Modeling the Three-Dimensional Chromatin Structure from Hi-C Data with Transfer Learning

Tristan Meynier Georges
ETH Zürich, IBM Research Zürich
tmeynier@student.ethz.com

Maria Anna Rapsomaniki
IBM Research Zürich
aap@zurich.ibm.com

Abstract

Recent studies have revealed the importance of three-dimensional (3D) chromatin structure in the regulation of vital biological processes. Contrary to protein folding, no experimental procedure that can directly determine ground-truth 3D chromatin coordinates exists. Instead, chromatin conformation is studied implicitly using high-throughput chromosome conformation capture (Hi-C) methods that quantify the frequency of all pairwise chromatin contacts. Computational methods that infer the 3D chromatin structure from Hi-C data are thus unsupervised, and limited by the assumption that contact frequency determines Euclidean distance. Inspired by recent developments in deep learning, in this work we explore the idea of transfer learning to address the crucial lack of ground-truth data for 3D chromatin structure inference. We present a novel method, *Transfer learning Encoder for CHromatin 3D structure prediction (TECH-3D)* that combines transfer learning with creative data generation procedures to reconstruct chromatin structure. Our work outperforms previous deep learning attempts for chromatin structure inference and exhibits similar results as state-of-the-art algorithms on many tests, without making any assumptions on the relationship between contact frequencies and Euclidean distances. Above all, TECH-3D presents a highly creative and novel approach, paving the way for future deep learning models.

1 Introduction

Eukaryotic genomes are complex, long molecules that carry all hereditary information of an organism in their sequence. In human cells, the ~ 2 -meter-long DNA molecule resides in a nucleus of $\sim 10 \mu\text{m}$ diameter [13]. To achieve this remarkable compression, DNA is packaged around proteins to form chromatin, which is in turn hierarchically organized in loops, topologically associated domains (TADs) and higher-order compartments [11]. This 3D organization allows the genome not only to fit within the confined nuclear space, but also to control *if*, *when* and *how* genetic information is accessed [2]. Indeed, a plethora of studies have highlighted that the 3D structure of chromatin is not random, but rather ensures timely and accurate completion of essential cellular processes, such as gene expression and DNA replication [18, 16]. Distortions of 3D chromatin structure have important implications in disease, leading to rewiring of regulatory interactions and linked to, among others, developmental defects, neurological disorders and, notably, a variety of cancer types [5, 10]. Hence, understanding 3D chromatin structure has the potential to not only lead to a mechanistic comprehension of biological processes, but also elucidate disease pathogenesis [1].

To study chromatin organization, early studies used imaging techniques such as fluorescence in situ hybridization (FISH) [4], which allows measurements of pairwise Euclidean distances between genomic loci. However, due to technical limitations in throughput and resolution, FISH-based methods cannot capture genome-wide organization. To fill this gap, the so-called high-throughput, high-resolution chromosome conformation capture technique (Hi-C) has been proposed [9]. In

contrast to FISH, Hi-C does not measure Euclidean distances but *frequencies of physical interactions*. A typical Hi-C experiment quantifies interaction frequencies (IFs) between N pairs of tiny DNA fragments (high-resolution) across the entire genome (high-throughput), and can be summarized into an $N \times N$ contact matrix (Figure 1). Hi-C can be performed either at a *single-cell* or *population* level. Since single-cell Hi-C contact matrices are sparse and noisy [19], inferring 3D chromatin structure from Hi-C data typically translates to recovering 3D structures from a single population Hi-C contact matrix (Figure 1).

Figure 1: Schematic overview of Hi-C data generation.

Related Work: Existing methods for 3D chromatin structure prediction take as input a population Hi-C matrix and either output a single *consensus* 3D structure, or an *ensemble* of possible 3D structures. Depending on how interaction frequencies are interpreted, these methods can also be categorized as *distance-*, *probability-* or *contact-based* [12]. Assuming that spatial distances and interaction frequencies are inversely correlated, distance- and probability-based methods first translate interactions to distances by employing a *transfer-function*. Then, the problem can be formulated either as traditional multidimensional scaling (MDS) [8] using the translated distances as constraints for distance-based methods (e.g., miniMDS [15]), or as probabilistic modeling for probability-based methods. On the other hand, contact-based methods directly map the interaction frequencies to an output structure(s) by employing manifold learning, with a notable example the method GEM [24] that exploits a modified version of stochastic neighborhood embedding (t-SNE) [22]. Recently, deep learning based methods have also been proposed. For example, REACH-3D [3] infers 3D chromatin structure by performing manifold learning on the Hi-C matrix using a recurrent neural network autoencoder.

However, current approaches suffer from several limitations. Firstly, with the exception of miniMDS that employs a divide and conquer strategy, existing methods are computationally demanding and do not scale well with high-resolution Hi-C data. Secondly, with the exception of GEM, most methods assume a fixed parametric transfer function mapping interaction frequencies to distances, but the exact mapping from interactions to distances differs between organisms and resolutions [23]. Thirdly, although REACH-3D circumvents both these limitations, its main drawback is the lack of ground-truth training data. Specifically, REACH-3D attempts to purposely overfit a single noisy Hi-C matrix, expecting that its latent representation can capture 3D structure. Since the model is unsupervised, its learning capacity is very limited and strongly depends on the noise level. Furthermore, REACH-3D ensures no control over the latent space, as its architecture cannot enforce it to actually encode for the 3D structure. The lack of ground truth data is thus a fundamental obstacle in the design of new deep learning models, not only limiting model validation and benchmarking, but also prohibiting the design of supervised approaches towards chromatin structure prediction. To address all these limitations, in this work we propose a novel deep learning approach called **Transfer learning Encoder for CHromatin 3D structure prediction (TECH-3D)**.

2 Methods

TECH-3D addresses the above limitations by exploiting *transfer learning*. We first go back to a more general and non-biological setting of **predicting any 3D structure from a synthetic contact frequency matrix**. This setting allows us to work in a (simpler) *supervised* setting and have complete control on the recovered structures. More formally, we define two problems:

- A:** Predicting 3D *chromatin* structure(s) from a single noisy *Hi-C* matrix
B: Recovering *synthetic* ground-truth 3D structures from *synthetic* *contact matrices*

Although solving problem **A** is the goal of TECH-3D, problem **B** is easier to solve and, if the set of possible structures synthetically generated in **B** contains the set of true biological structures found in **A**, then **B** is more *generic* than **A**, a favorable condition to use transfer learning and solve **A** by exploiting **B**. Concretely, we develop a neural network Encoder as illustrated in Figure 2, trained on synthetic data from **B** and tested on experimental Hi-C matrices. The Encoder (Figure 2) first maps the synthetic Hi-C matrix to a *consensus* structure in the unit ball and then generates a corresponding distance matrix. We consider two types of neural network architectures for the Encoder: a Feed Forward network or a bidirectional Long Short-Term Memory (Bi-LSTM) network [17], as illustrated in the Appendix (Figures 8 and 9, respectively). Algorithm 1 summarizes the main steps of TECH-3D, and Sections 2.1 and 2.2 describe in detail the synthetic data generation and the loss function of TECH-3D, respectively.

Figure 2: Overview of the TECH-3D implementation.

Algorithm 1 TECH-3D Framework

Input: An experimental $N \times N$ Hi-C matrix

Output: A predicted 3D consensus structure

- 1: **Generate** synthetic samples (contact matrix, consensus structure, distance matrix) of N bins
 - 2: **Train** the Encoder on the generated synthetic samples
 - 3: **Test** the Encoder on the input Hi-C matrix and **output** the resulting structure
-

2.1 Synthetic Dataset

To our knowledge, the only publicly available synthetic dataset in the literature is the one developed by Trussart et al. [21] that contains 100 ground-truth 3D structures mapping to a single Hi-C matrix at various resolutions. The data is obtained by generating an ensemble of 100 hypothetical 1-megabase-long chromosome structures, from which a single synthetic Hi-C matrix is computed by mimicking the actual Hi-C protocol. Since in the Trussart data all 100 synthetic structures map to the same Hi-C matrix, we cannot exploit this dataset as training samples for problem **B**. For this reason, we designed

our own procedure to generate synthetic data, where each structure comes with its own accompanying Hi-C matrix, as follows:

- 1: Produce a 3D structure S .
- 2: Compute the Euclidean distance matrix D from S .
- 3: Compute the contact matrix: $C = \frac{1}{D} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

We add noise to C such that the overall relationship between C and D resembles an inverse proportion while capturing biological variability. We consider two different heuristics to create structures, resulting in two different datasets, B_1 and B_2 , described in detail in subsections 2.1.1 and 2.1.2, respectively. Since TECH-3D heavily depends on training data, we thoroughly investigated which training dataset performs the best when testing the model on problem A.

2.1.1 Synthetic Random Walk Structures

We first considered Synthetic Random Walk Structures (B_1), generated by uniformly sampling, at each step N , coordinates x, y and z from the set S of combinations $[i, j, k]$, where $i, j, k \in \{-1, 0, 1\}$, with $[0, 0, 0]$ excluded (i.e., S has size $3^3 - 1$). More formally:

$$\begin{aligned} x_0, y_0, z_0 &= 0 \\ \text{step}_{i+1} &= \text{randomly choose a step in } S \\ [x_{i+1}, y_{i+1}, z_{i+1}] &= [x_i, y_i, z_i] + \text{step}_{i+1}, \quad i \in \{0, \dots, N-2\} \end{aligned}$$

Figure 3 gives an example of a generated sample, that consists of pairs of synthetic 3D structures (a) with their corresponding Hi-C matrices (b). The Synthetic Random Walk Structure lacks loops and is less smooth than a real genome. Furthermore, the distance between consecutive bins is not constant (this should be approximately the case in a true Hi-C experiment). However, it still shares some common properties with genomes: it exhibits *in cis* interactions (diagonal of Figure 3b), cluster-like domains that resemble TADs (squares around the diagonal of Figure 3b) and long-range interactions (off-diagonal elements in Figure 3b), as well as an elongated overall shape.

Figure 3: Output sample pair for the Synthetic Random Walk Structure heuristic for $N = 256$ bins.

2.1.2 Synthetic Biological Structures

In this heuristic, we aim to generate 3D structures that are *biologically relevant* and more realistic than Synthetic Random Walks. In particular, we design Synthetic Biological Structures (B_2) with a constant distance between consecutive bins and "smooth" loops. Starting from the origin $[0, 0, 0]$, we generate coordinates as follows:

Algorithm 2 We denote by $\vec{v}(r, \theta, \phi)$ the vector starting from the origin and pointing to (r, θ, ϕ) in spherical coordinates (see Appendix Figure 10). Additionally, let $\vec{v}_i$ be the vector starting from bin i and pointing to bin $i + 1$ in the synthetic structure. To generate a structure, the algorithm recursively computes vectors $\vec{v}_i$ for $i \in \{0, \dots, N-1\}$ starting with $\vec{v}_0 = \vec{v}(1, \theta_0, \phi_0)$, where the

angles $\theta_0 \in [0, \pi]$ and $\phi_0 \in [0, 2\pi]$ are chosen uniformly. Now, $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_{N-1}$ needs to be chosen so that conditions (1): constant distance between consecutive bins, and (2): "smooth" loops, are satisfied. Condition (1) translates to $\|\vec{v}_0\| = \|\vec{v}_1\| = \dots = \|\vec{v}_{N-1}\| = 1$. Satisfying condition (2) is more involved. If $\vec{v}_i$ and $\vec{v}_j$ are respectively the first and the last vectors of a loop, then:

- For $i \leq k < j$, the angle $\angle(\vec{v}_k, \vec{v}_{k+1})$ is small, resulting in smoothness inside the loop.
- For $i \leq k < j$, the vector $\vec{v}_{k+1}$ approximately points in the same direction as the previous vector $\vec{v}_k$ but with $\vec{v}_{k+1}$ slightly more rotated towards the center of the loop than $\vec{v}_k$. Formally, let θ_k and ϕ_k be the spherical coordinates of $\vec{v}_k$. Then, $\vec{v}_{k+1}$ has coordinates $\theta_k + \Delta\theta_{k+1}$ and $\phi_k + \Delta\phi_{k+1}$, where $\text{sign}(\Delta\theta_i) = \dots = \text{sign}(\Delta\theta_k) = \text{sign}(\Delta\theta_{k+1}) = \dots = \text{sign}(\Delta\theta_j)$, resulting in equal direction of rotation for all vectors in the loop. Similarly, $\text{sign}(\Delta\phi_i) = \dots = \text{sign}(\Delta\phi_k) = \text{sign}(\Delta\phi_{k+1}) = \dots = \text{sign}(\Delta\phi_j)$.

When and how to create a new loop? We create a new loop by randomly changing the sign of $\Delta\theta_{j+1}$ or $\Delta\phi_{j+1}$ with small probability. By doing this, the vector $\vec{v}_{j+1}$ does not follow the same "loop pattern" as $\vec{v}_j$. Concretely, the update rule is:

$$\begin{aligned}\vec{v}_1 &= \vec{v}(1, \theta_1, \phi_1) = \vec{v}(1, \theta_0 + \mathbb{1}_1 \Delta\theta_1, \phi_0 + \mathbb{1}'_1 \Delta\phi_1) \\ \vec{v}_2 &= \vec{v}(1, \theta_2, \phi_2) = \vec{v}(1, \theta_1 + \mathbb{1}_2 \Delta\theta_2, \phi_1 + \mathbb{1}'_2 \Delta\phi_2) \\ \vec{v}_3 &= \dots,\end{aligned}$$

$$\text{where: } \Delta\theta_i, \Delta\phi_i \sim U_{[0,a]}$$

$$\text{and: } \mathbb{1}_i, \mathbb{1}'_i \sim \{-1 \text{ with probability: } p_{\text{new loop}}, 1 \text{ with probability: } 1 - p_{\text{new loop}}\}$$

Parameters $p_{\text{new loop}}$ and a control the compactness and smoothness of the generated structures, with larger values of $p_{\text{new loop}}$ indicating stronger compactness and smaller values of a indicating higher smoothness. After experimenting with different values, we set $p_{\text{new loop}} = \frac{1}{5}$ and $a = 1$.

One can notice that generating genome-like structures that preserve biological features (loops, constant distance between consecutive bins, etc.) quickly becomes complex. Figure 4 gives an example of an output sample. As desired, the Synthetic Biological Structure is smoother than the Random Walk Structure and exhibits loops, constant distance between consecutive bins, and an elongated overall shape. However, it does not contain clear TAD-like clusters, or long-range interactions.

Figure 4: Output sample pair for the Synthetic Biological Structure heuristic for $N = 256$ bins.

2.2 Loss

The intuition for constructing an appropriate loss for TECH-3D is to measure the goodness of the predicted structure in terms of the distance between bins, the overall shape of the structure and its biological feasibility. The formula for the loss follows this idea and contains three terms for each feature, respectively:

$$\mathcal{L} = \mathcal{L}_d + \lambda_k \mathcal{L}_k + \lambda_b \mathcal{L}_b$$

To test for the correctness of the distances between bins, we use the mean-squared error (MSE) between the true and predicted distance matrices D and $\hat{D}$:

$$\mathcal{L}_d = MSE(D, \hat{D})$$

To test for the correctness of the overall shape, we compute the MSE between the true coordinates S and the predicted coordinates $\hat{S}$, where the two structures are first aligned in space using the Kabsch algorithm [7]. This metric, called the Kabsch Distance, is a common metric to compare 3D structures in space:

$$\mathcal{L}_k = MSE(S, \hat{S}_{aligned})$$

Finally, $\mathcal{L}_b$ checks for the presence of biological features in the predicted structure, only relevant on Synthetic Biological Structures. Hence, by default $\lambda_b = 0$ when training on Synthetic Random Walk Structures. $\mathcal{L}_b$ enforces the distance between consecutive bins to be approximately constant and the angle between consecutive vectors $\vec{v}_i$ and $\vec{v}_{i+1}$ to be small for $i \in \{0, \dots, N-1\}$ (where $\vec{v}_i$ is the vector pointing from bin i to $i+1$ in the sequence). The first objective is accomplished by minimizing $var(\|\vec{v}_i\|)$. For the second, we penalize the dot product $\vec{v}_i \cdot \vec{v}_{i+1}$ if it's negative. In this case, $\vec{v}_i$ and $\vec{v}_{i+1}$ point in opposite directions, which induces a zigzagging, non-smooth structure. Formally, we minimize $mean(ReLU(-\vec{v}_i \cdot \vec{v}_{i+1}))$ and the loss becomes:

$$\mathcal{L}_b = var(\|\vec{v}_i\|) + mean(ReLU(-\vec{v}_i \cdot \vec{v}_{i+1})),$$

where $\vec{v}_i$ is computed from $\hat{S}$.

3 Experiments and Results

Datasets The absence of real ground-truth biological structures necessitates alternative strategies for model training and validation. To circumvent this issue, we exploit our synthetic data generation framework with publicly available datasets for different tasks:

- **Our Synthetic Data:** The paired synthetic Hi-C matrices and structures generated in Section 2.1, used for TECH-3D training and optimization.
- **Trussart Synthetic Data:** The simulated dataset generated by Trussart et al. [21] (see Section 2.1), containing a single Hi-C matrix and 100 corresponding ground-truth structures (5 kilobase resolution), used for hyperparameter tuning and overall TECH-3D optimization. The Trussart structures are simulated so as to contain loops, TADs and long-range interactions, resembling real genomes.
- **Tanizawa Experimental Data:** The Fission Yeast dataset from Tanizawa et al. [20] that includes Hi-C measurements across the full fission yeast genome with a corresponding set of 18 microscopic FISH distances. We use the Hi-C matrix of 10 kilobases resolution and at timepoint $t=20$ minutes into the cell cycle for independent comparison with existing methods.

Implementation details The implementation is performed in PyTorch 1.9.0 [14]. As mentioned above, the network is first trained on our synthetic dataset that contains ground-truth structures, and the *weights* of the network are learned by gradient descent optimization. In particular, we use the PyTorch implementation of Adam [6] with the following parameters: a learning rate of 0.001, an exponential decay rate for the 1st moment estimates of 0.9, an exponential decay rate for the 2nd moment estimates of 0.999 and an epsilon value set to 10^{-8} . Additionally, we train with early stopping for a number of epochs between 10 and 400. To implement early stopping, the base dataset is divided into 80% training and 20% testing samples. The dataset contains either 1000 samples (Sections 3.1 and 3.2) or 500 samples (Section 3.3) with a batch size of 10. The base dataset has no *validation* samples, since model optimization is performed on biologically relevant data, as explained in Section 3.1 below. The experiments were ran on a 2.7 GHz Intel Core i7 processor with 16 GB memory, in less than an hour. The source code is available here: <https://github.com/PhosphorylatedRabbits/tech3d>.

3.1 TECH-3D optimization

Given the many design choices of architecture, training dataset and hyperparameter values, we first need to optimize training of TECH-3D, so that it can generalize well to unseen, biologically relevant data. Specifically, we determine the hyperparameters λ_k and λ_b (Section 2.2) as well as the best combination of Encoder (Feed Forward or Bi-LSTM) and dataset (Synthetic Random Walk - B_1 or Synthetic Biological - B_2) through testing on the Trussart sample. Although, as discussed, the Trussart dataset cannot be used to train TECH-3D, it offers realistic structures that can be used for model selection and hyperparameter tuning while ensuring that TECH-3D generalizes well to real Hi-C data. The results for all combinations of architecture, dataset and λ values are summarized in Table 1, where the minimum test Kabsch Distance (KB) obtained on the Trussart dataset after training over 1 to 50 epochs is reported. We can observe that overall, the best architecture and dataset combination is a Feed-Forward Encoder trained on Synthetic Random Walk Structures, with $\lambda_k = 0.1$ showing the minimal Kabsch distance across all experiments. Bi-LSTM Encoders show an overall worse performance, potentially because they focus on a close neighborhood of the genome, which creates difficulties in predicting long-range interactions. For the same reason, Synthetic Random Walk Structures constitute the best training dataset, as they contain more long-range interactions. Based on these results, for the remaining of this section we fix TECH-3D as a Feed Forward Encoder and the training set as Synthetic Random Walk Structures.

Table 1: Min Kabsch Distance (KB) obtained on the Trussart test sample at 5 kilobase resolution for various hyperparameter values, architectures and train dataset types.

		Architecture					
		Feed Forward			Bi-LSTM		
		λ_k	λ_b	Min KB	λ_k	λ_b	Min KB
Synthetic Training Datasets	Random Walk	0	0	0.0699	0	0	0.0870
		0.1	0	0.0660	0.1	0	0.0883
		1	0	0.0768	1	0	0.1227
		10	0	0.1583	10	0	0.1707
	Biological	0	0	0.0837	0	0	0.2053
		1	0	0.0974	1	0	0.2325
		1	1	0.0920	1	1	0.2316
		10	1	0.1583	10	1	0.1729
		10	10	0.1729	10	10	0.1757

3.2 Visualization of Predicted Structures

Predictions on our synthetic data After model optimization, we first analyse the structures predicted on our synthetic data. Although the performance on our synthetic data is not relevant from a biological point of view, it provides insights on the complexity of the problem. As seen in Figure 5, the 3D structure predicted by TECH-3D (Figure 5b) highly resembles the the original synthetic structure (Figure 5a). Whereas details are harder to predict because the loss becomes too small, the distance matrix is almost perfectly reconstructed (Figures 5c - 5e). The Kabsch distance on our synthetic structures is on average 6 times smaller than on the Trussart sample (reported in Table 1).

Figure 5: Ground-truth Synthetic Random Walk sample (left), TECH-3D prediction (middle) and squared error difference between the true and predicted distance matrices (right).

Predictions in the Trussart synthetic data We then assess the performance of TECH-3D on the Trussart synthetic data. Since the 100 Trussart structures are highly similar, we compute their mean (see *Mean Trussart Consensus* in Figure 6) to simplify visualizations. Figure 6 displays the ground-truth structure (upper-left), the TECH-3D prediction, the output of several benchmark algorithms (namely REACH-3D, GEM and miniMDS) and random predictions. TECH-3D shows the best performance in terms of Kabsch distance, followed by REACH-3D, GEM and miniMDS. Furthermore, TECH-3D accurately predicts the overall structure, particularly within TADs where complex patterns are reconstructed with detail. However, the predicted structure is not smooth and the distance between consecutive bins is not approximately constant. Surprisingly, miniMDS captures both those features, but its Kabsch distance is high since the predictions within the TADs are noticeably wrong. This is because the Kabsch distance tends to favor predictions where TADs are simply clusters of points, as it is the case for GEM and REACH-3D.

Figure 6: Output of various algorithms, ordered by Kabsch Distance (KD) with respect to the *Mean Trussart Consensus*.

Last, Figure 7 illustrates the distance matrices obtained from two different deep learning approaches: REACH-3D (unsupervised manifold-learning method with recurrent autoencoder) and TECH-3D (supervised method with transfer learning). We observe that REACH-3D clearly clusters the points within the TADs, as illustrated by the black squares on the distance matrix (Figure 7b). This is

characteristic of unsupervised learning, where bins are simply clusters of points inside the 3D latent space to minimize the reconstruction loss. As we expected, transfer learning (TECH-3D) clearly outperforms unsupervised learning (REACH-3D).

Figure 7: TECH-3D and REACH-3D output distance matrices. MSE with respect to the ground-truth *Mean Trussart Consensus* distance matrix given in the sublegends.

3.3 Evaluation on Experimental Data

Finally, we evaluate the TECH-3D performance on true experimental Hi-C data on the Tanizawa dataset. Since here no ground-truth structure is available, we use a small set of 18 genome-wide FISH distances experimentally measured by microscopy. Although the number of distances is very limited, by computing the correlation between actual FISH distances and predicted distances one can access the consistency of the prediction in a real genome. As seen in Table 2, both state-of-the-art algorithms GEM and miniMDS show very high correlation scores. The recent deep learning solutions have lower scores, but we observe a huge improvement from REACH-3D to TECH-3D — illustrating the higher learning capacity of transfer learning versus unsupervised learning.

Table 2: Correlation coefficients ρ between FISH and predicted distances for various algorithms

Algorithm	Pearson's ρ	Spearman's ρ
Perfect	1.0000	1.0000
GEM	0.9600	0.9567
MiniMDS	0.9092	0.9340
TECH-3D	0.7088	0.8287
REACH-3D	0.4831	0.5014
Random	-0.1044	-0.2012

Conclusions

In this work, we describe a creative and novel deep learning approach for genome structure inference, Transfer learning Encoder for CHromatin 3D structure prediction (TECH-3D). Our method employs transfer learning to overcome the lack of ground-truth data in the HiC setting — a major obstacle to the use of machine learning for this problem. When tested on publicly available synthetic structures, TECH-3D can accurately reconstruct them, outperforming state-of-the-art methods. When tested on microscopy measurements from real genomes, TECH-3D shows comparable performance as state-of-the-art algorithms and outperforms previous deep learning models. Above all, our work aims to present a highly creative perspective to the problem of inferring realistic 3D genome structures from Hi-C data, paving the way for future deep learning research in the field. Despite the methodological advancements achieved by TECH-3D, some aspects of the framework can be subject to improvement. Data generation is computationally intensive and can be enhanced by taking a divide-and-conquer approach, similar to miniMDS. Another promising direction could be to investigate new datasets, especially recent single cell Hi-C data (e.g., Dip-C [19], providing contacts for both sets of chromosomes (diploid) at a single-cell level.

References

- [1] Chiara Anania and Darío G. Lupiáñez. “Order and disorder: abnormal 3D chromatin organization in human disease”. en. In: *Briefings in Functional Genomics* (). DOI: 10.1093/bfgp/elz028. URL: <https://academic.oup.com/bfg/advance-article/doi/10.1093/bfgp/elz028/5727923> (visited on 03/02/2020).
- [2] Boyan Bonev and Giacomo Cavalli. “Organization and function of the 3D genome”. en. In: *Nature Reviews Genetics* 17.11 (Nov. 2016), pp. 661–678. ISSN: 1471-0064. DOI: 10.1038/nrg.2016.112. URL: <https://www.nature.com/articles/nrg.2016.112> (visited on 08/20/2018).
- [3] Bianca-Cristina Cristescu et al. “Inference of the three-dimensional chromatin structure and its temporal behavior”. In: *arXiv preprint arXiv:1811.09619* (2018).
- [4] Brandon D Fields et al. “A multiplexed DNA FISH strategy for assessing genome architecture in *Caenorhabditis elegans*”. In: *eLife* 8 (May 2019). Ed. by Brian D Slaughter et al. Publisher: eLife Sciences Publications, Ltd, e42823. ISSN: 2050-084X. DOI: 10.7554/eLife.42823. URL: <https://doi.org/10.7554/eLife.42823> (visited on 04/26/2020).
- [5] Geoff Fudenberg et al. “High order chromatin architecture shapes the landscape of chromosomal alterations in cancer”. en. In: *Nature Biotechnology* 29.12 (Dec. 2011), pp. 1109–1113. ISSN: 1546-1696. DOI: 10.1038/nbt.2049. URL: <https://www.nature.com/articles/nbt.2049> (visited on 08/20/2018).
- [6] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv:1412.6980 [cs]* (Jan. 2017). arXiv: 1412.6980. URL: <http://arxiv.org/abs/1412.6980> (visited on 10/15/2021).
- [7] Gerald R. Kneller. “Superposition of Molecular Structures using Quaternions”. In: *Molecular Simulation* 7.1-2 (May 1991). Publisher: Taylor & Francis _eprint: <https://doi.org/10.1080/08927029108022453>, pp. 113–119. ISSN: 0892-7022. DOI: 10.1080/08927029108022453. URL: <https://doi.org/10.1080/08927029108022453> (visited on 10/15/2021).
- [8] Joseph B. Kruskal. “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis”. In: *Psychometrika* 29.1 (1964). Publisher: Springer, pp. 1–27.
- [9] Erez Lieberman-Aiden et al. “Comprehensive mapping of long-range interactions reveals folding principles of the human genome”. eng. In: *Science (New York, N.Y.)* 326.5950 (Oct. 2009), pp. 289–293. ISSN: 1095-9203. DOI: 10.1126/science.1181369.
- [10] Darío G. Lupiáñez, Malte Spielmann, and Stefan Mundlos. “Breaking TADs: How Alterations of Chromatin Domains Result in Disease”. eng. In: *Trends in genetics: TIG* 32.4 (Apr. 2016), pp. 225–237. ISSN: 0168-9525. DOI: 10.1016/j.tig.2016.01.003.
- [11] Tom Misteli. “The Self-Organizing Genome: Principles of Genome Architecture and Function”. English. In: *Cell* 0.0 (Sept. 2020). Publisher: Elsevier. ISSN: 0092-8674, 1097-4172. DOI: 10.1016/j.cell.2020.09.014. URL: [https://www.cell.com/cell/abstract/S0092-8674\(20\)31155-7](https://www.cell.com/cell/abstract/S0092-8674(20)31155-7) (visited on 09/25/2020).
- [12] Oluwatosin Oluwadare, Max Highsmith, and Jianlin Cheng. “An Overview of Methods for Reconstructing 3-D Chromosome and Genome Structures from Hi-C Data”. In: *Biological Procedures Online* 21.1 (Apr. 2019), p. 7. ISSN: 1480-9222. DOI: 10.1186/s12575-019-0094-0. URL: <https://doi.org/10.1186/s12575-019-0094-0> (visited on 10/15/2021).
- [13] Jyotsana J. Parmar, Maxime Woring, and Christophe Zimmer. “How the Genome Folds: The Biophysics of Four-Dimensional Chromatin Organization”. In: *Annual Review of Biophysics* 48.1 (2019). _eprint: <https://doi.org/10.1146/annurev-biophys-052118-115638>, pp. 231–253. DOI: 10.1146/annurev-biophys-052118-115638. URL: <https://doi.org/10.1146/annurev-biophys-052118-115638> (visited on 10/15/2021).
- [14] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *arXiv:1912.01703 [cs, stat]* (Dec. 2019). arXiv: 1912.01703. URL: <http://arxiv.org/abs/1912.01703> (visited on 10/15/2021).
- [15] Lila Rieber and Shaun Mahony. “miniMDS: 3D structural inference from high-resolution Hi-C data”. In: *Bioinformatics* 33.14 (July 2017), pp. i261–i266. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx271. URL: <https://doi.org/10.1093/bioinformatics/btx271> (visited on 10/15/2021).

- [16] Michael I. Robson, Alessa R. Ringel, and Stefan Mundlos. “Regulatory Landscaping: How Enhancer-Promoter Communication Is Sculpted in 3D”. English. In: *Molecular Cell* 74.6 (June 2019), pp. 1110–1122. ISSN: 1097-2765. DOI: 10.1016/j.molcel.2019.05.032. URL: [https://www.cell.com/molecular-cell/abstract/S1097-2765\(19\)30404-6](https://www.cell.com/molecular-cell/abstract/S1097-2765(19)30404-6) (visited on 12/04/2019).
- [17] Alex Sherstinsky. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network”. In: *Physica D: Nonlinear Phenomena* 404 (Mar. 2020). arXiv: 1808.03314, p. 132306. ISSN: 01672789. DOI: 10.1016/j.physd.2019.132306. URL: <http://arxiv.org/abs/1808.03314> (visited on 10/15/2021).
- [18] Jiao Sima and David M Gilbert. “Complex correlations: replication timing and mutational landscapes during cancer and genome evolution”. In: *Current Opinion in Genetics & Development*. Genome architecture and expression 25 (Apr. 2014), pp. 93–100. ISSN: 0959-437X. DOI: 10.1016/j.gde.2013.11.022. URL: <http://www.sciencedirect.com/science/article/pii/S0959437X13001925> (visited on 05/02/2014).
- [19] Longzhi Tan et al. “Three-dimensional genome structures of single diploid human cells”. eng. In: *Science (New York, N.Y.)* 361.6405 (Aug. 2018), pp. 924–928. ISSN: 1095-9203. DOI: 10.1126/science.aat5641.
- [20] Hideki Tanizawa et al. “Architectural alterations of the fission yeast genome during the cell cycle”. en. In: *Nature Structural & Molecular Biology* 24.11 (Nov. 2017). Number: 11 Publisher: Nature Publishing Group, pp. 965–976. ISSN: 1545-9985. DOI: 10.1038/nsmb.3482. URL: <https://www.nature.com/articles/nsmb.3482> (visited on 06/25/2020).
- [21] Marie Trussart et al. “Assessing the limits of restraint-based 3D modeling of genomes and genomic domains”. eng. In: *Nucleic Acids Research* 43.7 (Apr. 2015), pp. 3465–3477. ISSN: 1362-4962. DOI: 10.1093/nar/gkv221.
- [22] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).
- [23] Siyu Wang, Jinbo Xu, and Jianyang Zeng. “Inferential modeling of 3D chromatin structure”. eng. In: *Nucleic Acids Research* 43.8 (Apr. 2015), e54. ISSN: 1362-4962. DOI: 10.1093/nar/gkv100.
- [24] Guangxiang Zhu et al. “Reconstructing spatial organizations of chromosomes through manifold learning”. eng. In: *Nucleic Acids Research* 46.8 (May 2018), e50. ISSN: 1362-4962. DOI: 10.1093/nar/gky065.

A Appendix

Figure 8: The Feed Forward Encoder architecture (example for $N = 16$). It applies three linear layers and a *ReLu* non-linearity function after each of them.

Figure 9: Overview of the Bi-LSTM Encoder architecture where RNN_{enc} is a LSTM cell (Figure 9b taken from Christopher Olah's blog (source: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>) and LNN_{enc} a linear network.

Figure 10: Visual example of the first steps of a *Synthetic Biological Structure* generation.