scientific reports

Check for updates

OPEN

VEHiCLE: a Variationally Encoded Hi-C Loss Enhancement algorithm for improving and generating Hi-C data

Max Highsmith✉ & Jianlin Cheng

Chromatin conformation plays an important role in a variety of genomic processes. Hi-C is one of the most popular assays for inspecting chromatin conformation. However, the utility of Hi-C contact maps is bottlenecked by resolution. Here we present VEHiCLE, a deep learning algorithm for resolution enhancement of Hi-C contact data. VEHiCLE utilises a variational autoencoder and adversarial training strategy equipped with four loss functions (adversarial loss, variational loss, chromosome topology-inspired insulation loss, and mean square error loss) to enhance contact maps, making them more viable for downstream analysis. VEHiCLE expands previous efforts at Hi-C super resolution by providing novel insight into the biologically meaningful and human interpretable feature extraction. Using a deep variational autoencoder, VEHiCLE provides a user tunable, full generative model for generating synthetic Hi-C data while also providing state-of-the-art results in enhancement of Hi-C data across multiple metrics.

Hi-C data, an extension of chromosome conformation capture assay (3C) is a biological assay which can be used to inspect the three-dimensional (3D) architecture of a genome[1]. Hi-C data can be used for downstream analysis of structural features of chromosomes such as AB compartment, Topological Associated Domains (TADs), loops, and 3D chromosome and genome models. Changes in chromosomal conformation have been empirically demonstrated to impact a variety of genomic processes including gene methylation and gene expression[2].

When analyzing Hi-C data, reads are usually converted into contact matrices, where each cell entry corresponds to the quantity of contacts between the two regions indexed by row and column[3,4]. The size of an individual region in this contact matrix is referred to as the resolution or bin size[4]. The smaller the bin size, the higher the resolution. The resolution of a contact matrix is usually selected based on the quantity of read pairs in an individual Hi-C experiment, with a higher quantity of read pairs permitting a higher resolution. Certain genomic features, such as TADs, can only be meaningfully identified using high resolution contact matrices, however if a matrix resolution is selected with insufficient read coverage the matrices can be overly sparse. One method to address this issue is to run additional Hi-C experiments, however because of experimental costs this is not always a feasible solution.

To solve this problem previous groups have utilized methods from the field of Image super-resolution to improve Hi-C contact matrix resolution. The first of these networks was HiCPlus[5], a simple neural network optimized using mean squared error (mse). HiCPlus was then improved upon by HiCNN[6] by adjusting network architecture. Next hicGAN[7] was proposed, introducing the use of Generative Adversarial Networks (GAN), which generated high resolution contact maps conditioned on low resolution input. The network DeepHiC[8] maintained the GAN loss function while extending it to also include a perceptual loss function derived from VGG-16 trained on image data. The model HiCSR[9] continued the advancement by introducing the use of a deep autoencoder as a feature extraction mechanism.

Our network, the Variationally Encoded Hi-C Loss Enhancer (VEHiCLE), extends the approach of conditional generative adversarial networks by using an integrated training approach inspired by literature in the domains of deep learning and genomics. First, VEHiCLE incorporates a variational autoencoder which extracts biologically meaningful features from Hi-C data. Second, VEHiCLE's decoder network is engineered to provide an easy to use generative model for Hi-C data generation which smoothly maps user tunable, low dimensional vectors to Hi-C contact maps independent of any low sampled input. Third, VEHiCLE incorporates a biologically

Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA. ✉email: mrh8x5@mail.missouri.edu

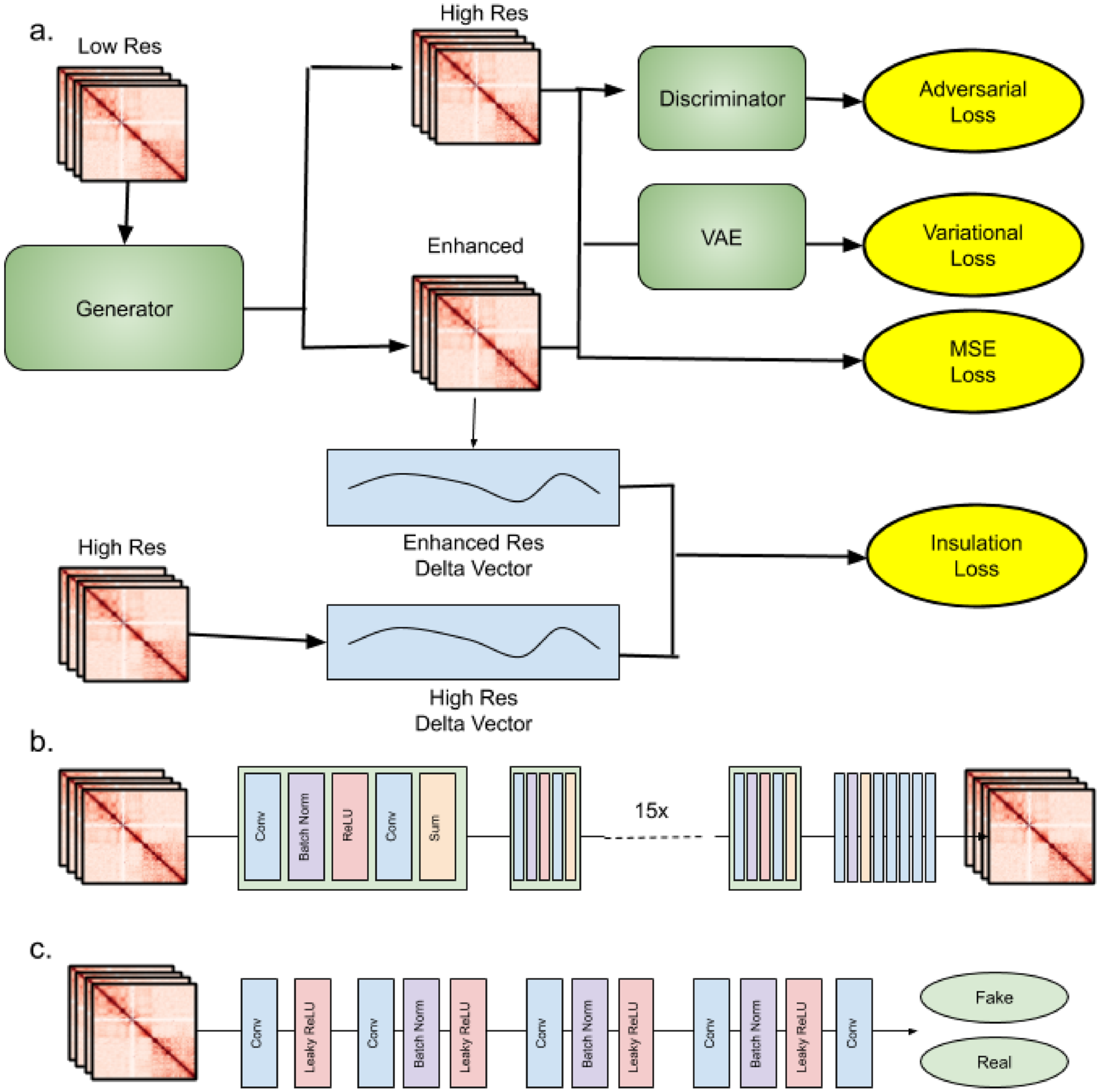

Figure 1. VEHiCLE architecture: (**a**) overview of training strategy, (**b**) generator architecture, (**c**) discriminator architecture.

explicit loss function based on Topologically Associated Domain identification to ensure accurate downstream genomic analysis.

VEHiCLE obtains state of the art results in the task of Hi-C super-resolution across a variety of metrics pulled from the domains of Image analysis and Hi-C quality/reproducibility. VEHiCLE enhanced data show successful retrieval of important downstream structures such as TAD identification and 3DModel generation while also providing novel human interpretability of its enhancement process.

Approach

Description of VEHiCLE network training.

Vehicle is trained as an adversarial network conditioned on low resolution input. The network is trained using a composite loss function made up of 4 sub loss functions: Adversarial loss, Variational loss, mean square error (MSE) loss, and Insulation loss. An overview of the training mechanism is displayed in Fig. 1a. The intellectual motivation for each of these loss functions is outlined below.

Adversarial loss function. Generative adversarial networks (GANs) are a popular deep learning based framework for generative modeling which has gained traction in a wide variety of tasks including image superresolution. GANs were first introduced to the field of Hi-C super resolution through hicGAN, and later improved upon in DeepHiC and HiCSR. A GAN uses two key networks: a generator G (Fig. 1b) and a discriminator D (Fig. 1c). The generator takes samples from an input distribution and generates enhanced matrices. The Discriminator is trained on a collection of inputs including real high resolution Hi-C samples as well as enhanced resolution Hi-C samples and attempts to determine whether individual samples are real or enhanced. The two networks are trained in a game where the generator is rewarded for successfully tricking the discriminator and the discriminator tries to minimize classification mistakes.

$$\text{min}_G \text{max}_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[log(1 - D(D(z))))] \tag{1}$$

The generator loss function is defined as:

$$L_{adb} = \sum_{n=1}^{N} -log(D(G(X_{low})) \tag{2}$$

Variational loss. Autoencoders are deep learning systems which map inputs from sample space to a condensed latent space via an encoder and then reconstruct images in sample space from the latent space using a decoder. The use of autoencoders for the task of Hi-C data super resolution was originally proposed in our preprint[10] for the task of denoising Hi-C data. They were then suggested by Dimmick et al.[9] as tools for training super resolution networks by using the features extracted by passing Hi-C data through a trained autoencoder as a loss function. In this manuscript we expand upon this strategy, but replace their network with a different flavor of network called the variational autoencoder[11].

Similar to vanilla autoencoders, variational autoencoders (VAE) aim to condense data into lower dimensional space, however they have the advantage of providing smooth feature representation which can permit the construction of powerful generative models. To obtain these advantages VAE rely upon a statistical method called variational inference[11]. This method frames the tasks of encoding and decoding as an ancestral sampling problem with two steps: First, a latent variable, z, is sampled from a prior distribution $P_\theta(z)$. Second, the observed variables, x, is drawn from a likelihood distribution $P_\theta(x|z)$.

To encode the observed variable, x, requires the computation of the posterior distribution $P_\theta(z|x)$. However because this is computationally intractable, instead one approximates the posterior by choosing a parametric family of recognition models $q_\phi(z|x)$ and selects parameters that minimize the divergence between the recognition model and the true underlying distribution via a probabilistic dissimilarity metric called KL-divergence,

$$D_{kl}(q_\theta(z|x)||p_\phi(z)) = \sum_{\mathbb{Z}} q(z) log\left(\frac{q(z)}{p(z)}\right) \tag{3}$$

By performing some algebra outlined in Kingma and welling[11] variational autoencoders are trained using the following loss function

$$L(\theta,\phi,x) = -D_{kl}(q_\phi(z|x)||p_\theta(z)) + \int_z q_\phi(z|x) log(p_\theta(x|z)) \tag{4}$$

The integral term on the far right of the loss function ensures that the reconstruction outputs of our networks are highly similar to their original inputs, while the KL divergence term causes the latent space distribution of values to closely resemble a vector of gaussian random variables. This imposition of gaussian similarity on the latent space results in advantages in quality of extracted features and the procurement of a generative model.

To create the variational loss function, we first train our variational autoencoder using high resolution contact matrices as both inputs and labels. In each experiment our VAE network is trained using the same chromosomes as the overall VEHiCLE network. The variational autoencoder maps vectors from data space into condensed latent space, which we interpret as a lower dimensional feature vector (Fig. 2a,b). Because the variational autoencoder training strategy imposes a Gaussian distribution of the latent space variables and because our decoder maps latent vectors back into data space in a relatively smooth manner we expect highly similar Hi-C contact matrices to contain similar latent space profiles.

We compute variational loss by passing both the enhanced Hi-C contact matrix and target high resolution Hi-C contact matrices through the backpropagatable encoder component of our variational autoencoder network, extracting latent dimensional representations. We then compute the mean differences between their latent feature vectors,

$$L_{vae} = \frac{1}{n}\sum_{i=1}^{n} \left|f_{encode}(X_{enhanced}) - f_{encode}(X_{target})\right| \tag{5}$$

where f_{encode} is the encoding function, defined by our trained encoder network.

Insulation score loss. Most of the previously proposed loss functions for developing Hi-C enhancement networks draw upon loss functions prolific in the fields of computer vision[5–9]. While there are certainly advantages to these strategies, they derive from assumed similarities between the tasks of image super-resolution and Hi-C super-resolution. However, the tasks are not synonymous. Hi-C contact matrices contain important information used for downstream feature analysis such as loop calling, TAD identification and 3D model construction. Consequently, images which are highly visually similar which are blurry, shift positions of structural features, or

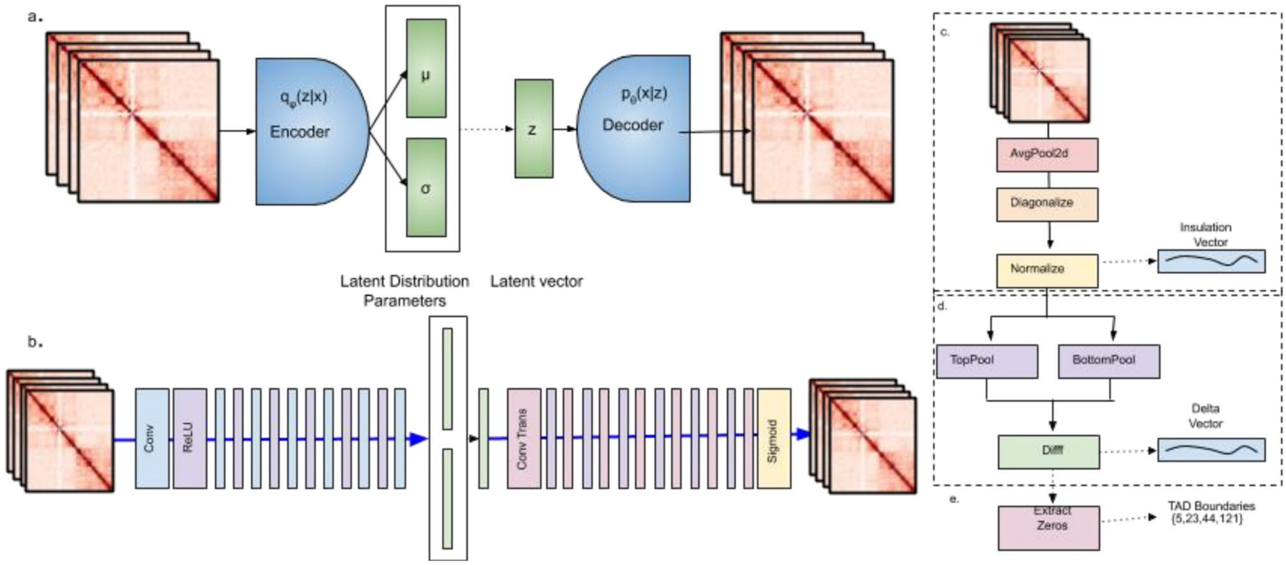

Figure 2. Variational autoencoder. (**a**) Overview of variational autoencoder approach. (**b**) VEHiCLE architecture. Tad loss evaluated using a feedforward implementation of Insulation loss computing, (**c**) insulation vector, (**d**) delta vector and (**e**) identification of TAD Boundaries.

contain noise might result in significant differences in downstream analysis. With this fact in consideration, we used domain knowledge of computational genomics to devise an insulation loss function, which directly trains networks to correctly identify downstream features, specifically TAD placement.

One well-established strategy for the identification of TADs is the use of insulation scores[12]. Insulation scores of a matrix are calculated by sliding a 20 bin (200 kb × 200 kb) window down the diagonal of a matrix and summing the signal across each bin, resulting in an insulation vector (Fig. 2c). This insulation vector is normalized by taking the log2 ratio of each bin's insulation score and the mean of all insulation scores on the chromosome. From the insulation vector a delta vector is computed by observing the change in signal strength 100 kb downstream and upstream of each bin on the insulation vector (Fig. 2d). This delta vector is treated as a pseudo-derivative, and identifies insulation valleys in the regions where the delta vector crosses the x-axis from negative values to positive values, indicating a relative minimum in insulation. TAD Boundaries are assigned to each insulation valley whose difference in strength between the nearest left local max and right local min was > 0.1 (Fig. 2e).

The insulation TAD calling procedure can be encoded into a single, back propagatable network up until extraction of the delta vector (Fig. 2c,d). We define insulation loss,

$$L_{ins} = \frac{1}{n}\sum_{i=1}^{n}\left|D_{vec}(X_{enhanced}) - D_{vec}(X_{target})\right| \tag{6}$$

where D_{vec} is a backpropagatable network which maps a contact matrix to a delta insulation vector.

Bin-wise mean squared error loss. Bin-wise mean square error loss is a thoroughly tested loss function used in previous Hi-C enhancement literature[6,8,9]. It contributes to maintaining visual similarity between enhanced and target Hi-C contact matrices.

$$L_{mse} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|X_{enhanced} - X_{target}\right| \tag{7}$$

Composite training function. To capitalize on the advantages of all four loss functions we incorporate them into our comprehensive training process. First the variational network is trained on the train and validation datasets. Then the trained encoder is used for L_{vae} along with the three other training losses to train the generator network, yielding our overall loss function

$$L_{tot} = \lambda_{adv}L_{adv} + \lambda_{mse}L_{mse} + \lambda_{vae}L_{vae} + \lambda_{ins}L_{ins} \tag{8}$$

where λ_x are hyperparameters used to determine loss contribution. We use $\lambda_{adv} = 0.0025$, $\lambda_{mse} = 1$, $\lambda_{vae} = 0.01$, $\lambda_{ins} = 1$.

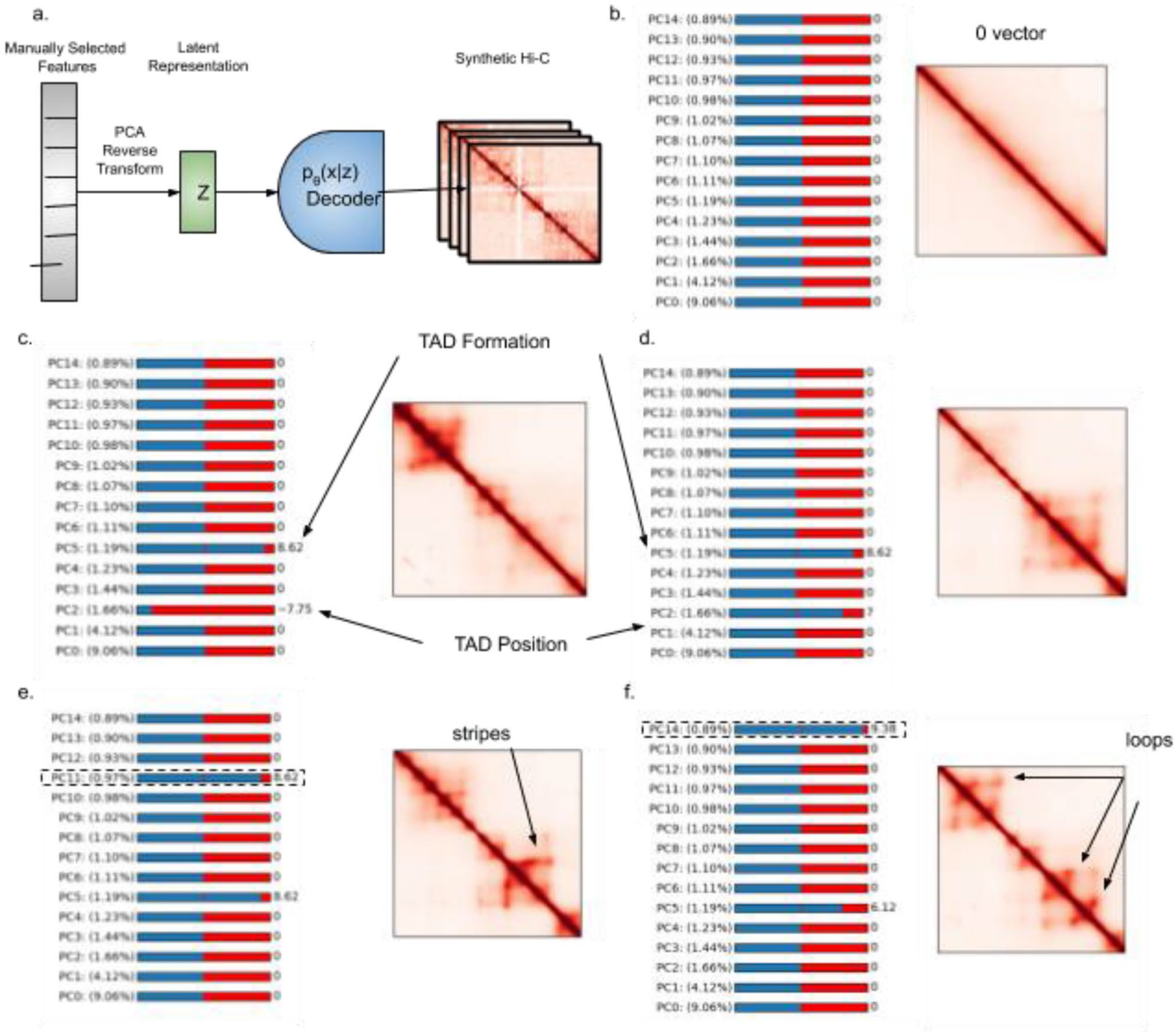

Figure 3. (**a**) Diagram of synthetic Hi-C generation tool, a user tunable zero-centered feature vector is transformed via PCA reverse transform to latent space and then passed through our tuned decoder network. (**b**) The 0 vector corresponds to a purely linear contact map. (**c**) Increasing value of PC5 results in generation of TADs. (**d**) Adjusting value of PC2 shifts position of TADs. (**e**) Adjusting PC11 creates stripes within TADS. (**f**) Adjusting PC14 develops loops within TADs.

Results

Latent space representations permit generation of synthetic Hi-C data.

The KL divergence term in the loss function of our variational autoencoder imposes constraints on the latent dimension, pushing our estimate for the prior $q(z|x)$ towards a vector distribution of Gaussian random variables. Because all latent vector variables fall within Gaussians centered around 0, most vectors near the center of these Gaussians can be successfully decoded into Hi-C space, resulting in a generative model for Hi-C data. We first perform principal component analysis (PCA) on our training set's learned, latent dimensional features. We then create a function mapping PCA values to the latent dimensional space. We then use our trained decoder network to transform the values in latent dimensional space into Hi-C space (Fig. 3a). The result is a function mapping a profile of PCA values to a 2.57 Mb × 2.57 Mb block of Hi-C data. We hook this function into an interactive matplotlib widget, permitting manual visualization of changes to generated Hi-C data as input variables are adjusted. In our widget we set a NUM_SLIDERS = 15 parameter to permit the manual tuning of PCA vector components. The widget passes a vector to our mapping function with user selected values in all manually adjusted components and dataset averages for all PCA's that are not manually selected or are above the NUM_SLIDERS component index threshold. The selection of 15 is arbitrary and can be manually increased by users interested in viewing the impact of adjusting higher PC values on the generated Matrix structure.

The zero vector results in a vanilla Hi-C map with interaction frequency between two regions following the inverse of genomic distance (Fig. 3b). The biological interpretation of some adjustable features remains elusive,

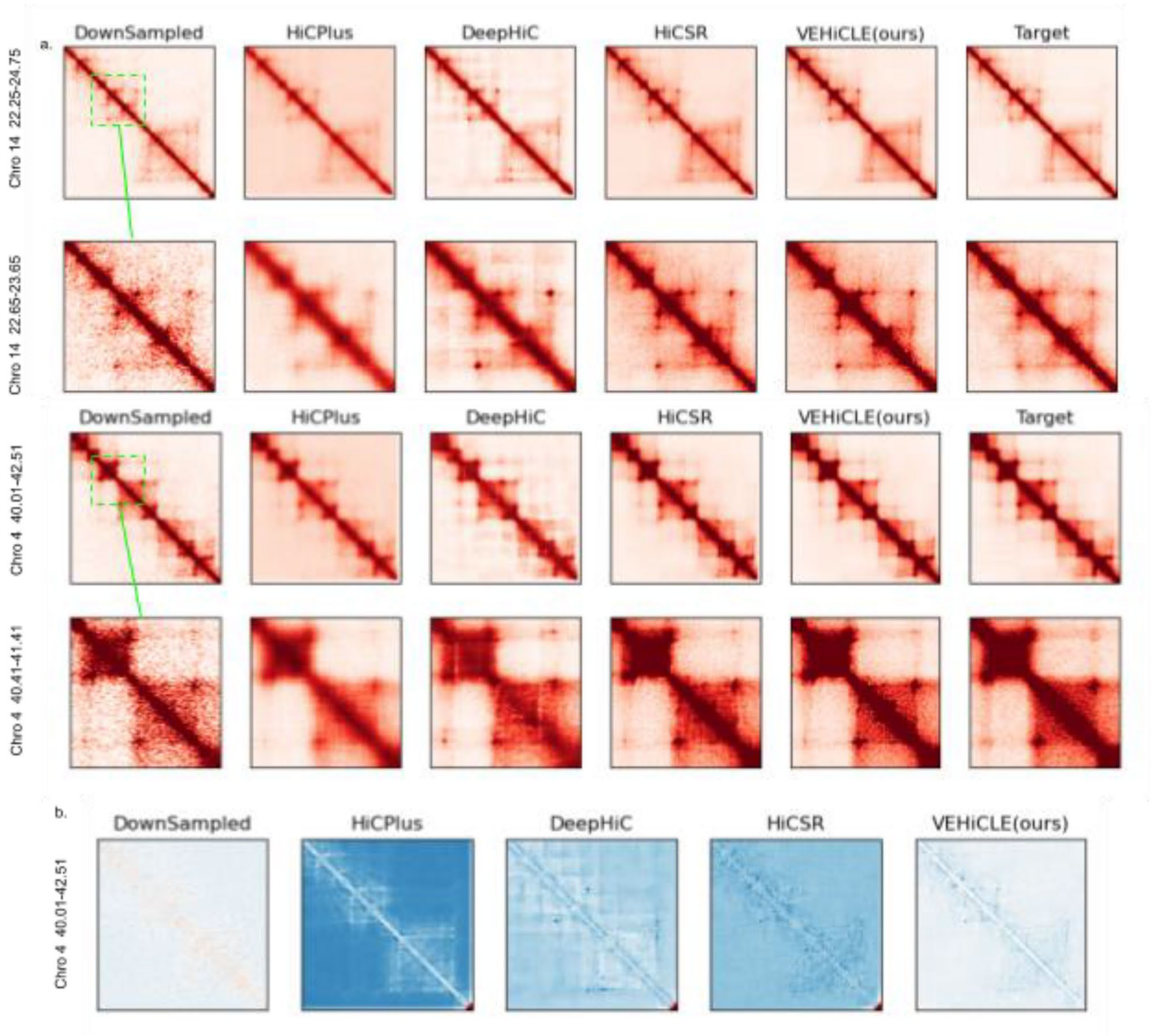

Figure 4. Visual Comparison of deep learning based methods for contact matrix enhancement. (**a**) Visual comparison of enhancement matrices. (**b**) Absolute difference matrices between target high resolution data and enhancement. All displayed matrices are derived from the GM12878 cell line. Architectures of previous models utilize original window size.

with changes to vector component values resulting merely in changes of diagonal signal strength or sporadic repositioning of contact regions. However, we observe that many of the tunable feature vector components correspond directly with biologically meaningful features in Hi-C space such as: formation of TADs, increasing TAD size (Fig. 3c), increasing TAD frequency, shifting TAD position (Fig. 3c,d), formation of genomic stripes (Fig. 3e)[13] and formation of chromatin loops[14] (Fig. 3f).

Low resolution Hi-C contact matrices enhanced by VEHiCLE appear visually competitive with other enhancement algorithms. We generate visual heatmaps of Hi-C contact maps of the GM12878 dataset using VEHiCLE as well as three other previously developed algorithms: HiCSR, DeepHiC and HiCPlus. We observe high visual similarity between reconstructions by VEHiCLE and other enhancement algorithms (Fig. 4a). We also subtracted high resolution contact maps from reconstructions by each tool to observe a visual difference matrix (Fig. 4b). Visually VEHiCLE appears competitive with existing algorithms.

Notes on evaluation metrics. One of the major differences between the VEHiCLE algorithm and previous Hi-C enhancement tools is that our architecture is trained to enhance 2.69 Mb × 2.69 Mb regions along diagonals of contact maps rather than splitting contact maps into 0.4 Mb × 0.4 Mb pieces, enhancing in a piecemeal fashion, and then reassembling (see "Methods"). This contribution permits the inclusion of more comprehensive information like TAD structure into training samples. However, it is possible to expand older architectures

	Downsampled	HiCPlus 40	DeepHiC 40	HiCSR 40	HiCPlus 269	DeepHiC 269	HiCSR 269	VEHiCLE
Chro 4								
PCC	0.7592	0.9103	0.9212	0.9285	0.9467	**0.9524**	0.9463	**0.9524**
SPC	0.6259	0.805	0.7715	0.8292	0.8646	**0.8837**	0.8719	**0.8739**
SSIM	0.2336	0.3284	0.3784	0.4346	0.3785	**0.4305**	**0.4526**	0.3978
MSE	0.0468	0.0163	0.0162	0.0114	0.0091	0.0083	**0.0097**	**0.0098**
SNR	306.65	514.24	516.847	619.41	700.549	733.5042	**673.73**	**670.9001**
Chro 14								
PCC	0.8682	0.9374	0.9481	0.9583	0.9716	0.975	**0.9753**	**0.9764**
SPC	0.6692	0.8159	0.7188	0.85	0.88	**0.9031**	0.888	**0.8892**
SSIM	0.3524	0.4022	0.5481	0.5877	0.644	**0.6588**	**0.67**	0.6439
MSE	0.0145	0.0117	0.0052	0.0041	0.0027	**0.0024**	**0.024**	**0.0024**
SNR	341.5712	380.041	554.9034	627.752	786	**830.2759**	847.28	**834.5026**
Chro 16								
PCC	0.8798	0.9327	0.9479	0.9602	0.9694	**0.9771**	0.8771	**0.9769**
SPC	0.6684	0.8097	0.6949	0.8496	0.8887	**0.9027**	0.8884	**0.8896**
SSIM	0.3901	0.3935	0.5618	0.5924	0.6913	**0.7058**	**0.7095**	0.6948
MSE	0.0118	0.0124	0.0047	0.0036	0.0027	**0.0021**	**0.0021**	**0.0022**
SNR	332.8447	318.105	517.1062	597.73	703.324	**808.4**	**810.28**	797.5162
Chro 20								
PCC	0.9075	0.9303	0.9507	0.9656	0.9692	0.9825	**0.983**	**0.983**
SPC	0.6866	0.827	0.6857	0.8631	0.9094	**0.9184**	0.9033	**0.905**
SSIM	0.4373	0.4082	0.6022	0.6432	0.7522	**0.7559**	**0.7619**	**0.7559**
MSE	0.0076	0.0124	0.0038	0.0027	0.0023	**0.0014**	**0.0013**	**0.0014**
SNR	364.83	282.43	510.35	608.04	662.656	850.7672	**886.20**	**868.3073**

Table 1. Comparison of multiple standard vision metrics across different super-resolution algorithms. Networks are trained using the training set chromosomes of the GM12878 cell line and evaluated on the test chromosome set of the GM12878 cell line. Top 2 scores for each metric are bolded.

to full 2.69 Mb × 2.69 Mb sizes rather than the condensed 0.4 Mb × 0.4 Mb window that appears in previous papers. In some cases this expansion of window size degrades older architecture performance, while in others it leads to enhancement. Thus, when comparing VEHiCLE to previous tools we include both original architectures without adjusting window size as well as alternative architectures trained using expanded window sizes.

Low resolution Hi-C contact matrices enhanced by VEHiCLE achieve strong similarity to high resolution contact matrices using multiple metrics. Using models trained and tested on the GM12878 cell line dataset We evaluated the effectiveness of VEHiCLE in predicting high resolution contacts using 5 common metrics: Pearson Correlation Coefficient (PCC), Spearman Correlation Coefficient(SPC), Mean Squared Error (MSE), Signal-to-noise ratio (SNR) and Structure Similarity Index (SSI) (see "Methods"). We compared VEHiCLE reconstructions to the lower resolution data as well as other super resolution methods (HiCPlus, DeepHic and HiCSR.) VEHiCLE enhanced contact matrices consistently showed improvement relative to low resolution data along all 5 metrics (Table 1). VEHiCLE frequently out-performed other Hi-C super resolution methods beating all older models with 0.4 Mb window size along every test chromosome in every vision metric (Table 1). VEHiCLE out performs both the original and expanded window HiCPlus model in every vision metric across every chromosome (Table 1). VEHiCLE remained competitive with 2.69 Mb window sized DeepHiC and HiCSR models scoring highest in PCC in 3 of the 4 test chromosomes and scoring in the top 2 for 80% of the metric-chromosome combinations, a higher consistency of top-2 performance than any of the previous models (Table 1).

Downsampled Hi-C contact matrices enhanced by VEHiCLE display significant improvement using Hi-C specific metrics. Using models trained on the GM12878 cell line dataset we next evaluated VEHiCLE reconstructions using 3 Hi-C specific metrics: GenomeDISCO, HiCRep and QuASAR-Rep (see "Methods"). VEHiCLE enhanced metrics remain competitive with other methods (Table 2). Furthermore, even in instances where VEHiCLE is outperformed by another algorithm, we consistently observe increased performance relative to original low resolution matrices. These results indicate biological consistency with VEHiCLE enhanced matrices.

VEHiCLE enhanced contact matrices effectively retrieve downstream features such as TADS. We identified TADs using the prolific insulation score method[12]. This method assigns an insulation score vector by sliding a window across the diagonal of the contact matrix, constructing an insulation difference vector, and using the zeros of the insulation difference vector to discover TAD boundaries. We used models

	Downsampled	HiCPlus 40	DeepHiC 40	HiCSR 40	HiCPlus 269*	DeepHiC 269	VEHiCLE
Chr 4							
GenomeDISCO	0.941	0.972	0.945	**0.98**	0.972	**0.98**	0.972
HiCRep	0.967	0.974	0.972	**0.989**	0.972	**0.99**	0.972
QuASAR-Rep	0.924	0.995	0.993	**0.995**	**0.995**	0.589	**0.995**
Chr 14							
GenomeDISCO	0.942	0.933	0.907	**0.979**	0.975	**0.977**	0.972
HiCRep	0.982	0.969	0.97	**0.991**	0.987	**0.992**	**0.991**
QuASAR-Rep	0.944	0.995	0.993	**0.996**	**0.996**	**0.996**	**0.996**
Chr 16							
GenomeDISCO	0.927	0.904	0.88	0.972	**0.967**	**0.972**	0.969
HiCRep	0.974	0.948	0.96	0.987	0.978	**0.988**	**0.987**
QuASAR-Rep	0.941	0.992	**0.99**	0.994	0.994	**0.995**	**0.995**
Chr 20							
GenomeDISCO	0.934	0.895	0.864	**0.974**	0.968	**0.973**	0.948
HiCRep	0.981	0.949	0.959	**0.988**	0.984	**0.989**	0.979
QuASAR-Rep	0.955	0.994	**0.99**	**0.996**	**0.996**	**0.996**	**0.996**

Table 2. Comparison of Hi-C superresolution algorithms using Hi-C reproducibility metrics. Networks are trained using the training set chromosomes of the GM12878 cell line and evaluated on the test chromosome set of the GM12878 cell line. Top 2 scores for each metric are bolded. *Our version of the HiCSR model with an expanded window size of 269 repeatedly failed to converge using these tools, thus we include only the authors original model for comparison.

trained on the GM12878 cell line and evaluated insulation on test chromosomes for the HMEC, K562 and IMR90 cell lines as well as the GM12878 cell line. We expand the test set to evaluate the effectiveness of our network at predicting downstream biological features like TADs when the model is trained on different cell lines which may have different TAD profiles.

We compare the insulation difference vector of each matrix-enhancement algorithm to the insulation difference vector of our high resolution contact matrix using the L2 norm dissimilarity metric. In many cases VEHiCLE enhanced insulation difference vectors have higher similarity to target matrices relative to other matrix enhancing algorithms (Table 3). Furthermore, even in instances where VEHiCLE is outperformed by another algorithm we consistently observe higher similarity between the target high resolution matrices and VEHiCLE enhanced matrices relative to low resolution matrices (Table 3).

3D chromatin model construction.

We tested the effectiveness of reconstructed data in building 3D structure models using the structural modeling tool 3DMax. We extracted constraints from the low resolution, high resolution and VEHiCLE-enhanced 2.57 Mb × 2.57 Mb regions of our test dataset chromosomes of the GM12878 dataset. From each constraint grouping we generated 3 models. We observed significantly higher visual similarity between VEHiCLE-enhanced and high-resolution matrices relative to low-resolution matrices (Fig. 5a). We then used the TM-score metric to quantify structural similarity of models[15]. We observed higher TM-scores between high resolution and VEHiCLE-enhanced matrices than between high resolution and low resolution models (Fig. 5b). We also observed higher TM-score similarities between models generated by the same VEHiCLE-Enhanced matrices relative to models generated by the same low resolution matrices, indicating VEHiCLE enhanced models are more consistent (Fig. 5c).

Discussion

One of the most common challenges in Deep Learning projects is the opaque nature of a neural network's inner functioning. Consequently, our ability to extract latent features and map them to biologically relevant structures provides a significant advance in increasing interpretability of Hi-C matrices. Our GUI tool can be used to generate Hi-C data through user tunable parameters with biologically relevant downstream structures such as TAD strength, TAD positioning, stripes and loops. Further inspection of these features has potential to enhance analysis of key characteristics of chromatin organization.

Our introduction of the Insulation loss sets a new precedent of utilizing biological knowledge in the training of Hi-C networks. This may open the door for future improvement of Hi-C data enhancement by utilizing other forms of domain knowledge to increase usability of deep learning enhanced matrices. Future loss functions could incorporate algorithms for identification of other important downstream features such as loops or stripes.

In addition to the increased interpretability and inclusion of domain knowledge, VEHiCLE obtains resolution enhancement results competitive with the state-of-the-art, often beating top algorithms on a variety of metrics, all while preserving the ability to convey meaningful structures such as TAD's and 3D structure in downstream analysis.

VEHiCLE's capacity to increase accuracy of insulation scores shows promise of utility for experimental biologists interested in chromosome architecture at specific genomic locations. By enhancing experimentally obtained

Norm of insulation score difference vectors								
	Downsampled	HiCPlus 40	DeepHiC 40	HiCSR 40	HiCPlus 269	DeepHiC 269	HiCSR 269	VEHiCLE
Chr 4								
GM18278	7.966	6.64	7.52	4.389	7.8217	**4.3782**	**4.7323**	4.763
K562	10.942	9.1133	9.957	7.605	10.688	7.976	**7.11**	**7.305**
IMR90	9.8344	8.5681	9.2244	5.9457	9.78	**5.736**	6.091	**5.5729**
HMEC	16.143	13.132	15.267	10.17	15.8212	11.6367	**10.1420**	**11.2512**
Chr 14								
GM18278	2.68	2.619	3.774	2.898	2.97	**2.305**	2.473	**2.3414**
K562	6.225	5.927	6.329	5.548	6.28	5.1104	**4.8282**	**4.868**
IMR90	4.3609	4.6005	5.1827	3.871	4.838	3.309	**3.284**	**3.244**
HMEC	9.214	8.34	9.549	7.448	9.2113	**6.9471**	**6.5814**	7.0423
Chr 16								
GM18278	4.162	3.467	3.769	3.099	4.3619	2.6623	**2.3862**	**2.4376**
K562	6.653	5.903	6.485	5.14	6.817	4.6	**4.465**	**4.572**
IMR90	5.806	5.0148	5.459	4.169	6.134	3.556	**3.117**	**3.376**
HMEC	8.957	8.353	8.799	7.527	9.1517	**6.4103**	**6.068**	6.4423
Chr 20								
GM18278	2.077	2.419	2.587	2.624	2.5274	**1.8383**	**1.807**	1.922
K562	5.316	4.835	5.021	4.307	5.4488	4.267	**3.811**	**3.908**
IMR90	2.888	3.522	3.444	3.083	3.5723	2.3699	**2.2602**	**2.3169**
HMEC	6.383	6.662	6.579	5.805	6.562	**4.7832**	**4.701**	4.8159

Table 3. L2 norm of TAD Insulation difference vectors against target insulation vectors. Networks are trained using the training set chromosomes of the GM12878 cell line and evaluated on the test chromosome sets of the K562, IMR90, HMEC and GM12878 cell line. The top 2 scores for each metric are bolded.

Hi-C data a biologist could observe the frequency with which a list of genes or cis regulatory elements are found near TAD boundaries. Such analysis could provide further insight into the role of structural organization in a genomic process. Additionally, VEHiCLE enhanced matrices could be used to generate more accurate 3D models when building visualizations of genomic structure. These visualizations may provide insight into the underlying machinery of a genomic process of interest.

Methods

Dataset assembly.

Like many of the previous Hi-C super resolution networks we train VEHiCLE on high and low resolution Hi-C data for the GM12878 cell line[16]. While previous work often split chromosomes into training, validation and testing sets in a sequential manner[8,9] we were concerned that differences in the 3D conformation of large vs small chromosomes[17] may contain implicit bias in contact map features that could confound training. Consequently we assembled training, validation and test sets in a non-sequential manner using chromosomes 1, 3, 5, 6, 7, 9, 11, 12, 13, 15, 17, 18, 19, 21 as our training set, chromosome 2, 8, 10, 22 as our validation set and chromosomes 4,14,16,20 as our test set.

Previous work on Hi-C super resolution consistently used network input window sizes of 0.4 Mb × 0.4 Mb at 10 kb resolution, requiring networks to split chromosome contact maps into 40 × 40 bin matrices[5–9]. While this strategy has seen relative success, a major disadvantage is that certain important features of Hi-C such as TADs can span ranges larger than 0.4 Mb, meaning that it is impossible for previous networks to explicitly encode important information about TAD organization. Furthermore, this informational bottleneck of constraining window sizes to 40 × 40 bins is not incumbent upon the employed super-resolution networks as work in the field of computer vision has demonstrated the effectiveness of GAN and VAE networks on significantly larger images. With these considerations in mind we instead built our network to accept 2.69 Mb × 2.69 Mb images, a range which is large enough to fully encompass the average TAD of length 1MB[18]. Observing 2.69 Mb × 2.69 Mb regions of Hi-C contact maps at range 10 kb results in submatrix images of 269 × 269 bin size. Because of the expanded window size we trained our network exclusively on diagonally centered submatrices, split by sliding a 269 × 269 window down the diagonal of each chromosome's Hi-C contact map. We move the window with a stride of 50 bins at a time, ensuring sufficient overlap between samples for our dataset to include all contacts between regions within 2 Mb of each other. This results in a total of 3309 training, 1051 validation, and 798 testing matrices.

Because the convolutional arithmetic of our GAN architecture results in a decrease in output matrices by 12 bins, our output matrices are of dimension 257 × 257. Our variational loss is based on reconstruction of matrices output by our GAN, thus when training our variational autoencoder we use the inner 257 × 257 bins of each 269 × 269 sample in our dataset.

All Models were trained using the GM12878 cell line. When evaluating vision metrics, Hi-C qc metrics and 3D model comparison we use the test chromosomes from the GM12878 cell line. For our insulation score analysis we extend our test set to include the K562, IMR90 and HMEC cell lines so as to verify the effectiveness of our network at retrieving information when trained on a different cell line. Both low resolution and high-resolution

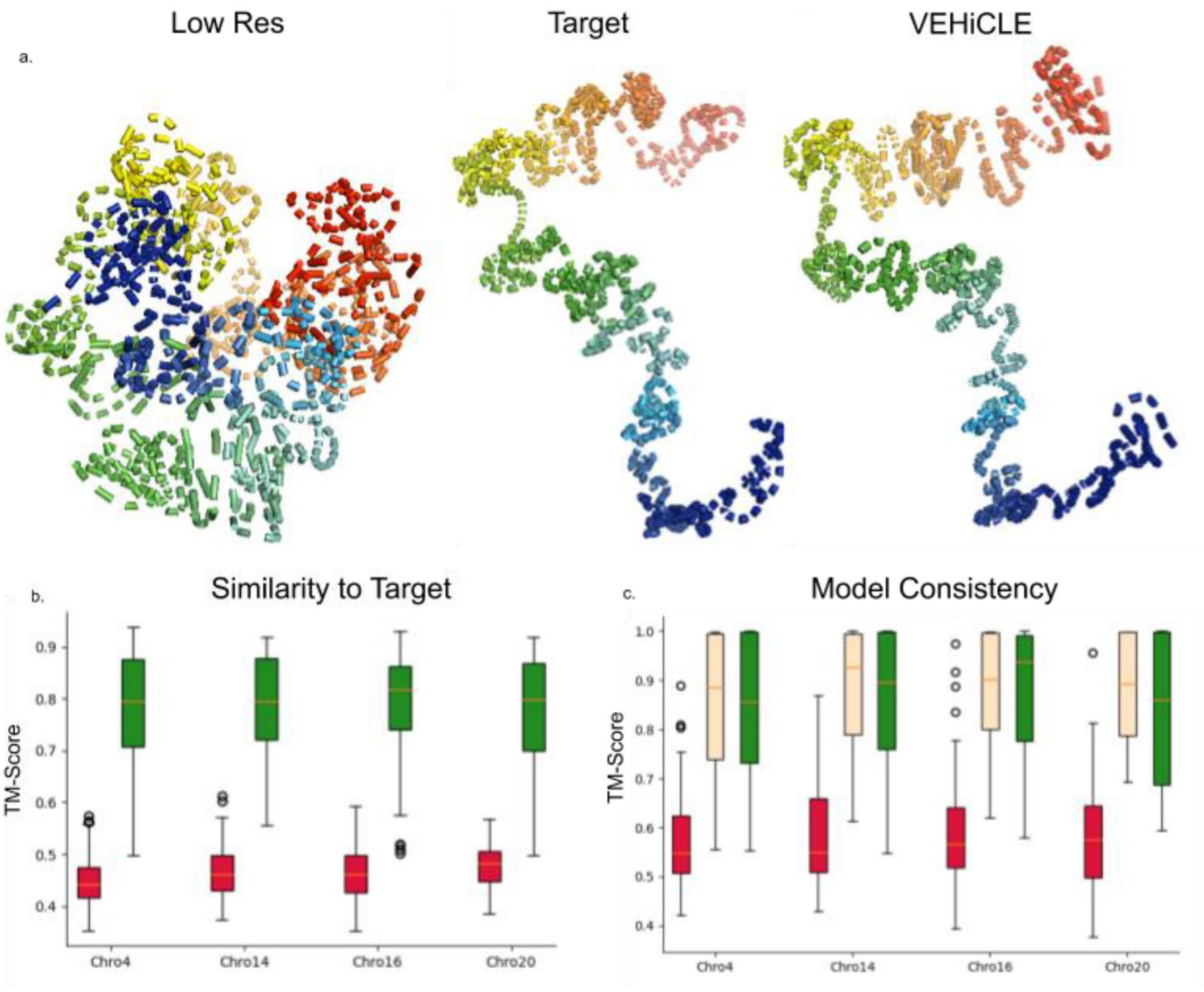

Figure 5. (**a**) 3D reconstruction of Chro 20 0.6–3.1 MB. (**b**) TM -score comparison of High Resolution structures to (red) Low resolution structures and (green) VEHiCLE enhanced structures. VEHiCLE enhanced scores are significantly higher (Wilcoxon rank sum p value < 1e–20). (**c**) Average TM-Score comparison of ingroup structures generated by same contact matrix (red) low res, (yellow) high res, (green) VEHiCLE enhanced. VEHiCLE enhanced scores are significantly better than low-resolution scores (wilcoxon rank sum p value < 1e–20) Structures are all generated from GM12878 cell line using the test chromosome set: 4, 14,16,20.

contact maps are normalized using the Knight–Ruiz algorithm, a standard normalization method in the Hi-C literature [19].

Variational autoencoder architecture. The VAE component of VEHiCLE utilizes two neural networks for the encoding and decoding components, where the encoder is trained for the parameters of q_0 and the decoder is trained to optimize the parameters of p_0. The VEHiCLE encoder network contains 7 convolutional layers with kernel counts: 32, 64, 128, 256, 256, 512, 512. Each convolutional layer is separated by leaky ReLU and batch normalization. The decoder network has 7 layers of convolution transpose with the kernel counts 512, 512, 256, 256, 128, 64, 32, also separated by leaky ReLU and batch norm functions. The decoder network is appended by a Sigmoid activation function placing outputs in the range of [0,1].

Generative adversarial network architecture. We use the discriminator and generator architecture defined in HiCSR, with the exception of our generator's output function, which is changed from tanh, to a sigmoid so that outputs are mapped to [0,1]. The generator architecture contains 15 residual blocks separated by skip connections, each containing 64 convolutional filters. The fully convolutional discriminator is a fully convolutional network with ReLU activation. Both the generator and discriminator are trained with batch normalization.

Other networks. We used the pytorch versions of HiCPlus, DeepHiC and HiCSR provided at https://github.com/wangjuan001/hicplus, https://github.com/omegahh/DeepHiC and https://github.com/PSI-Lab/

HiCSR. We first tested networks using their literature provided weights, however we obtained very poor performance because these networks were trained on alternative training sets with key characteristic differences from ours. First, their training sets had bin value ranges of [− 1,1], however our training datas range was [0,1] because negative values confound the probabilistically motivated VAE component. Second the input size of contact maps for previous networks was 40 × 40, while our network aims to incorporate surrounding genomic information and utilizes a larger window input size of 269 × 269. To provide more accurate comparison we trained networks on our own GM12878 Dataset. Because our networks accept a large scale input matrix 269 × 269, but other networks were built to accept 40 × 40 pieces, we trained other networks by splitting each 269 × 269 into 36 non-overlapping pieces. Evaluation of Hi-C metrics was performed by feeding split pieces through networks as necessary, then reassembling pieces and comparing full chromosome contact maps.

Standard evaluation metrics. We utilize 5 reproducibility metrics pulled from image-super resolution literature: Pearson Correlation Coefficient (PCC), Spearman Correlation Coefficient (SPC), Mean Squared Error (MSE), Signal-to-noise ratio (SNR) and Structure Similarity Index (SSI).

Mean squared error.

$$L_{mse}(x,y) = \sum_{i=1} (x_i - y_i)^2$$

Pearson correlation coefficient.

$$L_{pcc}(x,y) = \frac{\sum_{i=1}(x_i - x)(y_i - y)}{\sqrt{\sum_{i=1}(x_i - x)^2}\sqrt{\sum_{i=1}(y_i - y)^2}}$$

Spearman correlation coefficient. Spearman Correlation is similar to Pearson correlation differing in that it utilizes rank variables so as to evaluate monotonic relationship between the matrices without imposing a linearity condition that may not exist in nature.

$$L_{spc}(x,y) = \frac{\sum_{i=1}(rx_i - rx)(ry_i - ry)}{\sqrt{\sum_{i=1}(rx_i - rx)^2}\sqrt{\sum_{i=1}(ry_i - ry)^2}}$$

Signal-to-noise ratio. Signal-To-Noise Ratio uses a ratio of the clean signal to the difference between clean and noisy signals to represent how much signal is actually getting through. The higher the value of SNR the better quality the data.

$$L_{snr}(x,y) = \frac{\sum_{i,j} y_{i,j}}{\sqrt{\sum_{i,j}(x_{i,j} - y_{i,j})^2}}$$

Structural similarity index. SSI is calculated by sliding windows between images and averaging values. The constants C_1 and C_2 are used to stabilize the metric while the means, variances and covariances are computed via a Gaussian filter. We use the implementation of SSI developed by Hong et al.[8] (DeepHiC) keeping their default values for the size of sub-windows and variance value of gaussian filter at 11 and 3 respectively.

$$L_{ssi}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Hi-C reproducibility metrics. We consider 3 Hi-C specific reproducibility metrics: GenomeDISCO[20], HiCRep[21], and QuASAR-Rep[22]. We use the 3DChromatin_ReplicateQC[23] implementations of the metrics. This 3DChromatin_ReplicateQC repository also included metrics for the tool HiC-Spector[22], however we consistently obtained faulty values, even when using the repositories sample data and so we excluded HiC-Spector results from this analysis. When expanding previous models to a 269 × 269 window size the HiCSR model repeatedly failed to converge using these metrics, thus we only include the original 40 × 40 window version of HiCSR in our evaluation of Hi-C Reproducibility metrics. GenomeDISCO utilizes a random walk on a graph generated by contact maps to obtain a concordance score[23]. HiCRep develops a stratum adjusted correlation coefficient for matrix comparison by measuring weighted similarity of contacts in identified stratum[23]. QuASAR-Rep calculates a correlation matrix of interaction using weights based on enrichment[22].

Topologically associated domain identification. Topologically associated domains were identified using Insulation score as identified in Crane et al. We mimicked their procedure entirely with the exception that our initial insulation score window size was condensed to 20 bins instead of 50 because this demonstrated greater visual accuracy in TAD positioning[12].

Three-dimensional model reconstruction. To generate models, we utilize 3DMax[24] with out-of-the-box parameters of 0.6 conversion factor, 1 learning rate, and 10,000 max iteration. We create 3 models per input contact matrices. We generate models for every 5th 269 Mb × 269 Mb input matrix from our training dataset, because this skipping distance ensures coverage of each chromosome while minimizing model generation time. Similarity between structures was measured using TM-score[15].

Motivation for 269 × 269 window size. The decision to expand our window size to 2.69 × 2.69 Mb is multifaceted. Philosophically the decision to expand beyond the previous standard of 0.4 Mb × 0.4 Mb was to permit the inclusion of a wider range of genomic information in our deep learning methods.

From a technical standpoint our insulation score loss is based on the previously defined method for insulation determination[12]. Because calculation of insulation necessitates incorporation of boundary bins, the length of an insulation vector is always smaller than the dimension of a Hi-C contact maps axis with formula:

$$(\text{Len of insulation vector}) = (\text{Len of Hi-C Axis}) - (\text{insulation window}) - (2 * \text{delta window} - 1).$$

Thus, using a 20 kb insulation window and 10 kb Delta window with the previously applied 40 × 40 window would result in an insulation vector of length (40-20-19) = 1, which is only a scalar and would contain insufficient information for meaningful feature extraction.

The decision to use 269 as opposed to a different, large number is due to our variational autoencoder. While passing through the variational autoencoder the dimension of an input matrix is compressed with each incremental layer. It was essential that at each step the output dimension remained a whole number and that when the latent representation is decoded back into contact matrix space the reconstructed matrix be of the same dimension as its input. 257 was the smallest number which both spanned 2 Mb (a range that would encompasses nearly all TADs) and resulted in the same dimensional input and output at each layer of our variational autoencoder. We account for the 12 bin decrease in size that occurs by passing through our GAN, resulting in a 269 × 269 matrix.

Data availability

All Hi-C data were downloaded from the Gene Expression Omnibus (GEO) GSE63525. For the Hi Resolution Matrices of GM12878, IMR90, K562 and HMEC we used GSE63525_GM12878_insitu_primary+replicate_combined_30.hic, GSE63525_IMR90_combined_30.hic, GSE63525_K562_combined_30.hic and GSE63525_HMEC_combined_30.hic respectively. For low resolution matrices we used GSM1551550_HIC001_30.hic, GSM1551602_HIC053_30.hic, GSE63525_K562_combined_30.hic, and GSM1551610_HIC061_30.hic respectively.

Code availability

VEHiCLE was built using python. All experimental code as well as the VEHiCLE enhancement tool and Contact Matrix generating GUI are available at https://github.com/Max-Highsmith/VEHiCLE with zenodo https://zenodo.org/badge/latestdoi/339535370.

Received: 14 December 2020; Accepted: 10 March 2021
Published online: 23 April 2021

References

1. Lieberman-Aiden, E. *et al.* Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**, 289–293 (2009).
2. Miele, A. & Dekker, J. Long-range chromosomal interactions and gene regulation. *Mol. BioSyst.* **4**, 1046 (2008).
3. Oluwadare, O., Highsmith, M. & Cheng, J. an overview of methods for reconstructing 3-D chromosome and genome structures from Hi-C data. *Biol. Proced. Online* **21**, 7 (2019).
4. Lajoie, B. R., Dekker, J. & Kaplan, N. The Hitchhiker's guide to Hi-C analysis: Practical guidelines. *Methods* **72**, 65–75 (2015).
5. Zhang, Y. *et al.* Enhancing Hi-C data resolution with deep convolutional neural network HiCPlus. *Nat. Commun.* **9**, 750 (2018).
6. 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).
7. Liu, Q., Lv, H. & Jiang, R. hicGAN infers super resolution Hi-C data with generative adversarial networks. *Bioinformatics* **35**, i99–i107 (2019).
8. Hong, H. *et al.* DeepHiC: A generative adversarial network for enhancing Hi-C data resolution. *PLoS Comput. Biol.* **16**, e1007287 (2020).
9. Dimmick, M. C., Lee, L. J. & Frey, B. J. HiCSR: A Hi-C Super-Resolution Framework for Producing Highly Realistic Contact Maps. https://doi.org/10.1101/2020.02.24.961714.
10. Highsmith, M., Oluwadare, O. & Cheng, J. Deep Learning For Denoising Hi-C Chromosomal Contact Data. https://doi.org/10.1101/692558.
11. Kingma, D. P. & Welling, M. An Introduction to Variational Autoencoders. https://doi.org/10.1561/9781680836233 (2019).
12. Crane, E. *et al.* Condensin-driven remodelling of X chromosome topology during dosage compensation. *Nature* **523**, 240–244 (2015).
13. Kraft, K. *et al.* Serial genomic inversions induce tissue-specific architectural stripes, gene misexpression and congenital malformations. *Nat. Cell Biol.* **21**, 305–310 (2019).
14. Zhao, L. *et al.* Chromatin loops associated with active genes and heterochromatin shape rice genome architecture for transcriptional regulation. *Nat. Commun.* **10**, 3640 (2019).
15. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins Struct. Funct. Bioinform.* **68**, 1020–1020 (2007).
16. Rao, S. S. P. *et al.* A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* **159**, 1665–1680 (2014).

17. Trieu, T. & Cheng, J. 3D genome structure modeling by Lorentzian objective function. *Nucleic Acids Res.* **45**, 1049–1058 (2017).
18. Dali, R. & Blanchette, M. A critical assessment of topologically associating domain prediction tools. *Nucleic Acids Res.* **45**, 2994–3005 (2017).
19. Knight, P. A. & Ruiz, D. A fast algorithm for matrix balancing. *IMA J. Numer. Anal.* **33**, 1029–1047 (2013).
20. 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).
21. Yang, T. *et al.* HiCRep: Assessing the reproducibility of Hi-C data using a stratum-adjusted correlation coefficient. *Genome Res.* **27**, 1939–1949 (2017).
22. Sauria, M. E. G. & Taylor, J. QuASAR: Quality Assessment of Spatial Arrangement Reproducibility in Hi-C Data. https://doi.org/10.1101/204438.
23. Yardımcı, G. G. *et al.* Measuring the reproducibility and quality of Hi-C data. *Genome Biol.* **20**, 57 (2019).
24. Oluwadare, O., Zhang, Y. & Cheng, J. A maximum likelihood algorithm for reconstructing 3D structures of human chromosomes from chromosomal contact data. *BMC Genomics* **19**, 161 (2018).

Acknowledgements

This project is partially supported by two NSF grants (no. IOS1545780 and no. DBI1149224). It is also supported by the NIH through the T32 training grant GM008396.

Author contributions

M.H. and J.C. conceived the project. M.H. performed all experiments and drafted the manuscript. J.C. revised the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2021