Molecular Omics

ROYAL SOCIETY OF CHEMISTRY

RESEARCH ARTICLE

View Article Online
View Journal

Check for updates

Cite this: DOI: 10.1039/d0mo00031k

Enhancer recognition and prediction during spermatogenesis based on deep convolutional neural networks

Chengzhang Sun,†[a] Ning Zhang, †[a] Peng Yu, [a] Xiaolong Wu,[b] Qun Li, [a] Tongtong Li,[a] Hao Li,[a] Xia Xiao,[a] Abdullah Shalmani,[a] Leijie Li,[a] Dongxue Che,[a] Xiaodan Wang,[a] Peng Zhang,[a] Ziyu Chen,[a] Tong Liu,[c] Jianbang Zhao,[d] Jinlian Hua*[b] and Mingzhi Liao *[a]

Received 13th March 2020,
Accepted 28th May 2020

DOI: 10.1039/d0mo00031k

rsc.li/molomics

Motivation: enhancers play an important role in the regulation of gene expression during spermatogenesis. The development of ChIP-Chip and ChIP-Seq sequencing technology has enabled researchers to focus on the relationship between enhancers and DNA sequences and histone protein modifications. However, the prediction of enhancers based on the locally conserved DNA sequence and similar histone modification features is still unknown. Here, the present study proposed a convolutional neural network (CNN) model to predict enhancers that can regulate gene expression during spermatogenesis. Results: we have obtained a positive set of enhancers using the P300 locus, verified by experiments, while a negative set was constructed using the promoter as a non-enhancer locus. The model was trained on all types of specific cells during spermatogenesis independently, and the transfer learning strategy was used to fine-tune the model based on which the model can be trained and adapted to other cells quickly. We visualized the convolution layer of the trained model and aligned the predicted enhancer with the JASPAR database. The results showed that the model was highly matched with some important transcription factors during spermatogenesis, signifying the reliability of the model. Finally, we compared the CNN algorithm with the gkmSVM algorithm (Support Vector Machine). It is well known that CNN has better performance than the gkmSVM algorithm, especially in the generalization ability. Our work demonstrated their strong learning ability and the low CPU requirements for the experiment, with a small number of convolution layers and simple network structure, while avoiding overfitting the training data. At the end of the experiment, we used the trained model to build an enhancer recognition website for further research and communication.

Introduction

The enhancer is a DNA *cis*-acting element that enhances the transcriptional activity of promoters.[1] The enhancers consist of approximately 50–1500 bases and are considered one of the most critical regulatory elements in gene expression. Spermatogenesis is a complex process involving multiple stages and thousands of genes that are difficult to detect, particularly in mammals. Many studies have shown that testicular germ cell-specific genes are controlled by the proximal promoter, insulators, and distal enhancers.[2–4] The identification of enhancers may play a vital role in the investigation of the complex processes of spermatogenesis. However, the large scale identification of active enhancers across a variety of human tissues and cell lines is a difficult task due to expensive experimental approaches and time-consuming processes. So far, despite great efforts, the ENCODE and Roadmap projects were only able to carry out histone modification experiments in several hundred human cell lines, still far less than forming a comprehensive landscape of enhancers under different disease states, and subsequently preventing the deciphering of gene regulatory mechanisms.

Previous studies have reported some machine learning methods that use gene sequence and histone modification signatures to identify enhancer regions. Heintzman *et al.* analyzed 30 Mb sequences of the histone modification regions in human

[a] *College of Life Sciences, Northwest A&F University, Yangling, Shaanxi 712100, China. E-mail: liaomingzhi83@163.com*

[b] *College of Veterinary Medicine, Shaanxi Centre of Stem Cells Engineering & Technology, Northwest A&F University, Yangling, Shaanxi, 712100, China. E-mail: jlhua2003@126.com*

[c] *Chinese Academy of Sciences Center for Excellence in Molecular Plant Sciences, Shanghai Center for Plant Stress Biology, Chinese Academy of Sciences, Shanghai, 200031, China*

[d] *College of Information Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China*

† These authors contributed equally to this paper.

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

This journal is © The Royal Society of Chemistry 2020

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

HeLa cells and found that the active promoter was marked by histone H3 (H3K4) trimethylation, while the enhancer was marked by H3K4 monomethylation rather than trimethylation. Some studies used histone modification to predict *cis*-acting factors.[5] Won *et al.* used Hidden Markov Models (HMMs) to predict enhancers using three primary histone modifications.[6] The RFECS (Random Forest-based Enhancer identification from the Chromatin States) and SVM are also used to identify enhancers.[7,8] RFECS has improved a limited number of training samples in previous methods using random forests to determine the best combination of histone modifications to predict enhancers. Chen *et al.* used a deep CNN to predict enhancers *via* the integration of DNA sequences and DNase-seq data and discriminate disease-related enhancers.[9] Dikla Cohnet *et al.* trained deep CNN to identify enhancer sequences in multiple species. Chen *et al.* used multiple biological datasets including simulated sequences, *in vivo* binding data of single transcription factors, and genome-wide chromatin maps of active enhancers in 17 mammalian species.[10] Lee *et al.* developed a computational framework called kmer-SVM based on the as-known SVM to predict mammalian enhancers from background sequences.[11] They found that some predictive k-mer features are enriched in enhancers and have potential biological meaning. Ghandi *et al.* improved kmer-SVM by adopting another type of sequence feature called gapped k-mers.[12] Their method, known as gkmSVM, showed robustness in the estimation of k-mer frequencies and higher performance than kmer-SVM. However, k-mer features, though unbiased, may lack the ability to capture high order characteristics of enhancer sequences. So far, various studies have been performed to apply the most advanced deep learning methods in bioinformatics problems. For instance, a recent study proposed the use of Deep Neural Network (EP-DNN) to predict enhancers using sequences and four histone modifications.[13] It has been reported that PEDLA trained with 1,114-dimensional heterogeneous features showed excellent performance in H1 cells.[14]

According to the previous studies, machine learning could successfully be applied for the recognition of enhancers in the whole genome. Because enhancers are highly cellular or tissue-specific, and many enhancers only work in specific tissues and cells. Most of the current studies on enhancers are based on human H1 or CD4$^+$ T cells. Previous studies reported that the mechanism of gene regulation in spermatogenesis is so complex and different from that in somatic cells. So far, a few studies have been performed on the role of enhancers and other regulatory factors in the process of spermatogenesis.

Recent studies have evaluated the characteristics and properties of enhancers in cells; however, less studies about the function of enhancers have been processed in the complex biological process of spermatogenesis. Therefore, in the present study, we identified and analyzed the enhancers in specific cell types during spermatogenesis using the CNN model. It has been found that deep learning methods such as deep CNN and generative adversarial networks have achieved great success in various computer vision tasks.[15–18] We believe that image recognition is similar to enhancer recognition. We assumed that some of the same features of different pictures might be located at different positions in the picture. Inspired from the rotation of the picture and pixel deletion, we postulate that enhancers can also have these characteristics. Enhancers have different positions in sequences, and sequences can be rotated without changing the characteristics of enhancers. Based on the above findings, we claimed that the convolution neural network could be used to identify enhancers in specific cells during spermatogenesis.[19,20] The CNN method that is used in the present study requires less training data and time, and showed higher training accuracy compared with the traditional machine learning method. Moreover, CNN has a stronger generalization ability for different types of cells by using a transfer learning strategy in training. We visualized the convoluted nucleus layer and compared it with the JASPAR biological database to further explore its biological significance, which can provide a reference for the regulation process of spermatogenesis-related enhancers.[21]

Methods

Data collection and processing

We collected all throughput sequence datasets during mouse spermatogenesis, which can be divided into three parts: P300 datasets, histone modification datasets, and DNA sequences (known promoter, enhancer sequences, and random sequences). The first part of the Anti-P300/CBP antibody ChIP-seq was collected from Gene Expression Omnibus (GEO), NCBI with series GSE97703 and the P300 antibody binding site distribution is shown in Fig. 1.[22] The dataset for histone modifications was collected from GEO, NCBI with series GSE49624.[23] The datasets for enhancers were collected from VISTA Enhancer Browser,[24] which is considered as the central resource for experimentally verified mouse non-coding fragments with gene-enhancing activity. The promoter sequence was collected from the mouse promoter database (Eukaryotic Promoter Database), and random sequences were generated by generating random numbers from the mouse genome. The GSE49624 datasets consist of four types of histone modification for three types of cells during the spermatogenesis, including H3K4me3, H27me3, H3K4me1, and H3K27ac. The datasets were then divided into positive and negative sets: P300 and experimentally verified enhancers were used as positive sets, whereas the promoter and random sequences from the mouse genome were used as the negative sets. It has been investigated that P300/CBP is a transcriptional coactivator that works by binding to transcription factor activation domains and histone acetyltransferase (HAT) positions that show a specific nucleosome pattern. Recent studies have shown localized P300 in many active target gene promoter regions and enhancers.[25,26]

Histones are a type of necessary protein among which lysine and arginine show high abundance. Histone modifications and their dynamic changes play an important role in chromatin modification and gametophyte maturation in normal meiosis.[27] In the process of spermatogenesis, the methylation of H3K4, H3K9 or H3K27 results in temporary expression under strict regulation to ensure the correct operation of spermatogenesis. The methylation and acetylation at specific positions affect the

This journal is © The Royal Society of Chemistry 2020

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

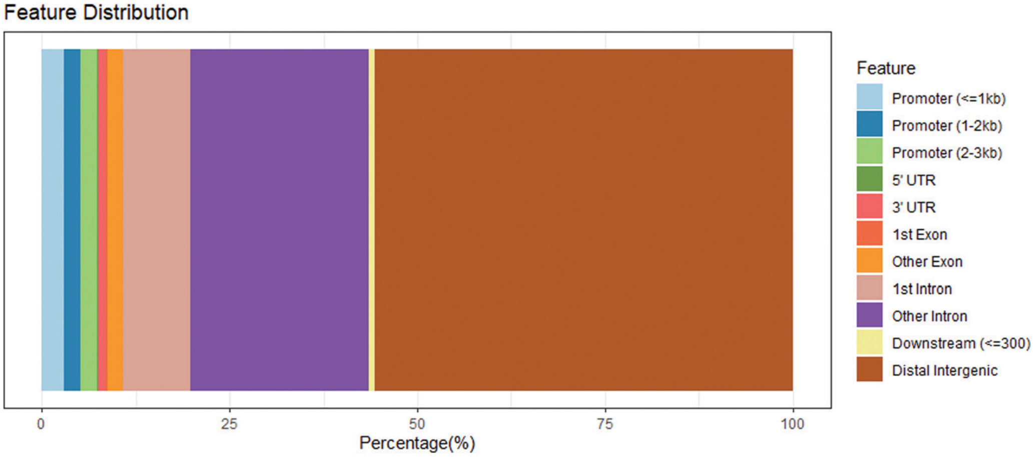


Fig. 1 The distribution of the P300 binding site in the genome. The P300 binding site mainly located in the distal intergenic region.

Table 1 The descriptions of datasets used for training and testing, including data types, GEO sample IDs, sequencing techniques and cell types. SC, AGSC and ST represent spermatocytes, spermatogonia and spermatids respectively

Title	Sample ID	Technique	Cell	Ref.
P300	GSM2575692	Illumina HiSeq 2000		22
H3K4me3	GSM1202705	Illumina HiSeq 2000	AGSC	23
H3K4me3	GSM1202706	Illumina HiSeq 2000	SC	23
H3K4me3	GSM1202707	Illumina HiSeq 2000	ST	23
H27me3	GSM1202708	Illumina HiSeq 2000	AGSC	23
H27me3	GSM1202709	Illumina HiSeq 2000	SC	23
H27me3	GSM1202710	Illumina HiSeq 2000	ST	23
H3K4me1	GSM1202711	Illumina HiSeq 2000	SC	23
H3K4me1	GSM1202712	Illumina HiSeq 2000	ST	23
H3K27ac	GSM1202713	Illumina HiSeq 2000	AGSC	23
H3K27ac	GSM1202714	Illumina HiSeq 2000	SC	23
H3K27ac	GSM1202715	Illumina HiSeq 2000	ST	23

expression of *cis*-acting elements in the sequence such as promoters and enhancers. Therefore, some studies revealed that these histone features could be vital for predicting the potential enhancers in the genome. For example, it was found that enhancers are marked by H3K4 monomethylation but not trimethylation.[5] The genome-wide ChIP-seq histone information used in this study is shown in Table 1.

The P300 and histone ChIP-seq data were processed. Initially, the ChIP-seq reads were filtered for quality and 3′ trimmed for adapter sequences using FastQC.[28] Reads were aligned to mouse assembly mm10 using bowtie2[29] with default parameters. We used MACS[30] to identify peaks from ChIP-Seq data with default parameters. Ultimately, we picked a 1 kb window around each P300 bound locus and histone marks to construct the training data. Bases of 1000 bp length upstream and downstream of the peak point serve as an enhancer sequence. Most of the current optimized sequence length selections are 1 kb because it is sufficient to contain the information of the enhancer (50–200 bp) and incurs minor errors.[7] So for each sample site, we used a sequence of 499 bp upstream and 500 bp downstream and considered the four histone modifications on this 1 kb sequence.

The principle and arithmetic of convolutional neural networks

The CNN is composed of many parts, such as the input layer, convolution layer, pooling layer, inception module, and full connection layer. The input layer refers to the layer in which the CNN receives the data. The function of the convolution layer is to extract the features of each layer, which has a convolution nucleus, similar to neurons. After feature extraction in the convolution layer, the output feature map was transferred to the pooling layer for feature selection and information filtering. Batch normalization (Batchnorm) can be used to reduce the amount by what the hidden unit values shift around (covariance shift). The batch normalization of networks can increase the training rates and reduces overfitting. For a mini-batch (Mini-batch gradient descent is a trade-off between stochastic gradient descent and batch gradient descent) $B = \{x_{1},\ldots,x_{m}\}$, we need to learn parameters γ and β, and obtain output y_i.

$$\hat{x} = \frac{x_i - \mu_\mathrm{B}}{\sqrt{{\sigma_\mathrm{B}}^2 + \varepsilon}} \quad (1)$$

$$y_i \leftarrow \gamma\hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \quad (2)$$

where μ_B is mini-batch mean and σ_B is variance. In the equation $y = \mathrm{BN}_{\gamma,\beta}(x)$, the parameters γ and β can be learned from the dataset. The BN (Batch Normalizing Transform) is the transformation process in the algorithm. Activation functions are usually used to ensure the nonlinearity of each layer of the whole model, such as the Rectified Linear Unit function (ReLU).[31] The ReLU ensures that the feature mapping is always positive. After ReLU, the model can extract relevant features and fit training data better.

$$\mathrm{ReLU} = \max(0, x) \quad (3)$$

Because the convolution layer acts on one window each time, the model is sensitive to the positions of the P300 binding sites. The sensitivity can be further improved after the pooling layer significantly. The Max-pooling function takes the maximum value of all neurons in the regions. We then used an across-

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

entropy loss function as the objective function for a classification network, which can be described as follows.

$$H(p,q) = -\sum_{x} p(x)\log q(x) \tag{4}$$

where $H(p,q)$ is the cross-entropy function, p is the target distribution and q is the approximation of the target distribution.

The Softmax function is mostly used in the final layer of a neural network-based classifier that gives the estimated class probabilities.[32] We used MXNet, which is a flexible and efficient software framework for deep learning, to build a neural network.[33] It is scalable and can be efficient for fast model training. In this study, the python program was used to build neural networks. The Gluon interface provided by MXNet can be used to write programs conveniently and effectively. The model was developed by using the sequential class of gluon interfaces to achieve the functions of Auto Grad, Conv2D, and MaxPool 2D.

$$y_i = \frac{e^{Z_i}}{\sum_j e^{Z_j}} \tag{5}$$

where Z_i is the input number of last layers in a neuron, and it's divided by the sum of all neuron results in the last layer to ensure that the final result is within 0–1, which can also be used as a probability of final prediction. $j \in \{0, 1\}$ is the label for the i-th category and y_i is the prediction confidence for the i-th category.

Transfer learning

Considering the high cost of obtaining training data, it is very important to reduce the need for effective training data. In this case, the method of transfer learning is more convenient and helpful. Due to the similar cell types between sperm cells and spermatocytes, transfer learning may improve the model efficiency, which does not need to initialize model parameters once again. So we trained the model based on the dataset of one type of cells to other types of cells with parameters fine-tuning automatically. This kind of transfer learning can reduce the training time of the model and can also get acceptable results.

Visualization of the convolution kernel

Considering the difficulty of understanding the features used in dense layers of a CNN, we sought all the possible input matrices that have positive activation values through the first convolutional layer, and then aggregated them into a positive weight matrix (PWM) which is used to represent a motif. We found all possible one-hot encoded input matrices in a shape with positive convolutional activations, which were represented as visual motifs. We took out the parameters of the first layer convolution kernel of a convolution neural network, with each convolution kernel in the shape of 8 × 1 × 8. Then, we extracted the parameters for which the shape is 4 × 1 × 8 for identifying the DNA sequence, retaining the positive value, and setting the negative value to 0.

Performance evaluation

Performance of binary classification was evaluated based on standard measures, including sensitivity (also known as recall), specificity and precision,

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

where TP is the number of residues correctly labelled as positives (true positives); TN is the number of residues correctly labelled as negatives (true negatives); FP is the number of misclassified negative cases (false positives), and FN is the number of misclassified positive cases (false negatives).

To evaluate the prediction performance, ROC analysis was carried out in the following analysis, which was generated from the pairs (1-specificity, sensitivity). The area under the curve (AUC) was calculated with the pROC package in the R program.

Comparison with the gkmSVM algorithm

The datasets were run on the same computer (Intel$^{®}$ Core(TM) i7-3770 CPU@3.40 GHz and 16 GB memory) with low CPU requirements. The personal computer was used for enhancer prediction, visualization of CNN, transfer learning, and the calculations of sensitivity (also known as recall), specificity and precision.

Data set construction and processing

In the present study, the construction of the data set was divided into two parts, namely the selection of feature attributes and division of data sets (positive and negative sets). To process the trained data samples into inputs that can be recognized by CNN, we need to specialize in the data. For each trained sequence, we scanned 1 window around each P300 binding locus, then we combined the 1 kb base sequence with the histone modification information to form a matrix of 8 × 1 × 1000 (8 is the channel, including A, T, C, G and four types of histones, whereas 1 × 1000 is the length of 1 kb sequence). The P300 locus regions were aligned with the histone modification regions (Fig. 2A). If a region is modified by histones, the characteristic value of the position is denoted by 1, whereas the 0 value means the region is not modified by histones. This process can also be called one-hot coding. We encoded four nucleic acid bases to form a matrix as the input to the CNN. The format of the input data is shown in Fig. 2.

The datasets were constructed for spermatids, spermatocytes and spermatogonia respectively. Each dataset includes a positive set, consisting of P300 and known enhancers, and a negative set, consisting of promoter sequences or random sequences from the whole mouse genome. Considering the number of non-enhancers

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

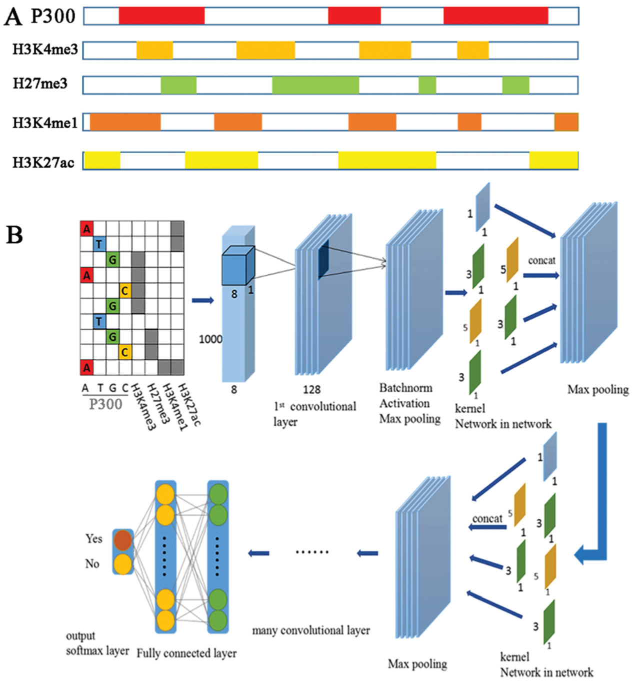


Fig. 2 Data preprocessing and CNN structure model. (A) The combination of the peak value of P300 and histone (H3K4me3, H27me3, H3K4me1, and H3K27ac) modification. The matching results are processed into the corresponding matrices, which are used as inputs of the neural network. Four colors indicate the four histone modification sites. (B) The structure of the convolution neural network model. First, the nucleotide sequences A, T, G, C, and four histone modifications were encoded by one-hot. Then it passes through a convolution kernel layer, followed by the Batch norm layer, activation function, and max-pooling layer. Then there are two networks in the network structures to extract local features. Then it passes from several convolution network layers, and finally connects to two fully connected layers, and the softmax function is used to output the final result.

is far more than that of known enhancers from the whole genome perspective, we set different ratios of positive and negative sets as 1 : 1, 1 : 4 and 1 : 9 respectively to test the stability of our models.

Build the convolutional neural networks

The deep CNN model is illustrated in Fig. 2B. Our CNN contains seven learned layers, two inception modules, three convolutional and two fully-connected. The input layer is an 8 × 1 × 1000 matrix, which is encoded by the one-hot method and represents the character of 1000 bp sequence. The convolutional layers contain different kernels with the same sliding step size 1. Behind each convolutional layer, there are the batch norm, activate and max-pooling layers. At the end of the neural network, there are two fully connected layers of size 512 and 256, respectively. Finally, the probability of the result was generated by the two Softmax layers.

There are two inception modules behind the first convolutional layer. The main idea of the inception module is to cover the optimal local sparse structure of the CNN by easily available dense components.[34] According to the characteristics of the input data, the filter sizes of the inception module are designed as the forms of 1 × 1, 1 × 3, and 1 × 5. After an inception network, max-pooling layers with stride 3 are used to reduce the resolution of the grid.

Results

Prediction of enhancers using convolutional neural networks

We selected the model in terms of accuracy using 5-fold cross-validation in both the training set and validation set. The training error rate curve is illustrated in Fig. 3A. According to the curve, we used the strategy of the early stop. Before the 37th epochs (dotted red line) the Adam algorithm is used and the learning rate is 0.0001.[31,35] After 37 epochs, the random gradient descent (SGD) algorithm is used, and the learning rate is 0.00005.[36] By changing the algorithm and learning rate of CNN, the convergence effect of the CNN is better, such as faster convergence.

The training and test set increase the high accuracy rate, which is more in line with the actual production situation.[37,38] In the spermatids datasets, the training and testing ratio was 1 : 9, whereas the average accuracy of the 5-fold cross-validation on the training set was 94.70%, and the average accuracy on the testing set was 94.10% (Fig. 3B). In the spermatocytes and spermatogonia datasets, we used the transfer learning strategy. Fine-tuning with existing parameters, the curve converges faster, and only needs eight epochs (Fig. 3C) (Table 2). Through fine-tuning the parameters, we got better models to fit other cells quickly. In the original sperm cell training set, the training convergence took 19.2

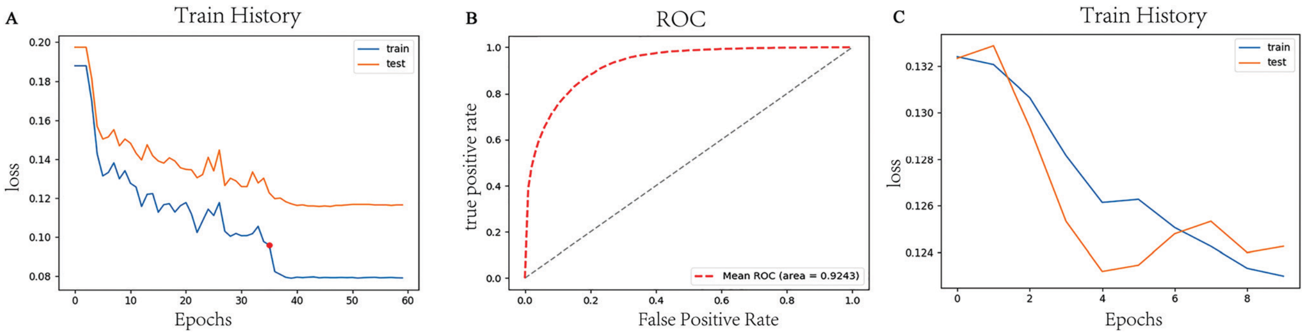


Fig. 3 (A) The relationship between training time and error rate. The vertical axis represents the change of the error rate on the training set and the test set, whereas the horizontal axis represents the number of training epochs. (B) The 5-fold cross-validation ROC curve of the CNN model on the test set. (C) The fine-tuning of the training accuracy curve of CNN in spermatogonia.

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

Table 2 The accuracy of each cell model on its own cell training set and test set

Cell	Positive : negative ratio	Train accuracy (%)	Test accuracy (%)
Spermatids	1 : 9	94.70	94.10
Spermatocyte		95.20	94.30
Spermatogonia		94.40	93.70
Spermatids	1 : 4	91.79	88.61
Spermatocyte		90.70	89.41
Spermatogonia		87.68	87.61
Spermatids	1 : 1	85.05	82.49
Spermatocyte		86.17	82.56
Spermatogonia		85.90	83.47

hours, but it took only 4.3 hours in the spermatocytes after using the transfer learning strategy, suggesting that the time performance has been greatly improved. In the spermatogonial dataset, the accuracy of CNN training was lower compared with the other two cells, probably due to the lack of H3K4me1 information, potentially signifying that histone modification information can be helpful for the recognition of enhancers.

Visualization of the convolution kernel

We built the parameters learned by the networks into a position weight matrix (PWM) and then visualized them. The PWM or position frequency matrix (PFM) methods have been applied to recognize the relationships between transcription factors and enhancers (Fig. 4).[39] Each convolutional layer of a CNN can be seen as a non-linear combination of many weighted position matrices. Therefore, the CNN may have better recognition of enhancers or transcription factors than PWM matrices. The visualization of motifs allows us to intuitively understand the sequence of enhancers and the CNN models, and it further facilitates our comparison with related enhancer or transcription factor bioinformatics databases.

The JASPAR core dataset contains a well-trained, non-redundant profile set derived from published and experimentally defined transcription factor binding sites in eukaryotes.[40] Tomtom is an algorithm to measure the motif-motif similarity, which can be used to search a database of motifs with a given query motif. The Tomtom sorts the topics in the database and produces an alignment for each significant match.[41] So we can use Tomtom to compare sequence patterns to a database of known patterns (*e.g.* JASPAR). Thus, these extracted motifs can be compared to the JASPAR database using Tomtom (Fig. 5A).

For each cell line, we used the Tomtom tool to compare the motif of the first convolutional transformation with the vertebrate motif database (JASPAR) and set the significance threshold *E* value to <1. By comparing with the JASPAR databases, many transcription factors related to the process of spermatogenesis can be found.[42,43] The results demonstrated that many of our

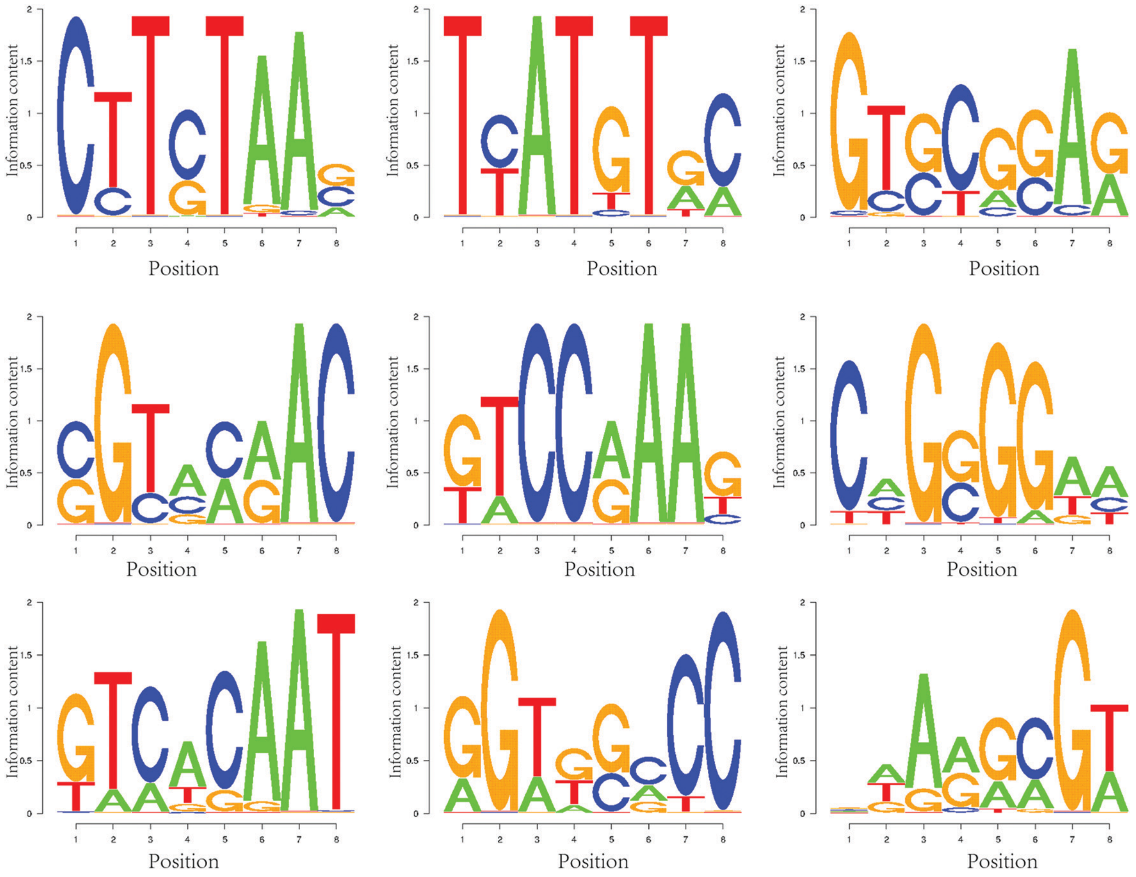


Fig. 4 The visualization of the first layers in the convolution kernel. The parameters from the convolution neural network are shown as the PWM logo.

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

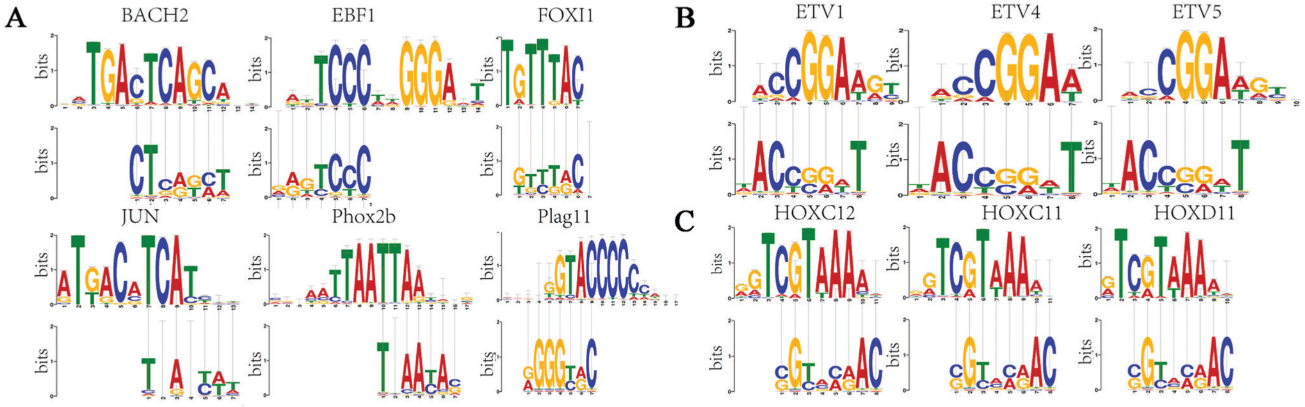


Fig. 5 Relationships between the first layer in the convolution kernel and transcription factors. The patterns obtained from our convolution neural networks are similar to some known transcription factors in JASPAR. (A) Motifs with high similarity to some known transcription factors. (B) Motifs with high similarity to the ETV family transcription factors. (C) Motifs with high similarity to the HOX family.

learned motifs have significant similarity to the biologically known motifs (Fig. 5). For example, one of the convolution kernel patterns has a higher similarity with the ETV family of transcription factors (Fig. 5B). The loss of ETV5 and receptor tyrosine kinase (RET) levels in neonatal mouse testicular germ cells could decrease the proliferation and cause abnormal spermatogenesis. Gaurav Tyagi *et al.* found that the loss of ETV5 may decrease the RET expression, which may inhibit the downstream GDNF/RET/GFRA1 signal.[44] The loss of ETV5 in SSCs may lead to the reduction of germ cell numbers during the early spermatogenesis.

Besides, one of the convolution kernel patterns has a higher similarity with the HOX family of transcription factors (Fig. 5C). The HOX gene encodes a class of transcriptional regulators with

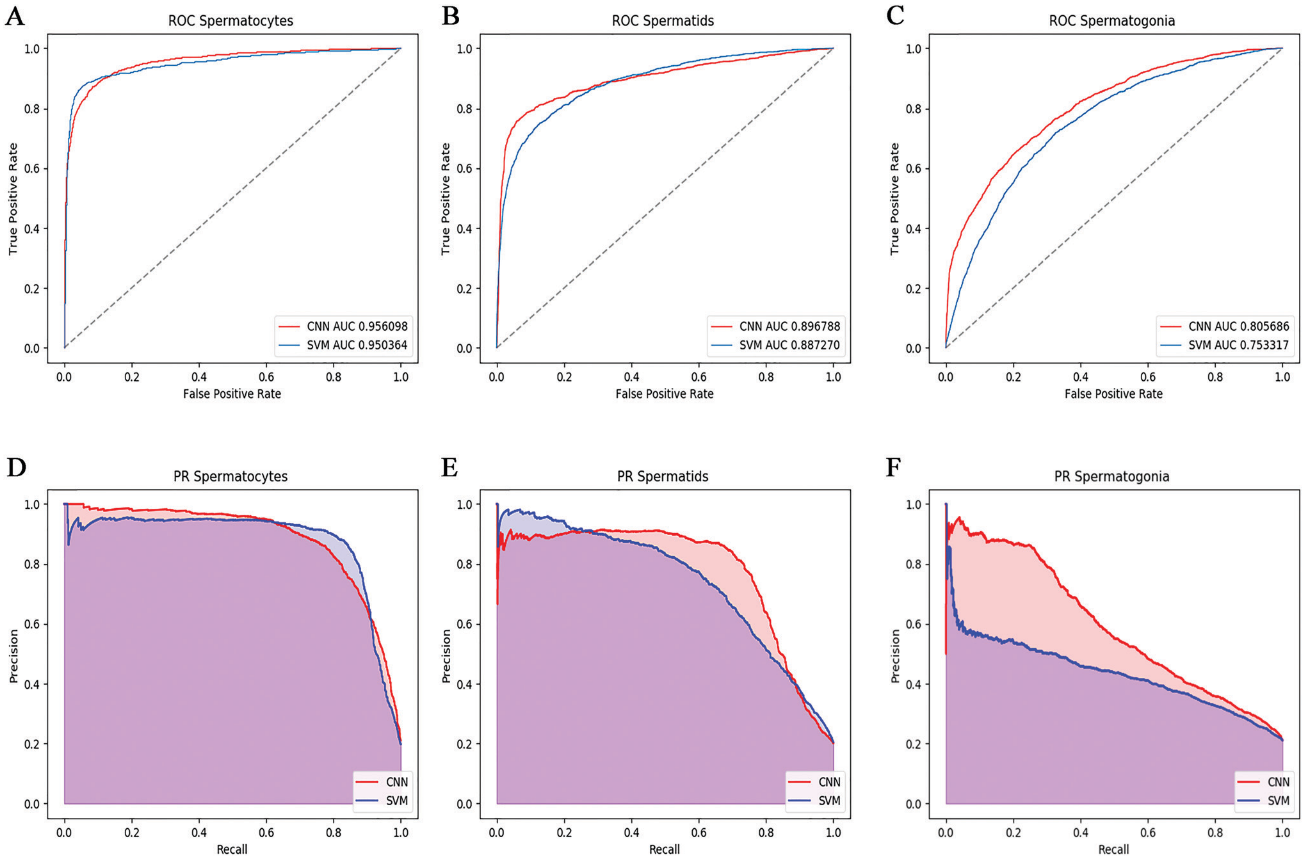


Fig. 6 Comparison of CNN and gkmSVM performance. (A) and (D) are the ROC and PR curves of the CNN and gkmSVM algorithms in spermatocytes. (B) and (E) are the ROC and PR curves of the CNN and gkmSVM algorithms in spermatozoa. (C) and (F) are the ROC and PR curves of the CNN and gkmSVM algorithms in spermatozoa. Red is the CNN, blue is gkmSVM.

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

homologous domains and similar DNA binding preferences. The HOX gene family is distributed in almost all eukaryotes. J. Chen's research reported that the HOXA1 and HOXB1 homeobox transcription factors directly regulate the expression of the EPHA2 gene in rhombomere 4.[45] Since the discovery of the HOX gene, it has attracted much attention because of its core role in regulating the diversity of the anterior and posterior axis phenotypes and specifying the characteristics of the axial skeleton in embryonic development. Furthermore, the convolution kernel patterns also have a high degree of matching with many transcription factors, such as SOX, NFY, and so many other transcription factors. These transcription factors play important roles in the regulation of the related gene expression.

Comparison with the gkmSVM algorithm

Finally, the predicted results of our algorithm were compared with the gkmSVM algorithm (Fig. 6). The gkmSVM is an SVM that predicts the functional genomic sequence elements using a short (6–8 bp) k-mer combination ranging from 500 to 2000 bp.[46] The gkmSVM algorithm only uses sequence information as the features so for ensuring fairness, our comparison with the CNN also just used these features. To explore the generalization ability of the two different algorithms in different cells, we used H3K27ac signals as enhancer bench datasets, which have been used as an important marker by many studies for enhancer identification.[47]

We used three different cell datasets to train their respective models and tested them on the corresponding cell test set. To evaluate the performance of the algorithm, we first draw the receiver operating characteristic curve (ROC) and calculate the area under the curve (AUC). Also, we calculate another criterion, the area under the precise recall curve (PR), which is less likely to cause inflation due to class imbalance compared with AUC.[48] The curves produced by our method climb much faster towards to top-left corner of the sub-plots, suggesting that our method can achieve a relatively high true positive rate at a relatively low false-positive rate (Fig. 6B and C). The precision–recall curves for individual cell lines also suggested the superiority of our method (Fig. 6E and F). Based on these results, we assumed that our deep learning model is more powerful than SVM-based methods.

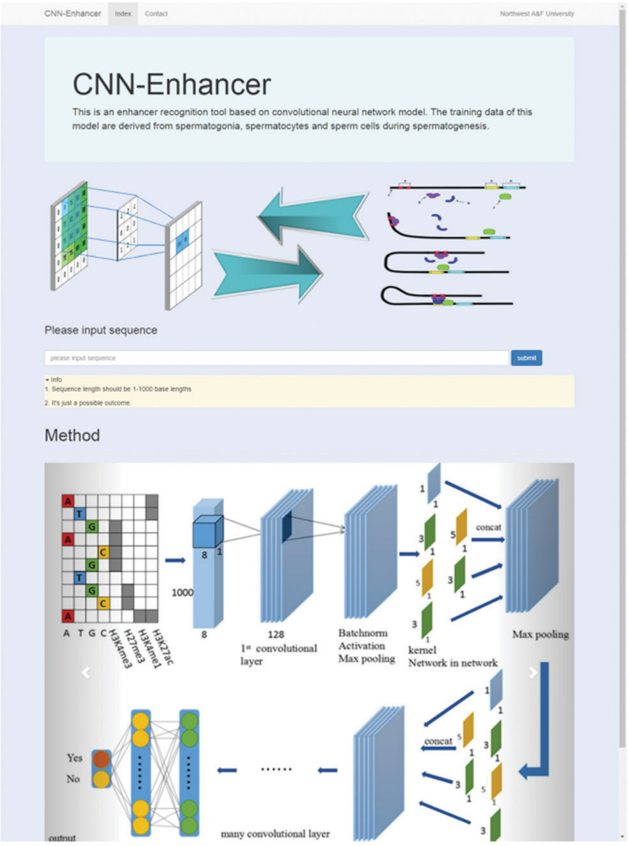
To assess the generalization capabilities of machine learning algorithms, we trained the model in a special cell to test the performance in other cells (Table 3). The results found that the accuracy of our model was 8% greater than the gkmSVM algorithm. Though our method showed obvious superiority on the whole, the accuracy of SVM is higher than that of CNN in spermatocytes. The reason needs to be explored in the future. Our study also observed that the prediction performance in the same cell was more powerful than prediction across the different cells.

Establishment of the enhancer recognition website

To show our results and help researchers in the same research field, we constructed an enhancer recognition website (http://203.195.175.196/) with the trained enhancer subconvolution neural network model. The website was built using the Flask framework, which is a concise and convenient Python Web framework. At present, the primary function of the website is to input a sequence of less than 1000 bp length into the input box of the homepage, which can generate a prediction probability after model calculation (Fig. 7). It is helpful using the probability from our model result to judge whether the sequence is an enhancer or not. The website also includes some information about the process, data, results and contact information related to the present experiment. Our website provides a communication platform for the related research of enhancer recognition based on machine learning, which may be helpful for the research of enhancers and *cis*-acting elements.

Table 3 Comparison of the AUC results for CNN and gkmSVM. Each row represents the cell type for training, and each column presents the cell type for the test. For each model, we used 5-fold cross-validation. Values in bold and italics indicate down and up, respectively

Algorithm		Spermatids	Spermatocyte	Spermatogonia
CNN	Spermatids	*0.8968*	*0.9149*	*0.6273*
	Spermatocyte	*0.8447*	*0.9561*	*0.6328*
	Spermatogonia	**0.5274**	*0.5738*	*0.8057*
SVM	Spermatids	0.8873	0.5192	0.543
	Spermatocyte	0.8208	0.9504	0.5109
	Spermatogonia	0.5847	0.5014	0.7533



CNN-Enhancer

This is an enhancer recognition tool based on convolutional neural network model. The training data of this model are derived from spermatogonia, spermatocytes and sperm cells during spermatogenesis.

Please input sequence

▾ Info

1. Sequence length should be 1-1000 base lengths

2. It's just a possible outcome.

Method

Fig. 7 Homepage of the developed website.

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

Conclusions and discussion

In this paper, we proposed a deep CNN to solve the enhancer identification problem in mouse spermatogenesis. We used P300 data, enhancer data that were experimentally verified, and four histone training datasets of three sperm cells during spermatogenesis. We compared the trained model with the traditional machine learning model gkmSVM. The CNN model outperformed the SVM algorithm in terms of performance, especially when it comes to across cell prediction. The CNN model has better prospects for predicting enhancers in different types of cells. During the experiment, the volume of our visual layer was compared with the bioinformatics database and its biological significance explored, which proved that the model could be used to study the mechanism of sperm enhancer, further providing statistical support. In the model training, we adopted the transfer learning strategy, and we noted that not only was the training speed of the CNN faster, but also the model generalization was stronger compared with other models. Researchers can use the present strategy based on the existing CNN model to quickly train a suitable model for their experiments.

Compared with using CPU only, the GPU can accelerate the training speed of the model. We can also try the Recurrent Neural Networks (RNN) algorithm. Generally speaking, the information sequence is rich in a large number of sequences. The information has a complex time correlation and overall logicality, and the information length varies. The traditional neural network can't be solved. Thus, RNN formally solves this sequence problem that arises at this historic moment.

In the future, we can use more enhancers to identify information, such as CpG Islands, evolutionary conservation, and sequence features. Also, CNN can map the whole genome of other types of chromatin elements and further annotate the whole genome regulatory code. We can further explore the role of recognition enhancers in spermatogenesis by conducting experimental studies. Our model provides the possibility to study the recognition and association for specific base expression regulated by multiple enhancers.

Author contributions

M. Z. L., Q. L., T. T. L., and J. L. H. conceived and designed the study. C. Z. S., H. L., X. X., X. L. W. and L. J. L. performed the experiments. X. L. W., D. X. C., X. D. W., P. Z., Z. Y. C., T. L., P. Y., J. B. Z., and N. Z. contributed to the data analyses. C. Z. S., N. Z. and M. Z. L., A. S., X. L. W. wrote the manuscript. All authors reviewed and commented on the manuscript, and approved it in its final form.

Conflicts of interest

The authors declare that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China [grant numbers 61772431 and 31572399]; the Natural Science Fundamental Research Plan of Shaanxi Province [grant number 2018JM6039]; and the Program of Shaanxi Province Science and Technology Innovation Team (2019TD-036).

References

1 B. R. Jasny and L. Roberts, *Science*, 2004, **306**, 629.
2 S. M. Kehoe, M. Oka, K. E. Hankowski, N. Reichert, S. Garcia, J. R. Mccarrey, S. Gaubatz and N. Terada, *Biol. Reprod.*, 2008, **79**, 921–930.
3 K. M. Lele and D. J. Wolgemuth, *Biol. Reprod.*, 2004, **71**, 1340–1347.
4 P. P. Reddi, C. J. Urekar, M. M. Abhyankar and S. A. Ranpura, *Ann. N. Y. Acad. Sci.*, 2010, **1120**, 95–103.
5 N. D. Heintzman, R. K. Stuart, G. Hon, Y. Fu, C. W. Ching, R. D. Hawkins, L. O. Barrera, S. V. Calcar, C. Qu, K. A. Ching, W. Wang, Z. Weng, R. D. Green, G. E. Crawford and B. Ren, *Nat. Genet.*, 2007, **39**, 311–318.
6 K. J. Won, I. Chepelev, B. Ren and W. Wang, *BMC Bioinf.*, 2008, **9**, 1–18.
7 M. Fernández and D. Mirandasaavedra, *Nucleic Acids Res.*, 2012, **40**, e77.
8 N. Rajagopal, W. Xie, Y. Li, U. Wagner, W. Wang, J. Stamatoyannopoulos, J. Ernst, M. Kellis and B. Ren, *PLoS Comput. Biol.*, 2013, **9**, e1002968.
9 S. Chen, M. Gan, H. Lv and R. Jiang, *bioRxiv*, 2018, 398115.
10 D. Cohn, O. Zuk and T. Kaplan, *bioRxiv*, 2018, 264200.
11 D. Lee, R. Karchin and M. A. Beer, *Genome Res.*, 2011, **21**, 2167–2180.
12 M. Ghandi, D. Lee, M. Mohammad-Noori and M. A. Beer, *PLoS Comput. Biol.*, 2014, **10**, e1003711.
13 S. G. Kim, M. Harwani, A. Grama and S. Chaterji, *Sci. Rep.*, 2016, **6**, 38433.
14 L. Feng, L. Hao, R. Chao, X. Bo and W. Shu, *Sci. Rep.*, 2016, **6**, 28517.
15 Y. Sun, X. Wang and X. Tang, Proceedings CVPR, 2014, pp. 1891–1898.
16 A. Krizhevsky, I. Sutskever and G. E. Hinton, Neural Information Processing Systems, 2012.
17 M. Lovino, G. Urgese, E. Macii, S. Di Cataldo and E. Ficarra, *Int. J. Mol. Sci.*, 2019, **20**, 1645.
18 H. Y. Lin, L. Q. Chen and M. L. Wang, *Sensors*, 2019, **19**, 2250.
19 J. Zhou and O. G. Troyanskaya, *Nat. Methods*, 2015, **12**, 931–934.
20 B. Alipanahi, A. Delong, M. T. Weirauch and B. J. Frey, *Nat. Biotechnol.*, 2015, **33**, 831.
21 A. Khan, O. Fornes, A. Stigliani, M. Gheorghe, J. A. Castro-Mondragon, R. van der Lee, A. Bessy, J. Cheneby, S. R. Kulkarni, G. Tan, D. Baranasic, D. J. Arenillas, A. Sandelin, K. Vandepoele, B. Lenhard, B. Ballester, W. W. Wasserman,

Published on 29 May 2020. Downloaded by Uppsala University on 7/20/2020 2:37:52 AM.

F. Parcy and A. Mathelier, *Nucleic Acids Res.*, 2018, **46**, D260–D266.

22 S. P. Wang, Z. Tang, C. W. Chen, M. Shimada, R. P. Koche, L. H. Wang, T. Nakadai, A. Chramiec, A. V. Krivtsov, S. A. Armstrong and R. G. Roeder, *Mol. Cell*, 2017, **67**(308–321), e306.

23 S. S. Hammoud, D. H. Low, C. Yi, D. T. Carrell, E. Guccione and B. R. Cairns, *Cell Stem Cell*, 2014, **15**, 239–253.

24 A. Visel, S. Minovitsky, I. Dubchak and L. A. Pennacchio, *Nucleic Acids Res.*, 2007, **35**, D88–D92.

25 N. Vo and R. H. Goodman, *J. Biol. Chem.*, 2001, **276**, 13505–13508.

26 N. D. Heintzman, R. K. Stuart, G. Hon, Y. Fu, C. W. Ching, R. D. Hawkins, L. O. Barrera, S. Van Calcar, C. Qu and K. A. Ching, *Nat. Genet.*, 2007, **39**, 311–318.

27 S. Kimmins and P. Sassonecorsi, *Nature*, 2005, **434**, 583–589.

28 B. Bioinformatics, Babraham Institute, Cambridge, UK, 2011.

29 B. Langmead and S. L. Salzberg, *Nat. Methods*, 2012, **9**, 357.

30 Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E. Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li and X. S. Liu, *Genome Biol.*, 2008, **9**, R137.

31 B. Yu and K. Kumbier, *Front. Inform. Tech. Electro. Eng.*, 2018, **19**, 6–9.

32 C. M. Bishop, *Learning*, Springer, New York, 2006.

33 T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang and Z. Zhang, 2015, arXiv preprint arXiv:1512.01274.

34 X. Jin, J. Chi, S. Peng, *et al.*, 2016 8th International Conference on Wireless Communications & Signal Processing (WCSP), 2016, pp. 1–6.

35 D. P. Kingma and J. Ba, 2014, arxiv.org/abs/1412.6980.

36 H. Robbins and S. Monro, *Ann. Math. Stat.*, 1951, **22**, 400–407.

37 J. A. Hanley and B. J. McNeil, *Radiology*, 1982, **143**, 29–36.

38 M. Buda, A. Maki and M. A. Mazurowski, *Neural Networks*, 2018, **106**, 249–259.

39 S. Aerts, H. J. Van, O. Sand and B. A. Hassan, *PLoS One*, 2007, **2**, e1115.

40 O. Fornes, J. A. Castro-Mondragon, A. Khan, R. van der Lee, X. Zhang, P. A. Richmond, B. P. Modi, S. Correard, M. Gheorghe, D. Baranasic, W. Santana-Garcia, G. Tan, J. Cheneby, B. Ballester, F. Parcy, A. Sandelin, B. Lenhard, W. W. Wasserman and A. Mathelier, *Nucleic Acids Res.*, 2020, **48**, D87–D92.

41 S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey and W. S. Noble, *Genome Biol.*, 2007, **8**, R24.

42 T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li and W. S. Noble, *Nucleic Acids Res.*, 2009, **37**, W202–W208.

43 S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey and W. S. Noble, *Genome Biol.*, 2007, **8**, R24.

44 G. Tyagi, K. Carnes, C. Morrow, N. V. Kostereva, G. C. Ekman, D. D. Meling, C. Hostetler, M. Griswold, K. M. Murphy, R. A. Hess, M.-C. Hofmann and P. S. Cooke, *Biol. Reprod.*, 2009, **81**, 258.

45 J. Chen and H. E. Ruley, *J. Biol. Chem.*, 1998, **273**, 24670–24675.

46 M. Ghandi, D. Lee, M. Mohammad-Noori and M. A. Beer, *PLoS Comput. Biol.*, 2014, **10**(7), e1003711.

47 M. P. Creyghton, A. W. Cheng, G. G. Welstead, T. Kooistra, B. W. Carey and E. J. Steine, *Proc. Natl. Acad. Sci. U. S. A.*, 2010, **107**, 21931–21936.

48 J. Davis and M. Goadrich, Proceedings of the 23rd international conference on Machine learning, ACM, 2006, pp. 233–240.