DATA MINING AND MACHINE LEARNING METHODS FOR CHROMOSOME CONFORMATION DATA ANALYSIS

A Dissertation

presented to the

Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment of the Requirements for the Degree
Doctor of Philosophy

---

by

OLUWATOSIN OLUWADARE

Professor Jianlin Cheng, Dissertation Supervisor

MAY 2019

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

DATA MINING AND MACHINE LEARNING METHODS FOR CHROMOSOME CONFORMATION DATA ANALYSIS

presented by Oluwatosin Oluwadare, a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

_____

Professor Jianlin Cheng

_____

Professor Marjorie Skubic

_____

Professor Dong Xu

_____

Professor Heather Hunt

DEDICATION

I dedicate this dissertation to my parents. Thanks for always giving me the best.

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my advisor and committee chair Dr. Jianlin Cheng. Without his guidance, suggestions, and support this dissertation would not have been possible. His mentoring has been instrumental to my research productivity and efficiency, and his view about problem solving has influenced me to always have a relentless positive attitude in all situations. I am glad to have had the opportunity to work in his lab.

I would like to thank my committee members Dr. Dong Xu, Dr. Marjorie Skubic and Dr. Heather Hunt, for providing scientific guidance, encouragement and advice throughout my time as a student.

Finally, I would like to thank the previous and current members of the Bioinformatics and Data Mining (BDM) lab for the supportive role they played while working with them in the lab. Through the collaboration, with Dr. Tuan A. Trieu, I learnt so many things about my research. I appreciate Dr. Renzhi Cao, Dr. Debswapna Bhattacharya, and Jie Hou, in the research group for their advice and support throughout my time in the group. I also enjoyed a friendly working environment with Anes Quadou, Tianqi Wu, Adil Al-Azzawi, Max Highsmith, and Chen (Chris) Chen. I consider it a privilege to have worked alongside each one of you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Sixteen years after the sequencing of the human genome, the Human Genome Project (HGP), and 17 years after the introduction of Chromosome Conformation Capture (3C) technologies, three-dimensional (3-D) inference and big data remains problematic in the field of genomics, and specifically, in the field of 3C data analysis. Three-dimensional inference involves the reconstruction of a genome's 3D structure or, in some cases, ensemble of structures from contact interaction frequencies extracted from a variant of the 3C technology called the Hi-C technology. Further questions remain about chromosome topology and structure; enhancer-promoter interactions; location of genes, gene clusters, and transcription factors; the relationship between gene expression and epigenetics; and chromosome visualization at a higher scale, among others.

In this dissertation, four major contributions are described, first, 3DMax, a tool for chromosome and genome 3-D structure prediction from Hi-C data using optimization algorithm, second, GSDB, a comprehensive and common repository that contains 3D structures for Hi-C datasets from novel 3D structure reconstruction tools developed over the years, third, ClusterTAD, a method for topological associated domains (TAD) extraction from Hi-C data using unsupervised learning algorithm. Finally, we introduce a tool called, GenomeFlow, a comprehensive graphical tool to facilitate the entire process of modeling and analysis of 3D genome organization. It is worth noting that GenomeFlow and GSDB are the first of their kind in the 3D chromosome and genome research field. All the methods are available as software tools that are freely available to the scientific community.

1 Introduction

After decades of research about the organization of the nucleus of the eukaryotic cell, there exists substantial evidence that the genome architecture plays a key role in nuclear functions. [1–8]. For instance, the spatial arrangement and proximity of genes has been linked to biological functions such as gene replication, regulation, and transcription. [6, 9–11].

The impact of genome architecture on nuclear processes spans multiple hierarchical levels, including the spatial compartmentalization of the process, the higher-order organization of chromatin and the arrangement of the genome within the nucleus. Despite the dynamic nature of their process components, processes such as transcription and DNA repair have been shown to be constrained to specific spatial locations rather than randomly dispersed throughout the nucleus. Genes tend to be more active in sparse euchromatin than dense heterochromatin, purportedly due to the impact of folding density on regulatory factor availability. The homogeneous topology of chromatin has the potential to capture nuclear proteins, affecting their probability of interaction with binding sites. Small, kilo-base sized chromatin loops can localize promoters with upstream elements, while larger mega-base sized loops can spatially segregate nuclear regions, imposing independence on different processes.

Understanding the 3-D organization of the eukaryotic genome is essential to explain the important chromosomal activities within the cell. Hence, a fundamental question in genome and biological studies is how the spatial conformation of the chromosome in the nucleus affects a number of genetic and biological functions such as gene regulation [12, 13], gene expression [14], transcription regulation [15], DNA repair, and DNA replication [16, 17].

Early studies of chromosome conformation relied on the use of cytogenetic techniques, such as Fluorescence In Situ Hybridization (FISH), which has been employed to detect the presence of a

specific chromosome region and the proximity between two regions in a genome sequence [18,19]. Fluorescence in situ hybridization uses fluorescent probes that bind to specific regions of a chromosome with a high degree of sequence complementarity. Using fluorescence microscopy, the location of the *loci* or DNA sequence with which a probe is expected to bind may be determined. This method is especially useful, as it allows direct, one-to-one estimation of genome *loci* proximity. However, due to technical limitations such as low-throughput, low resolution of FISH data, and probe requirements for every analysis, it is not optimal for examining multiple positions simultaneously. As a result, the method is not used when studying the organization of chromosomes at a genome-wide scale.

Other microscopy techniques that have been developed to study the chromatin organization are aimed at providing details about the genome positioning and activities. Some of these methods are called the Super Resolution Microscopy Strategies, as they were developed to provide imaging at a high resolution. Examples include Saturated Structured Illumination Microscopy (SSIM), Stimulated Emission Depletion (STED), and Ground State Depletion (GSD) [20, 21]. The introduction of Stochastic Super-Resolution Microscopy techniques such as Photo-Activated Localization Microscopy (PALM or FPALM), and Stochastic Optical Reconstruction Microscopy (STORM) produced a different set of ways for investigating the chromatin organization [22, 23]. Generally, the microscopy techniques for studying the chromatin organization could be categorized as light *versus* electron microscopy-based techniques. The more detailed description of the microscopy-based techniques for studying genome organization is given in the section "Genome Organization by Microscopy-Based Techniques".

In 2002, Dekker *et al.* [24] developed 3C, a high-throughput methodology that can be used to generate interaction frequency (IF) between nearby genomic *loci* in a cell population. Since then,

a number of 3C variants [25–27] such as 4C [28], 5C [29], Hi-C [30], TCC [31], ChIA-PET [32, 33] and, later on, single-cell Hi-C [34], have been developed to study the 3-D organization of the chromosome and genome. The development of 3C techniques has substantially benefited the study of the spatial proximity, interaction, and genome conformation of a number of cells. Today, Hi-C is the most widely used and well-known 3C variant. Using next-generation sequencing strategies, such as high-throughput and parallel sequencing, Hi-C enables researchers to profile read-pair interactions on an all-versus-all basis—that is, to profile interactions for all read pairs in an entire genome. It also allows them to detect and compute the number of interactions between fragments within a chromosome—*i.e.*, the intra-chromosome IF—or between different chromosomes—*i.e.*, the inter-chromosome interaction frequency. Fragments, alternatively known as bins or genomic *loci*, are the regions into which a chromosome have been divided into. Each fragment has a defined length or size which is the number of base pair (bp) in it. The size of the fragment is determined by the resolution, *e.g.* a 1 MB resolution signifies that 1,000,000 bp are contained within each fragment.

The IFs obtained are commonly represented in a two-dimensional matrix, also known as a contact matrix, with rows and columns representing the number of fragments in the chromosome or genome.

The Hi-C technique is especially relevant because the IFs it yields can be used to construct 3-D chromosome and genome structures. These structures, in turn, help explain a series of events such as genome folding, gene regulations, the connection between regulatory elements, and the higher-order structural features in the nucleus of a cell [1, 2, 14, 35, 36].

Within the past decade, a number of computational methods and algorithms have been proposed for the construction of chromosome and genome 3-D structures from Hi-C data. Most of these

methods adopt different strategies for 3-D structure prediction, have different technical requirements for algorithms, and use different noise reduction techniques to analyze Hi-C data. In this review, we categorize these methods based on how they model IF from Hi-C data, highlight a common approach to method evaluation and validation, and finally point to the future direction and challenges of chromosome and genome 3-D structure prediction.

1.1 Description of the Hi-C Experiment and Chromosomal Contact Map

Using next generation sequencing technology, the emergence of the Hi-C technique, an extension of 3C, has enabled the identification of the chromosome conformation at a genome wide scale [26, 27, 30, 37, 38]. Compared to other variant of the 3C technique, the Hi-C technique is the first method [30, 38] to capture chromosome conformation on a "all versus all" basis —that is, it can profile interactions for all read pairs in an entire genome. The Hi-C protocol begins by using formaldehyde to crosslink the cells, which results in the covalent linking of the chromosomal *loci* through their protein-DNA interactions. The cross-linked chromatin segment is then cut out with a restriction enzyme, and the segment restriction ends are marked by filing in with biotin-labeled nucleotides [25, 30]. Next, the resulting blunt-end segments are ligated randomly under appropriate conditions for ligation events between the cross-linked DNA segments. DNA is purified and sheared, and a biotin pull-down is performed to ensured that only the biotinylated junctions are selected for further high throughput pair-end sequencing and computational analysis. After the sequencing of the pair-reads, the generated output usually in .fastq format is mapped to a reference genome, filtered, and used to create a contact map [39]. Notable tools that support the mapping of the sequenced pair reads to generate contact map are GenomeFlow [40], Juicer [41], HiC-Pro [42], Hi-Cpipe [43], and HiCUP [44].

Interaction frequency (IF), sometimes referred to as contact frequency, is a measure of the number of interactions between a pair of chromosomal or genomic regions in the Hi-C data [45–48]. The combined contact counts for all pairwise regions or *loci* may be represented as a symmetric matrix to form an IF matrix of all interacting fragments. The IF matrix is sometimes also referred to as a contact matrix or contact map [30, 47]. A chromosome contact matrix is a n-by-n matrix representing the interaction of *loci* or chromosomal regions as captured in the Hi-C experiment [27, 30, 31, 49]. The rows and columns of the matrix correspond to the index of the equal-sized regions which partition the chromosome. The length of one equal-sized region (*e.g.*, 1 Mb base pair) is referred to as the resolution [30]. Each entry in the matrix represents a count of read pairs that connect two corresponding chromosome regions in a Hi-C experiment [30]. Alternatively, the contacts can be represented in a 3-column sparse matrix [49], where columns 1 and 2 refer to the genomic location or the fragment number of the interacting *loci* and column 3 represents the IF between them.

1.2 Polymer Model

Polymer models are based on the underlying idea that interactions between molecular subunits, such as monomers, result in large molecular structures known as polymers. This approach was adopted from polymer physics, a branch of statistical physics [50–52]. Polymers produced by living organisms are referred to as biopolymers. Two well-known examples of biopolymers are DNA and proteins, with nucleotides and amino acids as their monomers, respectively. Polymerization involves the combination of small molecules through chemical bonding to form a network at equilibrium called a polymer. Various authors have adopted two states of the polymer to model the architecture of chromosomal regions in a cell: the equilibrium globule [53, 54] and the fractal globule [37, 55, 56]. A characteristic feature of the equilibrium globule model is that it

is highly knotted [30]. Mirny [37] has pointed out that this configuration is disadvantageous, as it restricts genomic processes such as unfolding—an important property for gene activation—or refolding [57]. Alternatively, Barbieri *et al.* [55] showed that polymer collapse after exposure to a topological constraint can result in the formation of a long-lived, untangled, non-equilibrium configuration state called a "crumpled" or "fractal" globule. A fractal globule is knot-free, and it is organized such that it allows for unfolding or refolding processes while in a highly compact state. Hence, the polymer exhibits a "beads-on-a-string" configuration, with beads representing monomers connected by linkers; DNA connections in eukaryotic chromatin are similarly configured. The fractal globule can be illustrated as a dense multicolor ball of yarn, where each color has its own end, but one can pull out threads with a specific color and put them back in, without disturbing the structure of the overall ball at all. This important property makes the fractal globule suitable for organizing chromatin in a cell because this topology facilitates rapid and easy unfolding, refolding [58], and large-scale opening of genome *loci* loop that affects and explains biological processes, *e.g.* the connection of distal single-nucleotide polymorphisms (SNPs) with their target genes, gene activation, gene repression, or the cell cycle [59–63].

When studying these two globules, two biophysical properties are considered: the genomic distance between two *loci* and the probability of contact between them. It is worth noting that genomic distance (s) is measured by FISH, while the contact probability is obtained from chromosome conformation methods such as Hi-C. The equilibrium and fractal globules yield different estimates for these properties, and therefore also varying predictions on the three-dimensional distance between pairs of *loci*. Lieberman-Aiden *et al.* [30] and Mirny [37] report, through simulation, that equilibrium and fractal globule scaling for three dimensional-distance are $s^{1/2}$ and $s^{1/3}$ (s: genomic distance - number of nucleotides between two *loci*), respectively.

Equilibrium and fractal globule scaling for contact probability are $s^{-3/2}$ and s^{-1}, respectively. As shown in [37], the properties exhibited by the fractal globule model make it more effective at fitting Hi-C data than the equilibrium globule.

Some methods adopt the knowledge about polymer chain for chromosome structure representation by simulating a physically realistic, bead-chain polymer model of the 30-nm chromatin fiber [64, 65]. As a result, when constructing either a chromosome structure for instance, a locus for a chromosome is represented using a conventional beads-and-spring polymer model, where each bead represents a specific genomic location with well-defined initial and final genomic coordinates. Hence, viewing the chromatin fiber as a polymer model implies that conformation energies such as bending, stretching, and excluding energies of chromatin segments needs to be considered and integrated with the IF for 3-D structure reconstruction (Figure 1.1a).

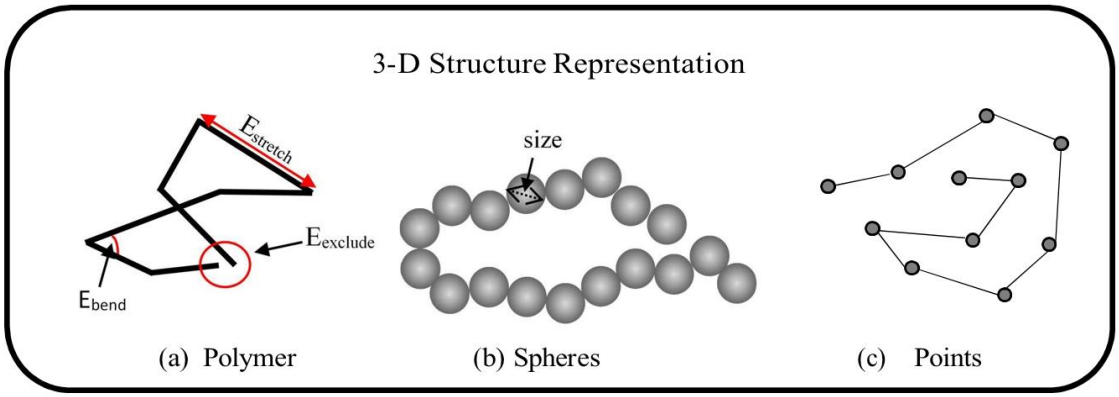


Figure 1.1. Chromosome and genome 3-D structure representation for models from Hi-C data. The different models used for representing 3-D chromosome and genome structure by various methods using Hi-C data for modeling chromosomes and genome 3-D structure. (a) polymer model (b) sphere (c) points.

1.3 Spheres and Points

An alternative structure representation model adopted by methods is representing the chromosome region or *loci* as series of connected spheres or interacting points. Methods using this approach presents the 3-D structure in a simplified model, where the spheres [66–68] or points [45, 46, 69, 70] are synonymous to a chromosome region or *loci* of a chromosome (Figure 1.1b, c). Using a bead on string configuration, each bead is modeled as a spherical shape with a defined radius, and an excluded volume used to penalize overlaps between two spheres. The defined radius and the sphere volume could consequently be considered as a restraint to be satisfied during the algorithm's 3-D structure reconstruction process. The Points representation represent the chromatin region simply as a point, with no radius nor volume, to mark the presence or absence of a *loci*.

1.4 Methodologies for Chromosome and Genome 3-D Structure Reconstruction

The methods for chromosome and genome 3-D structure inference are categorized below based on the IF modeling adopted by them. All methods adopt a stepwise approach to achieve the 3-D structure reconstruction, and a summarization of these steps is provided in Figure 1.2.

1.4.1 Distance-Based Methods

Over the years, a number of approaches have been proposed for chromosome 3-D structure inference from Hi-C contact data. A group of these methods involve a two-step process: (1) IF is converted to distance, ultimately defining the problem of 3-D genome or chromosome structure reconstruction as a problem of converting distances into 3-D coordinates; and (2) non-linear optimization is subsequently applied to the problem in order to find the genomic coordinates that satisfy converted distances. The most notable differences between these proposed methods are: (1) the way in which IF is converted into distance, and (2) the optimization technique used to infer the

3-D structure from *loci* distance. The aim of a distance-based modeling is to create a map that shows the relative spatial positioning of a number of objects whose inter-point distance is known. Additionally, representing chromosome structure prediction as a distance-based modeling problem is tempting because methods based on distances are simple and clear: there is no ambiguity regarding metric definition, and therefore the proximity between objects can eventually be derived. In relation to 3-D genome structure prediction, the distance-based approach makes it easier to handle a large spectrum of modeling problems at different Hi-C data resolutions.

The distance-based approach attempts to reproduce the original metric or distance as accurately as possible. The earliest application of the metric Multi-Dimensional Scaling (MDS) [82, 99] to chromosome 3-D conformation construction, known as 5C3D [45], assumed that the relationship of IF to distance between DNA fragments or *loci* follows an inverse relation; it then used an optimization approach to find the best 3-D conformation through a misfit objective function of the converted distance and the 3-D Euclidean distance between points. While this method was applied to the 5C variant of 3C data, it could be applied to Hi-C datasets as well. Similarly, in their work based on yeast 3-D genome structure reconstruction, Duan *et al.* [66] designed a metric that estimated the corresponding Euclidean distance from the mean of the curves obtained from two restriction enzyme libraries for each contact frequency. To aid modeling and ensure that intra- and inter-chromosomal features (*e.g.*, centromeres), distance, and properties were satisfied [66, 67], researchers introduced a series of constraints such as minimum and maximum distances between adjacent beads, minimum distances between pair beads to avoid overlapping and clashes, specific positioning of RNA coding regions, telomeres, and centromeres to guide the construction of the 3-D model; this constituted an improvement over the previous method. Duan *et al.* used IPOPT [71], an open-source software for nonlinear constrained optimization problems, to minimize the

objective function; this ensured that the predicted coordinates of two interacting *loci*, from which the distance between said *loci* in the 3-D structure is derived, closely matched the expected distance obtained from IF. Tanizawa *et al.* [67] developed a method similar to [66] to construct the 3-D structure of the fission yeast genome.

Although Lieberman-Aiden *et al.* [30] showed that IF can be used to determine the spatial distance between interacting *loci*, certain factors regarding this conversion are still worth considering. As shown by [76, 100–102] in their work, the IF-distance correlation might vary from one dataset resolution to another, and from one organism to another. Hence, an efficient method is required for a distance-based approach to generate a more reliable distance estimate from IF data. To solve this problem, Zhang *et al.* [76] made two novel propositions for the two-step genome structure prediction pipeline. First, they used a modified version of the golden section search method [103] to determine the best scale parameter, conversion factor (α), to convert IF to its approximate distance equivalent: $D_{ij} \propto F_{ij}^{-\alpha}$; this ensures that an appropriate conversion factor is obtained for each dataset. Secondly, for the 3-D structure prediction from a distance matrix, they presented an algorithm called ChromSDE (Chromosome Semi-Definite Embedding). Unlike earlier methods, ChromSDE relaxed the optimization problem to a semi-definite programming (SDP) problem. The proposed approach to IF-distance conversion defined by Zhang *et al.* introduced a new convention for defining the IF-distance relationship, followed by a series of distance-based algorithms that were subsequently developed.

According to Yaffe and Tanay [104], raw Hi-C data obtained from 3C experiments may contain numerous systematic biases, such as GC content, length of restriction fragments, and mappability between fragments. Long-range frequencies are typically noisy and unreliable; this represents a substantial drawback for the construction of 3-D chromosome and genome structures. In order to

overcome these limitations, a number of methods have been developed to pre-process Hi-C data through normalization [9, 42, 104–108] before using the data for 3-D reconstruction. Alternatively, certain algorithms for 3-D structure construction incorporate bias removal. Peng *et al.* [77] proposed a normalization approach to reduce experimental sequencing depth bias, which affects the IF yielded by Hi-C data and makes it hard to compare structures from data obtained from different experiments. The method, called AutoChrom3D, provides an automated pipeline for 3-D modeling, enabling structural comparison at various data resolutions. Two linear transformations were used to determine the frequency-distance correlation, and the structure was then predicted through nonlinear constrained optimization.

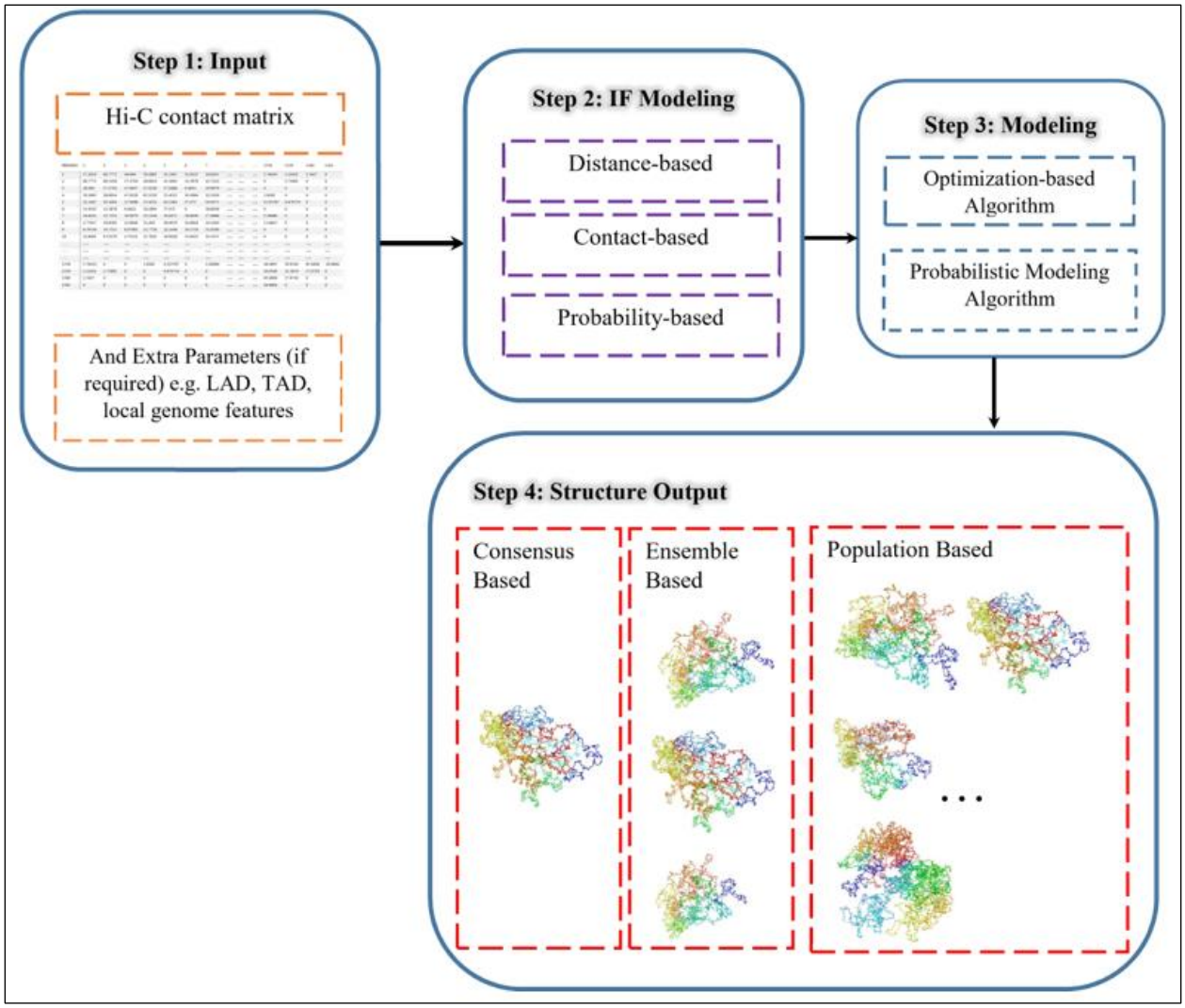


Figure 1.2. Chromosome and genome 3-D structure reconstruction workflow. A summarization of the steps for genome and chromosome 3-D structure taken by the different methods. Starting from the user input in Step 1: The input preparation, usually, Hi-C contact matrix or sometimes with extra parameters requirement. Step 2: One of the three IF modeling approach is used to represent the IF depending on the method's algorithm. Step 3: Modeling is done using defined sampling algorithms, and Step 4, a consensus average structure or a group of structure is generated depending on the method's structure class.

Shavit *et al.* [81] designed an MDS-based optimization approach that used FISH distance to guide the conversion of IF to Hi-C *loci* distances; this approach aimed to reduce noise, improve the data quality, ensure the consistency of data used for 3-D structure construction, and cover key functionality features in the Hi-C and FISH datasets, which will eventually overlap if these features are vital. Zou *et al.* [47] designed a flexible algorithm capable of handling biases introduced by restriction enzymes during Hi-C data sequencing. Restriction enzymes are known to have various cutting sites across the genome, so combining different Hi-C tracks provides further information about genomic *loci* for modeling. The tool developed by Zou *et al.*, called HSA, takes advantage of the uniqueness of the contact map obtained from different restriction enzymes in Hi-C experiments; it creates a generalized linear model through an iterative algorithm that combines simulated annealing and Hamiltonian dynamics. By using HSA, Zou *et al.* discovered that the obtained 3-D structure fits the contact map obtained from different restriction enzymes. Bau *et al.* [72] performed a log transformation and the Z-score computation to normalize the contact counts. They converted observed interactions between *loci* to points and spatial restraints, and used the Integrative Modeling Platform (IMP) [73] to produce possible confirmations that satisfies their defined constraints and maximizes their structure to fit the IF data. Each *loci* was first represented as a point connected by a "string" to create a pairwise interaction in which the length of the string depended on the number of interactions between the *loci*.

To date, a number of other distance-based methods have been developed. These algorithms create 3-D models by first converting contact frequency to distance [9, 46, 69, 70, 77, 88, 97, 109, 110] and then apply optimization to predict chromosome structure. Usually, these methods perform chromosome 3-D reconstruction by first defining a random 3-D structure; this structure coordinates are then updated by an objective function that is iteratively optimized until a

convergence condition is satisfied. Chromosome3D [46], applied a modified version of the distance geometry simulated annealing (DGSA) based method for chromosome and genome 3-D structure reconstruction from Hi-C data. The DGSA method has been popularly used for protein structure construction over the years and implemented in the Crystallography & NMR System (CNS) suite [111, 112]. The Hi-C distances are used as restraints for the defined simulated annealing (SA) optimization pipeline. SA is carried out through multiple steps of temperature change until the defined structure energy is optimally minimized. Because Chromosome3D uses one of the rigorously tested approaches in protein structure to inferring chromosome and genome 3-D structure, it is reliable and robust against noise in Hi-C data.

LorDG [69] introduced a novel method to address inconsistent chromosomal contacts generated from multi-cell Hi-C data. It used a nonlinear Lorentzian function as the objective function—to enforce the satisfaction of consistent restraints, which is resistant against noisy distance restraints. Unlike the square error function that is susceptible to outliers, LorDG aims to maximize the satisfaction of realistically satisfied restraints rather than unsatisfiable noisy ones. The objective function is optimized by the highly scalable adaptive step-size gradient descent method. Its resilience against noisy contacts and scalability make it a suitable method for constructing the structure of the entire genome involving noisy inter-chromosomal contacts. 3DMax [70] defined a maximum likelihood objective function for chromosome 3-D structure inference from Hi-C data. It is based on the simplified assumption that the contact data is normally distributed and that each Hi-C data point is conditionally independent given a structure. A log likelihood objective function for chromosome structure reconstruction was defined in order to determine the structure that maximizes the likelihood function. 3DMax uses a variant of gradient ascent called Adagrad [113] that adapts the learning rate to each objective function parameter automatically to regulate its

learning rate. 3-DMax is robust against noise and structural variability, and it is computationally fast and memory efficient.

miniMDS [92] and Hierarchical3DGenome [98] are the distance-based algorithms that reconstruct high-resolution 3-D models at the topologically associating domain (TAD) level. Eventually, these TAD models are assembled to form a complete, high-resolution 3-D chromosomal structure. After the assembly of TAD models, Hierarchical3DGenome uses the contacts between all regions in a chromosome to further refine the assembled whole chromosome model, which leads to high-resolution (*e.g.* 5 KB) models of good quality.

The conformational space of a chromosomal structure is large, given that Hi-C data are drawn from a population of cells, each with its own independent and unique 3-D structure. Hence, an ensemble of predicted structures obtained through so-called ensemble-based modeling appears to provide a better representation of chromosomal structure than a single structure obtained through consensus modeling. Unfortunately, like Hi-C data at large, this dataset contains a number of biases: the fact that it is noisy, coupled with other technical factors, makes it extremely difficult to determine the various unique 3-D structures of cells used in Hi-C experiments. Due to the drawbacks involved in using multi-cell Hi-C data, studying single-cell Hi-C data has become increasingly relevant [34]. In particular, it does not require designing an algorithm to satisfy the variability of each cell used in the Hi-C experiment. As expected, single-cell Hi-C datasets are sparser than multi-cell Hi-C datasets. Hence, conventional distance and restraint-based methods are not suitable for 3-D structure reconstruction based on these data. Carstens *et al.* [90] extended Rieping *et al.*'s [114] Bayesian probabilistic framework to statistically infer ensembles of 3-D chromosome structures from single-cell Hi-C data using MCMC sampling. They combined single-cell Hi-C contact information with FISH data and a csoarse-grained model of the chromatin fiber.

Lesne *et al.* [79] formulated a two-step algorithm known as "shortest-path reconstruction in 3-D" (ShRec3D), which combines the shortest-path distance between two points from graph theoretic methods with MDS to achieve chromosome reconstruction. This method is designed for both multi-cell and single-cell Hi-C data. In the case of single-cell Hi-C data, instead of distances between two points, binary numbers signify the presence or absence of interaction. ShRec3D+ [96] extended Lesne *et al.*'s algorithm by using a golden-section algorithm (an approach similar to Zhang *et al.* [76]) with an adaptable distance conversion factor for different Hi-C chromosome datasets. Wang *et al.* [64] proposed a method that combined knowledge of the conformational energy model of a chromatin structure and a Bayesian inference approach. They represented the chromosome structure as a polymer model with a conformational energy, and integrated the IF data as input for an expectation maximization based algorithm under a Bayesian like framework. They took advantage of the prior information about the conformation energy to construct a Bayesian inference of the chromatin structure. An approach proposed by Paulsen *et al.* [84] employed manifold-based optimization (MBO), which is basically the application of optimization techniques to the manifold of positive semi-definite matrices of fixed rank [115]. Paulsen *et al.* reported that MBO is capable of generating a consensus 3-D chromosome structure consistent with the original contact map.

Another approach for solving the distance-based problem is called Non-Metric MultiDimensional Scaling (NMMDS), which assumes that only distance ranks are known; distances themselves are not provided. The method aims to yield a map of these ranks [116, 117]. Using this approach, Ben-Elazar *et al.* [118] developed a method for structure prediction based on the hypothesis that a pair locus A with a higher IF is closer in 3-D space than any other locus pair B with a lower frequency.

Varoquaux *et al.* [78] also proposed an optimization method to solve the NMDS problem by minimizing the Shepard-Kruskal scaling cost function [119].

1.4.2 Contact Based Methods

Certain methods do not convert IF but use it directly for modeling. These methods are regarded as contact-based methods [15, 80, 83, 91, 93]. MOGEN [49, 80] used contact directly and designed an optimization-based approach that relied mostly on Hi-C intra- and inter-chromosomal contact data to build an ensemble of 3-D conformations for genome and chromosome structures. The contact-based optimization is carried out by the adaptive step-size gradient descent/ascent method that is highly scalable and therefore is well suited for large-scale genome structure modeling. MOGEN does not require two contacted regions to satisfy a specific distance as the distance-based approach does. Instead it only tries to make the distance between the two contact regions below a threshold (*i.e.* in contact). MOGEN is capable of producing ensemble models that are highly consistent with each other. MOGEN is also robust against noise in the data, particularly the noise in inter-chromosomal contacts, and therefore it is able to build 3-D structures of large genomes such as the entire human genome. Gen3D [83] used a series of meta-heuristic algorithms (*e.g.* genetic algorithms and simulated annealing) to infer 3-D structure from IF. Zhu *et al.* [93] proposed a manifold-based framework called GEM, which first uses IF to create an interaction network representing the spatial organization of the *loci* from Hi-C data. Zhu *et al.*'s aim was to use a manifold learning algorithm to uncover the low-dimensional (3-D) geometry embedded in a high-dimensional (Hi-C) space, while satisfying certain defined conformation energy requirements. An improvement over this method integrates Hi-C data with FISH data for 3-D structure inference [94]. To ensure the modeling of realistic structures consistent with cellular organization, Paulsen *et al.* [91] introduced Chrom3-D, a genome-modeling algorithm that

combines Hi-C and Lamina-associated domain (LAD) information from ChIP-seq data to generate an ensemble of 3-D genome structures in which *loci* and TAD positioning and interaction requirements are satisfied.

On the other hand, certain methods convert contact frequency into defined spatial restraints. As is the case with distance-based approaches, these restraints are satisfied through an optimization method. In their seminal study, Kalhor *et al.* [68] developed a 3C variant known as tethered conformation capture (TCC), aimed at increasing the signal-to-noise ratios in conformation capture experiments. This is relevant because it allows for a more accurate representation of IF, especially for genome structure analysis, where low inter-chromosomal interactions are recorded using existing approaches. Using TCC data, researchers proposed a novel modeling approach whereby a variety of genome structures were generated. This approach, called population-based modeling, produces a population of structures representative of genomic configuration and consistent with contact probability. Serra *et al.* [85] followed certain constraints in order to transform IF into spatial restraints; for instance, consecutive and non-consecutive *loci* were treated differently. As in the case of Bau *et al.* [72], these restraints were satisfied by using the IMP.

1.4.3 Probability Based Methods

Methods in this category define a probabilistic measure for contact frequency, hence their name. Using a probabilistic approach to model 3-D structures has a number of advantages; key among them is that such an approach allows uncertainties in experimental Hi-C data to be easily considered through probabilistic representation. In addition, statistical calculations of specific structural properties or noise sources can be carried out. Due to the fact that Hi-C data are drawn from cell populations, IF can be considered as an average; most probability-based methods assume that an ensemble of structures underlies a contact map. In addition, they consider the problem of

3-D structure inference as either a Bayesian inference problem or a maximum likelihood problem. However, some probabilistic modeling may be more time consuming than other methods.

Rousseau *et al.* [48] developed the first method in this category, called MCMC5C. They defined a probabilistic model of IF and used a Markov chain Monte Carlo (MCMC) sampling to generate an ensemble of structures. MCMC5C through a Gaussian model based on Hi-C data, whose variance was estimated using an improvised approach. A MCMC sampling-based algorithm was selected over alternatives methods because of its inherent ability to estimate the distribution of various structural properties. As previously mentioned, raw Hi-C data contain a number of systematic biases such as GC content, restriction enzyme cutting frequency, and sequence uniqueness [104]. These factors all need to be considered when designing a 3-D genome reconstruction method. To overcome these limitations, Hu *et al.* [75] proposed two Bayesian models for 3-D genome structure reconstruction from Hi-C data. Their methods combined bias removal with 3-D genome structure construction. They corrected known biases and used a Poisson model to fit contact data, an improvement over MCMC5C when it came to estimating the Gaussian variance. Varoquaux *et al.* [78] also defined a probabilistic model of IF. Similar to the model defined by Hu *et al.*, it defined the structure inference problem as a maximum likelihood problem and used an optimization method to solve it.

A typical drawback of high-resolution Hi-C data is the sparsity of long-range contacts on the contact matrix and the high proportion of zero-contact counts between *loci* in the matrix. Hence, certain existing methods might be incapable of modeling at a higher resolution. Park and Lin [87] proposed an algorithm that is robust to resolution specification and corrects known systematic biases. They modeled the contact count using a Poisson distribution and addressed excess zero

problems in high resolution datasets. They suggested that these problems could be solved by adjusting the Poisson distribution adopted for modeling.

Nagano *et al.* and Stevens *et al.* [34, 120, 121] applied a simulated annealing technique to sample single-cell datasets, while sometimes using contacts as distance restraints at different data resolutions. A novel study by Tjong *et al.* [86] has proposed a population-based modelling approach called PGS. Different from the ensemble-based approach—where a variety of structures with different variabilities are generated to simulate the heterogeneity of cells in the Hi-C experiment—the population of genome structures generated by PGS is consistent with the normalized contact probability matrix. Tjong *et al.* have formulated a probabilistic framework that uses an EM algorithm with constraint assignment at the E step and optimization of the structure population through simulated annealing and conjugate gradient descent at the M step. This method takes advantage of other external experimental data, such as lamina information for improved modeling. Rosenthal *et al.* [95] proposed an approach to recover missing contacts in single-cell Hi-C contact maps by filling missing parts with structures obtained from the corresponding cell populations, while imposing certain penalties on the generated structures.

1.5 Correcting Biases in Hi-C Data by Data Normalization

As is the case for most sequencing experiments, raw Hi-C data contain several systematic biases that could potentially affect the 3-D genome reconstruction. An inexhaustive list of these systematic biases include GC content, distance between restriction sites, restriction enzyme cutting frequency, sequence uniqueness, and experimental artifacts [104]. In a Hi-C experiment protocol, a minimum of 25 million cells was used to produce a Hi-C library [27, 30, 38, 69] with the goal of analyzing the contact frequencies between genomic sites in a cell population. One of the reasons

for using a population of cells in Hi-C experiments is more sequence reads can be produced from a population of cells than a single cell.

The number of paired-end reads linking two genomic regions is interpreted as the interaction frequency between two genomic regions. This implies that a higher interaction frequency on a contact map means that a higher read count was observed, and that the two regions are spatially close to each other. However, many of these systematic biases affect the observed Hi-C read counts for two interacting regions (or fragments) on a contact matrix [106]. Hence, when these biases are left unhandled, the 3-D model construction is predicated on inaccurate information and consequently may be adversely affected. Additionally, if the effect of duplication, deletion, inversion and ploidy is significant in the pair reads, this could cause a direct effect on the number of paired-end reads linking two genomic regions which will alter the derived contact map. Because the Hi-C contact data is used for 3-D genome modeling, the level of correctness of the Hi-C data largely determines the accuracy of the generated model.

To overcome these limitations, most 3-D reconstruction methods apply normalization methods that focus on removing biases introduced by experimental procedures and by intrinsic properties of the genome to preprocess the data [9, 42, 104–108]. With the application of a normalization and pre-processing technique before 3-D genome reconstruction, the noise and systematic biases introduced by external factors, such as DNA shearing, and cutting, during the Hi-C experiment makes the Hi-C data more suitable for chromosome/ genome 3-D structure reconstruction. Alternatively, some probability-based reconstruction methods handle the noise and biases differently by taking the biases into consideration in their algorithm design [75].

A common problem observed in some Hi-C data is the omission of the contact frequency of some genomic positions in the contact matrix. When this occurs, the reconstructed 3-D model from this

data varies across the different tools due to difference in the way the methods represent omissions in their 3-D model. Generally, this leaves some doubt about which 3-D model is better when this occurs.

1.6 Validation and Evaluation

According to the literature on chromosome and genome 3-D construction methods, algorithms are most often validated by a simulated dataset to assess their reconstruction ability, the consistency with the Hi-C data, known genome and chromosome structural features [49], or Fluorescence in situ hybridization (FISH) data. In the simulation case, most methods use a 3-D polymer model meant to serve as a gold standard model with which to compare the final 3-D reconstructed structure. A set of chromosomal contact data is then simulated from this structure, and a certain degree of Gaussian noise is often added to the data as well. The noise is usually added to assess the methods' responsiveness and accuracy to noisy data. Eventually, the algorithms' ability to reconstruct the true model is tested. A commonly used synthetic dataset is the one generated by Trussart *et al.* [122]. Trussart *et al.* created a series of simulated Hi-C contact matrices in which genomic architectures are pre-defined, and the noise level and structural variability (SV) are both simulated.

FISH provides a powerful tool for identifying the location of a DNA sequence. It is used to study the 3-D organization of chromosomes and genomes and determine the proximity of a gene relative to other genes through the use of fluorescent probes [123]. It has been determined to be much more accurate, simple, and reliable than all other molecular profiling techniques [124]. Hence, it is often used to determine the distance between *loci* in a genome and for single-cell analysis of gene and *loci* positioning [125–128]. However, its major limitations are low throughput and resolution at higher scales, such as the entire genome or an ensemble of cells. Nonetheless, FISH data can be

used to validate the distance between *loci* in a reconstructed 3-D structure at a lower scale. Given that the FISH method is considered reliable, it is useful in the study of chromosomal and genomic 3-D spatial organization when *loci* in the structure being evaluated are physically proximal.

Once the structure construction is complete, a method is often needed to assess its accuracy. The most common approach to structure evaluation is to calculate the Pearson Correlation Coefficient (PCC), the Spearman Correlation Coefficient (SCC), or the Root Mean Square Error (RMSE) of the distance representation of the Hi-C data and the Euclidean distance of the 3-D chromosomal structure. Since these metrics are obtained for distance, they are sometimes referred to as the distance Pearson Correlation Coefficient (dPCC), the distance Spearman Correlation Coefficient (dSCC), and the distance Root Mean Square Error (dRMSE). The value of dSCC and dPCC is in the range of − 1 to + 1, with higher values being preferable. In the case of dRMSE, on the other hand, a lower value is preferred. The latter may vary between 0—which signifies no difference between distances—and a large upper limit dependent on the number of fragments in the structure being compared when they are completely different. The dRMSE is also an appropriate metric to assess the similarity between 3-D structures. In order to do so, a linear transformation that includes translation, orthogonal rotation, and rescaling is performed on one of the structures, so that they are at the same 3-D-coordinate scale as in [49].

Let the pairwise distance between Hi-C data IF be represented by the vector $\{D_i, \ldots, D_n\}$ and the Euclidean distance between *loci* in a 3-D chromosome model be represented as $\{ED_i, \ldots, ED_n\}$, where n is the number of *loci* pairwise distances. The dSCC, dPCC, and dRMSE can be computed as shown below:

(1) The dPCC is defined as:

$$\text{dPCC} = \frac{\sum_{i=1}^{n}(D_i - \acute{D})(ED_i - \acute{ED})}{\sqrt{\sum_{i=1}^{n}(D_i - \acute{D})^2 \sum_{i=1}^{n}(ED_i - \acute{ED})^2}}$$

where:

- D_i and ED_i are single distance samples indexed with i,
- n is the number of *loci* pairwise distances,
- $\acute{D}$ and $\acute{ED}$represent sample means. $\acute{D} = \frac{1}{n}\sum_{i=1}^{n} D_i$, $\acute{ED} = \frac{1}{n}\sum_{i=1}^{n} ED_i$.

(2) The dSCC is defined as:

$$\text{dSCC} = \frac{\sum_{i=1}^{n}(A_i - \acute{A})(B_i - \acute{B})}{\sqrt{\sum_{i=1}^{n}(A_i - \acute{A})^2 \sum_{i=1}^{n}(B_i - \acute{B})^2}}$$

dSCC is calculated by converting distance variable D_i and ED_iinto ranked variables A_i and B_{ii}, and then, computing the dPCC between the ranked variables. Hence, the pairwise distances D_i and ED_i are converted into ranked variables A_i and B_i respectively,

where:

- A_i and B_i are the ranks of two distances, D_i and ED_i respectively.
- $\acute{A}$ and $\acute{B}$ represent sample means of rank. $\acute{A} = \frac{1}{n}\sum_{i=1}^{n} A_i$, $\acute{B} = \frac{1}{n}\sum_{i=1}^{n} B_i$.

(3) The dRMSE is defined as:

$$\text{dRMSE} = \sqrt{\frac{1}{n}\sum\left(D_{ij} - ED_{ij}\right)^2}$$

- where D_{ij} and ED_{ij} represent the pairwise distance between *loci* i and j of the Hi-C IF data and 3-D structure Euclidean distance
- n is the number of *loci* pairwise distances.

1.7 Outline

The content of each chapter in this dissertation is described as follow. Chapter 1, the introduction, gives the general background about Hi-C experiment, the data obtained from this protocol, the classification of the methods for 3D structure reconstruction using this data, and the approach for evaluating these methods. The main content of this chapter is from the following publication:

***Oluwadare, O**, Highsmith M, Cheng J. An Overview of Methods for Reconstructing 3-D Chromosome and Genome Structures from Hi-C Data. **Biological Procedures Online**. 2019 Dec;21(1):7. [147]*

Chapter 2 describes a tool for chromosome and genome 3D structure prediction from Hi-C data using an optimization algorithm, 3DMax [70]. The main content of this chapter is from the following publication:

***Oluwadare O**, Zhang Y, Cheng J. A maximum likelihood algorithm for reconstructing 3D structures of human chromosomes from chromosomal contact data. **BMC genomics**. 2018 Dec;19(1):161. [70]*

Chapter 3 describes, GSDB, a comprehensive and common repository that contains 3D structures for Hi-C datasets from novel 3D structure reconstruction tools developed over the years. The main content of this chapter is from the unpublished manuscript:

***Oluwadare O**, Max Highsmith, Cheng J. a database of 3D chromosome and genome structures reconstructed from Hi-C data. (To be submitted July/August 2019)*

Chapter 4 describes, ClusterTAD [177], a method for topological associated domains (TAD) extraction from Hi-C data using unsupervised learning algorithm. The main content of this chapter is from the following publication:

***Oluwadare O**, Cheng J. ClusterTAD: an unsupervised machine learning approach to detecting topologically associated domains of chromosomes from Hi-C data. **BMC bioinformatics**. 2017 Dec;18(1):480. [177]*

Chapter 5 describes GenomeFlow [40], a comprehensive graphical tool to facilitate the entire process of modeling and analysis of 3D genome organization. This tool provides a graphical user interface to analyze and study Hi-C data in 2D and 3D format, it's the first tool to provide this in the genome research field. The main content of this chapter is from the following publication:

Trieu T, **Oluwadare, O***, Wopata J, Cheng J. GenomeFlow: a comprehensive graphical tool for modeling and analyzing 3D genome structure. **Bioinformatics.** 2018 Sep 12.* **(* co-first author***) [40]*.

Chapter 6 describes the installation steps and the instructions for using the software tools developed for each of the methods above.

2 A maximum likelihood algorithm for reconstructing 3D structures of human chromosomes from chromosomal contact data

2.1 Abstract

The development of chromosomal conformation capture techniques, particularly, the Hi-C technique, has made the analysis and study of the spatial conformation of a genome an important topic in bioinformatics and computational biology. Aided by high-throughput next generation sequencing techniques, the Hi-C technique can generate genome-wide, large-scale intra- and inter-chromosomal interaction data capable of describing in detail the spatial interactions within a genome. These data can be used to reconstruct 3D structures of chromosomes that can be used to study DNA replication, gene regulation, genome interaction, genome folding, and genome function.

Here, we introduce a maximum likelihood algorithm called 3DMax to construct the 3D structure of a chromosome from Hi-C data. 3DMax employs a maximum likelihood approach to infer the 3D structures of a chromosome, while automatically re-estimating the conversion factor (α) for converting Interaction Frequency (IF) to distance. Our results show that the models generated by 3DMax from a simulated Hi-C dataset match the true models better than most of the existing methods. 3DMax is more robust to structural variability and noise. Compared on a real Hi-C dataset, 3DMax constructs chromosomal models that fit the data better than most methods, and it is faster than all other methods. The models reconstructed by 3DMax were consistent with fluorescent in situ hybridization (FISH) experiments and existing knowledge about the organization of human chromosomes, such as chromosome compartmentalization.

3DMax is an effective approach to reconstructing 3D chromosomal models. The results, and the models generated for the simulated and real Hi-C datasets are available here: http://sysbio.rnet.missouri.edu/bdm_download/3DMax/. The source code is available here: https://github.com/BDM-Lab/3DMax . A short video demonstrating how to use 3DMax can be found here: https://youtu.be/ehQUFWoHwfo .

2.2 Background

A set of all chromosomes within the nucleus of a eukaryotic cell constitutes its genome. Studies of the organization of chromosomes and genomes reveal that they are structurally organized within a cell [1-12]. Studies find that this organization influences many biological mechanisms such as DNA replication, DNA repair, DNA translocation, gene regulation, transcription efficiency, genome interpretation, epigenetic modification, and genome stability maintenance [1, 12]. The Fluorescent In situ Hybridization (FISH) [18.19] was often used in the investigation of the three-dimensional (3D) organization of a genome, but it cannot produce the layout of the genome structure at a large scale. The chromosome conformation capturing techniques such as 3C [24], 4C [28], 5C [29], and Hi-C [30] were developed to analyze the spatial organization of chromatin in a cell at a larger scale. The Hi-C technique can use next generation DNA sequencing to determine genome-wide spatial chromosomal interactions.

Much progress has been made in recent years on the study of chromosome and genome 3D structure modeling. Several methods have been proposed to construct the structure of an individual chromosome or an entire genome from chromosome conformation capturing data [66-70,72-81,83-98,109-110]. Some of these methods perform chromosome/genome 3D structure modeling in a two-step process, which involves converting interaction frequencies (IF) between fragment pairs in Hi-C data to distances between them, and then inferring the 3D structures that best satisfies

the distances. Methods that implement this two-step process are known as distance restraint-based methods. Several of such methods have been proposed, each of which varies in restraint representation and optimization methods adopted [83-98].

In [66], Duan *et al.* considered the genome 3D structure prediction problem as a constrained non-convex optimization problem, and hence used an optimization solver (open-source software) IPOPT [71] to solve it. Bau *et al.* [110] also treated the 3D modeling problem as an optimization problem, and used the Integrated Modeling Platform (IMP) [73] to construct 3D structure models. The MCMC5C [48] method designed a probabilistic model for the interaction frequency (contact) data, and thereafter used a Markov Chain Monte Carlo (MCMC) approach to generate a representative structure from the data. ChromSDE [76] formulated the 3D structure modeling problem as a non-convex non-linear optimization problem, but then relaxed it as a semi-definite programming (SDP) problem. Bayesian 3D constructor (BACH) [75] is another method that employs MCMC to infer the 3D structure by maximizing the likelihood of the observed Hi-C data following a Poisson regression approach. MOGEN [49,80] is a contact-based method that is different from the rest, because it does not require the conversion of interaction frequencies to distances before structure construction. ShRec3D [79] is a two-step algorithm that uses the shortest path algorithm to realize chromosome structure construction. LorDG [69] uses a Lorentzian objective function to construct the 3D model of a chromosome or genome.

Despite the significant progress made over the years, some of the distance-based chromosome structure modeling methods have several limitations: they may simply assume that the parameters used for converting interaction frequencies to distances are independent of input data and therefore are fixed for different datasets [80, 131], they may converge slowly (common for Markov chain

Monte Carlo (MCMC) approach [132-133]), and they sometimes require to adjust quite a few parameters [49,80], making it difficult to use.

In this paper, we introduce a new method called 3DMax that uses a maximum likelihood approach to infer the 3D structures of a chromosome from Hi-C data. In the 3DMax algorithm, the conversion factor (α) parameter to convert IF to its distance equivalent is determined automatically from the data. We show that 3DMax is relatively faster than most of the existing methods, and it only depends on optimizing the structural coordinate of predicted models through the least square residuals. 3DMax is capable of translating contact data of a chromosome, or genome into an ensemble of probable 3D conformations to approximate the dynamic 3D genome structures of a population of cells of the same type. Our experiment also demonstrates how parameters such as the learning rate and the convergence constant (epsilon) can impact the performance of a constructed model. We also demonstrate the effect of using different normalization method on the different chromosome 3D structure prediction algorithms. We benchmarked 3DMax with several popular methods [48, 75, 80, 110, 122], and the result showed that our method performed robustly in the presence of noise and structural variability. We applied our method to a synthetic chromosomal interaction dataset, and two experimentally generated Hi-C datasets: a karyotypically normal human lymphoblastic cell line (GM06990) and a malignant B-cell. We used the data from FISH experiments available for the cell lines as independent validations of the reconstructed 3D chromatin structures. We performed a comparative analysis of the performance of 3DMax and several existing 3D reconstruction methods on the Hi-C datasets normalized by three commonly used methods [9,104-105]. These experiments show that 3DMax is an effective method for reconstructing 3D chromosomal structures from Hi-C data.

2.3 Methods

Generally, before Hi-C data [30] are used for model construction, they are converted to a matrix form known as a contact matrix or a contact map.

2.3.1 Chromosome contact map

A chromosome contact map is a N * N matrix, extracted from a Hi-C data, showing the number of interactions between chromosomal regions. The size of the matrix (N) is the number of equal-size regions of a chromosome. The length of equal-size regions (*e.g.* 1Mb base pair) is called resolution. Each entry in the matrix contains a count of read pairs that connect two corresponding chromosome regions in a Hi-C experiment. Therefore, the chromosome contact matrix represents all the observed interactions between the regions (or bins) in a chromosome. The 3DMax algorithm takes as input a contact map to build the 3D structure of a chromosome.

2.3.2 Structure initialization

To structurally represent a chromosome, each of its regions (or bins) is represented by three coordinates (x, y, z) in 3D space. In 3DMax, the structure construction starts with a random initialization of the coordinates of all the regions such that they are in the range [-0.5, 0.5] as in [80].

2.3.3 Maximum likelihood objective function of a chromosome structure

We used a log likelihood function as an objective function to compute chromosome structures from a contact map. Let S stand for a 3D chromosome structure, and D represent the contact matrix data derived from a Hi-C dataset. The likelihood of S, P(D|S), can be expressed as the product of the probabilities of individual data points (interaction frequencies or distances) in D conditioned on the structure S, if the data points are conditionally independent of each other given a S. In 3DMax structure modeling, the input contact matrix is converted to spatial distances based on the

assumption that the IF and the distance have an inverse relationship [48,75,76,78,122]. The conversion method is explained in the Subsection "*conversion of interaction frequency to spatial distance"* later. By assuming that data points D_i in D are conditionally independent given a structure S, we defined the likelihood (L(S)) in Equation (1) as:

$$L(S) = P(D|S) = \prod_{i=1}^{n} P(D_i|S) \quad (1)$$

, where n represents the total number of data points to be considered, and D_i represents the i^{th} data point (*i.e.*, the distance between a pair of chromosomal regions derived from the contact matrix). Assumed that each data point i obeys the normal distribution, the probability of data point D_i can be described as:

$$P(D_i|S) \sim \frac{1}{\sigma\sqrt{2\pi}} . \exp\left(-\frac{1}{2\sigma^2}(D_i{}^s - D_i)^2\right) \quad (2)$$

, where $D_i{}^s$ which is the actual Euclidean distance of the pair of regions corresponding to D_i, computed from (*x*,*y*,*z*) coordinates of the two regions in 3D structure S as in [34]. σ^2 is the variance of the distance. By combining Equations (1) and (2), we obtain the likelihood estimate of a structure S:

$$L(S) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n . \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(D_i{}^s - D_i)^2\right) \quad (3)$$

By taking the logarithm of both sides of the Equation (3), we obtain the log likelihood objective function in Equation (4) for 3DMax chromosome structure reconstruction. Our goal is to find a structure S* that maximizes the likelihood function: L(S|D).

$$L\,(S) = -\frac{\sum_{i=1}^{n}(D_i{}^{s}-D_i)^2}{2\sigma^2} - n.log\sigma \qquad (4)$$

With the assumption that the data is normally distributed according to Equation (2), σ is calculated as in Equation (5):

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(Di^{s}-Di)^2}{n}} \qquad (5)$$

We eliminated the dependence of the objective function on σ parameter by plugging Equation (5) into the log likelihood objective function in Equation (4). Hence, the resulting objective function L(S) can be represented as in Equation (6). The objective function in Equation (6) depends only on the (*x,y,z*) coordinates of regions in the structure.

$$L\,(S) = = -\frac{n}{2} - \text{nlog}\sqrt{\frac{\sum_{i=1}^{n}(Di^{s}-Di)^2}{n}} \qquad (6)$$

2.3.4 Gradient ascent optimization Algorithm

We used the gradient ascent method to optimize the objective function iteratively until the 3DMax algorithm converges. 3DMax algorithm is considered converged, if the difference between the newly calculated log likelihood L(S) function value obtained with updated (x, y, z) coordinates and old L(S) function value of the previous step is less than a small constant value (epsilon). The determination of the epsilon value is described in the Results section.

Gradient ascent is an iterative optimization algorithm that moves in the direction of the function gradient. Using Equation (6) as the base equation, we calculated the partial derivative of the log likelihood function with respect to a region's x, y, and z coordinates in a 3D structure.

Once the partial derivative for each coordinate was obtained, we used the gradient ascent optimization method to adjust each coordinate to get a new structure S* that increases the likelihood. Equation (7) shows how the update was done, where λ is the learning rate, and S is the (x, y, z) coordinate vector in 3D space. If the learning rate is too small, it can result in a slow convergence to an optimal solution. But, if a larger learning rate is defined, the algorithm might oscillate around an optimal solution. There is no standard approach to choose λ value, but it is common to set a larger learning rate at the beginning of the optimization, and reduce it as the optimization progresses. The result of using the different types of learning rate is described in the Subsection "*choice of the learning rate*" in the Result section.

$$S^{(t+1)} = S^{(t)} + \lambda^{(t)} \nabla L(S^{(t)}) \qquad (7)$$

, where t is an iteration index, $S^{(t)}$ is the structure coordinate at an iteration index t, $\lambda^{(t)}$ is a learning rate at t that may vary as the iteration proceeds, and $\nabla L(S^{(t)})$ is the partial derivative of the log likelihood with respect to the coordinates in the structure.

In this work, we also implemented a variant of the 3DMax algorithm above, called 3DMax1, which performs an extra pre-processing and filtering of the input contact matrix when the input is noisy (*e.g.* having low IFs). Moreover, 3DMax1 uses a stochastic gradient ascent algorithm with per-parameter learning rate, which is called the adaptive Gradient algorithm (AdaGrad). The AdaGrad [113] is a gradient-based optimization that can adapt the learning rate to each parameter, it performs larger updates for infrequent or sparse parameters and smaller updates for frequent or less sparse parameters. And it often improves convergence performance over standard stochastic gradient ascent when dealing with sparse parameters [134]. Different from 3DMax that updates the values of all the structure parameters in S at once with the same learning rate λ, AdaGrad in

3DMax1 uses a different learning rate for every parameter in S at every time t . Let Equation (8) represent the gradient of the log likelihood for a parameter S_i at a time step t. Hence, the stochastic Gradient ascent in Equation (7) can be written as in Equation (9) for a parameter S_i in S.

$$g_{t,i} = \nabla L(S_i^{(t)}) \quad (8)$$

$$S_i^{(t+1)} = S_i^{(t)} + \lambda^{(t)} . g_{t,i} \quad (9)$$

In the update rule for AdaGrad, it modifies the learning rate λ at each time step for every parameter S_i based on the previously computed gradient for the parameter S_i. according to Equation (10)

$$S_i^{(t+1)} = S_i^{(t)} + \frac{\lambda}{\sqrt{G_{t,ii} + \varepsilon}} . g_{t,i} \quad (10)$$

Here, G_t is a diagonal matrix where each diagonal element i, i is the sum of the squares of the gradients w.r.t. S_i up to time step t according to Equation (11). While ε is a smoothing term that avoids division by zero (usually on the order of 1e−6).

$$G_t = \sum_{i=1}^{t} (g_i g_i) \quad (11)$$

In essence, G_t contains the sum of the squares of the past gradients for all the parameters in S along its diagonal. One of AdaGrad's main benefits is that it eliminates the need to manually tune the learning rate at each iteration.

2.3.5 Normalization of Hi-C data

Data normalization is necessary for Hi-C datasets, because there is a lot of noise in them. In this study, we used the Iterative Correction and Eigenvector decomposition (ICE) technique [105] as

the default technique to normalize the Hi-C data. The ICE technique was used to normalize the contact map derived from both the synthetic data and the experimental Hi-C data. The GM06990 Hi-C data was also normalized using the Yaffe and Tanay normalization technique [104]. The Yaffe and Tanay normalization technique normalizes the observed read counts by the expected read counts between the regions in a contact matrix. The other technique used to normalize the GM06990 Hi-C data is the Sequential Component Normalization (SCN) technique [9]. The results obtained by the three methods above are presented in the Results Section.

2.3.6 Conversion of interaction frequency to spatial distance

An important aspect of most distance restraint-based modeling approaches including 3DMax is to convert the interaction frequency (IF_{ij}) between two regions (i, j) in a contact matrix to a hypothetical Euclidean distance. An inverse relationship is assumed to exist between them. The relationship is usually defined as $1/IF^{\alpha}$, where IF is the interaction frequency, and α is called the conversion factor. According to [76], α cannot be too small because the spatial distance becomes independent of the interaction frequency as α approaches zero. And α also cannot be too large because in this situation a small change in interaction frequency(IF) could produce a significant difference in the spatial distances. Therefore, choosing a conversion factor that correctly represents the relationship between distance and interaction frequency is important. For 3DMax, we assume that the optimal α will be in the range [0.1, 2], which is consistent with the previous study [48] [76].

2.3.7 Measurement of model similarity and accuracy

We used the Pearson Correlation Coefficient (PCC), the Spearman's Correlation Coefficient (SCC), and the Root Mean Square Error (RMSE) to measure the similarities between chromosomal structures, and assess the accuracy of the constructed structures as in the previous studies

[48,64,66,75,76,78,80,110,122]. When these assessment methods are applied on a distance representation of a model, or a distance representation of Hi-C data, they are sometimes called the distance Pearson Correlation Coefficient (dPCC), the distance Spearman Correlation Coefficient (dSCC), and the distance Root Mean Square error (dRMSE), respectively. For instance, if we have two pairwise distance dataset from two models, $\{d_i, \ldots, d_n\}$ containing n values, and another pairwise distance dataset $\{D_i, \ldots, D_n\}$ containing n values, the dPCC, the dSCC and the dRMSE can be computed using the formulas given below.

(1) The distance Pearson Correlation Coefficient (dPCC) is defined as,

$$\text{dPCC} = \frac{\sum_{i=1}^{n}(d_i-\bar{d})(D_i-\bar{D})}{\sqrt{\sum_{i=1}^{n}(d_i-\bar{d})^2 \ \sum_{i=1}^{n}(D_i-\bar{D})^2}}$$

where:

- d_i and D_i are single distance samples indexed with i,
- n is the number of pairwise distance.
- $\bar{d}$ and $\bar{D}$ represent sample means. $\bar{d}\ \frac{1}{n}\sum_{i=1}^{n} d_i$, $\bar{D}= \frac{1}{n}\sum_{i=1}^{n} D_i$.

(2) The distance Spearman's Correlation Coefficient (dSCC) is defined as

$$\text{dSCC} = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2 \ \sum_{i=1}^{n}(Y_i-\bar{Y})^2}}$$

dSCC is calculated by converting distance variable d_i and D_i into ranked variables X_i and Y_i, and then, computing the dPCC between the ranked variables.

where:

- X_i and Y_i is the rank of two distance d_i and D_i respectively. Hence, X and Y is a vector of distance rank of the distance vector *d* and *D respectively*.
- $\bar{X}$ and $\bar{Y}$ represent sample means of rank. $\bar{X}$ $\frac{1}{n}\sum_{i=1}^{n} X_i$, $\bar{Y}$= $\frac{1}{n}\sum_{i=1}^{n} Y_i$.

(3) The distance Root Mean Square Error (dRMSE) is defined as,

dRMSE = $\sqrt{\frac{1}{n}\sum(d_{ij} - D_{ij})^2}$

- where d_{ij} and D_{ij} are the distance vector between regions i and j for the first model, and second model respectively.
- n is the number of pairwise distance.

The dSCC measures the similarity of the distance profiles of two 3D structures. The dSCC value varies between -1.0 and 1.0; the higher the dSCC value is, the more similar the two structures are. It is worth noting that, to determine the dRMSE of two structures, the structures must be compared at the same scale. For instance, assuming two structures are represented with coordinates S**'** and S $\in$ $R^{n \times 3}$, where S**'** is the model constructed by 3DMax, S is the known model from a simulated data, and n is the number of regions representing a chromosome. To calculate the dRMSE value, we performed linear transformations that includes translation, orthogonal rotation, and rescaling of the points in the matrix $R^{3 \times n}$ of structure S**'** in order to best match them with the points in matrix $R^{3 \times n}$ of structure S. The Procrustes function library defined in MATLAB [135-138] is used to do the transformation of the dimensions. After the transformation, the dRMSE value between the scaled structure S'' and the original structure S is calculated.

2.3.8 Datasets

The synthetic dataset from Trussart *et al.*, 2015 [122] is a series of simulated Hi-C contact matrices where the genomic architectures are pre-defined and the noise level and structural variability (SV) are both simulated. The contact maps, the original models and their reconstructed models used in this study were downloaded from http://sgt.cnag.cat/3dg/datasets/.

The real Hi-C data used in this study is from a normal GM06990 cell line and a malignant B-cell line. The normal GM06990 dataset was downloaded from the Gene Expression Omnibus (GEO) repository under the accession number GSE18199. Its raw and normalized interaction frequency matrices at 1-MB resolution [30] were downloaded from [139]. We used the normalization pipeline described in [9] and [105] to obtain normalized contact matrices. The raw contact matrices of the malignant B-cell 1-MB resolution were obtained from [140]. We used the pipeline [105] to normalize them. The Fluorescence In-Situ Hybridization (FISH) data of the GM06990 cell line is from [30]. Its FISH distances and contact maps were obtained from [47].

2.4 Results

We evaluated our method using a synthetic dataset (Trussart *et al.*, 2015) [122] and two real Hi-C datasets of the two cell lines: a karyotypically normal human lymphoblastic cell line (GM06990) [30] and the malignant B-cell of an acute lymphoblastic leukemia patient [140].

2.4.1 Parameter estimation

To use 3DMax, the conversion faction (α) needs to be defined. As the default, we set the α value to be in the range [0.1, 2] as explained in the Methods section. Another parameter we defined in 3DMax is the convergence constant called epsilon. To estimate the best epsilon value to use, we experimented on the GM06990_HindIII cell line dataset using six epsilon values, *i.e.*, 1, 0.5, 0.1, 0.01, 0.0001, and 0.00001(Table 1.1). According to our experiment, although the different epsilons

produced comparable dSCC average, the epsilon = 0.0001 has the highest average dSCC score. Hence, we set it as the default epsilon value for 3DMax. The number of ensemble structures (N) to generate per conversion factor is another parameter to be tuned. Table 2.2 shows the performance changes by setting different numbers of ensemble structures (NUM_STR). It is observed that a higher N value does not guarantee a significant increase in the accuracy. We set the default N to 5 in our implementation.

Table 2.1. The determination of the convergence constant (epsilon) values for the 3DMax algorithm. The dSCC value between the input distance matrix and the representative model for chromosome 1 – 22 of the GM06990 cell line using convergence constant (epsilon): 1, 0.5, 0.1, 0.01, 0.0001, and 0.00001 respectively. The average dSCC values across the chromosomes show that the results are highly comparable. The epsilon = 0.0001 has the highest average dSCC score, hence, we set it as the default epsilon value for 3DMax. The bold text represents the highest dSCC value.

Chromosome	**epsilon=1**	**epsilon=0.5**	**epsilon=0.1**	**epsilon=0.01**	**epsilon=0.0001**	**epsilon=0.00001**
1	0.8087	0.8088	0.8087	0.8087	0.8088	0.8087
2	0.8149	0.8149	0.8149	0.8149	0.8149	0.8149
3	0.8306	0.8306	0.8306	0.8306	0.8306	0.8306
4	0.8716	0.8716	0.8714	0.8663	0.8735	0.8714
5	0.8645	0.8645	0.8645	0.8646	0.8654	0.8645
6	0.8477	0.8479	0.848	0.8478	0.848	0.848
7	0.8302	0.8302	0.83	0.8302	0.831	0.8301
8	0.8701	0.8701	0.8701	0.8702	0.8701	0.8701
9	0.853	0.853	0.8495	0.8521	0.8532	0.8508
10	0.8538	0.8542	0.8541	0.8538	0.8538	0.8538
11	0.8431	0.8431	0.8431	0.8431	0.8433	0.8432
12	0.8576	0.8576	0.8578	0.8577	0.8578	0.8578
13	0.8581	0.8553	0.8582	0.8582	0.8584	0.8582
14	0.8785	0.8796	0.8797	0.8797	0.8797	0.8797
15	0.8593	0.8563	0.8588	0.8595	0.8565	0.8592
16	0.8441	0.8459	0.8458	0.8459	0.8458	0.8458
17	0.8359	0.836	0.8362	0.8362	0.8362	0.8361
18	0.8521	0.8537	0.8536	0.8535	0.8535	0.8534
19	0.8629	0.8669	0.8663	0.8665	0.8665	0.8664
20	0.8853	0.884	0.8842	0.8865	0.8867	0.8867
21	0.9019	0.8995	0.9016	0.9016	0.9017	0.9018
22	0.8657	0.8658	0.8672	0.8658	0.8659	0.8659
Average dSCC	0.8541	0.8541	0.8543	0.8542	**0.8546**	0.8544

Table 2.2. The comparison of the performance when a constant learning rate and a decreasing learning rate are applied. The comparison of the computing time and the average dSCC value obtained by using a constant or a decreasing learning rate for different input parameters for the chromosome 1 – 22 of the GM06990 cell line. We used the constant learning rate 0.0001, and we defined the initial_ $\lambda = 0.01$ for the decreasing learning rate. CHR represents the chromosome number, and NUM_STR represents the number of ensemble structures generated per conversion factor(α), ALPHA represents the conversion factor. The decreasing learning rate achieved a better computing speed in all the cases.

	Constant Learning Rate		**Decreasing Learning Rate**	
Input Parameters	**Running Time**	**Accuracy (Average dSCC)**	**Running Time**	**Accuracy (Average dSCC)**
CHR=1-22, NUM_STR = 1, ALPHA = constant	4 minutes	0.821	13 seconds	0.8493
CHR=1-22, NUM_STR = 1, ALPHA = [0.1, 2]	1 hour, 30 minutes	0.8456	3 minutes	0.8536
CHR= 1-22, NUM_STR = 5, ALPHA = [0.1, 2]	7 hours	0.8546	20 minutes	0.8546

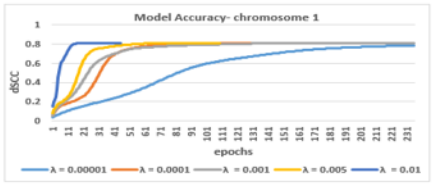
CHR = 1, NUM_STR = 1, ALPHA = constant	37 seconds	0.7556	2 seconds	0.8088
CHR = 1, NUM_STR = 5, ALPHA = [0.1, 2]	1 hour	0.7841	3 minutes	0.8088
CHR = 21, NUM_STR = 1, ALPHA = constant	0.7 second	0.8969	0.2 second	0.8995
CHR = 21, NUM_STR = 5, ALPHA = [0.1, 2]	36 seconds	0.9018	2 seconds	0.9018
CHR = 21, NUM_STR = 30, ALPHA = [0.1, 2]	4 minutes	0.9018	12 seconds	0.9018
CHR = 21, NUM_STR = 50, ALPHA = [0.1, 2]	6 minutes	0.9021	18 seconds	0.9018
CHR = 21, NUM_STR = 100, ALPHA = [0.1, 2]	12 minutes	0.9020	37 seconds	0.9020
CHR = 21, NUM_STR = 200, ALPHA = [0.1, 2]	24 minutes	0.9022	83 seconds	0.9020
CHR = 21, NUM_STR = 500, ALPHA = [0.1, 2]	1 hour	0.9022	3 minutes	0.9021

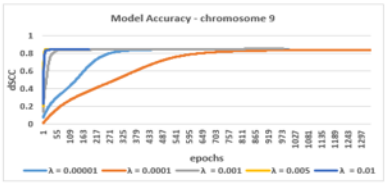
We executed all the other methods following the directions for parameter settings by their authors. All the parameters used to produce all the results are made available in the "parameters" directory of each method in the 3DMax website (http://sysbio.rnet.missouri.edu/bdm_download/3DMax/). For instance, to evaluate the MOGEN program, we used the parameters that produced the best result after trying multiple settings for the parameters required by the algorithm. The different parameters used to generate the MOGEN models, the input data, and the outputs for the three normalization methods for the GM06990 cell line are all available at the 3DMax website.

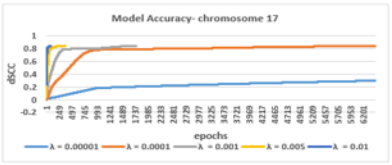
2.4.2 Choice of the learning rate

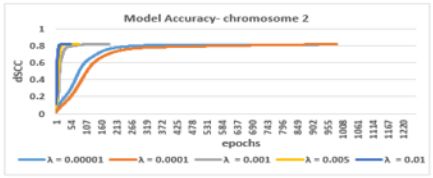
As mentioned in the Methods section, the choice of the best learning rate can sometimes be a difficult task. However, it is common practice to use either a preferable constant learning rate, or a decreasing learning rate.

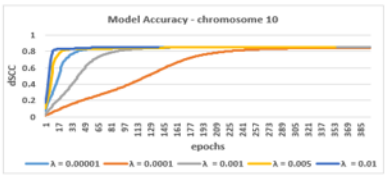
The constant learning rate uses a constant λ value through all the epoch steps for an algorithm. By experimenting with a range of learning rates in our work, Figure 2.1 shows the model accuracy for different constant learning rates for GM06990_HindIII cell chromosome 1 to 22 datasets. The result shows the impact of using the different learning rates for structure modeling. We observed that λ = 0.0001, 0.001, and 0.005 shows a consistent better performance than the other λ values across all the chromosomes. As observed in the Figure, the larger learning rate (λ =0.01) had the advantage of faster convergence in some chromosomes, but suffered fluctuations or even decreased performance at some point (Chromosome 5,11, and 20). The smaller learning rates resulted in slow convergence and sometimes does not converge with a good model accuracy as in the case of λ = 0.00001 (Chromosome 3,11,13,15,16-18, and 21).

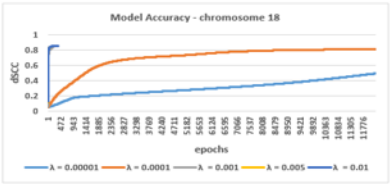

Model Accuracy- chromosome 1
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

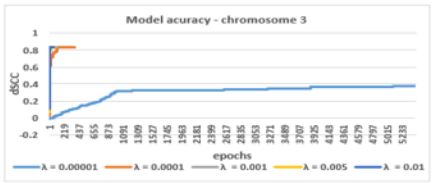

Model Accuracy - chromosome 9
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

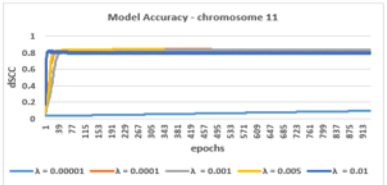

Model Accuracy- chromosome 17
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

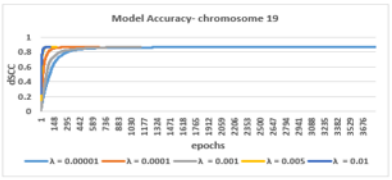

Model Accuracy- chromosome 2
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

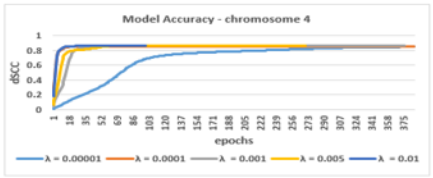

Model Accuracy - chromosome 10
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

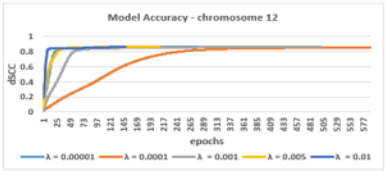

Model Accuracy - chromosome 18
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

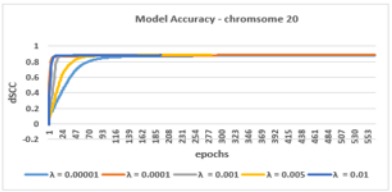

Model acuracy - chromosome 3
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

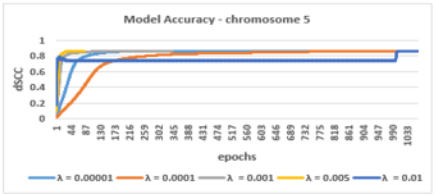

Model Accuracy - chromosome 11
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

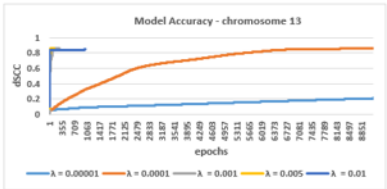

Model Accuracy- chromosome 19
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

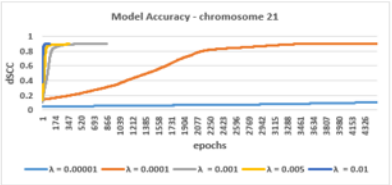

Model Accuracy - chromosome 4
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

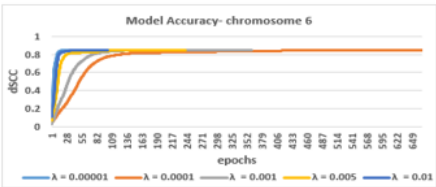

Model Accuracy - chromosome 12
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

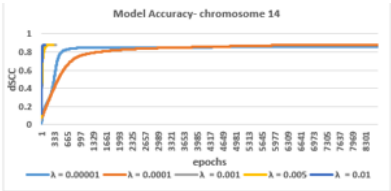

Model Accuracy - chromsome 20
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

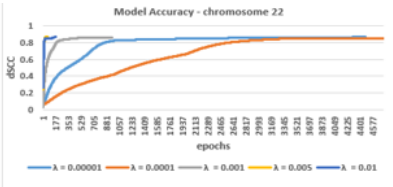

Model Accuracy - chromosome 5
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

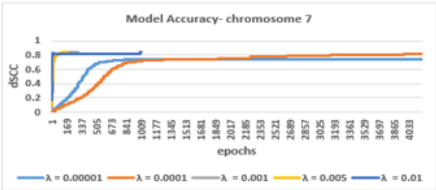

Model Accuracy - chromosome 13
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

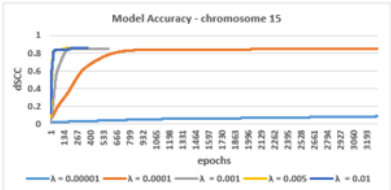

Model Accuracy - chromosome 21
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

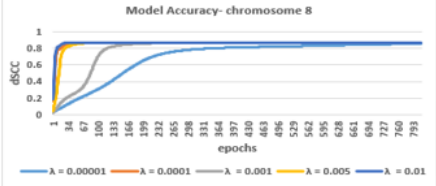

Model Accuracy- chromosome 6
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01


Model Accuracy- chromosome 14
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01


Model Accuracy - chromosome 22
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01


Model Accuracy- chromosome 7
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01


Model Accuracy - chromosome 15
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01


Model Accuracy- chromosome 8
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

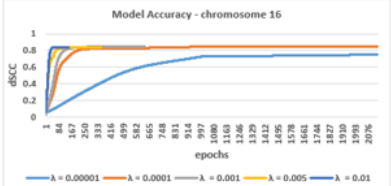

Model Accuracy - chromosome 16
dSCC
epochs
λ = 0.00001
λ = 0.0001
λ = 0.001
λ = 0.005
λ = 0.01

Figure 2.1. The comparison of the step by step model accuracy for different constant learning rate. The comparison of the dSCC model accuracy for five constant learning rates for GM06990_HindIII cell chromosome 1 to 22 dataset. We show the step by step dSCC till convergence for λ = 0.00001,0.0001, 0.001, 0.005 and 0.01 respectively for all the GM06990_cell chromosomes. The result shows that λ = 0.0001,0.001, and 0.005 had less fluctuations, and achieved a higher or similar dSCC value in cell chromosomes. Overall, the performance of 3DMax is comparable for each of the λ values. A higher dSCC value means the better accuracy.

Conversely, for ***the decreasing learning rate***, a typical way to implement it is to choose a starting learning rate and drop the learning rate by half every 70 epochs (in our algorithm). This approach is termed the step-based learning rate decay schedule. It takes the mathematical form below:

$$\lambda = \text{initial\_}\lambda * 0.5^{\frac{1+epoch}{70}}$$

In this work, we compared the result obtained by using the constant learning rate (λ =0.0001), and the decreasing learning rate methods in Figure 2.2. Interestingly, the results show that both methods achieved a comparable accuracy for all the chromosomes. However, in terms of the computing speed, 3DMax is faster when the decreasing learning rate is used than when the constant learning rate is used. The running time and accuracy of the two methods of setting learning rates are reported in Table 2.2. In 3DMax, we made the decreasing learning rate approach the default because it converges faster.

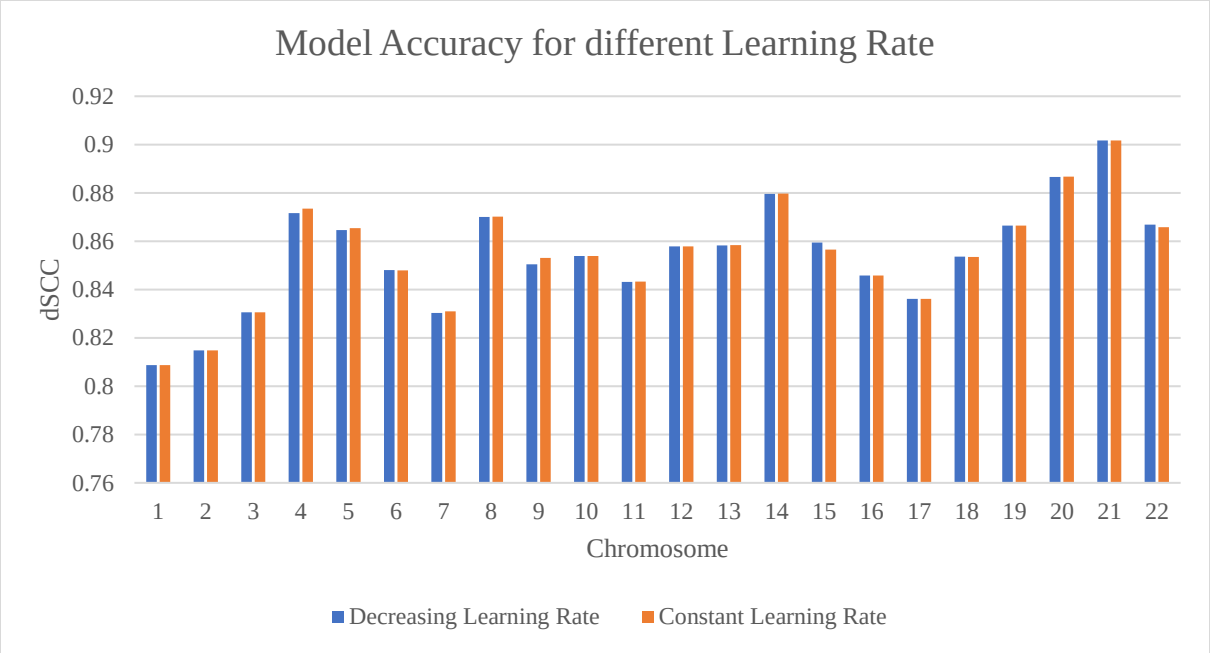


Figure 2.2. The comparison of the performance of 3DMax for constant and decreasing learning rates. Comparison of the result obtained by using the constant learning rate, and the decreasing learning rate shows that both methods achieved a comparable accuracy for all the chromosomes. A higher dSCC value means the better accuracy.

2.4.3 Assessment on simulated datasets

The synthetic dataset includes a series of Hi-C matrices simulated from the pre-defined chromosome structures with different noise levels and structural variability (SV) level. Each worm like chain chromosome structure has ~1 Mb base pairs and is represented by 202 regions of 5 Kb base pairs each. The simulated data can be classified into two categories based on the different architectures of the chromosome structures: Topological Associated Domains (TAD)-like architecture and Non-Topological Associated Domains (Non-TAD)-like architecture [141-143]. Each of these architectures has three structural density levels (40bp/nm, 75bp/nm and 150bp/nm),

resulting in six density-architecture combinations. The entire synthetic dataset contains 168 simulated Hi-C matrices in total, *i.e.*, six different combinations of density and architectures times seven levels of structural variability (SV) (denoted as 0, 1, 2, 3, 4, 5, 6) times four noise levels (*i.e.* 50, 100, 150 and 200). There are 28 simulated Hi-C contact matrices for each of the six density-architecture combinations. According to [48], the most difficult architecture to reconstruct is the 150bp/nm density with no TAD-like features because of its higher resolution and lack of regular TAD sub-structures.

We evaluated 3DMax on the 28 contact matrices (7 levels of structural variability with four noise levels each) of the synthetic dataset with resolution 150bp/nm for both TAD and non-TAD like feature architecture, respectively. The matrices were normalized with the ICE technique before they were used as input for 3DMax. To determine the best conversion factor (α) for model reconstruction, the dSCC value between the distance matrix generated from the input contact matrix and the Euclidean distance of the representative chromosomal model is computed. To determine the representative structure for an input matrix, we generated an ensemble of 50 structures and calculated the similarity between each structure in the ensemble with the input distance matrix. The structure with the highest dSCC value in the ensemble was chosen as the representative structure for the input contact matrix. We then computed the average dSCC value across the 28 contact matrices of the simulated data, with resolution 150bp/nm and TAD like feature architecture, for the conversion factor (α) in the range [0.1, 2] (Table 2.3). The result shows that α value 0.3 has the highest average dSCC value. We computed the average dSCC value between the models reconstructed by 3DMax and the true structures (*i.e.*, a set of 100 true structures for each structural variability level in the simulated dataset) for the α values in the range [0.1, 2] for the simulated data with resolution 150bp/nm and TAD like feature architecture (Table

4). The result also shows that the structures generated at α = 0.3 have the higher similarity to the true structures from simulated dataset than other α values. To compute the accuracy of 3DMax, we compared each structure in the generated ensemble with the true structures (*i.e.*, a set of 100 true structures for each structural variability level) by using the spearman correlation coefficient. We thereafter selected the reconstructed structure closest to a true structure from the ensemble. The spearman correlation coefficient of the selected structure and the true structure was averaged and used as the dSCC accuracy for the ensemble of generated 3DMax structures. The reconstruction accuracy (dSCC) for 3DMax at different levels of noise and structural variability (SV) for α = 0.3 shows that the accuracy of reconstructed models decreased as the structural variability level increased for each noise level (Figure 2.3). The reconstruction accuracy of structures generated by 3DMax is relatively high for different noise levels when the structural variability (SV) is low, while the average accuracy of structures decreases noticeably as the level of SV increases.

Table 2.3. The average dSCC value between the distance matrix and the representative model for 28 contact matrices with different conversion factor (α) values. The average dSCC value between the input distance matrix and the representative model for 28 contact matrices (7 levels of structural variability with four noise levels each) for the conversion factor (α): 0.1, 0.3, 0.5, 1.0, 1.5 and 2.0 respectively. The dataset has resolution 150bp/nm and TAD like feature architecture. The bold text represents the highest dSCC value.

Conversion factor(α)	**0.1**	**0.3**	**0.5**	**1.0**	**1.5**	**2.0**
dSCC	0.759	**0.768**	0.758	0.695	0.638	0.559

Table 2.4. The average dSCC value for the dataset with resolution 150bp/nm and TAD like feature architecture. The average dSCC value between 3DMax model and the known structure for 28 contact matrices (7 levels of structural variability with four noise levels each) for the conversion factor (α): 0.1, 0.3, 0.5, 1.0, 1.5 and 2.0 respectively. The dataset has resolution 150bp/nm and TAD like feature architecture. The bold text represents the highest dSCC value.

Conversion factor(α)	**0.1**	**0.3**	**0.5**	**1.0**	**1.5**	**2.0**
dSCC	0.564	**0.720**	0.697	0.650	0.650	0.495

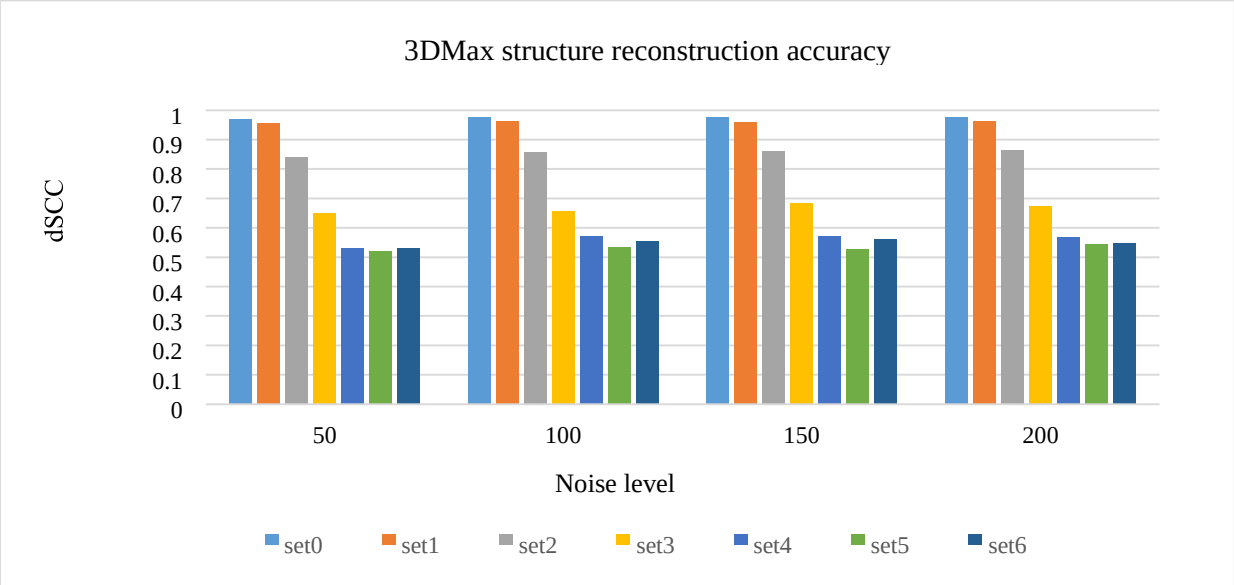


Figure 2.3. The dSCC accuracy of the structures generated by 3DMax for the synthetic data. The dSCC accuracy of the structures generated by 3DMax at different levels of noise and structural variability for conversion factor (α) = 0.3. The dataset has resolution 150bp/nm and TAD like feature architecture. Y-axis denotes the distance Spearman correlation coefficient (dSCC) score in the range [-1,1] and the X-axis denotes the noise level. Set 0-6 denotes seven different levels of structural variability in the increasing order. A higher dSCC value means the better accuracy.

Similarly, we evaluated 3DMax on 28 contact matrices of the synthetic dataset with resolution 150bp/nm and non-TAD like feature architecture. Table 2.5 shows the performance of 3DMax for different α values.

Table 2.5. The average dSCC value for the dataset with resolution 150bp/nm and non-TAD like feature architecture. The average dSCC value between 3DMax model and the known structure for 28 contact matrices (7 levels of structural variability with four noise levels each) for the conversion factor (α): 0.1, 0.3, 0.5, 1.0, 1.5 and 2.0 respectively. The dataset has resolution 150bp/nm and non-TAD like feature architecture. The bold text represents the highest dSCC value.

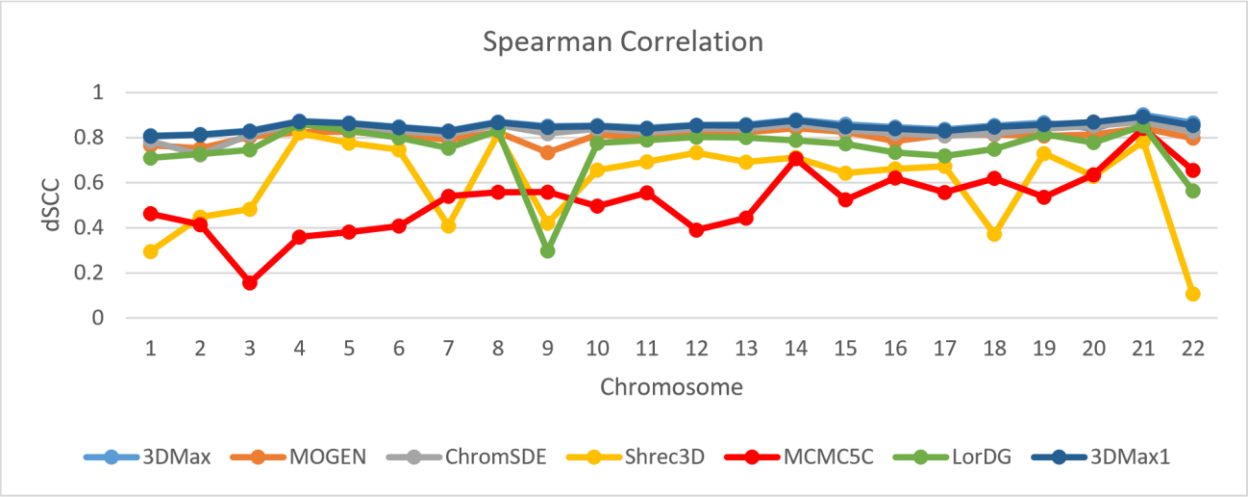
Conversion factor(α)	0.1	0.3	0.5	1.0	1.5	2.0
dSCC	0.583	**0.658**	0.634	0.566	0.518	0.429

2.4.4 Assessment on real Hi-C data

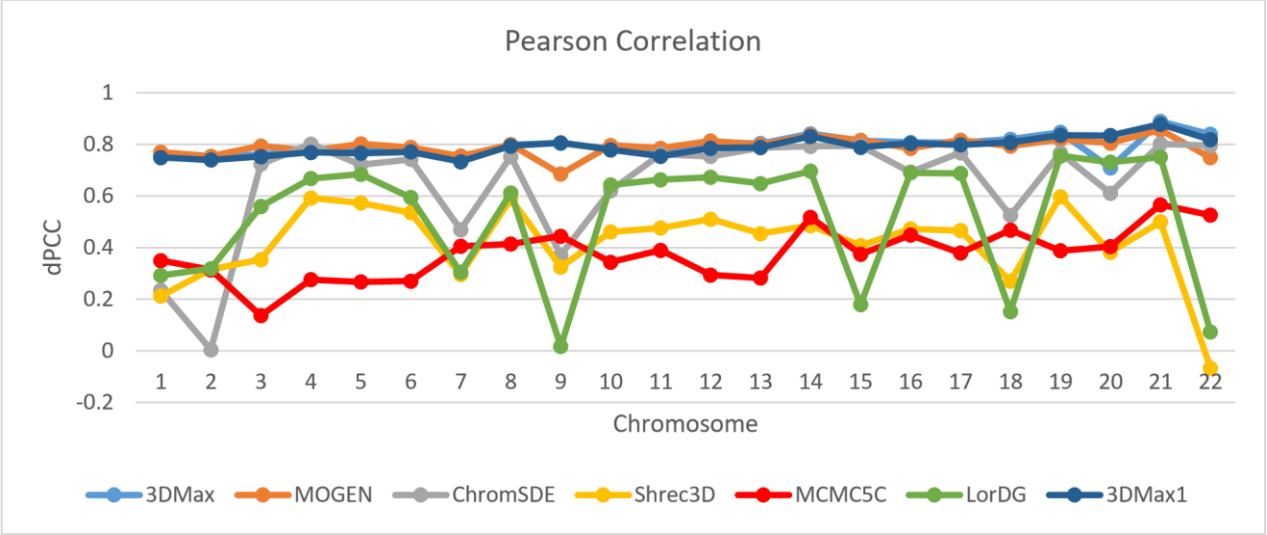
We applied 3DMax to a 1MB resolution Hi-C dataset of GM06990 cell line [144]. The Hi-C data for this cell line was generated with two different restriction enzymes: Ncol and HindIII. For comparison, we applied seven structure prediction methods 3DMax, 3DMax1 based on AdaGrad optimization algorithm, ShRec3D, ChromSDE, MCMC5C, MOGEN, and LorDG[69] to predict the 3D structure of chromosomes of this cell line. All the methods take as input an interaction frequency matrix normalized by using the normalization pipeline in [133]. We used the distance Spearman Correlation Coefficient (dSCC) and the distance Pearson Correlation Coefficient (dPCC) to assess the accuracy of these methods. The accuracy is determined by computing the

dSCC value between the distance matrix of the normalized frequency input matrix and the Euclidean distance calculated from the predicted 3D structures. Figure 2.5(a) shows that 3DMax outperforms the other methods by at least 4% across 22 pairs of non-sex chromosomes of the cell line. 3DMax obtained an average spearman correlation coefficient of 0.85 across all the chromosomes while the second highest among the other methods has the coefficient of 0.82. Figure 2.5(b) shows the Pearson correlation coefficient on the GM06990_HindIII cell. 3DMax obtained the highest average Pearson correlation coefficient of 0.795, which is better than the other methods.

(a)



(b)



(c)

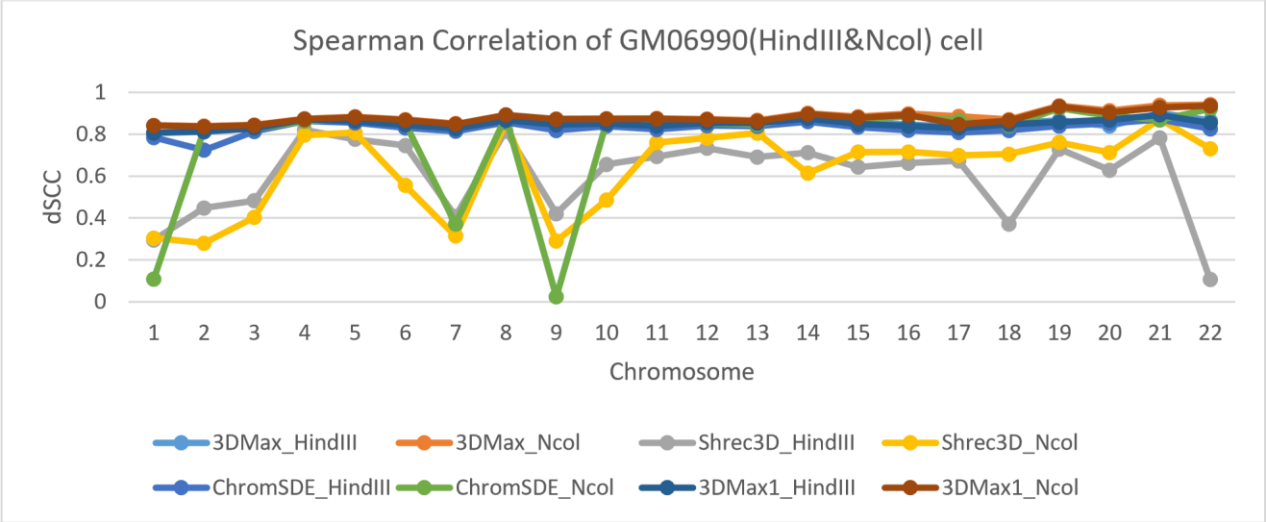


Figure 2.4. A comparison of the accuracy of different methods on real Hi-C datasets. (a) The Spearman Correlation Coefficient of 3DMax, 3DMax1, MOGEN, ChromSDE, ShRec3D, MCMC5C, and LorDG on the normalized contact maps of GM06990_HindIII cell. (b) The Pearson Correlation Coefficient of 3DMax, 3DMax1, MOGEN, ChromSDE, ShRec3D, MCMC5C, and LorDG on the normalized contact maps of GM06990_HindIII cell. (c) The Comparison of 3DMax, 3DMax1, ChromSDE and ShRec3D on the normalized contact maps of GM06990 HindIII and Ncol

cell. Y-axis denotes either the distance Spearman Correlation Coefficient (dSCC) score in the range [-1,1] or the distance Pearson Correlation Coefficient score (dPCC) in the range [-1,1]. X-axis denotes the Chromosome number. A higher dSCC value means the better accuracy.

In Figure 2.5(c) we compared the spearman correlation values of ShRec3D, ChromSDE, 3DMax, and 3DMax1 for the contact maps of GM06990 cell line with NcoI and HindIII restriction enzymes. 3DMax has the highest average dSCC value of 0.88 across the chromosomes of the cell line. Table 2.7 shows a tabular representation of the model accuracy comparison visualized in Figure 2.5.

Table 2.6. A comparison of the accuracy spread of the different methods on real Hi-C datasets. Top: The Spearman Correlation Coefficient of 3DMax, 3DMax1, MOGEN, ChromSDE, ShRec3D, MCMC5C, and LorDG on the normalized contact maps of GM06990_HindIII cell, and the Pearson Correlation Coefficient of 3DMax, 3DMax1, MOGEN, ChromSDE, ShRec3D, MCMC5C, and LorDG on the normalized contact maps of GM06990_HindIII cell. Bottom: The Comparison of dSCC values of 3DMax, 3DMax1, ChromSDE and ShRec3D on the normalized contact maps of GM06990 HindIII and Ncol cell. The values denote the distance Spearman Correlation Coefficient (dSCC) score in the range [-1,1] or the distance Pearson Correlation Coefficient score (dPCC) in the range [-1,1].

	SPEARMAN CORRELATION						
Chromosome	**3DMax**	**3DMax1**	**MOGEN**	**ChromSDE**	**Shrec3D**	**MCMC5C**	**LorDG**
1	0.8088	0.8062	0.7662	0.7845	0.2951	0.463	0.7101
2	0.8149	0.8126	0.7526	0.7245	0.4482	0.4143	0.7275
3	0.8306	0.828	0.8044	0.814	0.4827	0.1564	0.7459
4	0.8736	0.8715	0.8245	0.8636	0.8203	0.3595	0.8607
5	0.8653	0.8631	0.8266	0.8551	0.7762	0.3813	0.8317
6	0.848	0.845	0.8104	0.8303	0.7465	0.4078	0.8002
7	0.831	0.8278	0.7925	0.8144	0.4087	0.5402	0.7536
8	0.8701	0.8675	0.8236	0.857	0.8152	0.5584	0.8317
9	0.851	0.846	0.7339	0.8184	0.421	0.5584	0.2972
10	0.854	0.8505	0.8129	0.8392	0.6561	0.4967	0.7759
11	0.8433	0.8398	0.8003	0.823	0.6936	0.5559	0.7896
12	0.8558	0.8544	0.8259	0.8413	0.7332	0.3907	0.803
13	0.8584	0.8537	0.8242	0.8381	0.6917	0.4437	0.8007
14	0.8799	0.8754	0.8425	0.8605	0.7123	0.7065	0.7879
15	0.8592	0.8488	0.8255	0.8346	0.6432	0.5246	0.7725
16	0.8466	0.8397	0.7854	0.8188	0.6621	0.6208	0.7345
17	0.837	0.8298	0.8127	0.8083	0.6732	0.557	0.719
18	0.8537	0.8475	0.8139	0.8185	0.3717	0.6197	0.7492
19	0.8668	0.8579	0.8077	0.8397	0.73	0.5362	0.8152
20	0.8392	0.869	0.8146	0.8527	0.6291	0.6361	0.7779
21	0.9017	0.8925	0.8421	0.8704	0.7831	0.841	0.8532
22	0.866	0.8542	0.7977	0.8264	0.1065	0.6554	0.5639

	PEARSON CORRELATION						
Chromosome	**3DMax**	**3DMax1**	**MOGEN**	**ChromSDE**	**Shrec3D**	**MCMC5C**	**LorDG**
1	0.7611	0.7491	0.7697	0.2352	0.2125	0.3497	0.2922
2	0.7511	0.7401	0.7544	0.0042	0.3154	0.314	0.3187
3	0.7603	0.7532	0.7938	0.7238	0.3539	0.1368	0.5597
4	0.7813	0.7691	0.7739	0.8016	0.5922	0.2758	0.6675
5	0.779	0.7661	0.8021	0.7215	0.5732	0.2673	0.6845
6	0.7834	0.7709	0.7883	0.7422	0.5361	0.2705	0.5945
7	0.7471	0.7334	0.7549	0.4693	0.2961	0.405	0.3044
8	0.7994	0.794	0.7988	0.7533	0.5895	0.4138	0.6126
9	0.8046	0.8063	0.6852	0.3711	0.3253	0.4446	0.017
10	0.7836	0.7793	0.7965	0.6214	0.4614	0.3436	0.6428
11	0.7628	0.7542	0.7852	0.7624	0.4761	0.39	0.6636
12	0.8098	0.7856	0.813	0.7533	0.5106	0.2941	0.6727
13	0.8037	0.7887	0.7989	0.7875	0.4544	0.2824	0.6483
14	0.8411	0.8316	0.8357	0.7928	0.4855	0.518	0.6965
15	0.8137	0.7892	0.8165	0.7948	0.4078	0.3745	0.179
16	0.8075	0.804	0.7845	0.6925	0.4726	0.4489	0.6899
17	0.8069	0.7981	0.8164	0.768	0.4662	0.3793	0.6879
18	0.82	0.8079	0.7931	0.5246	0.2697	0.468	0.1519
19	0.847	0.8356	0.8204	0.7674	0.5972	0.3881	0.7552
20	0.7096	0.8347	0.8049	0.6113	0.3825	0.4039	0.731
21	0.8892	0.8784	0.8561	0.802	0.5	0.5663	0.7509
22	0.8396	0.8181	0.7486	0.7958	-0.067	0.5262	0.0737

	SPEARMAN CORRELATION OF GM06990 (HINDIII &NCOL) CELL							
Chromosome	**3DMax_HindIII**	**3DMax_Ncol**	**3DMax1_HindIII**	**3DMax1_Ncol**	**Shrec3D_HindIII**	**Shrec3D_Ncol**	**ChromSDE_HindIII**	**ChromSDE_Ncol**
1	0.8088	0.8432	0.8062	0.8412	0.2951	0.3043	0.7845	0.1085
2	0.8149	0.8387	0.8126	0.8367	0.4482	0.2797	0.7245	0.8228
3	0.8306	0.8447	0.828	0.8425	0.4827	0.403	0.814	0.8271
4	0.8736	0.874	0.8715	0.872	0.8203	0.796	0.8636	0.8624
5	0.8653	0.8836	0.8631	0.8816	0.7762	0.8077	0.8551	0.872
6	0.848	0.8701	0.845	0.8677	0.7465	0.556	0.8303	0.8539
7	0.831	0.8509	0.8278	0.8483	0.4087	0.3147	0.8144	0.3709
8	0.8701	0.8509	0.8675	0.8924	0.8152	0.8559	0.857	0.8832
9	0.851	0.8732	0.846	0.8721	0.421	0.2899	0.8184	0.0239
10	0.854	0.8753	0.8505	0.8723	0.6561	0.4865	0.8392	0.8603
11	0.8433	0.876	0.8398	0.8731	0.6936	0.76	0.823	0.8603
12	0.8558	0.873	0.8544	0.8698	0.7332	0.7819	0.8413	0.8531
13	0.8584	0.8665	0.8537	0.8621	0.6917	0.8064	0.8381	0.8457
14	0.8799	0.9	0.8754	0.8965	0.7123	0.6141	0.8605	0.887
15	0.8592	0.8842	0.8488	0.879	0.6432	0.715	0.8346	0.8707
16	0.8466	0.8975	0.8397	0.8921	0.6621	0.7156	0.8188	0.8856
17	0.837	0.8858	0.8298	0.8473	0.6732	0.6988	0.8083	0.866
18	0.8537	0.8701	0.8475	0.865	0.3717	0.7055	0.8185	0.8407
19	0.8668	0.936	0.8579	0.9324	0.73	0.7613	0.8397	0.925
20	0.8392	0.9133	0.869	0.9037	0.6291	0.7128	0.8527	0.8878
21	0.9017	0.9382	0.8925	0.9274	0.7831	0.873	0.8704	0.8688
22	0.866	0.9414	0.8542	0.9359	0.1065	0.7311	0.8264	0.922

On average, 3DMax's accuracy is at least 3% higher than the other methods. In addition, since each Hi-C data obtained with a restriction enzyme is an independent observation of the GM06690 cell, we checked the robustness of our method by comparing the predicted structure from Ncol with one from the HindIII enzyme. We compared the predicted structure of chromosome 19 of HindIII data and NcoI replicate data. The dSCC and dRMSE value of the comparison were 0.9 and 0.0064 respectively, suggesting the two models are very similar.

2.4.5 Comparison with existing methods on the simulated data

We compared 3DMax with three existing methods: MCMC5C [48], MOGEN [80], and ShRec3D [79]. We used each method to generate an ensemble of 50 structures for each input matrix. We compared each structure in the ensemble with the true structures (*i.e.*, a set of 100 true structures for each structural variability level) using Spearman correlation coefficient to select the reconstructed structure closest to a true structure from the ensemble. The Spearman correlation coefficient of the selected structure and the true structures is averaged and used as the dSCC accuracy for the method. For clarity, the comparison is grouped based on the noise level of the simulated data from 50 to 200. For the different noise levels, 3DMax is comparable to the top method - MOGEN when structural variability (sets 0-1) is low. And as the variability increases (especially sets 3-6), it outperforms all the other methods (Figure 2.4) most time. Table 2.6 shows a tabular representation of the dSCC values visualized in Figure 2.4, to show the dSCC values generated by all the algorithms.

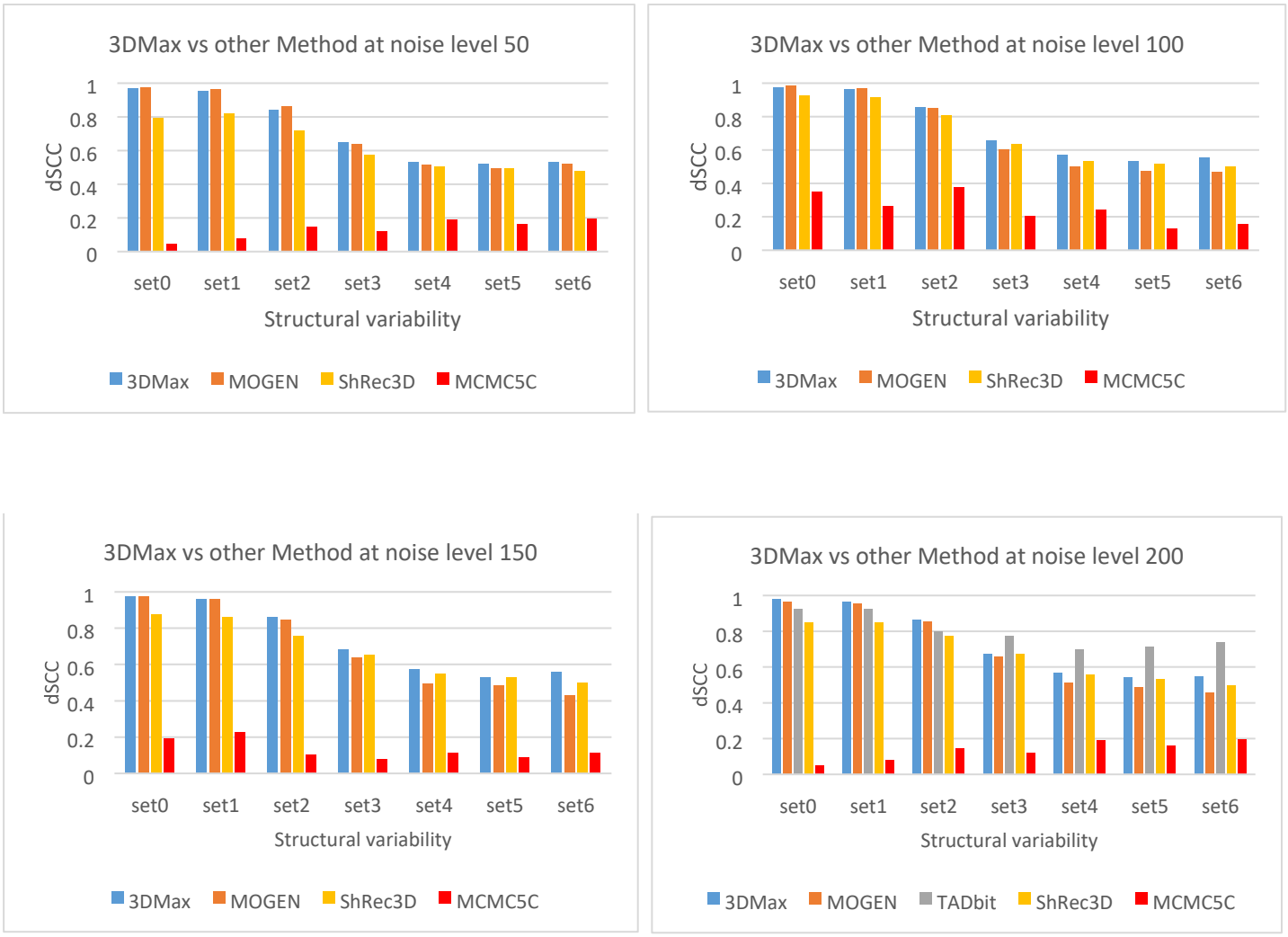


Figure 2.5. A comparison of the reconstruction accuracy of different methods on the synthetic dataset. The reconstruction accuracy for 3DMax, MOGEN, ShRec3D, and MCMC5C at different levels of noise and structural variability. The dataset has resolution 150bp/nm and TAD like feature architecture. Top-Left: comparison at Noise Level 50, Top-Right: comparison at Noise Level 100, Bottom-Left: comparison at Noise Level 150, Bottom-Right: comparison at Noise Level 200. Y-axis denotes the distance Spearman Correlation Coefficient (dSCC) score in the range [-1,1] and the X-axis denotes the structural variability level. Set 0-6 denotes seven different levels of structural variability in the increasing order. A higher dSCC value means the better accuracy.

Table 2.7. A comparison of the reconstruction accuracy spread of the different methods on the synthetic dataset. The reconstruction accuracy for 3DMax, MOGEN, ShRec3D, and MCMC5C at different levels of noise and structural variability. The dataset has resolution 150bp/nm and TAD like feature architecture. Noise Level 50: comparison of dSCC value at Noise Level 50, Noise Level 100: comparison of dSCC value at Noise Level 100, Noise Level 150: comparison of dSCC value at Noise Level 150, Noise Level 200: comparison of dSCC value at Noise Level 200. The table values denote the distance Spearman Correlation Coefficient (dSCC) score in the range [-1,1] and the SV denotes the structural variability level. Set 0-6 denotes seven different levels of structural variability in the increasing order. A higher dSCC value means the better accuracy.

SV	**Noise Level 50**			
	3DMax	**MOGEN**	**ShRec3D**	**MCMC5C**
set0	0.9708	0.9755	0.7928	0.0481
set1	0.9552	0.9648	0.8188	0.0779
set2	0.8405	0.8625	0.7175	0.1477
set3	0.6505	0.6406	0.5722	0.1201
set4	0.5302	0.5135	0.502	0.1916
set5	0.5211	0.4945	0.4938	0.1614
set6	0.5303	0.5211	0.4767	0.1938
SV	**Noise Level 100**			
	3DMax	**MOGEN**	**ShRec3D**	**MCMC5C**
set0	0.9753	0.9835	0.9239	0.3514
set1	0.963	0.968	0.9133	0.2642
set2	0.8578	0.8527	0.8072	0.3792

set3	0.6555	0.6039	0.6338	0.2068
set4	0.5703	0.4991	0.532	0.2456
set5	0.5342	0.4728	0.5183	0.1299
set6	0.5535	0.47	0.5026	0.1578
SV	**Noise Level 150**			
	3DMax	**MOGEN**	**ShRec3D**	**MCMC5C**
set0	0.976	0.9734	0.876	0.1933
set1	0.959	0.96	0.8613	0.2275
set2	0.8612	0.8485	0.7572	0.1016
set3	0.6821	0.6362	0.6546	0.0791
set4	0.5713	0.4915	0.5475	0.1146
set5	0.5285	0.4835	0.5268	0.0858
set6	0.5601	0.4318	0.5009	0.1106
SV	**Noise Level 200**			
	3DMax	**MOGEN**	**ShRec3D**	**MCMC5C**
set0	0.9771	0.9655	0.8499	0.0481
set1	0.9627	0.9533	0.8481	0.0779
set2	0.8634	0.8514	0.7743	0.1477
set3	0.6724	0.6606	0.6726	0.1201
set4	0.5679	0.5131	0.5559	0.1916
set5	0.5435	0.4886	0.5292	0.1614
set6	0.5487	0.4554	0.4992	0.1938

2.4.6 Consistency checking of models in ensembles

To assess the consistency of the structures generated by 3DMax, we compared 50 structures generated at the optimal α value for each chromosome for the GM06990_HindIII cell and the malignant B-cell, respectively. We used the dSCC value to measure the similarity between these structures. Figure 2.6 shows the average dSCC for each chromosome for Hi-C data of the GM06990_HindIII cell and the malignant B-cell respectively. The average dSCC between the models is > 0.9 for all the chromosomes, indicating chromosomal models generated by 3DMax are quite similar to each other.

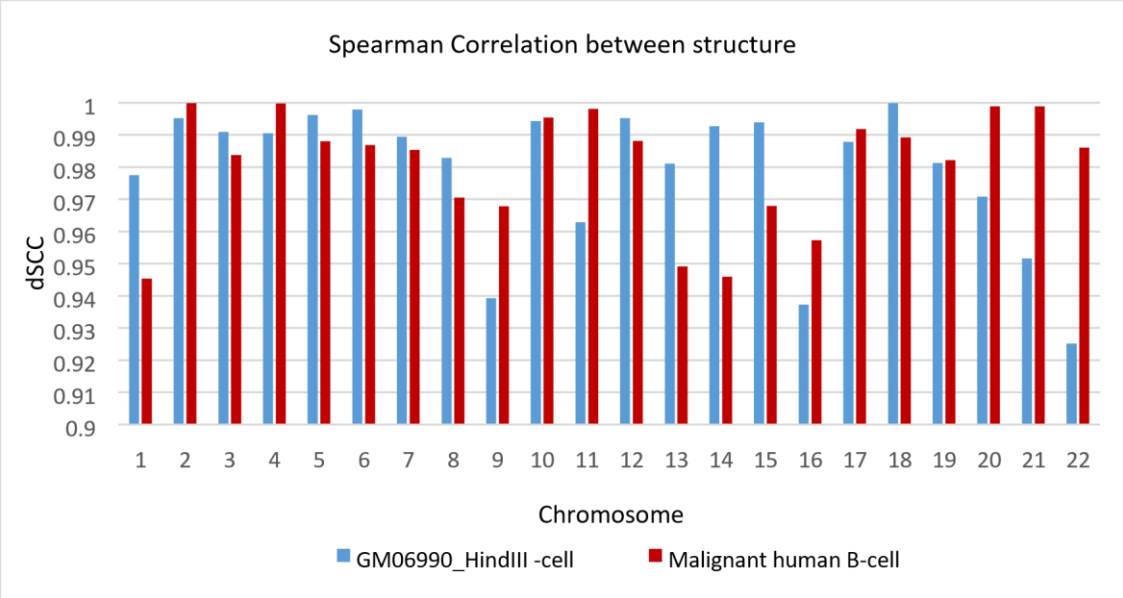


Figure 2.6. The similarity between structures generated by 3DMax. The average similarity for an ensemble of structures generated for the GM06990_HindIII cell and the malignant B-cell chromosomes using the optimal α value for each chromosome.

2.4.7 Comparative analysis of the performance of 3DMax, 3DMax1, MOGEN, ChromSDE, ShRec3D, MCMC5C, and LorDG on Hi-C data normalized with three popular normalization methods.

Due to biases in Hi-C experiments, Hi-C data is generally noisy. Some of these biases are associated with cutting frequencies of restriction enzymes, GC content and sequence uniqueness [104-105]. In order reduce the effects of these biases, the Hi-C data contact matrix is normalized to reflect the strength of the underlying chromosomal interactions more accurately.

We performed a comparative study of the performance of different 3D modeling methods when each of the three commonly used normalization techniques: Yaffe and Tanay [104] normalization technique, ICE (Iterative Correction and Eigenvector decomposition) technique [105], and Sequential Component Normalization (SCN) technique [9] is applied. Figure 2.5(a) shows the result obtained by using the Yaffe and Tanay normalization technique, where 3DMax outperformed the other methods. Table 2.8 shows the average dSCC value for different chromosomes for each of the normalization technique. 3DMax and 3DMax1 produces the best performance when the Yaffe and Tanay normalization technique is used, and the 3DMax1 produces the best performances when the ICE and SCN normalization method are used respectively. It is evident from the results that the normalization techniques have a significant impact on the performance of some 3D modeling methods.

Table 2.8. The average dSCC score of the chromosomal models of the GM06990 cell line reconstructed with three normalization techniques. The average dSCC scores of chromosomal models of the GM06990 cell line reconstructed by 3DMax, 3DMax1, MOGEN, ChromSDE, ShRec3D, MCMC5C, and LorDG with the three normalization methods. The top 2 scores for each normalization technique are highlighted in bold text.

	3DMax	3DMax1	MOGEN	ChromSDE	Shrec3D	MCMC5C	LorDG
Yaffe &Tanay	**0.85**	**0.85**	0.81	0.82	0.60	0.52	0.75
ICE	0.75	**0.85**	0.61	**0.83**	0.60	0.032	0.78
SCN	0.72	**0.85**	0.58	**0.83**	0.71	0.028	0.79

2.5 Discussion

2.5.1 Comparison of the computing performance of the different methods

To improve the computing performance and the usability of our algorithm, we also implemented the 3DMax algorithm in the Java programming language (available via http://sysbio.rnet.missouri.edu/bdm_download/3DMax). The performance comparison of the MATLAB and the Java programming versions for a GM06990_HindIII cell line dataset is shown in Figure 2.7. As shown in the Figure, the result produced by two separate Java implementation runs is consistent with those of the MATLAB implementation. We tested 3DMax and all other methods on an Intel Core i5-2400 3.10GHz computer with 8GB RAM.

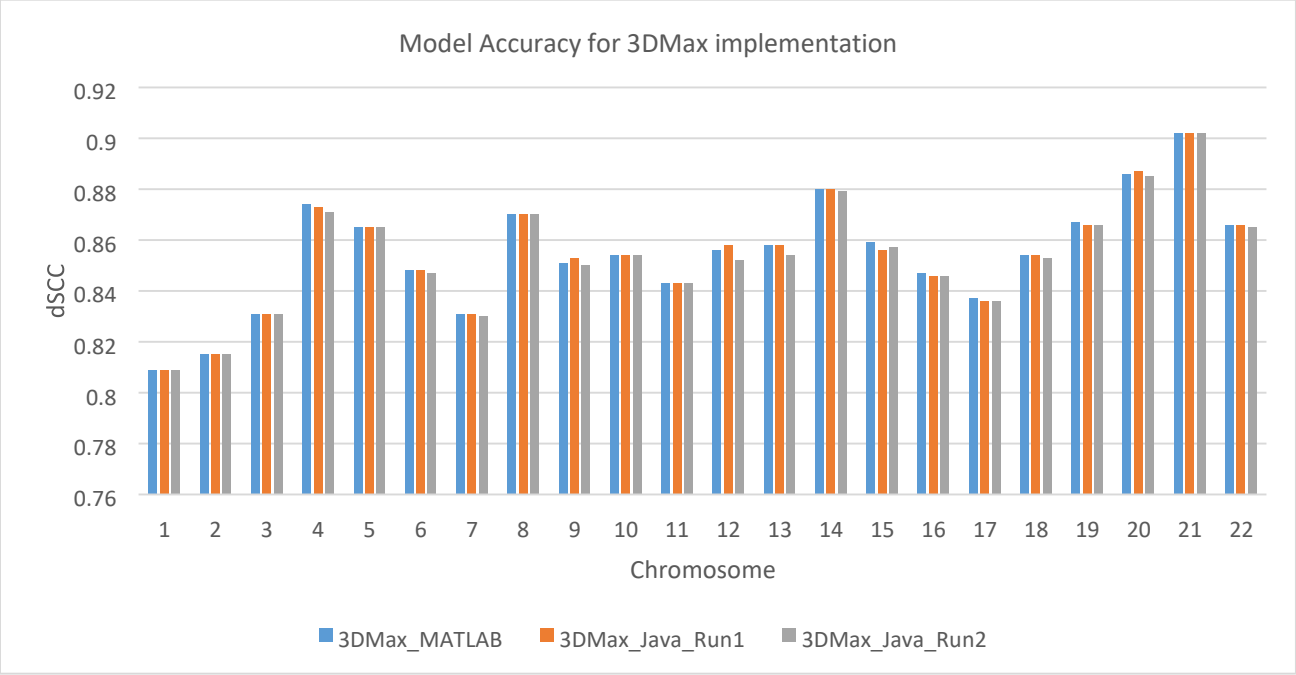


Figure 2.7. A comparison of the performance of 3DMax algorithm MATLAB and Java programing language implementation. The performance comparison of the MATLAB and the Java 3DMax implementation for a GM06990_HindIII cell line dataset. The Figure shows two different runs of the Java implementation compared against the MATLAB implementation. Models produced by both implementations are comparable with a similar accuracy. Y-axis denotes either the distance Spearman correlation coefficient (dSCC) score in the range [-1,1]. X-axis denotes the Chromosome number. A higher dSCC value means the better accuracy.

We compared 3DMax algorithm with the other algorithms mentioned above in terms of computation speed, and the memory cost. To do this, we benchmarked them against the chromosomes of GM06990_HindIII cell data. It takes 3DMax java implementation about 13 seconds to predict the structure for all the chromosomes of the entire genome when it uses a single conversion factor (α), while it generates a single structure for each chromosome. 3DMax uses about 20 minutes to generate the representative structures for the entire cell when it estimates the optimal conversion factor (α) in the range [0.1, 2].

Though ChromSDE produced one of the best results, it was memory intensive and slow to generate large structures. ChromSDE could not handle efficiently input data with > 400 bins on our machine with 8 GB RAM. We were only able to use ChromSDE to create structure on our server machine with 65GB RAM. It takes ChromSDE 20-25 hours to generate structure for the entire GM06990_HindIII cell data. MOGEN uses over 2 hours to generate the models for the cell line. It takes LorDG about 1 hour and 7 minutes to process the whole cell line. MCMC5C with the default parameters uses 1 hour and 19 minutes to generate the models. But to obtain better accuracy by increasing the number of iterations and the number of structures generated, the MCMC5C algorithm could run for > 18 hours before it converges.

2.5.2 Validation using FISH data

We validated the model of Chromosome 22 reconstructed by 3DMax with an independent FISH data for GM06990_HindIII cell. Four 3D FISH probes for four *loci* (L5, L6, L7, L8) of the consecutive positions alternate between two chromosome compartments (A and B) [30]. That is, locus L5 and locus L7 are in Compartment A, and locus L6 and locus L8 are in Compartment B. According to the FISH data, L7 is spatially closer to L5 than to L6, though L6 lies between L5 and L7 on the chromosome sequence. Likewise, L6 is spatially closer to L8 than to L7. To check if this holds in the reconstructed 3D model, we measured the distance between these *loci* on the predicted structure. Figure 2.8 shows a model constructed by 3DMax with the four probes L5, L6 L7, L8 colored green, blue, yellow, magenta respectively. The distances between these *loci*: L5 – L6, L5 – L7, L6 – L7, L6-L8 are reported. Indeed, the distance L5 – L7 was shorter than L5 – L6 and the distance L6 - L8 was shorter than L6 – L7. The 3D structure was visualized with Pymol [45].

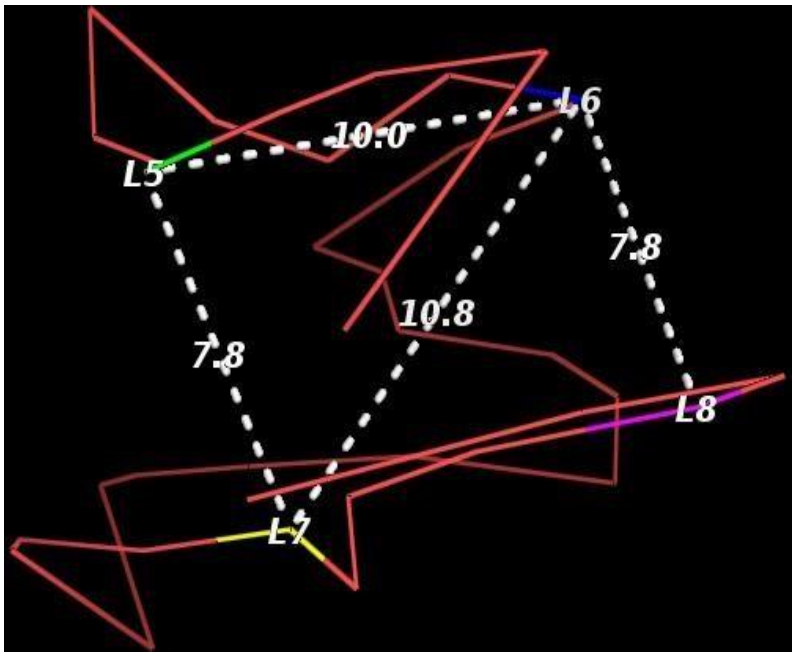


Figure 2.8. Validation with FISH data. Distances between four fluorescence *in situ* hybridization (FISH) probes in the model of Chromosome 22 reconstructed by 3DMax. L5, L6, L7 and L8 denote four probes. The distances between the probes are labelled along the virtual line segments connecting the probes.

2.6 Conclusions

We developed a new method (3DMax) based on the maximum likelihood inference to reconstruct the 3D structure of chromosomes from Hi-C data. 3DMax combines a maximum likelihood algorithm and a gradient ascent method to generate optimized structures for chromosomes. The results on synthetic datasets show that the method performs robustly in the presence of noise and structural variability. This method provides a way to automatically determine the best conversion factor (α) for any Hi-C contact data. The results on the real Hi-C datasets reveals that 3DMax can effectively reconstruct chromosomal models from Hi-C contact matrices normalized by different methods. We also show that a major strength of the 3DMax algorithm is that it is faster and has a low memory requirement compared to some other methods.

3 GSDB: a database of 3D chromosome and genome structures reconstructed from Hi-C data

3.1 Abstract

Advances in the study of Chromosome conformation capture(3C) technologies, such as Hi-C technique - capable of capturing chromosomal interactions in a genome-wide scale - have led to the development of several three-dimensional(3D) chromosome structure reconstruction methods from Hi-C data. The chromosome 3D structure is important because it has been shown to play a role in a variety of important biological activities occurring within the cell such as DNA replication, gene regulation, genome interaction, and gene expression. In recent years, numerous Hi-C datasets have been generated, and likewise, a number of genome structure construction algorithms have been developed. However, until now, there has been no freely available repository for 3D chromosome structures generated from Hi-C data. In this work, we outline the construction of such a database called the Genome Structure Database (GSDB) to create a comprehensive repository that contains 3D structures for Hi-C datasets from a variety of 3D structure reconstruction tools developed over the years. The Genome Structure Database is the first database to provide a repository for 3D chromosome and genome structures constructed from different Hi-C data reconstruction methods. Our database contains over 50,000 structures constructed by 12 start-of-the-art Hi-C data structure prediction methods for publicly used Hi-C datasets with varying resolution. The database is useful for the community to study the function of genome from a 3D perspective. GSDB is accessible at http://sysbio.rnet.missouri.edu/3dgenome/GSDB

3.2 Introduction

The three-dimensional (3D) organization of the genome plays a significant role in many diverse biological functions and processes including gene expression [14], regulation [12,13] and

transcriptional regulation [129]. Several studies of the architecture of the genome in the cell have linked genome structure to the mechanism of these functions; hence, it is essential to understand the spatial arrangement within the cell nucleus in order to fully elucidate this relation [25,47,87]. Early studies of the structure of the genome have relied on the use of microscopy techniques such as fluorescence in situ hybridization (FISH), a technique that employs fluorescence probes to detect the presence of a specific chromosome region and the proximity between two regions in a genome sequence [2,18-19]. Other microscopy methods developed to study the genome organization include stimulated emission depletion (STED) [20], stochastic optical reconstruction microscopy (STORM), and Photo-activated localization microscopy (PALM or FPALM) [22,23,145]. While these techniques have proven useful in providing insights into the organization of the genome for DNA fragments or chromatin regions, they are limited and unsuitable for an overall view of the genome-wide inter-and intra-chromosomal relationship study of the genome within the cell nucleus [145].

In order to capture these inter- and intra- chromosomal interactions, a variety of next-generation, high-throughput sequencing technologies have emerged including: 3C [24], 4C [28], 5C [29], Hi-C [30], TCC [31] and ChIA-PET [32,33]. Out of all these techniques, the Hi-C technique has seen a particularly high usage because of its ability to comprehensively map the chromatin interactions at a genome wide scale.

A Hi-C experiment results in the generation of an interaction frequency (IF) matrix for chromosomal regions (*loci*) within a chromosome or between any two chromosomes in a population of cells [30,40,69,78]. With the advancement of the Hi-C research, sophisticated tools such as GenomeFlow [40], Juicer [41], and HiC-Pro[42] have been developed to generate IF

matrices from raw sequence pair reads data[28]. Some methods represent the contact matrix as a sparse 3-column matrix where columns 1-2 denote the interacting *loci* and column 3 denotes the number of interactions (or contacts) between the corresponding *loci* in a Hi-C dataset [69,70,80]. Many methods have been developed for chromosome 3D structure reconstruction from chromosome conformation capture (3C) such as the Hi-C data. Generally, these data-driven methods can be grouped into three classes [147] based on how the IF is used for 3D structure construction: distance-based, contact-based and probability-based. First, distance-based methods implement the 3D structure construction through a two-step process. These methods convert the IF matrix to a distance matrix between *loci* based on an inverse relation observed from FISH 3D distance data [31]. An optimization function is thereafter used to infer a 3D structure from an initial random structure with the objective of satisfying the distances in the distance matrix as much as possible [45-47,69,70,76, 79,88,92,97]. Second, contact-based methods consider each chromosomal contact as a restraint and apply an optimization algorithm to ensure that the number of contacts in the input contact matrix is satisfied in the 3D structure [80,83,91,93]. Third, probability-based methods define a probability measure over the IF, by constructing the structure inference problem as a maximum likelihood problem and thereafter using a sampling *e.g.* Markov chain Monte Carlo (MCMC) or optimization algorithm to solve the prediction problem [75, 78, 86, 95].

Here, we present Genome Structure Database (GSDB), a novel database that contains the chromosome 3D structural models of publicly and commonly used Hi-C datasets reconstructed by twelve state-of-the-art 3D structure reconstruction algorithms at various Hi-C data resolution ranging from 25KB – 10MB. The database is organized such that users can view the structures online and download the 3D structures constructed for each dataset by all the reconstruction

methods. Our database is the first of its kind to provide a repository of 3D structures and the evaluation results for 3D structures constructed from different Hi-C data reconstruction methods all in one place.

3.3 Materials and Methods

3.3.1 Datasets

Our Hi-C data is pulled from a variety of sources which we list here. Some datasets were downloaded from the Gene Expression Omnibus (GEO) database, including the Hi-C contact matrices datasets (GEO accession Number: GSE63525) of cell line GM12878 Hi-C count matrices (MAPQ ≥30) from Rao *et al.* [144], normalized interaction matrices for each of the four cell types analysis -mouse ES cell, mouse cortex, human ES cell (H1), and IMR90 fibroblasts – (GEO accession Number: GSE35156) [141,148], and the Hi-C contact matrices datasets (GEO Accession Number: GSE18199) of karyotypically normal human lymphoblastic cell line(GM06990, K562)[30]. All other Hi-C datasets were obtained from the ENCODE project repository [149], and the GEO accession Number and the ENCODE ID for each dataset are available on the Genome Structure Database website.

3.3.2 Normalization

Hi-C data normalization is an important process in 3D structure reconstruction from Hi-C data, because the raw contact count matrix obtained from 3C experiments may contain numerous systematic biases, such as GC content, length of restriction fragments, and other technical biases that could influence the 3D structure reconstruction [9,104-106,150]. Consequently, all the contact matrices were normalized prior to applying the 3D structure reconstruction algorithms. GM12878 cell line datasets was normalized using the Knight–Ruiz normalization (KR) method [106] [144], and the normalized interaction matrices downloaded from Dixon *et al.* [141] were

normalized using Yaffe and Tanay normalization method [104.] The Vanilla Coverage (VC) technique [144] was used as the default technique to normalize all the other Hi-C datasets.

3.3.3 Database Implementation

The GSDB website interface was implemented using HTML, PHP and JavaScript, and the database was implemented in MySQL (https://www.mysql.com/). The online 3D structure visualization was done through 3Dmol viewer, a molecular visualization JavaScript library [151].

3.3.4 3D Modeling Algorithms Included

We used twelve existing algorithms for the 3D structure construction. We selected a mixture of distance-based, contact-based, and probability-based algorithms [147]. We first describe the distance-based algorithms. LorDG [69] uses a nonlinear Lorentzian function as the objective function with the main objective of maximizing the satisfaction of realistic restraints rather than outliers. LorDG uses a gradient ascent algorithm to optimize the objective function. 3DMax [70] used a maximum likelihood approach to infer the 3D structures of a chromosome from Hi-C data. A log-likelihood was defined over the objective function which was maximized through a stochastic gradient ascent algorithm with per-parameter learning rate [113]. Chromosome3D [46] uses Distance Geometry Simulated Annealing (*DGSA*) to construct chromosome 3D structure by translating the distance to positions of the points representing *loci*. Chromosome3D adopts the Crystallography & NMR System (CNS) suite [111] which has been rigorously tested for protein structure construction for the 3D genome structure prediction from Hi-C data. HSA [47] introduced an algorithm capable of taking multiple contact matrices as input to improve performance. HSA can generate same structure irrespective of the restriction enzyme used in the Hi-C experiment. miniMDS [92] proposed an algorithm to model Hi-C data by partitioning the contact matrix first into segments and building the 3D structure bottom-up from each segment

which are eventually aggregated to form a final 3D structure. ChromSDE [76] (Chromosome Semi-Definite Embedding) framed the 3D structure reconstruction problem as a semi-definite programming problem. Shrec3D [79] formulated the *3D* structure reconstruction problem *as* a graph problem and attempts to find the shortest-path distance between two nodes on the graph. The length of a link is determined as the inverse contact frequency between its end nodes. Each fragment is regarded as the nodes connected by a link. The represented 3D structure for a Hi-C data is one in which distance between the nodes is the shortest. InfoMod3DGen [64] converts the IF to a distance matrix and used an expectation-maximization (EM) based algorithm to infer the 3D structure.

In the contact-based category, we used MOGEN [80] and GEM [93] for the 3D structure reconstruction. MOGEN [80] does not require the conversion of IF to distances and is suitable for large-scale genome structure modeling. GEM [93] considers both Hi-C data and conformational energy derived from knowledge about biophysical models for 3D structure modeling. It used manifold learning framework, which is aimed at extracting information embedded within a high-dimensional space, in this case the Hi-C data.

Lastly, in the probability-based category, Pastis [78] defined a probabilistic model of IF and casted the 3D inference problem as a maximum likelihood problem. It defined a Poisson model to fit contact data and used an optimization algorithm to solve it. SIMDA3D[95] used a Bayesian approach to infer 3D structures of chromosomes from single cell Hi-C.

3.3.5 Computational Model Reconstruction

The GSDB chromosome structure generation was done on three server machines: a x86_64 bit Redhat-Linux server consisting of multi-core Intel(R) Xeon(R) CPU E7-L8867 @ 2.13GHz with 120 GB RAM, x86_64 bit Redhat-Linux server consisting of multi-core Intel(R) Xeon(R) CPU

E5649 @ 2.53GHz with 11GB RAM, x86_64 bit Redhat-Linux server consisting of multi-core AMD Opteron(tm) Processor 4284 @ 3.0GHz with 62GB RAM, and a high-performance computing cluster (Lewis) with Linux. Using a high-performance computing (HPC) cluster machine, we allocated 10 cores, 80G of memory, with a time limit of 2 days for each chromosome structure reconstruction task per algorithm. Structures not constructed within 48 hours were terminated.

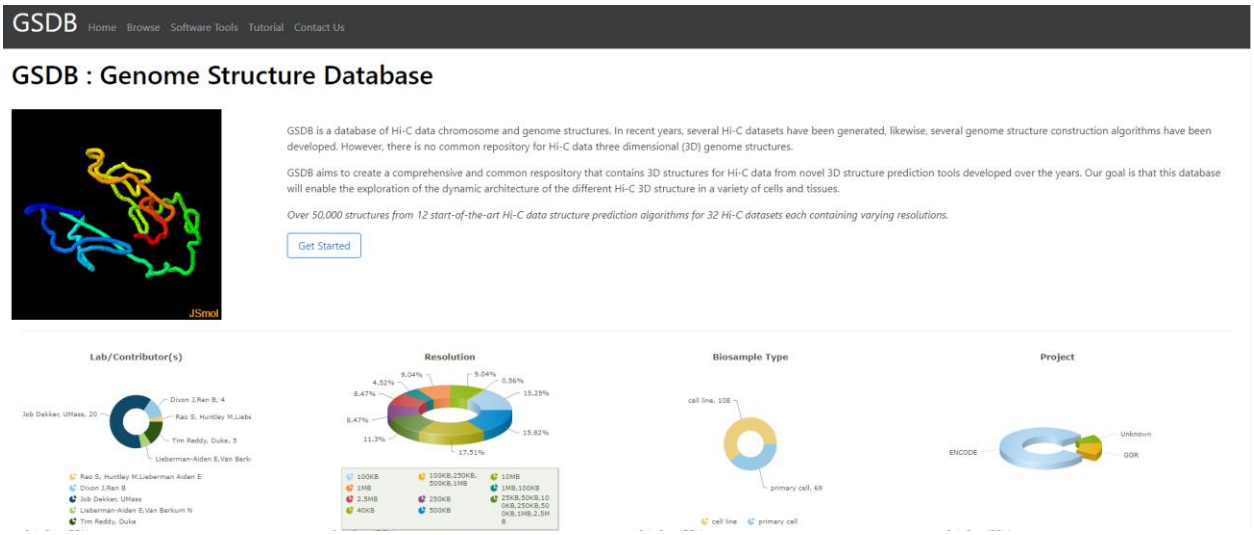


Figure 3.1. Highlights the two ways to access the database from the homepage. Clicking on the "Browse" menu in the Navigation tab or on the "Get started" button on the home page will load the Database search window.

3.4 Database Content and Usage

All the 3D structures in the GSDB have been pre-generated, so that the 3D structure visualization is faster and can be easily downloaded. The steps to navigating the database have been separated into five sections as follows:

i. Browse the database (Figure 3.1) ─ Click on “Browse” menu in the navigation bar to load the full list of the Hi-C datasets Alternatively, users can click on the “Get Started” button on the homepage.

ii. Search the database ─ The GSDB provides two ways to search for a Hi-C data and its corresponding 3D models:

   a. GSDB provides a summary of the information provided in the database through a Summary Pane. By clicking on a property/item in the summary, the user can search the database for all the Hi-C data containing this property and their corresponding 3D structural models. (Figure 3.2)

   b. Users can search the database by typing the keywords about the filename, title of Hi-C data, Hi-C data resolution, project that Hi-C data was generated from (*e.g.* ENCODE), project ID, and the GEO accession No in the “Search Pane” (Figure 3.2).

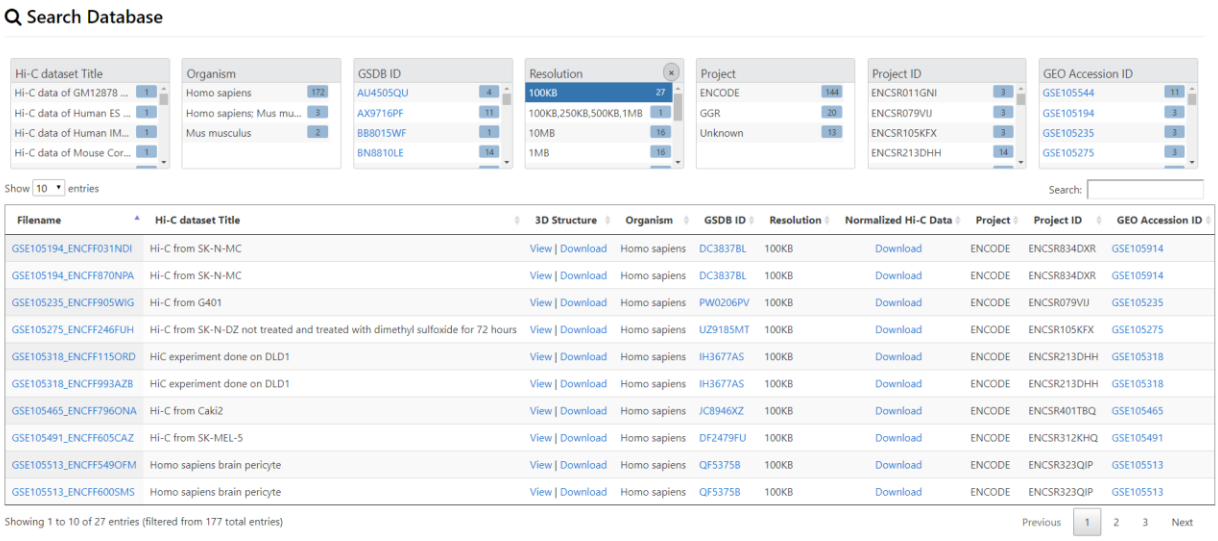


Figure 3.2. Data search and display. An example of data search using the two approaches for searching. First, search by clicking on an item on the "Summary Pane" highlighted in green. The figure shows when the user clicks on Resolution 100kb, all the datasets with 100kb resolutions are listed. Second, user can search by typing the key word in the "Search Pane" highlighted in red.

iii. 3D structure visualization and download ─ To view the details and structures for a Hi-C data, click on the "View" link in the "3D Structure Column" (Figure 3). The data information and visualization tab will be displayed (Figure 3.4). To show the 3D structure, select the algorithm, dataset, chromosome, and press Display. The structure will be displayed on the viewer. Users can download the 3D structures by clicking on the "Download" link in the "3D Structure Column" (Figure3). The normalized Hi-C datasets used for the 3D structure generation for all the algorithms can also be downloaded by clicking on the "Download" link in the "Normalized Hi-C Data" column (Figure 3).

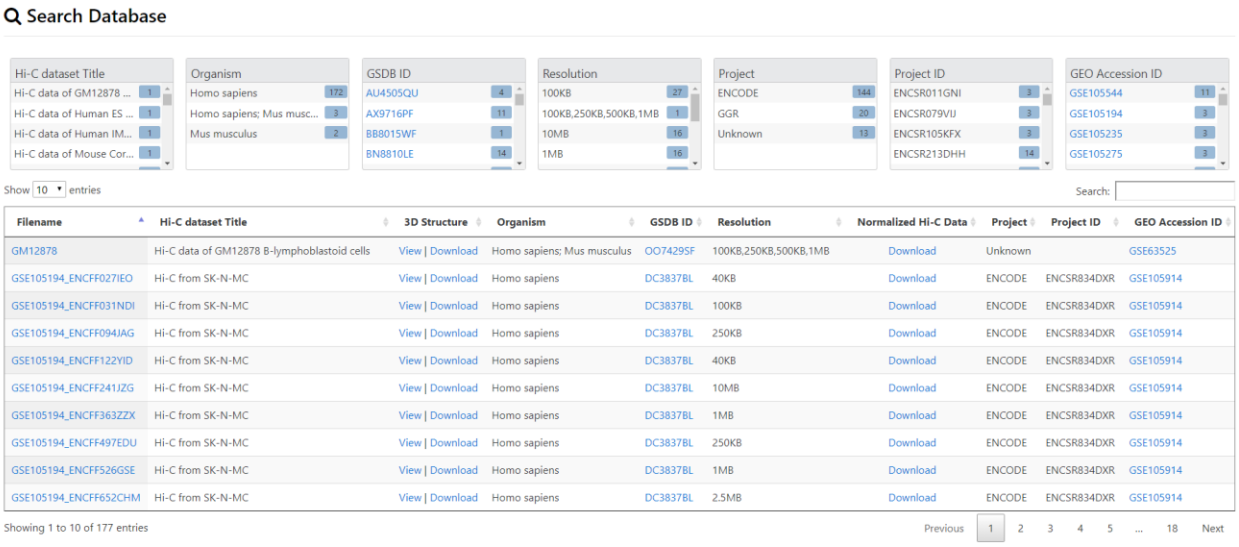


Figure 3.3. Displaying the database search window. In the "3D Structure" column, highlighted in red is the "View" link to display the 3D structure for a Hi-C data, highlighted in green is the

"Download" link to download the 3D structures constructed by the different algorithms for the Hi-C data. Pressing on the "Download" link will download the 3D structures for all the algorithms for a Hi-C data. In the "Normalized Hi-C Data" column, the "Download" link is highlighted in blue Pressing on the "Download" link will download the Normalized Hi-C data used for 3D structure construction.

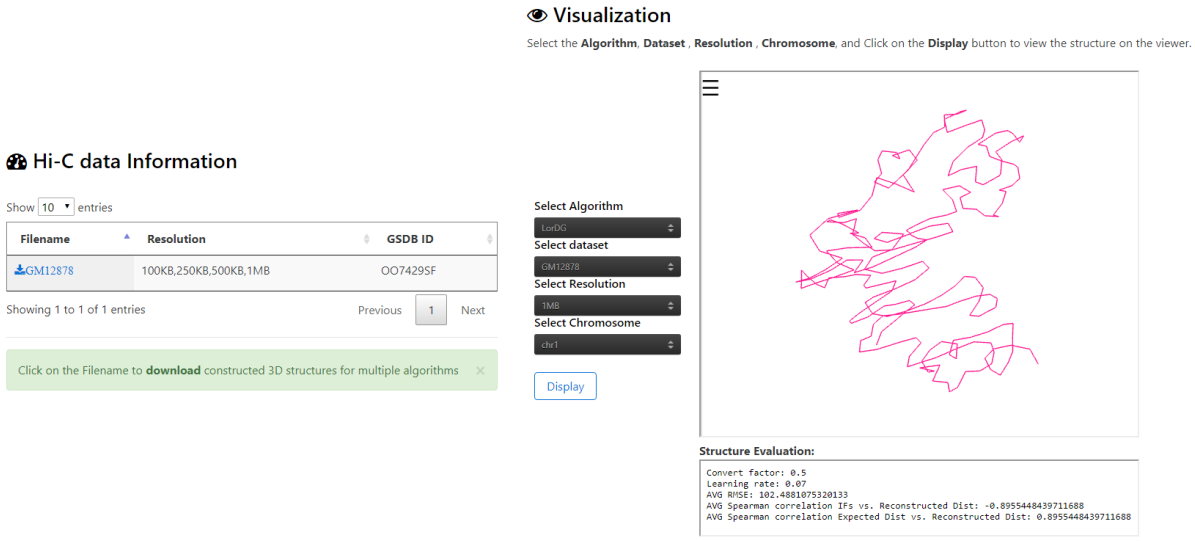


Figure 3.4. Data visualization. The figure shows the output when user click on the "View link" for the GM12878 dataset. The red highlight section shows the information about the Resolution(s) available for the Hi-C data. The blue highlight section Display the structure available for the Hi-C data. The green highlight section shows the evaluation result available for the Hi-C data. It displays the spearman correlation between the output structure and the input Hi-C data, and other evaluation result obtained. To evaluate each 3D structure, we computed the distance Spearman's Correlation Coefficient(dSCC) between reconstructed distances and distances obtained from the Hi-C datasets. The value of dSCC is in the range of -1

to +1, where higher value is better. For distance-based methods, we reported the conversion factor(α) used for the IF to distance conversion. For LorDG and 3DMax, that used gradient ascent optimization algorithm, we reported the learning rate used for the optimization process. The parameters used by each method to generate 3D structures are available on GSDB GitHub page.

3.5 Discussion and Future Development

The GSDB contains 3D structures generated from different Hi-C structure reconstruction algorithms for Hi-C data collected from multiple sources. To the best of our knowledge, it is the first repository for 3D structures generated from multiple Hi-C reconstruction algorithms. Currently, our database contains over 50,000 structures reconstructed for 32 Hi-C datasets by 12 modeling algorithms. The normalized Hi-C dataset used and 3D structures generated from all the algorithms are available to be downloaded. This database will enable the fast and easy exploration of the dynamic architecture of the different Hi-C 3D structure in a variety of cells to improve our understanding of the structural organization of various organism's chromosome and genome 3D structure. In addition, we envision that it will be helpful to researchers and scientist to keep track of the performance of the existing approaches for 3D structure construction, and also lead to the development of novel methods that outperform existing approaches. Future directions of the GSDB will include the integration of more algorithms and latest Hi-C datasets generated as the research in 3D structure construction expands.

4 ClusterTAD: an unsupervised machine learning approach to detecting topologically associated domains of chromosomes from Hi-C data

4.1 Abstract

With the development of chromosomal conformation capturing techniques, particularly, the Hi-C technique, the study of the spatial conformation of a genome is becoming an important topic in bioinformatics and computational biology. The Hi-C technique can generate genome-wide chromosomal interaction (contact) data, which can be used to investigate the higher-level organization of chromosomes, such as Topologically Associated Domains (TAD), *i.e.*, locally packed chromosome regions bounded together by intra chromosomal contacts. The identification of the TADs for a genome is useful for studying gene regulation, genomic interaction, and genome function.

Here, we formulate the TAD identification problem as an unsupervised machine learning (clustering) problem and develop a new TAD identification method called ClusterTAD. We introduce a novel method to represent chromosomal contacts as features to be used by the clustering algorithm. Our results show that ClusterTAD can accurately predict the TADs on a simulated Hi-C data. Our method is also largely complementary and consistent with existing methods on the real Hi-C datasets of two mouse cells. The validation with the Chromatin ImmunoPrecipitation (ChIP) Sequencing (ChIP-Seq) data shows that the domain boundaries identified by ClusterTAD have a high enrichment of CTCF binding sites, promoter-related marks, and enhancer-related histone modifications.

As ClusterTAD is based on a proven clustering approach, it opens a new avenue to apply a large array of clustering methods developed in the machine learning field to the TAD identification problem. The source code, the results, and the TADs generated for the simulated and real Hi-C datasets are available here: http://sysbio.rnet.missouri.edu/bdm_download/ClusterTAD/.

4.2 Background

A chromosome is known to occupy its own territory, and fold into a high-order, non-random structure in a nucleus [2]. The knowledge of the high-order organization of chromosomes is useful

for the understanding of genome folding, long-range gene interactions and regulations [152], DNA replication [153], and cellular functions [140,154]. To gain better insights into the organization of the chromosomes in a cell, a technology called the Chromosome Conformation Capture technique, such as 3C [24], 4C [28,131], 5C [29], Hi-C [30], has been developed to determine spatial chromosomal interaction within a chromosome region, a chromosome, or an entire genome. Particularly, the Hi-C technique [30] is capable of capturing genome-wide chromosomal interactions (or contacts) by cross linking interacting DNA fragments, excising them out, sequencing them, and mapping them to a reference genome. The sequence reads obtained by the Hi-C technique are read pairs that reveal the chromosomal locations, or regions within spatial proximity to each other. By taking advantage of the high-throughput next generation sequencing techniques, the Hi-C technique can generate genome-wide, large-scale intra- and inter-chromosome contact data that can describe the spatial interactions within a genome. This genome description can be made at a detailed level, if a sufficiently deep sequencing of interacting DNA fragments is carried out. The recent study of the Hi-C data revealed that the local regions in a chromosome tend to have a lot more contacts within them than between them. These regions with more within-interaction are called Topologically Associated Domains (TAD). TADs are considered to be the structural and functional unit (or module) of a chromosome. According to [11], these TADs are unchanged irrespective of cell differentiation, and they also contain gene clusters that are co-regulated. In recent years, the detection of topological domain has become an important problem in bioinformatics, and computational biology, and as a result, several methods for TAD identification have been developed [142,143,155-158].

In this work, we formulate the TAD detection problem as grouping or clustering spatially interacting chromosomal regions into clusters. With this formulation, the TAD detection problem

is tackled by unsupervised machine learning (clustering) methods. The rationale is that the chromosomal fragments within the same topological domain have many more interactions between them than those between different topological domains. Therefore, the fragments within the same topological domain tend to have similar interaction profiles than those from different topological domains. Based on this insight, we developed an algorithm to group chromosomal fragments (or regions) that have similar interaction profiles into clusters, which are used for detecting TADs. To prepare a Hi-C contact matrix data as input to a clustering algorithm, we introduce a new feature representation describing the interaction profiles of a chromosomal region, which is suitable for clustering. Our method - ClusterTAD can produce fine-scale TADs that are complementary and consistent with existing methods. Moreover, this approach opens a new avenue to apply many other well-studied clustering methods developed in the machine learning, and data mining community to the relatively new TAD detection problem.

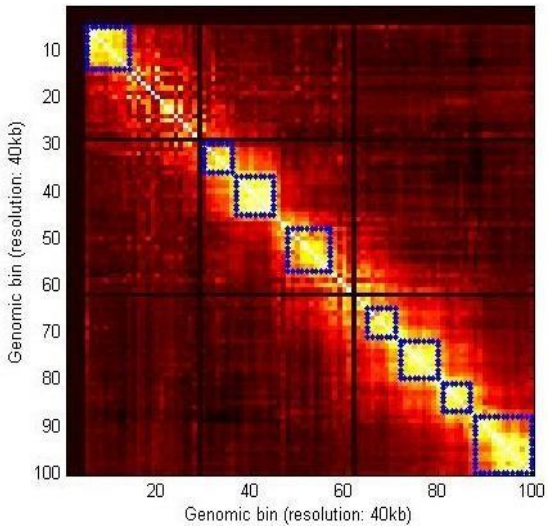
4.3 Methods

The input to our clustering-based TAD detection method (ClusterTAD) is a N by N intra-chromosomal contact matrix, M [30, 141], derived from Hi-C data, where N is the number of equal-sized regions of a chromosome. A chromosomal region is also referred to as a chromosomal bin or unit in some previous works [141,142]. The contact matrix, M, is a square matrix that represents all the observed interactions between the regions (or bins) in a chromosome. Therefore, the value of an element in the contact matrix, represented as M[i, j], records the interaction frequency between two regions (i and j) of a chromosome. As an example, Figure 4.1(a) shows the contact matrix of Chromosome 20 derived from the Hi-C data of the human embryonic stem cell (hESC) [148].

(a)

REGION	1	2	3	4	5	6	7	....	....	....	1558	1559	1560	1561
1	17.3033	60.7771	40.064	50.1865	31.2187	11.6525	16.8231	....	....	....	1.76454	1.13421	1.5427	0
2	60.7771	69.5456	57.3704	38.6014	41.4393	11.5878	12.7252	....	....	....	0	1.71961	0	0
3	40.064	57.3704	47.6947	47.9126	37.0488	9.4612	19.9679	....	....	....	0	0	0	0
4	50.1865	38.6014	47.9126	65.1259	55.4512	30.1894	32.1236	....	....	....	1.8182	0	0	0
5	31.2187	41.4393	37.0488	55.4512	63.5383	57.475	50.0471	....	....	....	0.523787	0.676754	0	0
6	11.6525	11.5878	9.4612	30.1894	57.475	0	58.6038	....	....	....	0	0	0	0
7	16.8231	12.7252	19.9679	32.1236	50.0471	58.6038	17.0688	....	....	....	2.28086	0	0	0
8	2.77917	16.8592	13.9646	21.419	28.0076	34.6818	34.1304	....	....	....	1.14815	0	0	0
9	9.79736	14.7515	8.07061	11.7756	23.3348	50.1716	51.6506	....	....	....	0	0	0	0
10	12.6001	9.55279	4.73121	11.7043	16.8228	12.8452	22.5471	....	....	....	0	0	0	0
....	....	....	....	....	....	....	....	....	....	....	....	....	....	....
....	....	....	....	....	....	....	....	....	....	....	....	....	....	....
....	....	....	....	....	....	....	....	....	....	....	....	....	....	....
1558	1.76454	0	0	1.8182	0.523787	0	2.28086	....	....	....	48.4995	39.0549	85.0608	26.6864
1559	1.13421	1.71961	0	0	0.676754	0	0	....	....	....	39.0549	31.2876	57.9716	0
1560	1.5427	0	0	0	0	0	0	....	....	....	85.0608	57.9716	0	0
1561	0	0	0	0	0	0	0	....	....	....	26.6864	0	0	0

(b)



(c)

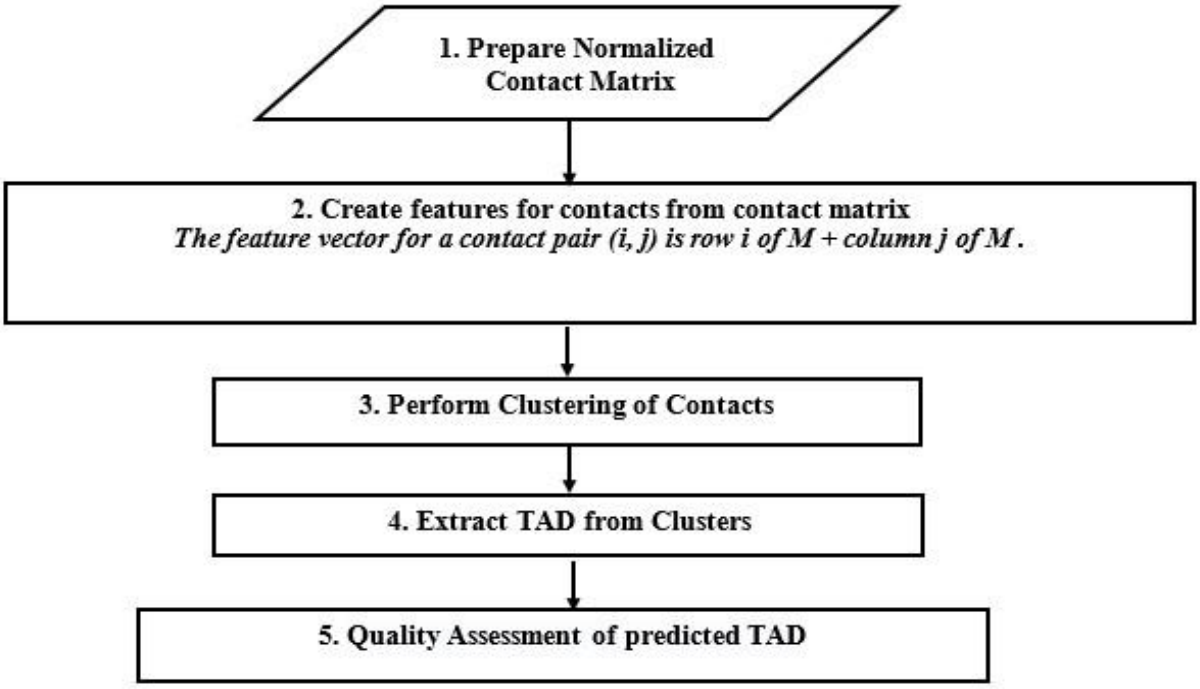


Figure 4.1. Chromosome contact matrix, TADs, and the workflow of ClusterTAD. (a) The contact matrix of Chromosome 20 of the human embryonic stem cell (hESC). The x and y-axes represent the regions of the chromosome. (b) Representation of TADs along the main diagonal of a heat map visualizing a 100 x 100 chromosomal contact matrix at 40 KB resolution. The intensity of colors represents the value of interaction frequency in the matrix. The blue squares along the main diagonal denote the identified TADs in the contact matrix. (c) The workflow of ClusterTAD.

Generally speaking, ClusterTAD takes a Hi-C data contact matrix as input, reformats the input data, and groups the contact pairs that are spatially close to each other into the same cluster. These groups are thereafter used to identify TADs. To provide a detailed clarification of the TAD detection problem, a visual representation of the TADs in a contact matrix is shown in Figure 4.1(b). The squares along the main diagonal of the contact matrix are the TAD identified for this

contact matrix. Figure 4.1(c) shows the workflow for ClusterTAD step by step. The specific steps of this workflow are described in detail below.

4.3.1 Step 1: Prepare Normalized Contact Matrices for Chromosomes

Given a Hi-C data and a specific resolution, we generate a contact matrix for each chromosome. To reduce noise and biases, a normalization method can be used to normalize the original contact counts to create a normalized contact matrix. In this work, we used the Hi-C datasets from Dixon et al [141], which had been binned at 40kb resolution, and normalized for sequencing bias using the method from Yaffe and Tanay [104].

4.3.2 Step 2: Create features for contacts in contact matrix

A key issue regarding clustering contacts into groups is determining the best way to define the informative features to represent each contact (i, j) involving two regions i, and j. In this work, we consider two pieces of information relevant to each contact (i ,j) as its features. Firstly, all the contact data on the i^{th} row in the contact matrix, M, to represent the contact profile of region i. Secondly, all the contact data on the j^{th} column of the contact matrix, M, to represent the contact profile of region j. Therefore, the feature vector for contact M [i, j] consists of 2N numbers, where N is the number of rows (or column) of the contact matrix. We used this feature representation because it includes all the contact profiles of the regions in contact; hence, making our feature informative and discriminative. Because a contact matrix is symmetric, only the contacts in the upper triangle of the contact matrix need to be considered. Since we only needed to group the regions along the main diagonal into clusters for TAD detection, we generated the features for only the contacts on the main diagonal to speed up clustering.

4.3.3 Step 3: Clustering

Once the feature generation for the contacts along the diagonal of the contact matrix is completed, a clustering method [172-174] is needed to cluster them into groups. Different types of clustering algorithms have been developed, which can be classified into the following categories: partitioning methods, hierarchical methods, model based methods, density-based methods, and grid-based methods [159]. In this work, we applied the hierarchical clustering method, Expectation-Maximization, and K-means clustering method combined with various distance metrics on a simulated Hi-C dataset. Our results in the Result Section shows that all the methods generate comparable results. To use ClusterTAD, the number of clusters, K, is the only parameter that needs to be defined. And the presumably best K value for a dataset can be estimated automatically by ClusterTAD for user's convenience (see the Results Section).

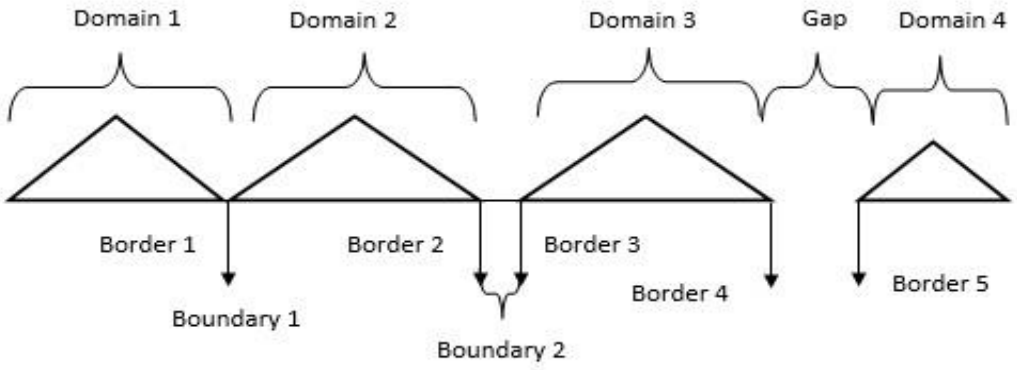
4.3.4 Step 4: Extract TAD from Contact Clusters

As shown in Figure 4.1(b), each square (TAD) highlighted on the contact matrix contains dense contacts within them, and sparse contacts between them. Therefore, a square can be considered as the cluster of contacts that have similar contact profiles. Hence, the contact clusters identified by ClusterTAD in Step 3 can be used to identify TADs.

Once the contacts on the main diagonal are assigned into clusters, we join the consecutive contacts on the main diagonal belonging to the same cluster into segments. Based on previously reported works and experimental findings [141-143,155], the minimum TD size is about 180 kb. We categorized the joined segments into three groups. The segments on the main diagonal that have zero contacts are labeled as "Gap regions". The segments greater than the minimum length are labeled as "TAD regions". The segments that have fewer than the minimum length of a TAD are

filtered out, and labelled as “Boundary regions”. Figure 4.2(a) visually explains the different types of segments defined for a dataset by ClusterTAD.

(a)



(b)

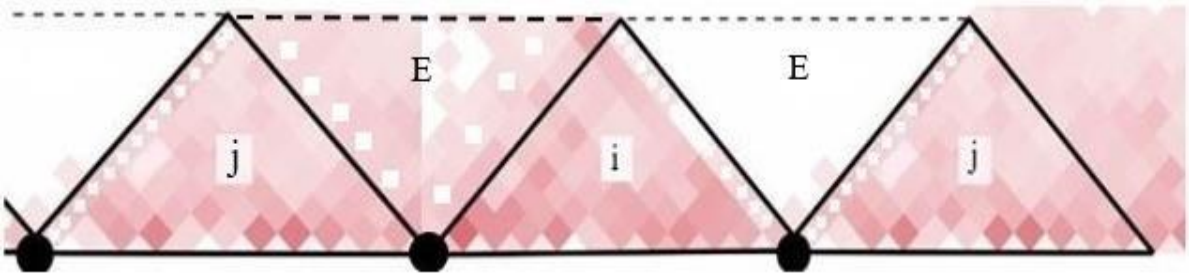


Figure 4.2. Illustration of the topologically associated domains. (a) Illustration of the basic elements related to TAD: domain, border, boundary, and gap. A domain is a TAD. A boundary is the chromosomal region between two consecutive TADs. The border marks the start/end of a domain. A gap is a point with no interaction in the contact matrix. (b) The calculation of TAD quality score. Two adjacent TADs are denoted as i and j. The area between TADs i and j that has few interactions is labeled as E. The intra(i) is the average contact frequency within a TAD (*e.g.* the area marked i). The inter(i, j) is the average contact frequency of the area marked as E. The difference of the two is the quality of TAD i.

4.3.5 Step 5: Evaluation of predicted TADs

An important characteristic of TADs is that, bins (regions) within a given TAD have similar contact frequency profiles, which are different from those of bins outside the TAD. Intuitively, maximizing the within-TAD similarity and minimizing the between-TAD similarity is important for evaluating the quality of TADs. Based on this property, we used the difference between the average of contact frequency of the bins in a TAD i, denoted as intra(i), and the average of contact frequency of the bins between TAD i and adjacent TAD j, denoted as inter (i, j) where |i-j| = 1 [14], to assess the quality of TAD assignments. This TAD quality score is represented in Equation (1) and visually represented in Figure 4.2(b).

$$\text{TAD}_i \text{ Quality} = intra(i) - inter(i,j) \quad (1)$$

Equation 1 is used to compute the quality of each TAD defined for a dataset. The overall quality score for a set of TADs defined for a contact matrix is their average quality score. Consequently, the set of TADs with the highest quality score is chosen as the representative domain set for a chromosome.

4.3.6 Datasets

The simulated dataset from Wang *et al.*, 2015 [143] is a 30-bin Hi-C contact matrix, in which the contacts were simulated from a chromosome structure with predefined topological domains. The contact matrix and the predefined domains of the simulated dataset were downloaded from [143].

The real Hi-C dataset used in this study is the Hi-C data of two mouse cells: the mouse embryonic stem cell and the cortex cell at a bin resolution of 40kb. The normalized contact matrices for these cells are available at [148].

The ChipSeq data used to analyze the enrichment of CTCF and other histone modifications is from Shen et al [160]. The raw data is available in the Gene Expression Omnibus (GEO) database with the GEO accession ID GSE29184. The extracted peaks for this ChipSeq data can be downloaded from [161].

4.4 Results and discussion

4.4.1 Determination of the parameter of ClusterTAD

ClusterTAD needs a single parameter, K (the number of clusters), to compute the set of TADs for a chromosome contact matrix. For most clustering algorithms, it is always important to find the "best" K parameter for a particular dataset, because this parameter influences the quality of the cluster analysis. However, it is worth mentioning that the definition of the "best" K parameter is usually subjective because the "right" number is often ambiguous [159]. Here, we use two well-known approaches to estimate the "best" possible value of K parameter as follows.

1) A method proposed by Han et al [159] assumes that each cluster for a dataset has about $\sqrt{2n}$ points for a dataset of n points, and the number of clusters can be estimated using Equation (2).

$$K = \sqrt{\frac{n}{2}} \qquad (2)$$

To allow some flexibility, we created a window around this estimated K value. We set the lower limit of the estimated number of clusters equal to K – 10, and upper limit equal to K + 10. We used this method as the default one for ClusterTAD for the real Hi-C data.

2) The elbow method [162, 163] is one of the oldest methods to determine the number of clusters. It chooses the number of clusters, K, such that increasing the number of clusters (K+1, K+2, …) results in no significant change in the within-cluster variance. Usually, it

starts at K = 2 and increases K with an increment of 1 to an upper limit, which is usually the number of instances in the dataset. The elbow is regarded as the point where adding another cluster does not improve the quality of clustering much. The elbow method can be computationally costly for large datasets, but extremely useful and efficient for small datasets.

4.4.2 Evaluation of the Clustering Quality

We used two different statistical evaluation measures to assess the quality of the clusters of chromosomal contacts.

(1) The Davies-Bouldin index [175] (DBI). DBI is defined as

$$\text{DBI} = \frac{1}{N} \sum_{i=1}^{N} D_i \qquad (3)$$

$$\text{where} \quad D_i = max_{j \neq i} R_{i,j} \ , \quad R_{i,j} = \frac{d_i + d_j}{d_{i,j}} \qquad (4)$$

Where d_i is the distance of elements in cluster i to its centroid. $d_{i,j}$ is the measure of the separation of clusters i, and j, equal to the distance between the centers of clusters i and j. A lower DBI score is preferred.

(2) The Silhouette Index [176] (SI). SI is defined as

$$\text{SI} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|C_i|} \sum_{j \in C_i} s_j \qquad (5)$$

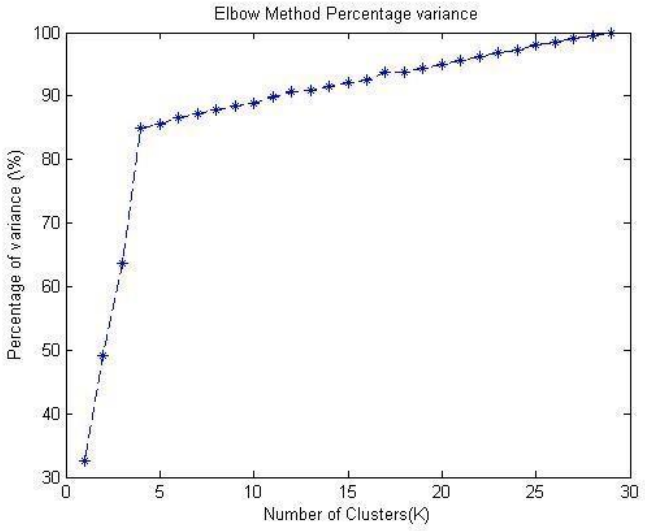
$$\text{where} \quad s_j = \frac{b_j - a_j}{max\{a_j, b_j\}} \qquad (6)$$

Where a_j is the average distance of data point j to all other data points within the same cluster (C_i). A smaller a_j value implies a better cluster assignment. b_j is the average distance of data point j to the data in the next best fit cluster for it or to another cluster with lowest average distance to j. The Silhouette coefficient value ranges between −1 and 1. A higher SI score is considered better.

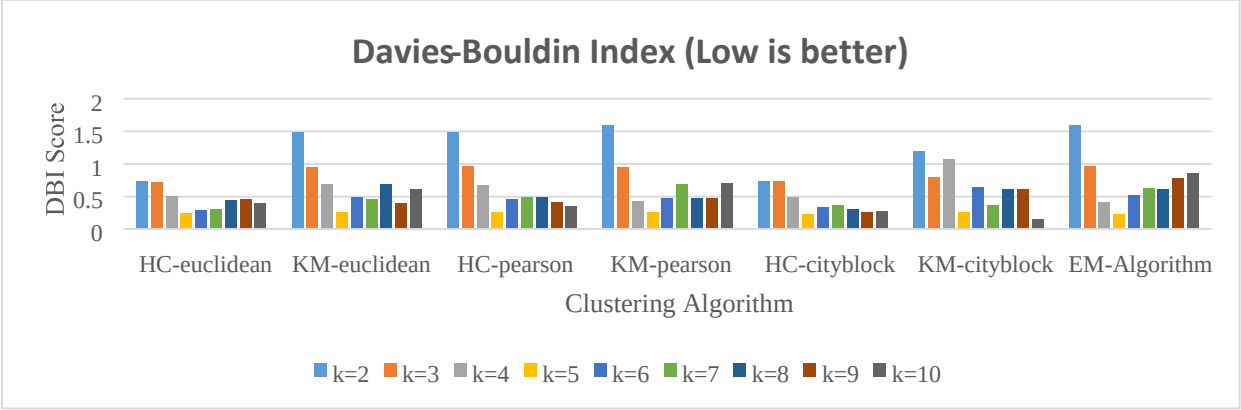
4.4.3 Assessment on the simulated dataset

We first evaluated our method on a simulated Hi-C contact matrix dataset [143]. We applied ClusterTAD on this dataset and compare its results with the known true results. We used three clustering algorithms with ClusterTAD to the dataset, including the k-means (KM) method, the hierarchical clustering (HC), and the Expectation Maximization (EM) algorithm. For the KM, and HC algorithms, we applied three distance metrics: the Euclidean-distance, the Pearson correlation distance, and the city-block distance. These algorithms require the number of cluster to be specified for them to be used. Firstly, using the Han *et al.* method, the number of clusters, K, can be estimated from the number of data points (n) in the dataset. Using Equation (2), we estimated the initial number of Cluster (K) to be 4. A window around the estimated K value specifies the range of the potential numbers of clusters to be tested in our clustering analysis. Secondly, using the elbow method, we plot the percentage of variance against the number of clusters for the dataset (Figure 4.3(a)). From the plot, we can infer that the elbow point is at 5.

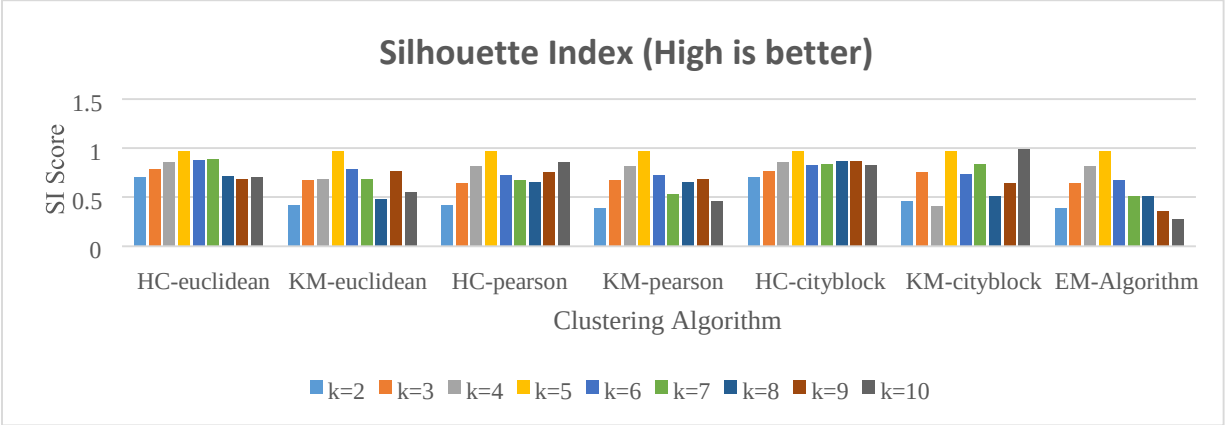
(a)


Elbow Method Percentage variance
Percentage of variance (%)
Number of Clusters(K)
100
90
80
70
60
50
40
30
0
5
10
15
20
25
30

(b)


Davies-Bouldin Index (Low is better)
DBI Score
2
1.5
1
0.5
0
HC-euclidean
KM-euclidean
HC-pearson
KM-pearson
HC-cityblock
KM-cityblock
EM-Algorithm
Clustering Algorithm
k=2
k=3
k=4
k=5
k=6
k=7
k=8
k=9
k=10

(c)


Silhouette Index (High is better)
SI Score
1.5
1
0.5
0
HC-euclidean
KM-euclidean
HC-pearson
KM-pearson
HC-cityblock
KM-cityblock
EM-Algorithm
Clustering Algorithm
k=2
k=3
k=4
k=5
k=6
k=7
k=8
k=9
k=10

(d)

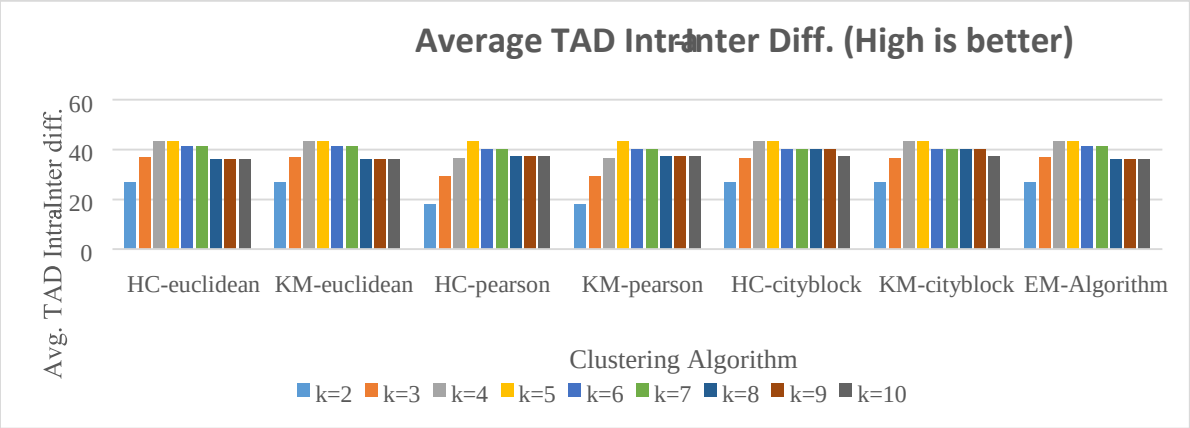


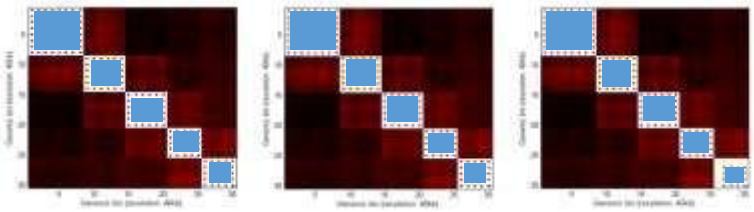
Figure 4.3. The results on the simulated dataset. (a) An elbow plot for the clustering results of ClusterTAD on the simulated dataset. The percentage of within-cluster variance is plotted against the number of clusters. The elbow point is at K = 5. (b) The Davies-Bouldin index (DBI) for the different clustering algorithms. (c) The Silhouette Index (SI) for the different clustering algorithms. (d) The average Intra-Inter difference scores for the TADs extracted by ClusterTAD with different combinations of clustering algorithms and distance metrics: HC-eulcidean, KM-eulidean, HC-pearson, KM-pearson, HC-cityblock, KM-cityblock, and the EM. HC denotes the hierarchical clustering algorithm, KM the K-means algorithm, and EM the expectation maximization algorithm. HC-euclidean represents the combination of the hierarchical clustering algorithm with Euclidean distance metric.

Once the number of clusters is defined, we performed the clustering on the simulated dataset using the three clustering algorithms above. We evaluated the quality of the clustering results using the Davies-Bouldin index (DBI) and Silhouette Index (SI). The results are shown in Figure 4.3(b, c). The best clustering quality is achieved at K = 5 for both DBI (Figure 4.3(b) and SI (Figure 4.3(c)) measures for most combinations of the algorithms and distance metrics.

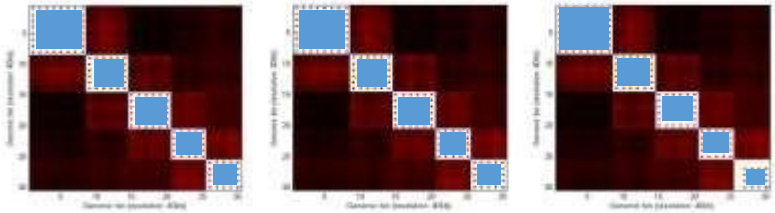
Once the clustering was done, we applied ClusterTAD to extract the TADs from the clustering results of all the algorithms, respectively. As described earlier, once the TAD is extracted, Equation (1) is used to evaluate the quality of the TADs. Figure 4.3(d) shows the Intra-Inter difference quality scores of TADs. The highest intra-inter difference was achieved with the different clustering algorithms at K = 5 regardless distance metrics used, showing the quality of TADs is consistent with that of the clustering results.

Figure 4.4(a-g) visualizes the TADs identified at K= 4 (left), K = 5 (middle) and K = 6 (right) by HC-euclidean, KM-eulidean, HC-pearson, KM-pearson, the HC-cityblock, KM-cityblock, and EM algorithm, respectively. The TADs are represented as blue squares on the contact heat maps. A TAD identified on each of the contact matrix is the blue region within the blue dots along the diagonal of the contact matrix heat map. These dots represent the boundary of the TAD, which forms squares on each of the contact matrix. Within this boundary are regions with more interactions to each other than to other areas on a contact matrix. Table 2.4.1 lists the TADs identified by each of the seven different algorithms visualized in the Figure 4.4. With this visualization, we were able to observe the consistency between the quality scores of TADs in Figure 4.3, and the true accuracy of TADs shown in Figure 4.4. The quality score is higher when the TAD result is more accurate. For instance, HC-euclidean at K = 4 and 5 in Figure 4.3(d) have the highest quality score, and their corresponding TADs are the same as the true TADs (Figure 4.4(a) left and middle). It is observed from Figure 4.4 that the seven different algorithms identify the same set of TADs when the number of clusters (K) equals to 5, which is consistent with the results in Figure 4.3 where the TADs produced by the seven algorithms have similar quality scores when K equals to 5.

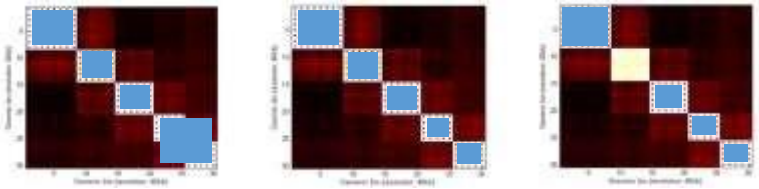
(a) K = 4 K = 5 K = 6



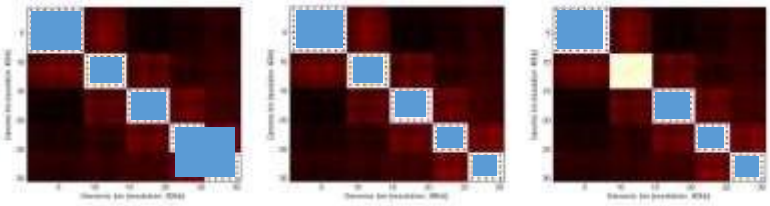
(b) K = 4 K = 5 K = 6



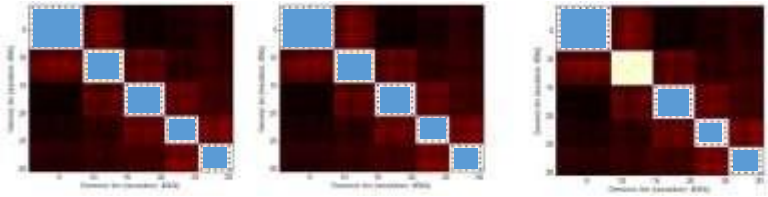
(c) K = 4 K = 5 K = 6



(d) K = 4 K = 5 K = 6



(e) K = 4 K = 5 K = 6



(f) K = 4 K = 5 K = 6

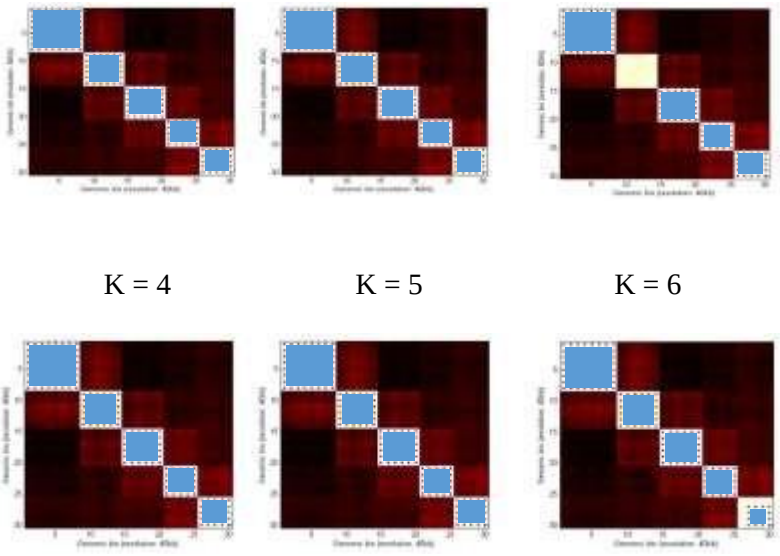


Figure 4.4. The visualization of the TADs extracted for one chromosome contact map in the simulated dataset. Rows a to g represents the TADs extracted for K= 4, K = 5 and K = 6 (from left, middle to right) for the following combinations of clustering algorithms and distance metrics: (a) HC-eulcidean, (b) KM- eulidean, (c) HC-pearson, (d) KM-pearson, (e) HC-cityblock, (f) KM-cityblock, and (g) EM. HC denotes the hierarchical clustering algorithm, KM the K-means algorithm, and EM the expectation maximization algorithm. HC-euclidean denotes the combination of the hierarchical clustering algorithm with the Euclidean distance metric. The left column visualizes the TADs extracted by the seven algorithms when K=4, the middle columns the TADs extracted when K=5, and the right column the TADs extracted when K=6. A TAD region identified on each contact heatmap is denoted by a blue square within the blue dots along its diagonal. The blue dots represent the boundary of a TAD region. The white squares along the diagonals are unrecognized TADs.

Table 4.1. The lists of TADs identified by the seven different algorithms in Figure 4.4. The table contains the lists of TADs extracted for K= 4, K = 5 and K = 6 (from left, middle to right) by the seven algorithms: (a) HC-eulcidean, (b) KM-eulidean, (c) HC-pearson, (d) KM-pearson, (e) HC-

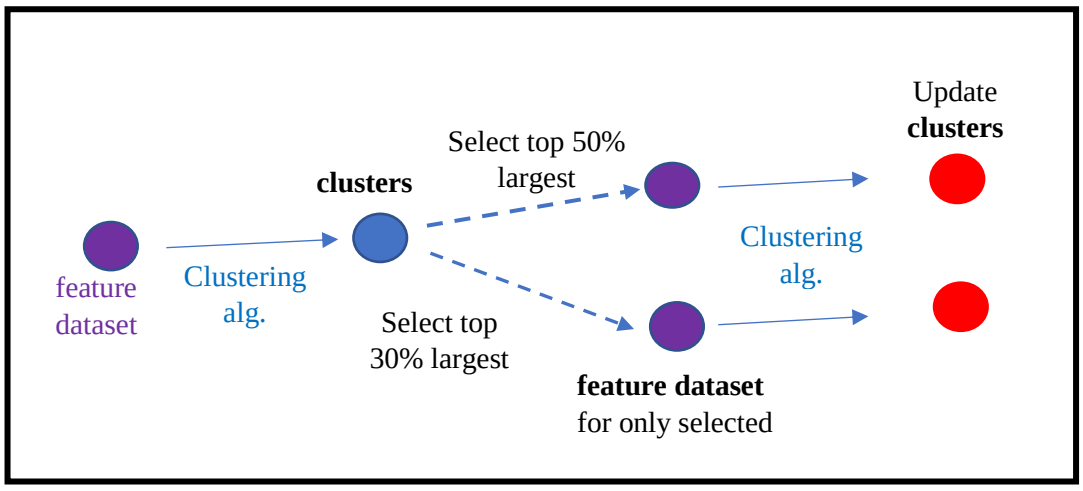
cityblock, (f) KM-cityblock, and (g) EM. HC denotes the hierarchical clustering algorithm, KM the K-means algorithm, and EM the expectation maximization algorithm. HC-euclidean denotes the combination of the hierarchical clustering algorithm and the Euclidean distance metric. A TAD is represented as {start, end}, where "start" is the TAD start region, and "end" is the TAD end region. The best TAD set for the synthetic data is {(1,8), (9,14), (15,20), (21,25), and (26,30)}.

Algorithm	**K = 4**	**K = 5**	**K = 6**
a	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (9,14), (15,20), (21,25), and (27,30)}.
b	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (9,14), (15,20), (21,25), and (27,30)}.
c	{(1,8), (9,14), (15,20), and (21,30)}.	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (15,20), (21,25), and (26,30)}.
d	{(1,8), (9,14), (15,20), and (21,30)}.	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (15,20), (21,25), and (26,30)}.
e	{{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (15,20), (21,25), and (26,30)}.
f	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (15,20), (21,25), and (26,30)}.
g	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (9,14), (15,20), (21,25), and (26,30)}.	{(1,8), (9,14), (15,20), (21,25), and (27,30)}.

4.4.4 Assessment of ClusterTAD on real Hi-C datasets

We tested ClusterTAD on the Hi-C data of two mouse cells: the mouse embryonic stem cell and the mouse cortex cell at a bin resolution of 40kb. We used the K-means algorithm with Euclidean distance metric for the clustering performed on the real Hi-C datasets. The first round of the application of ClusterTAD resulted in large, coarse clusters, and consequently large TADs. As illustrated in [142-143,155] that large TADs often have lower average interactions within TADs, in order to improve cohesiveness of TADs, we applied another round of clustering to large clusters generated in the first round. Figure 4.5(a) shows the workflow of multiple steps of clustering with ClusterTAD. Re-clustering of the existing clusters generates sub-clusters. To identify the set of clusters to be re-clustered from the results of the first round of clustering (ClusterTAD_1), we ranked the clusters generated from ClusterTAD_1 based on the number of points (regions) in each cluster. Then we selected the top 30% or 50% largest clusters for re-clustering with the same algorithm of ClusterTAD, such that at least 50% of clusters in the current round will be kept. The second round of clustering is denoted as ClusterTAD_2. The third and also last round of clustering operation is called ClusterTAD_3.

(a)



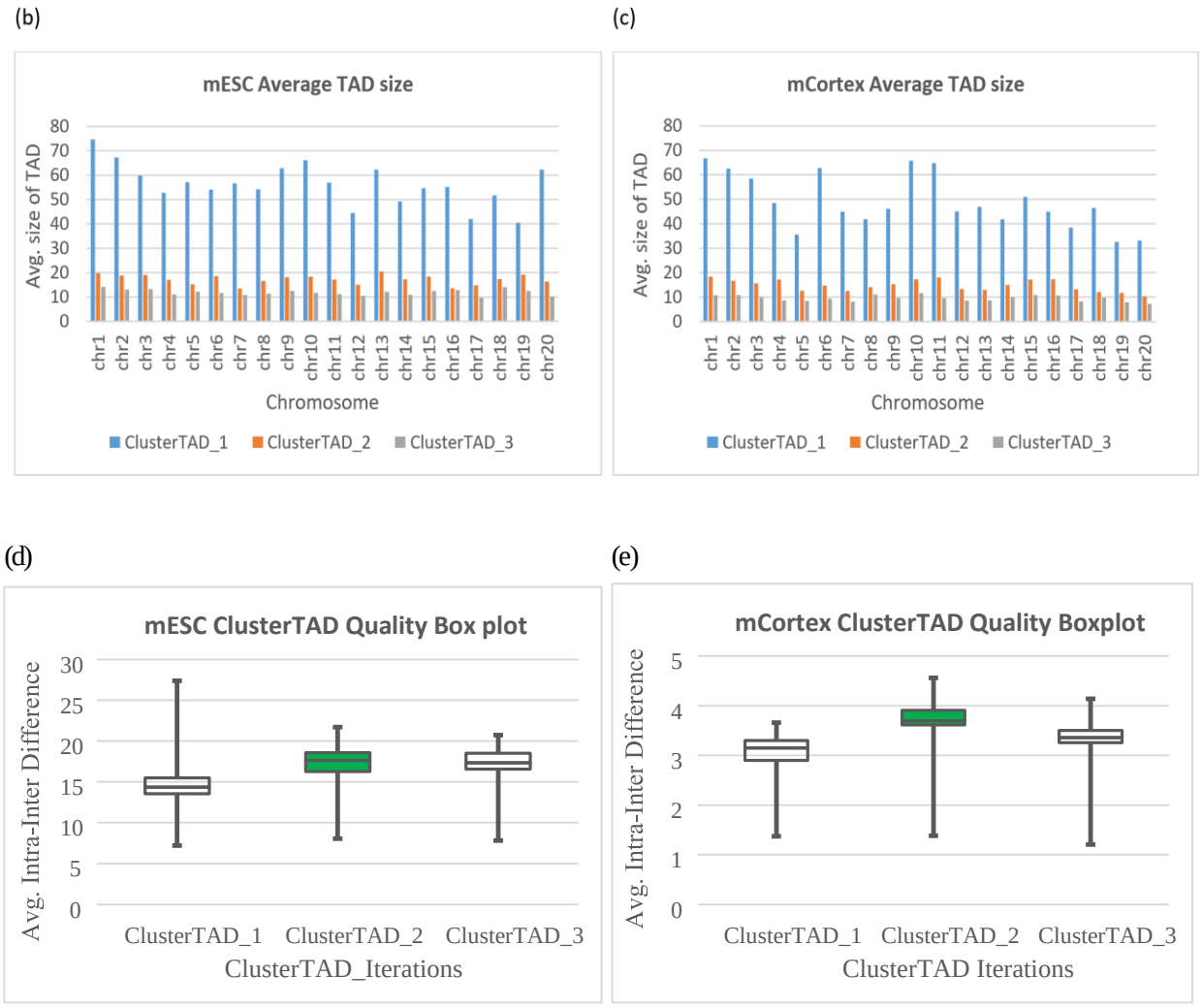
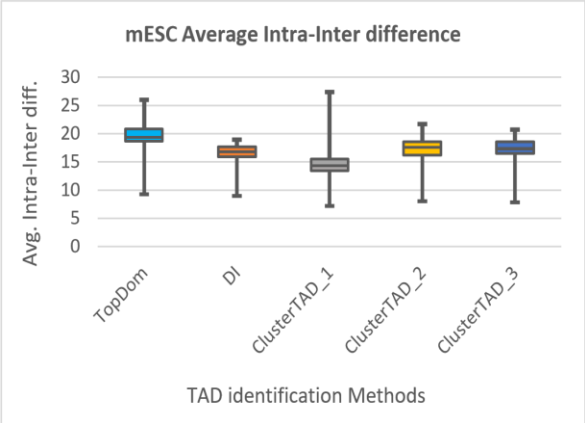


Figure 4.5. Evaluation on a real Hi-C dataset. (a) The workflow of the iterative application of ClusterTAD. (b) The average size of TADs identified for the mouse embryonic stem cell by three rounds of clustering of ClusterTAD (ClusterTAD_1, ClusterTAD_2, and ClusterTAD_3). (b) The average size of TADs identified for the mouse cortex cell by three rounds of clustering of ClusterTAD. (d) The box plot of the quality scores of TADs extracted for the mouse embryonic stem cell by the three rounds of clustering of ClusterTAD. (e) The box plot of the quality scores of TADs extracted for the mouse Cortex cell for the different clustering operations performed by ClusterTAD.

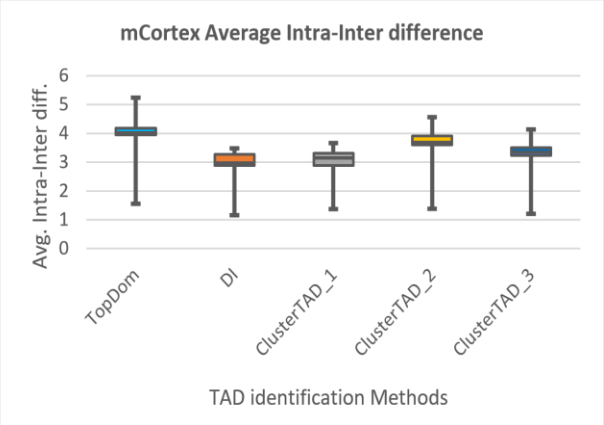
Figure 4.5(b, c) shows the average size of TADs generated in the three rounds of clustering. The average size of TADs decreases from one round to next round as expected. Figure 4.5(d, e) reports the inter-intra interaction frequency scores of TADs of the three rounds. ClusterTAD_2 consistently achieved the highest average score. Though ClusterTAD_3 has smaller TADs than ClusterTAD_2, its quality score is lower than ClusterTAD_2.

We compared ClusterTAD with the two other widely used methods: the directionality index (DI) method [141] and the TopDom [155] methods on the mouse Hi-C datasets. The results of DI and TopDom were obtained from their published data. Figure 4.6 shows the quality scores of TADs, the number of TADs, and the average size of TADs of the three methods. Generally speaking, DI detects TADs of larger sizes, TopDom identifies TADs of smaller size, and ClusterTAD produces the results in the middle. Figure 4.6(e) and 4.6(f) shows the average size of TADs identified by TopDom, DI, and ClusterTAD for the mESC, and mCortex cells respectively. The average size of the TADs produced by ClusterTAD is significantly smaller than DI, but somewhat larger than TopDom (Figure 4.6(e)) or comparable to it (Figure 4.6(f)). This is consistent with the observation that DI tends to detect TAD with large sizes, while TopDom tends to identify smaller TADs called sub-TADs. Since ClusterTAD tends to break larger TADs into smaller TADs to improve their cohesiveness, the average size of TADs identified by ClusterTAD is between DI and TopDom, while leaning more toward TopDom. Since the TADs identified by ClusterTAD and TopDom have a smaller size, they tend to have higher inter-intra interaction frequency scores.

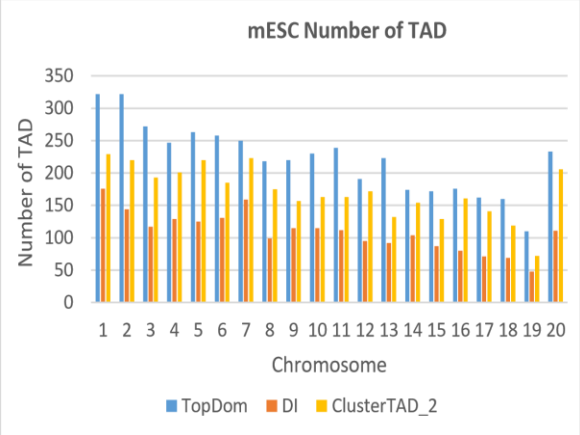
(a)


mESC Average Intra-Inter difference
Avg. Intra-Inter diff.
30
25
20
15
10
5
0
TopDom
DI
ClusterTAD_1
ClusterTAD_2
ClusterTAD_3
TAD identification Methods

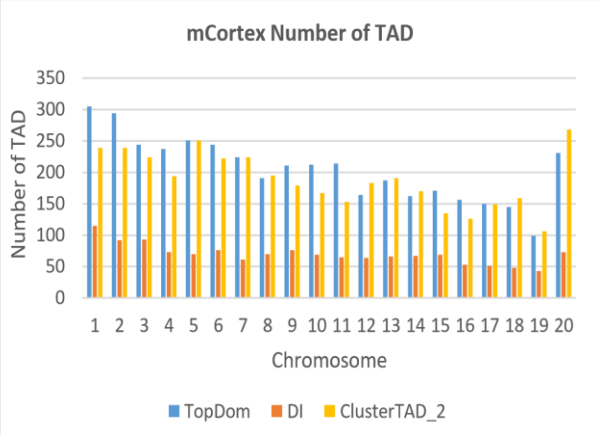
(b)


mCortex Average Intra-Inter difference
Avg. Intra-Inter diff.
6
5
4
3
2
1
0
TopDom
DI
ClusterTAD_1
ClusterTAD_2
ClusterTAD_3
TAD identification Methods

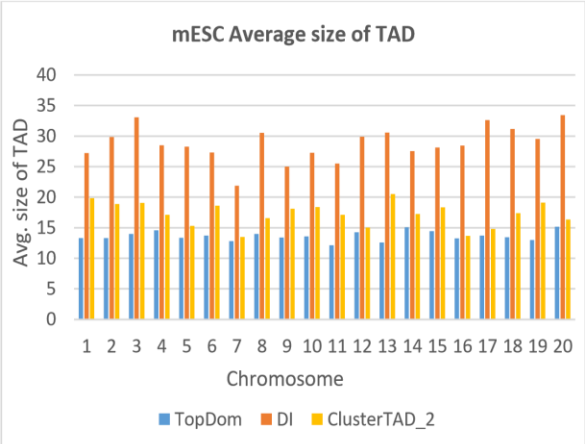
(c)


mESC Number of TAD
Number of TAD
350
300
250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Chromosome
TopDom
DI
ClusterTAD_2

(d)


mCortex Number of TAD
Number of TAD
350
300
250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Chromosome
TopDom
DI
ClusterTAD_2

(e)


mESC Average size of TAD
Avg. size of TAD
40
35
30
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Chromosome
TopDom
DI
ClusterTAD_2

(f)

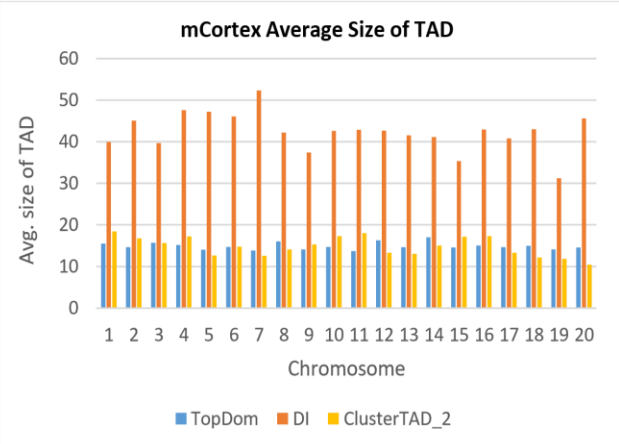
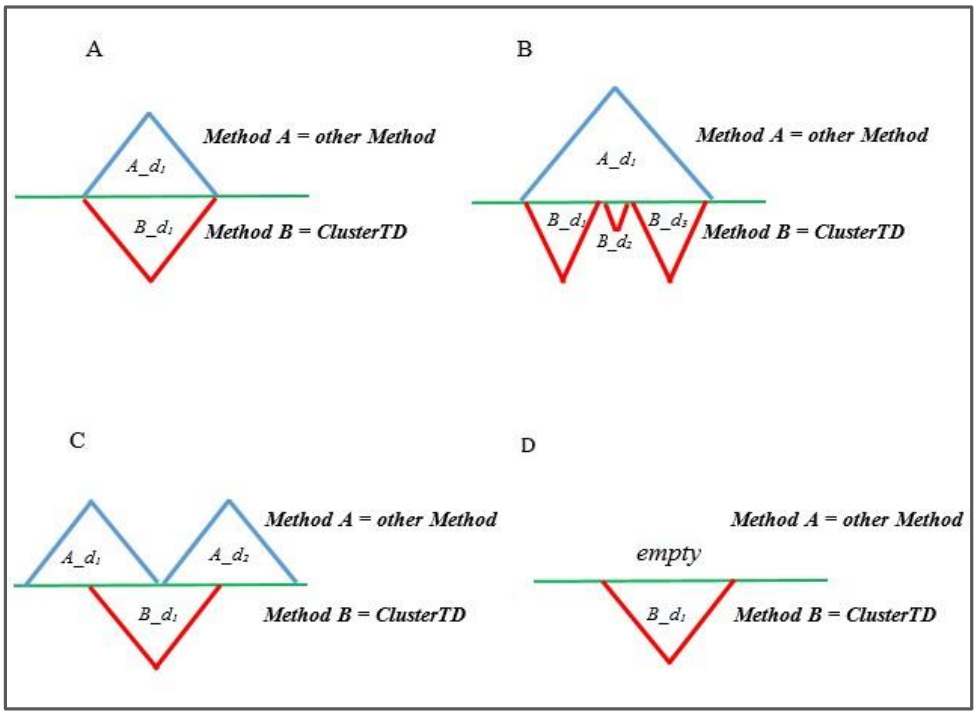

mCortex Average Size of TAD
Avg. size of TAD
60
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Chromosome
TopDom
DI
ClusterTAD_2

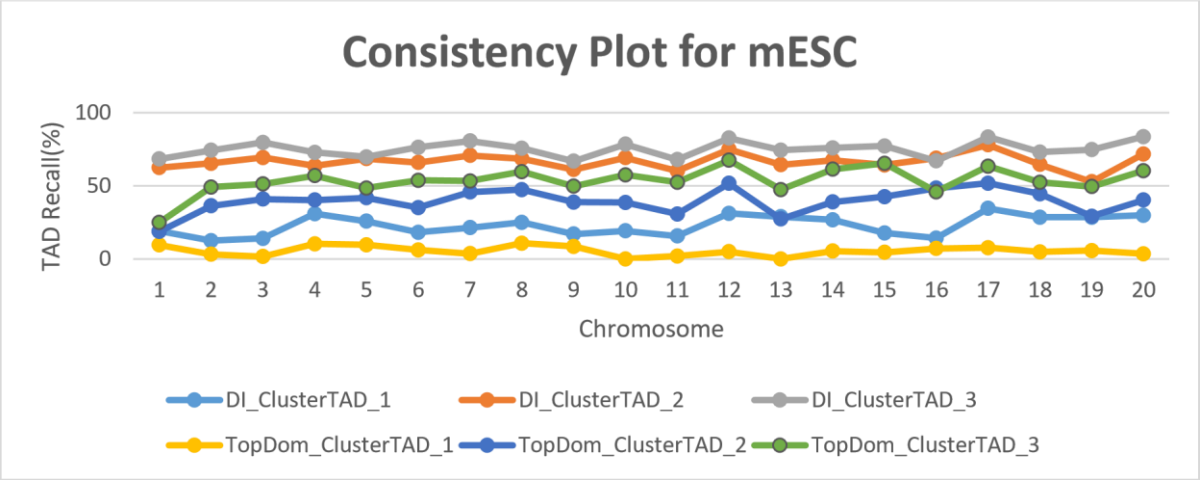
Figure 4.6. Comparison of the quality scores, numbers and average sizes of TADs identified by TopDom, DI, and ClusterTAD on two mouse cell lines. (a, b): The comparison of the intra-inter difference scores; (c, d): the number of TADs, and (e, f) the average size of TADs for the mESC and mCortex cells respectively.

We assessed how consistent the TADs detected by ClusterTAD are with those by DI and TopDom. The consistency check was carried out according to the method described in Figure 4.7(a). A TAD detected by method A is considered also detected by method B if the similarity between the TADs by method A and the TADs by method B falls in Case A or Case B in Figure 4.7(a). Figure 4.7(b, c) shows the percentage of TADs detected by ClusterTAD that were also detected by the other methods. A higher percentage of TADs identified by ClusterTAD was found by DI than by TopDom probably because the TADs predicted by TopDom were generally smaller. Overall, the three methods appear to produce the complementary results on the dataset.

(a)


A
Method A = other Method
A_d1
B_d1
Method B = ClusterTD
B
Method A = other Method
A_d1
B_d1
B_d2
B_d3
Method B = ClusterTD
C
Method A = other Method
A_d1
A_d2
B_d1
Method B = ClusterTD
D
Method A = other Method
empty
B_d1
Method B = ClusterTD

(b)


Consistency Plot for mESC
TAD Recall(%)
100
50
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Chromosome
DI_ClusterTAD_1
DI_ClusterTAD_2
DI_ClusterTAD_3
TopDom_ClusterTAD_1
TopDom_ClusterTAD_2
TopDom_ClusterTAD_3

(c)

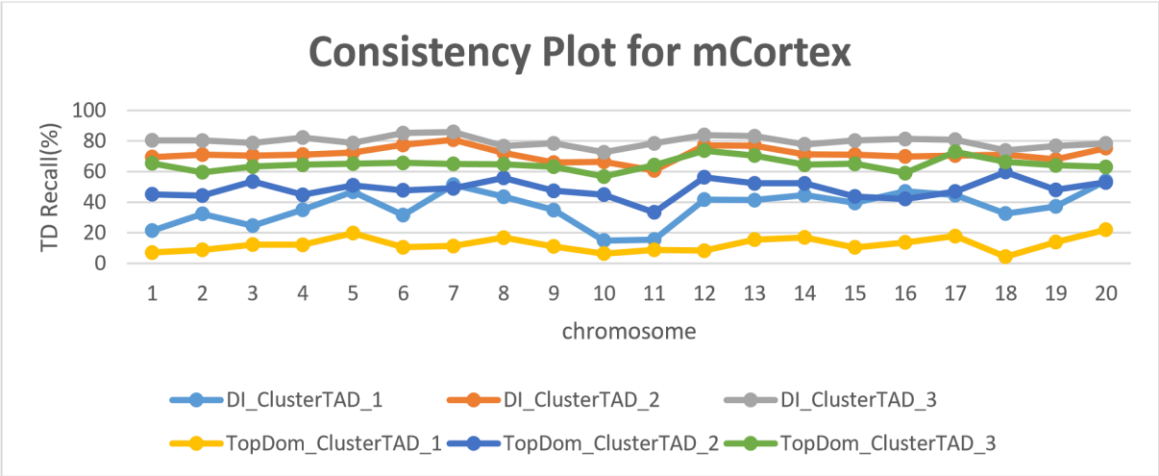


Figure 4.7. The analysis of the consistency between TADs identified by ClusterTAD and other methods on the two mouse cell lines. (a) Four different cases in which TADs detected by two different methods are compared with each other. **Case A**: This refers to the case in which the TAD identified in method B exactly matches those from another method A. The TADs detected by the two methods have the same boundaries. **Case B**: This refers to the case in which a TAD detected by method A contains two or more domains detected by method B. The smaller TADs detected by method B are called sub-TAD of the TAD detected by method A. **Case C**: This represents the conflicting case in which the domain detected by method A does not match or contain the domains detected by method B even though there is some overlap between them. **Case D**: This refers to the rare case in which the region is not assigned to a TAD by method A, but is assigned by a TAD by method B. (b) The percentage of TADs detected by ClusterTAD for the mESC cell line that were also detected by TopDom and DI. (c) The percentage of TADs detected by ClusterTAD for the mCortex cell line that were also detected by TopDom and DI.

4.4.5 Validation of ClusterTAD by the enrichment analysis of CTCF binding sites and histone modification marks in domain boundaries

Topologically Associated Domains (TADs) are known to have a high level of interactions within them, compared to those between them. Each domain is separated from each other by domain boundaries. Therefore, TAD boundaries can be regarded as an insulator that restricts interaction between a TAD and its adjacent TADs [141, 164]. And TAD boundaries are also known to have an enrichment of binding sites of CTCF – a genome architectural protein [156-158, 164, 165-168]. The binding sites of CTCF can be determined by a chromatin immunoprecipitation (ChIP) sequencing (ChIP-Seq) technique. We validated the result obtained from ClusterTAD by checking the enrichment of CTCF at the boundary between TADs for each of the mouse cells.

We used the dataset of the predicted cis-regulatory elements extracted from Chip-Seq data by Shen et al [160] to assess the abundance of CTCF binding sites at the domain boundaries of TADs. Though CTCF binding sites are largely found at domain boundaries, CTCF are also associated with some active histone modification to form the insulation in the domain boundaries. Hence, in addition to studying the CTCF enrichment in the boundaries, we also investigated the enrichment of promoter marks: RNA Polymerase II and H3K4me3, and enhancer-marks (H3K4me1 and H3K27ac). Using the Chip-Seq data, the peaks for the CTCF and histone modification marks were identified using MACS [169] with the default parameters and filtered by a p-value of 0.00001. Figure 2.8 shows the occurrence of high number of peaks (enrichment) for CTCF binding sites, and the histone modification marks at the boundaries of TADs identified for the two mouse cells by ClusterTAD, DI and TopDom, validating that the domain boundaries recognized by ClusterTAD are biologically relevant. According to the enrichment analysis in Figure 2.8, there was a reduction in the average number of peaks for the enhancer mark H3K27ac in the mouse

cortex cells than in the mESC cells, which is consistent with the previous discovery in [155]. In addition, the H3K4me1 peak enrichment in the mCortex cells was slightly higher than in the mESC cells. The enrichment of CTCF, H3K27ac, and H3K4me1 in the predicted TAD boundaries suggests that they may act as an insulator to separate TADs [141, 164]. The previous studies show that enhancers could activate transcription by bringing accessory transcription-related factors to gene promoters within their spatial proximity [170], even though the promoters may be sequentially far away from the enhancers in the linear genome sequence [171]. Hence, the high enrichment of the enhancer and promoter marks in the boundary regions suggests that some TAD boundary regions can be transcription activation sites.

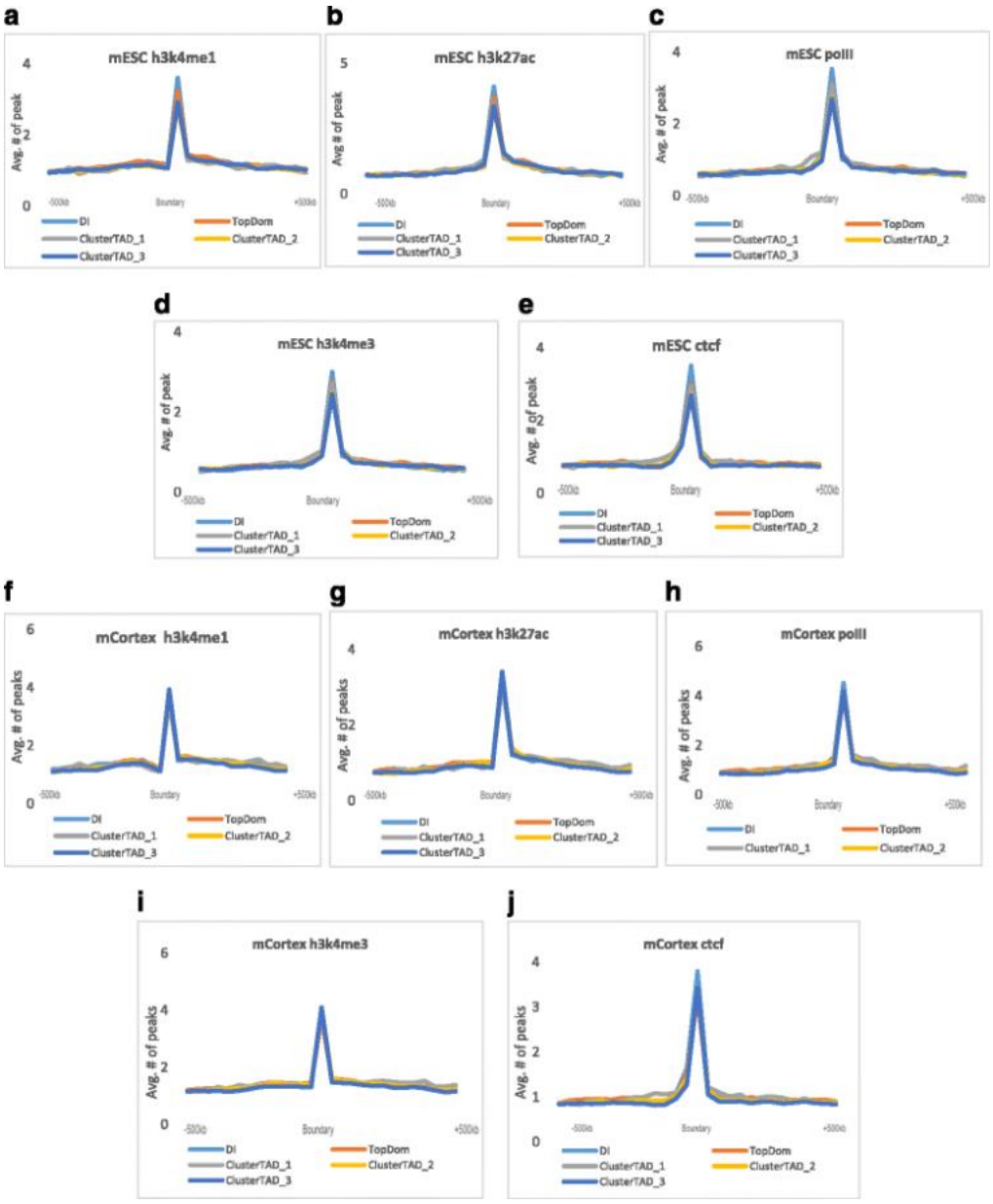


Figure 4.8. The enrichment analysis of active histone modification marks and CTCF binding sites at the domain boundary.

The average peak number of active histone modification marks (promoter marks (Polymerase II and H3K4me3) and enhancer marks (H3K4me1 and H3K27ac) and CTCF binding sites at the boundary regions identified by TopDom, DI and ClusterTAD for mouse Embryonic Stem Cell line (mESC) (a-e) and the mouse cortex cell line (mCortex) (f-j).

4.5 Conclusions

We introduce ClusterTAD, a new clustering-based method, to detect TADs from Hi-C data. ClusterTAD employs standard clustering algorithms to extract topological domains from Hi-C contact data. We show that ClusterTAD is consistent and complementary with existing methods. The TAD boundaries identified by ClusterTAD are validated by the enrichment analysis of CTCF binding sites and histone modification marks. It is easy to use ClusterTAD since it only requires one parameter – the number of cluster, and the parameter can be estimated automatically from the data. Moreover, ClusterTAD can be iteratively applied to divide larger clusters into small ones, which can be used to identify both large TADs and smaller sub-TADs. Finally, by formulating the TAD detection problem as a classic clustering problem through a novel representation of chromosomal contacts, an array of clustering methods in the field of machine learning can be applied to address the problem. We expect more sophisticated clustering algorithms will be used to improve TAD detection in the future.

5 GenomeFlow: A Comprehensive Graphical Tool for Modeling and Analyzing 3D Genome Structure

5.1 Abstract

Three-dimensional (3D) genome organization plays important functional roles in cells. User-friendly tools for reconstructing 3D genome models from chromosomal conformation capturing data and analyzing them are needed for the study of 3D genome organization. We built a comprehensive graphical tool (GenomeFlow) to facilitate the entire process of modeling and analysis of 3D genome organization. This process includes the mapping of Hi-C data to one-dimensional (1D) reference genomes, the generation, normalization and visualization of two-dimensional (2D) chromosomal contact maps, the reconstruction and the visualization of the 3D models of chromosome and genome, the analysis of 3D models, and the integration of these models with functional genomics data. This graphical tool is the first of its kind in reconstructing, storing, analyzing and annotating 3D genome models. It can reconstruct 3D genome models from Hi-C data and visualize them in real-time. This tool also allows users to overlay gene annotation, gene expression data and genome methylation data on top of 3D genome models. The source code and user manual: https://github.com/jianlin-cheng/GenomeFlow

5.2 Introduction

The three-dimensional genome organization is important for cellular function [30,144]. Chromosome Conformation Capture techniques like Hi-C [30] have enabled the study of 3D genome organization in high resolution and with high throughput. Graphical tools such as [41] have been developed to process and analyze Hi-C data. Several algorithms have been proposed to reconstruct 3D genome models from Hi-C data[76-79]. But there is no graphical tool with an easy user interface to build, analyze 3D genome models and integrate modeling with functional

genomics data. To fill the gap, we built a comprehensive tool, GenomeFlow, for processing Hi-C data, reconstructing and analyzing 3D genome models, and integrating 3D models with functional genomics data.

5.3 Function

The function of GenomeFlow is organized in three categories: 1D function, 2D function, and 3D function. 1D function allows users to map raw Hi-C pair-end reads to a reference genome to identify chromosomal contacts. 2D function is used to create, normalize and visualize contact matrices. 3D function is for reconstructing and analyzing 3D models. Figure 5.1-5.3 shows four typical 2D and 3D functional features: visualization of contact matrix and topologically associating domains (TADs), 3D model reconstruction, chromatin loop identification, and model annotation. The most important function features of GenomeFlow are described below.

5.3.1 1D Functions

5.3.1.1 Index reference genome and Map Hi-C reads to 1D genome sequence

GenomeFlow has a function to create an index for a specific specie reference genome. In addition, GenomeFlow allows users to map raw pair-end Hi-C reads to a reference genome to generate chromosomal contacts. It also allows users to filter, and convert mapped single, or pair-end reads to an easy to read text format. GenomeFlow calls an external genome mapping tool such as Bowtie2 [178] or Bwa [179] to perform the indexing and mapping operations. Thereafter, it stores the mapped data file in a text file format, which can be used by the 2D function to generate contact matrices. GenomeFlow allows the user to perform these tasks step by step or in one go with our 1D function, HiC-Express. It is worth noting that this 1D function is optional since users can load their already mapped files into GenomeFlow to use its 2D and 3D function below.

5.3.2 2D Functions

5.3.2.1 Conversion of mapped reads to 2D contact matrices

GenomeFlow provides a function to convert a mapped Hi-C reads from the text format into a compressed file containing chromosomal contact matrices in the binary hic format [41]. The function normalizes chromosomal contacts with Knight-Ruiz matrix balancing normalization [144] and Vanilla-Coverage normalization [30]. It also provides options to create contact matrices at specific resolutions and for specific chromosomes. Users can specify a contact threshold or a mapping quality (MAPQ) score threshold. The function is an extension of the Pre function of Juicer [41]. Moreover, GenomeFlow provides a graphical user interface (GUI) to validate input and makes it easy for users to use the function.

5.3.2.2 Extracting 2D contact matrices from a compressed contact file

A binary contact file in hic format can contain several contact matrices at different resolutions, each of which are normalized by different normalization methods. It is often large and not human readable. GenomeFlow can read the header of the file first to display information about the genome version, chromosomes, resolutions and normalization methods. Users can then choose the contact matrix of a chromosome at a resolution and normalization method of choice to be exported to a text file in the sparse matrix format. Users can also choose to export a contact matrix of a fragment of a chromosome.

5.3.2.3 Normalization, visualization and analysis of 2D contacts

GenomeFlow provides a function to normalize contact matrices in sparse matrix format using the ICE normalization [101]. GenomeFlow can visualize a contact matrix as a heat map, where numeric values in the input contact matrix are represented as colors according to a selected color gradient (Figure 5.1**)**.

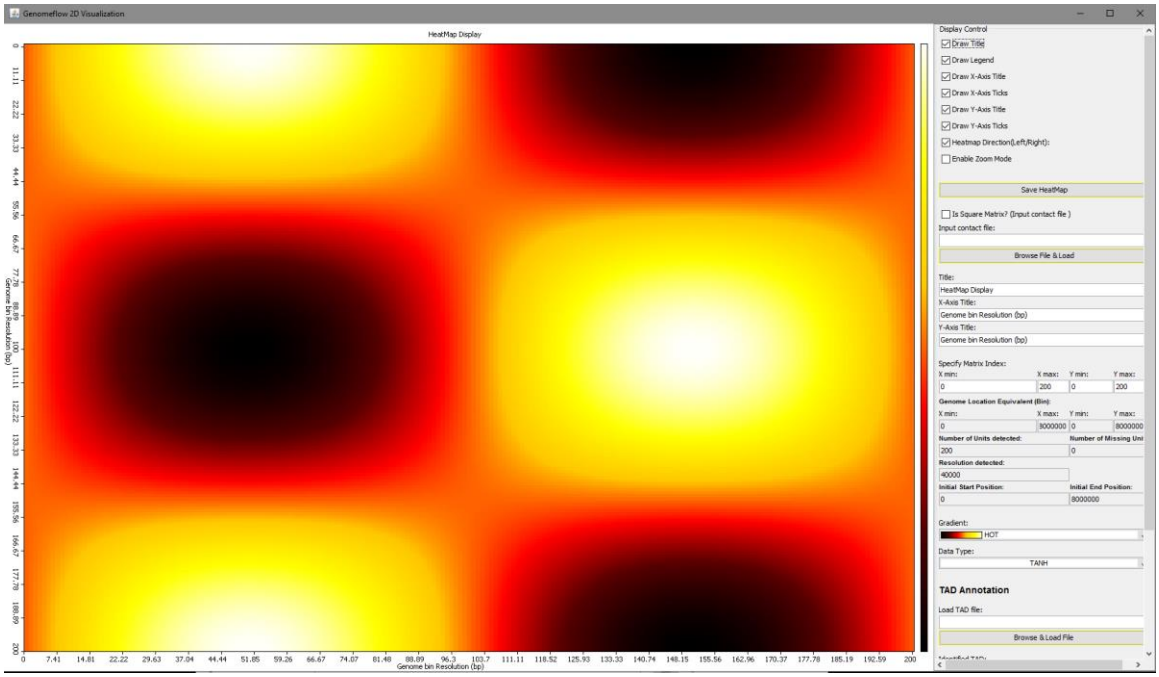


Figure 5.1. Visualization of Hi-C dataset in 2D Format

5.3.2.4 Identification of topological association domains from 2D contact matrices

GenomeFlow uses the ClusterTAD [177] algorithm to identify topological associated domains (TADs) from chromosomal contact matrices. Once the TADs of a contact matrix have been identified, they can be visualized on the heatmap of the contact matrix (**Figure 5.2)**.

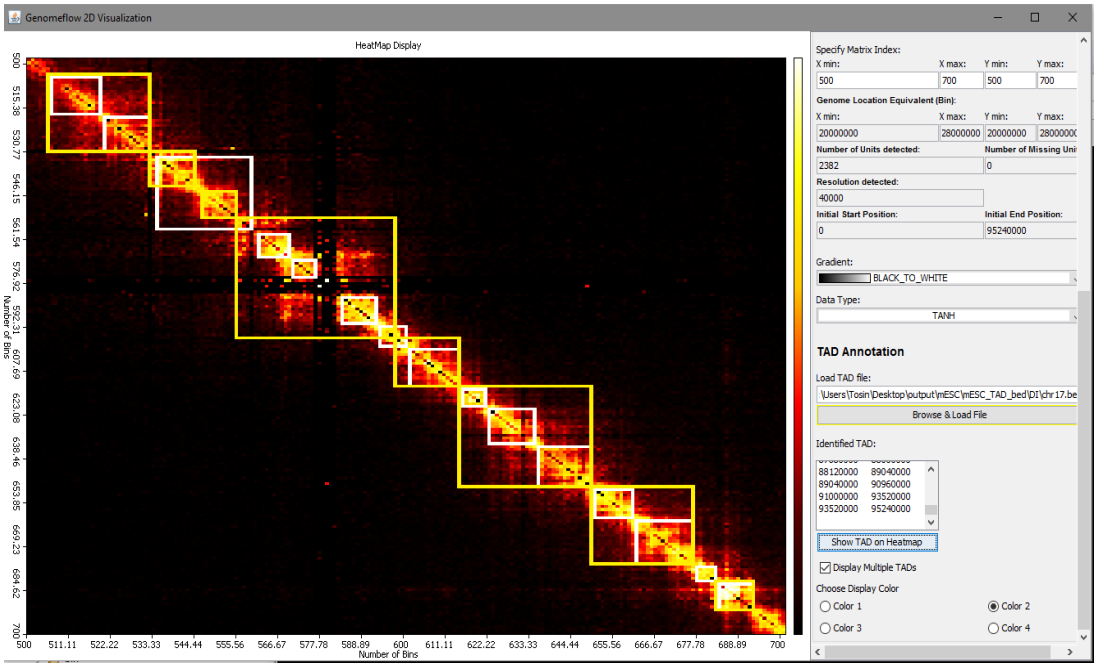


Figure 5.2. Demonstration of TAD Annotation on a 2D Heatmap. The yellow and white squares show the annotation of TADs identified by two different methods on a heatmap.

5.3.3 3D Functions

5.3.3.1 Reconstruction of 3D genome model in real time

GenomeFlow implements two 3D genome reconstruction algorithms (LorDG [69] and 3DMax [70]) to reconstruct 3D genome models from contact matrices. Both functions have user-friendly GUI and visualize how models are being reconstructed in real time. The input of the function is a contact matrix in sparse matrix format, such as one extracted from a the hic format file. Figure 5.3 shows the GUI of reconstructing 3D models. The output 3D models are stored in the GSS format files [180] that contain both *x*, *y*, *z* coordinates and genomic locations of *loci*, chromosome number, and genome version. There is currently no file format specially designed for 3D genome models other than the Genome Scale System (GSS) format. Compared to the PDB format for storing

protein structures, the GSS format can store much larger structures of a chromosome or genome in high resolution and can include extra genomic information needed for function analysis. GenomeFlow can visualize and analyze 3D genome models in GSS format *and* integrate them with other genomics data such as gene expression and methylation data.

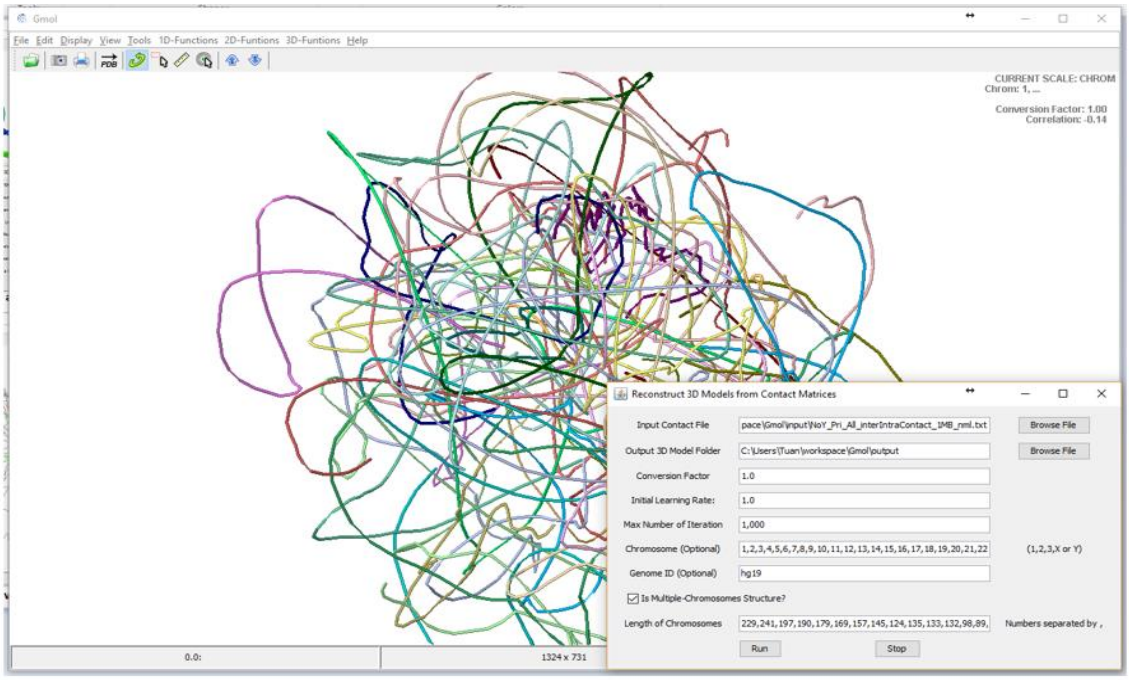


Figure 5.3. 3D model Structure reconstruction in real time

5.4 Conclusion

We introduced a comprehensive software tool for processing Hi-C data and reconstructing and analyzing 3D genome models. It provides a user-friendly GUI to carry out many steps of analysis and modeling of 3D genome conformation. Users without prior knowledge of the 3D genome can use the tool to build and analyze a 3D genome in their research and work.

5.5 An example user case, using GenomeFlow for TAD Annotation

First, a user needs to identify the Topological Associated domains (TAD) in their contact matrix.

A user can do this through the function “Identify TAD” provided in the 2D Functions menu in GenomeFlow. Once the user clicks on the “Identify TAD” function, the Figure 5.4 is displayed. The necessary information for each of the fields in Figure 5.4 is highlighted in Table 5.1. The algorithm used for the TAD identification is ClusterTAD [177].

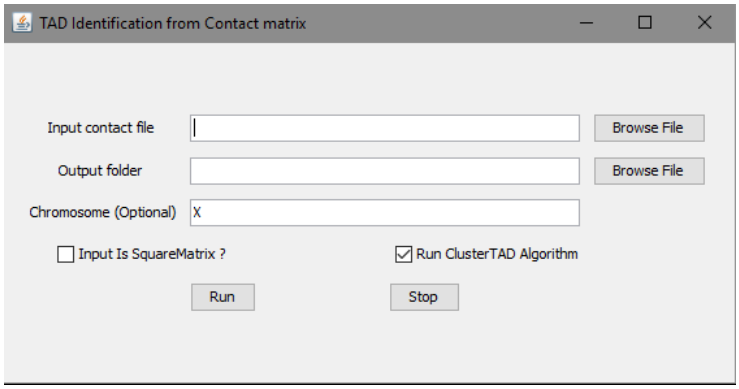


Figure 5.4. Identifying TADs on a contact matrix

Table 5.1. Description of the required information for TAD identification window in GenomeFlow

Field	Description
Input contact file	An input file in any of the format described above
Output folder	Directory to output the comparison report
Is SquareMatrix?(Input contact file)	Allows the user to specify if the input is a Square matrix (a full matrix) or a sparse matrix. If checked, it displays a textbox for the user to specify the matrix resolution.
Data Resolution	It is visible only if *Is SquareMatrix?* is checked. It allows user specify resolution for the input matrix.

Chromosome (optional)	Allows user to specify the chromosome data
Run ClusterTAD Algorithm	The default algorithm used for TAD identification from the input contact Matrix
Run	To start the identification process. A progress bar is displayed to show the steps taken by the TAD identification algorithm.
Stop	During the identification, if this button is pressed, the program will stop.

Once the TADs in the contact matrix have been identified, the user can move to the next step of annotating the TADs on the contact matrix. Launch the GenomeFlow 2D visualization window, by clicking on GenomeFlow function, “Visualize Dataset” (Figure 5.5). If the input matrix is a N x N square matrix, the user needs to specify the Hi-C contact matrix resolution and load the contact matrix by clicking on the “Browse File & Load” button on the Display Control window on the right (Figure 5.6).

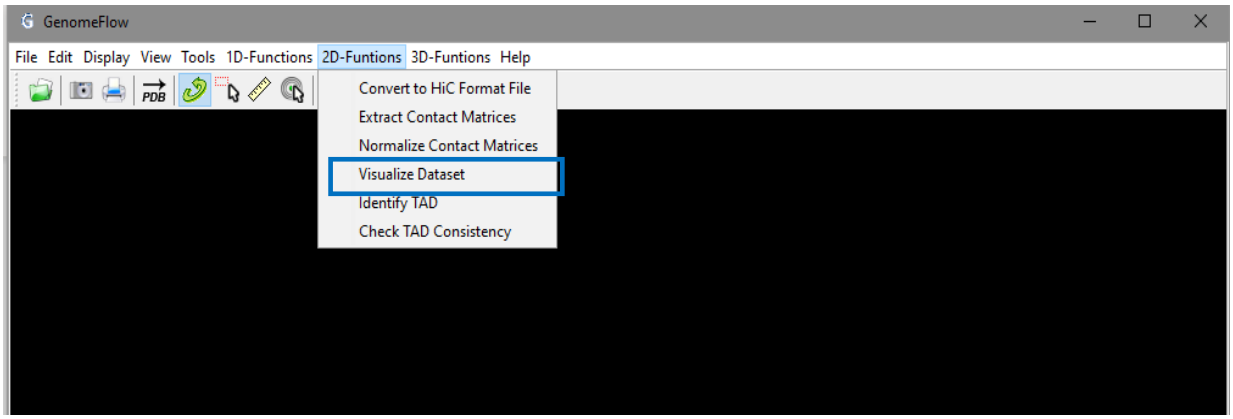


Figure 5.5. Demonstrating how to display the GenomeFlow 2D visualization window.

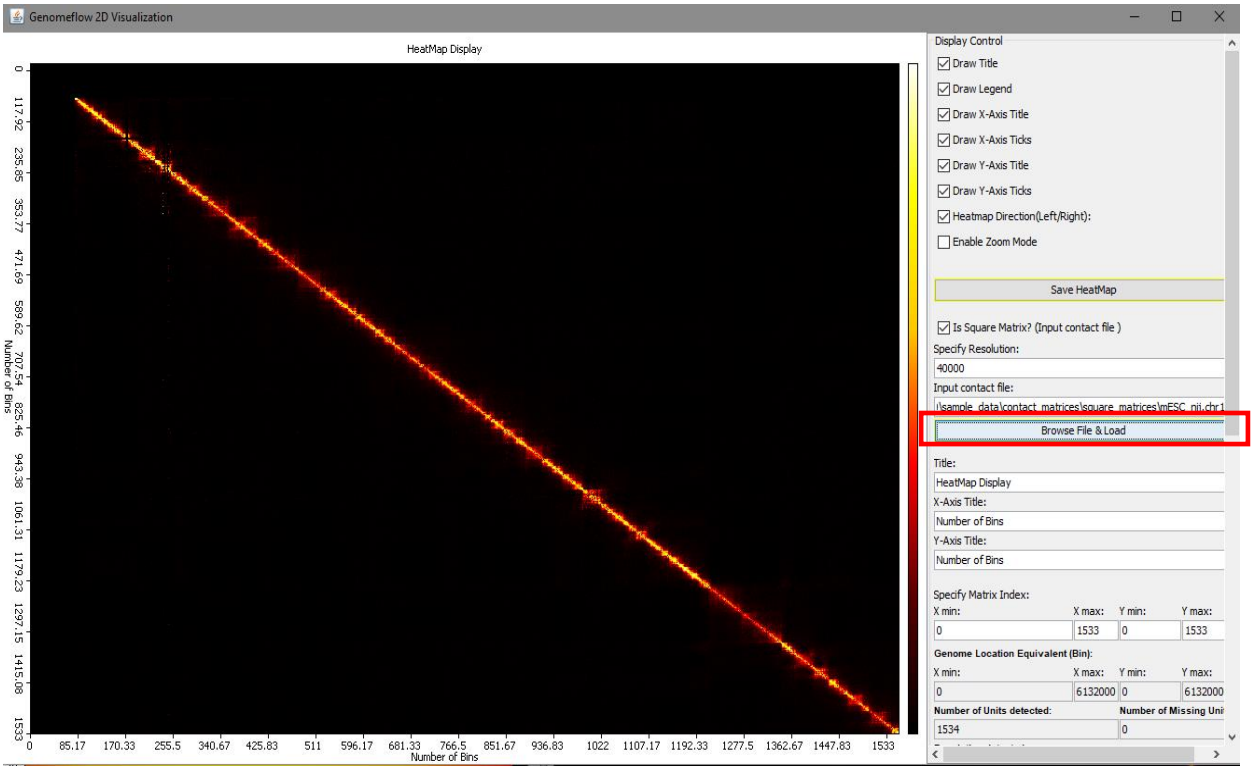


Figure 5.6. A contact matrix represented as a heatmap on the GenomeFlow 2D display window. User clicks on the highlighted button, Browse File & Load, to select the contact matrix file and display it on the heatmap display window.

Next, to annotate the contact matrix with identified TADs, the following steps will be performed. In the "Display Control" on the right, the user needs to load the TADs by clicking on the "Browse File & Load", in the TAD Annotation section in the Display control (Figure 5.7). Once the TADs have been loaded, click on the show TAD on heatmap to annotate the contact matrix with the identified TADs (Figure 5.8). The squares highlighted with white boundaries are the domains identifies for the contact matrix.

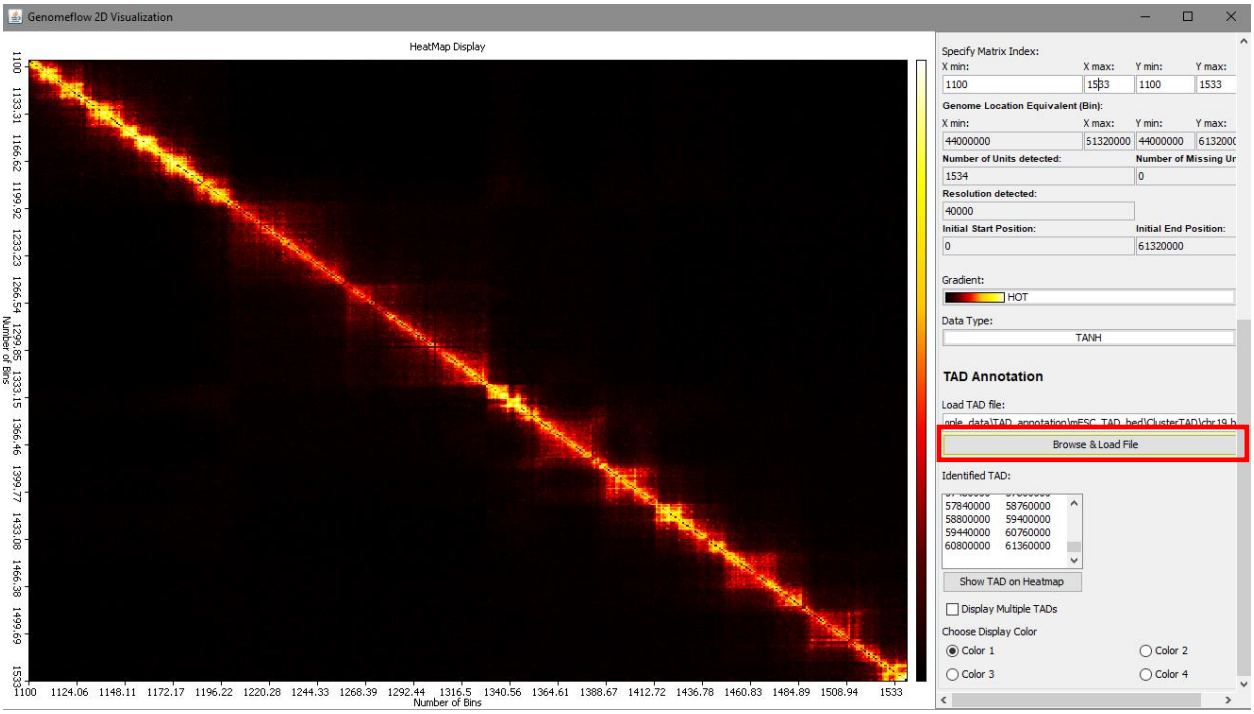


Figure 5.7. Loading the identified TAD into 2D visualization window

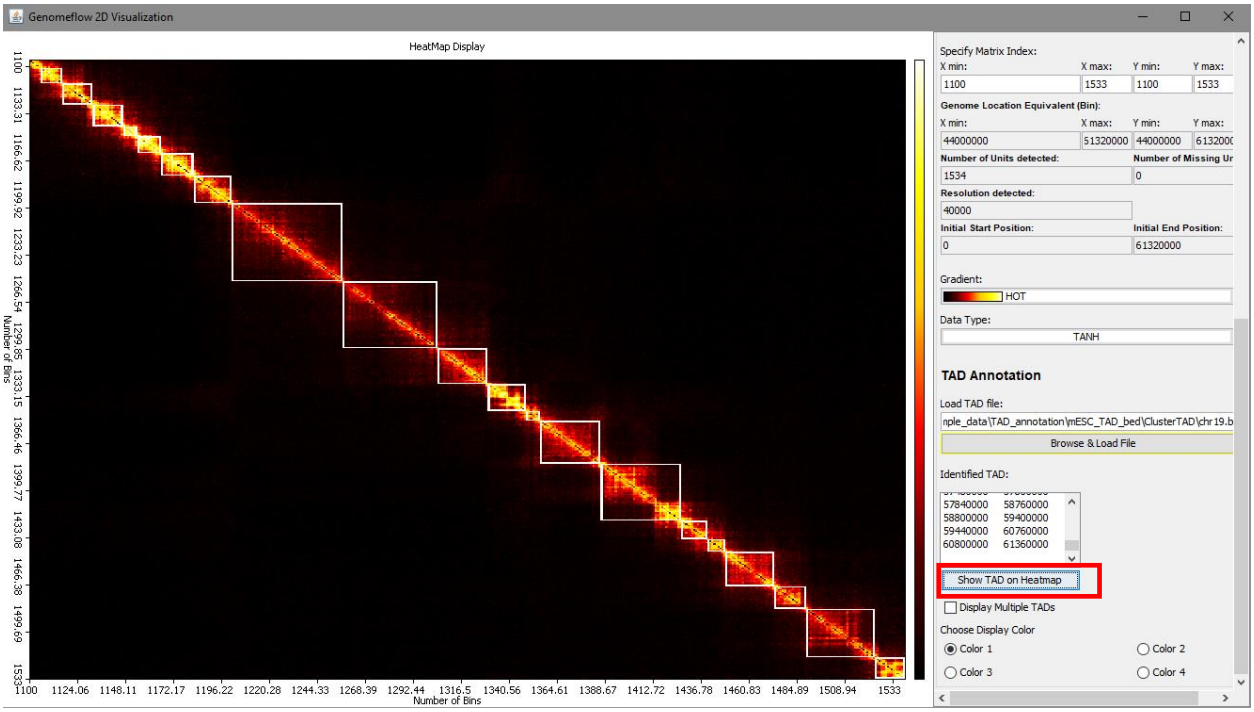


Figure 5.8. Demonstration of TAD Annotation on 2D Heatmap

Next, if the user has TADs identified by another methods, for example Dixon et al method, DI [141]. The user can display both TADs from different methods on the same 2D Heatmap by clicking on the “Display Multiple TADs” check box and choosing a different color to annotate it. In Figure 5.9, the TADs by ClusterTAD are in white, Color 1, and DI is highlighted in yellow, Color 2.

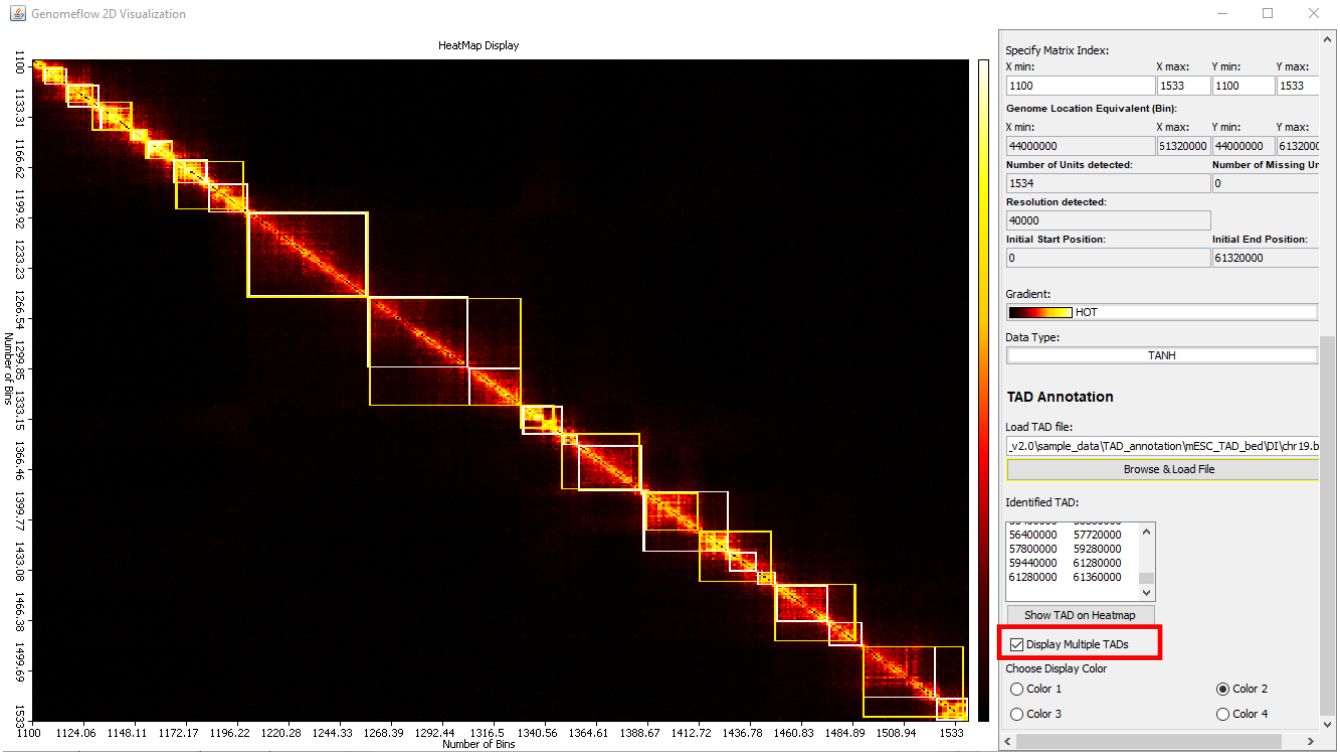


Figure 5.9. TAD Annotation on 2D Heatmap using TADs identified by ClusterTAD [177], in white and the DI [141] in yellow.

6 Tools for 3D structure reconstruction and feature extraction

6.1 Basic dependencies

All the methods were developed in Java. Users need to have a Java Development Kit (JDK) installed before the executables for the program can be used. Download JDK from here: Java 1.7 or 1.8 JDK. (Alternative link for Ubuntu/LinuxMint). Minimum system requirements for running Java can be found at http://java.com/en/download/help/sysreq.xml.

6.2 3DMax

6.2.1 Installation

The Java source codes for 3DMax is available at https://github.com/BDM-Lab/3DMax.

Download the latest version of the 3DMax executable file from https://github.com/BDM-Lab/3DMax/releases.

6.2.2 Usage

To run the tool, open a command line interface and type:

java -jar 3DMax.jar parameters.txt

, where the **"**paramaters.txt" contains parameters required to run to program.

The Parameters that are configured in the 'parameters.txt' file are

- NUM: Number of models to be generated
- OUTPUT_FOLDER: Output folder to store generated 3D structure
- INPUT_FILE: A normalized A 3-column Hi-C contact file (position_1 position_2 interaction frequency) or a N x N square matrix. N is the number of equal-sized fragments in the chromosome.
- CONVERT_FACTOR: the factor used to convert IF to distance, distance = 1/(IF^factor), when not specified, the program will search for it in range [0.1, 2.0], step = 0.1
- CHROMOSOME_LENGTH: If there is only one chromosome in the data, this

parameter should be omitted. It is required if the contact matrix contains data of several chromosomes. This parameter specifies a sequence of numbers, each representing the length of a chromosome in the input data. Numbers are separated by commas. The length of a chromosomes is the number of points required to represent the chromosome and therefore, the regions without contacts with other regions (such as centromeres) are not considered when calculating the length.

- VERBOSE: A true or false value to indicate if gradient lengths are displayed during the optimization.
- LEARNING_RATE: The initial learning rate for the optimization, if the optimization fails, reducing this value may help.
- MAX_ITERATION: Maximum number of iterations, the optimization may converge with a smaller number of iterations than this number.

6.2.3 Input Matrix File Format

3DMax allows two formats:

- Tuple Input format(*preferred*) : A hi-C contact file, each line containing 3 numbers (separated by a space) of a contact, position_1 position_2 interaction-frequency
- Square Matrix Input format: The square matrix is a comma separated N by N intra-chromosomal contact matrix derived from Hi-C data, where N is the number of equal-sized regions of a chromosome.

6.2.4 Example Input Hi-C data

Hi-C datasets we used in out study can be downloaded from here http://sysbio.rnet.missouri.edu/bdm_download/3DMax/

6.2.5 Output

3DMax produces 4 files"

- *.pdb: contains the model and can be visualized by pyMol, Chimera or GenomeFlow
- *_log_a_number.txt: contains the settings used to build the model and Spearman's correlation of reconstructed distances and input IFs
- *_log.txt: NUM > 1, the files contain settings and average root means square error (RMSE) and average correlation of Spearman's and Pearson's correlations of separate models
- *_coordinate_mapping.txt: contains the mapping of genomic positions to indices in the model. Notice that indices start from 0, while in pyMol or Chimera, id starts from 1

6.3 ClusterTAD

6.3.1 Installation

The Java source codes for ClusterTAD is available at https://github.com/BDM-Lab/ClusterTAD.

Download the latest version of the 3DMax executable file from https://github.com/BDM-Lab/ClusterTAD/releases.

6.3.2 Usage

To run the tool, open a command line interface and type:

java -jar ClusterTAD.jar Input-Matrix-file Matrix-Resolution

The Parameters are as follow:

- Input-Matrix-file: A tab separated N by N intra-chromosomal Hi-C contact matrix.
- Matrix-Resolution: The contact Matrix Resolution.

6.3.3 Input Matrix File Format

The input to ClusterTAD is a tab separated N by N intra-chromosomal contact matrix derived from Hi-C data, where N is the number of equal-sized regions of a chromosome

6.3.4 Example Input Hi-C data

In our study, we used the normalized Hi-C here : http://chromosome.sdsc.edu/mouse/hi-c/download.html

6.3.5 Output

ClusterTAD generates 2 folders in the Output directory. They are :

6.3.5.1 Clusters

- It contains a *.txt* file that contains the cluster assignment for the diagonal for all the K values considered for the clustering operation

6.3.5.2 TADs

- It contains the *.txt* files listing the TADs extracted from each clustering and re-clustering done.
- It contains the Best TAD identified based on the Quality score, labeled as "BestTAD_[name-of-input-file]_K=.txt".
- It contains a *.txt* file which contains a list of the extracted TAD Quality scores, file name = [name-of-input-file]_TAD_QualityScore_List.

6.4 GenomeFlow

6.4.1 Quick Start

1. Verify that you have installed the **basic dependencies above**.
2. To use the 1D-Function that provides reference genome indexing, alignment of fastq files and filtering of alignment files, follow the instructions here for the dependencies download and installation. *This step is required only for the 1D-Function tools*
3. Download the latest GenomeFlow Tools jar

4. Run GenomeFlow:
    - Windows OS: double-click the **genomeflow.bat** script
    - *Linux/UNIX* based OS: execute the script, **genomeflow.sh**

6.4.2 Dependencies and Installation

6.4.2.1 Operating System (OS)

A Linux, UNIX, or Mac OS X environment is required to use the 1D-Functions.

It is strongly recommended to work under a Mac OS X or a Linux/UNIX-based operating system, such as Ubuntu, Centos/Red Hat, Solaris.

If you are using a Windows operating system, install Cygwin first. Cygwin is a free software that provides a UNIX-like environment on Windows. The Cygwin install package can be found at *http://www.cygwin.com/*. Once Cygwin is installed, place your work in the Cygwin directory.

6.4.2.2 Download External Tools

- Download BWA (*http://bio-bwa.sourceforge.net/*) **OR** Bowtie2 (*http://bowtie-bio.sourceforge.net/index.shtml*) *for indexing and alignment creation.*
- Bowtie2 supports multiple OS, download the version for your OS. That is:
    - Download bowtie2- version number-macos-x86_64 for MacOS
    - Download bowtie2- version number- linux-x86_64 for Linux
    - Download bowtie2- version number- mingw-x86_64 for Mingw/Cygwin
- Download Samtools (*http://samtools.sourceforge.net/*)
- We tested on the following versions for each one of the tools: bwa-0.7.17, bowtie2-2.3.4-*, and samtools-1.6.
- You can also download the installation files for these tools from here: *http://sysbio.rnet.missouri.edu/bdm_download/GenomeFlow/External_Tools/*

6.4.2.3 Installing External Tools

6.4.2.3.1 BWA

- Open a Unix Terminal
- Change directory to the downloaded bwa-* directory. For example:
  - cd *bwa-0.7.17*
- Type **make** once you are inside the bwa directory. For example:
  - make
- This operation produces a binary file: **bwa**
  - In Unix based Operating system: **bwa**
  - In Cygwin/Mingw: **bwa.exe**
- Give executable permission to the **binary file**. For example:
  - In Unix based Operating system: chmod +x **bwa**
  - In Cygwin/Mingw: chmod +x **bwa.exe**

6.4.2.3.2 Bowtie2

- Open a Unix Terminal
- Change directory to the downloaded Bowtie2-* directory. For example:
  - cd *bowtie2-2.3.4-linux-x86_64*
- Give executable permission to the binary file. For example:
  - In Unix based Operating system/ Cygwin/Mingw: chmod +x **bowtie2***

6.4.2.3.3 Samtools

- Open a Unix Terminal
- Change directory to the downloaded samtools-* directory. For example:
  - cd *samtools-1.7*
- Type **./configure** once you are inside the samtools directory. For example:
  - ./configure
- After configuration is completed, type **make**. For example:
  - make
- This operation produces a binary file: **samtools**
  - In Unix based Operating system: **samtools**
  - In Cygwin/Mingw: **samtools.exe**

- Give executable permission to the **binary file**. For example:
    - In Unix based Operating system: chmod +x **samtools**
    - In Cygwin/Mingw: chmod +x **samtools.exe**

6.4.2.3.4 Gawk

- Open a Unix Terminal
- Install gawk
- For Linux Users:
    - Type **sudo apt-get install gawk**
- For Max OSX Users:
    - Follow the instructions here: http://macappstore.org/gawk/
- For Cygwin/Mingw Users:
    - Open a Unix terminal
    - Download *gawk-4.2.1.tar.gz*
        - *wget https://ftp.gnu.org/gnu/gawk/gawk-4.2.1.tar.gz*
    - Unzip the *gawk-4.2.1.tar.gz*
        - *tar -xvpzf gawk-4.2.1.tar.gz*
    - Change directory to the *gawk-4.2.1 directory*
        - cd *gawk-4.2.1*
    - Type **./configure** once you are inside the *gawk-4.2.1* directory.
        - ./configure
    - After configuration is completed, type **make**. For example:
        - make && make check

6.4.3 Usage

A comprehensive documentation of the various functions provided in GenomeFlow is available here https://github.com/jianlin-cheng/GenomeFlow/wiki

6.5 Conclusion and Future insights

Despite the improvement in 3-D structure modeling approaches, the lack of a real structure with which to contrast these models remains a challenge. In particular, it is currently difficult to confirm the true modeling capability of 3-D genome methods. Although the introduction of 3-D-FISH data and Hi-C data for joint modeling has received some attention recently [94], there is no sufficient 3-D-FISH data to guide most modeling on Hi-C data and to thoroughly validate the quality of computational models. The development of more advanced genome/chromosome imaging techniques will further improve the validation of 3-D genome models. In addition, other high-throughput sequencing data such as functional genomics and epigenomics data can be used to validate the biological validity of 3-D genome/ chromosome models by exploring their correlation with 3-D genomes.

Another challenge is to reconstruct high-resolution 3-D models of large genomes from Hi-C data, which are needed for studying detailed interactions between genes and regulatory elements, due to enormous time complexity and data sparsity associated with high-resolution modeling. Only a few methods [98] was designed to build high-resolution (*e.g.* 5 KB) models.

Finally, it is important to make 3-D genome modeling methods easy for biomedical scientists to use in their research. To this end, a few tools have been designed to visualize 3-D genome models [88, 89, 170–174]. Recently, GenomeFlow [40] provides a comprehensive graphical environment for users to process Hi-C data, generate chromosomal contact maps, build 3-D models, and apply 3-D models to integrate various omics data. More efforts of making 3-D genome modeling accessible to general users are still needed.

BIBLIOGRAPHY

[1] Misteli T. Beyond the sequence: cellular organization of genome function. Cell. 2007;128(4):787–800.

[2] Cremer T, Cremer C. Chromosome territories, nuclear architecture and gene regulation in mammalian cells. Nat Rev Genet. 2001;2(4):292.

[3] Branco MR, Pombo A. Chromosome organization: new facts, new models. Trends Cell Biol. 2007;17(3):127–34.

[4] Hakim O, Misteli T. SnapShot: chromosome conformation capture. Cell. 2012;148(5):1068–e1.

[5] Osório J. Chromosome biology: moving a TAD closer to unravelling chromosome architecture. Nat Rev Mol Cell Biol. 2015;16(12):701.

[6] Dekker J, Mirny L. The 3D genome as moderator of chromosomal communication. Cell. 2016;164(6):1110–21.

[7] Shen Y, Yue F, McCleary DF, Ye Z, Edsall L, Kuan S, Wagner U, Dixon J, Lee L, Lobanenkov VV, Ren B. A map of the cis-regulatory sequences in the mouse genome. Nature. 2012;488(7409):116.

[8] Makova KD, Hardison RC. The effects of chromatin organization on variation in mutation rates in the genome. Nat Rev Genet. 2015;16(4):213. 9.

[9] Cournac A, Marie-Nelly H, Marbouty M, Koszul R, Mozziconacci J. Normalization of a chromosomal contact map. BMC Genomics. 2012;13(1): 436.

[10] Wasserman WW, Sandelin A. Applied bioinformatics for the identification of regulatory elements. Nat Rev Genet. 2004;5(4):276.

[11] Taberlay PC, Achinger-Kawecka J, Lun AT, Buske FA, Sabir K, Gould CM, Zotenko E, Bert SA, Giles KA, Bauer DC, Smyth GK. Three-dimensional disorganization of the cancer genome occurs coincident with long-range genetic and epigenetic alterations. Genome Res. 2016;26(6):719–31.

[12] Dekker J. Gene regulation in the third dimension. Science. 2008;319(5871): 1793–4.

[13] Dekker J, Marti-Renom MA, Mirny LA. Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. Nat Rev Genet. 2013;14(6):390.

[14] de Laat W, Grosveld F. Spatial organization of gene expression: the active chromatin hub. Chromosom Res. 2003;11(5):447–59.

[15] Gorkin DU, Leung D, Ren B. The 3D genome in transcriptional regulation and pluripotency. Cell stem cell. 2014;14(6):762–75.

[16] Woodcock CL, Dimitrov S. Higher-order structure of chromatin and chromosomes. Curr Opin Genet Dev. 2001;11(2):130–5.

[17] Chromatin WA. San Diego: Structure and Function. San Diego, CA: Academic Press; 1998.

[18] Langer-Safer PR, Levine M, Ward DC. Immunological method for mapping genes on Drosophila polytene chromosomes. Proc Natl Acad Sci. 1982; 79(14):4381–5.

[19] Amann R, Fuchs BM. Single-cell identification in microbial communities by improved fluorescence in situ hybridization techniques. Nat Rev Microbiol. 2008;6(5):339.

[20] Westphal V, Rizzoli SO, Lauterbach MA, Kamin D, Jahn R, Hell SW. Video-rate far-field optical nanoscopy dissects synaptic vesicle movement. Science. 2008;320(5873):246–9.

[21] Hell SW, Wichmann J. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. Opt Lett. 1994;19(11):780–2.

[22] Betzig E, Patterson GH, Sougrat R, Lindwasser OW, Olenych S, Bonifacino JS, Davidson MW, Lippincott-Schwartz J, Hess HF. Imaging intracellular fluorescent proteins at nanometer resolution. Science. 2006;313(5793):1642–5.

[23] Huang B, Babcock H, Zhuang X. Breaking the diffraction barrier: superresolution imaging of cells. Cell. 2010;143(7):1047–58.

[24] Dekker J, Rippe K, Dekker M, Kleckner N. Capturing chromosome conformation. Science. 2002;295(5558):1306–11.

[25] de Wit E, de Laat W. A decade of 3C technologies: insights into nuclear organization. Genes Dev. 2012;26(1):11–24.

[26] Han J, Zhang Z, Wang K. 3C and 3C-based techniques: the powerful tools for spatial genome organization deciphering. Mol Cytogenet. 2018;11(1):21.

[27] Schmitt AD, Hu M, Ren B. Genome-wide mapping and analysis of chromosome architecture. Nat Rev Mol Cell Biol. 2016;17(12):743.

[28] Simonis M, Klous P, Splinter E, Moshkin Y, Willemsen R, De Wit E, Van Steensel B, De Laat W. Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture–onchip (4C). Nat Genet. 2006;38(11):1348.

[29] Dostie J, Richmond TA, Arnaout RA, Selzer RR, Lee WL, Honan TA, Rubio ED, Krumm A, Lamb J, Nusbaum C, Green RD. Chromosome conformation capture carbon copy (5C):

a massively parallel solution for mapping interactions between genomic elements. Genome Res. 2006; 16(10):1299–309.

[30] Lieberman-Aiden E, Van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie BR, Sabo PJ, Dorschner MO, Sandstrom R. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science. 2009;326(5950):289–93.

[31] Kalhor R, Tjong H, Jayathilaka N, Alber F, Chen L. Genome architectures revealed by tethered chromosome conformation capture and populationbased modeling. Nat Biotechnol. 2012;30(1):90.

[32] Fullwood MJ, Liu MH, Pan YF, Liu J, Xu H, Mohamed YB, Orlov YL, Velkov S, Ho A, Mei PH, Chew EG. An oestrogen-receptor-α-bound human chromatin interactome. Nature. 2009;462(7269):58.

[33] Li G, Fullwood MJ, Xu H, Mulawadi FH, Velkov S, Vega V, Ariyaratne PN, Mohamed YB, Ooi HS, Tennakoon C, Wei CL. ChIA-PET tool for comprehensive chromatin interaction analysis with paired-end tag sequencing. Genome Biol. 2010;11(2):R22.

[34] Nagano T, Lubling Y, Stevens TJ, Schoenfelder S, Yaffe E, Dean W, Laue ED, Tanay A, Fraser P. Single-cell Hi-C reveals cell-to-cell variability in chromosome structure. Nature. 2013;502(7469):59.

[35] Ron G, Globerson Y, Moran D, Kaplan T. Promoter-enhancer interactions identified from Hi-C data using probabilistic models and hierarchical topological domains. Nat Commun. 2017;8(1):2237.

[36] Fraser P, Bickmore W. Nuclear organization of the genome and the potential for gene regulation. Nature. 2007;447(7143):413.

[37] Mirny LA. The fractal globule as a model of chromatin architecture in the cell. Chromosom Res. 2011;19(1):37–51.

[38] Van Berkum NL, Lieberman-Aiden E, Williams L, Imakaev M, Gnirke A, Mirny LA, Dekker J, Lander ES. Hi-C: a method to study the three-dimensional architecture of genomes. J Vis Exp. 2010;6(39):e1869.

[39] Ay F, Noble WS. Analysis methods for studying the 3D architecture of the genome. Genome Biol. 2015;16(1):183.

[40] Trieu T, Oluwadare O, Wopata J, Cheng J. GenomeFlow: a comprehensive graphical tool for modeling and analyzing 3D genome structure. Bioinformatics. 2018; https://doi.org/10.1093/bioinformatics/bty802.

[41] Durand NC, Shamim MS, Machol I, Rao SS, Huntley MH, Lander ES, Aiden EL. Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. Cell Syst. 2016;3(1):95–8.

[42] Servant N, Varoquaux N, Lajoie BR, Viara E, Chen CJ, Vert JP, Heard E, Dekker J, Barillot E. HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. Genome Biol. 2015;16(1):259.

[43] Castellano G, Le Dily F, Pulido AH, Beato M, Roma G. Hi-Cpipe: a pipeline for high-throughput chromosome capture. bioRxiv. 2015:020636.

[44] Wingett S, Ewels P, Furlan-Magaril M, Nagano T, Schoenfelder S, Fraser P, Andrews S. HiCUP: pipeline for mapping and processing Hi-C data. F1000Research. 2015:4.

[45] Fraser J, Rousseau M, Shenker S, Ferraiuolo MA, Hayashizaki Y, Blanchette M, Dostie J. Chromatin conformation signatures of cellular differentiation. Genome Biol. 2009;10(4):R37.

[46] Adhikari B, Trieu T, Cheng J. Chromosome3D: reconstructing threedimensional chromosomal structures from Hi-C interaction frequency data using distance geometry simulated annealing. BMC genomics. 2016;17(1): 886.

[47] Zou C, Zhang Y, Ouyang Z. HSA: integrating multi-track Hi-C data for genome-scale reconstruction of 3D chromatin structure. Genome Biol. 2016;17(1):40.

[48] Rousseau M, Fraser J, Ferraiuolo MA, Dostie J, Blanchette M. Three-dimensional modeling of chromatin structure from interaction frequency data using Markov chain Monte Carlo sampling. BMC Bioinformatics. 2011;12(1):414.

[49] Trieu T, Cheng J. Large-scale reconstruction of 3D structures of human chromosomes from chromosomal contact data. Nucleic Acids Res. 2014;42(7):e52.

[50] Flory PJ. Principles of Polymer Chemistry. Ithaca: Cornell University Press; 1953.

[51] Gennes PG d. Scaling Concepts in Polymer Physics. Ithaca: Cornell University Press; 1979.

[52] Doi M, Edwards SF. The Theory of Polymer Dynamic. Oxford: Clarendon; 1986.

[53] Mateos-Langerak J, Bohn M, de Leeuw W, Giromus O, Manders EM, Verschure PJ, Indemans MH, Gierman HJ, Heermann DW, Van Driel R, Goetze S. Spatially confined

folding of chromatin in the interphase nucleus. Proceedings of the National Academy of Sciences. 2009:pnas-0809501106.

[54] Münkel C, Langowski J. Chromosome structure predicted by a polymer model. Phys Rev E. 1998;57(5):5888.

[55] Barbieri M, Chotalia M, Fraser J, Lavitas LM, Dostie J, Pombo A, Nicodemi M. A model of the large-scale organization of chromatin. Biochem Soc Trans. 2013;41:508–12.

[56] Grosberg AY, Nechaev SK, Shakhnovich EI. The role of topological constraints in the kinetics of collapse of macromolecules. J Phys. 1988; 49(12):2095–100.

[57] Bölinger D, Sułkowska JI, Hsu HP, Mirny LA, Kardar M, Onuchic JN, Virnau P. A Stevedore's protein knot. PLoS Comput Biol. 2010;6(4):e1000731.

[58] Van Holde KE. Chromatin: Springer series in molecular biology. New York: Springer-Verlag; 1988.

[59] Woodcock CL, Ghosh RP. Chromatin higher-order structure and dynamics. Cold Spring Harb Perspect Biol. 2010;2(5):a000596.

[60] Sewitz SA, Fahmi Z, Lipkow K. Higher order assembly: folding the chromosome. Curr Opin Struct Biol. 2017;42:162–8.

[61] Dina C, Meyre D, Gallina S, Durand E, Körner A, Jacobson P, Carlsson LM, Kiess W, Vatin V, Lecoeur C, Delplanque J. Variation in FTO contributes to childhood obesity and severe adult obesity. Nat Genet. 2007;39(6):724.

[62] Scuteri A, Sanna S, Chen WM, Uda M, Albai G, Strait J, Najjar S, Nagaraja R, Orrú M, Usala G, Dei M. Genome-wide association scan shows genetic variants in the FTO gene are associated with obesity-related traits. PLoS Genet. 2007;3(7):e115.

[63] Norton HK, Phillips-Cremins JE. Crossed wires: 3D genome misfolding in human disease. J Cell Biol. 2017;216(11):3441–52.

[64] Wang S, Xu J, Zeng J. Inferential modeling of 3D chromatin structure. Nucleic acids research. 2015;43(8):e54.

[65] Hua N, Tjong H, Shin H, Gong K, Zhou XJ, Alber F. Producing genome structure populations with the dynamic and automated PGS software. Nat Protoc. 2018;13(5):915.

[66] Duan Z, Andronescu M, Schutz K, McIlwain S, Kim YJ, Lee C, Shendure J, Fields S, Blau CA, Noble WS. A three-dimensional model of the yeast genome. Nature. 2010;465(7296):363.

[67] Tanizawa H, Iwasaki O, Tanaka A, Capizzi JR, Wickramasinghe P, Lee M, Fu Z, Noma KI. Mapping of long-range associations throughout the fission yeast genome reveals global genome organization linked to transcriptional regulation. Nucleic Acids Res. 2010;38(22):8164–77.

[68] Kalhor R, Tjong H, Jayathilaka N, Alber F, Chen L. Solid-phase chromosome conformation capture for structural characterization of genome architectures. Nat Biotechnol. 2012;30(1):90.

[69] Trieu T, Cheng J. 3D genome structure modeling by Lorentzian objective function. Nucleic Acids Res. 2016;45(3):1049–58.

[70] Oluwadare O, Zhang Y, Cheng J. A maximum likelihood algorithm for reconstructing 3D structures of human chromosomes from chromosomal contact data. BMC Genomics. 2018;19(1):161.

[71] Wächter A, Biegler LT. On the implementation of an interior-point filter linesearch algorithm for large-scale nonlinear programming. Math Program.

[72] 2006;106(1):25–57.

[73] Baù D, Sanyal A, Lajoie BR, Capriotti E, Byron M, Lawrence JB, Dekker J, MartiRenom MA. The three-dimensional folding of the α-globin gene domain reveals formation of chromatin globules. Nat Struct Mol Biol. 2011;18(1):107. 73. Alber F, Dokudovskaya S, Veenhoff LM, Zhang W, Kipper J, Devos D, Suprapto A, Karni-Schmidt O, Williams R, Chait BT, Rout MP. Determining the architectures of macromolecular assemblies. Nature. 2007;450(7170):683.

[74] Meluzzi D, Arya G. Recovering ensembles of chromatin conformations from contact probabilities. Nucleic Acids Res. 2012;41(1):63–75.

[75] Hu M, Deng K, Qin Z, Dixon J, Selvaraj S, Fang J, Ren B, Liu JS. Bayesian inference of spatial organizations of chromosomes. PLoS Comput Biol. 2013; 9(1):e1002893.

[76] Zhang Z, Li G, Toh KC, Sung WK. Inference of spatial organizations of chromosomes using semi-definite embedding approach and Hi-C data. In: Annual international conference on research in computational molecular biology. Berlin, Heidelberg: Springer; 2013. p. 317–32.

[77] Peng C, Fu LY, Dong PF, Deng ZL, Li JX, Wang XT, Zhang HY. The sequencing bias relaxed characteristics of Hi-C derived data and implications for chromatin 3D modeling. Nucleic Acids Res. 2013;41(19):e183.

[78] Varoquaux N, Ay F, Noble WS, Vert JP. A statistical approach for inferring the 3D structure of the genome. Bioinformatics. 2014;30(12):i26–33.

[79] Lesne A, Riposo J, Roger P, Cournac A, Mozziconacci J. 3D genome reconstruction from chromosomal contacts. Nat Methods. 2014;11(11):1141.

[80] Trieu T, Cheng J. MOGEN: a tool for reconstructing 3D models of genomes from chromosomal conformation capturing data. Bioinformatics. 2015;32(9): 1286–92.

[81] Shavit Y, Hamey FK, Lio P. FisHiCal: an R package for iterative FISH-based calibration of Hi-C data. Bioinformatics. 2014;30(21):3120–2.

[82] de Leeuw J. Applications of convex analysis to multidimensional scaling. In: van Cutsem B, *et al.*, editors. Recent advantages in Statistics. Amsterdam: North Holland Publishing Company; 1977.

[83] Nowotny J, Ahmed S, Xu L, Oluwadare O, Chen H, Hensley N, Trieu T, Cao R, Cheng J. Iterative reconstruction of three-dimensional models of human chromosomes from chromosomal contact data. BMC Bioinformatics. 2015;16(1):338.

[84] Paulsen J, Gramstad O, Collas P. Manifold based optimization for single-cell 3D genome reconstruction. PLoS Comput Biol. 2015;11(8):e1004396.

[85] Serra F, Baù D, Goodstadt M, Castillo D, Filion GJ, Marti-Renom MA. Automatic analysis and 3D-modelling of Hi-C data using TADbit reveals structural features of the fly chromatin colors. PLoS Comput Biol. 2017;13(7): e1005665.

[86] Tjong H, Li W, Kalhor R, Dai C, Hao S, Gong K, Zhou Y, Li H, Zhou XJ, Le Gros MA, Larabell CA. Population-based 3D genome structure analysis reveals driving forces in spatial genome organization. Proc Natl Acad Sci. 2016;113(12):E1663–72.

[87] Park J, Lin S. Impact of data resolution on three-dimensional structure inference methods. BMC Bioinformatics. 2016;17(1):70.

[88] Szalaj P, Michalski PJ, Wróblewski P, Tang Z, Kadlof M, Mazzocco G, Ruan Y, Plewczynski D. 3D-GNOME: an integrated web service for structural modeling of the 3D genome. Nucleic Acids Res. 2016;44(W1):W288–93.

[89] Szałaj P, Tang Z, Michalski P, Pietal MJ, Luo OJ, Sadowski M, Li X, Radew K, Ruan Y, Plewczynski D. An integrated 3-dimensional genome modeling engine for data-driven simulation of spatial genome organization. Genome Res. 2016; https://doi.org/10.1101/gr.205062.116.

[90] Carstens S, Nilges M, Habeck M. Inferential structure determination of chromosomes from single-cell Hi-C data. PLoS Comput Biol. 2016;12(12): e1005292.

[91] Paulsen J, Sekelja M, Oldenburg AR, Barateau A, Briand N, Delbarre E, Shah A, Sørensen AL, Vigouroux C, Buendia B, Collas P. Chrom3D: three-dimensional genome modeling from Hi-C and nuclear lamin-genome contacts. Genome Biol. 2017;18(1):21.

[92] Rieber L, Mahony S. miniMDS: 3D structural inference from high-resolution Hi-C data. Bioinformatics. 2017;33(14):i261–6.

[93] Zhu G, Deng W, Hu H, Ma R, Zhang S, Yang J, Peng J, Kaplan T, Zeng J. Reconstructing spatial organizations of chromosomes through manifold learning. Nucleic Acids Res. 2018;46(8):e50.

[94] Abbas A, He X, Zhou B, Zhu G, Ma Z, Gao JT, Zhang MQ, Zeng J. Integrating Hi-C and FISH data for modeling 3D organizations of chromosomes. bioRxiv. 2018;1:318493.

[95] Rosenthal M, Bryner D, Huffer F, Evans S, Srivastava A, Neretti N. Bayesian Estimation of 3D Chromosomal Structure from Single Cell Hi-C Data. BioRxiv. 2018;1:316265.

[96] Li J, Zhang W, Li X. 3D genome reconstruction with ShRec3D+ and Hi-C data. IEEE/ACM Trans Comput Biol Bioinform. 2018;1;15(2):460–8.

[97] Hua KJ, Ma BG. EVR: Reconstruction of Bacterial Chromosome 3D Structure Using Error-Vector Resultant Algorithm. bioRxiv. 2018;1:401513.

[98] Trieu T, Oluwadare O, Cheng J. Hierarchical Reconstruction of HighResolution 3D Models of Large Chromosomes. Scientific reports. 2019;9(1):4971.

[99] Borg I, Groenen P. Modern multidimensional scaling: theory and applications. J Educ Meas. 2003;40(3):277–80.

[100] Ay F, Bunnik EM, Varoquaux N, Bol SM, Prudhomme J, Vert JP, Noble WS, Le Roch KG. Three-dimensional modeling of the P. falciparum genome during the erythrocytic cycle reveals a strong connection between genome architecture and gene expression. Genome Res. 2014;24:974.

[101] Le TB, Imakaev MV, Mirny LA, Laub MT. High-resolution mapping of the spatial organization of a bacterial chromosome. Science. 2013;342(6159): 731–4.

[102] Fudenberg G, Mirny LA. Higher-order chromatin structure: bridging physics and biology. Curr Opin Genet Dev. 2012;22(2):115–24.

[103] Kiefer J. Sequential minimax search for a maximum. Proc Am Math Soc.1953;4(3):502–6.

[104] Yaffe E, Tanay A. Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. Nat Genet. 2011;43(11):1059.

[105] Imakaev M, Fudenberg G, McCord RP, Naumova N, Goloborodko A, Lajoie BR, Dekker J, Mirny LA. Iterative correction of Hi-C data reveals hallmarks of chromosome organization. Nat Methods. 2012;9(10):999.

[106] Hu M, Deng K, Selvaraj S, Qin Z, Ren B, Liu JS. HiCNorm: removing biases in Hi-C data via Poisson regression. Bioinformatics. 2012;28(23):3131–3.

[107] Servant N, Varoquaux N, Heard E, Barillot E, Vert JP. Effective normalization for copy number variation in Hi-C data. BMC Bioinformatics. 2018;19(1):313.

[108] Stansfield JC, Cresswell KG, Vladimirov VI, Dozmorov MG. HiCcompare: an Rpackage for joint normalization and comparison of HI-C datasets. BMC Bioinformatics. 2018;19(1):279.

[109] Serra F, Di Stefano M, Spill YG, Cuartero Y, Goodstadt M, Baù D, MartiRenom MA. Restraint-based three-dimensional modeling of genomes and genomic domains. FEBS Lett. 2015;589(20):2987–95.

[110] Baù D, Marti-Renom MA. Genome structure determination via 3C-based data integration by the integrative modeling platform. Methods. 2012;58(3): 300–6.

[111] Brunger AT. Version 1.2 of the Crystallography and NMR system. Nat Protoc. 2007;2(11):2728.

[112] Brünger AT, Adams PD, Clore GM, DeLano WL, Gros P, Grosse-Kunstleve RW, Jiang JS, Kuszewski J, Nilges M, Pannu NS, Read RJ. Crystallography & NMR system: a new software suite for macromolecular structure determination.Acta Crystallogr Sect D. 1998;54(5):905–21.

[113] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res. 2011;12(Jul):2121– 59.

[114] Rieping W, Habeck M, Nilges M. Inferential structure determination. Science. 2005;309(5732):303–6.

[115] Mishra B, Meyer G, Sepulchre R. Low-rank optimization for distance matrix completion. In: 50th IEEE Conference on Decision and Control and European Control Conference 2011 Dec 12: IEEE; 2011. p. 4455–60.

[116] Kruskal JB. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika. 1964;29(1):1–27.

[117] Shepard RN. The analysis of proximities: multidimensional scaling with an unknown distance function. I Psychometrika. 1962;27(2):125–40.

[118] Ben-Elazar S, Yakhini Z, Yanai I. Spatial localization of co-regulated genes exceeds genomic gene clustering in the Saccharomyces cerevisiae genome. Nucleic Acids Res. 2013;41(4):2191–201.

[119] Agarwal S, Wills J, Cayton L, Lanckriet G, Kriegman D, Belongie S. Generalized non-metric multidimensional scaling. In: Artificial Intelligence and Statistics; 2007. p. 11–8.

[120] Stevens TJ, Lando D, Basu S, Atkinson LP, Cao Y, Lee SF, Leeb M, Wohlfahrt KJ, Boucher W, O'Shaughnessy-Kirwan A, Cramard J. 3D structures of individual mammalian genomes studied by single-cell Hi-C. Nature. 2017; 544(7648):59.

[121] Nagano T, Lubling Y, Várnai C, Dudley C, Leung W, Baran Y, Cohen NM, Wingett S, Fraser P, Tanay A. Cell-cycle dynamics of chromosomal organization at single-cell resolution. Nature. 2017;547(7661):61.

[122] Trussart M, Serra F, Baù D, Junier I, Serrano L, Marti-Renom MA. Assessing the limits of restraint-based 3D modeling of genomes and genomic domains. Nucleic Acids Res. 2015;43(7):3465–77.

[123] Osborne CS, Chakalova L, Brown KE, Carter D, Horton A, Debrand E, Goyenechea B, Mitchell JA, Lopes S, Reik W, Fraser P. Active genes dynamically colocalize to shared sites of ongoing transcription. Nat Genet.2004;36(10):1065.

[124] Gozzetti A, Le Beau MM. Fluorescence in situ hybridization: uses and limitations. In Seminars in hematology 2000 Oct 1 (Vol. 37, No. 4, pp. 320– 33). WB Saunders.

[125] Ferrai C, de Castro IJ, Lavitas L, Chotalia M, Pombo A. Gene positioning. Cold Spring Harb Perspect Biol. 2010;2:a000588.

[126] Holwerda S, De Laat W. Chromatin loops, gene positioning, and gene expression. Front Genet. 2012;3:217.

[127] Geyer PK, Vitalini MW, Wallrath LL. Nuclear organization: taking a position on gene expression. Curr Opin Cell Biol. 2011;23(3):354–9.

[128] Yokota H, Van Den Engh G, Hearst JE, Sachs RK, Trask BJ. Evidence for the organization of chromatin in megabase pair-sized loops arranged along a random walk path in the human G0/G1 interphase nucleus. J Cell Biol. 1995; 130(6):1239–49.

[129] Miele A, Dekker J. Long-range chromosomal interactions and gene regulation. Molecular biosystems. 2008;4(11):1046-57.

[130] Van Steensel B, Dekker J. Genomics tools for unraveling chromosome architecture. Nature biotechnology. 2010 Oct;28(10):1089.

[131] Zhao Z, Tavoosidana G, Sjölinder M, Göndör A, Mariano P, Wang S, Kanduri C, Lezcano M, Sandhu KS, Singh U, Pant V. Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra-and interchromosomal interactions. Nature genetics. 2006 Nov;38(11):1341.

[132] Mossel E, Vigoda E. Limitations of Markov chain Monte Carlo algorithms for Bayesian inference of phylogeny. The Annals of Applied Probability. 2006;16(4):2215-34.

[133] Cole SR, Chu H, Greenland S, Hamra G, Richardson DB. Bayesian posterior distributions without Markov chains. American journal of epidemiology. 2012 Feb 3;175(5):368-75.

[134] Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, Senior A, Tucker P, Yang K, Le QV, Ng AY. Large scale distributed deep networks. InAdvances in neural information processing systems 2012 (pp. 1223-1231).

[135] Kendall DG. A survey of the statistical theory of shape. Statistical Science. 1989 May 1:87-99.

[136] Bookstein, Fred L. *Morphometric Tools for Landmark Data*. Cambridge, UK: Cambridge University Press, 1991.

[137] Seber, G. A. F. *Multivariate Observations*. Hoboken, NJ: John Wiley & Sons, Inc., 1984.

[138] MATLAB version 7.10.0. Natick, Massachusetts: The MathWorks Inc., 2010.

[139] GM, 06990 Normalized HiC Data. http://compgenomics.weizmann.ac.il/tanay/?page_id=283.

[140] Wang Z, Cao R, Taylor K, Briley A, Caldwell C, Cheng J. The properties of genome conformation and spatial gene interaction and regulation networks of normal and malignant human cell types. PloS one. 2013 Mar 11;8(3):e58793.

[141] Dixon JR, Selvaraj S, Yue F, Kim A, Li Y, Shen Y, Hu M, Liu JS, Ren B. Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature. 2012 May;485(7398):376.

[142] Lévy-Leduc C, Delattre M, Mary-Huard T, Robin S. Two-dimensional segmentation for analyzing Hi-C data. Bioinformatics. 2014 Aug 22;30(17):i386-92.

[143] Wang Y, Li Y, Gao J, Zhang MQ. A novel method to identify topological domains using Hi-C data. Quantitative Biology. 2015 Jun 1;3(2):81-9.

[144] Rao SS, Huntley MH, Durand NC, Stamenova EK, Bochkov ID, Robinson JT, Sanborn AL, Machol I, Omer AD, Lander ES, Aiden EL. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell. 2014 Dec 18;159(7):1665-80.

[145] Rust MJ, Bates M, Zhuang X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). Nature methods. 2006 Oct;3(10):793.

[146] Williamson I, Berlivet S, Eskeland R, Boyle S, Illingworth RS, Paquette D, Dostie J, Bickmore WA. Spatial genome organization: contrasting views from chromosome conformation capture and fluorescence in situ hybridization. Genes & development. 2014 Dec 15;28(24):2778-91.

[147] Oluwadare O, Highsmith M, Cheng J. An Overview of Methods for Reconstructing 3-D Chromosome and Genome Structures from Hi-C Data. Biological Procedures Online. 2019 Dec;21(1):7.

[148] GSE35156, Normalized Hi-C data. http://chromosome.sdsc.edu/mouse/hi-c/download.html. Accessed 10 Apr 2019.

[149] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012 Sep;489(7414):57.

[150] Knight PA, Ruiz D. A fast algorithm for matrix balancing. IMA Journal of Numerical Analysis. 2013 Jul 1;33(3):1029-47.

[151] Rego N, Koes D. 3Dmol. js: molecular visualization with WebGL. Bioinformatics. 2014 Dec 12;31(8):1322-4.

[152] Ferraiuolo MA, Rousseau M, Miyamoto C, Shenker S, Wang XQ, Nadler M, Blanchette M, Dostie J. The three-dimensional architecture of Hox cluster silencing. Nucleic acids research. 2010 Jul 24;38(21):7472-84.

[153] Ryba T, Hiratani I, Lu J, Itoh M, Kulik M, Zhang J, Schulz TC, Robins AJ, Dalton S, Gilbert DM. Evolutionarily conserved replication timing profiles predict long-range chromatin interactions and distinguish closely related cell types. Genome research. 2010 Jun 1;20(6):761-70.

[154] Taylor, K.H., Briley, A., Wang, Z., Cheng, J., Shi, H. and Caldwell, C.W., 2013, January. Aberrant epigenetic gene regulation in lymphoid malignancies. In *Seminars in hematology* (Vol. 50, No. 1, pp. 38-47). WB Saunders.

[155] Shin, H., Shi, Y., Dai, C., Tjong, H., Gong, K., Alber, F. and Zhou, X.J., 2015. TopDom: an efficient and deterministic method for identifying topological domains in genomes. *Nucleic acids research*, p.gkv1505

[156] Mizuguchi T, Fudenberg G, Mehta S, Belton J-M, Taneja N, Folco HD, FitzGerald P, Dekker J, Mirny L, Barrowman J, *et al.* Cohesin-dependent globules and heterochromatin shape 3D genome architecture in S. pombe. Nature. 2014

[157] Lajoie B.R., Dekker J., Kaplan N. The Hitchhiker's guide to Hi-C analysis: practical guidelines. Methods. 2015;72:65–75.

[158] Crane E., Bian Q., McCord R.P., Lajoie B.R., Wheeler B.S., Ralston E.J., Uzawa S., Dekker J., Meyer B.J. Condensin-driven remodelling of X chromosome topology during dosage compensation. Nature. 2015;523:240–244.

[159] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[160] Shen Y, Yue F, McCleary DF, Ye Z, Edsall L, Kuan S, Wagner U, Dixon J, Lee L, Lobanenkov VV, Ren B. A map of the cis-regulatory sequences in the mouse genome. Nature. 2012 Aug;488(7409):116.

[161] GEO19184, ChipSeq data , http://chromosome.sdsc.edu/mouse/download.html , Accessed 30 May, 2017.

[162] Ng, Andrew. "Clustering with the k-means algorithm." *Machine Learning* (2012).

[163] Ketchen Jr, David J., and Christopher L. Shook. "The application of cluster analysis in strategic management research: an analysis and critique." *Strategic management journal* (1996): 441-458.

[164] Van Bortle K, Nichols MH, Li L, Ong CT, Takenaka N, Qin ZS, Corces VG. Insulator function and topological domain border strength scale with architectural protein occupancy. Genome biology. 2014 May;15(5):R82.

[165] Phillips JE, Corces VG. CTCF: master weaver of the genome. Cell. 2009 Jun 26;137(7):1194-211.

[166] Guelen L, Pagie L, Brasset E, Meuleman W, Faza MB, Talhout W, Eussen BH, de Klein A, Wessels L, de Laat W, van Steensel B. Domain organization of human chromosomes revealed by mapping of nuclear lamina interactions. Nature. 2008 Jun;453(7197):948.

[167] Handoko L, Xu H, Li G, Ngan CY, Chew E, Schnapp M, Lee CW, Ye C, Ping JL, Mulawadi F, Wong E. CTCF-mediated functional chromatin interactome in pluripotent cells. Nature genetics. 2011 Jul;43(7):630.

[168] Holwerda SJ, de Laat W. CTCF: the protein, the binding partners, the binding sites and their chromatin loops. Philosophical Transactions of the Royal Society B: Biological Sciences. 2013 Jun 19;368(1620):20120369.

[169] Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, Liu XS. Model-based analysis of ChIP-Seq (MACS). Genome biology. 2008 Nov;9(9):R137.

[170] Calo E, Wysocka J. Modification of enhancer chromatin: what, how, and why?. Molecular cell. 2013 Mar 7;49(5):825-37.

[171] Maston GA, Evans SK, Green MR. Transcriptional regulatory elements in the human genome. Annu. Rev. Genomics Hum. Genet.. 2006 Sep 22;7:29-59.

[172] Berkhin P. A survey of clustering data mining techniques. InGrouping multidimensional data 2006 (pp. 25-71). Springer, Berlin, Heidelberg.

[173] Jain, Anil K., and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[174] Xu D, Tian Y. A comprehensive survey of clustering algorithms. Annals of Data Science. 2015 Jun 1;2(2):165-93.

[175] Davies DL, Bouldin DW. A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence. 1979 Apr(2):224-7.

[176] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics. 1987 Nov 1;20:53-65.

[177] Oluwadare O, Cheng J. ClusterTAD: an unsupervised machine learning approach to detecting topologically associated domains of chromosomes from Hi-C data. BMC bioinformatics. 2017 Dec;18(1):480.

[178] Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature methods. 2012 Apr;9(4):357.

[179] Li H, Durbin R. Fast and accurate long-read alignment with Burrows–Wheeler transform. Bioinformatics. 2010 Mar 1;26(5):589-95.

[180] Nowotny J, Wells A, Oluwadare O, Xu L, Cao R, Trieu T, He C, Cheng J. GMOL: an interactive tool for 3D genome structure visualization. Scientific reports. 2016 Feb 12;6:20802.

VITA

Oluwatosin Oluwadare was born and raised in Akure, Nigeria. He obtained his bachelor's degree in Computer Science (CS) from the Federal University of Technology, Akure (FUTA), and his master's degree in CS from the University of Texas, Arlington (UTA) in 2012 and 2015 respectively. While at UTA, his research was focused on artificial intelligence and ambient intelligence. At UTA he worked on smart care medical technologies to improve health care for the elderly using intelligent sensor for fall prediction and comprehensive gait analysis at the UTA Learn Lab.

He started his Ph.D. studies in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia in the fall of 2015. With his research interest in bioinformatics, machine learning, and data mining, he has proposed, and been published in reputable journals, including BMC Genomics, BMC Bioinformatics, Bioinformatics and Scientific Reports, novel methods for genome feature extraction, 3D structure prediction, 3D genome structures visualization, and machine learning application in bioinformatics.

His long-term goal is to develop computational methods that can be used to study and extract useful information for disease diagnosis and prediction from experimental genomic data.