GrapHiC: An integrative graph based approach for imputing missing Hi-C reads

Ghulam Murtaza[1] Justin Wagner[2] Justin M. Zook[2]
Ritambhara Singh[1,3*]
[1]Department of Computer Science, Brown University
[2]Material Measurement Laboratory, National Institute of Standards and Technology
[3]Center for Computational Molecular Biology, Brown University

Abstract

Hi-C experiments allow researchers to study and understand the three-dimensional (3D) organization of the genome at various structural scales. Unfortunately, sequencing costs and technical constraints severely restrict analyses that can be reliably performed at high resolutions. Existing deep learning methods to improve the resolution of Hi-C matrices utilize conventional convolutional neural network (CNN) based frameworks. These methods treat Hi-C data as an image, which imposes a strict 2D euclidean structure that, in reality, represents a 3D structure. Moreover, CNNs rely on smaller structural features to construct a global representation that might be missing altogether when the input Hi-C data is very sparse. We propose GrapHiC, a Hi-C imputation framework that utilizes a conditional Graph Autoencoder network to impute Hi-C contact matrices. We reformulate Hi-C data as a graph, where each node represents binned genomic regions, and we assign edge weights using the observed Hi-C contacts between those regions. Our formulation captures the 3D structure accurately and integrates the sparse input Hi-C signals with easy-to-obtain 1D genomic signals incorporating cell-type-specific information when the Hi-C signal is noisy or outright missing. Our experiments on five GM12878 datasets with varying sparsity levels suggest that GrapHiC provides $\sim 3.4\%$ improvement in the Hi-C similarity scores over state-of-the-art deep learning models. We observe the highest performance boost of 26% on the sparsest dataset. More importantly, our framework allows us to predict the Hi-C reads reliably in a cell line with missing Hi-C data by using a surrogate Hi-C map and auxiliary genomic signals. Thus, our generalizable framework can accelerate and make the research on 3D genomics more accessible.

*Corresponding Author. Email: ritambhara@brown.edu

1 Introduction

Research on genome organization has established its role in gene expression regulation [1], and how perturbations in this organization can lead to disease onset [2, 3, 4]. Hi-C, a high-throughput chromosome conformation capture experiment, allows researchers to understand and study genome organization. The Hi-C experiment, conceptually thought of as a function, takes in the cell genome and produces an array of paired-end reads. Each paired-end read contains sequences of DNA interacting in the 3D space. A common way to aggregate the data produced by the Hi-C experiment is to store it in a contact map of size $N \times N$. Each row and column correspond to fixed-width N windows ("bins") in the range of 1 kbps to 1 mbps depending on the number of reads tiled along the genomic axis. The values in the contact map are counts of read pairs that fall into the corresponding bins The study of these contact maps has revealed important structural features such as topologically associated domains (TADs) [5] and enhancer-promoter interactions [6] that are involved in gene regulation. Therefore, Hi-C experiments have proven to be crucial in helping us understand the interplay of the spatial structure and gene regulation.

Unfortunately, due to the quadratic scaling of reads in Hi-C protocol, most experiments produce sparse read counts that require a larger bin size (typically more than $\geq$ 40 kb [7]) to account for the experimental noise. Consequently, any downstream analysis conducted on such maps misses out on the finer structural features, such as enhancer-promoter interactions that typically occur in the 5 kbp to 10 kbp range [6]. Constructing Hi-C contact maps with sufficient resolution often requires billions of reads [6], which is often infeasible. We believe this technical limitation of the Hi-C protocol restricts the thorough analysis of the 3D conformation of DNA.

To make Hi-C analysis more accessible, researchers have proposed several deep learning-based computational methods [7, 8, 9, 10, 11, 12] to impute the value of undetected reads in the Hi-C protocol. These methods formulate Hi-C read imputation task as an image super-resolution improvement task [13]. On a high level, these methods extract a feature set by using convolution operations from the sparse input map and try to reconstruct a denser output contact map conditional on the extracted features. Subsequent models to HiCPlus [7] added additional model parameters [8, 9] or tailored the loss objective [10, 11, 12] to assist the model in extracting better features for reconstruction. A recent work [14] has shown that these methods struggle to generalize to real-world sparse Hi-C contact maps. Specifically, the performance of these methods degrades as the sparsity of the input contact maps increases.

We believe that these methods fail to generalize for two primary reasons. 1) The image formulation of the Hi-C data enforces a strict 2D euclidean structure; whereas in reality, Hi-C represents a 3D structure. This formulation forces the Convolutional Neural Network (CNN) model to attend to features that are local in DNA distance space rather than their distance in 3D space and consequently causes the model to miss out on global structural properties. 2) In the case of sparse Hi-C inputs, the local features that CNNs attend to might be entirely missing; in that case, whatever internal representation it constructs would be unrealistic, and hence the imputed maps could be inaccurate.

To solve these limitations, we propose GrapHiC to impute Hi-C contact maps, which makes three significant contributions. First, GrapHiC reformulates the Hi-C data as a position-aware graph where each node represents a genomic region. The observed reads between those regions correspond to the weight of the edge connecting those regions. The graph formulations has been used previously for gene expression prediction [15, 16], chromatin state prediction [17], Micro-C prediction [18], and many other tasks [19, 20, 21, 22, 23]. However, none of the existing methods have considered the importance of the positional information of the node representing genomic region. Using graph-based positional encodings as features, GrapHiC establishes the importance of positional information in imputing missing contacts. Secondly, we propose to solve the problem of sparse Hi-C input by using cheaper-to-obtain ChIP-seq signals from targeting proteins known to mediate chromatin structure. We believe this auxiliary data assists our model in imputing Hi-C reads, mainly when input Hi-C data is missing. Lastly, GrapHiC proposes a generative graph-autoencoder that first encodes the input graph into a latent representation. This latent representation encodes the likelihood of two nodes interacting in 3D space conditional on their distance on a DNA string, their structural neighborhood, and the proteins mediating the structure around them. Then the graph decoder utilizes this latent representation to impute a denser Hi-C contact map.

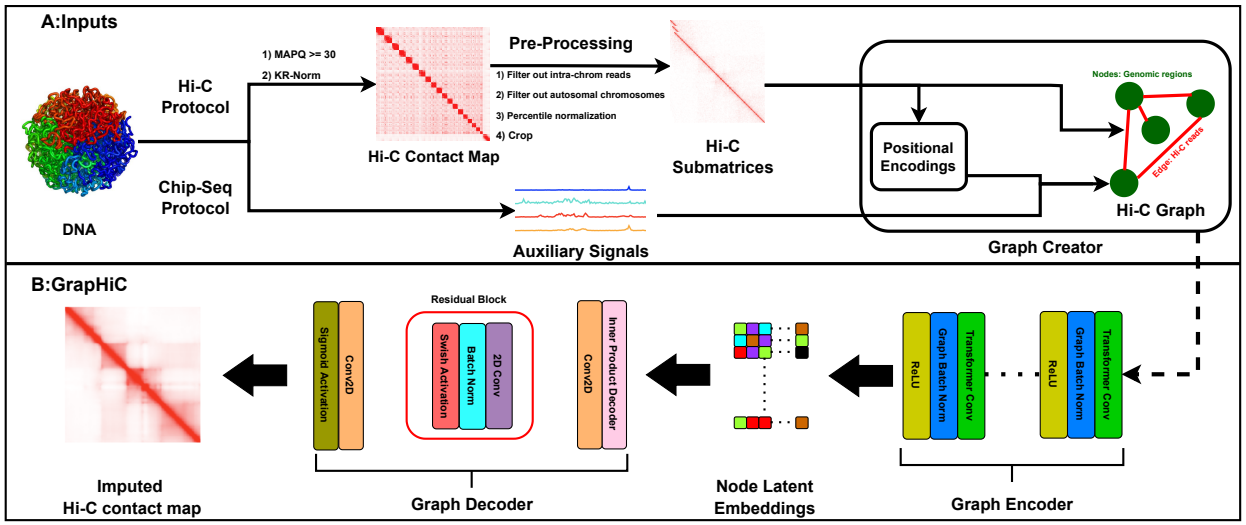

Figure 1: **Overview of GrapHiC pipeline: A** Inputs and pre-processing pipeline: In this portion of the pipeline, we normalize Hi-C contact maps and create Hi-C graph using Hi-C data, auxiliary genomic signals and positional encodings. **B** GrapHiC: we use the sparse input Hi-C graph and generate a denser Hi-C graph using a graph generative model.

Our results show that GrapHiC outperforms HiCNN (the best state-of-the-art method [14]) on a Hi-C similarity metric by 3.4% on average on GM12878 Hi-C datasets, with the most significant improvement of 24% for a dataset downsampled by a factor of 1/100. Similarly, the 3D chromatin models reconstructed with GrapHiC show better accuracy in 4 out of 5 sparse GM12878 datasets. Moreover, GrapHiC provides an opportunity to accurately impute cell-type specific contact maps with auxiliary genomic signals and a surrogate Hi-C contact map. Thus, we believe that GrapHiC is a helpful tool for imputing Hi-C reads for cell types with missing Hi-C data. Lastly, our ablation analyses show that although GrapHiC capitalizes on additional auxiliary genomic signals, it can still recover finer 3D structures missing in the input Hi-C with as little as one genomic signal.

2 Related Work

Existing state-of-the-art methods model the Hi-C imputation task as an image super-resolution improvement task [13]. They utilize CNNs to improve the Hi-C resolution, treating a contact map as an image. CNNs extract a feature set from the input low-read-count contact map and then use those features to predict a high-read-count contact map. For example, HiCPlus [7], utilized a 3-layer CNN and optimized this model using a Mean Squared Error (MSE) loss. HiCNN [8] extended this framework to a 54-layer architecture with residual layers to stabilize training and improve performance. HiCNN2 [9] further expanded to an ensemble of models that contained HiCNN, HiCPlus, and a modified version of Very Deep Super Resolution (VSDR) framework [24] to assist the model in learning hierarchical features to impute realistic Hi-C contacts. Alternatively, HiCGAN [10] argued for improving the loss term to help the CNN models to learn better feature representations. HiCGAN employed Generative Adversarial Network [25] (GAN) to jointly optimize the generator that imputes missing reads and the discriminator that generates higher loss on unrealistic contact maps. DeepHiC [11] extended HiCGAN's loss term by adding Perceptual loss [26] and TVLoss [27] to assist the discriminator loss in forcing the generator to predict more realistic contact maps. VeHiCLe[12] improved on DeepHiC by replacing Perceptual loss with Hi-C domain-specific similarity loss. Moreover, VeHiCle added Insulation loss [28] to force the model to attend to the underlying biological structures

(specifically topologically associated domains) to impute biologically informative Hi-C reads.

A recent work by Murtaza *et al.* [14] evaluated these methods on real-world sparse Hi-C datasets and found that the performance of these methods degraded substantially as the sparsity of the input increased. We have included a result from this work in the Supplementary Section S1 for ease of reference. Based on these observations, we argue that formulating the Hi-C read imputation problem as an image super-resolution problem imposes a strict two-dimensional euclidean structure on the Hi-C data that, in reality, represents a three-dimensional structure. This two-dimensional structure forces the Hi-C data between DNA regions to be ordered based on their positions on the DNA rather than their distance in 3D space. We hypothesize this representation can result in CNNs missing out on the global structure, given they are forced to attend to smaller local features. In the case of sparse inputs, these features might be outright missing, and any imputation in that scenario would not generalize. We propose reformulating the Hi-C as a graph to learn better representations. This representation allows us to keep our model small (two encoder layers followed by five residual layers in the decoder) rather than relying on deeper networks such as 54-layer HiCNN [8] or using complex loss functions to learn those representations.

Another set of existing methods use data from cheaper-to-conduct experiments, capturing 1D genomic information, to impute Hi-C data which is typically much more expensive to obtain. Akita [29] uses DNA sequence information to predict Hi-C contact maps. However, relying on only DNA sequences as inputs will fail to capture the cell-specific differences in the data arising from epigenomic changes. To effectively capture these cell-specific differences, HiCReg [30] utilized ChIP-seq data targeting 14 genomic signals ranging from histone modifications, DNA accessibility, and Transcription Factor (TF) binding sites to predict Hi-C contacts. Although HiCReg does capture cell type-specific information, it misses out on crucial structural information in Hi-C data. We combine both cheaper-to-acquire ChIP-seq experiments with sparse architectural conformation data from Hi-C experiments in a single framework to impute Hi-C reads. This approach allows us to minimize the financial cost of acquiring accurate high-read-count Hi-C datasets.

3 Methods

3.1 Datasets and Preprocessing

Following existing Hi-C read imputation methods [7, 8, 10, 11, 12, 31] we used GM12878 and K562 cell line datasets from Rao *et al.* [6] as our target high-read-count matrices. We curated an additional six Hi-C datasets from ENCODE [32] and NCBI [33] portals as our real-world low-read-count datasets with a sparsity of 1/10 to 1/100 compared to the target Hi-C datasets. We filtered out reads with MAPQ value < 30 to remove reads that may be incorrectly mapped and binned the remaining reads at 10KB resolution to create our two-dimensional contact maps. We performed a KR-normalization of these contact maps to balance reads across all the bins. Moreover, we filtered out all inter-chromosomal reads, including reads from X, Y chromosomes, and mitochondria, and divided the contact map for the remaining autosomal chromosomes. Finally, we normalized each contact map between 0 and 99.75th percentile value and cropped into sub-matrices of size 200×200.

For our auxiliary genomic signals, we used the 14 one-dimensional inputs used by HiCReg [30]. HiCReg curated a collection of signals targeting epigenetic factors, transcription factors, and chromatin accessibility markers, essentially forming a comprehensive set of features related to the 3D organization of the DNA. Similar to our Hi-C data, we binned all the data in 10kbp resolution, normalized them in the 0 to 99.75th percentile range, and cropped them into sub-ranges of size 200 that aligned with Hi-C submatrices. We divided chromosome [1-8, 12-18] as training, [8, 10, 19, 22] as validation and [9, 11, 20, 21] as testing chromosome sets. We have summarized our inputs' pre-processing workflow in the Figure 1 **A** and their statistics in the Supplementary Tables S1, S2, and S3.

3.2 GrapHiC

Our GrapHiC framework can be divided into two main components - 1) a graph creator module and 2) a graph generative model. The first component - **graph creator** - is responsible for graph formulation. The graph creator, shown in Figure 1 **A**, constructs a Hi-C graph $G = (V, E)$ with $N = |V|$ nodes, each corresponding to a genomic region and E edges with weights corresponding to the observed reads of the contact map. Most of the existing Hi-C graph formulations in the literature [19, 34, 21, 20] have a similar setup and assume that all the nodes (genomic regions) have the same properties. The graph G created by these methods contains all the 3D structural information provided in a given contact map. However, those formulations miss out on the crucial positional information of these nodes with respect to each other. This positional information is essential because analysis of chromatin structure [6] has shown that genomic regions close by are more likely to interact. To encode this information, we utilize a recently proposed graph positional encoding scheme [35] that assigns eigenvectors of the laplacian of the graph as node attributes. These positional encodings represent distance of the nodes in the graph space. The idea is similar to the positional encodings used in Transformers [36] that describe the location or position of an entity in a sequence so that each position is assigned a unique representation. Our graph representation provides us with a unique opportunity to augment the Hi-C data with auxiliary data curated from cheaper-to-conduct ChIP-Seq experiments targeting proteins that are well known to mediate chromatin structure. We concatenate the auxiliary genomic signals (used by HiCReg [30]) with the positional embeddings in our node attribute vector to impute Hi-C reads, particularly in regions where Hi-C data is sparse.

The second component is the **graph generative model** that follows the Graph Autoencoder (GAE) [37] architecture, as shown in Figure 1 **B**, with updates to adapt it to Hi-C read imputation task. As a default, GAE has two major components: a Graph Encoder and a Graph Decoder. The Graph Encoder constructs latent representations at each node, where these representations capture each node's functional relationship in the context of its attributes with its neighborhood. The Graph Decoder then predicts the likelihood of a connection between nodes using an inner product of each pair of nodes' latent representations. This decoding formulation assumes that nodes with similar latent representations are more likely to be connected in the generated graph. This autoencoder formulation assumes that input and output graphs are unweighted, which is not the case in Hi-C, and thus, we made some key additions to the framework.

We replaced the graph convolutions [38] with graph transformer convolutions [39] in the graph encoder. Graph transformer convolutions learn the new representation of the node i, x'_i, by aggregating over all N nodes with x_N attributes as follows:

$$x'_i = W_1 x_i + \sum_{j \neq i} a_{i,j}(W_2 x_j + W_6 e_{i,j}) \tag{1}$$

$$a_{i,j} = softmax(\frac{(W_3 x_i^T)(W_4 x_j + W_6 e_{i,j})}{\sqrt{d}}) \tag{2}$$

Here, $e_{i,j}$ are edge weights connecting node i and j, $a_{i,j}$ are attention coefficient attributes, d is a scaling parameter and W_k are learnable parameters

Transformer convolution operation updates the node attributes of the current node by combining the current node's attributes x_i with attributes of neighboring nodes $j \neq i$ and edge weight connecting them $e_{i,j}$ (Eq. (1)). This combination is scaled by their attention coefficient $a_{i,j}$ (Eq. (2)). Attention allows the model to focus on the most relevant node interactions by weighing them differently, with weights ranging between $0 - 1$ due to the $softmax$ function. In the biological context, this latent representation encodes the likelihood of two nodes interacting in 3D space conditional on their distance on the DNA string, their structural neighborhood, and the proteins mediating the structure around them.

The original implementation of the Graph Decoder gave us a probability of the existence of an edge between two nodes. However, we need the graph decoder to predict edge weight instead. To that end, we added a convolution layer that transforms the probability of an edge between nodes to their predicted weights. We also stacked residual convolutional blocks, similar to DeepHiC [11], that provide additional parameters to impute realistic Hi-C reads. Lastly, we replaced the training objective from Binary Cross

Entropy (BCE) to Mean Squared Error (MSE) loss to optimize the encoder and decoder to impute realistic Hi-C reads. MSE loss is defined as:

$$L_{MSE}(x, y) = \sum_{i=1}^{N} (x_i - y_i)^2 \tag{3}$$

We picked a simple MSE loss because it has been shown [40] that having complex nuanced loss functions increases the likelihood of the model overfitting to the training objective. This problem is more relevant in the graph domain because graph spectral convolutions tend to overfit the underlying geometry [41].

We summarize our training process in the Supplementary Section S3. We evaluated positional encoding methods, merging positional encodings and auxiliary signals, graph convolution operations, the value of Hi-C data in imputing Hi-C reads, node embedding size, loss function, encoder blocks, and residual blocks. In summary, we found an architecture with one encoder block and five decoder blocks with 32-node embedding dimensions optimal for keeping parameters small while getting a state-of-the-art performance.

3.3 Baselines

We retrained both HiCNN[8] and HiCReg[30] with our pre-processed Hi-C data, as mentioned earlier, to ensure that the imputed Hi-C reads followed the same distribution. Given that HiCNN predicts outputs of size 40×40 and HiCReg imputes reads for a single bin, we first imputed reads for the entire chromosome. Next, we cropped out sub-matrices of 200×200 to ensure a fair comparison against GrapHiC's imputed samples. We kept the same training, validation, and testing chromosomes while retraining both HiCNN and HiCReg. We changed the optimizer from SGD to ADAM for HiCNN and updated the batch size to utilize all the GPU memory available to improve the training time. We did not change the architecture or rest of the hyperparameters for either HiCNN or HiCReg.

3.4 Evaluation metrics

We present results of two metrics to estimate the quality of the imputed Hi-C contact maps. First, we compare the Hi-C similarity using GenomeDISCO [22], which utilizes random graph walks of increasing lengths to compare the similarity of the underlying structural features. GenomeDISCO produces a similarity score between 0 and 1, where the higher score represents higher similarity. Secondly, we use 3D reconstruction accuracy to estimate the structural similarity of the reconstructed chromatin from the imputed Hi-C reads. We construct 3D models from the Hi-C contact maps using 3DMax [42] with its default parameters and compare the reconstruction accuracy of the predicted against a target using TM-Score [43]. A higher TM-Score corresponds to greater similarity between the structural features. In the domain of protein structure analysis, a score greater than 0.5 corresponds to structures having similar underlying features [43]. In the Supplementary Evaluations in sections S3, S4 and S5. we include additional metrics used in other studies such as HiCRep[44], HiC-Spector [23], QuASAR-Rep[45], MSE, Pearson's Correlation Coefficient (PCC) and Structural Similarity Index Metric (SSIM) [46].

4 Results

4.1 GrapHiC generalizes to a broad range of input sparsity-levels in Hi-C data

We compare the performance of GrapHiC against two relevant state-of-the-art methods. First, we compare our performance with HiCNN [8] because a recent evaluation [14] showed that HiCNN provides the best imputation generalizability across a broad range of sparse real-world Hi-C datasets. Second, we compare our performance with HiCReg [30], a method that utilizes the same 1D genomic signals without a sparse Hi-C contact map to impute Hi-C reads. We summarize our findings in Figure 2.

As shown in Figure 2 **A**, we observe that GrapHiC generates Hi-C contact matrices that are more visually similar. We choose to visualize this region because it depicts a wide range of architectural features

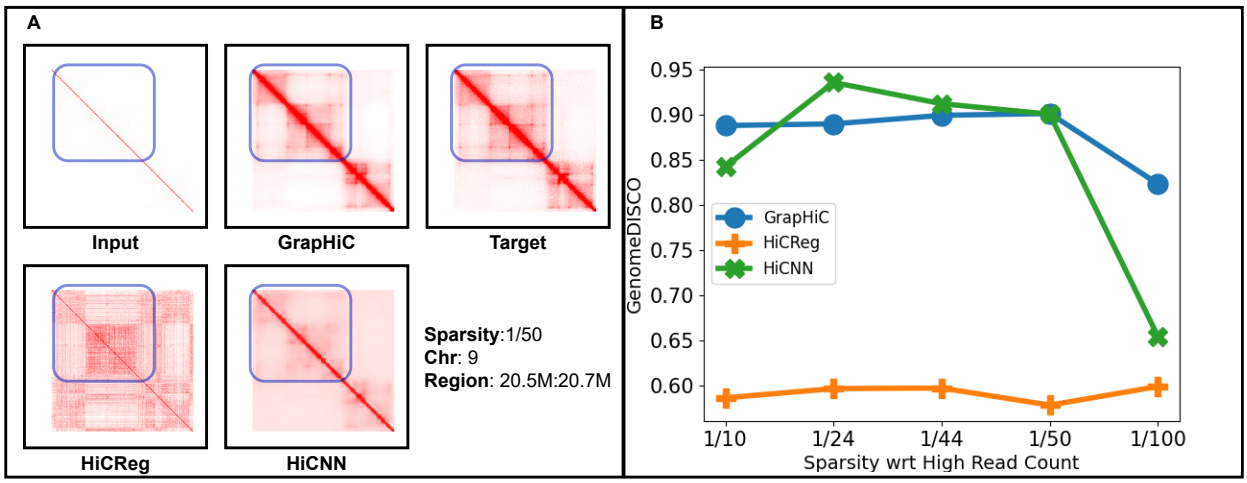

Figure 2: **Comparison to HiCNN and HiCReg on for sparsity levels and GenomeDISCO metric:** In this experiment, we compare the performance of GrapHiC against HiCNN (that uses Hi-C data only) and HiCReg (that uses auxiliary genomic information only). Qualitatively **A**, GrapHiC shows better visual reconstruction, specifically in the region highlighted blue. We visualize the same region for rest of the sparsity in supplementary section. In our quantitative results **B**, we find that on the GenomeDISCO metric, GrapHiC outperforms HiCNN in 3 out of 5 sparsity levels showing the most improvement in 1/100 sparse case.

(like loops and TADs), and both HiCNN and GrapHiC have the most similar score for 1/50 sparsity-level of inputs. Specifically for the section highlighted in blue, GrapHiC's imputed contacts map shows a clear presence of chromatin loops [47] at the boundaries of TADs [5] in comparison to HiCNN. For HiCNN, we observe that most of the finer structural features are muted away because either the signal is missing or noisy. For quantitative analysis, we aggregate the GenomeDISCO scores across all genomic regions in the test set chromosomes (Figure 2 **B**). We observe that GrapHiC outperforms HiCReg on all levels of sparsity and HiCNN on three out of five sparsity levels. Furthermore, our results show that GrapHiC provides a substantially better similarity score on 1/100 sparse input. It achieves a score of **0.8199** compared to **0.6577** from HiCNN's imputations. These results provide empirical evidence that GrapHiC can learn robust global and local representations regardless of sparsity. Results and visualizations for the rest of the reproducibility metrics – HiC-Spector, HiC-Rep, and QuaSAR-Rep, are in the Supplementary Section S4. Our visualizations show that GrapHiC imputations are visually more similar to high-read-count maps for all sparsity levels. We also observe better aggregate reproducibility score improvement of 5.4% and 16.62% over HiCNN for HiCRep and HiC-Spector scores. We notice a small (1.8%) decrease in QuASAR-Rep scores against HiCNN for the first four sparsity-levels. QuASAR-Rep failed to produce a reproducibility score for HiCNN samples imputed from 1/100 sparse dataset, where we generally observe the most improvement for GraphHiC.

4.2 GrapHiC imputations generate accurate 3D chromatin structures

We conduct a downstream analysis of the imputed Hi-C contact matrices to establish that the predicted reads from the GrapHiC model can assist with reliable 3D modeling of the chromatin architecture. VeHiCLe paper [12] performed an identical experiment to establish that VeHiCLe produces biologically informative Hi-C samples. If the 3D structure produced from imputed reads is similar to that produced from high-read-count Hi-C data, one can assume that the underlying biological properties would be conserved because these properties predominantly rely on 3D conformation. We use the 3DMax tool [42] to construct the 3D structure for the 200×200 sub-matrices for all the imputed Hi-C matrices and compare their reconstruction accuracy using TM-Score. Analysis of TM-Scores in the protein domain have demonstrated that a score

greater than 0.5 suggests a highly similar underlying structural geometry [43]. As done previously [12], we assume that TM-score behaves the same way for chromatin structure comparison as it would for proteins. To visualize these models we utilize PyMol toolkit with the "technical" preset. We summarize results of our analysis in Figure 3 and Table 1.

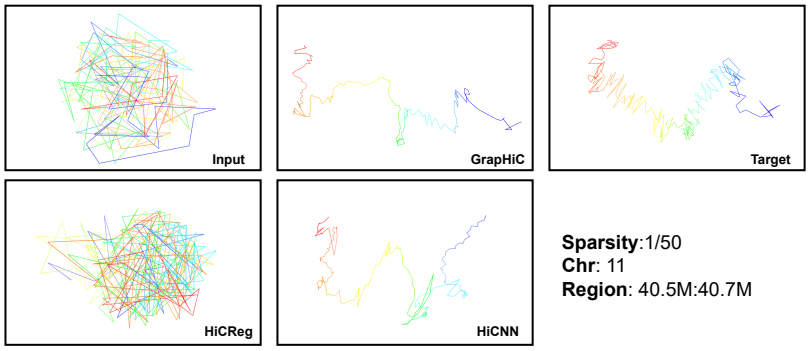

Figure 3: **Biologically informative 3D reconstruction:** In this experiment, we establish that imputed Hi-C contact maps are biologically informative since they exhibit a similar 3D structure to the high-read-count Hi-C contact maps. Our qualitative results suggest that both and HiCNN show reasonable reconstruction accuracy

Sparsity wrt High Read Count	1/14	1/24	1/44	1/50	1/100
GrapHiC	**0.7872**	0.7826	**0.7817**	**0.7907**	**0.6483**
HiCNN	0.776	**0.8087**	0.7707	0.7546	0.6212
HiCReg	**0.4103**	0.4103	0.4119	0.4098	0.4117

Table 1: **3D reconstruction aggregate score** Our quantitative analysis aggregate over all samples suggests that GrapHiC's imputed 3D structures show better structural similarity than HiCNN in four datasets, with HiCNN outperforming on one of the less sparse 1/24 datasets.

Visual analysis of the generated models on one 2 mbp region shows that HiCReg struggles to recover any meaningful representation while both HiCNN and GrapHiC's predicted structures show good similarity with target 3D structure. Our aggregate TM-score analysis over all 2 mbp samples across sparsity levels, as summarized in Table 1, shows that GrapHiC outperforms HiCNN in four out of five datasets supporting our results from the previous experiment. Based on these results we assert that GrapHiC can impute Hi-C contact maps that are biologically informative.

4.3 GraphHiC predicts cross cell-type Hi-C contact maps

In this section, we test the generalizability of GrapHiC for cross-cell-type Hi-C imputation. We use K562 as our target cell type for high-read-count Hi-C prediction. Our experiment is set up as follows: (1) First, we predict missing Hi-C reads in K562 conditional on a low-read-count K562 Hi-C contact map and K562 auxiliary genomic signals. (2) Next, we predict missing Hi-C reads with K562 auxiliary genomic signals but with a surrogate contact map from the GM12878 cell line. The purpose of this experiment is twofold: (1) to establish that GrapHiC generalizes to inputs from other cell lines, and (2) GrapHiC can impute missing reads for cell samples that do not have any Hi-C data available using the 1D genomic signals. We summarize our findings of this experiment in Figure 4. Our imputed contact maps show a denser distribution of reads

compared to the K562 high-read-count dataset [6], we believe our qualitative results shown in Table 2 hold limited value.

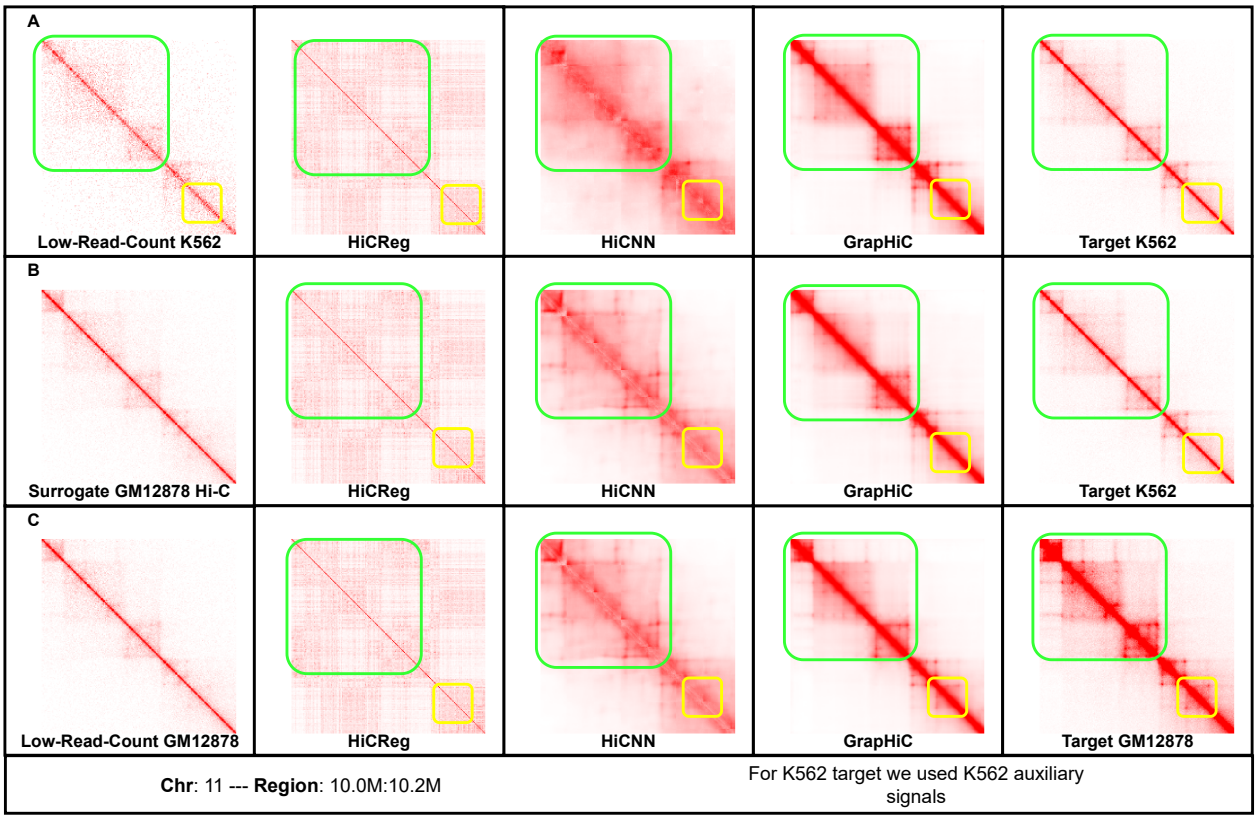

Figure 4: **Generalization to other cell lines:** In this experiment, we establish that GrapHiC generalizes to inputs from other cell lines. In tile **A**, we impute K562 reads conditional on low-read-count K562 Hi-C and K562 auxiliary signal. In tile **B**, we impute K562 reads conditional on a surrogate GM12878 Hi-C and K562 auxiliary signals. In tile **C**, we impute GM12878 reads conditional on GM12878 Hi-C and GM12878 auxiliary signals. Our results show that the region highlighted green is conserved across both cell lines, and GrapHiC accurately imputes missing reads in that region. Moreover, we show that regions highlighted in yellow do not have loops in the K562 high-read-count cell line, and GrapHiC can predict their absence in both low-read-count and surrogate Hi-C cases.

Figure 4 shows an imputed contact map for a region in Chr-11 under three different cases. In panel **A**, we show the predicted Hi-C contact map conditional on a low-read-count with 1/14 sparsity K562 (the most sparse experimental K562 dataset with respect to high-read-count K562 dataset that is available) Hi-C contact map and K562 auxiliary signals. Here we essentially capture the case when we have a low-read-count Hi-C dataset available for a cell line. In panel **B**, we show the predicted Hi-C contact map conditional on a surrogate GM12878 1/10 sparse contact map and K562 auxiliary signals. This case is analogous to imputing Hi-C contact maps when a Hi-C dataset is unavailable for the target cell line. In panel **C**, we show the predicted Hi-C contact map for the GM12878 cell line. We argue that in cases where architecture is similar across the cells (highlighted in green), GrapHiC can accurately impute the missing reads conditional on both low-read-count or a surrogate contact map. Moreover, when architectural features differ, GrapHiC can still accurately impute Hi-C cell-specific reads regardless of the type of input contact map. For example, the region highlighted yellow has loops in the GM12878 cell line while they are absent in K562 cell line GrapHiC

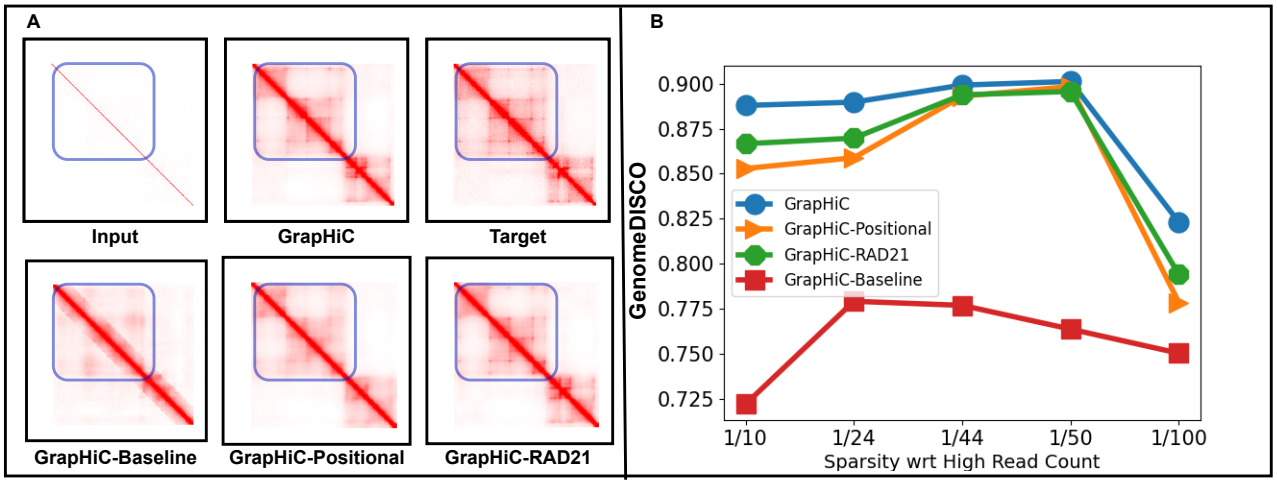

Figure 5: **Ablation Analysis:** In this experiment we try to provide insights into different components of GrapHiC and how they help GrapHiC to perform better. We show that adding the positional encodings add the most performance **B** over the baseline GrapHiC that has neither positional encodings nor auxiliary signals. Moreover, our qualitative analysis suggests that adding RAD21 signal allows GrapHiC to detect loop positions more accurately as highlighted in the blue region.

can accurately predict their absence in tile **A** and **B** that uses K562 specific auxiliary signals. Compared to our related works, HiCNN and HiCReg, we show better visual reproducibility because we can recover finer features more reliably in all three cases. Lastly, although we believe that qualitative results hold limited value in this analysis, in our aggregate GenomeDISCO score, we observe that GrapHiC performs better than both HiCNN and HiCReg. Furthermore, we find that GrapHiC shows a very similar GenomeDISCO score in both K562 low-read-count cases and GM12878 surrogate cases providing additional support for our takeaways from qualitative results.

	K562-Low-Read-Count	**GM12878-Surrogate**
GrapHiC	**0.8461**	**0.8433**
HiCNN	0.6643	0.835
HiCReg	0.5645	0.5621

Table 2: We evaluate how well our cross-cell type predictions perform compared to each other on four reproducibility metrics. Applicability of these metrics is questionable since the target high-read-count matrices are sparser than the generated matrices.

4.4 Ablation analysis highlights the importance of novel positional encoding

As our last set of experiments, we conduct an ablation on our graph formulation on GM12878 datasets to validate the utility of additions we made to Hi-C graph formulation. We retrain three new versions of GrapHiC – 1) GrapHiC without positional information and auxiliary signals in the Hi-C graph, 2) GrapHiC with positional information only, and 3) GrapHiC with positional information and only one auxiliary signal RAD21 that we empirically found to be the single most informative of the fourteen we considered, as summarized in Supplementary Table S5. Figure 5 shows the ablation analysis results.

GenomeDISCO scores, shown in panel **B**, for these four versions across five levels of sparsity suggests that adding the positional information to the Hi-C graph (shown by the line 'GrapHiC-Positional') improves

the performance substantially (by 15.13%) over the baseline Hi-C graph formulation (depicted by the line 'GrapHiC-Baseline'). Moreover, aggregate GenomeDISCO scores suggest that adding auxiliary signals improves performance as we move from adding one in 'GrapHiC-RAD21' to the proposed GrapHiC version. Lastly, upon visual inspection of imputed contact maps, in panel A, we find (in the region highlighted in blue) that both GrapHiC and GrapHiC-RAD21 have sharper features corresponding to chromatin loops [47] around the TAD boundaries compared to GrapHiC-Positional. These features are known to be mediated by RAD21 protein [48] and consequently provide evidence that adding auxiliary signals assists GrapHiC in predicting features that might have otherwise been missing sparse Hi-C inputs.

5 Discussion

In our framework, GrapHiC, we have presented a robust method to formulate Hi-C data as a graph. On top of being a more accurate representation of chromatin structure, GrapHiC provided us a unique opportunity to merge multiple measurement modalities such as DNA accessibility information, TF binding sites, Histone Modifications, and spatial arrangement of DNA into a single representation learning framework. We believe this formulation can be utilized for tasks other than Hi-C read imputation, such as cell-phase identification, cell clustering, chromatin loops, and TAD boundary identification because GrapHiC constructs an information-rich representation.

Our evaluations show that GrapHiC is resilient against the sparsity of the input Hi-C data and can reliably impute Hi-C reads that are biologically informative even in cases when Hi-C experiment data is unavailable. We believe GrapHiC can extend the efforts of Avocado [49] to impute missing Hi-C experiments for all the cell lines on the ENCODE portal [32]. However, there are still some unanswered questions related to how GrapHiC generalizes to Hi-C data generated by variants of Hi-C protocol or single-cell HiC. Moreover, it is unclear on what sparsity or coverage GrapHiC fails and what guarantees we can provide for Hi-C data in regions containing structural variants, which are especially common in some types of cancer.

Regardless, GrapHiC democratizes access to Hi-C data by allowing researchers to impute dense Hi-C contact maps either conditional on their low-read-count Hi-C data or just using signals from ChIP-seq experiments. We believe that showing our method generalizes to a wide range of sparsity takes away the guesswork and provides a unified, robust approach to impute missing Hi-C reads that also work across different cell types.

Acknowledgements. We would like to thank the Brown Computational Biology Core for their help in setting up and running different frameworks. We would also like to acknowledge contributions of all Singh Lab members in providing invaluable feedback on various aspects of this work. Specifically, Atishay Jain, Hyeyeon Hwang and Pinar Demetci for their direct feedback on graph formulation and Hi-C data.
Certain commercial equipment, instruments, or materials are identified to specify adequately experimental conditions or reported results. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment, instruments, or materials identified are necessarily the best available for the purpose.
Funding. Ghulam Murtaza's effort on this project was funded by the NIST PREP grant GR5245041. Ritambhara Singh's contribution to the work is supported by NIH award 1R35HG011939-01.

References

[1] Mora, A., Sandve, G. K., Gabrielsen, O. S. & Eskeland, R. In the loop: Promoter–enhancer interactions and bioinformatics. *Briefings in Bioinformatics* (2015).

[2] Lauderdale, J. D., Wilensky, J. S., Oliver, E. R., Walton, D. S. & Glaser, T. 3' deletions cause aniridia by preventing pax6 gene expression. *Proceedings of the National Academy of Sciences* **97**, 13755–13759 (2000).

[3] Fantes, J. *et al.* Aniridia-associated cytogenetic rearrangements suggest that a position effect may cause the mutant phenotype. *Human molecular genetics* **4**, 415–422 (1995).

[4] Kleinjan, D. A. *et al.* Aniridia-associated translocations, dnase hypersensitivity, sequence comparison and transgenic analysis redefine the functional domain of pax6. *Human molecular genetics* **10**, 2049–2059 (2001).

[5] Dixon, J. R. *et al.* Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* **485**, 376–380 (2012).

[6] Rao S, e. a. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* **159**, 1665–1680 (2014).

[7] Zhang, Y. *et al.* Enhancing hi-c data resolution with deep convolutional neural network hicplus. *Nature communications* **9**, 1–9 (2018).

[8] Liu, T. & Wang, Z. HiCNN: a very deep convolutional neural network to better enhance the resolution of Hi-C data. *Bioinformatics* **35**, 4222–4228 (2019).

[9] Liu, T. & Wang, Z. Hicnn2: Enhancing the resolution of hi-c data using an ensemble of convolutional neural networks. *Genes* **10** (2019).

[10] Liu, Q., Lv, H. & Jiang, R. hicgan infers super resolution hi-c data with generative adversarial networks. *Bioinformatics* **35**, i99–i107 (2019).

[11] Hong, H. *et al.* Deephic: A generative adversarial network for enhancing hi-c data resolution. *PLOS Computational Biology* **16** (2020).

[12] Highsmith, M. & Cheng, J. Vehicle: A variationally encoded hi-c loss enhancement algorithm. *Scientific Reports* (2020).

[13] Dong, C., Loy, C. C., He, K. & Tang, X. Image super-resolution using deep convolutional networks (2015). URL `https://arxiv.org/abs/1501.00092`.

[14] Murtaza, G., Jain, A., Hughes, M., Varatharajan, T. & Singh, R. Investigating the performance of deep learning methods for hi-c resolution improvement. *bioRxiv* (2022). URL `https://www.biorxiv.org/content/early/2022/08/05/2022.01.27.477975`.

[15] Ashoor, H. *et al.* Graph embedding and unsupervised learning predict genomic sub-compartments from hic chromatin interaction data. *Nature Communications* **11** (2020).

[16] Karbalayghareh, A., Sahin, M. & Leslie, C. S. Chromatin interaction–aware gene regulatory modeling with graph attention networks. *Genome Research* (2022).

[17] Lanchantin, J. & Qi, Y. Graph convolutional networks for epigenetic state prediction using both sequence and 3d genome data. *Bioinformatics* **36**, i659–i667 (2020).

[18] Feng, F., Yao, Y., Wang, X. Q. D., Zhang, X. & Liu, J. Connecting high-resolution 3d chromatin organization with epigenomics. *Nature communications* **13**, 1–10 (2022).

[19] Lee, D.-I. & Roy, S. Grinch: Simultaneous smoothing and detection of topological units of genome organization from sparse chromatin contact count matrices with matrix factorization. *Genome Biology* **22** (2021).

[20] Ashoor, H. *et al.* Graph embedding and unsupervised learning predict genomic sub-compartments from hic chromatin interaction data. *Nature Communications* **11** (2020).

[21] Fotuhi Siahpirani, A., Ay, F. & Roy, S. A multi-task graph-clustering approach for chromosome conformation capture data sets identifies conserved modules of chromosomal interactions. *Genome Biology* **17** (2016).

[22] Ursu, O. *et al.* Genomedisco: A concordance score for chromosome conformation capture experiments using random walks on contact map graphs. *Bioinformatics* **34**, 2701–2707 (2018).

[23] Yan, K.-K., Yardımcı, G. G., Yan, C., Noble, W. S. & Gerstein, M. Hic-spector: A matrix library for spectral and reproducibility analysis of hi-c contact maps. *Bioinformatics* **33**, 2199–2201 (2017).

[24] Kim, J., Lee, J. K. & Lee, K. M. Accurate image super-resolution using very deep convolutional networks (2015). URL `https://arxiv.org/abs/1511.04587`.

[25] Goodfellow, I. J. *et al.* Generative adversarial networks (2014). URL `https://arxiv.org/abs/1406.2661`.

[26] Johnson, J., Alahi, A. & Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution (2016). URL `https://arxiv.org/abs/1603.08155`.

[27] Javanmardi, M., Sajjadi, M., Liu, T. & Tasdizen, T. Unsupervised total variation loss for semi-supervised deep learning of semantic segmentation (2016). URL `https://arxiv.org/abs/1605.01368`.

[28] Crane, E. *et al.* Condensin-driven remodelling of x chromosome topology during dosage compensation. *Nature* **523**, 240–244 (2015).

[29] Fudenberg, G., Kelley, D. R. & Pollard, K. S. Predicting 3d genome folding from dna sequence with akita. *Nature Methods* **17**, 1111–1117 (2020).

[30] Zhang, S., Chasman, D., Knaack, S. & Roy, S. In silico prediction of high-resolution hi-c interaction matrices. *Nature Communications* **10** (2019).

[31] Dimmick, M. C., Lee, L. J. & Frey, B. J. Hicsr: a hi-c super-resolution framework for producing highly realistic contact maps. *bioRxiv* (2020). URL `https://www.biorxiv.org/content/early/2020/07/07/2020.02.24.961714`.

[32] Luo, Y. *et al.* New developments on the encyclopedia of dna elements (encode) data portal. *Nucleic Acids Research* **48** (2019).

[33] Barrett, T. *et al.* Ncbi geo: Archive for functional genomics data sets—update. *Nucleic Acids Research* **41** (2012).

[34] Li, A. *et al.* Decoding topologically associating domains with ultra-low resolution hi-c data by graph structural entropy. *Nature Communications* **9** (2018).

[35] Dwivedi, V. P. *et al.* Benchmarking graph neural networks (2020). URL `https://arxiv.org/abs/2003.00982`.

[36] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding (2018). URL `https://arxiv.org/abs/1810.04805`.

[37] Kipf, T. N. & Welling, M. Variational graph auto-encoders (2016). URL `https://arxiv.org/abs/1611.07308`.

[38] Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks (2016). URL `https://arxiv.org/abs/1609.02907`.

[39] Shi, Y. *et al.* Masked label prediction: Unified massage passing model for semi-supervised classification. *arxiv* (2020). URL `https://arxiv.org/abs/2009.03509`.

[40] Kornblith, S., Chen, T., Lee, H. & Norouzi, M. Why do better loss functions lead to less transferable features? (2020). URL `https://arxiv.org/abs/2010.16402`.

[41] Monti, F. *et al.* Geometric deep learning on graphs and manifolds using mixture model cnns (2016). URL `https://arxiv.org/abs/1611.08402`.

[42] Oluwadare, O., Zhang, Y. & Cheng, J. A maximum likelihood algorithm for reconstructing 3d structures of human chromosomes from chromosomal contact data. *BMC Genomics* **19** (2018).

[43] Zhang, Y. Tm-align: A protein structure alignment algorithm based on the tm-score. *Nucleic Acids Research* **33**, 2302–2309 (2005).

[44] Yang, T. *et al.* Hicrep: Assessing the reproducibility of hi-c data using a stratum-adjusted correlation coefficient. *Genome Research* **27**, 1939–1949 (2017).

[45] Yardımcı, G. G. *et al.* Measuring the reproducibility and quality of hi-c data. *Genome Biology* **20** (2019).

[46] Wang, Z., Bovik, A., Sheikh, H. & Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**, 600–612 (2004).

[47] Rao & et al. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* **159**, 1665–1680 (2014).

[48] Pugacheva, E. M. *et al.* Ctcf mediates chromatin looping via n-terminal domain-dependent cohesin retention. *Proceedings of the National Academy of Sciences* **117**, 2020–2031 (2020).

[49] Schreiber, J., Durham, T., Bilmes, J. & Noble, W. S. Avocado: A multi-scale deep tensor factorization method learns a latent representation of the human epigenome. *Genome Biology* **21** (2020).