The authors have successfully addressed my comments. I recommend the article for publication.

I had suggested to the authors to do a careful copy-edit. There were some small edits that I still recommend, which are in bold in the text below.

==========

Introduction

The high-throughput chromosome conformation capture (Hi-C) technique [1] is a genome-wide technique used to investigate three-dimensional (3D) chromatin conformation inside the nucleus. It has facilitated the identification and characterization of multiple structural elements, such as the A/B **compartments** [1], topological associating domains (TADs) [2, 3], enhancer-promoter **interactions** [4] and stripes [5] over recent decades. In practice, Hi-C data is conventionally stored as a pairwise read count matrix $Mn \times n$, where Mij is the number of observed interactions (read-pair count) between genomic regions i and j, and the genome is partitioned into n fixed-size bins (e.g., 25 kb). Bin size (i.e., resolution), is a crucial parameter for Hi-C data analysis, as it directly affects the results of downstream analysis, such as predictions of enhancer-promoter interactions [6-11] or identification of TAD boundaries [6, 12-16]. Depending on sequencing depths, the size of commonly used bins ranges from 1 kb to 1 Mb. Because of the high cost of sequencing, most available Hi-C datasets have relatively low resolution, such as 25 kb or 40 kb [17]. Sequencing high-resolution Hi-C matrices demands sufficient sequencing coverage; otherwise, the contact matrix would be extremely sparse and contain excessive stochastic noise. When sequencing Hi-C data, billions of read-pairs are typically necessary to achieve truly genome-scale coverage at kilobase-pair resolution [18], and the cost of Hi-C experiments generally scales quadratically with the desired level of resolution [19]. Low-resolution data may be sufficient for detecting large-scale genomic patterns such as A/B compartments, but the decrease in resolution when analyzing Hi-C data may prevent identification of fine-scale genomic elements such as sub-TADs [20, 21] and enhancer-promoter interactions, **and** even lead to inconsistent results when detecting interactions and TADs in replicated samples [22]. Therefore, developing a computational model to impute a higher-resolution Hi-C contact matrix from currently available Hi-C datasets show its potency and usefulness.

Several pioneering works on solving problems related to low-resolution Hi-C data have recently emerged. Li et al. proposed deDoc for detecting megabase-size TAD-like domains in ultra-low resolution Hi-C data [23]. Zhang et al. proposed a deep learning model called HiCPlus to enhance Hi-C matrices from low-resolution Hi-C data [17]. HiCPlus showed that chromatin interactions can be predicted from their neighboring regions, by using the convolutional neural network (CNN) [24]. Carron et al. proposed a computational method called Boost-HiC for boosting reads counts of long- range contacts [25]. And Liu et al. proposed HiCNN [26] which is a 54-layer CNN and achieved better performance than HiCPlus. While these results were encouraging, three problems still exist in Hi-C data resolution enhancement algorithms. First,

Hi-C data contain numerous high-frequency details (Mij and its nearby values are very large, while values in neighboring regions are small) and sharp edges, which are usually considered to indicate the presence of enhancer-promoter loops, stripes, and TAD boundaries. Models **that rely** on regression and mean squared error (MSE) loss, which is thought to yield solutions with overly smooth textures [27], are likely to smooth these features. Thus, we seek to develop a model which is capable of predicting data with a sharp or degenerated distribution. Second, the structural patterns and textures of Hi-C data are abundant. The hypothesis

space, which is controlled by the number of parameters, should be able to capture richer structures as it grows [28]. It is possible that increasing the depth of network would increase accuracy [29], while ensuring the model's generalizability and restraining the overfitting problem. The final critical problem is the stochastic noise in Hi-C data. An effective model should be able to predict solutions resides on the manifold of target data and thus diminish stochastic noise (i.e., capability for denoising) [30, 31].

In order to make accurate prediction of high-resolution Hi-C data from low-coverage sequencing samples against these three problems. We developed a deep learning model which employed the state-of-the-art generative adversarial network (GAN), in combination with some advanced techniques in deep learning field. Goodfellow et al. first introduced the GAN model for estimating generative models with an adversarial process [32]. The GAN architecture allows the generative net to easily learn target data distribution, even sharp or degenerated distribution. GAN has been used for various applications and is showing its huge potency. For instance, Mirza et al. proposed the conditional GAN (cGAN) of which the generator learns the data distribution upon conditional inputs [33]. Li and Wand described the usage of GANs to learn a mapping from one manifold to another [34]. Another inspiring work for us was described by Ledig et al. [35], who proposed SRGAN to generate photo-realistic super-resolution images. Besides, He et al. introduced the concept of residual learning and proved that an ultra-deep neural network could be easily trained via residual learning and achieve superior performance [36]. Also, researchers started to design task-specified loss functions, using not only MSE loss (i.e., L2 loss) but other losses like perceptual loss [37] as well, and gain surprising advancements [38].

In this paper, we propose a GAN-based method DeepHiC to enhance the resolution of Hi-C data. Using low-coverage Hi-C matrices (obtained by downsampling original Hi-C reads) as input, we demonstrate that DeepHiC is capable of reproducing high-resolution Hi-C matrices. DeepHiC-enhanced data achieve high correlation and structure similarity index (SSIM) compared with original high-resolution Hi-C matrices. And even using as few as 1% original reads, while no previous methods enhancing data of this depth, DeepHiC is still capable of inferring high-resolution data and achieves the correlation and SSIM score as good as the real high-resolution replicated assay. Compared with previous methods, our method is more accurate in predicting high-resolution Hi-C data, even in fine-grained details, and performed better when **applied** to different cell lines. Enhancements of DeepHiC improve the accuracy of downstream analysis such as identification of chromatin loops and detection of TADs. In this study, we applied DeepHiC to Hi-C data in mouse embryonic development and demonstrated that, compared with the original low-coverage Hi-C data, DeepHiC-enhanced Hi-C data enables

the identification for chromatin loops that are similar to those identified in deeply sequenced Hi-C data. Besides, we also develop a web-based tool (DeepHiC, http://sysomics.com/deephic) that allows researchers to enhance their own Hi-C data with just a few clicks. In summary, this work introduces an effective model for enhancing Hi-C data resolution and establishes a new framework for prediction of a high-resolution Hi-C matrix from low-coverage data.

Parameters training of DeepHiC model

In **the** current study, we propose a conditional generative adversarial network (cGAN), DeepHiC, for enhancing Hi-C data from low-resolution samples. It contains a generative network G and a discriminative network D. The former takes low-resolution data as **input** and imputes the enhanced **output**, while the latter is only employed during training process as a discriminator for reporting the differences between enhanced outputs and real high-resolution Hi-C data to the network G, which form the adversarial training (Fig 1a). Also, in order to alleviate the overly-smooth problem caused by MSE loss, we utilized the perceptual loss to capture structure features in Hi-C contact maps and the total variation (TV) loss for suppressing artifacts [39]. The detailed architecture of DeepHiC is depicted in S1 Fig. The GAN framework benefits G network by efficiently capturing the distribution of target data (even very sharp or degenerate distributions) [32] and favors solutions **residing** on the manifold of target data.
We trained DeepHiC on chromosomes 1-14 and tested on chromosomes 15-22 in the GM12878 cell line dataset during the training process (see Materials and Methods: Implementation of DeepHiC and evaluation). For low-resolution (low-coverage) data with different downsampling ratios, we obtained their corresponding trained models separately. We evaluated the structure similarity index (SSIM) scores in the test set during training. Higher SSIM scores between enhanced output and real high-resolution Hi-C indicate greater structural similarity. For low-resolution data from different downsampling ratios, SSIM scores in the test set increased gradually and converged when DeepHiC was trained in 200 epochs (S2 Fig), as well as another metric related to MSE (S3 Fig). Generator loss in both the training and test sets decreased simultaneously during the training process (S4 Fig). These results indicate that the model converged successfully in training without overfitting. Furthermore, we tested various splits of training and test sets, like a 5-fold cross validation. Performances in **the** test set were consistent across different dataset splits, showing that our model is capable of capturing common information from the different training sets and its parameters could be stably derived with no relation to training/test set we used (S5 Fig). We also trained the generator net as a regression model without the adversarial part, but SSIM scores in the test set vibrated substantially (S6 Fig). These results suggest that the GAN-based framework efficiently restrains the over-fitting phenomenon and its necessity for prediction. Besides, DeepHiC is also could be trained in IMR90 or K562 dataset (S7 Fig).
In **the** prediction step, we divided the large Hi-C matrix into small squares as model inputs. For **a** fair comparison in **the** following analysis, we divided the low-resolution Hi-C matrix into 0.4 Mb × 0.4 Mb sub- regions (40 × 40 bins in 10-kb resolution) same with what HiCPlus does. Then the completed enhanced Hi-C matrix could be obtained by reconstructing all enhanced sub-regions after prediction (Fig 1b) (see Methods: dividing and reconstructing matrices).

DeepHiC reproduces high-resolution Hi-C from as few as 1% downsampled reads

We used the high-resolution Hi-C data in the GM12878, K562 and IMR90 cell lines from Rao's Hi-C (access code GSE63525) in our experiments. Datasets pertaining to different cell types are denoted as GM12878, GM12878R, K562, and IMR90 for convenience (GM12878R represents the replicated assay in the GM12878 cell line). First, we constructed high-resolution (10-kb) contact matrices using all the reads from the raw data. Then we downsampled the reads to different ratios (ranges from 1:10 to 1:100) of the original reads to simulate the low-resolution Hi-C data. We also constructed contact matrices at the same bin size. Therefore, we obtained paired high-resolution and low-coverage Hi-C data (both were binned at 10-kb). The original experimental high-resolution data were regarded as ground truth in the following analysis, while the low-coverage data were enhanced by DeepHiC using the trained model.
Fig 1c shows the model's enhancements in a 1Mb sub-region (100 bins) on chromosome 22 in the test set (GM12878 cell line). Comparing with the real 10-kb Hi-C data, DeepHiC-enhanced matrices recover patterns such as chromatin loops and TADs successfully from low-coverage inputs. Quantitatively, DeepHiC-enhanced data achieve the correlations as good as the experimental replicate (i.e., GM12878R), even **though** they were predicted from 1% downsampled data (Fig 1d). It also shows the same result in SSIM measure (S8 Fig). These results indicate that the DeepHiC model is capable of reproducing high-resolution Hi-C data with high similarity even using 1% downsampled reads. Because the high-resolution data we used is at 10-kb resolution, it implies that our method could enhance 1Mb resolution Hi-C data to 10-kb resolution with high quality. And there is no available imputation algorithm for enhancing Hi-C data from such a sequencing depth before.
In the following analyses, we trained DeepHiC in 1/16 downsampled data for fairly comparing with other baseline methods such as HiCPlus, Boost-HiC, and HiCNN (see Methods). The trained model we used was trained on data of chromosome 1-14 in GM12878 dataset. SSIM scores converged at 0.9 in remaining chromosomes (S9 Fig). For remaining chromosomes' data in GM12878 dataset, as well as the whole GM12878R, K562, and IMR90 datasets, we applied the trained model to their downsampled data, then evaluated the performance with taking the real high-resolution data as the ground truth.

Enhancements of low-resolution data

We first investigate the enhancements afforded by DeepHiC by visualizing data in the form of heatmaps (see S1 Note for colorbar settings). Fig 2a shows three 1-Mb-width sub-regions (arranged by rows) on chromosomes 16, 17, and 22 which **were** extracted from the test set in the GM12878 dataset. The real high-resolution examples marked as "Original" in the first column contain clear individual chromatin loops and TAD structures, while low-coverage examples marked as "Downsampled" (second column) have abundant noise and less clear TAD structures. We found that DeepHiC-enhanced data (last column) could accurately restore the patterns and textures which are exactly **the** same as those in real high-resolution data. Baseline

models' results are shown in the third to fifth columns. Noting that Boost-HiC was specifically developed for enhancing long-range contacts [25]. So, it makes sense that Boost-HiC **has** slight changes in short-range contacts (third column). The HiCPlus-enhanced data marked as "HiCPlus" (fifth column) contains much less noise and more visible TAD structures, but refined structures such as chromatin loops are replaced by smooth textures. So does the HiCNN (fifth column), which is a deeper CNN and relies on MSE loss as well. **In terms of fine-grained details, we** scrutinized smaller 0.3 Mb × 0.3 Mb (30 × 30 bins) sub-regions from these three examples in real high-resolution Hi-C and DeepHiC-enhanced Hi-C, as illustrated in Fig 2b. High similarity between experimental high-resolution data and DeepHiC-enhanced data was observed. Sharp edges in heatmaps, which are deemed difficult to recover in practice, were accurately recovered by DeepHiC. We also visualized three sub-regions from the GM12878R dataset (S10 Fig), three sub-regions from the K562 dataset (S11 Fig), and three sub-regions from the IMR90 dataset (S12 Fig). And DeepHiC outperforms baseline models in all four datasets. The SSIM scores for downsampled, HiCPlus-enhanced, HiCNN-enhanced, and DeepHiC-enhanced data, as compared with real high-resolution data for these three sub-regions were 0.20, 0.64, 0.59, and 0.89 on average, respectively.

DeepHiC outperformed other methods in terms of genome-wide similarity **to ground truth deeply sequenced data**

Furthermore, we quantitatively investigated genome-wide performance for all four datasets. We calculated SSIM scores for downsampled and various model-enhanced data, as compared with real high-resolution data for all 1 Mb × 1 Mb (100 × 100 bins) sub-regions with non-overlap at the diagonal across the entire genome (S13 Fig). Fig 3a shows that DeepHiC-enhanced matrices had the highest SSIM scores for all 23 chromosomes in the GM12878 dataset. Average values for downsampled, HiCPlus-enhanced, HiCNN-enhanced, and DeepHiC-enhanced data were 0.15, 0.71, 0.66, and 0.89, respectively. SSIM scores derived from DeepHiC, HiCPlus, and HiCNN are denoted as $SSIM_{deep}$, $SSIM_{icplus}$ and $SSIM_{icnn}$, respectively. Fig 3b shows the differences between these scores for all 4 datasets covering all chromosomes. Their absolute values are shown in S14 Fig. The comparison results show that DeepHiC achieves greater similarity than HiCPlus and HiCNN.
We also computed the Pearson correlation coefficients between the experimental high-resolution, downsampled, baselines-enhanced, and DeepHiC-enhanced matrices at each genomic distance, which also performed in previous studies. As shown in Fig 3c, the DeepHiC-enhanced matrices obtained higher correlation coefficients (~5%) than the HiCPlus-enhanced matrices at all genomic distances of interest from 50 kb to 1 Mb. This region included proximal and distal regions. We also computed the differences between correlations derived from DeepHiC with those derived from HiCPlus/HiCNN, which are denoted as r_{deep} and r_{icplus} / r_{icnn}, respectively. Then we investigated the distribution of differences in all four datasets by boxplots, with extremely small p-values obtained for that r_{deep} are significantly higher than r_{icplus} / r_{icnn} (paired t-test, pair number = 96), as shown in Fig 3d. Their absolute values are shown in S15 Fig. The results of similarity and correlation comparison revealed our model's advantages in restoring high-resolution Hi-C. More importantly, advantages across

various cell lines revealed that DeepHiC can be used to enhance the Hi-C matrix for other cell types.
We omitted comparison with Boost-HiC considering that it aims to enhance long-range contacts. Evaluation of Boost-HiC is plotted in S14 Fig and S15 Fig. Besides, we also investigated the performance of detecting A/B compartments for DeepHiC and Boost-HiC, because the latter is reported for it. S16 Fig shows our model achieves comparative performance in detecting A/B compartments, considering our model is trained using short-range contacts.
Besides, we applied DeepHiC to data from various downsampled ratios (e.g., 1/25, 1/36), while still using the trained model derived from 1/16 downsampled data. S17 Fig shows that DeepHiC still achieves greater correlation coefficients. These results suggest that DeepHiC could be employed to enhance low-coverage sequencing data, rather than just enhancing data with a particular ratio. Thus, we used the same downsampling and predicting procedure to make predict on more cell types' data (including mouse cell line CH12-LX) from Rao et al [4], as shown in S18 Fig. And correlations across cell types suggest that our model also preserve the specificities between cell lines (S18 Fig. d). Further, performances on Hi-C data prepared using 6-cutter enzyme revealed that our model is also applicable to 6-cutter enzyme prepared Hi-C data (S19 Fig).

Significant interactions in high-resolution Hi-C were accurately recovered from DeepHiC-enhanced matrices

After demonstrating that DeepHiC can restore high-resolution Hi-C from low-resolution data, we investigated whether these enhanced high-resolution matrices could facilitate the identification of significant chromatin interactions. For this purpose, we used Fit-Hi-C software to obtain significant intra-chromosomal interactions. We applied Fit-Hi-C to Hi-C data present above, in four datasets, using the same parameters (Methods). Statistical confidence values (i.e., q-values) for all loci-pairs were acquired by Fit-Hi-C. We kept the predicted significant interactions (q-value < 1×10^{-6}) for genomic distances from 20 kb to 1 Mb for further comparative analysis. At first, we visualized three 1 Mb-wide sub-regions. Significant interactions are presented in yellow in the upper triangles of heatmaps (Fig 4a). Compared with the real high-resolution data, only DeepHiC-enhanced matrices yield consistent results in recognizing significant interactions. And the yellow-marked anchors are indeed significant interactions by observing the lower triangular parts of heatmaps. The numbers of interactions in these three sub- regions (denoted as I, II and III) derived from various contact matrices are presented in S20 Fig. HiCNN and HiCPlus-enhanced matrices identified few loci-pairs, while the experimental and DeepHiC-enhanced matrices identified about 40 loci-pairs, respectively. Fig 4a presents the significant interactions identified in real high-resolution Hi-C gathered in 8, 20, and 11 clusters, respectively. However, for low-resolution Hi-C, few interactions were identified. For HiCPlus-enhanced Hi-C, only six clusters were recovered. Surprisingly, DeepHiC-enhanced Hi-C recovered nearly all clusters (35 in total) and no false-positive cluster was added. Because Fit-Hi-C calculated the significance of all loci-pairs within the genomic distance of interest, we performed a genome-wide comparative analysis by analyzing the significance

matrices formed with q-values. We calculated the similarity of significance matrices, as previously performed for Hi-C matrices. Fig 4b shows the Pearson correlation coefficients for significance matrices in the GM12878 dataset at each genomic distance. Same results of comparisons between the other three datasets are presented in S21 Fig. We observed that q-values derived from DeepHiC-enhanced data were more similar to the real high-resolution data than any others for the entire dataset. We also compared the overlap of identified interactions with real high-resolution data at each genomic distance, as shown in Fig 4c. The Jaccard index (JI) of identified interactions between DeepHiC-enhanced data and real high-resolution data was higher at each genomic distance. In addition to using the aforementioned threshold for q-values, we tried more thresholds by scanning various false discovery rates (FDR), ranging from 0.001 to 0.05, with step size of 0.001. We evaluated the overlap of identified interactions according to FDR scanning. We found that DeepHiC outperformed others (Fig 4d). These results suggested that DeepHiC-enhanced Hi-C data are more accurate in predicting chromatin loops and yield less artifact noise.
Next, we compared the **significant interactions** identified in these Hi-C matrices with the identified **interactions** by CTCF chromatin interaction analysis by paired-end tagging sequencing (ChIA- PET) in the K562 cell line, **for** which related data is available in the ENCODE project. ROC analysis was performed **in the** same **way as is described** in HiCPlus, **using** the identified CTCF-mediated chromatin **interactions** from ChIA-PET as true positives. As for negatives, we randomly selected **the** same number of loci pairs that were not predicted to be interacting pairs by ChIA-PET and **that had the** same **distance** distribution with positives (10 repeats). We then plotted the ROC (receiver operating characteristic) curve and calculated the area under the ROC curve (AUC) for each. As shown in Fig 4e, CTCF interacting pairs and non-interacting pairs were separated from the DeepHiC-enhanced matrix in the predicted results (average AUC = 0.825). We also observed that the AUC score for the DeepHiC-enhanced matrix was significantly higher than both the AUC derived from the HiCPlus/HiCNN-enhanced matrix (p-value = 0, paired t-test) as well as the AUC derived from the downsampled matrix (p-value = 0, paired t-test).

DeepHiC is more precise in detecting TAD boundaries

The detection of TADs is not as sensitive to resolution decline as algorithms for detecting TADs, we obtained roughly the same results when using the Hi-C data with various downsampling ratios [23]. However, we found that some refined TAD structures were shifted-even wrongly detected-in low- resolution data. Therefore, we continually assessed the performance of DeepHiC in recovering TADs, especially in fine-scale TADs. We calculated the Δ score of insulation scores across the entire genome for all four datasets (Methods). The zero-points within monotonic rising intervals are considered to be TAD boundaries. Fig 5a illustrates the insulation Δ scores derived from experimental high-resolution, downsampled, HiCPlus/BoostHiC/HiCNN-enhanced, and DeepHiC-enhanced Hi-C matrices, on chromosome 22, in the region between 20-22.7 Mb, from the GM12878 dataset. The trends seemed similar, but enlarged views around the zero-points revealed that DeepHiC obtained the closest location of zero-points, while downsampled Hi-C and HiCPlus-enhanced Hi-C had bias of 20-50 kb. The

Pearson correlation coefficients between Δ scores derived from experimental Hi-C and those derived from non-experimental Hi-C were 0.937, 0.953, and 0.992 for downsampled, HiCPlus-enhanced, and DeepHiC-enhanced data, respectively.
As for the two segmentations formed by TAD boundaries, we calculated all split points' distances and all intervals' overlap with another segmentation (see Methods), then **investigated** the properties of the resulting arrays. As shown in Fig 5b, we illustrated the distribution of all boundaries' distances from $Sdown$, $Sboost$hic, Shicplus, Shicnn, and $Sdeep$hic to $Sorigin$ in the GM12878 dataset via box plot. Boundary segmentations were derived from corresponding data. The distances of DeepHiC- enhanced data were significantly smaller than those of Boost-HiC-enhanced data (p-value = 1.4 ×10−40,, Wilcoxon rank-sum test), those of HiCPlus-enhanced data (p-value = 7.1 × 10−14, Wilcoxon rank-sum test), those of HiCNN-enhanced data (p-value = 0.035, Wilcoxon rank-sum test) and those of downsampled data (p-value = 1.3 × 10−193, Wilcoxon rank-sum test). We also investigated the distribution of the overlap of segmentations vs. experimental high-resolution data (Fig 5c). The results showed that our model had a high proportion of high JI (p-value < 1 × 10−20 for downsampled/BoostHiC-enhanced/HiCPlus-enhanced data, < 0.001 for HiCNN-enhanced data, Mann Whitney U-test), which indicates that more TADs are precisely matched with those in real Hi-C data. Same results of comparisons for other cell types are illustrated in S22 Fig.

DeepHiC **enhances** prediction of chromatin loops in mouse early embryonic developmental stages

DeepHiC can be used to enhance the resolution of existing time-resolved Hi-C data obtained through early embryonic growth. These data are prone to low resolution due to limited cell population (40-kb in [40]). Therefore, algorithms for detecting significant interactions, when applied to these data, may produce results with a relatively high false positive rate. We demonstrate that DeepHiC can be applied to Hi-C data of mouse early embryonic development to enable identification of significant chromatin interactions with a considerably lower false positive rate. We applied Fit-Hi-C to both original low-resolution Hi-C contact matrices and DeepHiC-enhanced contact matrices (Fig 6a) and kept pairs of loci with q-values lower than a preset cut-off (0.5 percentile) as significant interactions (predicted loops). Chromatin loops regulate spatial enhancer-promoter contacts and are relevant to domain formation [4, 41], and anchors of Fit-Hi-C predicted significant interactions co-localize with open chromatin regions including insulators, enhancers, and promoters. In deeply sequenced Hi-C data of GM12878 cell line, significant interactions identified by Fit-Hi-C are significantly enriched in gene promoter and open chromatin regions compared to shuffled control (S23 Fig). Therefore, we evaluate the similarity of Fit-Hi-C significant interactions identified on mouse embryonic development Hi-C data to those identified in high-resolution Hi-C data according to the fraction of all significant interactions that connect promoter regions, as well as by the fraction connecting two accessible chromatin regions marked by ATAC-seq peaks. As shown in Fig 6b, significant interactions identified using DeepHiC enhanced Hi-C data are more likely to anchor at gene promoters than those identified using original Hi-C data. They are also more likely to co-localize with open chromatin regions at both of their anchoring loci than those predicted with original Hi-C data (Fig

6c). We mainly focused on the 8-cell stage and beyond because Hi-C data from earlier stages only demonstrate weak TADs and depleted distal chromatin interactions [40]. To generate control datasets, we randomly repositioned all predicted significant interactions for original Hi-C data, while maintaining the distance between anchors of each significant interaction, using the "shuffle" command in Bedtools [42]. We repeated this process 20 times to generate 20 random significant interaction datasets. We found that the fraction of predicted significant interactions that connected accessible loci was significantly higher for DeepHiC-enhanced Hi-C data, compared with random control data. Using an example at chromosome 5, we showed that significant interactions predicted using original Hi-C data were highly separated (Fig 6d). This is inconsistent with the known characteristics of significant interactions, as they are mostly located within TADs and are frequently observed as strong apexes of TADs and sub- TADs [4, 43]. Figure 6c shows that significant interactions as predicted using DeepHiC-enhanced Hi- C data are predominantly located within TADs, and at the apexes of TADs, where they co-localize with open chromatin regions. Therefore, DeepHiC is a powerful tool for studying chromatin structure during mammalian early embryonic development.

Discussion

Hi-C is commonly used to map 3D chromatin organization across the genome. Since its introduction in 2009, this method has been updated many times in order to improve its accuracy and resolution. However, owing to the high cost of sequencing, most available Hi-C datasets have relatively low resolution (40-kb to 1-Mb). The low-resolution representation of Hi-C data limits its application in studies of genomic regulatory networks or disease **mechanisms**, which require robust, high-resolution 3D genomic data.

In this study, we proposed a deep learning method, DeepHiC, for predicting experimentally-realistic high-resolution data from low-resolution samples. Our approach can produce estimates of experimental high-resolution Hi-C data with high similarity, using 1% sequencing reads. DeepHiC is built on state-of-the-art techniques from the deep learning discipline, including the GAN framework, residual learning, and perceptual loss. With using of the GAN framework, carefully designed net architecture, and loss functions in DeepHiC, it becomes possible to predict high-resolution Hi-C with high structural similarity of 0.9 to real high-resolution Hi-C. This approach may be used to accurately predict chromatin interactions, even in fine detail. Because of the huge quantity of parameters (~121,000) included in the network, DeepHiC may be used to approximate the real data, and to make predictions in other cell or tissue types. More importantly, enhancements afforded by DeepHiC favor the identification of significant chromatin interactions and TADs in Hi-C data. Finally, we also applied DeepHiC to Hi-C data pertaining to mouse early embryonic developmental stages, **for which** only low- coverage sequencing data were available, and enhancements afforded by DeepHiC facilitated identification of significant chromatin contacts for these data.

DeepHiC provides a GAN-based framework with which to enhance Hi-C data, and even other omics data. **The** GAN framework is a state-of-the-art technique in **the** deep learning field in recent years. The idea of adversarial training **allows** the deep model to capture learnable

patterns efficiently and stably. DeepHiC is trained with real high-resolution data as target and is therefore a supervised learning paradigm. The quality of **the** target determines the upper-bound efficiency of the model. Here we used the **most deeply sequenced GM12878 cell line data** as a training set. It would be possible to retrain or fine-tune the model if more accurate Hi-C data were available, potentially reaching restriction- fragment resolution. Besides, we also performed a quasi-autoencoder training by taking low-coverage data to be both input and target, like an auto encoder, for ensuring that our model improves the sequencing depth rather than simply **cleaning**the data (S24 Fig). DeepHiC could be used not only to enhance existing low-resolution Hi-C data but also to reduce the experimental cost of sequencing in future Hi-C assays. As our method outperforms baseline methods, current low coverage Hi-C data could benefit from this improvement in performance. For example, Hi-C data imputed by our method can be used to identify significant interactions and TADs more similar to those identified with deeply sequenced Hi-C data. In some **circumstances, in which** the limitation on the number of cells stands in the way of producing high resolution Hi-C data, our method could provide an alternative solution to this problem. **In addition**, we develop a web-based tool (DeepHiC, http://sysomics.com/deephic) that allows researchers to enhance their own Hi-C data with just a few clicks. And the enhancement procedure **runs** in 3-5 minutes using single CPU (for example, enhancement on chromosome 1 of human will cost 4.7 minutes using a Xeon CPU E5-2682 v4 @ 2.5GHz). It will be faster when using a GPU (22s for Nvidia 1080ti). **We** trained several models based on various downsampled data. Translating the downsampling ratios to read coverage or data distribution is indispensable for users. We discuss the strategy of choosing **between models** in S2 Note. A caveat is that the low-coverage Hi-C data of input **should have** more than 10% non-zero **entries**.

In conclusion, DeepHiC introduced the GAN framework for enhancing the resolution of Hi-C interaction matrices. By utilizing the GAN framework and other techniques such as residual learning, DeepHiC can generate high-resolution Hi-C data using a low fraction of the original number of sequencing reads. DeepHiC can easily be used in a number of Hi-C data analysis pipelines, and prediction could be executed quickly in minutes on **the** human genome.

Materials and methods

Hi-C data sources and processing

The high-resolution (10-kb) Hi-C data used for training and evaluating were obtained from GEO (https://www.ncbi.nlm.nih.gov/geo/) under accession number GSE63525. The primary Hi-C data in GM12878 cell line (HIC001-018) is denoted as GM12878 dataset in this paper, and the corresponding replicate (HIC019-029) is denoted as GM12878R dataset for convenience. The high-resolution Hi-C contact maps for each dataset were derived from reads with mapping quality > 30. And we used the KR-normalization scheme [44] for normalized data. Corresponding low-resolution data were simulated by randomly downsampling the sequencing reads to different ratios range from 1:10 to 1:100 (i.e., 1% reads). Downsampled data would typically be processed at lower resolution because of the shallower sequencing depths. In our experiments, low-resolution contact maps were built using the same bin size as used for

high-resolution Hi-C to fit the models' requirement. All resolution enhancing methods compared in our study used this same procedure as reported in HiCPlus [17] to ensure fair comparisons. Hi-C data pertaining to mouse embryonic development were obtained from GEO under accession number GSE82185. Hi-C matrices of 10-kb bin size were created using the HOMER (http://homer.ucsd.edu/homer/) analyzeHiC command with the following parameters: -res 10000 – window 10000.
ChIA-PET data for the CTCF target in the K562 cell line was obtained from ENCODE (https://encodeproject.org) under accession number ENCSR000CAC (file: ENCFF001THV.bed.gz). Chromosome regions were mapped to Hi-C bins with which they overlapped in ROC analysis. ATAC-seq data on mouse early embryonic development was obtained from GEO under accession number GSE66390. Data of DNase-seq peaks on GM12878 cell line was obtained from ENCODE under accession number ENCSR000EMT.
For Hi-C matrices in training, outliers are set to the allowed maximum by setting the threshold be the 99.9-th percentile. For example, 255 is about the average of 99.9-th percentiles for 10-kb Hi-C data, so all values greater than 255 are set to 255 for 10-kb Hi-C data. Then all Hi-C matrices are rescaled to values ranging from 0 to 1 by min-max normalization [45] to ensure the training stability and efficiency. Besides, cutoff values for downsampled inputs of our model were 125, 100, 80, 50, and 25 for 1/10, 1/16, 1/25, 1/50, and 1/100 downsampled ratios.

DeepHiC architecture

In general, DeepHiC is a GAN model that comprises a generative network called generator and a discriminative network called discriminator. The generator tries to generate enhanced outputs that approximate real high-resolution data from low-resolution data, while the discriminator tries to tell generated data apart from real high-resolution data and reports the difference to the generator. The contest (hence "adversarial") between generator and discriminator promotes the generator learns to map from conditional input to a data distribution of interest.
As depicted in S1 Fig, the generator net (G) is a convolutional residual network (first row), while the discriminator net (D) is a convolutional neural network (second row). The G net takes low-resolution matrices (X) as input and outputs enhanced matrices ($\hat{Y}$) with identical size. The adversarial component, the D net, takes the enhanced output $\hat{Y}$ and the real high-resolution data (Y) as input and outputs 0-1 labels. The green arrowed lines describe how data are processed in DeepHiC. The G net, employs two layers: the convolutional layer (blue block) and the batch normalization (BN) layer [46] (yellow block). Together with elementwise sum operation (green ball) and skip-connection operation (green polyline), some of these layers form the residual blocks (ResBlocks) [47]. There are five successive ResBlocks in G. As for the activation function (pink block), we elected to use the Swish function [48] instead of the Rectified Linear Unit (ReLU) for activating some layers. The Swish function is defined as:
$f(x) = x \cdot \sigma(\beta x)$,
where $\beta = 1$ and σ is the sigmoid function. Swish has been shown to works better than ReLU in deep
models [49]. Note that the final outputs of G are scaled by:
$g(x) = \tanh(x) + 1.2$

Thus, elements in output matrices range from 0 to 1. In general, the G net contains about 121,000 parameters. The D network is a convolutional network similar with the VGG network [50]. The number of kernels in a convolutional layer is depicted via block width: the more kernels, the wider the width of the block. The final output of D is a scalar value ranges from 0 to 1 by a sigmoid function. More details of the hyperparameters of network architectures, such as kernel size and filter numbers, are summarized in S1 Table and S2 Table.

To establish the GAN paradigm for training (Fig 1a), we employed both the generator net G and the discriminator net D. The G net aims to generate enhanced outputs by approximating to the real high-resolution matrices Y, while the D net attempts to distinguish the real Y from the generated $\hat{Y}$. In the D net, the value of output $\hat{y} = D(\hat{Y})$ is considered to be the probability of $\hat{Y}$ to be real data. Divergences between $\hat{Y}$ and Y, as well as the probability of $\hat{Y}$ to be real data, are minimized according to a carefully designed loss function. Besides, these two networks are trained alternatively by the backpropagation algorithm.

Loss functions in DeepHiC

A critical point when designing a deep learning model is the definition of the loss function. Many methods have recently been proposed to stabilize training [51, 52] and improve the quality of synthesized images [37] by the GAN model. For DeepHiC, the binary cross entropy loss function for the D network was used to measure the error of output, as compared with the assigned labels. Because real and generated high-resolution data are paired in practice, it can be described as:

$$LD = N1\sum \log(\hat{y}i) + \log(1 - yi),\ i$$

where i is the index for pairs of real and generated data, and N is the number of pairs. Here we used $y = D(Y)$ and $\hat{y} = D(\hat{Y})$.

For generator loss, we used four loss functions, which were added to yield a final objective function. Firstly, we used MSE to measure the pixel-wise error between predicted Hi-C matrices and real high-resolution matrices, defined as:

$$\frac{N}{i=1}$$

which is also called L2 loss. The MSE loss function is broadly used for regression problems, while the

fact that MSE loss does not correlate well with the human perception of image quality [53] and overly smooths refined structures in images [27]. We also employ perceptual loss [37], however, based on the feature layers of the VGG16 network. We used total variation (TV) loss , derived from the total variation denoising technique, so as to suppress noise in images [54]. Final generator loss is yielded in combination with adversarial (AD) loss derived from D network and defined as:

$$LG = lMSE + \alpha \cdot lVGG + \beta \cdot lTV + \gamma \cdot lAd.$$

Note that $lAd = (\sum i\ \hat{y}i)/N$ without logarithmic transformation, which allows for fast and stable training

of the G net [51]. Hyperparameters α, β, γ are scale weights that range from 0 to 1.

Implementation of DeepHiC and performance evaluation

DeepHiC is implemented in Python scripts with PyTorch 1.0 [55]. After splitting GM12878 dataset into a training set and a test set, the model was trained on the training set and tested on the test set during training process. The final model we used was trained on chromosomes 1-14. We divided contact matrices where the genomic distance between two loci is < 2 Mb, as the average size of TAD is < 1 Mb and there are few significance interactions outside TADs, thus could be omitted for training. The Adam optimizer [56] is used with a batch size of 64, and all networks are trained from scratch, with a learning rate of 0.0001. We trained the networks with 200 epochs. In order to yield loss terms on the same scale, the hyperparameters for generator loss were set as α = 0.006, β = 2 × 10−8, and γ = 0.001. All training process were performed using an NVIDIA 1080ti GPU. A python code for model training and prediction is available at https://github.com/omegahh/DeepHiC.

In order to assess the efficiency of DeepHiC during training, we performed an improved measure called structure similarity index (SSIM) [57] to measure the structure similarity between different contact matrices. The SSIM score is calculated by sliding sub-windows between images. The measure for comparison of two identically sized sub-windows, x and y (from two images) is:

$$SSIM(x,y) = (2\mu x\mu y + C1)\cdot(2\sigma xy + C2)\ ,\ (\mu x2 + \mu y2 + C1)\cdot(\sigma x2 + \sigma y2 + C2)$$

where mean (μ), variance (σ), and covariance (σxy) are computed with a Gaussian filter. They measure the differences of luminance, contrast, and structure between two images, respectively.

$C1$, $C2$ are constants to stabilize the division with a weak denominator. In our experiments, the size of sub-windows and the variance value of Gaussian kernel are set as 11 and 3, respectively. And all compared matrices are rescaled by min-max normalization to same range to eliminate the differences of luminance in order to compare the contrast and structure differences.

Dividing and reconstructing matrices

We divided the whole Hi-C contact maps into equal-sized square submatrices to be used as model inputs. It reduces the time and memory cost in batch training. The size of submatrices determines the features' dimension of each sample. Here we used the same size of 0.4 Mb × 0.4 Mb as described in HiCPlus, note that other choices such as 0.3Mb or 0.5Mb is also applicable in our workflow. So, each submatrix contains 40 × 40 = 1600 pixels at 10 kb resolution. As shown in Fig 1b, the intact low-resolution Hi-C matrix was divided into non-overlapping sub-regions, then enhanced sub-regions were predicted from them (with outlier squashed and min-max normalization performed) by the generator network of DeepHiC. Finally, the high-resolution sub-matrices predicted were merged into a chromosome-wise Hi-C matrix, as the final enhanced output. Because our model is trained based on the contact maps where two bins < 2Mb genomic distance, we made the genome-wide predictions also on data where two bins < 2Mb.

Identifying chromatin **interactions** and detecting TAD boundaries

Chromatin **interactions** [7] are identified using the commonly used software: Fit-Hi-C. We parallelized the software for faster running speed and suitable for our data. The modified code is available in https://github.com/omegahh/pFitHiC. Fit-Hi-C parameters were set as follows: $resolution$ = 10kb, $lowerbound$ = 2, $upperbound$ = 120, $passes$ = 2, $noOfBins$ = 100. Significance was calculated only for intra-chromosome interactions. Since our model's output ranges from 0 to 1, we converted them to integer by multiplying 255 to be Fit-Hi-C inputs.
TADs were detected using the insulation score algorithm [14] with minor modifications: the width of the window used when calculating insulation score was set to 5 times of Hi-C matrix resolution to better detect the boundaries of finer-domain structures. We computed the delta score using insulation score of 5 nearest loci upstream and of 5 nearest loci downstream. We identified TADs as the genome region between center of 2 adjacent boundaries and regions containing low-coverage bins were excluded.

Measurements for two TAD segmentations

We investigated the consistency of segmentations formed by different TAD boundaries in the genome. Here we calculated the distance of two segmentations and the corresponding overlap, defined as follows. We denote the two segmentations as S and T, which are formulated in sets consisting of their split points:
$S = \{s1,s2,...,sn\}$, $T = \{t1,t2,...,tm\}$,
where m, n are numbers of split points. Thus, we could calculate the distance from one split point $si \in S$ to segmentation T, as follows:
$d(si,T) = \min d(si,tj),\ \forall j = 1,2,...,m.$
The overlap of an interval $IS = (si,si+1)$ from S, compared with T, could be measured as follows:
$JI(IS,T) = \max JI(IS,IT),\ with\ IT = (tj,tj+1),\ \forall j = 1,2,...,m-1,$

Implementation of baseline models
For baseline models, we only performed comparisons on data downsampled to 1/16 **reads** as they commonly used in their study [17, 25, 26]. The python source code for HiCPlus was obtained from https://github.com/zhangyan32/HiCPlus_pytorch, together with the codes for data processing and pre-trained model parameter file. We obtained HiCPlus results using the downloaded source code and pre-trained model parameter file. The scheme of data downsampling and reconstructing were implemented according to the description in its paper [17]. For Boost-HiC, the python source code was obtained from https://github.com/LeopoldC/Boost-HiC and implemented with $alpha$ = 0.2. For HiCNN, we obtained its implementation code from http://dna.cs.miami.edu/HiCNN/ and pretrained model parameters from http://dna.cs.miami.edu/HiCNN/checkpoint_files/. We used the "HiCNN_16" for experiments for 1/16 downsampled data.

Acknowledgements

We thank the Aiden Lab, the Wei Xie lab, and ENCODE Consortium for high-quality data.