RESEARCH

Open Access

CGLoop: a neural network framework for chromatin loop prediction



Junfeng Wang¹, Lili Wu¹, Jingjing Wei², Chaokun Yan³, Huimin Luo³, Junwei Luo^{1*} and Fei Guo⁴

Abstract

Background Chromosomes of species exhibit a variety of high-dimensional organizational features, and chromatin loops, which are fundamental structures in the three-dimensional (3D) structure of the genome. Chromatin loops are visible speckled patterns on Hi-C contact matrix generated by chromosome conformation capture methods. The chromatin loops play an important role in gene expression, and predicting the chromatin loops generated during whole genome interactions is crucial for a deeper understanding of the 3D genome structure and function.

Results Here, we propose CGLoop, a deep learning based neural network framework that detects chromatin loops in Hi-C contact matrix. CGLoop combines the convolutional neural network (CNN) with Convolutional Block Attention Module (CBAM) and the Bidirectional Gated Recurrent Unit (BiGRU) to capture important features related to chromatin loops by comprehensively analyzing the Hi-C contact matrix, enabling the prediction of candidate chromatin loops. And CGLoop employs a density based clustering method to filter the candidate chromatin loops predicted by the neural network model. Finally, we compared CGloop with other chromatin loops prediction methods on several cell line including GM12878, K562, IMR90, and mESC. The code is available from https://github.com/wllwuliliwll/CGLoop.

Conclusions The experimental results show that, loops predicted by CGLoop show high APA scores and there is an enrichment of multiple transcription factors and binding proteins at the predicted loops anchors, which outperforms other methods in terms of accuracy and validity of chromatin loops prediction.

Keywords Neural network, Chromatin loop, Hi-C, Density clustering, Attention mechanism

Background

Chromatin is not linearly arranged in the nucleus but exists in the nucleus in a multiple folded and entangled state in space, thus forming complex 3D structure of the genome. The folded state of the genome in 3D space

*Correspondence:

² College of Chemical and Environmental Engineering, Anyang Institute

of Technology, Anyang 455000, China

leads to the formation of interactions between local or distal regions of chromatin, which in turn regulates the gene transcription process, so the folded structure of the genome directly affects the transcription and expression of genes, and is closely related to the regulation of cellular functions [1].

Over the past decades, a variety of approaches have emerged to study the intrinsic complexity of 3D genomes, among which, many of them originated from the concept of chromosome conformation capture technology (3 C) [2], and with the development of this technology, researchers have revealed the multilevel structure of 3D genome organization, such as A/B compartments, topological association structure thresholds (TADs) and chromatin loops [1].



© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

Junwei Luo

luojunwei@hpu.edu.cn

¹ School of Software, Henan Polytechnic University, Jiaozuo 454003, China

³ School of Computer and Information Engineering, Henan University, Kaifeng 475001, China

⁴ School of Computer Science and Engineering, Central South University, Changsha 410083, China

As the fundamental component of the 3D genome structure, chromatin loops have been interpreted as structures formed by pairs of chromatin anchors that are far apart in linear distance from each other in the genome but are spatially close to each other. Chromatin loops are formed through complex molecular interactions and structural regulation, and studies have shown that the structure of chromatin loops is closely associated with a variety of binding factors that include, but are not limited to, histone modifying enzymes, transcription factors, and chromosomal regulatory proteins [3, 4]. The chromatin loop structure can maintain a high degree of chromatin organization and functionality, thereby affecting gene expression and chromosome function.

With the rapid development of next-generation sequencing technologies, a series of derived technologies have emerged for in-depth study of the structure and function of chromatin loops. High-throughput chromosome conformation capture (Hi-C) [5, 6] is a derivative of chromosome conformation capture technology, which obtains information on pairwise interacting segments of chromatin by biotechnologically immobilizing chromatin segments that produce contacts, and it has been widely applied to a variety of biological informatics fields. Combining high-throughput sequencing technology and bioinformatics analysis methods to study the spatial information of chromatin loops on a genome-wide scale can help to deeply understand gene expression networks and provide important theoretical support for life science research.

Currently, several research methods on chromatin loops have been built on the basis of Hi-C technology [5, 7, 8]. They can be broadly categorized into two groups, the first utilizing signal enrichment statistics and the second based on machine learning and deep learning.

There are several approaches that utilize signal enrichment statistics, for example, Fit-Hi-C [9] performs statistical confidence estimation for mid-range intrachromosomal contacts by combining the polymer loops effect and the bias previously observed in the Hi-C dataset, and it identifies significant interactions between chromatin segments. HiCCUPS [1] incorporates localized background into its framework and applies the Poisson test and a modified Benjamini-Hochberg to determine significance, identifying enriched regions as chromatin loops. SIP [10] achieves the prediction of chromatin Loops by identifying strongly saturated points in the Hi-C matrix, and SIP adopted a region-maximization detection algorithm to filter false candidate chromatin loops. Chromosight [11], inspired by computer vision, used a balanced normalization procedure to attenuate the experimental bias and extracted the relevant focal peaks to determine the location of chromatin loops. MUSTACHE [12] uses a scale-space theory of the contact matrix to report locally enriched pixels as chromatin loops, with good results in a prediction task. With the wide application of machine learning and deep learning technology, a series of new chromatin loop prediction methods have emerged in society. Notably, Peakachu [13], as a supervised learning method, predicts chromatin loops in the genome-wide contact matrix by constructing a random forest framework. DeepLUCIA [14], based on deep learning, achieves the prediction of chromatin loops in 3D genomes by learning genomic sequence features and epigenomic information features of Hi-C paired-end reads. The DeepLoop [15] method consists of two parts: the LoopDenoise model for image noise reduction and the LoopEnhance model for signal enhancement, and it implements chromatin interaction mapping from lowdepth Hi-C data. The two-branch network of GILoop [16] extracts pixel-level features and edge-informative features from two different view representations of images and graphs, respectively, and it identifies genomewide chromatin loops. Another new deep learning-based framework, DLoopCaller [17], predicts chromatin loops by integrating raw Hi-C contact matrix data and accessible chromatin landscape data.

Although these computational methods have made great progress, they still have some drawbacks, such as inadequate feature extraction and high false positives. All these limitations have stimulated the development of computational analysis. Considering the power of deep learning in capturing the features of complex data, there is still a great potential for using deep learning methods to adequately capture the features of chromatin loops, and improve the prediction of chromatin loops.

Here, we propose CGLoop, a deep learning method based on convolutional neural networks (CNN) [18, 19] and the cyclic neural network variant, bidirectional gated recurrent units (BiGRU) [20–22], for predicting chromatin loops based on Hi-C contact matrix. CGLoop uses the convolutional neural network with Convolutional Block Attention Module (CBAM) [22-24] to extract the local features of the matrix and then combines BiGRU to obtain the sequential feature variant among adjacent regions. Then, CGLoop obtains the candidate chromatin loops with prediction scores and clustering to filter the false candidate chromatin loops based on the density. In the experiment part, we conducted a series of validation analyses for chromatin loops prediction, such as APA analysis [1, 25, 26], transcription factor binding analysis [27-29], and binding protein enrichment analysis [6, 26, 27, 30, 31], etc., and the results demonstrated that CGLoop has a good performance compared with other methods.

Methods

CGLoop is a deep learning method capable of identifying chromatin loops in 3D genomes. CGLoop takes Hi-C contact matrix as input, and regards the prediction of chromatin loops as a binary classification problem. The prediction of chromatin loops by CGLoop is mainly divided into five steps: (i) Generating sub-matrices. Chromatin loops correspond to the elements of the Hi-C contact matrix. And centered on the elements, the Hi-C contact matrix is cut to Generate submatrices of size 21 \times 21. (ii) Extracting the local features. CGLoop uses the convolutional neural network and CBAM (CNN-CBAM) to capture the local features of each submatrix. (iii) Extracting sequential features among adjacent regions. CGLoop uses BiGRU to obtain the sequential features contained in adjacent regions inside each submatrix. (iv) Prediction. By extracting the features of the submatrix, CGLoop estimates and outputs the probability that the chromatin interaction fragments corresponding to the center of the submatrix form a chromatin loop. (v) Clustering. The candidate chromatin loops obtained in the previous steps are clustered based on density to obtain the final chromatin loop predictions. The workflow of CGLoop is shown in Fig. 1.

Generating submatrices

CGLoop uses the Hi-C contact matrix M as input, and the Hi-C data sources are provided in Table S1 of Supplementary file 1. The results of all methods in this manuscript were performed at 5 kb resolution and are based on the same Hi-C data, Cool data obtained by the Hicexplorer method [32]. For a chromosome, we split it into subregions with the same length (resolution, 5 kb in default), and each sub-region refers to a bin. In the Hi-C contact matrix, *M*[*i*, *j*] represents the contact frequency between *i-th* bin and *j-th* bin. B_{ii} represents the bin-pair, which is composed by the *i*-th bin and *j*-th bin, and it also corresponds to the coordinates of the Hi-C contact matrix, as shown in Fig. 2. On the linear chromosome, B_{ii} corresponds to the coordinates of the two fragments (i-th bin and j-th bin) that generate interaction. In the Hi-C contact matrix, B_{ii} corresponds to the coordinate of the center of the submatrix in the Hi-C contact matrix. That is, B_{ii} represents a coordinate, so B_{ii} can also be described as [i, j]. Based on M, we predict whether each bin-pair is the two fragments that form the anchors of the chromatin loop. In order to remove the systematic bias in the Hi-C data, CGLoop first uses KR [33] normalization to process the original Hi-C contact matrix.

We select the center of the submatrix from the upper right corner area of the Hi-C contact matrix with a step length of 1. In addition, it is generally believed that there are usually strong interactions at the chromatin loop anchors [1]. However, if the bin-pair has a small distance in the linear chromosome, they also typically have strong interactions. Meanwhile, if the bin-pair has large distance in the linear chromosome, the probability that they form a loop is low [13, 34, 35]. We experimented with different threshold settings, see Table S5 of Supplementary file 1, here we set the default threshold between 30 Kb- 3 Mb. So, if the B_{ij} is used to generate the submatrix, it should satisfy the following conditions:

$$B_{ij} = [i, j], where M[i, j] > 1 \text{ and } \frac{lower}{res} \le i - j \le \frac{upper}{res}$$
(1)

the M[i,j] denotes the contact frequency at [i,j] in the Hi-C contact matrix, and *res* denotes the resolution. *lower* represents the minimum distance between anchors(default is 30,000, 3 Kb), *upper* represents the maximum distance between anchors(default is 3,000,000, 3 Mb).

Next, we use B_{ij} as the center of the submatrix, and we define the submatrix with MS_{ij}. Thus, MS_{ij} is the submatrix that satisfies the constraint condition (formula 1). Here, we define d as half the number of rows (or columns) of the submatrix. We construct the MS_{ii} to represent the characteristics of surrounding contact and $MS_{ii} = M[i-d:i+d+1, j-d:j+d+1]$, which means elements from (i-d)-th row to (i+d)-th row, and (j-d)-th column to (j+d)-th column in M, and it also means that the number of rows (columns) of the submatrix is 2 d + 1. We refer to the neutron matrix size settings of different methods, and try different d value settings, where d= 10 in the model training to obtain a more stable effect. Under different resolutions, we set different submatrix sizes, and the evaluation results of the corresponding models are shown in Table S3 of Supplementary file 1. We comprehensively considered resource consumption and training efficiency, and finally, we set d = 10 to generate a submatrix with 21 rows and 21 columns for the analysis of this method.

Feature extraction

Deep learning models have powerful feature capture and fusion capabilities and can independently extract deep features from complex data [36–38], which is why deep learning is widely used in different fields at present. Convolutional neural networks (CNN) have the advantages of local awareness and position invariance in capturing matrix data [22, 39–41], so it becomes the choice of CGLoop in processing matrix data. The attention mechanism can assign different attention weights according to different representations of the input data, enabling the model to selectively focus on information that is more



Fig. 1 The workflow of CGLoop. **a** Generating submatrices. **b** Extracting the local features. F is the feature matrix of the input of CBAM, F" is the feature matrix of the output of CBAM, and M_{CM} is the feature matrix of the output of CNN-CBAM. **c** Extracting sequential features among adjacent regions. M_{BG} is the feature matrix of BiGRU's output. **d** model prediction. **e** Clustering

important to the current task [22]. BiGRU can capture the timing features of the input data in both forward and backward directions [21, 42].

Here, CGLoop chooses to use the CNN, CBAM, and BiGRU as the main architecture. CGLoop builds the model based on the architecture LSnet [43] and makes several improvements on this basis. Instead of using multi-layer standard convolution layers (Conv2D) in the original model, CGLoop reduces the number of convolution layers and incorporates deep separable convolution (SeparableConv2D) [44] to reduce the amount of computation and number of parameters while maintaining a high feature extraction capability. It has been found that the convolutional pattern and the way of combining different modules affect the performance of the model, especially for our research content: when the number



Fig. 2 The correspondence of B_{ij} on Hi-C contact matrix and linear chromosome. The blue line represents the chromosome, and the yellow line represents the fragments of the chromosome. The red box represents the submatrix (whose size is $(2 d + 1) \times (2 d + 1)$), and the coordinates of the green square in the Hi-C contact matrix correspond to B_{ii}

of convolutional layers is less, the model performance is rather better. Combined with our task requirements, CGLoop adjusts the number of different convolutional layers and the embedding position of each module, and it introduces the CBAM module after one layer of convolution, which effectively concentrates the more important feature regions. In addition, CGLoop also adjusts the structure of the BiGRU layer, which reduces the number of neurons in the fully connected layer. In conclusion, CGLoop takes the hybrid convolution strategy and the efficient embedding between different modules as the core, breaks the dependence of the traditional model on high parameters and high complexity, and realizes the optimal balance between computational efficiency and feature extraction performance.

Specifically, CGLoop builds a CNN-CBAM layer composed of two layers of CNN and one layer of CBAM and connects the CNN-CBAM layer in series with the BiGRU layer. During each MS_{ij} is processed by the convolutional layers, CBAM is used to focus on salient features. After processing by the CNN-CBAM layer, a feature matrix is output. Then, the feature matrix after flattening, is input into the BiGRU layer for learning the sequential features between elements of the feature matrix. Finally, the prediction results are obtained through the fully connected layers. With these improvements, our model shows high accuracy and robustness while reducing computational complexity, especially when dealing with tasks with smaller input features matrix, showing higher efficiency and performance.

Extracting the local features

CGLoop uses the CNN-CBAM layer to capture the local features. For each MS_{ij} , CGLoop first captures the local information using the convolution operation and then adopts the MaxPooling layer to reduce the spatial dimension of the feature. The convolution operation satisfies Eqs. (2, 3):

$$V = conv2(W, X) + bias$$
⁽²⁾

$$Y = \varphi(V) \tag{3}$$

where *W* is the convolution kernel matrix, *X* is the input matrix, *Y* is the output matrix, *bias* is the bias term, and

 ϕ (*V*) is the activation function, where the convolution output *V* is elu activated.

In CGLoop, in order to further reduce the computational complexity and improve the efficiency of feature extraction, we introduce SeparableConv2D. Separable Convolution decomposes the standard convolution into two steps: Depthwise Convolution and Pointwise Convolution. First, Depthwise Convolution performs convolution operation on each input channel separately to capture the spatial features within the channel; then, Pointwise Convolution integrates the information from each channel through a 1×1 convolution kernel to realize cross-channel feature fusion. This structure effectively reduces the number of parameters and computations while retaining the ability of the convolutional layer to express features.

CBAM can help the model to focus on important features, and CGLoop transports the convolved and pooled feature matrix F into the CBAM layer to further extract features from the two dimensions of the channel attention module and spatial attention module. The specific calculation formula is shown in Eqs. (4-7) [43]:

$$Mc(F) = Channel(F) \tag{4}$$

$$Ms(F) = Spatial(F) \tag{5}$$

$$F' = Mc(F) \otimes F \tag{6}$$

$$F'' = Ms(F') \otimes F' \tag{7}$$

Where *F* is the feature matrix after convolution and pooling, *F*" denotes the output matrix processed by CBAM, and Mc(F) and Ms(F) denote the outputs of the channel attention module and spatial attention module, respectively. \otimes denotes the elemental multiplication.

Finally, the feature matrix F'' after the CBAM layer is convolved and pooled again to obtain the CNN-CBAM processed matrix M_{CM} . The feature matrix M_{CM} is flattened and input to the subsequent feature extraction module.

Extracting sequential features among adjacent regions

Neighboring regions at the centroid of the submatrix MS_{ij} tend to show higher contact intensity. In addition, contact features are also evident in the lower left background region of the submatrix [1, 26, 45]. Therefore, CGLoop employs BiGRU to dissect the sequence relationships among the neighbors inside the submatrix. Here, each feature matrix (M_{CM}) is flattened into a sequence of feature vectors fed into the BiGRU layer to extract the sequential features of the internal neighborhood of each sample. The input matrix is processed not

only by the forward GRU, but also by the backward GRU. Specifically, the updated formula for the forward GRU satisfies (8), and the updated formula for GRU satisfies (9) [42, 46]:

$$\overrightarrow{h_t} = GRU(x_t, \overrightarrow{h_{t-1}}) \tag{8}$$

$$\overleftarrow{h_t} = GRU(x_t, \overleftarrow{h_{t-1}}) \tag{9}$$

Where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ denote the left-to-right and rightto-left hidden states, respectively, GRU denotes the GRU unit, and x_t denotes the *t*-th element in the input sequence.

Finally, the hidden states of the forward and backward GRUs are spliced together to obtain the final hidden state h_t :

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right] \tag{10}$$

where [;] denotes the vector splicing operation. Finally, the feature matrix processed by the two-layer BiGRU is passed into the model prediction module.

In each GRU cell, updating the hidden state depends on the operation of the input door and the reset door. The update gate z_t controls how the current hidden state h_t interpolates between the previous hidden state h_{t-1} and the candidate hidden state \tilde{h}_t controls the degree to which the previous hidden state h_{t-1} is reset when the candidate hidden state \tilde{h}_t is calculated [47]. The formula is as follows:

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t] + b_z) \tag{11}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{12}$$

$$\widetilde{h_t} = tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h)$$
(13)

$$h_t = (1 - z_t) \cdot \widetilde{h_t} + z_t \cdot h_{t-1} \tag{14}$$

Where h_{t-1} represents the hidden state of the previous step, x_t represents the current input, W_z , W_r , W_h correspond to the weight matrixs of the update gate z_t , the reset gate r_t and the candidate hidden state \tilde{h}_t respectively, and b_z , b_r and b_h are the bias of them. σ is the Sigmoid function and *tanh* is the hyperbolic tangent function.

Prediction

The fully connected layer takes the feature matrix output from the previous layer as input and makes the final prediction. Using the dropout layer in the process to regularize the network. Finally, after sigmoid activation, the model outputs a predicted score in the interval [0, 1] that identifies the probability that the sample is a chromatin loop.

Clustering

The analysis reveals that the features of the center of the matrix and its surrounding pixels possess a high degree of similarity. In the Hi-C contact matrix, multiple binpairs belonging to a loop may be reported as multiple loops. Therefore, we need to filter multiple prediction loops belonging to the same loop and output a representative loop. Considering that the density-based clustering algorithm can cluster the clustered sample points into a class [1, 6, 32, 48–50], the clustering method in Peakachu [13] was used here to obtain the optimized chromatin loop positions. The clustering threshold parameters of CGLoop in different cell lines are shown in Table S2 of Supplementary file 1.

Model training and evaluation

Construction of positive and negative samples

We use ChIA-PET(CTCF) and HiChIP(H3 K27ac) data to label positive and negative samples, and these data sources are provided in Table S1 of Supplementary file 1.

We preprocessed the data of corresponding positional columns in CTCF ChIA-PET and H3 K27ac HiChIP separately. First, it was mapped to 5 kb resolution, and we removed the data rows that were nested within each other in the two datasets and finally merged them to obtain a dataset integrating two enrichment factors, CTCF and H3 K27ac. Then, the interactions covering multiple bins are split into bin-bin interactions to get the combined data. We can obtain all bin-pairs from the Hi-C contact matrix. These bin-pairs are divided into two sets, one set includes all positive bin-pairs, other set includes all non-positive bin-pairs.

We named the positive sample submatrix as *MSP* and the positive sample coordinates as *BP* (*n* in total). We generated *MSP*, which is centered at *BP*, if *BP* can be found in the combined data above. BP_{ij} implies the positive sample at B_{ij} of the Hi-C contact matrix, and BN_{ij} implies a negative sample at B_{ij} of the Hi-C contact matrix. The bin-pair distance of BP_{ij} is |j-i|.

For the acquisition of negative samples, We referred to the method proposed by Shen et al. [35]. Similar to the positive sample, We named the negative sample submatrix as MSN, and named the negative sample coordinates as BN. BN was obtained in three ways: (1) Randomly select $2 \times n$ bin-pairs (BS) from non-positive bin-pairs, which have the same bin-pair distance with each sub-set of BP;(2) Randomly select $1 \times n$ bin-pairs (BL) from nonpositive bin-pairs, which have the larger bin-pair distance than the maximum bin-pair distance of BP; (3) Select $1 \times n$ bin-pairs (*BR*) from non-positive bin-pairs, which have the random bin-pair distance. Therefore, BN satisfies Eq. (15).

$$BN = 2BS + BL + BR \tag{15}$$

Each chromosome was de-weighted and cleaned according to the above samples sampling requirements to obtain positive and negative samples with *BP:BN* roughly 1:3.

Construction of training and validation samples

CGLoop used chromosomes 1–19 on the GM12878 cell line as training and validation samples for the model. Here, the positive and negative samples from chromosomes 1–19 are randomly divided into five equal parts, respectively. The training set is obtained by taking four parts of the positive and negative samples and merging them respectively. The remaining samples are used as the validation set, i.e., the training set and validation set satisfy 4:1. After the above processing, 298,060 training samples and 74,516 validation samples were obtained, respectively. Our model training process was run on an NVIDIA GeForce RTX 4090, in addition to a detailed analysis of resource consumption. The results are shown in Table S3 and S4 of Supplementary file 1.

Loss function

Since CGLoop treats the prediction of chromatin loops as a binary classification task, the model is trained here using binary cross-entropy loss (*BCELoss*), defined as follows:

$$BCELoss = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - (p(y_i)))$$
(16)

where *N* denotes the number of samples, y_i denotes the true label of the *i*-th sample, and $p(y_i)$ is the predicted probability of the *i*-th sample.

Furthermore, *BCELoss* uses the Adam optimization algorithm [51, 52]. The model uses ReduceLROnPlateau to adjust the learning rate, which in turn improves its performance [53, 54]. CGLoop saved the best-performing model parameters and applied them to subsequent tests.

Results

To confirm the validity of the CGLoop method, we first evaluated its performance using a selection of extracted test samples. Then, on the full samples of multiple chromosomes, CGLoop was compared with the other methods for chromatin loops prediction. These methods included Mustache, Chromosight, as well as Peakachu and DLoopCaller. CGLoop was evaluated with these methods on several cell lines (GM12878, K562, IMR90, and mESC) by Aggregation Peak Analysis (APA), Binding Factor Enrichment Analysis, Promoter and Enhancer Binding Analysis, Loops Overlap Analysis, Loops Distance Analysis, and other evaluative analyses.

Model test

CGLoop randomly selected 22,769 samples in the sample set of chromosomes 20, 21, and 22 of the GM12878 cell line, of which 7,032 were positive samples, and 15,737 were negative samples. The best performing model parameters were loaded, and those samples were fed into the model for testing to obtain 22,769 predicted scores located between 0 and 1. The predicted scores, categorical labels, frequency of contact at the center of the matrix, and location information are saved as the resulting output of the CGLoop model.

Accuracy, precision, recall, f1-score, and PRAUC were used as assessment metrics for model testing. The results show that on the randomly selected part of the test set, the PRAUC of our method reaches 0.934, the Accuracy reaches 0.911, and the precision, recall, and f1-score are all above 0.855, which shows that the CGLoop method achieves a more accurate prediction performance on the randomly selected dataset.

Candidate loops prediction

The ultimate goal of CGLoop is still to realize the prediction of chromatin loops on the whole chromosome. Here, we generated whole chromosome prediction samples by using all B_{ij} on human chromosomes 20, 21, and 22 (mouse 17, 18, and 19) as the centroid of MS_{ij} . The samples were fed into the already trained model and the results with predicted scores were produced.

The analysis revealed that the samples with three chromosomes on GM12878 had higher prediction scores than the other cell lines, and we speculate that non-similarity between cell lines contributed to this difference. Even so, regardless of the cell line, samples with high scores on a single chromosome showed a strong chromatin loop signal. Therefore, CGLoop selected samples with relatively high prediction scores as candidate chromatin loops to be input into the subsequent clustering process. The predicted number of chromatin loops on different cell lines is shown in Table 1.

Aggregation peak analysis

Aggregation Peak Analysis (APA) is used to identify and quantify aggregation peaks in chromatin. "Peaks" are usually indicative of regions of high signal intensity on the genome, representing a concentration of certain gene regulatory elements [1, 55]. The spatial aggregation of chromatin can be explored by APA analysis.

| Table 1 | The predicted number of chromatin loops on | different |
|------------|--|-----------|
| cell lines | | |

| Chrname | GM12878 | K562 | IMR90 | mESC | | |
|---------|---------|------|-------|------|--|--|
| Chr20 | 2301 | 323 | 530 | - | | |
| Chr21 | 852 | 129 | 199 | - | | |
| Chr22 | 1668 | 348 | 339 | - | | |
| Chr17 | - | - | - | 447 | | |
| Chr18 | - | - | - | 370 | | |
| Chr19 | - | - | - | 308 | | |
| Sum | 4821 | 800 | 1068 | 1125 | | |

APA_score reflects the contrast between the signal in the center region and the background signal, here, it represents the ratio of the contact frequency of the center element of a particular size matrix to the average contact frequency of the lower left background matrix. In order to calculate the APA_score, we refer to the methodology proposed by Rao et al. in their study [1]. Specifically, we chosed the average matrix size of 11 ×11, and the lower-left background matrix is defined as 3×3 region of the average matrix. And the APA_Score satisfies Eq. (17). We use APA_score to quantify the extent to which loops identified by CGLoop are supported by Hi-C contact frequency signals. The results of the APA analysis of the different methods on the GM12878 test set are shown in Fig. 3.

$$APA_Score = \frac{avg[w,w]}{\frac{1}{cw \times cw} \sum_{i=1}^{cw} \sum_{j=1}^{cw} lowerpart[i,j]}$$
(17)

where *avg* is the mean matrix(size of 11), which corresponds to chromatin loops, *avg[w, w]* is the contact frequency at the center of the *avg*, and *lowerpart* is the lower-left matrix of the *avg* (size of *cw*, and *cw* = 3).

The chromatin loop prediction results of CGLoop, Peakachu, DLoopCaller, Mustache, and Chromosight were sorted according to the prediction scores in ascending order and analyzed by APA, and the APA scores obtained for limiting the number of chromatin loops are shown in Fig. 2. The figure shows that the loops predicted by CGLoop presented higher APA scores compared to other methods at different sampling rates, and the APA scores gradually decreased as the number of chromatin loops increased. Among chromatin loops up to 5000, the APA scores of the loops predicted by CGLoop were not lower than 1.47. Visualizing the APA maps of all loops predicted by each method on the three chromosomes, the results are shown in Fig. 3B-F, which shows the features of matrix centroid and lower left background enrichment, which



Fig. 3 APA analysis of different methods on GM12878 cell. A APA scores of different methods under the condition of limiting the number of chromatin loops; B, C, D, E, F APA visualization maps of all chromatin loops predicted by Peakachu, DLoopCaller, Chromosight, Mustache, CGLoop on three chromosomes

is consistent with what we previously learned about the features of chromatin loops.

Enrichment analysis

Enrichment analysis of structural proteins

CTCF (CCCTC-binding factor), as a transcription factor, is able to bind to specific regions on chromatin and generate binding sites. These binding sites can form physical contacts with distal regulatory elements (e.g., enhancers) [30, 56, 57], allowing DNA fragments from different regions to come in close proximity to each other, ultimately forming chromatin loop structures. H3 K27ac is a histone modification mark that is often found in active regions of regulatory elements (e.g., enhancers and promoters) [58, 59], and in chromatin loops, the presence of H3 K27ac can indicate the active state of certain regions of chromatin.RAD21 and SMC1 are core components of the structural cell complex (cohesin complex), and they are involved in the construction of DNA helical structures [60, 61]. They promote the formation and stabilization of chromatin loops by aggregating different DNA fragments.



Fig. 4 Enrichment factor analysis of chr20, chr21, and chr22 chromatin loops on GM12878 by different methods. A Enrichment of CTCF transcription factors in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; B Enrichment of H3 K27ac-binding proteins in three chromatin loops of chr20, chr21, and chr22 by different methods; C enrichment of RAD21-binding protein in three chromatin loops of chr20, chr21, and chr22 by different methods; C enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different methods; D enrichment of SMC1-binding protein in three chromatin loops of chr20, chr21, and chr22 on GM12878 by different met

Therefore, the number of bindings of enriched factors, such as CTCF, H3 K27ac, RAD21, and SMC1, on the predicted results reflects the quality of the predicted chromatin loops. The reliability of the loop prediction method can be assessed by statistically analyzing the number of these binding events.

We downloaded multiple target datasets of binding factors from publicly available websites, including CTCF ChIA-PET, H3 K27ac HiCHIP, RAD21 ChIA-PET, and SMC1 HiCHIP. By calculating the matching number of prediction loops and target factors, the enrichment statistics are realized [34]. Accumulate the number of matches for each predicted loop with the target factors to obtain the number of matches between the prediction loops and the target factors [62, 63].

Here, we ranked the chromatin loops predicted by the different methods in order of prediction scores and visualized the enrichment of the top 2000 predicted loops separately. As shown in Fig. 4A, CGLoop binds more CTCF transcription factors at 5 kb resolution. As the number of predicted loops increased, the number of binding factors gradually increased, and the enrichment growth rate gradually slowed down. In addition, as shown in Fig. 4(B-E), CGLoop's prediction loops showed obvious enrichment effects of H3 K27ac, RAD21, and SMC1 binding factors.

Enrichment analysis of promoters and enhancers

Enhancers can enhance the transcriptional activity of nearby genes, and promoters are the starting points of transcription. The formation of chromatin loops requires the interaction between promoters and enhancers [64]. Here, we used the enhancer and promoter location information extracted from ChromHMM annotation [54] to verify the accuracy of chromatin loops predicted by CGLoop.

We analyzed the proportion of regulatory elements on chromatin loops. As can be seen from Fig. 5, most loops identified by CGLoop are mediated by enhancers, and about 30% of loops have no regulatory elements, which is similar to other methods. This is consistent with the proportion of chromatin loop regulatory elements reported by Rao et al. [1]in the GM12878 cell line. However, N–N accounts for the largest proportion of the loops identified by DLoopCaller. These results suggest that CGLoop is able to predict enhancer regulated chromatin loops with high sensitivity.

Quantitative analysis of overlapping loops Quantitative analysis of absolute overlap

We defined loops predicted by two methods are considered"absolutely overlap"if they are located in the same bin. We visualized the absolute overlap of loops



Fig. 5 Statistical analysis of the number of promoter and enhancer binding on the GM12878 dataset



Fig. 6 The number of overlapping chromatin loops predicted by different methods under absolute overlap conditions on chromosome chr20,21,22 of GM12878

predicted by different methods. As shown in Fig. 6, the number of loops predicted by the different methods varies significantly, which we attribute to the presence of prediction bias at high resolution. The results show that 724 of the chromatin loops predicted by CGLoop have the absolute overlap with other methods.

Quantitative analysis of mismatch overlap

"mismatch overlap" is defined as the difference between the left (and right) anchor positions of the two loops being no greater than 5 kb. Chromatin loops with higher prediction scores within 5000 were selected, and the mismatch overlap of the loops predicted by different methods were compared. As shown in Fig. 7, the overlap rate of the chromatin loops identified by CGLoop with the standard set is about 33%(The data labeled as'Replicloops'in the figure were obtained from Rao et al. (2014) [1], and we defined this dataset as'Replicloops'in our study). As the number of chromatin loops increased, the overlap rate gradually decreased, which also confirmed that the higher the prediction score, the more likely the predicted loop is true. Here, CGLoop still shows excellent predictive performance compared to other methods.

Analysis of Recovery Efficiency Metric (REM)

Recovery Efficiency Metric (REM) analysis is primarily utilized to assess the biological consistency and detection performance of loop prediction methods. REM integrates recovery rate with the number of predicted loops. Normalizing the recovery rate mitigates biases arising from varying numbers of loops predicted by different methods, thereby facilitating a fair comparison of their performance [65]. Specifically, recovery analysis quantifies the method's ability to identify specific biomarkers (e.g., CTCF, H3 K27ac, Rad21) by calculating the overlap ratio between predicted loops and reference data. The implementation of REM prevents methods from overstating their detection capabilities due to excessive loop predictions, enhancing the scientific rigor and reliability of the analysis results. On chr20,21,22 of GM12878, we comparatively analyzed the REM of different methods for CTCF, H3 K27ac, and Rad21 targets. As shown in Fig. 8,



Fig. 7 The overlap between different chromatin loop datasets under the condition of allowing 5 kb mismatch. A The overlap rate between different chromatin loop datasets. B The number of chromat in loops overlapping with different methods. In order: CGLoop and Peakachu, CGLoop and Chromosight, CGLoop and Mustache, CGLoop and DLoopCaller, CGLoop and Positive, Peakachu and Positive, Chromosigh and Positive, Mustache and Positive, DLoopCaller and Positive, respectively



Fig. 8 Visualization of REM(Recovery efficiency rate) for chromatin loops predicted by diffirent methods in the GM12878 cell line(chr20,21,22) under the target factors CTCF, Rad21, and H3 K27ac, respectively



Fig. 9 CTCF peaks in the neighborhood of the chromatin loop anchor identified by CGLoop on the GM12878 cell line

the overlap ratio of CGLoop is relatively low, which may be due to the relatively large number of predicted loops.

Anchor peak analysis

The peak height in CTCF ChIP-seq experiments usually reflects the CTCF binding strength at that genomic location, and sites with higher CTCF binding strength are more likely to be involved in the formation of chromatin loops [66]. Therefore, CGLoop analyzed CTCF binding peaks at chromatin loop anchors and their flanking regions. As shown in Fig. 9, the loops predicted by CGLoop show a trend of peak at the anchor points and slowing down around it. Compared with other methods, under the condition that the number of prediction loops is roughly the same, the peak performance in CGLoop is the most obvious, showing the highest peak.

Distance distribution analysis

In order to explore the distribution of chromatin loops(chr20,21,22) predicted by CGLoop and other methods on GM12878, the data were statistically analyzed according to the distribution of anchor distances of'[0, 250],"(250, 500],"(500, 1000],"(1000, -]'(in kb). As shown in Fig. 10, CGLoop has the similar distance distribution to the chromatin loops predicted by peakachu and mustache, with short-range loops ([0, 250]) accounting for the largest proportion. The analysis found that most (about 55%) of the chromatin loops predicted by CGLoop ranged from 0 to 250 kb, belonging to short-range loops, and 13.7% of the loops belonged to long-spaced loops (500 kb to 1000 kb). Notably, the loops predicted by Chromosight are all short-range loops, and DLoopCaller predicts more long-range loops. Distance distribution analysis of chromatin loops predicted by different methods on other cell lines is shown in Supplementary Fig S4 of Supplementary file 1.

Chromatin loops on Hi-C contact heatmap

Chromatin loops predicted by different methods on GM12878 are mapped onto the Hi-C contact heat map. Each coordinate in the hic heat map corresponds to the location of a pair of chromatin interaction fragments. As shown in the Fig. 11: CGLoop compared positive, CGLoop compared Peakachu, CGLoop compared Mustache, CGLoop compared Chromosight, and CGLoop compared DLoopCaller. We mapped the positive sample set, and chromatin loops predicted by CGLoop, Peakachu, Mustache, Chromosight, and DLoopCaller to the hic heat map and compared them. The results show a high agreement between the chromatin loops predicted by CGLoop and the other datasets.

Experimental analyses across cell lines and species

In order to validate that our method is not limited to a single cell line or species, we preprocessed the Hi-C data obtained for a human leukemia cell line (K562), a normal human embryonic lung fibroblast cell line (IMR90), and a mouse embryonic stem cell line (mESC) following the same process as previously described. We selected the previously trained model, predicted and clustered the preprocessed samples, and finally conducted the subsequent validation analysis. The results of peaks analysis



Fig. 10 The distance distribution between the left and right anchors of the chromatin loops predicted by different methods on chr20,21,22 of GM12878. a denotes the minimum value of the distance, b denotes the maximum value of the distance, and [a, b] means the absolute difference between the left and right anchor positions of the chromatin loop is within a to b



Fig. 11 Hi-C contact heat maps of chromatin loops predicted by different methods on GM12878. The dots above each heat map represent the other chromatin loops data (blue), and the dots below represent the chromatin loops predicted by CGLoop (black). In order: chromatin loops predicted by CGLoop (black) vs Positive samples set (blue); chromatin loops predicted by CGLoop (black) vs chromatin loops pre



Fig. 12 Analysis on other cell lines. Analysis of CTCF peak aggregation in chromatin loop neighborhoods (A-C) and CTCF binding at chromatin loop anchors (D, E) on different cell lines

and transcription factor analysis on other cell lines are shown in Fig. 12.

The experimental results showed that the chromatin loops predicted by CGLoop got a favorable performance on several cell lines, and all of them showed significant enrichment of binding factors such as transcription factors and binding proteins. Additional validation analysis results for different cell lines are shown in Supplementary Fig. 1-4 of Supplementary file 1. In conclusion, our method can still identify loops relatively accurately for data on other cell lines.

Discussion and conclusion

Chromatin loop prediction using neural networks can facilitate the development of research related to 3D genome. Most classical methods for predicting chromatin loops suffer from inaccurate loop identification, and the development of deep learning has inspired the emergence of a new generation of chromatin loop prediction methods. In this study, we developed a new method for predicting chromatin loops based on neural networks, CGLoop, which utilizes convolutional neural networks and recurrent neural networks to capture deep features from Hi-C interaction frequency data to achieve the prediction of chromatin loops.

We learned that CTCF ChIA-PET data in Peakachu contains more long-range loops, while H3 K27ac HiChIP

has more short-range loops [13], so in this study, CTCF ChIA-PET data and H3 K27ac HiChIP data are used to generate positive and negative samples. In CGLoop, A two-layer convolutional neural network (CNN layer) with a nested attention mechanism (CBAM layer) was used to extract local features from the samples, and the recurrent neural network (BiGRU layer) was used to capture sequential features. This model combined spatial and sequential information to help mine the data information more deeply. Finally, the final chromatin loop prediction results are obtained from the candidate loops by a density-based clustering algorithm, improving chromatin loop predictions'accuracy and portability.

To verify the validity of the method, we performed some evaluation experiments such as APA analysis, binding factor enrichment analysis, loops overlap analysis, and loops distance analysis, and applied the CGLoop method to other cell lines. The results of a series of experiments show that, our method possesses good robustness and can locate the anchor positions of chromatin loops with high resolution, whether in different species, different cell lines of the same species, or on different chromosomes.

Although CGLoop achieves good performance, there are still areas that need to be optimized and improved: (1) When generating test samples, since we predict all the samples of the whole chromosome at 5 kb resolution, the

data volume is huge, so it is very time-consuming to generate the small matrix samples, and the data preprocessing algorithm can be optimized to improve the efficiency of data generation. (2) Currently, we analyze the chromatin loop information of pairwise contact, in fact, there are many three or more chromatin loop anchor contacts in the 3D space, so the prediction method can be adjusted appropriately to adapt to the higher-order chromatin loops prediction work.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12864-025-11531-y.

Supplementary Material 1.

Authors' contributions

JFW, LLW and JWL participated in the analysis of the experimental results. JFW, LLW performed the implementation, prepared the tables and figures, and summarized the results of the study. JJW, HML, FG, and CKY checked the format of the manuscript. All authors have read and approved the final manuscript for publication.

Funding

This research was supported by the Innovative Research Team of Henan Polytechnic University (Grant No. T2021 - 3).

Data availability

All data used in this paer are shown in the Supplementary file 1.

Declarations

Ethics approval and consent to participate Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 25 August 2024 Accepted: 25 March 2025 Published online: 05 April 2025

References

- Rao SS, Huntley MH, Durand NC, Stamenova EK, Bochkov ID, Robinson JT, Sanborn AL, Machol I, Omer AD, Lander ES. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell. 2014;159(7):1665–80.
- Dekker J, Rippe K. Dekker M. Kleckner N: Capturing chromosome conformation science. 2002;295(5558):1306–11.
- Fudenberg G, Imakaev M, Lu C, Goloborodko A, Abdennur N, Mirny LA. Formation of chromosomal domains by loop extrusion. Cell Rep. 2016;15(9):2038–49.
- Nuebler J, Fudenberg G, Imakaev M, Abdennur N, Mirny L. Chromatin organization by an interplay of loop extrusion and compartmental segregation. Biophys J. 2018;114(3):30a.
- Lieberman-Aiden E, Van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie BR, Sabo PJ, Dorschner MO. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science. 2009;326(5950):289–93.

- Dixon JR, Selvaraj S, Yue F, Kim A, Li Y, Shen Y, Hu M, Liu JS, Ren B. Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature. 2012;485(7398):376–80.
- Liu L, Han K, Sun H, Han L, Gao D, Xi Q, Zhang L, Lin H. A comprehensive review of bioinformatics tools for chromatin loop calling. Brief Bioinform. 2023;24(2):bbad072.
- Forcato M, Nicoletti C, Pal K, Livi CM, Ferrari F, Bicciato S. Comparison of computational methods for Hi-C data analysis. Nat Methods. 2017;14(7):679–85.
- Ay F, Bailey TL, Noble WS. Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts. Genome Res. 2014;24(6):999–1011.
- Rowley MJ, Poulet A, Nichols MH, Bixler BJ, Sanborn AL, Brouhard EA, Hermetz K, Linsenbaum H, Csankovszki G, Aiden EL. Analysis of Hi-C data using SIP effectively identifies loops in organisms from C. elegans to mammals. Genome Res. 2020;30(3):447–58.
- Matthey-Doret C, Baudry L, Breuer A, Montagne R, Guiglielmoni N, Scolari V, Jean E, Campeas A, Chanut PH, Oriol E. Computer vision for pattern detection in chromosome contact maps. Nat Commun. 2020;11(1):5795.
- 12. Roayaei Ardakany A, Gezer HT, Lonardi S, Ay F. Mustache: multi-scale detection of chromatin loops from Hi-C and Micro-C maps using scale-space representation. Genome Biol. 2020;21:1–17.
- Salameh TJ, Wang X, Song F, Zhang B, Wright SM, Khunsriraksakul C, Ruan Y, Yue F. A supervised learning framework for chromatin loop detection in genome-wide contact maps. Nat Commun. 2020;11(1):3428.
- Yang D, Chung T, Kim D. DeepLUCIA: predicting tissue-specific chromatin loops using Deep Learning-based Universal Chromatin Interaction Annotator. Bioinformatics. 2022;38(14):3501–12.
- Zhang S, Plummer D, Lu L, Cui J, Xu W, Wang M, Liu X, Prabhakar N, Shrinet J, Srinivasan D. DeepLoop robustly maps chromatin interactions from sparse allele-resolved or single-cell Hi-C data at kilobase resolution. Nat Genet. 2022;54(7):1013–25.
- Wang F, Gao T, Lin J, Zheng Z, Huang L, Toseef M, Li X, Wong K-C. GlLoop: Robust chromatin loop calling across multiple sequencing depths on Hi-C data. Iscience. 2022;25(12):105535.
- Wang S, Zhang Q, He Y, Cui Z, Guo Z, Han K, Huang D-S. DLoopCaller: A deep learning approach for predicting genome-wide chromatin loops by integrating accessible chromatin landscapes. PLoS Comput Biol. 2022;18(10): e1010572.
- LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.
- Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017;60(6):84–90.
- Chung J, Gülçehre Ç, Cho K, Bengio Y: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR. 2014:abs/1412.3555.
- 21. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process. 1997;45(11):2673–81.
- 22. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. CoRR. 2015:abs/1409.0473.
- Woo S, Park J, Lee J-Y, Kweon IS: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV): 2018; 2018: 3–19.
- Zhu B, Hofstee P, Lee J, Al-Ars Z: An attention module for convolutional neural networks. In: Artificial Neural Networks and Machine Learning– ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part I 30: 2021: Springer; 2021: 167–178.
- Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W. Model-based analysis of ChIP-Seq (MACS). Genome Biol. 2008;9:1–9.
- Dekker J, Mirny L. The 3D genome as moderator of chromosomal communication. Cell. 2016;164(6):1110–21.
- Jin F, Li Y, Dixon JR, Selvaraj S, Ye Z, Lee AY, Yen C-A, Schmitt AD, Espinoza CA, Ren B. A high-resolution map of the three-dimensional chromatin interactome in human cells. Nature. 2013;503(7475):290–4.
- Gibcus JH, Dekker J. The hierarchy of the 3D genome. Mol Cell. 2013;49(5):773–82.

- Handoko L, Xu H, Li G, Ngan CY, Chew E, Schnapp M, Lee CWH, Ye C, Ping JLH, Mulawadi F. CTCF-mediated functional chromatin interactome in pluripotent cells. Nat Genet. 2011;43(7):630–8.
- Sanborn AL, Rao SS, Huang S-C, Durand NC, Huntley MH, Jewett AI, Bochkov ID, Chinnappan D, Cutkosky A, Li J. Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. Proc Natl Acad Sci. 2015;112(47):E6456–65.
- Rao SS, Huang S-C, St Hilaire BG, Engreitz JM, Perez EM, Kieffer-Kwon K-R, Sanborn AL, Johnstone SE, Bascom GD, Bochkov ID. Cohesin loss eliminates all loop domains. Cell. 2017;171(2):305–320. e324.
- Wolff J, Backofen R, Grüning B. Loop detection using Hi-C data with HiCExplorer. Gigascience. 2022;11:giac061.
- Knight PA, Ruiz D. A fast algorithm for matrix balancing. IMA J Numer Anal. 2013;33(3):1029–47.
- 34. Zhang Y, Blanchette M. Reference panel guided topological structure annotation of Hi-C data. Nat Commun. 2022;13(1):7426.
- Shen J, Wang Y, Luo J. CD-Loop: a chromatin loop detection method based on the diffusion model. Front Genet. 2024;15:1393406.
- LeCun Y. Bengio Y. Hinton G: Deep learning nature. 2015;521(7553):436–44.
- 37. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504–7.
- Vincent P, Larochelle H, Bengio Y, Manzagol P-A: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning: 2008; 2008: 1096–1103.
- LeCun Y, Kavukcuoglu K, Farabet C: Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE international symposium on circuits and systems: 2010: IEEE; 2010: 253–256.
- Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J. Recent advances in convolutional neural networks. Pattern Recogn. 2018;77:354–77.
- Rawat W, Wang Z. Deep convolutional neural networks for image classification: A comprehensive review. Neural Comput. 2017;29(9):2352–449.
- Luo J, Gao R, Chang W, Wang J. LSnet: detecting and genotyping deletions using deep learning network. Front Genet. 2023;14:1189775.
- 44. Khan ZY, Niu Z. CNN with depthwise separable convolutions and combined kernels for rating prediction. Expert Syst Appl. 2021;170: 114528.
- Fudenberg G, Mirny LA. Higher-order chromatin structure: bridging physics and biology. Curr Opin Genet Dev. 2012;22(2):115–24.
- She D, Jia M. A BiGRU method for remaining useful life prediction of machinery. Measurement. 2021;167: 108277.
- Liu J, Yang Y, Lv S, Wang J, Chen H. Attention-based BiGRU-CNN for Chinese question classification. J Ambient Intell Humaniz Comput. 2019;10.
- Ester M, Kriegel H-P, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. Inkdd. 1996;96:226–31.
- Schubert E, Sander J, Ester M, Kriegel HP, Xu X. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS). 2017;42(3):1–21.
- Zhang P, Wu H. IChrom-deep: an attention-based deep learning model for identifying chromatin interactions. IEEE Journal of Biomedical and Health Informatics. 2023;27(9):4559–68.
- 51. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. CoRR. 2014:abs/1412.6980.
- Smith LN. A disciplined approach to neural network hyper-parameters: part 1—learning rate, batch size, momentum, and weight decay. CoRR. 2018;abs/1803.09820.
- Xu Z, Dai AM, Kemp J, Metz L. Learning an adaptive learning rate schedule. CoRR. 2019;abs/1909.09712.
- 54. Moreira M, Fiesler E. Neural Networks with Adaptive Learning Rate and Momentum Terms. IDIAP Technical Report. 1995;95–04.
- Durand NC, Robinson JT, Shamim MS, Machol I, Mesirov JP, Lander ES, Aiden EL. Juicebox provides a visualization system for Hi-C contact maps with unlimited zoom. Cell Syst. 2016;3(1):99–101.

- Phillips JE, Corces VG. CTCF: master weaver of the genome. Cell. 2009;137(7):1194–211.
- 57. Li Y, Haarhuis JH, Sedeño Cacciatore Á, Oldenkamp R, van Ruiten MS, Willems L, Teunissen H, Muir KW, de Wit E, Rowland BD. The structural basis for cohesin–CTCF-anchored loops. Nature. 2020;578(7795):472–6.
- Zhang Y, Wong C-H, Birnbaum RY, Li G, Favaro R, Ngan CY, Lim J, Tai E, Poh HM, Wong E. Chromatin connectivity maps reveal dynamic promoter– enhancer long-range associations. Nature. 2013;504(7479):306–10.
- Creyghton MP, Cheng AW, Welstead GG, Kooistra T, Carey BW, Steine EJ, Hanna J, Lodato MA, Frampton GM, Sharp PA. Histone H3K27ac separates active from poised enhancers and predicts developmental state. Proc Natl Acad Sci. 2010;107(50):21931–6.
- Nasmyth K, Haering CH. Cohesin: its roles and mechanisms. Annu Rev Genet. 2009;43:525–58.
- 61. Gligoris T, Löwe J. Structural insights into ring formation of cohesin and related Smc complexes. Trends Cell Biol. 2016;26(9):680–93.
- Rodrigues ÉO. Combining Minkowski and Chebyshev: New distance proposal and survey of distance metrics using k-nearest neighbours classifier. Pattern Recogn Lett. 2018;110:66–71.
- Burr, T. Pattern Recognition and Machine Learning. J Am Stat Assoc. 2008;103(482):886–7.
- 64. Krivega I, Dean A. Enhancer and promoter interactions—long distance calls. Curr Opin Genet Dev. 2012;22(2):79–85.
- Chowdhury HM, Boult T, Oluwadare O. Comparative study on chromatin loop callers using Hi-C data reveals their effectiveness. BMC Bioinformatics. 2024;25(1):123.
- Kim TH, Abdullaev ZK, Smith AD, Ching KA, Loukinov DI, Green RD, Zhang MQ, Lobanenkov VV, Ren B. Analysis of the vertebrate insulator protein CTCF-binding sites in the human genome. Cell. 2007;128(6):1231–45.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.