BMC Bioinformatics

RESEARCH

Open Access

DconnLoop: a deep learning model for predicting chromatin loops based on multi-source data integration

Junfeng Wang[1,2], Kuikui Cheng[1], Chaokun Yan[3], Huimin Luo[3] and Junwei Luo[2*]

*Correspondence:
luojunwei@hpu.edu.cn

[1] School of Physics and Electronic Information Engineering, Henan Polytechnic University, Jiaozuo 454003, China
[2] School of Software, Henan Polytechnic University, Jiaozuo 454003, China
[3] School of Computer and Information Engineering, Henan University, Kaifeng 475001, China

Abstract

Background: Chromatin loops are critical for the three-dimensional organization of the genome and gene regulation. Accurate identification of chromatin loops is essential for understanding the regulatory mechanisms in disease. However, current mainstream detection methods rely primarily on single-source data, such as Hi-C, which limits these methods' ability to capture the diverse features of chromatin loop structures. In contrast, multi-source data integration and deep learning approaches, though not yet widely applied, hold significant potential.

Results: In this study, we developed a method called DconnLoop to integrate Hi-C, ChIP-seq, and ATAC-seq data to predict chromatin loops. This method achieves feature extraction and fusion of multi-source data by integrating residual mechanisms, directional connectivity excitation modules, and interactive feature space decoders. Finally, we apply density estimation and density clustering to the genome-wide prediction results to identify more representative loops. The code is available from https://github.com/kuikui-C/DconnLoop.

Conclusions: The results demonstrate that DconnLoop outperforms existing methods in both precision and recall. In various experiments, including Aggregate Peak Analysis and peak enrichment comparisons, DconnLoop consistently shows advantages. Extensive ablation studies and validation across different sequencing depths further confirm DconnLoop's robustness and generalizability.

Keywords: Chromatin loops, Multi-source data, Deep learning, Feature integration, Clustering

Background

Understanding the three-dimensional organization of chromatin within the cell nucleus is crucial for elucidating the regulatory mechanisms of gene expression and genome function [1].Recent advancements in high-throughput chromosome conformation capture techniques, such as Hi-C [2], have significantly enhanced our ability to explore chromatin architecture within the nucleus, particularly at high-resolution levels. Chromatin structure is further divided into A/B compartments, Topologically

Associating Domains (TADs), and chromatin loops, reflecting regions of chromatin openness and compaction [3–5].

Epigenetics explores heritable modifications in gene function that occur without changes to the underlying DNA sequence. Central to this field are structural components such as DNA-binding proteins and RNAs, alternative DNA conformations, and chemical modifications like methylation [6]. High-throughput sequencing technologies have revolutionized our ability to study these epigenetic factors [7, 8]. For instance, ATAC-Seq (Assay for Transposase-Accessible Chromatin using sequencing) profiles regions of open chromatin, enabling the assessment of chromatin accessibility and the identification of transcription factor binding sites [9]. Chromatin Immunoprecipitation followed by sequencing (ChIP-seq) maps protein-DNA interactions across the genome, providing insights into the regulatory roles of proteins such as transcription factors and histones [10]. Hi-C, a genome-wide chromosome conformation capture technique, examines the three-dimensional architecture of chromatin by quantifying physical interactions between genomic loci that are spatially proximal within the nucleus but may be distant along the linear genome sequence [11]. Collectively, these methodologies deepen our understanding of the complex regulatory networks and spatial organization that underpin genomic function.

Chromatin looping is a critical component of the genome's three-dimensional organization, enabling distal genomic elements to interact functionally despite their linear separation. Mediated by architectural proteins such as CTCF and cohesin complexes, these loops bring enhancers, silencers, and promoters into close spatial proximity, facilitating precise regulation of gene expression [12]. The dynamic modulation of chromatin loops allows cells to respond to developmental cues and environmental stimuli by altering transcriptional programs [13]. Disruptions or aberrations in chromatin looping can lead to misregulation of gene expression networks, contributing to the onset and progression of various diseases [14–16].

For instance, mutations in enhancer elements that affect their looping interactions with target gene promoters have been linked to developmental disorders. In holoprosencephaly, mutations in the SBE2 enhancer disrupt its interaction with the SHH gene promoter, leading to forebrain malformations [17]. Similarly, alterations in the ZRS enhancer can impede its regulatory loop with the SHH promoter in limb buds, resulting in limb malformations such as preaxial polydactyly type 2 (PPD2) [18]. In cancer biology, enhancer hijacking or duplication events can create aberrant loops that enhance oncogene expression. A notable example is the duplication of enhancers near the MYC gene in lung adenocarcinoma, which leads to its overexpression and drives tumorigenesis [19].

Exploring the methodologies used to study chromatin loops is essential. Understanding how these loops are formed, maintained, and altered requires robust techniques that can capture the dynamic interactions within the nucleus. Current methods for detecting chromatin loops can be broadly divided into two main categories: unsupervised and supervised approaches, with the latter further subcategorized based on the type of input data utilized.

(1) Unsupervised Methods; Unsupervised methods rely on statistical models or computational algorithms to identify chromatin loops without requiring labeled datasets. These methods focus on detecting significant interaction peaks from Hi-C contact matrices by modeling background noise, biases, or using image-based correlation techniques.

HiCCUPS [3, 20] is a peak-finding algorithm based on the Poisson distribution that identifies peaks as chromatin interactions by comparing the abundance of reads with local neighborhoods (horizontal, vertical, lower-left, and doughnut). Fit-Hi-C [21] assigns statistical confidence estimates to mid-range intrachromosomal contacts by jointly modeling the random polymer looping effect and previously observed technical biases in Hi-C datasets. HiC-ACT [22] assumes statistical significance between all chromatin interactions and uses the aggregated Cauchy test (ACT) method to learn from adjacent loci statistics to achieve peak identification. HiCExplorer [23] employs negative binomial distribution modeling and the Wilcoxon rank-sum test to identify enriched Hi-C interactions by analyzing candidate regions within their genomic neighborhoods, effectively distinguishing significant interaction peaks from background noise. InferLoop [24] method enhances signal by grouping adjacent cells into respective bins and uses a metric akin to a perturbed Pearson correlation coefficient to derive loop signals. Chromosight [25] employs a computer vision-based approach to detect specific templates (e.g., loops or boundary cores) in images (i.e., Hi-C contact matrices) by correlating each sub-image with the template and selecting the sub-images with the highest correlation as the template's representation. Mustache [26] utilizes the scale-space theory from computer vision to detect blob-like objects (i.e., loop structures) in Hi-C contact matrices, determining the presence of loops by evaluating the statistical significance of local enrichment of pixels.

(2) Supervised Methods; Supervised methods depend on labeled training datasets and are designed to predict chromatin loops by leveraging machine learning or deep learning models. These methods are categorized based on the type of input data they integrate.

(i) Methods Using Hi-C Contact Matrices;

Peakachu [27] uses a supervised random forest classification framework for predicting chromatin loops from genome-wide contact maps, capable of identifying a unique set of short-range interactions. RefHiC [28] leverages the advantages of deep learning and high-quality Hi-C datasets from reference panels, utilizing attention mechanisms to identify loop structures in the genome. Be-1DCNN [29] uses a bagging ensemble learning strategy and one-dimensional convolutional neural networks (1DCNN) to enhance the accuracy and reliability of chromatin loop prediction. GILoop [30], a dual-branch neural network model, leverages graphical and image representations of Hi-C contact matrices to identify genome-wide CTCF-mediated chromatin loops. CD-loop [31] uses a pre-trained estimation model on Hi-C contact matrices, combined with the denoising process of a diffusion model, to predict chromatin loops.

(ii) Methods Integrating Hi-C Contact Matrices and Multi-omics Data;

LoopPredictor [32] adopts an integrated machine learning model, combines the H3K27ac/YY 1 HiChIP dataset and an ensemble of multi-omics features based on Random Forests and Gradient Boosted Regression Trees to predict long-range enhancer-mediated loops. DLoopCaller [33] employs a deep learning framework to integrate Hi-C contact matrices and accessible chromatin data to predict chromatin loops.

Previous studies have observed that CTCF (CCCTC-binding factor), a key structural protein involved in the formation of chromatin loops, binds to specific DNA sequences and acts as an insulator, forming loops that connect distant genomic regions within the chromatin. Co-oriented CTCF binding sites are notably enriched at the anchor points of these loops across various cell types [34–36]. Therefore, we leverage the spatial information of potential loop anchors provided by CTCF binding regions obtained from CTCF ChIP-seq data to enhance the identification of chromatin loops.

Chromatin loops frequently connect enhancers and promoters, facilitating regulatory interactions that modulate gene expression [37]. Regions of accessible chromatin are often associated with active regulatory elements such as enhancers, and ATAC-seq data can assess chromatin accessibility, revealing regions where transcription factors and regulatory proteins bind [38, 39]. By overlaying ATAC-seq data with Hi-C contact matrices, it is possible to precisely identify open chromatin regions overlapping with loop anchors, thereby uncovering the interactions between enhancers and promoters.

In this paper, we developed a method, named DconnLoop, based on deep learning model, which integrates Hi-C contact matrices, ATAC-seq data, and CTCF CHIP-seq data to extract and fuse features, and accurately identify chromatin loops.

Methods

DcoonLoop adopts Hi-C contact matrices, ATAC-seq data and CTCF ChIP-seq data as input, and the process of DconnLoop is shown in Fig. 1. In this study, we used 10 kb resolution bins to construct the input sub-matrices. The steps of DconnLoop are as follows: (1) Generating sub-matrices. For one bin-pair (one chromatin loop) in the Hi-C contact matrix, DconnLoop constructs three sub-matrices based on Hi-C contact matrix, ATAC-seq data and CTCF ChIP-seq data. Meanwhile, DconnLoop sets filtering conditions to select high-confidence sub-matrices. (2) Extracting Feature. DconnLoop uses the ResNet model, Directional Prior Extraction, Sub-path Direction Excitation Model,

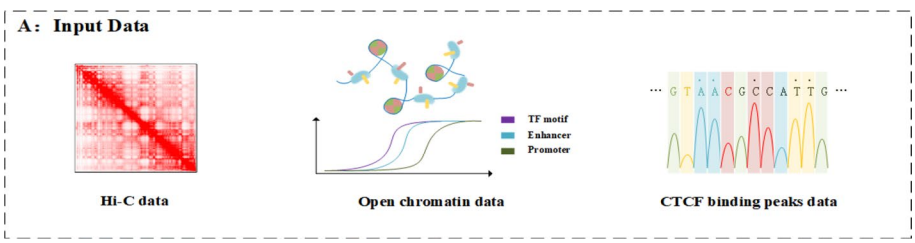

Fig. 1 The workflow of DconnLoop. **A** The input data includes Hi-C contact matrices, open chromatin data from ATAC-seq, CTCF ChIP-seq binding peak data. **B** Generate model input submatrix. **C** Extracting Feature. **D** Predicting candidate loop. **E** Clustering

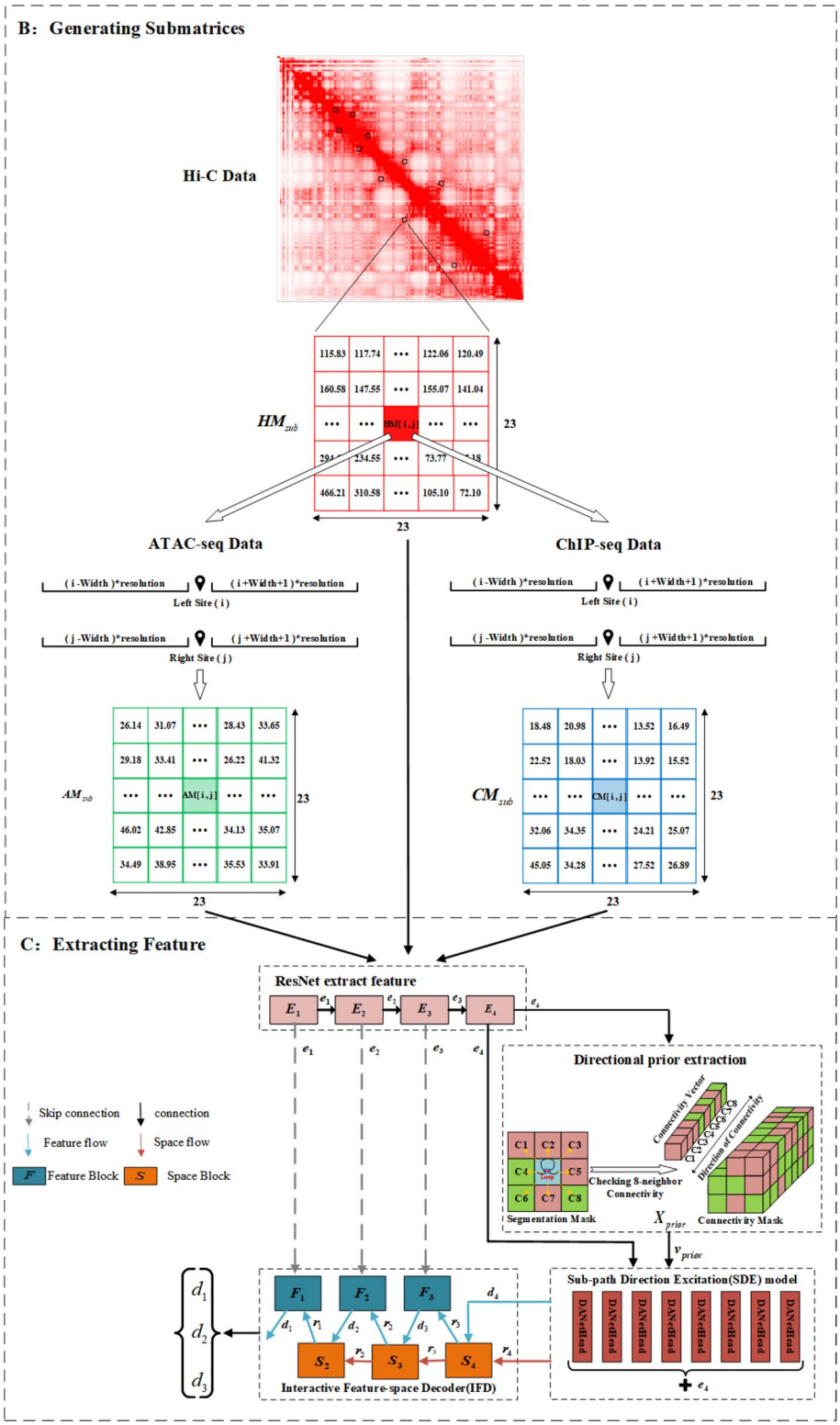

Fig. 1 continued

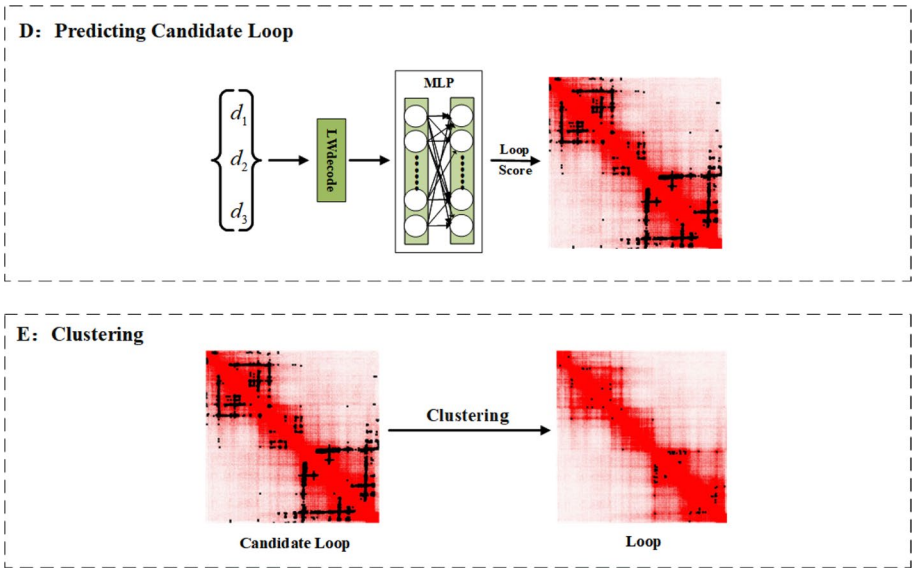

Fig. 1 continued

and Interactive Feature-space Decoder for feature extraction and fusion of the input sub-matrices. (3) Predicting candidate loop. DconnLoop adopts MLP model to score each chromatin loop, based on the features extracted by previous step, and then identifies the candidate loops. (4) Clustering. Due to experimental noise and sequencing technology effects, it may lead to false positives or redundant positive candidate loops, clustering can be used to group adjacent candidate loops and identify the most representative loops. The detail of DcoonLoop is illustrated below.

Generating sub-matrices

The input of DconnLoop includes Hi-C contact matrix, ATAC-seq data and CTCF ChIP-seq data. In Hi-C contact matrix **HM**, $HM[i, j]$ refers to the interaction number between the i-th bin and j-th bin.

Given the nature of chromatin loops, which are predominantly found within a 2 MB range near the diagonal of the Hi-C contact matrix [3], DconnLoop limits its prediction scope to between 11 and 300 bins offset from the diagonal. To ensure the robustness of loop predictions and to minimize false positives, we applied a Poisson distribution model for significance testing on each interaction pair within this prediction range. In this process, calculating the expected value $\lambda_{i,j}$ for each interaction pair is a critical step. The expected value $\lambda_{i,j}$ reflects the expected interaction frequency between any two positions within a specified offset distance $d=|i-j|$ in the contact matrix.

Specifically, for each offset d, we initially calculate the average value of the diagonal elements as the initial estimate of the expected value. This preliminary step allows us to capture the overall trend without being overly influenced by extreme values at specific positions. Subsequently, we adjust these expected values according to the corresponding weights, which are derived from the Knight-Ruiz (KR) normalization of the Hi-C contact matrix. These weights ($w[i]$ and $w[j]$) represent correction factors for bins i and j respectively, which are computed to ensure that the sum of contacts in each row and column of

the matrix are balanced, thus correcting for systemic biases such as sequencing depth and local coverage variability. The final expected value $\lambda_{i,j}$ for each bin pair (i, j) is calculated using the following formula:

$$\lambda_{i,j} = \lambda_d \times \frac{1}{w[i] \times w[j]}$$

where λ_d represents the expected value at offset d, and $w[i]$ and $w[j]$ are the weights for positions i and j, respectively, derived through the KR normalization process. These weights ensure that biases introduced by non-uniform coverage across different bins are accounted for, providing a balanced and unbiased estimation of chromatin interactions.

Finally, using this adjusted expected value, we can calculate the p-value for each interaction pair:

$$p_{i,j} = P(X \geq HM[i,j] | \lambda_{i,j})$$

Here, $p_{i,j}$ represents the probability that, under the condition where the parameter is $\lambda_{i,j}$, the random variable X is greater than or equal to the observed interaction count $HM[i, j]$. We consider interaction pairs with p-values less than 0.01 as statistically significant and retain them for further analysis.

Next, for each significant interaction pair (i, j) filtered by p-value, a 23×23 Hi-C submatrix was generated by extending 11 bins in all directions—up, down, left, and right—from the center $bin_{i,j}$ within the Hi-C contact matrix, as shown in Fig. 1B. To ensure the generation of valid submatrices $\mathbf{HM_{sub}}$ in the Hi-C contact matrix **HM**, we implemented stringent screening conditions. These conditions include: Boundary Check (The selected window must remain entirely within the bounds of the Hi-C contact matrix **HM**, ensuring that the submatrix does not exceed the matrix boundaries and that no out-of-bounds interactions are included. This is achieved by filtering out interaction pairs at the edges of the matrix, where the full 23×23 window cannot be generated), Non-zero Element Proportion (To avoid processing matrices with excessive noise or background values, we excluded submatrices with a low proportion of non-zero values. Specifically, submatrices were filtered if the number of non-zero elements was less than 10% of the total number of elements in the window. This ensures that the submatrix contains a meaningful amount of interaction data rather than sparse or insignificant values), Center Signal Significance (We further ensured that the signal at the center of the window was significantly higher than the surrounding background to exclude potential low-quality or noise-dominated data. This was done by comparing the center value of the submatrix with the mean value of the surrounding area. If the ratio of the center value to the mean of the surrounding area was less than 0.1, the submatrix was excluded. This threshold helps ensure that only submatrices with biologically relevant signals are retained, and noise-dominated submatrices are discarded).

For the final retained submatrices, we performed normalization using the expected values λ_d calculated based on the diagonal offsets within the genomic regions in the matrix.

$$HM_{norm}[i,j] = \frac{HM[i,j]}{\lambda_d}$$

For the submatrix $\mathbf{HM}_{\mathbf{sub}}$, we produce two other submatrices $\mathbf{AM}_{\mathbf{sub}}$ and $\mathbf{CM}_{\mathbf{sub}}$ based on ATAC-seq data and ChIP-seq data, as shown in Fig. 1B. The specific steps are as follows(See Supplementary Figure S1 for more details). (1) Determining the Range and Extracting Signals. The center coordinates (i, j) used to generate $\mathbf{HM}_{\mathbf{sub}}$ are mapped to the corresponding left locus i and right locus j in the ATAC-seq and ChIP-seq data, respectively. The signal values within a window of width 11 around these loci are extracted at a resolution of 10 kb. (2) The extracted signals from the ATAC-seq and ChIP-seq data are segmented into 23 parts, each representing a 10 kb region. (3) Bilateral Filtering and Weight Assignment. After segmentation, a bilateral filter is applied to each segment to assign weights, balancing smoothing with the preservation of signal edges. First, a smoothed version of each segment is computed using a Gaussian filter to provide a baseline and mitigate high-frequency noise. The difference between the original signal and the smoothed values is then calculated, indicating local variations and significant changes. This difference is subsequently filtered again with another Gaussian filter to refine the weights, which emphasize significant changes while reducing the influence of less informative data points. These refined weights are used to compute a weighted average for each segment, resulting in a single representative value. Consequently, each extracted region is converted into a 23-dimensional vector, with each element representing the weighted average of a segment, effectively capturing the relevant signal characteristics. (4) Transposing and Dot Product to Generate Submatrix. Once the 23-dimensional vectors for both the left and right sites (i and j) are obtained, we use a transposed dot product operation to generate the final submatrices, $\mathbf{AM}_{\mathbf{sub}}$ and $\mathbf{CM}_{\mathbf{sub}}$. The dot product operation results in a 23×23 matrix, where each element represents the interaction between segments from the left and right loci.

Extracting features

DconnLoop uses the ResNet model, Directional Prior Extraction, Sub-path Direction Excitation Model, and Interactive Feature-space Decoder for feature extraction and fusion of the input multi-source data, as shown in Fig. 1C. In the model, the Directional Prior Extraction module extracts directionality information by examining the connectivity of each feature element with its eight neighboring elements. This generates a feature mask vector containing 8 channels, where each channel represents the connectivity of the element in a specific direction. This connectivity information helps the model determine which elements have more significant associations, potentially indicating physical chromatin loop contacts. Subsequently, the Sub-path Direction Excitation (SDE) module further processes these directionality features through a multi-path feature processing mechanism, which includes the Position Attention Module (PAM) and the Channel Attention Module (CAM). The Position Attention Module captures spatial dependencies, while the Channel Attention Module emphasizes the directional connectivity features. This mechanism allows the input feature map to highlight significant regions within the feature matrix that may represent chromatin loop areas. This approach effectively utilizes directional connectivity to capture significant features of chromatin loop regions, enhancing the model's accuracy in predicting chromatin loop locations. By using directional relationships and attention mechanisms, the model can pinpoint

prominent areas within the feature map, facilitating the identification of potential chromatin loops.

(1) ResNet module. For the input sub-matrices $\mathbf{HM_{sub}}$, $\mathbf{AM_{sub}}$ and $\mathbf{CM_{sub}}$, ResNet [40] is used as the backbone network to efficiently represent the input features by leveraging the advantages of residual learning. Initially, the input feature maps undergo initial convolution and pooling. Then, three residual layers are used, employing 3, 4, and 5 residual blocks, respectively, to progressively deepen the network and extract and accumulate feature information at different spatial scales and depths.

$$x_{i+1} = F(x_i) = x_i$$

This formula demonstrates the relationship between the input and output feature maps at each layer within the ResNet network. Specifically, x_i refers to the input feature map at the i-th layer. The residual mapping, $F(x_i)$, indicates the transformation applied to the input x_i after it passes through the residual block. Finally, the output feature map xi_{+1} at the $i+1$-th layer is derived by summing the input feature map and the residual mapping.

(2) Directional prior module. The deepest feature maps extracted by the network are subjected to directionality prior extraction and re-encoding, aimed at capturing the directional features inherent in the data [41]. As shown in Fig. 1C. For each element of the matrix, the connectivity with its eight surrounding neighbors (including the top, bottom, left, right, and four diagonal directions) is checked. The connectivity status in each direction is encoded and stored in the corresponding channel. The final connection mask transforms each element into a vector containing 8 channels, with each channel corresponding to the connectivity status of one neighboring direction.

Next, we upsample the extracted directional information $\boldsymbol{X_{prior}}$ to match the input size and re-encode it using a 1×1 convolution kernel $\boldsymbol{W_1}$. Subsequently, global average pooling is applied to compress the spatial information, followed by a convolution kernel $\boldsymbol{W_2}$ to transform it into a feature map with the same number of channels as the original latent feature $\boldsymbol{e_4}$.

$$\tilde{X} = W_1 \boldsymbol{X_{prior}}$$

$$GAP(X) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X(i,j)$$

$$v_{prior} = \delta(\boldsymbol{W}_2 GAP(\tilde{\boldsymbol{X}}))$$

where H and W are the height and the width of a feature map, $\boldsymbol{W_1} \in R^{Ce_4 \times C_k}$, $\boldsymbol{W_2} \in R^{Ce_4 \times Ce_4}$, $\mathrm{C_k}$ is the channel of $\boldsymbol{X_{prior}}$, and δ is the ReLu activation, $\tilde{\boldsymbol{X}}$ represents the re-encoded features, $\boldsymbol{v_{prior}}$ represents the directional prior.

(3) Sub-path Direction Excitation module (SDE). The SDE module divides the input latent features $\boldsymbol{e_4}$ and directional prior $\boldsymbol{v_{prior}}$ into eight parts using channel slicing.

Each part is processed using a multi-path feature processing mechanism (DANet-Head), which includes a Position Attention Module (PAM) and a Channel Attention Module (CAM). These mechanisms are utilized to extract significant spatial and channel features, particularly when dealing with data with complex spatial and directional relationships. The feature maps from each sub-path are then concatenated and fused with the original input feature map, thus achieving feature enhancement.

$$\boldsymbol{PCM} = \phi^{-1}(\boldsymbol{V} \cdot \sigma(\phi(\boldsymbol{Q}) \cdot \boldsymbol{K})) + \gamma \boldsymbol{X}$$

$$\boldsymbol{CAM} = \phi^{-1}(\sigma(max(\boldsymbol{Q} \cdot \boldsymbol{K}) - \boldsymbol{Q} \cdot \boldsymbol{K}) \cdot \phi(\boldsymbol{X})) + \gamma \boldsymbol{X}$$

where the input feature map $X \in R^{B \times C \times H \times W}$; $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ represent the feature representations of Query, Key, and Value, respectively. The term $\gamma \boldsymbol{X}$ denotes the residual connection, where γ is a learnable parameter used to adjust the relationship between the input features and the attention-enhanced features. The function $\sigma(.)$ refers to the Softmax function, which is used to generate attention weights. The functions $\phi(.)$ and $\phi^{-1}(.)$ represent the reshape and inverse reshape operations, respectively, Specifically:

(1) Reshape Operation $\phi(.)$: This operation is used to transform the input feature map $X \in R^{B \times C \times H \times W}$ into a suitable shape for attention calculation. For the Position Attention Module (PAM), the feature map is reshaped into $\phi(\boldsymbol{Q}) \in R^{B \times H \times W \times C'}$, where C' is the reduced number of channels obtained through convolution. In the Channel Attention Module (CAM), the reshape operation is used to flatten the spatial dimensions of the feature map, resulting in $\phi(\boldsymbol{X}) \in R^{B \times C \times (H \times W)}$.

(2) Inverse Reshape Operation $\phi^{-1}(.)$: After the attention mechanism has been applied, the output needs to be reshaped back to its original dimensions to ensure compatibility with subsequent layers in the network. The inverse reshape operation restores the feature map from the flattened representation to its original form, $\phi^{-1}(.) \in R^{B \times C \times H \times W}$

The reshape and inverse reshape operations ensure that the feature map has the correct dimensions for both the attention calculation and subsequent integration with residual connections.

(4) Interactive Feature-space Decoder (IFD) module. The model's feature representation capabilities are enhanced through mechanisms of multi-scale feature processing and fusion. The IFD module consists of three Space Blocks and three Feature Block. Each Space Block uses a Context Encoder to extract global features from the outputs of the previous stage ($\boldsymbol{r_4}$, $\boldsymbol{r_3}$, $\boldsymbol{r_2}$) and a Content Encoder to extract local features from the current level's feature map ($\boldsymbol{d_4}$, $\boldsymbol{d_3}$, $\boldsymbol{d_2}$). The global and local features' relevance is computed and integrated through weighted fusion.

$$\boldsymbol{n_i} = GAP(\boldsymbol{r_i})$$

In the space flow, r_i represents the direction-enhanced feature map that is processed within the Space Block. The Global Average Pooling (GAP) operation is applied to r_i to generate the directional embedding n_i. This embedding retains the directional information from the feature map. The resulting n_i serves as a high-level directional representation, which will be used in subsequent steps to enhance the directional information within the main feature map.

$$d'_i = W_{d_i} d_i, \quad n'_i = W_{n_i} n_i$$

The projection uses two 1×1 convolutional projectors, $W_{d_i} \in R^{C_{d_i} \times C_{d_i}}$ and $W_{n_i} \in R^{C_{d_i} \times C_{d_i}/2}$, which compress the input feature map d_i and directional embedding n_i into the same dimension.

$$\alpha_{nd} = \sigma\left(d'_i \cdot n'_i\right)$$

The representations $d_i^{'}$ and $n_i^{'}$ are the re-projected main feature map and directional embedding obtained from the previous step. Their similarity is computed through a dot product operation, and the result is passed through a Sigmoid activation function σ to produce a normalized attention map α_{nd}. This attention map α_{nd} emphasizes the direction-related information within the feature map while suppressing irrelevant information, allowing the model to better focus on the relevant features, thereby improving the accuracy of classification or detection tasks.

$$r_{i-1} = \alpha_{nd} \cdot d'_i$$

The attention map α_{nd} is applied to $d_i^{'}$ through a dot product operation, resulting in an enhanced directional feature map r_{i-1}. The final r_{i-1} feature map embeds directional information and can be used for feature flow processing in the next layer.

For each Feature Block, features extracted at different depths from the backbone network (ResNet)—namely (e_1, e_2 and e_3)—and the directionality-prior-enhanced features processed by the spatial blocks (r_3, r_2 and r_1) are fused at the same dimension through skip connections.

Predicting candidate loop

We integrate the outputs d_1 to d_3 after feature fusion from each feature block in the IFD module to obtain the final output, as shown in Fig. 1D. For the integrated feature outputs d_1 to d_3, we constructed a lightweight decoder (LWdecoder). This decoder generates multiple decoding paths by performing upsampling and convolution operations on input feature maps of different scales, and fuses the features from different levels. This allows the model to effectively integrate multi-scale information, enhancing its ability to recognize chromatin loops.

Subsequently, the final fused feature map is passed through convolution, flattening, fully connected layers, and nonlinear activation, with a Sigmoid activation function applied to achieve probabilistic prediction for binary classification.

Obtaining final loop based on clustering

To ensure the reliability and biological significance of these predicted interaction pairs (high confidence loops), further clustering analysis was conducted, as shown in Fig. 1E. Clustering analysis helps to integrate spatially adjacent or nearby interaction pairs, identifying regions with frequently occurring interactions. These regions are more likely to represent actual chromatin loops, thereby enhancing the confidence in the predictions.

Specifically, we integrated density estimation and density-based clustering algorithms to enhance the accuracy and robustness of clustering. Initially, we retained high-scoring candidate loops by applying a confidence score threshold. Next, we identified high-frequency anchor points separately at both the left and right anchor regions. These high-frequency anchor points were then combined as boundary points, and HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [42] was applied to locally cluster within these high-frequency combined regions. HDBSCAN extends the traditional DBSCAN algorithm, transforming it into a hierarchical clustering algorithm capable of extracting hierarchical clusters at different density levels. It provides more robust and efficient clustering results when handling complex and density-uneven datasets.

After executing local clustering, a certain amount of redundant predictions remained, necessitating further refinement. We constructed a KDTree to rapidly query the distances between the current loop at position i and other loops at positions j within its neighborhood. For each loop, the local density ρ was estimated based on the weighted sum of distances from all loops within its neighborhood, and the minimum distance $\delta = |i\text{-}j|$ to a neighboring loop with higher density was calculated.

$$\rho_i = \sum_{j \in N(i)} e^{-(\frac{\delta}{R})^2} \cdot IF[j]$$

Here, ρ_i represents the local density estimate of loop i; $N(i)$ denotes the neighborhood set of loop i; R indicates the size of the neighborhood range; and IF stands for interaction frequency. This algorithm effectively selects loops that are both in close proximity and exhibit high interaction frequencies. By employing a data-driven approach to determine the thresholds for ρ and δ, the method can better adapt to the variations and characteristics of different datasets.

Generating training data and training

In our preliminary dataset, CTCF mediates long-range chromatin interactions, playing a key role in the loop extrusion model and the formation of topologically associating domains (TADs). On the other hand, H3K27ac marks active enhancers and promoters, capturing potentially more dynamic and shorter-range promoter-enhancer interactions [27]. Therefore, we integrate CTCF ChIA-PET [43] and H3K27ac HiChIP [44] data to construct a non-redundant interaction set. This integration leverages the strengths of both experimental methods to provide a more comprehensive detection of chromatin interactions.

We mapped the interaction coordinate pairs from the integrated CTCF ChIA-PET and H3K27ac HiChIP data to the Hi-C contact matrix and apply the methods described in

the 'Generating Submatrices' subsection to generate the Hi-C submatrices $\mathbf{HM}_{\mathbf{sub}}$, followed by rigorous screening based on several strict criteria, including boundary checks to ensure that the submatrices did not exceed the Hi-C contact matrix boundaries, non-zero element proportion filtering to exclude sparse submatrices with insufficient interaction data, and ensuring center signal significance, where the central value of the submatrix was required to be significantly higher than the surrounding background values. After these filtering steps, we normalized the submatrices based on the relative diagonal offset of their center position within the Hi-C contact matrix, ultimately generating the final set of positive Hi-C submatrices for training.

Subsequently, we mapped the effective interaction coordinate pairs, used for generating positive Hi-C submatrices, to the ATAC-seq and ChIP-seq datasets. By applying the methods for generating submatrices $\mathbf{AM}_{\mathbf{sub}}$ and $\mathbf{CM}_{\mathbf{sub}}$ described in the "Generating Submatrices" section, such as determining the range and extracting signals, dividing the signal range into bins based on the resolution, performing bilateral filtering within each bin to collect weights for weighted averaging, and using transposing and dot products to generate submatrices, positive sample submatrices from the ATAC-seq and ChIP-seq datasets were also generated.

To generate negative samples interaction coordinates sites, we employed a random sampling strategy based on distance distribution and long-range interactions to capture different types of negative interactions. First, we applied Gaussian kernel density estimation (KDE) to the pairwise distances between the midpoints of the positive sample coordinates, modeling the probability density function (PDF) of these distances. We then sampled from this distribution to generate negative samples interaction coordinates sites. This step ensures that the negative samples interaction coordinates sites reflect similar genomic distance distributions as the positive samples interaction coordinates sites but lack their biological significance. Second, we randomly sampled long-range interactions, which are less likely to form actual loop structures due to their large distances. By combining these two components, we created a diverse set of negative samples interaction coordinates sites.

Given the limited number of positive interaction sites, we sampled negative interaction sites at five times the number of positive interaction sites to maximize the utilization of the available genomic data. Subsequently, we applied the methods outlined in the "Generating Submatrices" section to generate submatrices for Hi-C, ATAC-seq, and ChIP-seq datasets from these negative interaction sites. Specifically, we followed the same process used for the positive samples, ensuring that the negative samples were processed in a manner consistent with the positive samples, enabling fair comparisons and maintaining biological relevance. The computational time required for generating these training datasets and the data volumes are detailed in Supplementary Tables S1 and S2, respectively.

To address the potential issue of overly distinguishable negative samples interaction, we adjusted the proportion of short-range and long-range interaction in the negative interaction set. We ensured that short-range interaction dominate the negative interaction set, with a higher proportion of short-range interaction than those found in the positive interaction set(See Supplementary Figure S2A). These short-range interaction are defined by anchor points that are closer in distance and near the Hi-C diagonal. This

adjustment makes the negative sample interaction set more similar to the positive interaction set in terms of distance distribution and biological significance.

Additionally, we conducted K-means clustering (See Supplementary Figure S2B) and regression model analyses (See Supplementary Figure S2C, D) on the distance and interaction frequency features of both positive and negative samples interaction sites. Our results show that the negative samples interaction sites are now more difficult to distinguish from the positive samples interaction sites, demonstrating that the revised negative sample interaction sites generation strategy leads to a more balanced and biologically meaningful dataset. Following the modification of our sampling strategy, we conducted comprehensive retraining and evaluation procedures on the adjusted dataset. The detailed results are presented in Supplementary Figures S3-S6, including ablation studies with different input data combinations (Supplementary Figure S3), performance comparisons with logistic regression (Supplementary Figure S4), cross-cell type and cross-species predictions (Supplementary Figure S5), and comparative analyses with other state-of-the-art tools (Supplementary Figure S6).

In this study, we employed Leave-One-Out Cross-Validation (LOO-CV) for training and validating the model, as illustrated in Fig. 2. Specifically, we partitioned the dataset by alternately assigning 22 chromosomes to the training set and 1 chromosome to the test set in each iteration of cross-validation. Within each training set, we separated all positive and negative samples, and the negative samples were further split using five-fold cross-validation (KFold) into five subsets (Negative1 to Negative5). For each fold, we combined all positive samples with one subset of negative samples, with 80% of the data used for training and 20% for validation. In each iteration, we trained five models (Model1 to Model5) and selected the best-performing model based on validation performance (by specify the metric, e.g., F1-score) as the final model for that fold. The selected best model was then applied to predict on the balanced test set to obtain the final results.

By combining all positive samples with different sets of negative samples across multiple iterations, this sample partitioning method maximizes the utilization of positive data, mitigating potential biases from the imbalance between positive and negative

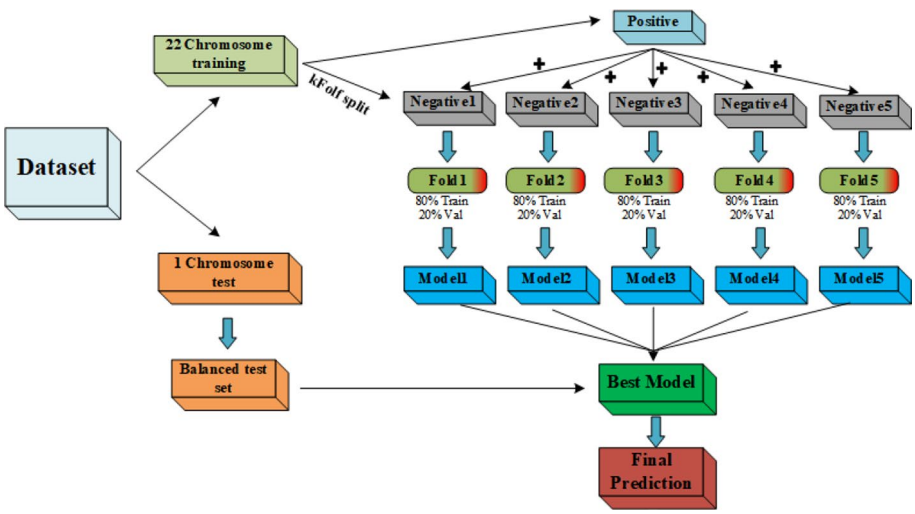

Fig. 2 Dataset partitioning and training process

sample ratios [45]. Additionally, training the model on different subsets of negative samples in each iteration introduces greater variability, which improves its ability to generalize and distinguish between positive and negative samples, enhancing the model's robustness and stability across different data distributions.

To prevent overfitting during training, we implemented early stopping and learning rate adjustment strategies. Early stopping was triggered if the validation loss failed to improve after five consecutive epochs, terminating the training process. Additionally, we applied the CosineAnnealingWarmRestarts [46] learning rate schedule, which cyclically adjusts the learning rate by gradually decreasing it within each cycle and resetting it at the beginning of the next cycle, helping the model escape local minima and achieve better generalization. All training procedures were conducted on a Linux system with an RTX 4090 GPU.

The selection of hyperparameters was guided by a combination of empirical experimentation and established best practices for training deep learning models. Specifically, the batch size was set to 256, after testing several values to strike a balance between memory usage and model performance. The initial learning rate was set to 0.001, based on preliminary trials, and the learning rate decay was modeled using a polynomial schedule to ensure smooth convergence. A momentum value of 0.9 and a weight decay of 0.0005 were chosen based on values commonly used in similar studies [33]. The number of training epochs was set to 30, with early stopping applied to prevent overfitting.

Results

Detail of dataset

All datasets utilized in this study are summarized in Supplementary Table S3 in the Supplementary Materials. For all raw Hi-C data within the data file, we converted the format to cool and applied KR normalization using the 'hicConvertFormat' command from the HiCExplorer [23] tool. All experiments were conducted using contact matrices at a resolution of 10 kb.

Ablation experiment analysis

To validate that the integration of multiple source data (Hi-C, ATAC-seq, and ChIP-seq) significantly enhances model performance, we conducted ablation experiments comparing multi-source data, single-source data, and dual-source data. We specifically analyzed their performance on GM12878 cells by examining the Precision-Recall Curve (PRAUC), F1-score, and Matthews Correlation Coefficient (MCC).

The tests and comparative analyses were performed on chromosomes 15, 16, and 17. As shown in Fig. 3A, the model achieved the best performance across all evaluation metrics when utilizing the integrated multi-source data. Even when using only Hi-C data, our model maintained high performance, with PRAUC and F1-score both exceeding 97% and 93%, respectively. As depicted in Fig. 3B, the overall performance of the curve slightly decreased, particularly in regions with high recall rates, indicating that a single data type might not capture the full range of feature information. When using dual-source data combinations, as shown in Figs. 3C and 3D, the model's performance was intermediate between the results obtained from using all data and single-source data, suggesting that two data types can complement each other but are still not

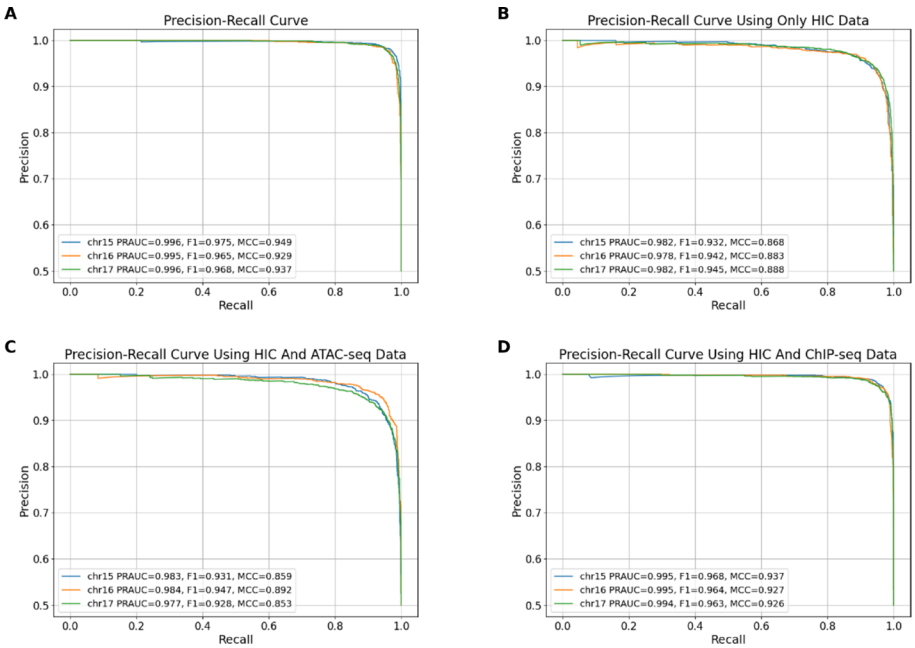

Fig. 3 Ablation experiments using multi-source data versus single Hi-C data and dual data combinations. **A** Model performance with all source data. **B** Model performance using only Hi-C data. **C** Model performance using a combination of Hi-C data and ATAC-seq data. **D** Model performance using a combination of Hi-C data and ChIP-seq data

as comprehensive as the full dataset. Additionally, the combination of ChIP-seq and Hi-C data exhibited performance close to that of the full dataset. This is because ChIP-seq data identifies transcription factor binding sites on the genome, and the CTCF transcription factor is considered a key regulator in chromatin loop structures. This suggests that features derived from CTCF ChIP-seq data play a crucial role in constructing chromatin loops, consistent with previous findings [47, 48].

Comparison of model performance

To evaluate the performance of the DconnLoop model, we compared it with existing deep learning methods, DLoopCaller, and machine learning methods, Peakachu, on GM12878 cells. In Table 1, we used the original same input data and compared the models under their respective standard input configurations: DconnLoop utilized multi-source data inputs comprising three data types, DLoopCaller employed the specified dual data combination input, and Peakachu utilized the specified Hi-C data input, using only MCC (Matthews Correlation Coefficient) as the evaluation metric. From the test results, the DconnLoop model demonstrated significant superiority across all evaluation metrics, particularly in terms of higher F1 score, PRAUC (Precision-Recall Area Under Curve), and MCC. This advantage may be attributed to DconnLoop's ability to integrate multi-source data in feature extraction and fusion, capturing and understanding the complexity of chromatin interactions more comprehensively. In contrast, DLoopCaller and Peakachu may have certain limitations in data input diversity and the depth of feature extraction.

Table 1 Model performance using each tool's respective input specifications

Chromosome	Method	PRAUC	F1 Score	MCC
Chr 15	**DconnLoop**	**0.996**	**0.975**	**0.949**
	DLoopCaller	0.932	0.911	0.838
	Peakachu	0.966	0.905	0.705
Chr 16	**DconnLoop**	**0.995**	**0.965**	**0.929**
	DLoopCaller	0.921	0.876	0.790
	Peakachu	0.966	0.905	0.706
Chr 17	**DconnLoop**	**0.996**	**0.968**	**0.937**
	DLoopCaller	0.932	0.921	0.848
	Peakachu	0.966	0.905	0.707

Table 2 Model performance using only Hi-C data as input

Chromosome	Method	PRAUC	F1 Score	MCC
Chr 15	**DconnLoop**	**0.982**	**0.932**	**0.868**
	DLoopCaller	0.925	0.889	0.807
	Peakachu	0.966	0.905	0.705
Chr 16	**DconnLoop**	**0.978**	**0.942**	**0.883**
	DLoopCaller	0.924	0.887	0.804
	Peakachu	0.966	0.905	0.706
Chr 17	**DconnLoop**	**0.982**	**0.945**	**0.888**
	DLoopCaller	0.932	0.908	0.830
	Peakachu	0.966	0.905	0.707

Subsequently, we further compared the three models using only Hi-C data as input, as shown in Table 2. Even under the condition of a single Hi-C data input, the DconnLoop model still outperformed the other two methods, demonstrating its strong generalization ability and robustness in handling single data types. This further validates the broad applicability and superior performance of the DconnLoop model under different data combinations and input conditions.

To ensure fair comparison and validate our method's superiority, we conducted additional experiments using unified filtering criteria (as described in Generating Sub-matrices section for submatrix sample filtering) across all supervised learning methods. We applied our filtering standards, which consider biological factors such as Hi-C matrix boundaries, signal-to-noise ratios, and known distance constraints in chromatin loop formation, to generate a standardized dataset. Using this unified dataset, we compared DconnLoop with Peakachu and DLoopCaller (detailed results in Supplementary Figures S10).

For unsupervised methods (Mustache and Chromosight), which operate directly on Hi-C contact matrices without requiring training data, we evaluated their predictions through comprehensive biological feature analyses, including APA scores, CTCF binding site enrichment, and experimental validation support.

Cross-cell line and cross-species predictive ability of the model

To evaluate whether a model trained on one cell type can be applied to other cell types, we utilized a model trained on human lymphoblastoid cell line GM12878 using CTCF ChIA-PET and H3K27ac HiChIP data to predict loops in human leukemia cells (K562), human embryonic stem cells (H1ESC), and mouse embryonic stem cells (MESC).

For the K562 cell dataset, we generated data using loops validated by CTCF ChIA-PET and H3K27ac HiChIP experiments. The H3K27ac HiChIP data [44] was extracted from the Hi-C pipeline of filtered HiChIP sequencing reads using juicer_tools [3]. For the experimentally validated loops in H1ESC and MESC, we used CTCF ChIA-PET and SMC1 HiChIP, respectively.

As shown in Figs. 4A, B, and C, our model demonstrated high Precision-Recall Curve performance across different cell and species types. The consistent prediction performance in different human cell types (H1ESC and K562) indicates that a model trained on one human cell type can generalize well to other human cell types. Although there was a slight decline in performance in mouse embryonic stem cells (MESC), the model still maintained high precision and recall, suggesting that while cross-species prediction poses greater challenges, models trained on human cell types still exhibit potential for application in mouse cells. This also highlights the differences between using control groups derived from ChIA-PET and HiChIP experimental techniques alone versus using them in combination.

Analysis of chromatin loop detection results

We conducted a comparative analysis of the detection results from all tools on chromosomes 15, 16, and 17 in GM12878 cells. In the binary classification predictions by Peakachu, DLoopCaller, and DconnLoop, the probability values assigned to each pixel can serve as filtering criteria; setting a higher probability threshold results in fewer but higher quality loops. We standardized the settings using a probability threshold of 97%, which is the optimal threshold for Peakachu's performance in GM12878 cells. For Mustache and Chromosight, we used their default optimal P-value and Pearson parameter settings. In all subsequent experiments, we compared all prediction results across the three chromosomes.

Quantitative analysis

Different tools exhibit both commonalities and distinctions in predicting chromatin loops. As shown in Fig. 5A, we used a KD tree to query overlapping matching regions between each tool and others within a matching radius of one bin. The central region is

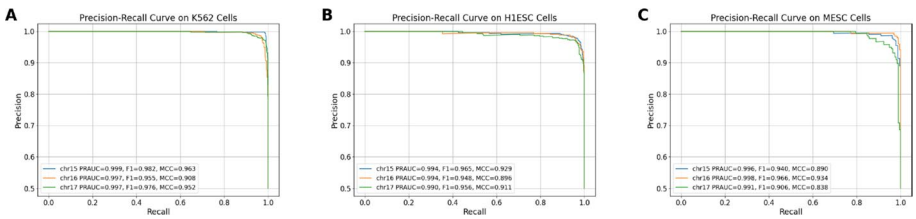

Fig. 4 Cross-cell line and cross-species model testing. **A** Test evaluation on human K562 cells. **B** Test evaluation on human H1ESC cells. **C** Test evaluation on mouse MESC cells

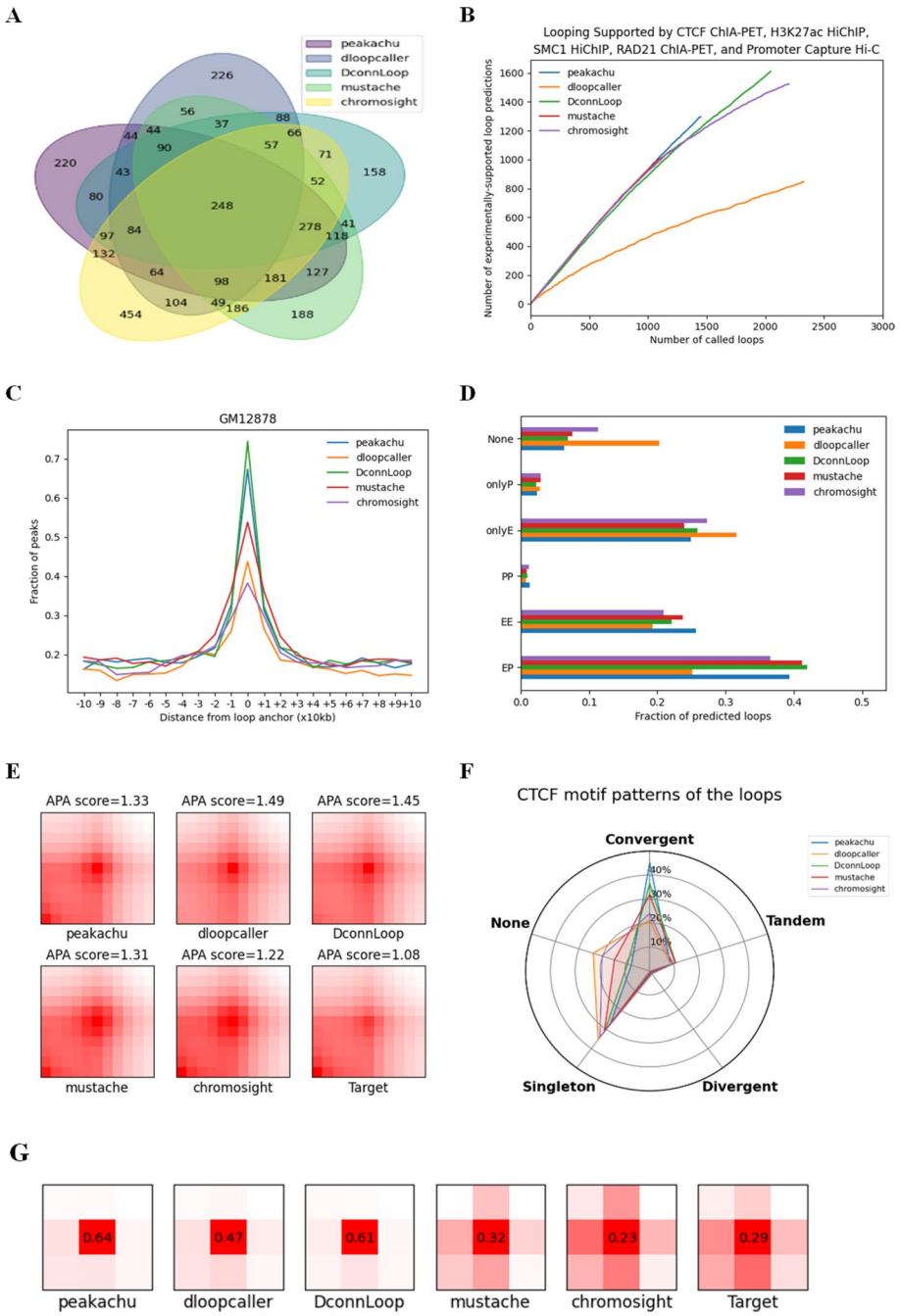

Fig. 5 Comparison of chromatin loop detection by DconnLoop, Peakachu, dloopcaller, Mustache, and Chromosight on GM12878 cells. **A** Venn diagram of loops predicted by different tools. **B** Number of loops supported by ChIA-PET, HiChIP, and Capture Hi-C experimental techniques among all detected results on chromosomes 15, 16, and 17. **C** Enrichment level of ChIP-seq peaks at CTCF binding sites. **D** Enrichment level of regulatory elements. **E** Aggregate peak analysis profiles for target (ChIA-PET and HiChIP identified) and annotated loops. **F** CTCF motif orientation of loops. **G** Local significance analysis of loops. **H**, **I** Distance distribution of loops. **J** Loop radius statistics. **K**, **L** Visualization comparison of detection results from different tools in small regions on chromosomes 15 and 16

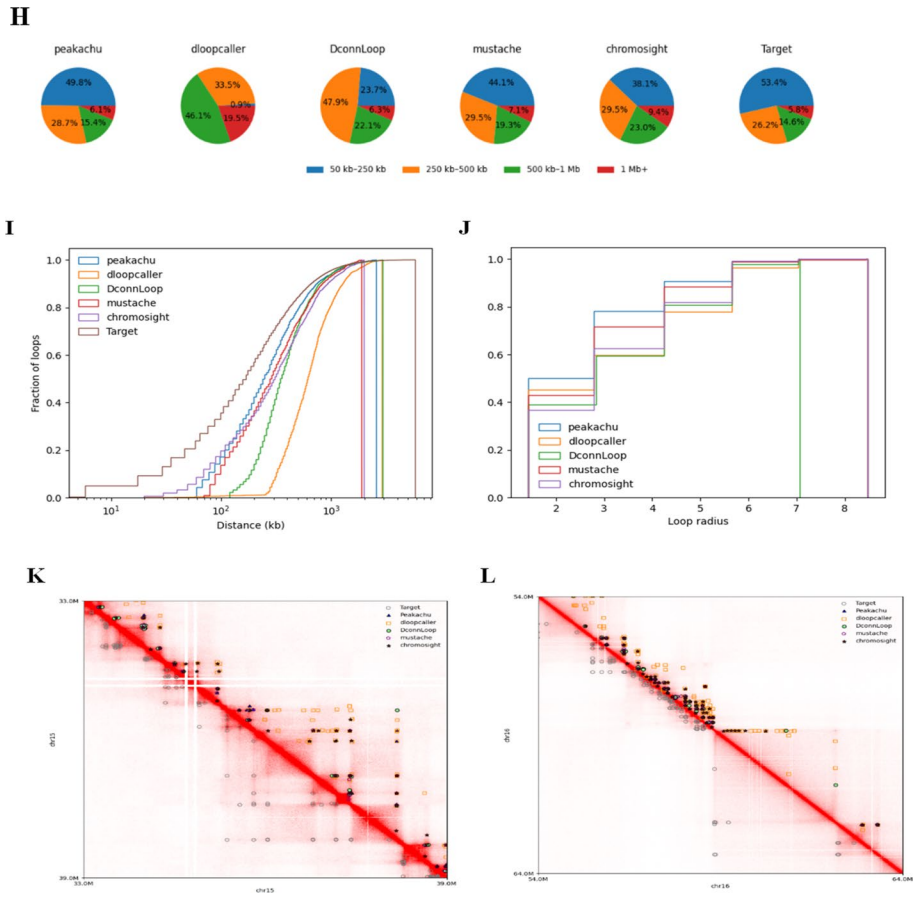

Fig. 5 continued

considered significant across all tools. DconnLoop displayed a larger number of overlaps in multiple regions, indicating a high consistency with predictions from other tools. Additionally, each tool demonstrated unique predictive capabilities, reflecting the distinct focus and strengths of different methodologies in chromatin loop detection.

Enrichment experimental analysis We validated the reliability of the results from all tools by comparing them with target loops identified by various experimental techniques. In Fig. 5B, we combined and de-redundified target loops identified by CTCF ChIA-PET [43], H3K27ac HiChIP [44], SMC1 HiChIP [21], RAD21 ChIA-PET [49], and Promoter Capture Hi-C [50] based on resolution. Using multiple experimental datasets ensured broad coverage of various types of loops. By matching the prediction results with target loops using the RefHiC's [28] experimental method, we found that DconnLoop exhibited the most support for target loops.

CTCF binding site analysis In chromatin interaction maps generated by techniques such as Hi-C, chromatin loops often manifest as peaks, where the loci of the peaks correspond to the anchor points of the loops [3]. In Fig. 5C, we analyzed the peak enrichment at chromatin loop anchors predicted by different tools near CTCF binding sites. All tools exhibited a significant increase in peak enrichment at anchor points, forming

sharp peaks, which indicates that all tools predicted numerous chromatin loop anchors at CTCF binding sites. DconnLoop demonstrated higher peak enrichment near CTCF binding sites, suggesting greater reliability in chromatin loop detection.

A key protein involved in the formation of chromatin loops is the CCCTC-binding factor (CTCF), a highly conserved zinc-finger DNA-binding protein. CTCF recognizes and binds to specific DNA sequences, with a consensus motif typically represented as 5′-CCACNAGGTGGCAG-3′. This sequence is asymmetric and non-palindromic, meaning each CTCF binding site has a specific orientation or polarity along the DNA strand. When considering pairs of CTCF binding sites on the same chromosome, there are four possible orientation configurations: (1) both sites oriented in the same direction on one DNA strand, (2) both oriented in the same direction on the complementary strand, (3) oriented towards each other (convergent), and (4) oriented away from each other (divergent). Empirical studies have demonstrated that chromatin loops predominantly form between CTCF sites in a convergent orientation—that is, the binding motifs face each other [3].

In Fig. 5F, both Peakachu and DconnLoop performed well in detecting convergent CTCF motifs. However, in detecting tandem CTCF motifs, DconnLoop and Mustache had the highest and nearly equal proportions. Peakachu and DconnLoop had lower proportions of anchors without CTCF motifs or with CTCF motifs only at one anchor, mirroring the trend observed with GILoop [30].

Regulatory element enrichment analysis The anchors of loops typically include a known promoter (annotated by ENCODE's ChromHMM) and another known enhancer [51]. In Fig. 5D, DconnLoop identified the highest proportion of enhancer-promoter loop structures, while also having a relatively low proportion of anchors without regulatory elements. This indicates that DconnLoop effectively captures enhancer-promoter interactions during chromatin loop detection.

Furthermore, the functional annotation analysis of predicted loop anchors (Supplementary Figure S7) demonstrates that although the model was trained with CTCF ChIP-seq and ATAC-seq data, it successfully captures diverse regulatory interactions. The IGV browser tracks [52] reveal enrichment of various functional elements at predicted loop anchors, including architectural proteins (CTCF, RAD21, and SMC3), active histone modifications (H3K27ac, H3K4me1, and H3K4me3), and repressive histone modification (H3K27me3). This comprehensive functional enrichment at loop anchors further validates the reliability of our predictions and highlights DconnLoop's ability to identify biologically meaningful chromatin interactions.

Aggregate peak analysis and distance distribution Aggregate Peak Analysis (APA) is a method used to evaluate and compare the overall enrichment of multiple peaks (e.g., chromatin loops) within chromatin structure. This is achieved by comparing the aggregated matrix of the obtained peak set with the aggregated matrix after shifting the peak set towards the bottom left corner, thereby assessing the significance of the aggregated peaks [3]. The analysis window size and the distribution of anchor distances have a significant impact on the peak significance analysis. In the APA analysis shown in Fig. 5E, to accommodate the input specifications of Peakachu, we standardized the aggregation matrix size

to 11 × 11 and performed a significance comparison for predicted loops within the range of 100 kb-1 MB across different tools and target loops (a non-redundant merge of CTCF ChIA-PET and H3K27ac HiChIP). Notably, dloopcaller and DconnLoop exhibited higher significance for their aggregated peaks.

In APA analysis, apart from the window size, the distribution of anchor distances is a crucial influencing factor. As illustrated in Figs. 5H, I, and J (with Fig. 6J using the blob_log method to calculate the loop radius within the local Hi-C contact matrix), dloopcaller and DconnLoop demonstrated fewer short-range interaction loops (< 250 kb), whereas other tools and target loops predominantly consisted of short-range interaction loops located near the diagonal. The Hi-C signals in the diagonal vicinity are often densely packed with significant background noise, making it challenging to distinguish peak significance within the local background when using larger aggregation matrices in APA analysis.

Therefore, in Fig. 5G, we analyzed only the 8 surrounding pixels around the loop neighborhood, evaluating the significance of the center region by assessing the intensity distribution of the central position and its surroundings. We conducted a comprehensive statistical analysis for all tools and target loops. Peakachu and DconnLoop showed a higher frequency of maximum values in the central region, indicating that their predicted loops exhibit higher interaction frequencies at the center, aligning with the criterion of loops as high-pixel points.

Hi-C heat map analysis We visualized the loops predicted by different tools and the target loops in specific chromosomal regions of chromosomes 15 and 16, spanning 33M-39M and 54M-64M, respectively (Figs. 5K and L). Peakachu showed a higher distribution near the diagonal, indicating its sensitivity to high-frequency interaction regions. Both dloopcaller and Chromosight also exhibited higher distributions near the diagonal but with some results scattered, potentially indicating some false detections. DconnLoop's detection results were more concentrated and had a higher overlap with the target loops, suggesting higher detection accuracy. Mustache detected fewer loops, and the results were more dispersed, indicating relatively lower accuracy.

Performance under different sequencing depths To test the impact of different sequencing depths on DconnLoop performance, we employed the FAN-C [53] method to downsample the Hi-C contact matrices of GM12878 cells at 10 kb resolution. We filtered out low-coverage bins and restored the coverage to the original read count. Using Peakachu's depth method, we counted the effective read pairs in both the original and downsampled datasets. The original dataset contained approximately 2000 million (M) effective read pairs, while the downsampled datasets at depths of 90%, 70%, 50%, 20%, and 1.5% contained approximately 1800 M, 1400 M, 1000 M, 400 M, and 30 M effective read pairs, respectively.

We trained and tested the models on these downsampled datasets to evaluate their performance with reduced data volumes. As shown in Figs. 6A, D, G, J, and M, the models trained on datasets with different sequencing depths maintained robust performance on chromosomes 15, 16, and 17, with PRAUC scores consistently above 98% and F1-scores above 91%. The downsampling operation involved random sampling of

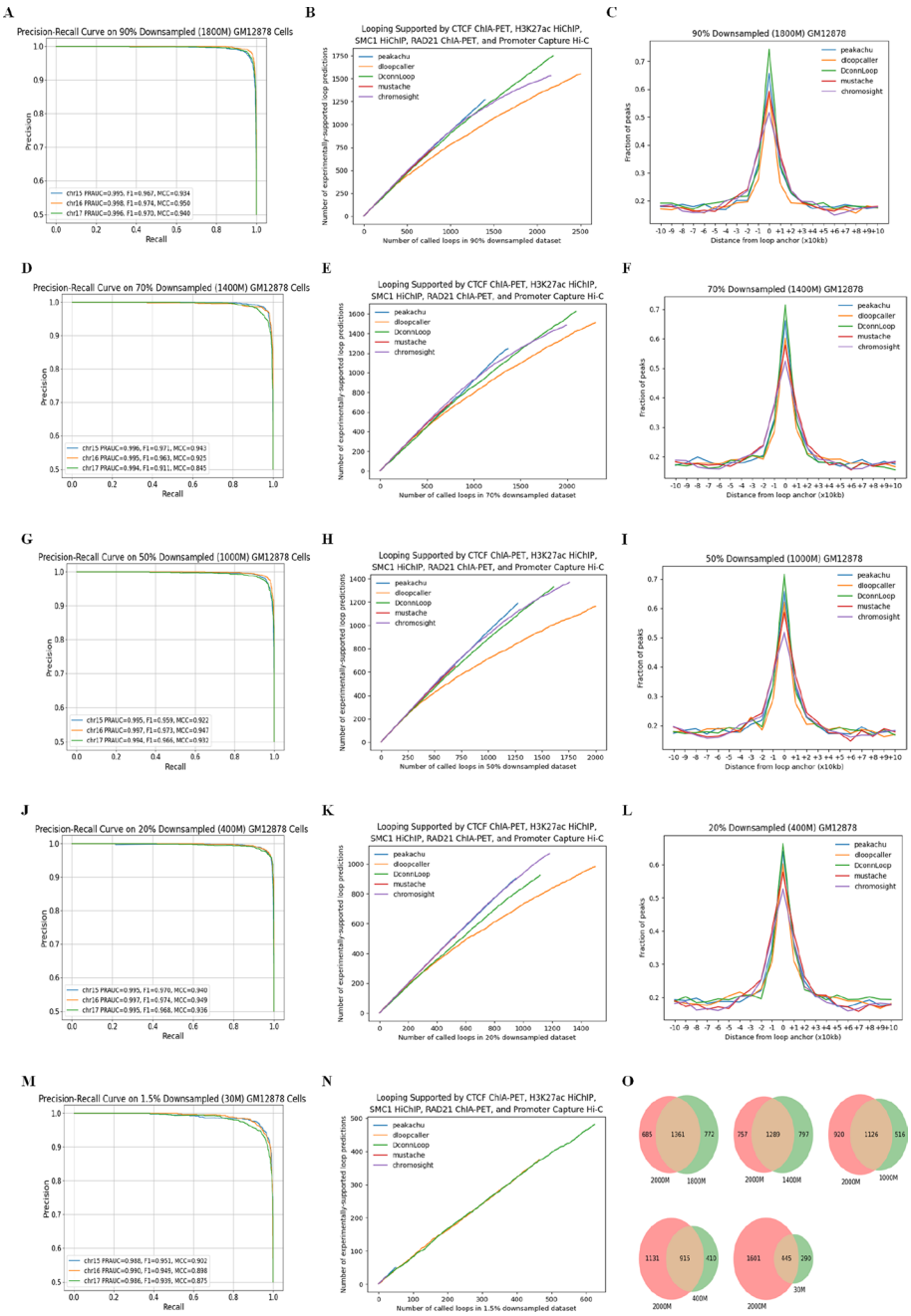

Fig. 6 **A**, **D**, **G**, **J**, **M** Model performance on data with different sequencing depths. **B**, **E**, **H**, **K**, **N** Number of predicted loops matching experimental controls on data with different sequencing depths. **C**, **F**, **I**, **L** Peak content at predicted loop anchors on data with different sequencing depths. **O** Venn diagram of loops predicted by DconnLoop on data with different sequencing depths

read coverage within each bin across the chromosome, which could result in differences in data volume between chromosomes. This might lead to imbalanced training data, slightly affecting model performance. However, selecting an appropriate downsampling rate can reduce noise in the original data, thus enhancing the model's robustness and generalizability.

Next, we utilized different tools to predict chromatin loops on chromosomes 15, 16, and 17 across datasets with varying sequencing depths. Throughout this process, we consistently applied the parameters from the original data. Due to DloopCaller's predictions in the downsampled data containing a substantial number of false positives, we only retained its high-confidence regions. As shown in Figs. 6B, E, H, K, and N, our method demonstrated higher robustness in predictions across the 90%, 70%, and 1.5% downsampled datasets. Notably, in the 1.5% downsampled dataset with only 30 million effective read pairs, we identified the most loops (625), which were highly consistent with experimentally validated loops. To assess the reliability of the predicted loops in the downsampled data, we conducted CTCF peak enrichment analysis at the loop anchors. Due to the low prediction counts (fewer than 100) by most other tools in the 1.5% downsampled dataset, the analysis was performed only on other downsampled datasets. As illustrated in Figs. 6C, F, I, and L, our method's predicted loop anchors exhibited the highest CTCF peak enrichment ratios across different sequencing depths.

In Fig. 6O, we evaluated the overlap between the loops predicted by DconnLoop in the downsampled datasets and those predicted from the original dataset. The results indicated that, for loops predicted at various sequencing depths, there was over 60% overlap with those predicted from the original dataset containing 2000 million effective read pairs. Therefore, even in low-coverage data, the detection of numerous loops does not introduce significant false positives. This demonstrates that DconnLoop maintains high robustness and reliability across different sequencing depths.

To further evaluate DconnLoop's performance in real-world scenarios where models trained on high-depth data might need to predict loops in lower-depth datasets, we conducted additional cross-depth prediction experiments. We trained the model on the original GM12878 dataset and tested its performance on downsampled datasets from both GM12878 and K562 cells (Supplementary Figures S8 and S9). These cross-depth experiments further validated DconnLoop's capability to maintain reliable predictions even when applied to datasets with substantially different sequencing depths, reinforcing its practical utility in diverse research scenarios.

Discussion

In this study, we developed the DconnLoop model by integrating Hi-C, open chromatin, and histone modification data to construct chromatin loop features. The model uses multi-path feature processing and multi-scale integration to improve the accuracy of predicted loops. In the training process, we generated multiple negative samples from a limited set of positive samples and used proportionate training to optimize the binary classification model. For genome-wide predictions, we applied density estimation and density-based clustering to identify the most representative loops among significant candidates. These technical innovations facilitated the model's strong performance across various cell types, species, and sequencing depths.

When compared with other existing tools, our detailed biological feature analysis—including significance testing, anchor peak enrichment, motif pattern analysis, distance distribution characteristics, and regulatory element content evaluation—demonstrated that DconnLoop has certain advantages in predicting biologically relevant chromatin loops.

Given the current limitations of functional genomics data, we aim to incorporate additional data features, such as DNA methylation and RNA expression from transcriptomics. These additional data sources will provide more comprehensive insights into the formation and function of chromatin loops. For instance, DNA methylation patterns can reveal the activity status of gene regulatory regions, while RNA-seq technology can offer a complete gene expression profile, aiding in the understanding of gene expression under different biological conditions. By integrating these data features, we aim to more accurately identify and interpret the biological functions and regulatory mechanisms of chromatin loops.

Conclusion

The DconnLoop model presents a novel approach for constructing and predicting chromatin loops by integrating multi-source genomic data. Our results indicate that the model performs consistently well across various conditions, and it shows advantages in key biological feature analyses compared to other existing tools.

Moving forward, we aim to enhance the model by integrating additional data types, including DNA methylation and transcriptomics, to further improve prediction accuracy and biological interpretability. These data sources will enable us to gain deeper insights into the regulatory mechanisms underlying chromatin loop formation, offering a more comprehensive understanding of their roles in gene regulation.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06092-6.

Supplementary file 1

Author contributions

JFW and KKC participated in the design of the study and the analysis of the experimental results. KKC and JFW performed the implementation. KKC and JWL prepared the tables and figures. CKY and HML summarized the results of the study and checked the format of the manuscript. All authors read and approved the final manuscript.

Funding

This research was supported by the Henan Provincial Department of Science and Technology Research Project (Grant No. 242102210110).

Availability of data and material

The input data used for the experiments can be downloaded from the **supplementary materials** of the paper. Please refer to the **supplementary materials** section for further details on data availability. The source code is available from GitHub at https://github.com/kuikui-C/DconnLoop.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 27 September 2024 Accepted: 19 February 2025
Published online: 01 April 2025

References

1. Bonev B, Cavalli G. Organization and function of the 3D genome. Nat Rev Genet. 2016;17(11):661–78.

2. Lieberman-Aiden E, Van Berkum NL, Williams L, et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science. 2009;326(5950):289–93.
3. Rao SSP, Huntley MH, Durand NC, et al. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell. 2014;159(7):1665–80.
4. Gorkin DU, Leung D, Ren B. The 3D genome in transcriptional regulation and pluripotency. Cell Stem Cell. 2014;14(6):762–75.
5. Forcato M, Nicoletti C, Pal K, et al. Comparison of computational methods for Hi-C data analysis. Nat Methods. 2017;14(7):679–85.
6. Gong H, Yang Y, Zhang S, et al. Application of Hi-C and other omics data analysis in human cancer and cell differentiation research. Comput Struct Biotechnol J. 2021;19:2070–83.
7. Liu L, Han K, Sun H, et al. A comprehensive review of bioinformatics tools for chromatin loop calling. Brief Bioinform. 2023;24(2):bbad072.
8. Tan J, Shenker-Tauris N, Rodriguez-Hernaez J, et al. Cell-type-specific prediction of 3D chromatin organization enables high-throughput in silico genetic screening. Nat Biotechnol. 2023;41(8):1140–50.
9. Buenrostro JD, Wu B, Chang HY, et al. ATAC-seq: a method for assaying chromatin accessibility genome-wide. Curr Protoc Mol Biol. 2015;109(1):1–21.
10. Schmidt D, Wilson MD, Spyrou C, et al. ChIP-seq: using high-throughput sequencing to discover protein–DNA interactions. Methods. 2009;48(3):240–8.
11. Van Berkum NL, Lieberman-Aiden E, Williams L, et al. Hi-C: a method to study the three-dimensional architecture of genomes. JoVE (J Vis Exp). 2010;39:e1869.
12. Sexton T, Cavalli G. The role of chromosome domains in shaping the functional genome. Cell. 2015;160(6):1049–59.
13. Greenwald WW, Chiou J, Yan J, et al. Pancreatic islet chromatin accessibility and conformation reveals distal enhancer networks of type 2 diabetes risk. Nat Commun. 2019;10(1):2078.
14. Ding H, Luo J. MAMnet: detecting and genotyping deletions and insertions based on long reads and a deep learning approach. Brief Bioinform. 2022;23(5):bbac195.
15. Luo J, Ding H, Shen J, et al. BreakNet: detecting deletions using long reads and a deep learning approach. BMC Bioinform. 2021;22:1–13.
16. Gao R, Luo J, Ding H, et al. INSnet: a method for detecting insertions based on deep learning network. BMC Bioinform. 2023;24(1):80.
17. Jeong Y, El-Jaick K, Roessler E, et al. A functional screen for sonic hedgehog regulatory elements across a 1 Mb interval identifies long-range ventral forebrain enhancers. Development. 2006;133:761.
18. Lettice LA, Heaney SJH, Purdie LA, et al. A long-range Shh enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. Hum Mol Genet. 2003;12(14):1725–35.
19. Zhang X, Choi PS, Francis JM, et al. Identification of focally amplified lineage-specific super-enhancers in human epithelial cancers. Nat Genet. 2016;48(2):176–82.
20. Durand NC, Shamim MS, Machol I, et al. Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. Cell Syst. 2016;3(1):95–8.
21. Ay F, Bailey TL, Noble WS. Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts. Genome Res. 2014;24(6):999–1011.
22. Lagler TM, Abnousi A, Hu M, et al. HiC-ACT: improved detection of chromatin interactions from Hi-C data via aggregated Cauchy test. Am J Hum Genet. 2021;108(2):257–68.
23. Wolff J, Backofen R, Grüning B. Loop detection using Hi-C data with HiCExplorer. Gigascience. 2022;11:giac061.
24. Zhang F, Jiao H, Wang Y, et al. InferLoop: leveraging single-cell chromatin accessibility for the signal of chromatin loop. Brief Bioinform. 2023;24(3):bbad166.
25. Matthey-Doret C, Baudry L, Breuer A, et al. Chromosight: a computer vision program for pattern detection in chromosome contact maps. bioRxiv. 2020;14(7):679.
26. Roayaei Ardakany A, Gezer HT, Lonardi S, et al. Mustache: multi-scale detection of chromatin loops from Hi-C and Micro-C maps using scale-space representation. Genome Biol. 2020;21:1–17.
27. Salameh TJ, Wang X, Song F, et al. A supervised learning framework for chromatin loop detection in genome-wide contact maps. Nat Commun. 2020;11(1):3428.
28. Zhang Y, Blanchette M. Reference panel guided topological structure annotation of Hi-C data. Nat Commun. 2022;13(1):7426.
29. Wu H, Zhou B, Zhou H, et al. Be-1DCNN: a neural network model for chromatin loop prediction based on bagging ensemble learning. Brief Funct Genomics. 2023;22(5):475–84.
30. Wang F, Gao T, Lin J, et al. GILoop: robust chromatin loop calling across multiple sequencing depths on Hi-C data. iScience. 2022;25(12):105535.
31. Shen J, Wang Y, Luo J. CD-Loop: a chromatin loop detection method based on the diffusion model. Front Genet. 2024;15:1393406.
32. Tang L, Hill MC, Wang J, et al. Predicting unrecognized enhancer-mediated genome topology by an ensemble machine learning model. Genome Res. 2020;30(12):1835–45.
33. Wang S, Zhang Q, He Y, et al. DLoopCaller: a deep learning approach for predicting genome-wide chromatin loops by integrating accessible chromatin landscapes. PLoS Comput Biol. 2022;18(10):e1010572.
34. Chowdhury HMAM, Boult T, Oluwadare O. Comparative study on chromatin loop callers using Hi-C data reveals their effectiveness. BMC Bioinform. 2024;25(1):123.
35. Banigan EJ, van den Berg AA, Brandão HB, et al. Chromosome organization by one-sided and two-sided loop extrusion. Elife. 2020;9:e53558.
36. Liu S, Cao Y, Cui K, et al. Hi-TrAC reveals division of labor of transcription factors in organizing chromatin loops. Nat Commun. 2022;13(1):6679.
37. Gong H, Li M, Ji M, et al. MINE is a method for detecting spatial density of regulatory chromatin interactions based on a MultI-modal NEtwork. Cell Rep Methods. 2023;3(1):100386.

38. Zhang H, Li F, Jia Y, et al. Characteristic arrangement of nucleosomes is predictive of chromatin interactions at kilobase resolution. Nucleic Acids Res. 2017;45(22):12739–51.
39. Luo Y, Zhang Z. DeLoop: a deep learning model for chromatin loop prediction from sparse ATAC-seq data. bioRxiv. 2023;22:226.
40. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In proceedings of the IEEE conference on computer vision and pattern recognition. 2016; 770–778.
41. Yang Z, Farsiu S. Directional connectivity-based segmentation of medical images. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023; 11525–11535.
42. McInnes L, Healy J, Astels S. hdbscan: hierarchical density based clustering. J Open Source Softw. 2017;2(11):205.
43. Tang Z, Luo OJ, Li X, et al. CTCF-mediated human 3D genome architecture reveals chromatin topology for transcription. Cell. 2015;163(7):1611–27.
44. Mumbach MR, Satpathy AT, Boyle EA, et al. Enhancer connectome in primary human cells identifies target genes of disease-associated DNA elements. Nat Genet. 2017;49(11):1602–12.
45. Zhang P, Wu H. Ichrom-deep: an attention-based deep learning model for identifying chromatin interactions. IEEE J Biomed Health Inf. 2023;27:4559.
46. Loshchilov I, Hutter F. Sgdr: Stochastic gradient descent with warm restarts[J]. arXiv preprint arXiv:1608.03983, 2016.
47. Zhang P, Wu Y, Zhou H, et al. CLNN-loop: a deep learning model to predict CTCF-mediated chromatin loops in the different cell lines and CTCF-binding sites (CBS) pair types. Bioinformatics. 2022;38(19):4497–504.
48. Kai Y, Andricovich J, Zeng Z, et al. Predicting CTCF-mediated chromatin interactions by integrating genomic and epigenomic features. Nat Commun. 2018;9(1):4221.
49. Heidari N, Phanstiel DH, He C, et al. Genome-wide map of regulatory interactions in the human genome. Genome Res. 2014;24(12):1905–17.
50. Cairns J, Freire-Pritchett P, Wingett SW, et al. CHiCAGO: robust detection of DNA looping interactions in Capture Hi-C data. Genome Biol. 2016;17:1–17.
51. Hoffman MM, Ernst J, Wilder SP, et al. Integrative annotation of chromatin elements from ENCODE data. Nucleic Acids Res. 2013;41(2):827–41.
52. Robinson JT, Thorvaldsdóttir H, Turner D, et al. igv. js: an embeddable JavaScript implementation of the integrative genomics viewer (IGV). Bioinformatics. 2023;39(1):830.
53. Kruse K, Hug CB, Vaquerizas JM. FAN-C: a feature-rich framework for the analysis and visualisation of chromosome conformation capture data. Genome Biol. 2020;21:1–19.

Publisher's Note