03385.

Hornik,K. (1991) Approximation capabilities of multilayer feedforward networks. *Neural Networks*, **4**, 251–257.

Ioffe,S. and Szegedy,C. (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Bach,F. and Blei,D. (eds) *Proceedings of the 32nd International Conference on Machine Learning*, Volume 37 of Proceedings of Machine Learning Research, pp. 448–456, Lille, France.

Kidder,B.L. *et al.* (2011) ChIP-Seq: technical considerations for obtaining high-quality data. *Nat. Immunol.*, **12**, 918–922.

Man,T.K. and Stormo,G.D. (2001) Non-independence of Mnt repressor-operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nucleic Acids Res.*, **29**, 2471–2478.

Merkel,D. (2014) Docker: lightweight linux containers for consistent development and deployment. *Linux J.*, **2014**, 2.

Nickolls,J. *et al.* (2008) Scalable parallel programming with CUDA. *Queue*, **6**, 40–53.

Rohs,R. *et al.* (2010) Origins of specificity in protein-DNA recognition. *Annu. Rev. Biochem.*, **79**, 233–269.

Simard,P.Y. *et al.* (2003) Best practices for convolutional neural networks applied to visual document analysis. In: *Proceedings of the Seventh International Conference on Document Analysis and Recognition*. pp. 958–963. IEEE, Edinburgh, Scotland.

Stewart,A.J. *et al.* (2012) Why transcription factor binding sites are ten nucleotides long. *Genetics*, **192**, 973–985.

Wang,J. *et al.* (2012) Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome Res.*, **22**, 1798–1812.

Weirauch,M.T. *et al.* (2013) Evaluation of methods for modeling transcription factor sequence specificity. *Nat. Biotechnol.*, **31**, 126–134.

Welling,M. and Teh,Y.W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11*, pp. 681–688, Omnipress, USA.

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