CHROMFORMER: A transformer-based model for 3D genome structure prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent research has shown that the three-dimensional (3D) genome structure is strongly linked to cell function. Modeling the 3D genome structure can not only elucidate vital biological processes, but also reveal structural disruptions linked to disease. In the absence of experimental techniques able to determine 3D chromatin structure, this task is achieved computationally by exploiting chromatin interaction frequencies as measured by high-throughput chromosome conformation capture (Hi-C) data. However, existing methods are unsupervised, and limited by underlying assumptions. In this work, we present a novel framework for 3D chromatin structure prediction from Hi-C data. The framework consists of, a novel *synthetic data generation module* that simulates realistic structures and corresponding Hi-C matrices, and CHROMFORMER, a transformer-based model to predict 3D chromatin structures from standalone Hi-C data, while providing local structural-level confidence estimates. Our solution outperforms existing methods when tested on unseen synthetic data, and achieves comparable results on experimental data for a full eukaryotic genome.

1 Introduction

A plethora of recent studies have illustrated that the 3D genome organization affects if, when, and how genetic information is expressed [2] to ensure controlled execution of essential cellular processes [20, 19]. As genome misfolding is increasingly linked to different diseases [5, 13], modeling the 3D structure of chromatin can not only provide mechanistic insights on vital cell processes, but also identify structural disease biomarkers [1]. However, there exists no method to experimentally determine 3D genome structure. Chromatin organization is studied implicitly using high-throughput chromosome conformation capture (Hi-C) experiments [12] that produce a *contact map* containing interaction frequencies among tiny DNA fragments (loci) across the genome. In absence of experimental methods, a plethora of computational approaches that infer 3D chromatin structure from Hi-C contact maps have emerged [17]. Most methods map interaction frequencies to 3D Euclidean distances via a parametric *transfer function* and use them as constraints to solve an optimization problem [18]. However, the transfer function differs between organisms and resolutions [26]. Recently, data-driven methods employing manifold learning to learn a 3D representation of Hi-C data without a transfer function assumption have emerged, such as GEM [27]) and REACH-3D [3]. Still, in the absence of ground truth, the learning capacity of these unsupervised methods is limited, as their architecture is not designed to generate encodings resembling 3D structures. In this work we propose a novel framework for predicting 3D chromatin structure, inspired by TECH-3D, a state-of-the-art transfer learning method [6]. In contrast to TECH-3D, we propose a synthetic data generation module that simulates biologically-informed 3D structures and corresponding Hi-C matrices using optimal transport (OT), and CHROMFORMER, a transformer-based model that predicts 3D structures from Hi-C matrices, while also estimating the confidence of the prediction.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

2 Methods

Our proposed framework, presented in Figure 1, consists of a *synthetic data generation* component that simulates paired chromatin structures and Hi-C matrices, and CHROMFORMER, a transformer-based model that is trained on the synthetic data and learns to map Hi-C matrices to 3D chromatin structures. CHROMFORMER also outputs loci-level confidence values using a calibration network that corrects the predicted confidence logits to estimated confidence values.

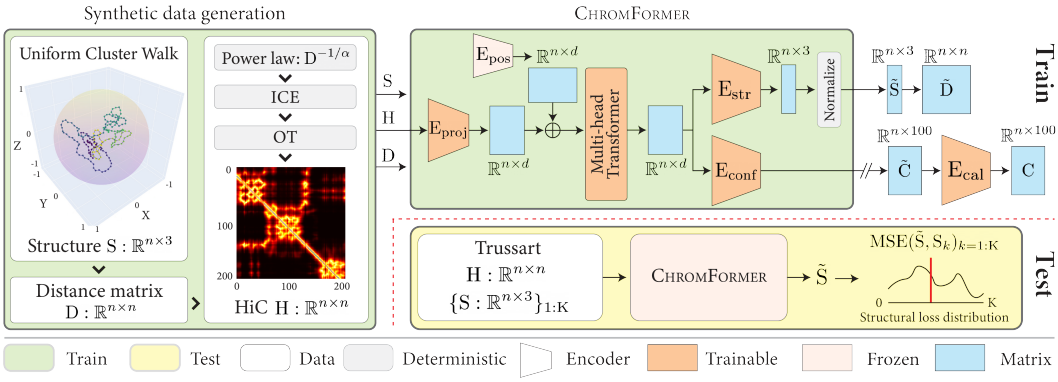


Figure 1: Overview of our proposed chromatin structure prediction framework.

Synthetic Data Generation: To address the lack of ground truth data, in our framework we first propose a Uniform Cluster Walk to generate pairs of synthetic 3D chromatin structures and Hi-C matrices. To resemble real 3D chromatin structures, the synthetic structures need to satisfy the following constraints: i) smoothness and compactness, ii) equidistance of consecutive loci, and iii) presence of clustered regions resembling topologically associated domains (TADs) [16]. To this end, given a number of n loci, we generate a structure $\mathrm{S} \in \mathbb{R}^{n\times 3}$, while employing rejection sampling at different levels to ensure biological plausibility. More formally, for a given locus $\{p_i, \vec{q}_i\}$, we first sample $r \sim \mathcal{U}(-1, 1)$ where $||r||_2 \leq 1$ to ensure isotropy. $\vec{q}_{i+1}$ is then chosen as $(1-\delta)\vec{q}_i + \delta\vec{r}/||r||_2$, where δ is a smoothness parameter, followed by L_2 normalization. The next locus p_{i+1} is derived as $p_i + \vec{q}_{i+1}$. To ensure S is compact, p_{i+1} is subject to rejection sampling according to a Gaussian $\mathcal{N}(p_0, \sigma)$, where p_0 is the starting locus and σ controls the degree of compactness. p_{i+1} is rejected if,

$$\exp\left(-\|p_{i+1} - p_0\|_2/\sigma\right)^2 / \exp\left(-\left\|(p_i - p_0) \times \max\left[0, 1 - 1/\|p_i - p_0\|_2\right]\right\|_2 / \sigma\right)^2 \leq \rho \quad (1)$$

where, ρ is a sampled probability. Further, we create a TAD of length $l << n$, starting at p_i, by using rejection sampling with a Gaussian $\mathcal{N}(p_i, \nu)$ and Equation 1. As TADs are more compact, we set $\nu < \sigma$. We enforce a distance of at least k loci after a TAD ending before being eligible to form another TAD. More details on the structure generation are given in the Appendix A.1. S is then centered and normalized by the maximum L_2 norm of loci in S. The corresponding distance matrix D and Hi-C matrix H are generated using pairwise Euclidean distances between all loci in S, and a power law $\mathrm{D}^{-1/\alpha}$, respectively, where α is a tunable parameter. We normalize H using iterative correction and eigenvector decomposition (ICE) normalization [14] and employ optimal transport (OT) [25] to match the synthetic to real Hi-C matrix. First, the real Hi-C is min-max normalized, and then OT is applied to transform the frequency distribution of the values in the synthetic Hi-C matrix to the frequency distribution of the values in the real Hi-C matrices (more details in the Appendix A.2). Notably, the usage of OT mitigates the distribution gap without any training and without requiring the real chromatin structures.

CHROMFORMER: The synthetic data generator outputs sets of three matrices, $\mathrm{S} \in \mathbb{R}^{n\times 3}$, $\mathrm{D} \in \mathbb{R}^{n\times n}$, and $\mathrm{H} \in \mathbb{R}^{n\times n}$. CHROMFORMER operates on H to predict $\tilde{\mathrm{S}}$, $\tilde{\mathrm{D}}$ and confidence $\tilde{\mathrm{C}} \in \mathbb{R}^{n\times 100}$. CHROMFORMER builds on the hypothesis that the relative spatial distribution influences the formulation of a 3D chromatin structure, and we use a Transformer [24] to exploit this hypothesis. First, CHROMFORMER employs $E_{\text{proj}} : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{n\times d}$ to project H onto lower dimensional embeddings $\mathrm{E} \in \mathbb{R}^{n\times d}$, $d < n$. E is summed with the positional embeddings from $E_{\text{pos}} : \mathbb{Z}^n \rightarrow \mathbb{R}^{n\times d}$ to include the loci spatial information. E is processed by a Multi-head Transformer, consisting of two transformer encoders, which utilizes self-attention to learn the inter-loci relations and contextualizes

E. Subsequently, E is processed by $E_{\text{str}} : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times 3}$, consisting of a linear decoder, to predict 3D loci coordinates. These coordinates are centered and divided by the maximum loci L_2 norm to produce $\tilde{\text{S}}$. $\tilde{\text{D}}$ is derived from $\tilde{\text{S}}$ by computing pairwise Euclidean distances. A major challenge in the addressed task is the lack of ground truth 3D chromatin structures, which inhibits assessment of model predictions. As a proxy, we approximate the prediction uncertainty by producing loci-level confidence scores. Formally, we linearly project E to $\tilde{\text{C}}$, which represents the non-normalised predictions/logits for belonging to 100 confidence interval bins. $\tilde{\text{C}}$ is used for confidence score optimization and model calibration as described below.

Loss objectives: We optimize $\mathcal{L} = \mathcal{L}_d + \lambda_k \mathcal{L}_k + \lambda_c \mathcal{L}_c$, consisting of three loss terms, to tune CHROMFORMER. $\mathcal{L}_d$ is the mean squared error (MSE) between D and $\tilde{\text{D}}$ to ensure the distance preservation. The predicted structure $\tilde{\text{S}}$ can be oriented and scaled differently with respect to S while being similar in structure. To mitigate this, we use the Kabsch algorithm [10] to align $\tilde{S}$ to S, resulting in $\tilde{S}_{\text{aligned}}$. $\mathcal{L}_k$ computes the MSE between S and $\tilde{S}_{\text{aligned}}$. $\mathcal{L}_c$ aims to ensure the correct prediction loci confidence, computed using a cross entropy between Softmax($\tilde{\text{C}}$) and estimated confidence scores C. C is derived using a modified AlphaFold's [9] Local Distance Difference Test [15]. First, an inverse relative error ReLU $\left(1 - |\text{D} - \tilde{\text{D}}|/\text{D}\right)$ is computed to assign low confidence to large deviations between D and $\tilde{\text{D}}$. The error is row-normalized excluding self-interacting diagonal to produce the scores. C is defined as the one hot encoding of $\lfloor \text{scores} \times 100 \rfloor$. All implementation details are given in the Appendix A.3.

Calibration network: In the absence of real ground truth structures, it is crucial to define loci-level confidence scores. $\mathcal{L}_c$ is weighted low during model training to give more emphasize to accurate structure prediction that induces sub-optimal confidence $\tilde{\text{C}}$. We design a calibration network to correct for this and improve confidence estimates on unseen real test data. Additionally, the network can also correct for overconfident predictions in $\tilde{\text{C}}$. To train the calibration network, we use C and $\tilde{\text{C}}$ from the synthetic validation set v, which are then split into $\text{C}v_{\text{tr}}$, $\text{C}v_{\text{val}}$, $\tilde{\text{C}}v_{\text{tr}}$, and $\tilde{\text{C}}v_{\text{val}}$ for network training and validation. We train three networks supporting three calibration techniques, *i.e.*, temperature scaling [7], isotonic calibration [8], and beta calibration [11]. The networks transform $\tilde{\text{C}}v_{\text{tr}}$ to calibrated[$\tilde{\text{C}}v_{\text{tr}}$] to match $\text{C}v_{\text{tr}}$. The performance of the networks is quantified by, MSE(calibrated[$\tilde{\text{C}}v_{val}$], $\text{C}v_{val}$) $-$ MSE($\tilde{\text{C}}v_{val}$, $\text{C}v_{val}$). Upon training, these networks are applied on $\tilde{\text{C}}$, predicted by CHROMFORMER, for the real test data.

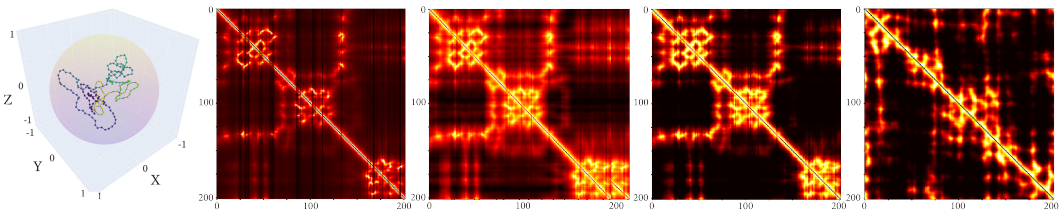
3 Results and Discussion

Public Datasets: We exploit the following publicly available datasets: the **TRUSSART dataset** [23], a set of 100 simulated structures that contain loops, TADs and long-range interactions paired with a single Hi-C matrix (202 loci at 5 kilobase resolution), and the **TANIZAWA dataset** [22], a real experimental single Hi-C measurement across the full fission yeast genome that contains 1258 loci at 5 kilobase resolution together with 18 ground truth pair-wise 3D Euclidean distances as measured by Fluorescence In Situ Hybridization (FISH).

Synthetic dataset: Using the Uniform Cluster Walk algorithm, we generated 1000 and 500 structures with 202 and 1258 loci to match the TRUSSART and TANIZAWA datasets, respectively (see Appendix Figure 4). We follow a 4:1 train-validation split for training CHROMFORMER. The validation set is further split into a 9:1 ratio for the calibration.

Evaluation metrics: For the TRUSSART data, the predicted $\tilde{\text{S}}$ by CHROMFORMER is aligned with the $\{\text{S}_i\}_{i=1:100}$ ground truth structures using the Kabsch algorithm and the mean of $\{\text{MSE}(\text{S}_i, \tilde{\text{S}})\}_{i=1:100}$ is used to quantify the model performance. For the TANIZAWA data, we match the experimentally determined 18 ground truth distances to the equivalent entries in the predicted $\tilde{\text{D}}$, and compute the Pearson correlation between the two sets. To quantify the calibration networks, we compute $\text{MSE}(\text{C}, \text{Softmax}(\tilde{\text{C}}))$ on the Trussart dataset. ReLU $\left(1 - |\text{D} - \tilde{\text{D}}|/\text{D}\right)$ is row-normalized to get C, where D is derived from the mean of the Trussart structures.

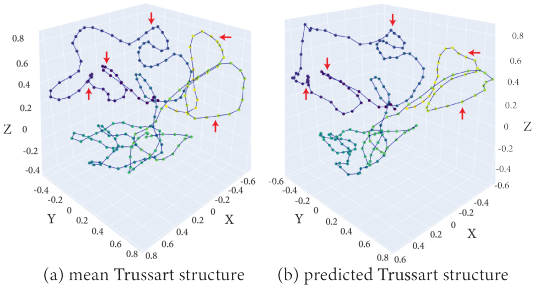
Qualitative and quantitative analysis: Figure 2 displays a synthetic structure S and its corresponding Hi-C matrix H. We clearly observe that S is smooth with equidistant consecutive points and that it includes chromatin loops and TAD-like domains, that are also visible in H. The Hi-C matrices in



(a) synthetic structure (b) synthetic HiC (c) synthetic HiC+ICE (d) synthetic HiC+ICE+OT (e) Trussart HiC

Figure 2: Synthetic structure (a) and Hi-C (d), and its evolution (b-d) w.r.t TRUSSART Hi-C (e).

Figure 2(b) and (e) highlight the distribution gap between the synthetic and the real Trussart data, which is progressively reduced following ICE (c) and OT (d), leading to a final Hi-C matrix that is closer to the real data distribution while retaining the initial Hi-C information. Appendix Figure 5 presents more sample examples. Figure 3 presents the ground-truth Trussart structure (mean of 100 simulations) and the structure as predicted by CHROMFORMER using the Trussart Hi-C. We observe that the predicted structure greatly resembles the real one as it preserves all loops and TADs. Predicted structures by the competing methods are presented in Appendix Figure 7.



(a) mean Trussart structure (b) predicted Trussart structure

Figure 3: Mean ground truth vs. predicted TRUSSART structure. Arrows (in red) show sample regions where loops and TADs are preserved.

Methods	TRUSSART [↓] (Kabsch distance)	TANIZAWA [↑] (Pearson's ρ)
Random	0.2233	-0.1044
MiniMDS	0.1439	0.9092
GEM	0.1437	**0.9600**
REACH-3D	0.0595	0.4831
TECH-3D	0.0642	0.7088
Ours	**0.0205**	0.9047

Table 1: Quantitative benchmarking of our proposed method with competing algorithms on the test datasets. Best scores in **bold**.

Quantitative results on the test datasets are presented in Table 1. CHROMFORMER achieves the lowest Kabsch distance on the Trussart data, outperforming competing methods by a large margin. On the Tanizawa data, CHROMFORMER outperforms the state-of-the-art method TECH-3D [6], with comparable performance to MiniMDS [18] and GEM [27]. Still, qualitative results (Appendix Figure 8) demonstrate that CHROMFORMER reconstructs smooth and loopy structures with a clear distinction between chromosomes, while miniMDS outputs a structure which is a compact set of points without loops and biological feasibility. Among the calibration networks, isotonic calibration results in the best score, followed by beta calibration. The average ground truth confidence, and the average predicted confidence before and after isotonic calibration are 88.90%, 87.40%, and 88.24%, respectively. The average uncalibrated confidence is already good, and the calibration step brings it closer to the upper bound (Appendix Table 2 and Figure 9). On the MSE metric, isotonic calibration produces a score of 8.64 compared to 10.39 before calibration.

4 Conclusion and Future work

In this work we present a novel approach for reconstructing 3D genome structures that overcomes the lack of ground-truth data using a biologically-informed synthetic data generation and a transformer-based architecture. Our model can accurately reconstruct publicly available synthetic structures, outperforming state-of-the-art methods, and shows comparable performance to state-of-the-art algorithms on experimental microscopy measurements from real genomes. A limitation of our framework is data generation, a computationally intensive process that needs to be tuned to tested datasets. We are currently tuning our model hyperparameters and extending its validation to other datasets, especially single-cell Hi-C data [21].

References

[1] Chiara Anania and Darío G Lupiáñez. Order and disorder: abnormal 3d chromatin organization in human disease. *Briefings in functional genomics*, 19(2):128–138, 2020.

[2] Boyan Bonev and Giacomo Cavalli. Organization and function of the 3d genome. *Nature Reviews Genetics*, 17(11):661–678, 2016.

[3] Bianca-Cristina Cristescu, Zalán Borsos, John Lygeros, María Rodríguez Martínez, and Maria Anna Rapsomaniki. Inference of the three-dimensional chromatin structure and its temporal behavior. *arXiv:1811.09619*, 2018.

[4] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[5] Geoffrey Fudenberg, Gad Getz, Matthew Meyerson, and Leonid Mirny. High-order chromatin architecture determines the landscape of chromosomal alterations in cancer. *Nature Precedings*, pages 1–1, 2011.

[6] Tristan Meynier Georges and Maria Anna Rapsomaniki. Modeling the three-dimensional chromatin structure from hi-c data with transfer learning. *bioRxiv*, 2021.

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning (ICML)*, pages 1321–1330, 2017.

[8] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Smooth isotonic regression: a new method to calibrate predictive models. *AMIA Summits on Translational Science Proceedings*, 2011:16, 2011.

[9] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[10] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[11] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics (AISTATS)*, pages 623–631, 2017.

[12] Erez Lieberman-Aiden, Nynke L Van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R Lajoie, Peter J Sabo, Michael O Dorschner, et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950):289–293, 2009.

[13] Darío G Lupiáñez, Malte Spielmann, and Stefan Mundlos. Breaking tads: how alterations of chromatin domains result in disease. *Trends in Genetics*, 32(4):225–237, 2016.

[14] Hongqiang Lyu, Erhu Liu, and Zhifang Wu. Comparison of normalization methods for hi-c data. *BioTechniques*, 68(2):56–64, 2020.

[15] Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

[16] Tom Misteli. The Self-Organizing Genome: Principles of Genome Architecture and Function. *Cell*, 0(0), September 2020. Publisher: Elsevier.

[17] Oluwatosin Oluwadare, Max Highsmith, and Jianlin Cheng. An overview of methods for reconstructing 3-d chromosome and genome structures from hi-c data. *Biological procedures online*, 21(1):1–20, 2019.

[18] Lila Rieber and Shaun Mahony. minimds: 3d structural inference from high-resolution hi-c data. *Bioinformatics*, 33(14):261–266, 2017.

[19] Michael I Robson, Alessa R Ringel, and Stefan Mundlos. Regulatory landscaping: how enhancer-promoter communication is sculpted in 3d. *Molecular cell*, 74(6):1110–1122, 2019.

[20] Jiao Sima and David M Gilbert. Complex correlations: replication timing and mutational landscapes during cancer and genome evolution. *Current opinion in genetics & development*, 25:93–100, 2014.

[21] Longzhi Tan, Dong Xing, Chi-Han Chang, Heng Li, and X. Sunney Xie. Three-dimensional genome structures of single diploid human cells. *Science (New York, N.Y.)*, 361(6405):924–928, August 2018.

[22] Hideki Tanizawa, Kyoung-Dong Kim, Osamu Iwasaki, and Ken-ichi Noma. Architectural alterations of the fission yeast genome during the cell cycle. *Nature structural & molecular biology*, 24(11):965–976, 2017.

[23] Marie Trussart, François Serra, Davide Bau, Ivan Junier, Luis Serrano, and Marc A Marti-Renom. Assessing the limits of restraint-based 3d modeling of genomes and genomic domains. *Nucleic acids research*, 43(7):3465–3477, 2015.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[25] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[26] Siyu Wang, Jinbo Xu, and Jianyang Zeng. Inferential modeling of 3d chromatin structure. *Nucleic acids research*, 43(8):e54–e54, 2015.

[27] Guangxiang Zhu, Wenxuan Deng, Hailin Hu, Rui Ma, Sai Zhang, Jinglin Yang, Jian Peng, Tommy Kaplan, and Jianyang Zeng. Reconstructing spatial organizations of chromosomes through manifold learning. *Nucleic acids research*, 46(8), 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] Not a theoretical work
   (b) Did you include complete proofs of all theoretical results? [N/A] Not a theoretical work
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] testing data is public and cited, synthetic data can be generated from code, code will be included on acceptance of the paper, instructions for reproduction are provided
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] Preliminary work, so hyperparametrising was not yet done
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] Preliminary work, so running multiple experiments multiple time was not done yet
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] license is in the code which will be released upon acceptance
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Data is open source used by many other models without leaking any personal information
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Not applicable to our work

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Not applicable to our work

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Not applicable to our work

A Appendix

A.1 Uniform Cluster Walk algorithm

Algorithm 1: Synthetic structure generation algorithm.

Inputs:

- n: number of loci in a synthetic structure
- σ: degree of compactness of the structure
- ν: degree of compactness of a TAD
- β: probability of creating a TAD
- δ: smoothness coefficient between consecutive loci
- l: number of loci in a TAD
- k: number of aging loci before being eligible to start a new TAD

Initialize:

- p_0: starting locus 3D co-ordinates
- $\vec{q}_0$: starting locus trajectory vector

function GET_NEXT_LOCUS($p_{in}, \vec{q}_{in}, p_{center}, v$) ▷ Get the next locus co-ordinates
  $\rho \sim [0, 1]$
  while $\frac{\exp\left(-\|p_{out}-p_{center}\|_2/v\right)^2}{\exp\left(-\left\|(p_{in}-p_{center})\times\max\left[0,1-1/\|p_{in}-p_{center}\|_2\right]\right\|_2 / v\right)^2} \leq \rho$ **do** ▷ Gaussian rejection criterion
    $r \sim \mathcal{U}(-1, 1)$
    while $\|r\|_2 \geq 1$ **do**
      $r \sim \mathcal{U}(-1, 1)$
    end while
    $\vec{q}_{out} \leftarrow (1-\delta)\vec{q}_{in} + \delta\frac{\vec{r}}{\|r\|_2}$ ▷ Smoothness constraint
    $p_{out} \leftarrow p_{in} + \vec{q}_{out}$
  end while
  return p_{out}, q_{out}
end function

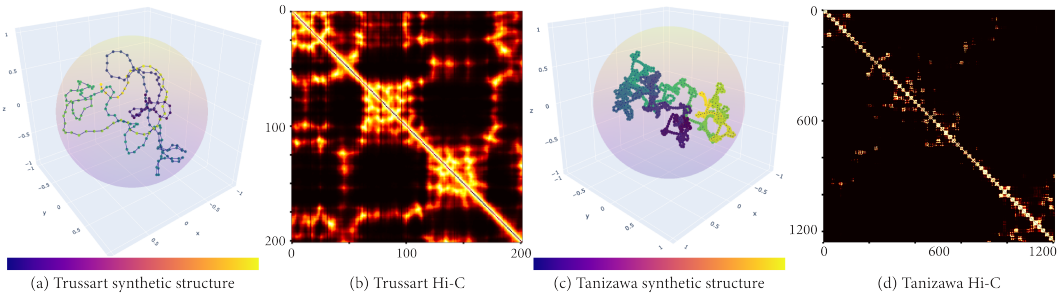
$i_{tadBegin} \leftarrow 0$
$i_{tadEnd} \leftarrow 0$
for $i = 1$ to n **do** ▷ Loop to derive structure defining loci
  if $i < i_{tadEnd}$ **then**
    continue;
  end if
  $t \sim [0, 1]$
  if $t > \beta$ and $i < n - l$ and $i > i_{tadBegin}$ **then** ▷ Begin a TAD
    $i_{tadBegin} \leftarrow i + l + k$
    $i_{tadEnd} \leftarrow i + l$
    $p_{center} \leftarrow p_{i-1}$
    for $j = 0$ to $l - 1$ **do**
      $p_{i+j}, \vec{q}_{i+j} \leftarrow$ GET_NEXT_LOCUS($p_{i+j-1}, \vec{q}_{i+j-1}, p_{center}, \nu$)
    end for
  else
    $p_i, \vec{q}_i \leftarrow$ GET_NEXT_LOCUS($p_{i-1}, \vec{q}_{i-1}, p_0, \sigma$)
  end if
end for

A.2 Optimal transport

We apply OT as an unsupervised domain adaptation method to minimize the differences in values between the synthetic and the real Hi-C matrices by matching the frequency distribution of the synthetic and real values. Figure 5 shows the histograms of the synthetic Hi-C values before and after applying OT. In particular, given a set of synthetic and real Hi-C matrices, we first compute the histograms of the frequencies of the values in the two sets by using 30 uniformly-spaced bins. Then, the histograms are normalized to have a total mass of 1, so that they can be considered discrete probability distributions. Given the two normalized histograms α, β, their supports A, B with equal size ($|A| = |B| = 30$), and squared euclidean distance costs $c : A \times B \to \mathbb{R};\ c(a, b) = (a - b)^2$, the OT problem is defined as follows:

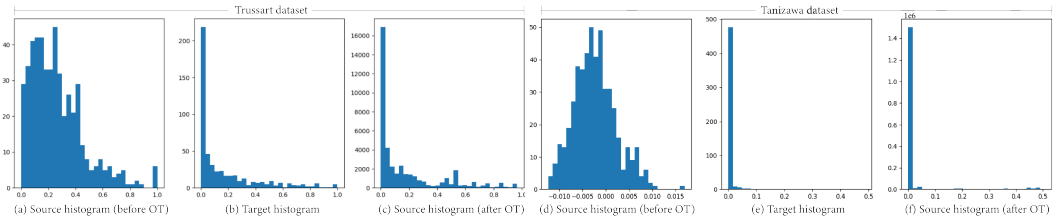
$$\min_{\Gamma} \sum_{(a,b)\in A\times B} c(a,b)\Gamma(a,b) \text{ subject to:}$$
$$\sum_{b\in B} \Gamma(a,b) = \alpha(a) \quad \forall a \in A$$
$$\sum_{a\in A} \Gamma(a,b) = \beta(b) \quad \forall b \in B$$
$$\Gamma(a,b) \geq 0 \quad \forall a \in A, b \in B$$

The solution of the above linear program is the optimal transport plan Γ that maps each bin $a \in A$ of the source histogram α to one or more bins $b \in B$ of the target histogram β. The value of each $\Gamma(a, b)$ denotes the amount of transported mass from $\alpha(a)$ to $\beta(b)$. Finally, the source histogram is transported to the target histogram by using the optimal transport plan Γ. With the transported histogram, the values of the synthetic Hi-C matrices are transformed to match the frequency distribution of the values of the real Hi-C matrices. For the implementation, we used the OT solver from Python Optimal Transport [4] library. Notably, OT transformation is independent of the specific structure of the target Hi-C, and only relies on the frequency distribution of values in the target Hi-C matrices. Figure 2 shows the effect of the OT transformation with source Hi-C (c), target Hi-C (e), and transported source Hi-C (d).



(a) Trussart synthetic structure (b) Trussart Hi-C (c) Tanizawa synthetic structure (d) Tanizawa Hi-C

Figure 4: Pairs of synthetic structures and Hi-C matrices for TRUSSART and TANIZAWA.



(a) Source histogram (before OT) (b) Target histogram (c) Source histogram (after OT) (d) Source histogram (before OT) (e) Target histogram (f) Source histogram (after OT)

Figure 5: Frequency distribution of Hi-C values for TRUSSART and TANIZAWA datasets.

A.3 Implementation details

On a local CPU, it takes 30 minutes to generate the Trussart synthetic data, and 6 hours to generate the Fission Yeast synthetic data. To create the TRUSSART and TANIZAWA synthetic structures, the selected parameters were:

Parameters	TRUSSART	TANIZAWA
# of loci n	202	1258
smoothness δ	0.45	0.2
degree of compactness of a structure σ	4	6
probability of creating a TAD	0.1	0.8
degree of compactness of a TAD σ	1.5	1
# of loci per TAD l	30	15
# of aging loci k	30	5
power law α	1	1

CHROMFORMER model and training parameters were the same for both the datasets:

- embedding dimension d: 100
- E_{proj}: number of layers = 1, hidden dimensions = 100
- # transformer blocks: 2 (dimension of the feedforward networks 100 and 48, respectively)
- # heads in a transformer block: 2
- E_{str}: number of layers = 1, hidden dimensions = 100
- E_{conf}: number of layers = 1, hidden dimensions = 100
- E_{cal}: number of layers = 1, hidden dimensions = 100
- Training: #epochs: 100, batch size: 10, optimized: Adamw with lr: 0.0005, weight decay: 1e-5
- Loss weights: $\lambda_k = 0.1$ and $\lambda_c = 0.1$

Training on a local CPU takes 1 and 5 hours for TRUSSART and TANIZAWA, respectively. Training on a NVIDIA P100 GPU with POWER9 CPU takes 20 and 50 minutes for TRUSSART and TANIZAWA datasets, respectively.

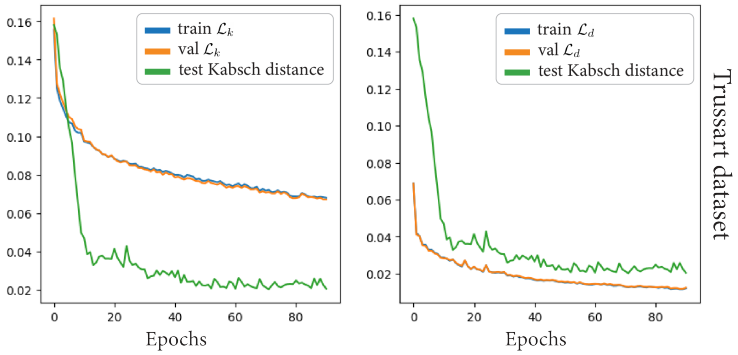
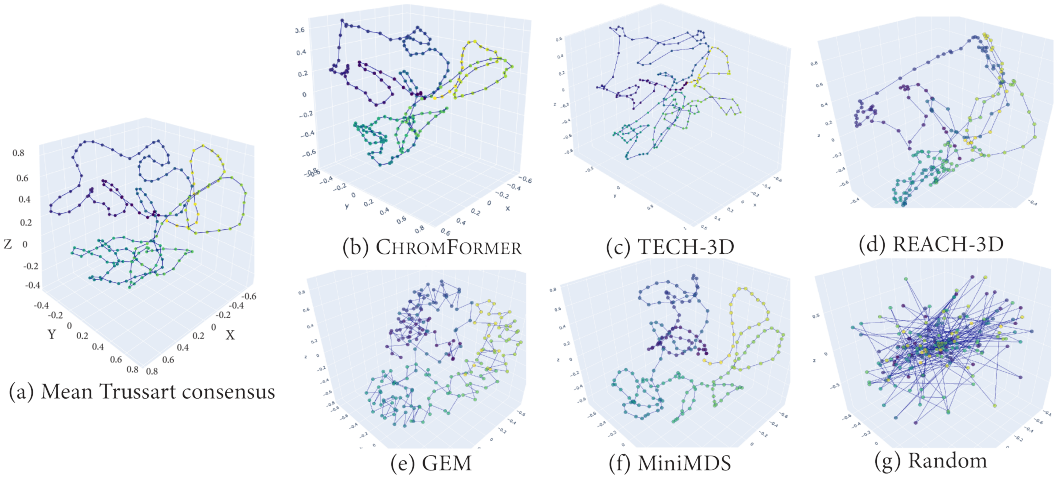


Figure 6: Evolution of $\mathcal{L}_k$ and $\mathcal{L}_d$ losses (train and validation), and evolution of Kabsch distance on test during model training on TRUSSART.

A.4 Quantitative and qualitative results

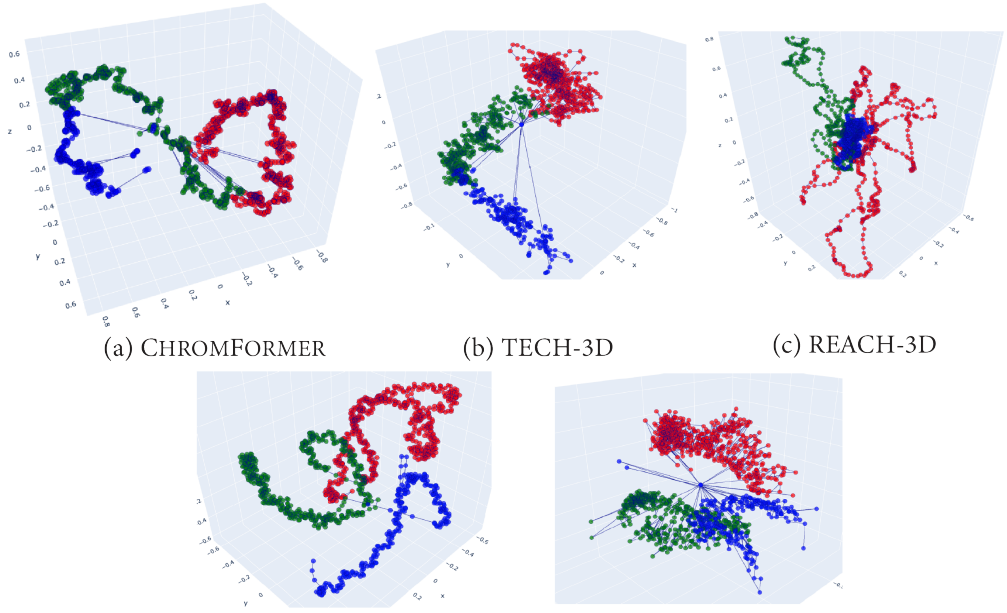
Methods	average(Softmax($\tilde{C}$)) [in %][↑]	MSE(C, 100 × Softmax($\tilde{C}$)) [↓]
Ground truth	88.90	0.00
Before calibration	87.40	10.39
Temperature calibration [7]	87.16	11.16
Isotonic calibration [8]	**88.24**	**8.64**
Beta calibration [11]	88.17	8.72

Table 2: Quantitative results for the calibration networks on TRUSSART data. Best scores in **bold**.



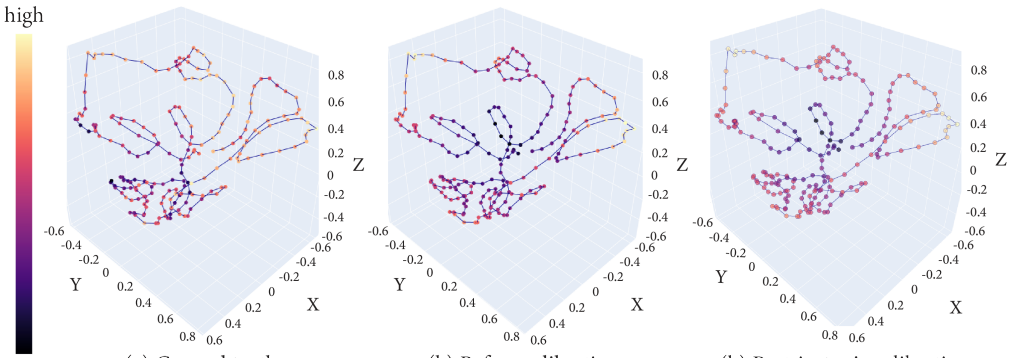
(a) Mean Trussart consensus (b) ChromFormer (c) TECH-3D (d) REACH-3D (e) GEM (f) MiniMDS (g) Random

Figure 7: Predicted chromatin structures on Trussart by different algorithms.



(a) ChromFormer (b) TECH-3D (c) REACH-3D (d) GEM (e) MiniMDS

Figure 8: Predicted chromatin structures on Tanizawa by different algorithms.



(a) Ground truth loci-level confidence (b) Before calibration loci-level confidence (b) Post-isotonic calibration loci-level confidence

Figure 9: Heatmaps of loci-level confidence on Trussart dataset.