BMC Genomics

SOFTWARE

Open Access

HiCDiffusion - diffusion-enhanced, transformer-based prediction of chromatin interactions from DNA sequences

Mateusz Chiliński[1,2,3] and Dariusz Plewczynski[1,3*]

Abstract

Prediction of chromatin interactions from DNA sequence has been a significant research challenge in the last couple of years. Several solutions have been proposed, most of which are based on encoder-decoder architecture, where 1D sequence is convoluted, encoded into the latent representation, and then decoded using 2D convolutions into the Hi-C pairwise chromatin spatial proximity matrix. Those methods, while obtaining high correlation scores and improved metrics, produce Hi-C matrices that are artificial - they are blurred due to the deep learning model architecture. In our study, we propose the HiCDiffusion, sequence-only model that addresses this problem. We first train the encoder-decoder neural network and then use it as a component of the diffusion model - where we guide the diffusion using a latent representation of the sequence, as well as the final output from the encoder-decoder. That way, we obtain the high-resolution Hi-C matrices that not only better resemble the experimental results - improving the Fréchet inception distance by an average of 11 times, with the highest improvement of 56 times - but also obtain similar classic metrics to current state-of-the-art encoder-decoder architectures used for the task.

Keywords 3D genomics, Hi-C, Machine learning, Artificial intelligence

Introduction

With the development of the experimental methods, the spatial organisation of the chromatin became of high importance and interest to the scientific community. The spatial landscape is vital for understanding how genetic machinery works [1] - and can be used in e.g. precision medicine [2], for example, by improving gene expression prediction [3]. However, those spatial experiments are expensive and time-consuming to run. Thus, multiple efforts have been made to predict the spatial organisation of chromatin in the nucleus using various machine-learning models. The first ones relied on epigenetic signals (e.g. ChIP-Seqs, or ATAC-Seqs) or peaks [4–7]. However, the community wanted to use the most easily accessible genomic data, i.e. DNA sequence to identify in silico the spatial landscape of chromatin for specific cell lines. Multiple algorithms were proposed, providing end-to-end solutions from DNA-Seq to chromatin 3D interactions prediction. Such predictions, based only on the sequence, are of high interest because they allow the incorporation into the model genome sequence mutations that might cause changes in the spatial organisation [8], thus changing the expression or behaviour of genetic machinery

*Correspondence:
Dariusz Plewczynski
Dariusz.Plewczynski@pw.edu.pl
[1]Laboratory of Bioinformatics and Computational Genomics, Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw 00-662, Poland
[2]Section for Computational and RNA Biology, Department of Biology, University of Copenhagen, Copenhagen, Denmark
[3]Laboratory of Functional and Structural Genomics, Centre of New Technologies, University of Warsaw, Warsaw 02-097, Poland

significantly. To account for that, differences between the reference genome and an individual one (obtained by reads from an experiment) can be calculated. Those include Single Nucleotide Polymorphisms (SNPs), short indels (insertions or deletions below 50-100bps), and Structural Variants (SVs; insertions, deletions, duplications, copy number variations and other complex rearrangements) - and they describe how an individual differs from the reference. Applying those variants to the reference creates a personalised genome of an individual. These statistical models allow to discover the spatial landscape of chromatin for a given individual using only DNA sequence - which is relatively easy to obtain and process. Multiple bioinformatics pipelines have been proposed to identify single nucleotide mutations and structural variants from raw DNA-seq data [9–11]. Those pipelines automate the whole variant discovery, taking as input in most cases either raw reads from DNA-Seq experiment (in the form of fastq files) or already-aligned data, and perform a series of calls to multiple algorithms for the variant discovery. The user is left with a list of variants, which can be used later for further analyses. We envision the entire in silico workflow for in-silico prediction of chromatin structure (Hi-C pairwise contact matrix) from DNA sequence, starting from DNA-Seq experimental data, running sequence variant discovery, mapping the variants to the reference genome, and running the selected 3D predictive computational model on personalised DNA sequence. Those Hi-C prediction models include convolutional networks - e.g., ChiNN [12] uses DNA sequences from open chromatin regions to predict the chromatin interactions. The classifier (that uses the features discovered by CNN) was xgboost. The authors obtained high metrics; however, they did not report any heat maps, which are of interest to us the most. Another approach is using transfer learning-based convolutional networks - like the authors of DeepC [13]. In that case, the procedure is split into two stages - in the first training phase, the 1kbp sequence is taken as an input, and the fully connected layers predict chromatin features (936 different tracks - primarily obtained from ENCODE). Afterwards, in the second phase, the learnt parameters of the convolutional layers are used for training the final model - in which the sequence of 1Mbp is taken, then transformed using convolutional layers trained in phase 1, and then another dilated convolutional layers were applied, and the final output was in form of predicted interaction strengths, which could be then arranged as heatmaps. Another concept used is convolutional encoder-decoder architectures, which include Akita and Orca [14, 15]. The key concept there was to first use encoder, that takes 1D DNA sequence and processes it into hidden representation using 1D convolutions, and then decoder,, which then operates on 2D heatmaps (with addition of hidden dimensions). In case of Akita, the initial region was 1Mbp, and in caase of Orca, it was up to 256Mbp, depending on the resolution we are interested in. Such an approach was considered state-of-the-art, till the transformer-based methods were established. Including transfer learning application of DNABERT [16] for prediction of chromatin interactions [17], that was able to recreate ChIA-PET interactions, however, as with ChINN, it was not used for production of HiC matrices, only the point interactions. Finally, convolutional encoder-decoder, transformer-based architectures were introduced with the publication of C.Origami [18], which outperformed all the other models in various metrics. Those studies have achieved high metrics, including the Pearson correlation coefficient and the stratum-adjusted correlation coefficient (SCC) between the real and predicted heatmaps. In terms of their metrics, each architectural improvement yielded improvement - starting with Akita and DeepC, which obtained very similar results and were developed simultaneously, one obtaining its power from the encoder-decoder architecture, and the other from transfer learning, going through Orca, which used and improved encoder-decoder architecture,, and finally arriving at C.Origami, that outperformed all the other models' metric because of the use of a transformer. However, while highly informative, none of those studies addressed the visual quality of the predicted heatmaps - and because of that, they look artificially produced. The reason for that is in the common part of all the networks - namely, convolution layers. While they are extremely useful for the prediction of the genetic landscape, they are responsible for the smoothening effect, which makes the output highly informative but very distinct from the real heatmaps. Distinguishing between the real Hi-C matrix and one predicted by any of the presently available tools is not difficult - as they seem to be highly blurred. While those methods preserve the most critical parts of the matrices, they lack quality assessed easily by human perception.

To address this challenge, we have decided to use the recent theoretical achievements from the computer vision field. One of the many problems that computer vision is currently facing is generating images of high quality. However, in our problem, we want to strongly guide the generation in order to reflect the actual physical nature of the underlying polymer. Thus, we are primarily interested in conditional architectures that allow us to improve the quality of the Hi-C matrices, not generate random pairwise contact maps. Multiple neural network architectures have been proposed to face this problem in computer vision - Generative Adversarial Network [19] being a prime example. However, lately, diffusion-based models [20] have gained tremendous popularity due to much higher performance. There have

been multiple implementations, each adding some value to the architecture and allowing to obtain even higher similarity metrics. Examples include Stable Diffusion [21] or DALL-E 2 [22]. However, those are still used primarily for generating new images from text description - and not improving existing ones. That is why we have decided to implement the deblurring diffusion model [23–25] to guide the network in improving the quality of the given heatmap.

Using previously established encoder-decoder architectures, we aimed to improve the predicted heatmaps' quality using transfer learning as a core to a deblurring, diffusion-based model. We have developed the HiCDiffusion model, which provides high metrics as obtained by other encoder-decoder-based algorithms and significantly improves the predictions' human-readable quality. Our approach makes the predicted heatmaps almost indistinguishable from the real data, which we have measured using Fréchet inception distance [26]. That metric, widely used in computer vision and image generation, decreases when the quality increases. The main advantage of using FID is the fact that it does not compare the output with the input in a pixel-by-pixel way, like the traditional metrics (e.g., squared error), but rather it compares the mean and standard deviation of a specific function (in our case, and in most real usages - deepest layer of Inception v3 model) - allowing the comparisons to be made between the real distribution and the generated distribution (see **Methods** for formal introduction) in a computer vision-specific domain. To put it into perspective when dealing with the Hi-C experiments, we have blurred an example real Hi-C map and calculated FID scores of the original and augmented data. The results are shown in Fig. 1.

In the case of FID, we deal with two distributions - one real and one generated. In the case of the situation where the real distribution is equal to the generated one, we get FID equal to 0.0 - which is the highest possible FID score. That is also the case in our example - when we take real Hi-C data and compare it to itself, we get FID equal to 0.0. The following steps are blurring the Hi-C matrix and comparing the real distribution (composed of the real data), with the blurred matrix. We have shown 3 examples with σ = (1, 3, 7). In the first case, where the blur is not that intense, as we are using σ=1, the FID score equals 18.3. In the case of the higher blur, with σ=3, we are getting a much more augmented matrix - and the FID score rises to 61.5. In the last example, we have used Gaussian blur with σ=7, and the FID score is 70.6. We can clearly see that, indeed, with the higher blurring of the matrix, we are getting a higher FID score. That is why, in our study, the goal was to decrease the FID score, that is, to deblur the Hi-C matrix generated by convolutional encoder-decoder architectures.

Results

The Hi-C Diffusion model that we propose is composed of multiple components (see Fig. 2). The first part, encoder-decoder architecture, is similar to the current state-of-the-art tools [15, 18]. The input to the network is a genomic sequence - one-hot encoded, and the final output is the Hi-C matrix. We first use an encoder composed of residual 1D convolutions that transfer 1D sequence into latent space; furthermore, we use a transformer encoder to allow the model to learn long-range context. Such latent representation is then cast into the 2D matrix, and the decoder, composed of 2D convolutions with exponentially growing dilation, produces the

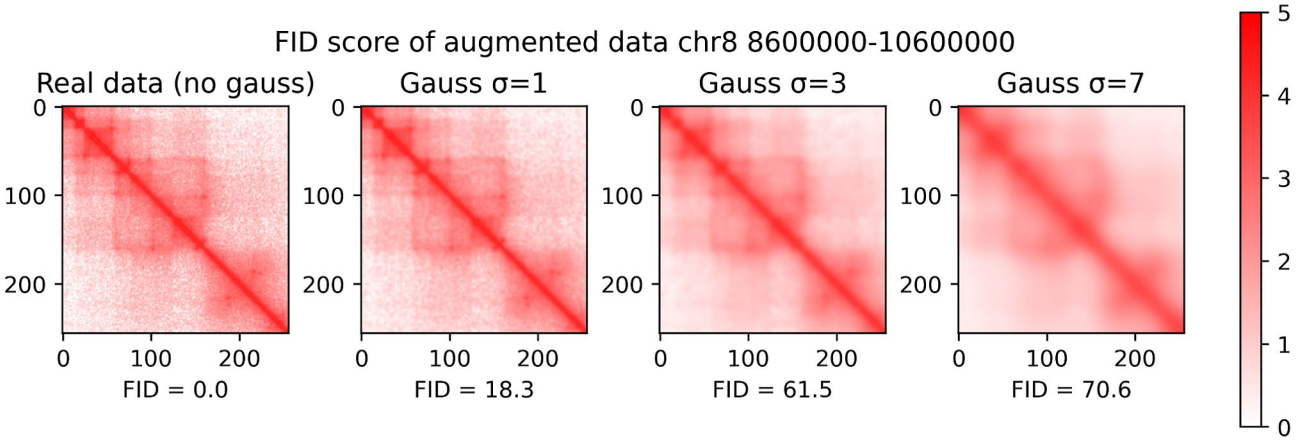

Fig. 1 FID score of the real and augmented data - example of chr8 8 600 000–10 600 000. The first example is the real data - we compare the real data distribution with the "generated" distribution, which is also the real data. We can see that the FID score is then 0.0 - because the distributions are precisely the same. We have applied Gaussian blur to this example - in the case of blur with σ = 1, the FID score increases to 18.3; in the case of the blur with σ = 3, it's 61.5, and in the last case, when Gaussian blur is performed with σ = 7, we are getting FID equal to 70.6. We can clearly see that the blurring increases the FID score significantly while reducing the data quality

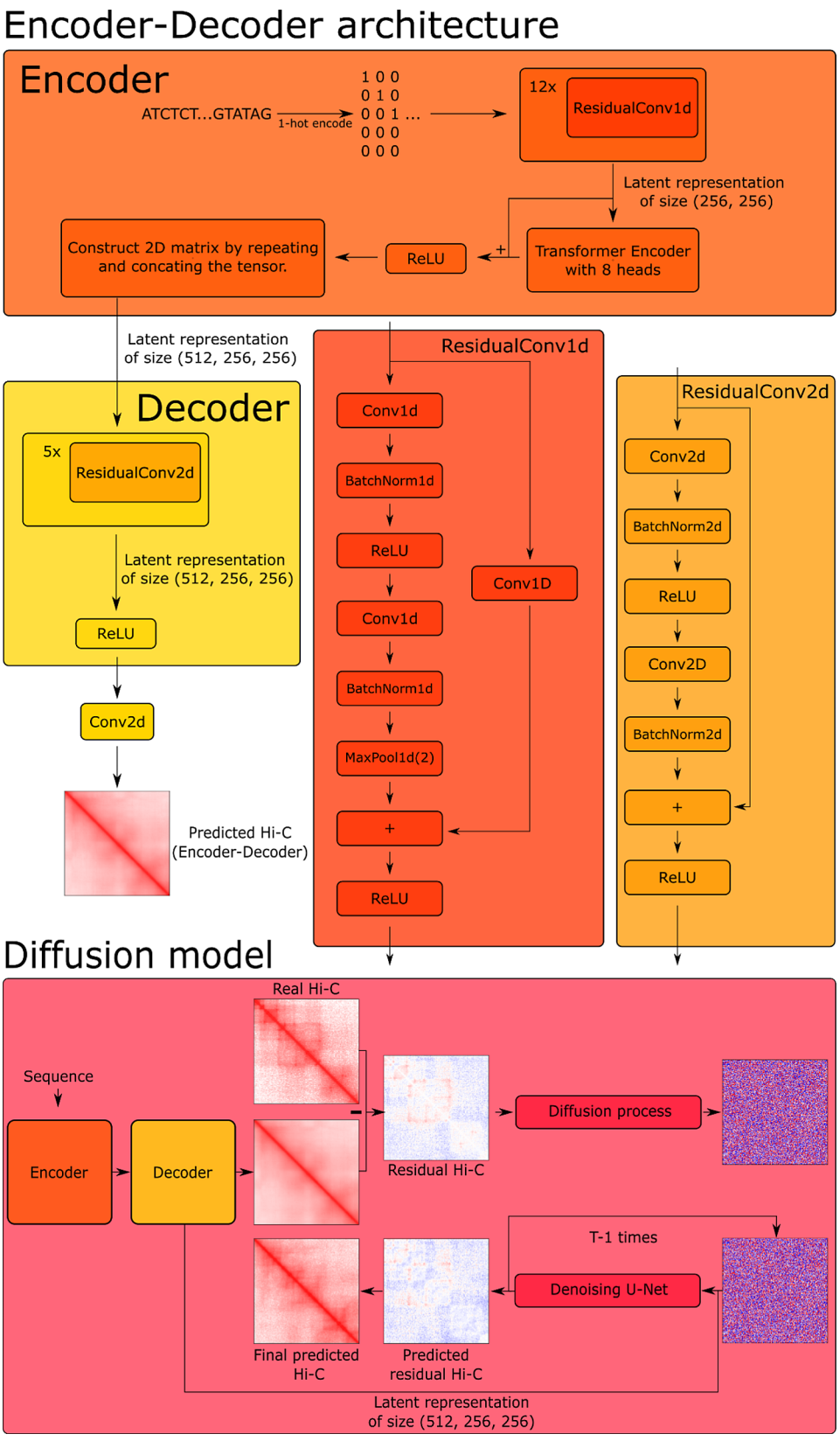

Fig. 2 The architecture of HiC Diffusion. The example used to visualise the prediction & real data is chr8, position 8 600 000–10 600 000

final matrix. The last part of the encoder-decoder architecture is a convolution that transfers the latent space's final representation into a classic Hi-C matrix of size 256×256 with one channel. The second network, which uses transfer learning (by taking pre-trained encoder-decoder architecture), is the diffusion model. Based on the previous findings about diffusion networks, we have decided that the input to the network will be residual between the real Hi-C map and the final prediction of the encoder-decoder network. Then, we apply Gaussian noise to the residual Hi-C map, and the denoising U-Net is trained to predict the noise, making it easy to obtain the actual residual Hi-C (by subtracting predicted noise from the noised residual Hi-C). The network takes as input the noised residual (or, in case of inference - the random input) and the latent representation of the Hi-C heatmap predicted by encoder-decoder architecture. The latent representation stores the knowledge about the sequence and its meaning in the context of the whole genomic window. It can be used for multiple downstream tasks. In our case, it guides diffusion to create a heat map that is as close as possible; however, it can be easily applied to classification problems as well (see Supplementary Materials). Using hidden representation is necessary to guide the diffusion, thus creating a conditional diffusion model. For more information on the technical details of the architecture, see Methods and Fig. 2.

In our study, we have decided to use the context of 2,097,152 nucleotides of sequence and predict the same Hi-C region. To validate the model thoroughly, we have performed 22-fold cross-validation - creating 22 models, each with one chromosome excluded for testing purposes (see **Methods**). Our primary motivation behind this approach was to ensure that the model works no matter which chromosomes are used for training and which for validation/testing. In a standard use case, one model is sufficient for downstream analysis. We have calculated the Pearson correlation coefficients for each of the examples in the testing set (for each model), SCC (Stratum-adjusted correlation coefficient), and the FID score. To compare ourselves with the current state-of-the-art tool, C.Origami, we have trained the model ourselves, according to the authors' recommendations; we have used two approaches - in one, we have excluded the epigenetic signal that they used - to keep the results consistent with our findings (see **Methods**), and second one, with epigenetic signal that they presented as the final model. We calculated and compared the Pearson correlation coefficients (as well as SCCs) to our model. The results are consistent and very similar - as in our work, we were aiming to obtain similar metrics in terms of correlation, in our case, even more challenging, i.e. **without epigenomic profiles** used as an additional input apart from the DNA sequence. Then, we calculated the FID scores for all the datasets obtained using HiCDiffusion and C.Origami. We have obtained an average improvement of FID score by 12 times in case of comparison between sequence-only models - with the highest improvement in chr7 (by 88 times). In case of comparison of our sequence-only model to C.Origami enhanced with epigenetics, the average improvement of FID score was by 11 times, and the highest improvement was also obtained in chr11 (by 56 times). The visualisation of those results can be seen in Fig. 3, and detailed per-chromosome statistics can be found in Supplementary Figs. 1–3.

The model was further tested in downstream analysis tasks. The first one was TAD calling. The obtained insulation scores were very similar to those observed in the experimental data. The average Pearson correlation score for the insulation (predicted/real) was 0.63 for chromosome 8. The full results from the analysis, with the violin plot of the correlations, as well as two examples, can be seen in Fig. 4. In the presented examples, we can see that the algorithm correctly finds TAD boundaries in both cases - however, in the case of the real data, in chr8:130 380 000-132 380 000, we detect two TAD boundaries - which are very close to each other and are seen as one in the predicted data. More sophisticated TAD calling algorithms could also be applied to the output of the model, provided the algorithms work per heatmap (as the scope of interest in case of the predicted data is 2Mbps - and we are sliding across the diagonal for the prediction of each next heatmap).

Another analysis, as presented in Fig. 5, showed that it is possible to call loops from the predicted heatmaps. However, for the HiCDiffusion model, that task is much harder - as we are dealing with the sequence-only model, and epigenetic tracks that can indicate very strongly looping factors (e.g., CTCF) are not present in the input data.

To see the behaviour of the model in the case of the mutations, we have simulated the event of transduplication of a region resembling a small TAD. To do so, we have used the model to predict the genomic window of chr8:32 1000 000–34 100 000 - firstly, using wild type sequence, and secondly, applying a transduplication of chr8:33 450 000–33 650 000 to chr8:32 750 000–32 950 000. The input to the model in the case of the wild type was the raw sequence, and in the case of the perturbed experiment - the sequence at chr8:32 750 000–32 950 000 was replaced with the one present at chr8:33 450 000–33 650 000. The results of such an operation can be seen in Fig. 6 (see Supplementary Materials for comparison with C.Origami model), where we can see that the change not only modified the region directly affected but also isolated the top-left corner more (which is shown as the increase of contacts within the top-left corner TAD,

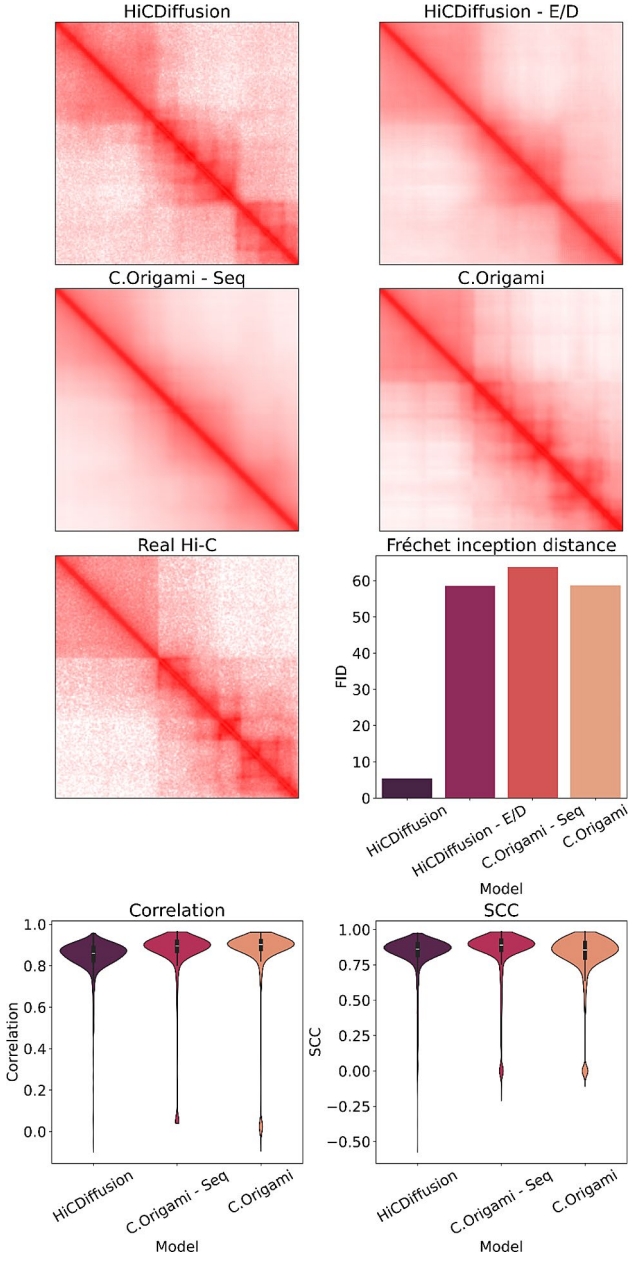

Fig. 3 In the upper part of the figure, an example output of the 3 models is presented - full HiCDiffusion, HiCDiffusion - only encoder and decoder, and C.Origami (version with epigenetics and sequence, and version with only sequence), along with the real Hi-C matrix. All example heatmaps are chr8, position 21 100 000–23 100 000. The lower part of the chart presents the Pearson correlation coefficient, based on data from all chromosomes, the stratum-adjusted correlation coefficient (SCC) - calculated in similar way, and an average FID obtained by the models (the average is taken from the per-chromosome metric)

as well as decrease of contacts downstream from it), and created a new TAD in the middle of the heatmap.

The final analysis that we have undertaken was the question - how well would our method perform on different organisms? To answer this question, we have taken a B cell derived cell line from a mouse [27], and tested our model on all the chromosomes. We have found that even if it's a different organism, the strength of the model persisted - we obtained a genome-wide Pearson correlation score of 0.847 and stratum-adjusted correlation coefficient of 0.761. The cumulative results of the analysis, as well as an example heatmap, can be seen in Fig. 7. For detailed per-chromosome results, see Supplementary Fig. 4.

Methods

Data processing

The first step of processing the data is creating the genomic windows analysed in the study. The sliding window that is used for the processing of the chromosomes is set to 500kbp. We load the reference genome (GRCh38) and use pyranges [28] to exclude telomeres and centromeres from the analysis. Then, the sequence is onehot encoded. The Hi-C matrices used in this research are taken from C.Origami [18] paper - we used GM12878 [29] cell line to allow us to compare our findings with that current state-of-the-art tool. We take precisely 2,097,152 base pairs of sequence and predict the Hi-C matrix of the same region (resized to 256×256 region) - the resolutions were chosen to easily and straightforwardly use convolutions in the network architecture.

Training, testing, and validation data

To show the predictive power of the method, we divided the dataset into training (used for training), validation (used for choosing the best models - both encoder/decoder and diffusion), and testing (separate, used only for final testing) datasets. This division ensures that the deep learning model is generalising well and that we are unbiased toward examples occurring in the training data. To test the model entirely, we decided to create 22 models - in each, the training, validation, and testing datasets are composed of different chromosomes. Such an approach allows us to be sure that no matter which chromosomes we use for training/validation/testing, the model is still trained properly and maintains its generalisation power. For a standard use that does not require such thorough testing, one model is entirely sufficient. For each case, we take chromosome i as the testing chromosome, chromosome i+1 as the validation, and the remaining chromosomes compose the training dataset. In the case of testing the last chromosome, chr22, the validation chromosome is chr21. We excluded sex chromosomes from the analysis.

Architecture of the model

The architecture of the model (see Fig. 2) uses the concept of transfer learning. Firstly, we use encoder-decoder architecture very similar to the ones previously published - e.g. in C.Origami [18] or Akita [15]. The encoder first

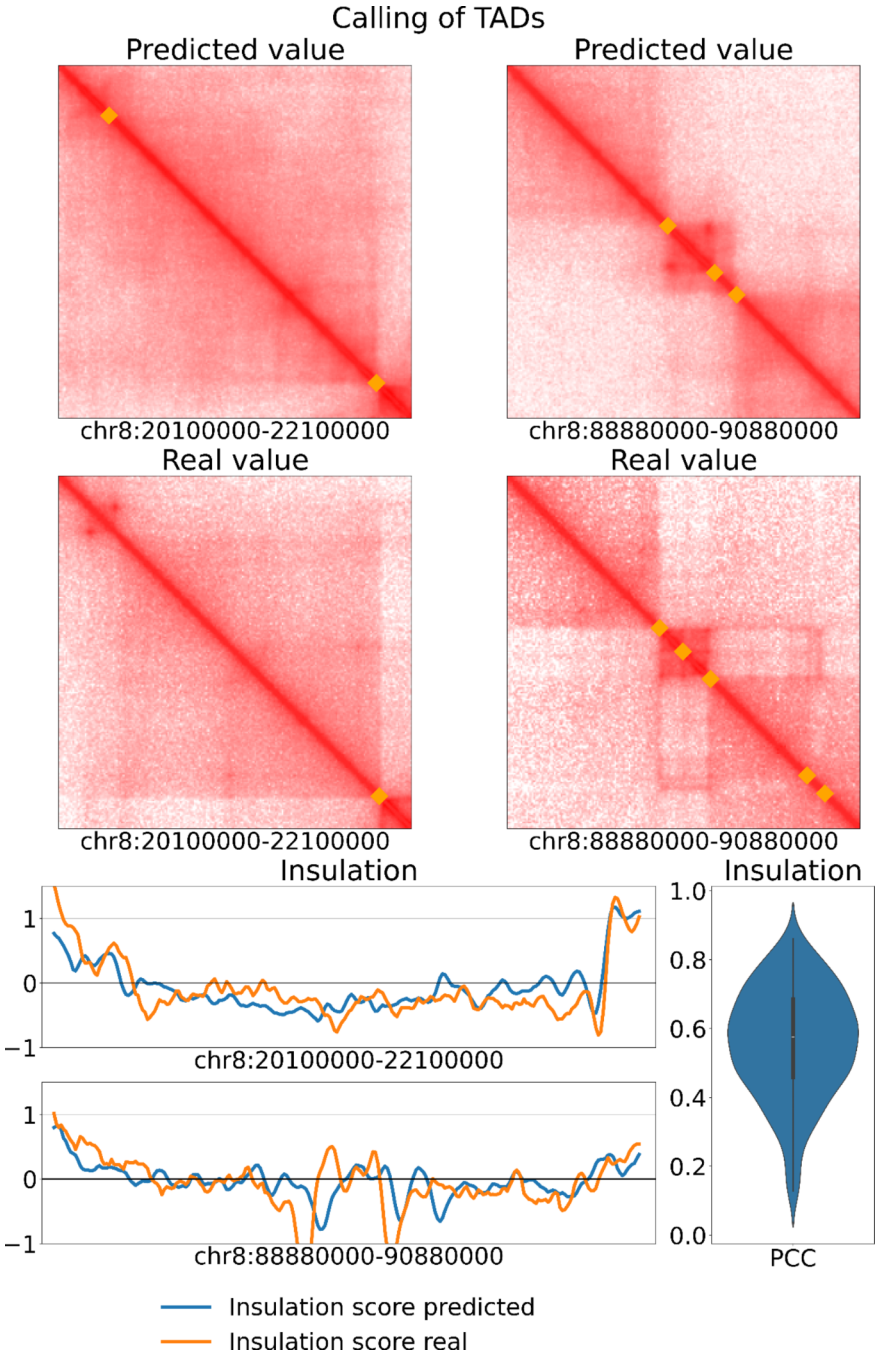

Fig. 4 TAD calling analysis in chromosome 8. The Pearson correlation coefficient between predicted and real insulation scores averaged 0.63. The two examples show TAD boundaries (orange squares) in chr8:20 100 000–22 100 000 and chr8:88 880 000–90 880 000

converts the 1D genomic sequence into a sequence of 256, with 256 channels. That is done using 13 residual blocks, out of which each is composed of convolution (converting input channels into output channels, with the kernel of size 3 and padding of size 1), batch normalisation, ReLU function, another convolution (this time preserving the number of channels, with kernel of size 3, and padding of size 1), batch normalisation, and maxpooling. Additionally, the initial data provided to the residual block is downscaled using convolution (converting input channels directly into output channels, with the kernel of size 3, and padding of size 1). That downscaled data is added to the result from the previously explained sequence of transformations. The final step of the residual block is applying the ReLU function to the output. The input channels of the residual blocks used in the encoder are: (5, 32, 32, 32, 64, 64, 64, 128, 128, 256, 256, 256, 256), and the output channels are: (32, 32, 32,

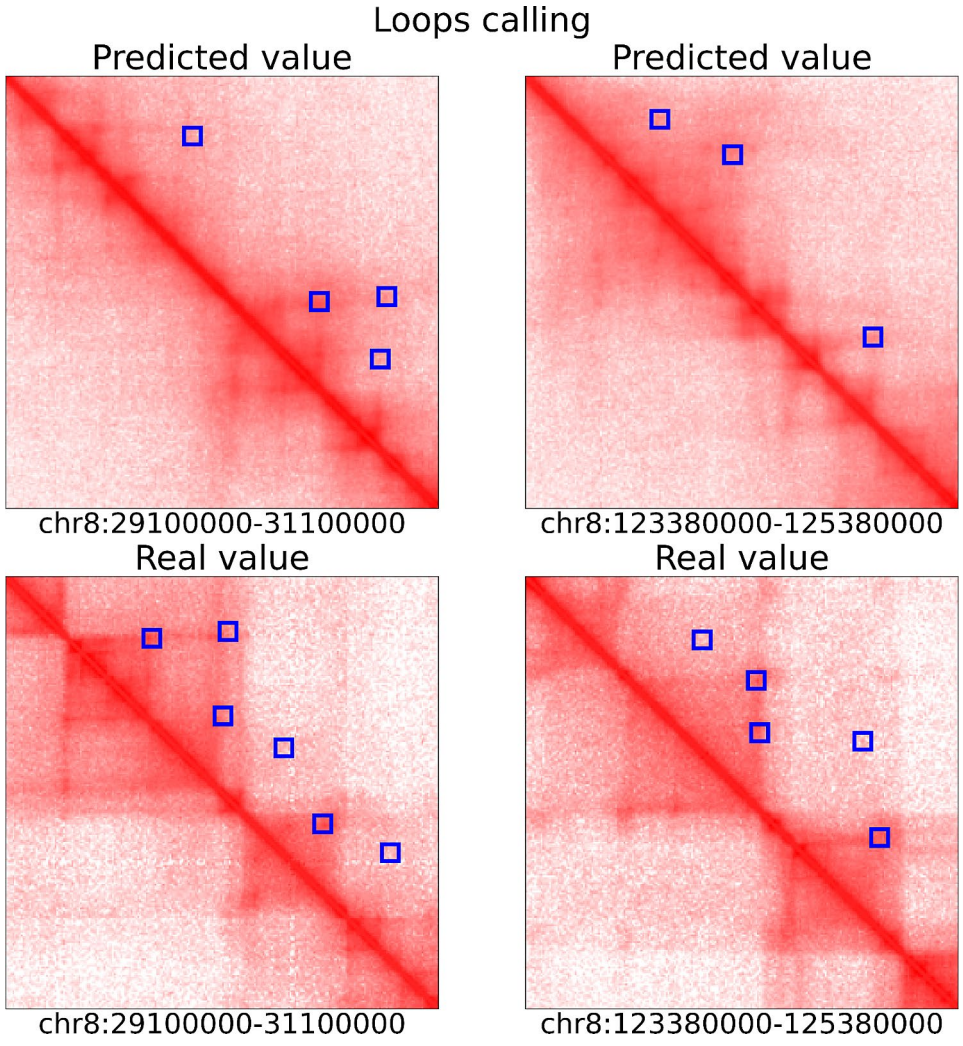

Fig. 5 Loops called on example regions - Chr8:29 100 000–31 100 000 and chr8:123 380 000-125 380 000 - on the real and predicted values

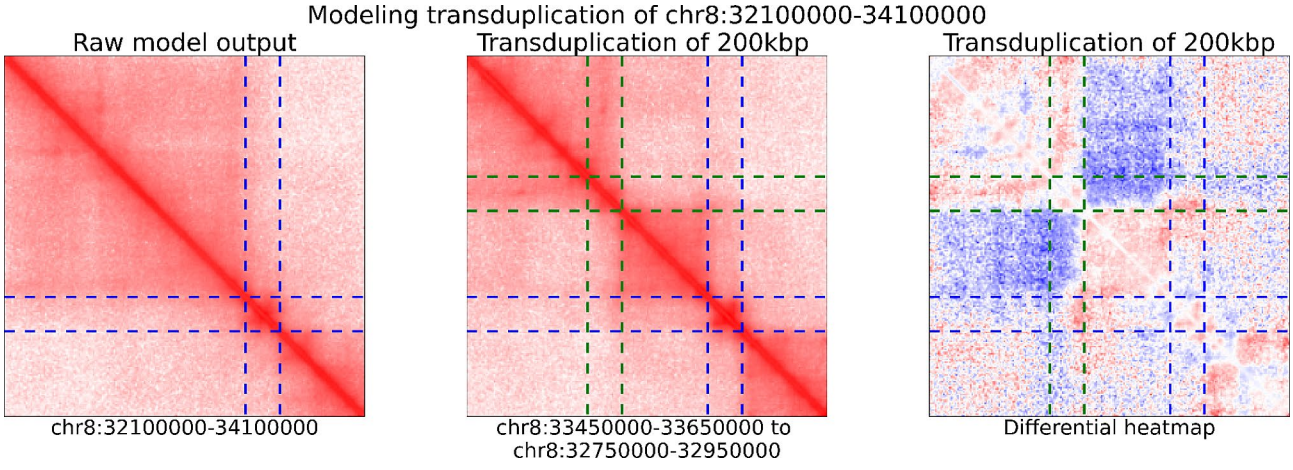

Fig. 6 Modelled transduplication of chr8:32 100 000–34 100 000 region. The first heatmap is raw model output - with no changes in the sequence. The second heatmap shows the output of the model with chr8:32 750 000–32 950 000 replaced by a sequence of chr8:33 450 000–33 650 000

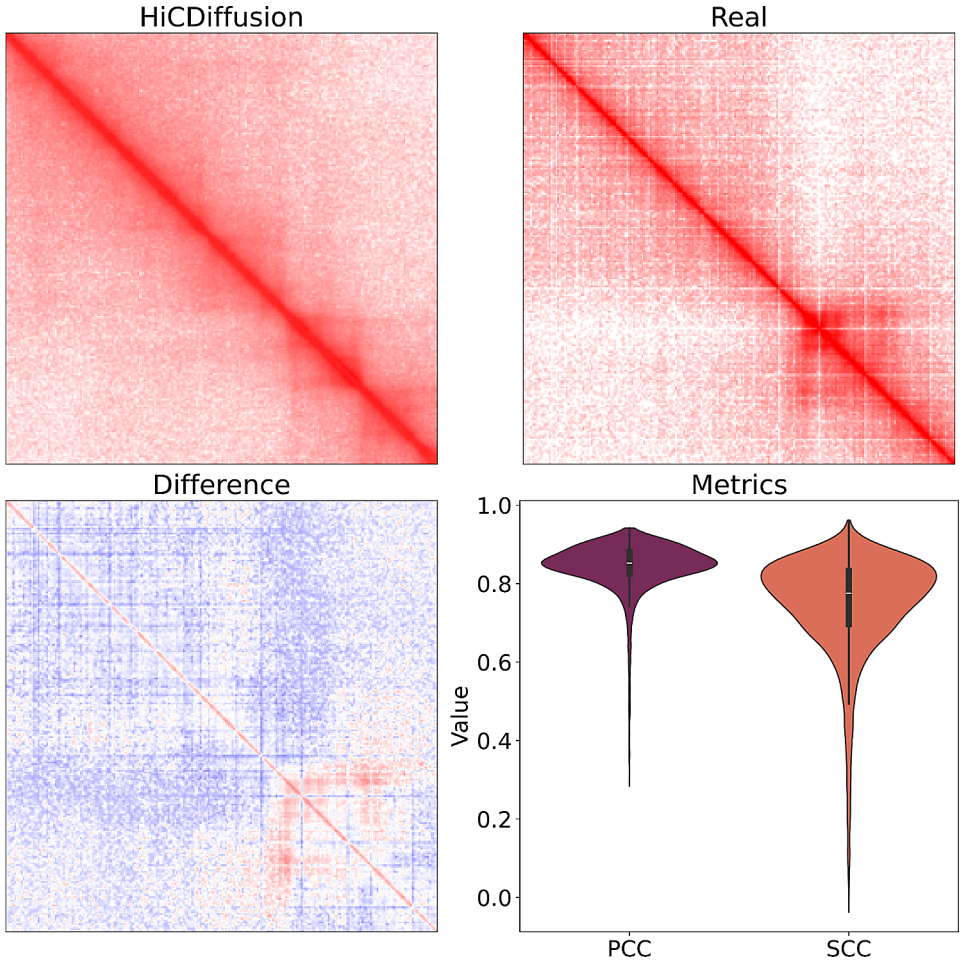

Fig. 7 Results of applying the human model to the mouse data. Pearson correlation coefficient and SCC were calculated for all heatmaps from all chromosomes. The example shows chr1:29 500 000–31 500 000 from mouse predicted by the human model

64, 64, 64, 128, 128, 256, 256, 256, 256, 256). The encoder has one additional layer - the transformer encoder - that helps to learn the appropriate context of the latent representation produced by convolutions. The transformer encoder used in our study uses 8 heads. However, we also use residual connections around the transformer to preserve original data and help with the vanishing gradient problem. Finally, we apply ReLU to the output and expand the result to the 2D matrix with 512 dimensions - by repeating the (256, 256) output vertically and horizontally. That procedure leaves us with two outputs of (256, 256, 256) - one for repeating vertically and one for horizontally. We further concatenate it to the final output of the encoder of (512, 256, 256).

The decoder architecture is composed of 2D residual convolutions - each of which is composed of 2D convolution (with 512 input and output channels, kernel size of 3, and padding of 1), batch normalisation, ReLU function, and again exactly the same 2D convolution, and batch normalisation. The output of the 2D residual block is created by applying the ReLU function to the sum of the output of the previously explained sequence and the input to the residual block - this time, unlikely in the Encoder, without any downscaling. It is crucial in the decoder to use various options for the dilation parameter - it has been done in the previously mentioned papers. It is also done in our work to propagate information through the whole heatmap. For each next layer, the dilation parameter is set to (2, 4, 8, 16, 32) and represents exponential growth. Finally, after the decoder is done, we apply an additional convolutional layer (kernel size of 3 and padding of 1) that changes dimensions from 512 to 1, leaving us with the heatmap of size (256, 256). We apply L1 loss function to match the original heatmap.

The previously explained encoder-decoder architecture is then used for further processing by using transfer learning - we pre-train the encoder-decoder and use it for the final architecture based on diffusion models. The final architecture first computes the latent representation (output of the encoder & decoder - with 512 channel), and the final heatmap from the encoder-decoder architecture. We compute the residual heatmap

by taking the difference between the real heatmap and the one produced by the encoder-decoder architecture. The network's target is predicting the residual heatmap, which shows us how to improve the prediction of the encoder-decoder architecture. The diffusion models work in two steps - the first is the model's training, and the second is inference. The training comes first so that it will be explained first as well. We take the residual heatmap and apply Gaussian noise multiple times - up to T. Then, we take such a noised image and use a denoising U-Net network. We try to predict the noise added to the image. The implementation of the diffusion model is pytorch reimplementation [30] of the original paper [20]. We have made a few changes to make it work with the Hi-C data. Firstly, the U-Net is conditioned by taking latent encoder-decoder output (512, 256, 256). It is then converted in the initial phase of using U-Net to (32, 256, 256) using convolution with a kernel equal to 7, and padding of 3. The original image provided to U-Net (noised image) is also processed using convolution to sizes of (32, 256, 256). The condition and noised image upscaled to 32 channels are then concatenated and fed into the network to represent the conditional diffusion process. We use then MSE loss function to match with the desired output. After training the diffusion model, we can predict the residual Hi-C matrix using inference mode - that is, in the form of guided diffusion using encoder-decoder output from random noise, which allows us to obtain more realistically looking Hi-C matrices.

Testing the models

All the models were tested using independent chromosomes not used in the training or validation procedure. We have calculated the Pearson correlation coefficient and SCC for each example and Fréchet inception distance to compare the quality of the images. We have run a version of C.Origami that includes only sequence (without CTCF and ATAC-Seq signals), and compared Pearson correlation coefficient scores, SCCs, and FID scores.

Fréchet inception distance (FID)

Fréchet inception distance is a metric used most often in computer vision to measure the quality of the images. It is formally defined [31] as:

$$FID = (inf_{\gamma \in \Gamma(\mu,\upsilon)} \int_{\Re^n \times \Re^n} ||x-y||^2 d\gamma(x,y))^{\frac{1}{2}}$$

Where $\Gamma(\mu, \upsilon)$ is the set of all measures on $\Re^n \times \Re^n$ with μ as first marginal factor, and υ as the second. In our work, we are using torchmetrics [32], which solves the aforementioned equation for multidimensional Gaussian distributions as:

$$FID = ||\mu - \mu_w||^2 + tr(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{\frac{1}{2}})$$

In which μ is the mean, and Σ is the variance of a multivariate normal distribution estimated from Inception v3 [33] features calculated on real data (true Hi-C matrices) and μ_w and Σ_w are corresponding values for the multivariate normal distribution estimated on the generated data (predicted Hi-C matrices). Since this metric is taking images, we need to normalise our output to the [0, 1] range and add 2 additional channels - which is done by repeating the data. The output of the procedure is called the FID score, and the lower it is, the higher quality the image is - as it is comparing the distributions of the generated images to the real ones (or, in our case, Hi-C matrices).

Comparisons to other tools

To compare the results to the C.Origami, the current state-of-the-art tool for predicting 3D interactions, we have downloaded the original software with the dataset. However, we have used two comparison models since our study focuses on pure sequence-to-interaction relationships. In the first one, we modified the tool to accept DNA sequence and train only on it. The second one was the full C.Origami architecture - the one that includes CTCF and ATAC-Seq epigenetic signals. Then, we trained both C.Origami models, with validation chromosome chr10, and testing chromosome chr15. This approach allowed us to get accurate metrics for testing and validation chromosomes and the supremum of the metrics in the case of training chromosomes. Furthermore, this model was used to predict precisely the same windows as in our models. We have calculated the Pearson correlation coefficient and SCCs for all the predictions and FID scores for all chromosomes.

Stratum-adjusted correlation coefficient (SCC)

The Pearson correlation coefficient, while used in many studies, is often criticised in the case of comparisons between Hi-C matrices. We have decided to use a stratum-adjusted correlation coefficient (SCC) [34, 35] to show the usefulness of our algorithm. We have used Python implementation of the algorithm [36]. Since our matrices are in the form of 256×256, we have used h=2 for smoothening and the maximum distance of 16 (equivalent to more than 100kbp).

TADs calling

To call TADs, we used the engine of the TADBit [37]. We have implemented loading the numpy / tensor arrays into their data structures, as the original software takes as the input. cool files. We have then calculated the insulation score for each of the heatmaps independently, and transformed it into TAD borders. We have calculated pearson

correlation of the insulation scores for all heatmaps (we have filtered out the ones that in real Hi-C data are starting/ending with zeros, as well as the ones that contain more than 5% of zero values).

Loops calling

To show that, in principle the loop calling works on the modelled data, we have implemented a proof-of-concept algorithm similar to HiCCUPS/BH-FDR [29] algorithm. However, in our case we only considered the four regions (donut, top-bottom, left-right, and left-bottom), without applying any corrections for ease of implementation, and to show that the loop calling is actually possible on the predicted data. Additionally, since HiCDiffusion does not use any epigenetic information, it is at strong disadvantage in comparison to methods that get CTCF-related information, like epigenetic tracks, or just regular peak callings.

Simulating trans-duplication

To simulate transduplication, a model trained on all chromosomes except chromosome 8 was taken into account. To illustrate that the model can correctly predict the effects of such a mutation, chr8:32 100 000–34 100 000 region was taken as an example. We have then copied the sequence of 200kbp, precisely chr8:33 450 000–33 650 000, and inserted it instead of sequence present at chr8:32 750 000–32 950 000. The model was then used to predict the in-sillico effect of such trans-duplication.

Testing on the mouse model

To test the human-derived model on the mouse data, we took the model trained on all chromosomes except chromosome 8 and tested it among all chromosomes of the mouse (as it is a completely different organism, the training/validation data is independent of the mouse). We used the mm10 reference genome and excluded centromeres and telomeres from the analysis. As the human model is trained on GM12878 data (which is lymphoblastoid cells), we have decided to use a similar cell line in mouse. We have taken B cell derived cell line [27] from the 4DN database (mcool file accession number: 4DNFIPNP9H9T, experiment accession number: 4DNESYX7AQRY). We have calculated the correlation scores similarly as in the case of the human data.

Discussion

In our study, we have created a novel deep learning architecture, HiCDiffusion, that is based on the latest advances in the field of computer vision and Artificial Intelligence. The model, composed of an encoder, transformer encoder, decoder, and a diffusion network, has surpassed the quality metric - FID score on average by 12 times (sequence-only comparison; by 11 times in case of C.Origami with epigenetics), while in the best-case chromosome, improvement was 88 times (sequence-only comparison as well; 56 in case of epigenetics-enhanced C.Origami version).

The level of the artificiality of the in silico HiC images (true quality) can also be seen with the human eye. In the case of the current models, like the aforementioned C.Origami or Akita, the model output is blurred. It can be easily distinguished from real data. In the case of our model, we have obtained quality that can be easily confused with the experimental data while obtaining all the metrics that have made previously proposed models great. Our method is the next step towards obtaining a reliable and functional universal predictor of the spatial organisation of the chromatin within the nucleus, using purely DNA sequence - which would make connecting population studies with 3D genomics much easier and, foremost - cheaper.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12864-024-10885-z.

Supplementary Material 1

Acknowledgements

Not applicable.

Author contributions

M. C. conceived and implemented the algorithm under D. P.'s supervision. M. C. performed all the experiments. M. C. wrote the manuscript under D. P.'s supervision. All authors read and approved the final manuscript.

Funding

This work has been supported by National Science Centre, Poland (2019/35/O/ST6/02484 and 2020/37/B/NZ2/03757). It was co-supported by the Polish National Agency for Academic Exchange (PPN/STA/2021/1/00087/DEC/1). The work has been co-supported by Enhpathy - "Molecular Basis of Human enhanceropathies" funded by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 860002 and National Institute of Health USA 4DNucleome grant 1U54DK107967-01 "Nucleome Positioning System for Spatiotemporal Genome Organization and Regulation". Research was co-funded by the Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme. Computations were performed thanks to the Laboratory of Bioinformatics and Computational Genomics, Faculty of Mathematics and Information Science, Warsaw University of Technology, using the Artificial Intelligence HPC platform financed by the Polish Ministry of Science and Higher Education (decision no. 7054/IA/SP/2020 of 2020-08-28).

Data availability

The GM12878 Hi-C data used for comparison with C.Origami and testing of the method is available under GEO accession number GSE63525. The preprocessing steps are described in the C.Origami paper. Mouse dataset was taken from 4DN database (mcool file accession number: 4DNFIPNP9H9T, experiment accession number: 4DNESYX7AQRY). Project name: HiCDiffusion. Project home page: https://github.com/SFGLab/HiCDiffusion Operating system(s): Platform independent. Programming language: python. Other requirements: python 3.10 higher, conda. License: MIT. Any restrictions to use by non-academics: MIT license

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 21 July 2024 / Accepted: 9 October 2024
Published online: 15 October 2024

References

1. Bouwman BAM, Crosetto N, Bienko M. The era of 3D and spatial genomics. Trends Genet. 2022;38:1062–75.
2. Chen M, Liu X, Liu Q, Shi D, Li H. 3D genomics and its applications in precision medicine. Cell Mol Biol Lett. 2023;28:19.
3. Chiliński M, Lipiński J, Agarwal A, Ruan Y, Plewczynski D. Enhanced performance of gene expression predictive models with protein-mediated spatial chromatin interactions. Sci Rep. 2023;13:11693.
4. Belokopytova PS, Nuriddinov MA, Mozheiko EA, Fishman D, Fishman V. Quantitative prediction of enhancer-promoter interactions. Genome Res. 2020;30:72–84.
5. Li W, Wong WH, Jiang R. DeepTACT: predicting 3D chromatin contacts via bootstrapping deep learning. Nucleic Acids Res. 2019;47:e60.
6. Whalen S, Truty RM, Pollard KS. Enhancer-promoter interactions are encoded by complex genomic signatures on looping chromatin. Nat Genet. 2016;48:488–96.
7. Al Bkhetan Z, Plewczynski D. Three-dimensional Epigenome Statistical Model: genome-wide chromatin looping prediction. Sci Rep. 2018;8:5217.
8. Chiliński M, Sengupta K, Plewczynski D. From DNA human sequence to the chromatin higher order organisation and its biological meaning: using biomolecular interaction networks to understand the influence of structural variation on spatial genome organisation and its functional effect. Seminars in Cell & Developmental Biology. Elsevier; 2022. pp. 171–85.
9. Chiliński M, Plewczynski D. ConsensuSV—from the whole-genome sequencing data to the complete variant list. Bioinformatics. 2022;38:5440–2.
10. Poplin R, Chang P-C, Alexander D, Schwartz S, Colthurst T, Ku A, et al. A universal SNP and small-indel variant caller using deep neural networks. Nat Biotechnol. 2018;36:983–7.
11. Van der Auwera GA, O'Connor BD. Genomics in the Cloud: using Docker, GATK, and WDL in Terra. O'Reilly Media, Inc.; 2020.
12. Cao F, Zhang Y, Cai Y, Animesh S, Zhang Y, Akincilar SC, et al. Chromatin interaction neural network (ChINN): a machine learning-based method for predicting chromatin interactions from DNA sequences. Genome Biol. 2021;22:226.
13. Schwessinger R, Gosden M, Downes D, Brown RC, Oudelaar AM, Telenius J, et al. DeepC: predicting 3D genome folding using megabase-scale transfer learning. Nat Methods. 2020;17:1118–24.
14. Zhou J. Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. Nat Genet. 2022;54:725–34.
15. Fudenberg G, Kelley DR, Pollard KS. Predicting 3D genome folding from DNA sequence with Akita. Nat Methods. 2020;17:1111–7.
16. Ji Y, Zhou Z, Liu H, Davuluri RV. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. Bioinformatics. 2021. https://doi.org/10.1093/bioinformatics/btab083.
17. Chiliński M, Halder AK, Plewczynski D. Prediction of chromatin looping using deep hybrid learning (DHL). Quant Biol. 2023;0:0.
18. Tan J, Shenker-Tauris N, Rodriguez-Hernaez J, Wang E, Sakellaropoulos T, Boccalatte F, et al. Cell-type-specific prediction of 3D chromatin organization enables high-throughput in silico genetic screening. Nat Biotechnol. 2023;41:1140–50.
19. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM. 2020;63:139–44.
20. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Adv Neural Inf Process Syst. 2020;33:6840–51.
21. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. arXiv [cs CV]. 2021;:10684–95.
22. Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M. Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv [cs.CV]. 2022.
23. Kawar B, Elad M, Ermon S, Song J. Denoising Diffusion Restoration Models. arXiv [eess.IV]. 2022;:23593–606.
24. Ren M, Delbracio M, Talebi H, Gerig G, Milanfar P. Multiscale structure guided Diffusion for Image Deblurring. arXiv [cs CV]. 2022;:10721–33.
25. Lee S, Chung H, Kim J, Ye JC. Progressive deblurring of Diffusion models for Coarse-to-fine image synthesis. arXiv [cs.CV]. 2022.
26. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. arXiv [cs. LG]; 2017.
27. Vian L, Pękowska A, Rao SSP, Kieffer-Kwon K-R, Jung S, Baranello L, et al. The energetics and physiological impact of Cohesin Extrusion. Cell. 2018;173:1165–e7820.
28. Stovner EB, Sætrom P. PyRanges: efficient comparison of genomic intervals in Python. Bioinformatics. 2020;36:918–9.
29. Rao SSP, Huntley MH, Durand NC, Stamenova EK, Bochkov ID, Robinson JT, et al. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell. 2014;159:1665–80.
30. Wang P, denoising-diffusion-pytorch. Implementation of Denoising Diffusion Probabilistic Model in Pytorch. 2020.
31. Fréchet M, Sur MF. Sur la distance de deux lois de probabilité. 1957. https://hal.science/hal-04093677/document
32. Detlefsen NS, Borovec J, Schock J, Jha AH, Koker T, Di Liello L, et al. Torchmetrics-measuring reproducibility in pytorch. J Open Source Softw. 2022. https://doi.org/10.21105/joss.04101.
33. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. arXiv [cs CV]. 2015;:2818–26.
34. Yang T, Zhang F, Yardımcı GG, Song F, Hardison RC, Noble WS, et al. HiCRep: assessing the reproducibility of Hi-C data using a stratum-adjusted correlation coefficient. Genome Res. 2017;27:1939–49.
35. Lin D, Sanders J, Noble WS. HiCRep.py: fast comparison of Hi-C contact matrices in Python. Bioinformatics. 2021;37:2996–7.
36. Matthey-Doret C. Hicreppy: Python reimplementation of hicrep with compatibility for sparse matrices. Github; 2022.
37. Serra F, Baù D, Goodstadt M, Castillo D, Filion GJ, Marti-Renom MA. Automatic analysis and 3D-modelling of Hi-C data using TADbit reveals structural features of the fly chromatin colors. PLoS Comput Biol. 2017;13:e1005665.

Publisher's note