frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 14 December 2022
DOI 10.3389/fgene.2022.1081842

OPEN ACCESS

EDITED BY
Quan Zou,
University of Electronic Science and Technology of China, China

REVIEWED BY
Wen Zhang,
Huazhong Agricultural University, China
Feng Fang,
University of Houston, United States
Qiong Zhang,
Affiliated Hospital of Nantong University, China

*CORRESPONDENCE
Minzhu Xie,
xieminzhu@hunnu.edu.cn

SPECIALTY SECTION
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

RECEIVED 27 October 2022
ACCEPTED 28 November 2022
PUBLISHED 14 December 2022

CITATION
Chen Y, Xie M and Wen J (2022), Predicting gene expression from histone modifications with self-attention based neural networks and transfer learning.
Front. Genet. 13:1081842.
doi: 10.3389/fgene.2022.1081842

Predicting gene expression from histone modifications with self-attention based neural networks and transfer learning

Yuchi Chen, Minzhu Xie* and Jie Wen

College of Information Science and Engineering, Hunan Normal University, Changsha, China

It is well known that histone modifications play an important part in various chromatin-dependent processes such as DNA replication, repair, and transcription. Using computational models to predict gene expression based on histone modifications has been intensively studied. However, the accuracy of the proposed models still has room for improvement, especially in cross-cell lines gene expression prediction. In the work, we proposed a new model TransferChrome to predict gene expression from histone modifications based on deep learning. The model uses a densely connected convolutional network to capture the features of histone modifications data and uses self-attention layers to aggregate global features of the data. For cross-cell lines gene expression prediction, TransferChrome adopts transfer learning to improve prediction accuracy. We trained and tested our model on 56 different cell lines from the REMC database. The experimental results show that our model achieved an average Area Under the Curve (AUC) score of 84.79%. Compared to three state-of-the-art models, TransferChrome improves the prediction performance on most cell lines. The experiments of cross-cell lines gene expression prediction show that TransferChrome performs best and is an efficient model for predicting cross-cell lines gene expression.

KEYWORDS

gene expression, histone modification, deep learning, transfer learning, convolutional neural network

Introduction

Understanding the patterns of gene regulation has been one of the major focuses of biological research. A variety of biological factors are thought to be involved in the regulation of gene expression. The regulatory factors usually include transcription factors, cis-regulatory elements, and epigenetic modifications. As a type of epigenetic modifications, histone modification plays an important role in gene expression regulation (Gibney and Nolan, 2010). Nucleosome is the building block of a chromosome, which consists of an octamer of histones and 147 base pair (bp) DNA wrapping around the octamer. Since histone is a core component of the nucleosome, histone modifications directly affect the structure of chromatin and control the expression intensity of nearby genes. Recently, a great number of researches have shown that histone modifications have a great impact on gene expression, chromosome inactivation, replication, and cell differentiation (Krajewski, 2022; Lin et al., 2022). There are a variety of histone modification marks at different chromosomal locations, and there may be a set of "codes" of histone modifications to control gene expression (Peterson and Laniel, 2004). Due to the high-throughput sequencing technologies, a huge amount of histone modifications data and gene expression data are available, and using computational algorithms to predict gene expression based on histone modifications is feasible.

To date, a variety of computational methods have been used to predict gene expression based on gene regulatory factors. For example, Beer and Tavazoie, (2004) used Bayesian networks to predict gene expression from DNA sequences. Ouyang et al. (2009) used a linear regression model to predict gene expression based on 12 transcription factors. Zeng et al. (2020) combined the information of proximal promoter and distal enhancer to predict gene expression. In 2010, Karlić et al.(2010) found histone modification levels and gene expression are well correlated, and derived quantitative models to predict gene expression from histone modifications. Li et al. (2015) used a machine learning method to predict gene expression in lung cancer from multiple epigenetic data such as CpG methylation, histone H3 methylation modification and nucleotide composition. In 2016, Singh et al. (Singh et al., 2016) used a convolutional neural network (CNN) DeepChrome to predict gene expression based on five critical histone modification marks. To improve prediction accuracy, they (Singh et al., 2017) integrated attention mechanism into a neural network and proposed a prediction model AttentiveChrome. Temporal Convolutional Network (Zhu et al., 2018; Kamal et al., 2020) is also utilized to predict the gene expression from histone modifications. In 2022, Hamdy et al. (2022) proposed three variations of CNN models called ConvChrome.

Though the above methods have achieved good performances, there are still room for improvement, and some recently emerging technologies have provided some ways. When models are trained and tested on different cell lines which is knowns as cross-cell lines prediction, the model performance is always compromised. For example, compared to training and testing on the same cell line dataset, the average prediction accuracy of DeepChrome trained on other cell lines is 2.3% lower. Because of the large variety of cell lines, it is difficult to obtain histone modification data and gene expression data for all types of cell lines. Therefore predicting gene expression using models trained on other cell lines is useful and in urgent need.

Transfer learning is a machine learning technique in which a model trained on a specific task is reused as part of the training process for another similar task (Tan et al., 2018). Transfer learning allows training and prediction using the dataset from different sources with similar characteristics and significantly reduces dataset bias. Transfer learning has achieved great success in prediction tasks that require learning transfer features (Sun et al., 2022; Zhu et al., 2022). In the field of bioinformatics, transfer learning enables existing trained models to efficiently work on similar datasets that are lack of labels, which reduces the cost of biological experiments.

TABLE 1 Five core histone modification marks and their functional categories.

Histone mark	Associated regions	Functional category
H3K27me3	Polycomb repression	Repressor mark
H3K36me3	Transcribed regions	Structural mark
H3K4me1	Enhancer regions	Distal mark
H3K4me3	Promoter regions	Promoter mark
H3K9me3	Heterochromatin regions	Repressor mark

In the paper, we propose a neural network model TransferChrome with self-attention mechanism and transfer learning to predict gene expression based on histone modifications data. TransferChrome uses neural network layers with self-attention mechanism to capture global contextual information of data. In order to correct the data bias of cross-cell lines gene expression prediction, we used transfer learning. The experimental results show that TransferChrome achieved an average Area Under the Curve (AUC) score of 84.79%, which is better than other 3 state-of-the-art similar models. The cross-cell lines prediction experiments also show that TransferChrome outperforms other models.

Materials and method

Data collection and processing

The experimental data comes from the Roadmap Epigenome Project (REMC) (Kundaje et al., 2015), which consists of 56 cell lines' histone modifications data and the corresponding normalized RPKM expression data of 17170 samples. Same as DeepChrome (Singh et al., 2016), five histone modification marks that play important roles in gene expression were selected for our experiments. These 5 marks include H3K4me3, H3K4me1, H3K36me3, H3K27me3, and H3K9me3. Their functional categories are summarized in Table 1. Each sample in the dataset represents a gene. The data of one sample include the five histone modification marks signal within 10000bp upstream and downstream of the transcription start sites (TSSs) of the corresponding gene.

According to DeepChrome (Singh et al., 2016), the 10000 bp is equally divided into 100 bins and the histone modifications data of one sample is encoded into an $n \times m$ matrix x, where n is

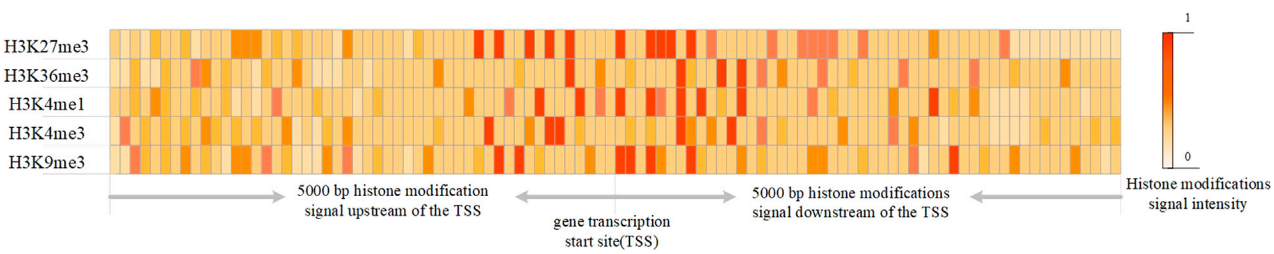

FIGURE 1
The data structure representing histone modifications.

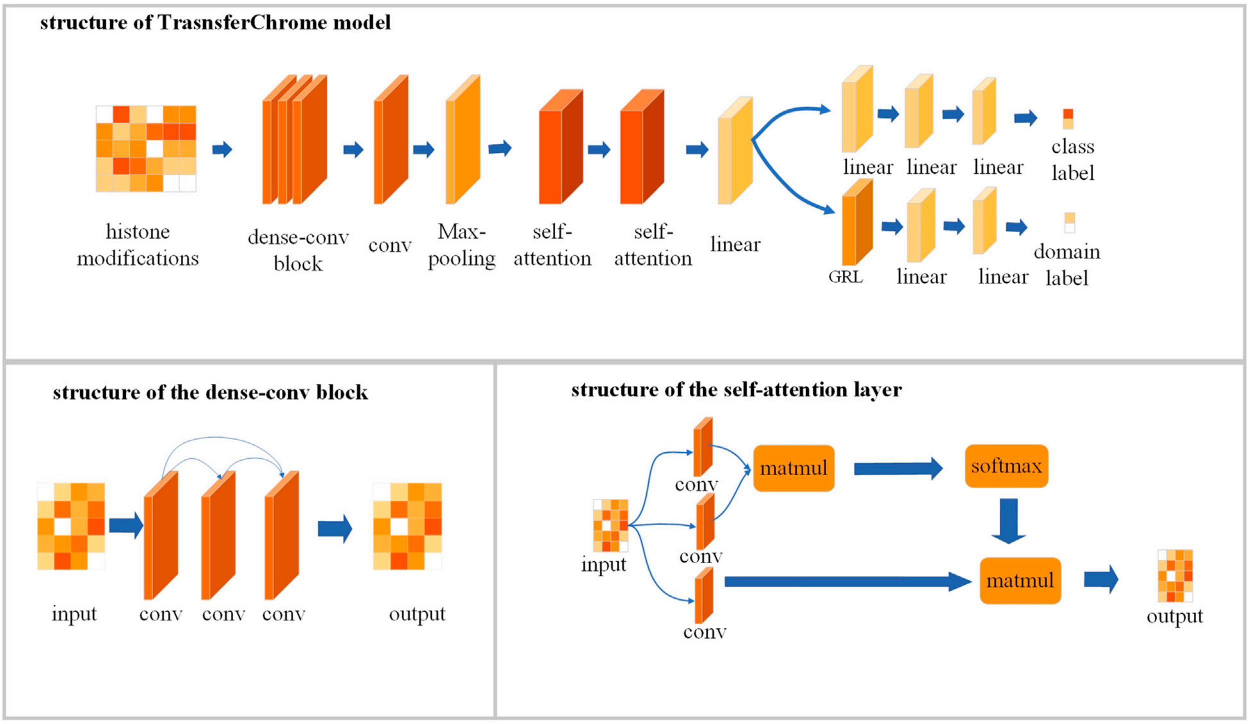

FIGURE 2
The model structure of TransferChrome.

the number of histone modification marks and m is the number of bins (see Figure 1). The histone sequencing data provided by REMC were quantified by BEDTools into histone modification signals. Therefore, $x_{i,j}$ represents the signal of the j-th histone modification mark in the i-th bin.

Since the normalization of training data can speed up the convergence of model training and allows the model to fit the data better (Singh and Singh, 2020), the z-score method is used to normalize the data for each histone modification mark as Eq. 1. In Eq. 1, $\hat{x}_{i,h}$ represents the normalized signal of h-th histone modification mark in the i-th bin. $\bar{x}_h$ and σ_h denote the mean and standard deviation of the signals of the h-th histone modification mark of all genes in a cell line.

$$\hat{x}_{i,h} = \frac{x_{i,h} - \bar{x}_h}{\sigma_h} \tag{1}$$

According to previous studies (Singh et al., 2016), each gene is assigned a label based on its expression value. The median of expression values of all genes in a given cell line is denoted as t. If the expression value of a gene is higher than or equal to t, it is labeled with 1; otherwise it is labeled with 0.

To avoid interfering from adjacent genes, those genes whose TSSs are within 5000 bp downstream of previous kept genes' TSSs are deleted. At last there are 17170 genes remained.

Design of neural network model

As shown in Figure 2, TransferChrome is composed of multiple modules: a feature extraction module, a label classification module and a domain classification module. The feature extraction module is used to calculate the latent features of data. It includes a dense-conv block, a 1D convolutional layer, a 1D max pooling layer, two self-attention layers, and a linear layer (also called fully connected layer or dense layer). The label classification module predicts a gene expression label. It includes three linear layers. The domain classification module predicts a domain label, which includes a gradient reversal layer (GRL) and two linear layers. It learns transfer features, which allows the model to achieve better performance in cross-cell lines gene expression prediction.

Dense-connected convolutional layer for extracting local features of data

To make the model better capture the features of the data, we optimized the convolutional neural network in the feature extraction module. The convolutional neural network uses feature detectors, also known as convolution kernals or filters to capture data's features. According to its size, a convolution kernal will aggregate all the information in the receptive field to extract a corresponding feature. By increasing the number of convolutional layers, a model can learn more complex features. However, the deepened network structure easily ignores the features captured by earlier convolutional layers, which usually represent the simple but also basic features of the data. Densely Connected Convolutional Networks (DenseNet) (Huang et al., 2017) uses a method called dense connectivity pattern enhances the reusability of features. Compared with the classic Convolutional Network, DenseNet connects convolutional layers densely so that the feature extracted by each layer could be used repeatedly. DenseNet performs a deep supervision and strengthen the weights of features captured by earlier convolutional layers. Inspired by DenseNet, a dense-conv block that contains several densely connected convolutional layers is used to extract features in our model. A dense-connected convolutional layer is a convolutional layer which connected to all other convolutional layers directly. It means that the input of a dense connected convolutional layer not only comes from its adjacent convolutional layer, but also from other preceding layers. The dense-conv block allows the model to learn the complex features of the data while also ensuring that the low-level convolutional layer retains a greater influence in the model's decision-making.

Let x_l be the output of the lth densely connected convolutional layer. The input of the lth dense connected convolutional layer contains the outputs of all the previous l–1 layers as Eq. 2 shows.

$$x_l = H([x_0, x_1, \ldots\ldots x_{l-1}]) \tag{2}$$

$[x_0, x_1, \ldots\ldots x_{l-1}]$ refers to the concatenation of the feature-maps output from preceding layers. The composite function H concludes a rectified linear unit (RELU) and 1D convolutional layer.

In TransferChrome, the dense-conv block consists of three dense-connected convolutional layers (kernal number = 32, 16, 8 and kernal length = 5, 5, 5). The dense-conv block is followed by a convolutional layer (kernal number = 50 and kernal length = 5) and a max-pooling layer (kernal length = 2). A dropout layer is added after each convolutional layer and the dropout rate is 0.4. The output of the max-pooling layer is input into a following self-attention layer.

Self-attention layer for aggregating global information

Regulatory factors at different locations may interact and act on gene expression together. Therefore effective integration of upstream and downstream information in the genome usually leads to better computational results (Ji et al., 2021). The Transformer (Vaswani et al., 2017) is an efficient neural network model which has achieved good results in many fields such as natural language processing (Devlin et al., 2019) and image recognition (Dosovitskiy et al., 2021). The self-attention mechanism used in Transformer can effectively integrate data's global features. The self-attention mechanism also has been widely adopted in the field of Bioinformatics (Avsec et al., 2021; Ji et al., 2021). For example, the researchers used this mechanism to significantly improve the regulatory elements prediction from genomic DNA sequences (Avsec et al., 2021). Previous experiments (Singh et al., 2017, 2016) have illustrated that histone modifications closer to gene's TSS have greater influence in the gene expression. We add self-attention mechanism to the model, and use a position encoding function to concatenate input data with relative distance information. The relative distance information contains the relative distance between each point and the TSS point, and the output of the position encoding function is denoted by x.

TransferChrome contains two self-attention layers to capture the long-distance dependence. The function of each self-attention layer is as Eqs 3–6.

$$Q = conv_q(x) \tag{3}$$

$$K = conv_k(x) \tag{4}$$

$$V = conv_v(x) \tag{5}$$

$$Attention(Q, K, V) = soft\max(QK^T)V \tag{6}$$

Each self-attention layer uses three one-dimensional convolutional layers (kernal length = 1) $conv_q$, $conv_k$, and $conv_v$ to calculate a query matrix Q, a key matrix K and a latent variable matrix V, respectively. The number of output channels of $conv_q$ and $conv_k$ is half of the number of input channels, and the number of output channels of $conv_v$ is equal to the number of input channels. Then the self-attention layer calculate data's attention score matrix QK^T by multiplying (matmul) Q and K. Finally, the attention score matrix will be normalized by a softmax function, and multiplied with V. At the end of the feature extraction module, there is a linear layer following the last self-attention layer. The feature extraction module outputs a low-dimensional feature vector. Then the feature vector is inputted into the label classification module and the domain classification module at the same time.

Label classification module and domain classification module

The label classification module predicts the gene expression label of the sample, which is the main task of our model. It is a binary classification task with 0 and 1 represent low expression and high expression, respectively. According to Long et al. (2015) and Ganin and Lempitsky (2015), domain adaption can improve prediction accuracy in transfer learning. Therefore, TransferChrome uses a domain classification module for cross-cell lines prediction. It contains a GRL and two linear layers. GRL acts as an identity transform in the forward propagation of the model. In the backward propagation, GRL takes the gradient from the subsequent layer and multiplies it by a parameter $-\lambda$ with $\lambda > 0$ and passes it to the preceding layer.

The domain classification module predicts whether the sample belongs to the target domain or the source domain. Source domains are the cell lines whose genes have gene expression labels, and the cell lines whose gene have no known gene expression information are called target domains. It is also a binary classification task with 0 and 1, where 0 indicates that the sample is from the target domain and 1 indicates that the sample is from the source domain. In cross-cell lines prediction, we try to extract those features that can not be used to discern the data domain.

Model training

For model training, we chose cross entropy as the loss function:

$$L = -[y \log \hat{y} + (1-y)\log(1-\hat{y})]. \tag{7}$$

Let G_f and θ_f be the function and the parameters of the feature extraction module, respectively. Let G_d (G_y) and θ_d (θ_y) be the function and the parameters of the domain (label) classification module, respectively. For the single cell line gene expression prediction task, the optimization goal of model training is to minimize the loss L_y of the label classification module without considering the domain classification module.

For the cross-cell lines gene expression prediction task, we train TransferChrome using the complete dataset from a source domain and a part of the dataset from a target domain to capture transfer features in different cell lines, and aim to minimize the objective function in Eq. 8.

$$\begin{aligned} E(\theta_f, \theta_y, \theta_d) &= \sum_{i=1}^{N} L_y\big(G_y\big(G_f(x_i;\theta_f);\theta_y\big), y_i\big) \\ &\quad -\lambda \sum_{i=1}^{N} L_d\big(G_d\big(G_f(x_i;\theta_f);\theta_d\big), d_i\big) \\ &= \sum_{i=1}^{N} L_y^i(\theta_f,\theta_y) - \lambda \sum_{i=1}^{N} L_d^i(\theta_f,\theta_d), \end{aligned} \tag{8}$$

where L_d is the loss function of the domain classification module.

In the training process, stochastic gradient descent (SGD) is used to update θ_y and θ_d to minimize the label classification loss L_y and L_d. In the backward propagation, the first layer GRL of the domain classification module reverses the gradient by multiplying a negative number $-\lambda$ and backward propagates it to the feature extraction module. After the model training, G_f is expected to extract transfer features in different cell lines. In the training process, the learning rate is set to 0.001, momentum is 0.85, and weight decay is 0.001. We set the max training epochs to 200 and adopted early stop strategy.

In the following single-cell line prediction experiments, each cell line data was partitioned into a training set, a validation set and a test set as DeepChrome (Singh et al., 2016). For cross-cell lines prediction, we used the source domain data and half of the target domain data to train our model, and used the other half of the target domain data as the test set.

Experiments

Comparison with other existing state-of-the-art methods

To evaluate the effectiveness of TransferChrome, we compared it with three state-of-the-art models (DeepChrome, AttentiveChrome, and ConvChrome_CNN1D). DeepChrome (Singh et al., 2016) is a convolutional neural network. It consists of a convolutional layer (convolution kernal size is 10, the number of convolution kernals is 50), a max pooling layer (convolution kernal size is 10), and two fully connected layers (the number of units is 900, 125). AttentiveChrome (Singh et al., 2017) is a recurrent neural network that uses two attention mechanisms. ConvChrome (Hamdy et al., 2022) includes three

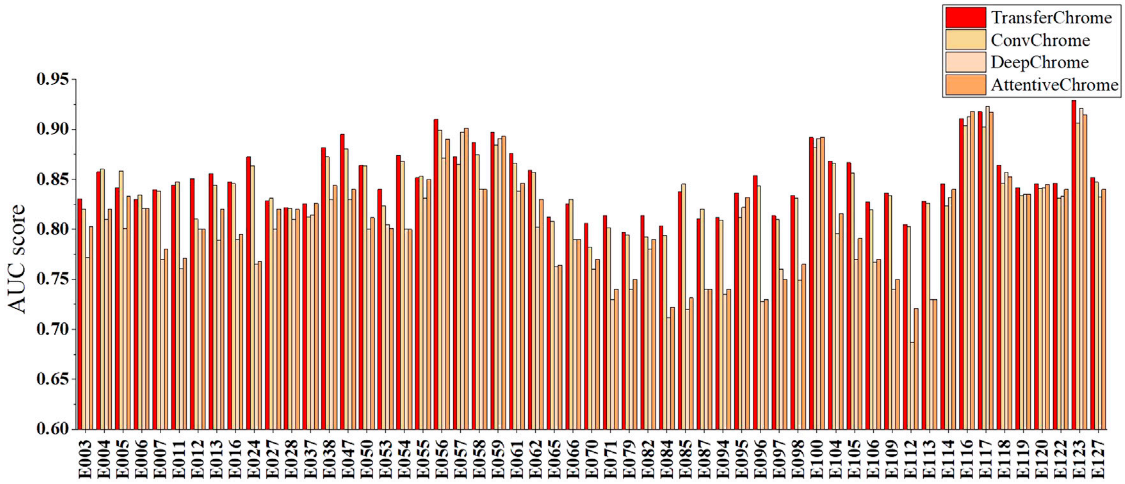

FIGURE 3
Single cell line gene expression prediction performance comparison on 56 cell lines of the models.

TABLE 2 The minimum, mean, maximum, and median of the AUC scores of single cell line gene expression prediction of the models on 56 cell lines.

	Min	Mean	Max	Median
TransferChrome	0.7972	0.8479	0.9289	0.8449
ConvChrome	0.7820	0.8399	0.9061	0.8386
DeepChrome	0.6871	0.8003	0.9236	0.8019
AttentiveChrome	0.7221	0.8093	0.9197	0.8216

variations of CNN models, among which ConvChrome_CNN1D achieved the best performance. In the experiments, all models used the same five types of core histone modifications from the REMC project to predict gene expression. In the experiments of single-cell gene expression prediction, we did not use the domain classification module of TransferChrome and only used the label classification module. DeepChrome was implemented and trained according to Singh's paper (Singh et al., 2016). For AttentiveChrome, we used the trained model downloaded from http://kipoi.org/models/AttentiveChrome/. Because

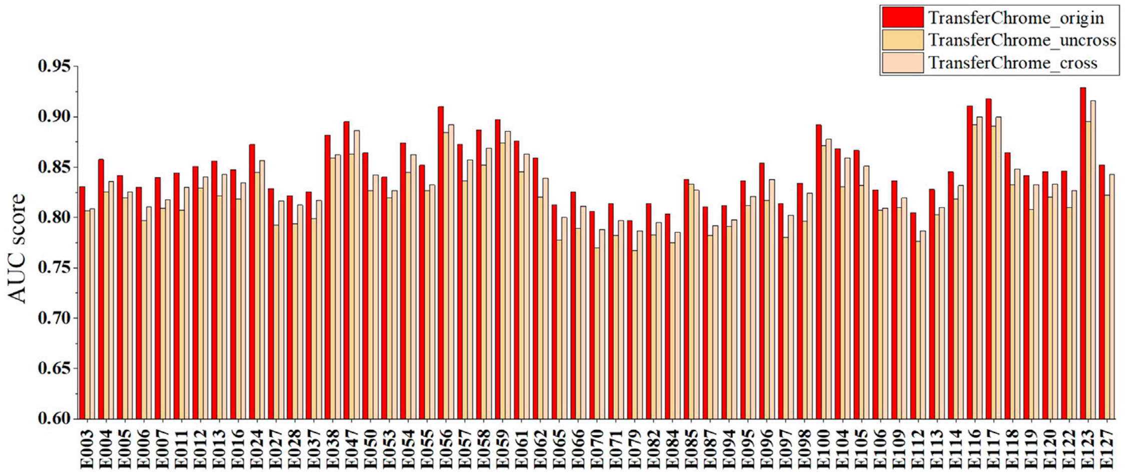

FIGURE 4
The performance comparison of different versions of TransferChrome: TransferChrome_cross, TransferChrome_uncross and TransferChrome_origin.

TABLE 3 Comparison of the minimum, mean, maximum, and median of the AUC scores of TransferChrome, DeepChrome and ConvChrome in single cell line and cross-cell lines gene expression predictions.

	Min	Mean	Max	Median
TransferChrome_origin	0.7972	0.8479	0.9289	0.8449
TransferChrome_uncross	0.7672	0.8185	0.8950	0.8182
TransferChrome_cross	0.7866	0.8330	0.9156	0.8300
DeepChrome	0.6871	0.8003	0.9236	0.8019
DeepChrome_uncross	0.6652	0.7737	0.8951	0.7710
ConvChrome	0.7820	0.8399	0.9061	0.8386
ConvChrome_uncross	0.7571	0.8111	0.8791	0.8076

ConvChrome's code and data are not available, We implement ConvChrome_CNN1D with PyTorch. We used the AUC score as our evaluation metric. Experimental results on 56 cell lines are shown in Figure 3 and Table 2, compared with the other models, TransferChrome improved the prediction accuracy on most cell lines. TransferChrome has a significant improvement in average AUC compared to DeepChrome and AttentiveChrome. Compared with ConvChrome, TransferChrome also has better performance.

Cross-cell lines gene expression prediction performance comparison

For the performance comparison in cross-cell lines gene expression prediction, we arbitrarily selected a cell line (E085) as the source domain and each one of other cell lines as the target domain. Figure 4 and Table 3 show the experimental results. In Figure 4 and Table 3, TransferChrome_cross means that TransferChrome was trained and tested on different cell lines. TransferChrome_uncross and TransferChrome_origin did not use the domain classification module, and TransferChrome_origin was trained and tested on the same cell lines, while TransferChrome_uncross was trained and tested on different cell lines. Table 3 shows the performance comparison of TransferChrome, DeepChrome and ConvChrome in cross-cell lines gene expression prediction. In Table 3, DeepChrome and ConvChrome indicate that the models were trained and tested on the same cell line, while DeepChrome_uncross and ConvChrome_uncross indicate that those models were trained with E085 cell line's data but were tested on other cell lines.

The results have shown that the average AUC of TransferChrome_uncross trained in a E085 cell line and tested on another dropped by 2.9% compared to those of TransferChrome_origin trained and tested on a same cell line. Similarly, the average AUCs of DeepChrome_uncross and ConvChrome_uncross dropped by 2.6% and 2.9% compared to those of DeepChrome and ConvChrome, respectively. Though TransferChrome_cross did not achieve the same effect as TransferChrome_origin, the average AUC drop is reduced to 1.5%, which showed that using domain classification module indeed improves the performance in cross-cell lines prediction.

Contributions of dense connectivity pattern and different position encoding functions

We carried experiments to see whether dense connectivity pattern and different position encoding functions have obvious impact on the performance of TransferChrome. A total of 9 cell lines out of 56 with worst (E079, E084, E112), median (E114, E120, E128), and best (E116, E117, E123) AUC scores were selected for ablation experiments.

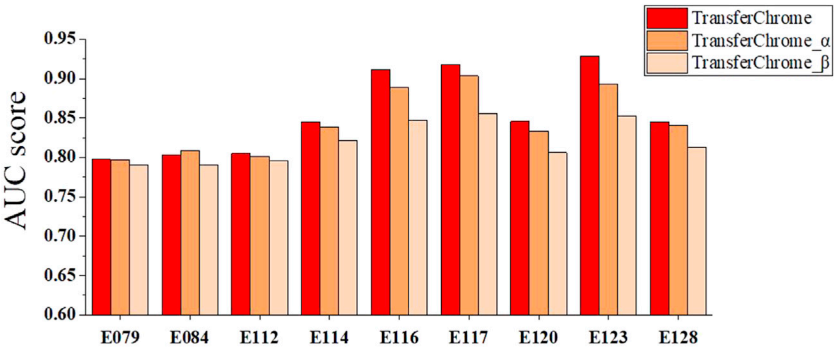

FIGURE 5
The performance comparison of different versions of TransferChrome: TransferChrome, TransferChrome_α and TransferChrome_β on 9 cell lines (E079, E084, E112, E114, E120, E128, E116, E117, E123).

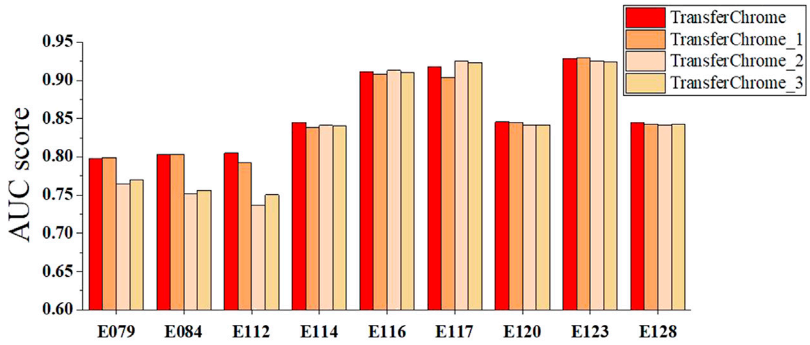

FIGURE 6
The performance comparison of different versions of TransferChrome: TransferChrome, TransferChrome_1, TransferChrome_2 and TransferChrome_3 on 9 cell lines (E079, E084, E112, E114, E120, E128, E116, E117, E123).

We compared three TransferChrome model variations to discuss the contribution of different position encoding functions. The position encoding function adopted by TransferChrome calculates the relative distance between TSS and bins. TransferChrome_α use sinusoidal position encoding (Vaswani et al., 2017) as the position function. Sinusoidal position encoding function calculates position information with a mix of sine and cosine functions. TransferChrome_β is the TransferChrome without adding position information. The experimental results are shown in Figure 5.

Histone modifications at different positions have different importance for gene expression prediction. Since TransferChrome_β ignores sequence position information, TransferChrome_β performance worser than TransferChrome and TransferChrome_α. Meanwhile, histone modifications which are close to TSS might have more significant effect on gene expression (Cheng et al., 2011). Accordingly, bins near to TSS should be assigned with higher weights for gene expression prediction (Singh et al., 2016). TransferChrome makes good use of relative distances between bins and TSS and performs better than TransferChrome_α.

We also conducted comparative experiments to discuss the contribution of the dense connectivity pattern and convolutional layer kernal numbers. As shown in Figure 6, four TransferChrome model variations are compared. TransferChrome has a dense-conv block, which has three dense-connected convolutional layers with different kernal numbers (32, 16, 8). TransferChrome_1 changes the structure of the dense-conv block. Dense-conv block of TransferChrome_1 uses three dense connected convolutional layers with 50 kernals. TransferChrome_2 and TransferChrome_3 do not use dense-conv block. TransferChrome_2 only has a convolutional layer with 50 kernals. TransferChrome_3 uses three convolutional layers with 50 kernals. Figure 6 shows the experimental results of above models. On 3 cell lines E079, E084 and E112, Transferchrome and TransferChrome_1 perform significantly better than others. TransferChrome uses fewer kernals than TransferChrome_1 but achieves a similar performance.

Conclusion and discussion

We proposed a new model called TransferChrome to predict gene expression levels based on histone modifications. TransferChrome uses self-attention mechanism to capture the long-distance dependence, and to learn hidden information features from the histone modifications data. Furthermore, TransferChrome adopts dense connectivity pattern to improve the feature exaction ability of convolutional neural network. Experimental results on the benchmark dataset of 56 cell lines showed that TransferChrome performed better than other 3 similar state-of-the-art models. To improve cross-cell lines gene expression prediction performance, TransferChrome uses transfer learning. Transfer learning makes the model capable of learning common features among different cell lines and reduces the data biases of different cell lines. Our experiments demonstrated that TransferChrome achieved the best accuracy in cross-cell lines gene expression prediction. We believe that it is useful to use transfer learning to improve cross-cell lines prediction accuracy. So far, gene expression prediction methods from histone modification data are mostly based on the five core histone modification marks. In future work, we will use more information from the histone modification data to predict gene expression. We also intend to increase the interpretability of the model in order to analyze the contribution of different histone modification marks on gene expression prediction.

Data availability statement

This study processed and analyzed publicly available data sets. These data can be found here: https://egg2.wustl.edu/roadmap/webportal/index.html.

Author contributions

YC and MX conceived the study and the conceptual design of the work. YC implemented the TransferChrome model and drafted the manuscript. JW collected the data and tested the model's performs. MX and YC polished the manuscript. All authors have read and approved the manuscript.

Funding

This work is supported by the National Natural Science Foundation of China under Grant 62172028 and Grant 61772197.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2022.1081842/full#supplementary-material

References

Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., et al. (2021). Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* 18, 1196–1203. doi:10.1038/s41592-021-01252-x

Beer, M. A., and Tavazoie, S. (2004). Predicting gene expression from sequence. *Cell.* 117, 185–198. doi:10.1016/S0092-8674(04)00304-6

Cheng, C., Yan, K.-K., Yip, K. Y., Rozowsky, J., Alexander, R., Shou, C., et al. (2011). A statistical framework for modeling gene expression using chromatin features and application to modencode datasets. *Genome Biol.* 12, R15–R18. doi:10.1186/gb-2011-12-2-r15

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). "Bert: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (Stroudsburg, Pennsylvania, USA: Association for Computational Linguistics), 4171–4186.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). "An image is worth 16x16 words: Transformers for image recognition at scale," in International Conference on Learning Representations.

Ganin, Y., and Lempitsky, V. (2015). "Unsupervised domain adaptation by backpropagation," in Proceedings of the 32nd International Conference on Machine Learning. Editors F. Bach and D. Blei (Cambridge, MA: PMLR), 1180–1189.

Gibney, E., and Nolan, C. (2010). Epigenetics and gene expression. *Heredity* 105, 4–13. doi:10.1038/hdy.2010.54

Hamdy, R., Maghraby, F. A., and Omar, Y. M. (2022). Convchrome: Predicting gene expression based on histone modifications using deep learning techniques. *Curr. Bioinform.* 17, 273–283. doi:10.2174/1574893616666211214110625

Huang, G., Liu, Z., Maaten, L. V. D., and Weinberger, K. Q. (2017). "Densely connected convolutional networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Honolulu, HI, USA: IEEE Computer Society), 2261–2269. doi:10.1109/CVPR.2017.243

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. (2021). Dnabert: Pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics* 37, 2112–2120. doi:10.1093/bioinformatics/btab083

Kamal, I. M., Wahid, N. A., and Bae, H. (2020). "Gene expression prediction using stacked temporal convolutional network," in 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), 402–405.

Karlić, R., Chung, H.-R., Lasserre, J., Vlahoviček, K., and Vingron, M. (2010). Histone modification levels are predictive for gene expression. *Proc. Natl. Acad. Sci. U. S. A.* 107, 2926–2931. doi:10.1073/pnas.0909344107

Krajewski, W. A. (2022). Histone modifications, internucleosome dynamics, and dna stresses: How they cooperate to "functionalize" nucleosomes. *Front. Genet.* 13, 873398. doi:10.3389/fgene.2022.873398

Kundaje, A., Meuleman, W., Ernst, J., Bilenky, M., Yen, A., Heravi-Moussavi, A., et al. (2015). Integrative analysis of 111 reference human epigenomes. *Nature* 518, 317–330. doi:10.1038/nature14248

Li, J., Ching, T., Huang, S., and Garmire, L. X. (2015). Using epigenomics data to predict gene expression in lung cancer. *BMC Bioinforma.* 16, S10–S12. doi:10.1186/1471-2105-16-S5-S10

Lin, W., Song, C., Meng, H., Li, N., and Geng, Q. (2022). Integrated analysis reveals the potential significance of hdac family genes in lung adenocarcinoma. *Front. Genet.* 13, 862977. doi:10.3389/fgene.2022.862977

Long, M., Cao, Y., Wang, J., and Jordan, M. (2015). "Learning transferable features with deep adaptation networks," in Proceedings of the 32nd International Conference on Machine Learning. Editors F. Bach and D. Blei (Cambridge, MA: PMLR), 97–105.

Ouyang, Z., Zhou, Q., and Wong, W. H. (2009). Chip-seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. *Proc. Natl. Acad. Sci. U. S. A.* 106, 21521–21526. doi:10.1073/pnas.0904863106

Peterson, C. L., and Laniel, M.-A. (2004). Histones and histone modifications. *Curr. Biol.* 14, R546–R551. doi:10.1016/j.cub.2004.07.007

Singh, D., and Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Appl. Soft Comput.* 97, 105524. doi:10.1016/j.asoc.2019.105524

Singh, R., Lanchantin, J., Robins, G., and Qi, Y. (2016). Deepchrome: Deep-learning for predicting gene expression from histone modifications. *Bioinformatics* 32, i639–i648. doi:10.1093/bioinformatics/btw427

Singh, R., Lanchantin, J., Sekhon, A., and Qi, Y. (2017). "Attend and predict: Understanding gene regulation by selective attention on chromatin," in *Advances in neural information processing systems*. Editors I. Guyon, U. V. Luxburg, S. Bengio,

H. Wallach, R. Fergus, S. Vishwanathan, et al. (Red Hook, NY, USA: Curran Associates, Inc.), 30.

Sun, J., Dodlapati, S., and Jiang, Z. C. (2022). Completing single-cell dna methylome profiles via transfer learning together with kl-divergence. *Front. Genet.* 13, 910439. doi:10.3389/fgene.2022.910439

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). "A survey on deep transfer learning," in *International conference on artificial neural networks*. Editors V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis (Berlin, Germany: Springer International Publishing), 270–279.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in *Advances in neural information processing systems*. Editors I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, et al. (Red Hook, NY, USA: Curran Associates, Inc.).

Zeng, W., Wang, Y., and Jiang, R. (2020). Integrating distal and proximal information to predict gene expression via a densely connected convolutional neural network. *Bioinformatics* 36, 496–503. doi:10.1093/bioinformatics/btz562

Zhu, L., Kesseli, J., Nykter, M., and Huttunen, H. (2018). "Predicting gene expression levels from histone modification signals with convolutional recurrent neural networks," in *EMBEC & NBC 2017*. Editors H. Eskola, O. Väisänen, J. Viik, and J. Hyttinen (Singapore: Springer Singapore), 555–558.

Zhu, X., Gu, Y., and Xiao, Z. (2022). Herbkg: Constructing a herbal-molecular medicine knowledge graph using a two-stage framework based on deep transfer learning. *Front. Genet.* 13, 799349. doi:10.3389/fgene.2022.799349