Supplementary Materials

### The supplementary table of accession number list

**Table S1**. Accession number list of cell lines used in experiments.

|  |  |
| --- | --- |
| Cell line | ChIA-PET read1/read2 |
| A549 | ENCFF985XQV/ENCFF994RVV |
| A673 | ENCFF967ILW/ENCFF790CGR |
| AG04449 | ENCFF868WGL/ENCFF430KRH |
| AG04450 | ENCFF581VKF/ENCFF387IFJ |
| BJ | ENCFF634VYO/ENCFF902YQJ |
| Caco-2 | ENCFF544SMM/ENCFF109BXE |
| GM10248 | ENCFF803DOQ/ENCFF149FYK |
| GM12878 | ENCFF409UJV/ENCFF810FRZ |
| HCT116 | ENCFF869HLZ/ENCFF933EOS |
| HepG2 | ENCFF285PTY/ENCFF042GDX |
| IMR-90 | ENCFF932VWG/ENCFF908YMO |
| K562 | ENCFF915YCD/ENCFF245TLN |
| MCF-7 | ENCFF592KNN/ENCFF342KQM |
| MCF-10A | ENCFF520GGF/ENCFF290IDP |
| OCI-LY7 | ENCFF593CIG/ENCFF261EWD |
| Panc1 | ENCFF331HZQ/ENCFF937UXS |
| PC-3 | ENCFF131SQE/ENCFF392ZNZ |
| PC-9 | ENCFF781PRX/ENCFF615NPK |
| RWPE1 | ENCFF096ZQA/ENCFF422NCN |
| SK-N-SH | ENCFF084YMC/ENCFF440ERH |

### The supplementary table of accession number list

**Table S2**. Accession number list of cell lines used in experiments.

|  |  |  |  |
| --- | --- | --- | --- |
| Cell line | ChIA-PET read1/read2 | ChIP-seq accession number | DNase accession number |
| AG04450 | ENCFF581VKF/ENCFF387IFJ | ENCFF873FBD | ENCFF816RDX |
| BJ | ENCFF634VYO/ENCFF902YQJ | ENCFF518RUC | ENCFF812QOW |
| Caco-2 | ENCFF544SMM/ENCFF109BXE | ENCFF378WSO | ENCFF579UXQ |
| GM12878 | ENCFF409UJV/ENCFF810FRZ | ENCFF796WRU | ENCFF759OLD |
| HCT116 | ENCFF869HLZ/ENCFF933EOS | ENCFF463FGL | ENCFF240LRP |
| IMR-90 | ENCFF932VWG/ENCFF908YMO | ENCFF203SRF | ENCFF221SZF |
| K562 | ENCFF915YCD/ENCFF245TLN | ENCFF221SKA | ENCFF274YGF |
| MCF-7 | ENCFF592KNN/ENCFF342KQM | ENCFF278FNP | ENCFF522NDW |

### Hyperparameter settings

In large kernel fire block of anchor score model, the receptive fields of the three branches are configured as 1, 3, and 13. In the branch with a receptive field of 3, the convolutional kernel size is set to 3. The branch with a receptive field of 13 uses dilated convolution with a kernel size of 5 and a dilation factor of 3, in place of standard convolution with a kernel size of 13. For the branch with a kernel size of 3, a combination of 1\*3 and 3\*1 convolution layers are utilized instead of a 3\*3 convolution layer to reduce trainable parameters. Similarly, for the branch with a kernel size of 5, a combination of 1\*5 and 5\*1 convolution layers are employed to reduce trainable parameters while preserving feature extraction capability.

The input channel numbers for the three large kernel fire blocks of anchor score model are 64, 128, and 256, and channel transformations are realized through two scale blocks, transitioning from 64 to 128 and from 128 to 256. To monitor potential overfitting, an early stopping callback function based on validation loss is implemented, with the patience parameter set to 40. This implies that training will halt if the validation loss fails to exhibit a significant decrease over 40 consecutive epochs. The initial learning rate is set at 0.0001, complemented by the CosineAnnealingLR as the learning rate scheduler function. This scheduler function automatically adjusts the learning rate periodically based on training epochs. Regularization techniques are employed to enhance model performance, including the use of Batch Normalization to reduce the risk of overfitting and accelerate training. The application of these techniques allows the model to maintain simplicity and ease of use while achieving higher performance and improved generalization capability. The anchor score model contains approximately 310,000 trainable parameters, well within the range suitable for training on a single GPU. Conse-quently, during experimentation, the anchor score model is trained efficiently on a NVIDIA GeForce RTX 3060 12G GPU.

In large kernel fire block of OCR score model, the receptive fields of the three branches are configured as 1, 3, and 13. The input channel numbers for the three large kernel fire blocks of OCR score model are 64, 128, and 256, and channel transformations are realized through two scale blocks, transitioning from 64 to 128 and from 128 to 256. The OCR score model contains approximately 630,000 trainable parameters, making it suitable for training on a single GPU. In the experiments, the OCR score model is trained using an NVIDIA GeForce RTX 3060 12G GPU.

The models are implemented in a Python 3.8 environment, utilizing PyTorch 1.12 as the backend framework.

### Detailed configuration of the comparison method

The CTCF-MP method uses three features—sequence features, functional genomic features, and a DNA sequence embedding tensor—as inputs for XGBoost (eXtreme Gradient Boosting) to predict chromatin loops. Sequence features include basic sequence-related features, such as loop length and CTCF motif orientation patterns. To ensure a fair comparison, functional genomic features are consistently set as DNase hypersensitive sites' signal values in this experiment. The embedded DNA sequences represent sequence features learned by a deep learning model and are represented as tensors. The DeepCTCFLoop method utilizes only the DNA sequence as the input feature and employs a deep learning model to predict chromatin loops. The DeepLUCIA method uses both DNA sequence and functional genomic features as inputs, employing a neural network model for predicting chromatin loops. The CharID method uses both sequence and functional genomic features as inputs, which are fed into a gradient boosting tree classifier for predicting chromatin loops. The Lollipop method also uses sequence and functional genomic features, employing a random forest classifier for predicting chromatin loops.