Graph-Based Genome Inference from Hi-C Data

Yihang Shen[1], Lingge Yu[1], Yutong Qiu[1], Tianyu Zhang[2], and Carl Kingsford[1(✉)]

[1] Ray and Stephanie Lane Computational Biology Department, Carnegie Mellon University, Pittsburgh 15213, USA
carlk@cs.cmu.edu

[2] Department of Statistics and Data Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, USA

Abstract. Three-dimensional chromosome structure plays an important role in fundamental genomic functions. Hi-C, a high-throughput, sequencing-based technique, has drastically expanded our comprehension of 3D chromosome structures. The first step of Hi-C analysis pipelines involves mapping sequencing reads from Hi-C to linear reference genomes. However, the linear reference genome does not incorporate genetic variation information, which can lead to incorrect read alignments, especially when analyzing samples with substantial genomic differences from the reference such as cancer samples. Using genome graphs as the reference facilitates more accurate mapping of reads, however, new algorithms are required for inferring linear genomes from Hi-C reads mapped on genome graphs and constructing corresponding Hi-C contact matrices, which is a prerequisite for the subsequent steps of the Hi-C analysis such as identifying topologically associated domains and calling chromatin loops. We introduce the problem of genome sequence inference from Hi-C data mediated by genome graphs. We formalize this problem, show the hardness of solving this problem, and introduce a novel heuristic algorithm specifically tailored to this problem. We provide a theoretical analysis to evaluate the efficacy of our algorithm. Finally, our empirical experiments indicate that the linear genomes inferred from our method lead to the creation of improved Hi-C contact matrices, which are more effective in accurately capturing the structures of topologically associated domains.

Keywords: Hi-C · Genome Graph · Dynamic Program · NP-completeness

1 Introduction

The spatial arrangement of chromosomes plays an important role in many crucial cellular processes including gene transcription [12,32], epigenetic modification [17], and replication timing [28]. This complex structure can be discovered

Y. Shen, L. Yu and Y. Qiu—These authors contributed equally to this work.

J. Ma (Ed.): RECOMB 2024, LNCS 14758, pp. 115–130, 2024.
https://doi.org/10.1007/978-1-0716-3989-4_8

through Hi-C [25], a high-throughput variant of the chromosome conformation capture technique [6], which has become a prevalent tool in the study of genomic organization. The Hi-C process yields read pairs representing spatial contacts between two genomic loci. These contacts can be identified by aligning each end of a read pair to the reference genome. These aligned read pairs facilitate subsequent analyses, such as identifying topologically associated domains (TADs) [5,7,10,23,26], which are contiguous regions on chromosomes with more frequent contacts, and calling chromatin loops [33], which are pairs of genomic loci that lie far apart along the linear genome but are in close spatial proximity.

Hi-C pipelines use a linear reference genome such as Genome Reference Consortium Human Build 38 (GRCh38) as the template against which to align reads. However, these linear references do not incorporate the genetic diversity within populations. Consequently, aligning reads from genomes that diverge from the linear reference genome can result in reads either not aligning at all or being mapped to incorrect genomic locations. This issue is exacerbated when analyzing Hi-C reads from cancer samples, which frequently exhibit structural variations, including copy number variations and substantial translocations. The misalignments, arising due to structural variations, can confound the interpretation of Hi-C data, potentially producing features that may be mistaken for other biological signals, such as chromatin loops [39]. Given that read alignment is always the initial step in Hi-C analysis, errors at this stage can proliferate, leading to inaccuracies throughout the downstream analyses. To rectify the Hi-C analysis of cancer cell lines, substantial efforts have been made to develop algorithms to identify structural variations and rearrange the cancer genomes from Hi-C data, sometimes with the help of other data types such as whole genome sequencing to enhance accuracy and precision [34,39,42] of Hi-C analysis of cancer cell lines. However, these steps still rely on the linear reference genome.

The concept of pan-genome has been introduced to address the shortcomings of linear references. The pan-genome is a collection of DNA sequences that incorporates both common DNA regions and sequences unique to each individual [16,40]. These DNA sequences can be represented by genome graphs, which are graph-based data structures amalgamating the linear reference alongside genetic variations and polymorphic haplotypes [1]. Numerous computational techniques have been published in the domain of genome graphs, addressing various aspects such as efficient genome graph construction [14,15,20,27,29], graph-based genome alignment [21,30,38] and graph-based structural variation and haplotype analyses [4,9,19]. These studies have substantiated that genome graphs can enhance the analysis of genome sequencing data. Moreover, Liao et al. [24] has illustrated the potential of genome graphs in improving the analysis of various other data types, including RNA-seq, ChIP-seq, and ATAC-seq. However, there has been no research exploring the enhancement of Hi-C analyses through the use of genome graphs.

Leveraging genome graphs as the reference can enhance the accuracy of mapping Hi-C reads. However, a challenge arises from the erroneous information in the graphs post-read alignment, attributed to structural variations present in

the genome graphs but absent in the actual linear genome of the Hi-C sample. Besides, these graphs are not immediately applicable for subsequent standard Hi-C analysis steps like TAD identification [10,41] and chromatin loop calling [2], given their inherent dependence on linear genomes and the corresponding Hi-C contact matrices, the two-dimensional matrices representing chromatin interactions. A critical component to address this is to infer the appropriate, sample-specific linear genome from Hi-C reads mapped on genome graphs. These inferred genomes, which are more congruent with the Hi-C samples' unknown ground truth genomes than traditional linear reference genomes, account for polymorphisms and structural variations specific to the given sample. By using these reconstructed genomes to create Hi-C contact matrices and incorporating these matrices into subsequent analyses, we can enhance the precision of Hi-C studies. This method offers a more sample-specific genomic representation, addressing the shortcomings inherent in using standard linear reference genomes.

We investigate the problem of genome sequence inference from Hi-C data on directed acyclic genome graphs. The problem of inferring genome sequences from genome graphs and whole genome sequencing data was explored in previous work such as Ebler et al. [8]. Yet, the challenge of addressing this problem using Hi-C data remains unexplored. We propose a novel problem objective to formalize our inference problem. To infer the genome, we choose the best source-to-sink path in the directed acyclic graph that optimizes the confidence of TAD inference on the genomes. We show that optimizing this objective is NP-complete, a complexity that persists even with directed acyclic graphs. We develop a greedy heuristic for the problem and theoretically show that, under a set of relaxed assumptions, the heuristic finds the optimal path with a high probability. We test our processing pipeline and genome inference algorithms on cancer Hi-C samples K-562. Results on these samples show that compared to using the traditional linear reference genomes, the linear genomes inferred from our method create improved Hi-C contact matrices, which are more effective in accurately capturing the structures of TADs, attesting to the algorithm's potential to enhance the precision and reliability of genomic studies. The source code of our method is available at https://github.com/Kingsford-Group/graphhic. More algorithmic and experimental details are in Shen et al. [37].

2 Inferring the Sample Genome from Hi-C Data with Genome Graphs

Typical Hi-C processing pipelines, such as HiC-Pro [35], mainly consist of two steps: (i) aligning each end of raw read pairs to the linear reference genome; (ii) constructing a two-dimensional contact matrix that describes the interactions between pairs of genomic regions. A contact matrix is derived from the alignment results, wherein each entry contains the number of read pairs between two genomic bins—intervals with a fixed length such as 10 kilobases. This contact matrix is used as an input for downstream analyses, such as TAD identification. However, current Hi-C analysis pipelines are unable to process data when genome graphs are used as the reference instead of linear reference.

By using the graph-based Hi-C processing pipeline proposed in Shen et al. [36], we are able to build a genome graph and a Hi-C contact matrix M. Each dimension of this matrix represents nodes of the graphs, and each matrix entry is the number of read pairs with ends mapped to the corresponding nodes. The nodes are ordered in their topological order in the genome graph.

2.1 Problem Definition of Genome Inference

Given a directed acyclic genome graph G with a source node s and a sink node t, and the contact matrix M derived from the graph-based Hi-C pipeline [36], the objective of genome inference is to find a s-t path in G such that the concatenated DNA sequences represented by nodes in the selected path is the most similar sequence to the actual genome of the Hi-C sample. Ideally, two primary criteria should be fulfilled by this reconstructed path: (i) it should encapsulate as many mapped Hi-C read pairs as feasible, and (ii) the distribution of these mapped pairs ought to echo the distinctive spatial structures of chromosomes, especially the topologically associated domains (TADs or "domains" for brevity). Motivated by these prerequisites, our approach toward genome inference encompasses a simultaneous inference of the s-t path and the corresponding TADs from G.

Let P_{st} be the collection of all s-t paths in G, and let D_p be a set of domains along path $p \in P_{st}$. The i-th domain on path p is defined as a subpath $d_i^p = [a_i^p, b_i^p]$ that starts at node a_i^p and ends at b_i^p, where a_i^p and b_i^p are nodes on p. We require that domains of a path do not overlap with each other. Furthermore, the boundaries of two consecutive domains d_i^p and d_{i+1}^p, b_i^p and a_{i+1}^p, must be two nodes connected with an edge on path p. The first and the last domain are $d_1^p = [s, b_1^p]$ and $d_{|D_p|}^p = [a_{|D_p|}^p, t]$.

We infer the s-t path representing the actual genome from a Hi-C sample by solving the following problem:

Problem 1. We are given a directed, acyclic graph $G = (V, E)$ with a source node s and a sink node t, a pre-computed function $\mu : \mathbb{N} \to \mathbb{R}_{\geq 0}$, a real number $\gamma \geq 0$, and a cost function $c : V \times V \to \mathbb{R}_{\geq 0}$ that maps every pair of nodes to a non-negative cost. c is symmetric in a sense that $c(v, v') = c(v', v)$. The goal is to find a s-t path $p = \{v_1, v_2, \ldots, v_{|p|}\}$ over all s-t paths and a set of consecutive domains D_p on p that optimize

$$\max_{p \in P_{st}} \max_{D_p} \sum_{[a_i^p, b_i^p] \in D_p} f([a_i^p, b_i^p]), \tag{1}$$

where f is defined as:

$$f(p') := \frac{1}{|p'|^\gamma} \sum_{v_i, v_j \in p', 1 \leq i \leq j \leq |p'|} c(v_i, v_j) - \mu(|p'|). \tag{2}$$

The cost function $c(v_i, v_j)$ can be defined as the values of entries in the contact matrix $M(v_i, v_j)$ for node v_i and node v_j. f quantifies the quality

of a TAD as the normalized number of interactions within the subpath p'. $\sum_{v_i,v_j \in p', 1 \leq i \leq j \leq |p'|} c(v_i, v_j)$ in Eq. (2) is the total number of interactions between nodes that are both in path p'.

The total number of interactions is normalized by two factors. First, the total number is zero-centered by a pre-computed $\mu(|p'|)$, which is the expected interaction frequency within a path with $|p'|$ nodes. Then, it is normalized by the number of nodes in p' ($|p'|$) scaled by a factor of γ. This normalization prevents the identification of TAD domains with excessively large sizes. Larger values of γ typically lead to finding smaller domains.

2.2 The Hardness of the Problem

The objective function (1) of Problem 1 is derived from that of Filippova et al. [10]. However, Filippova et al. [10] employs polynomial-time dynamic programming to infer TADs based on reads mapped to a linear reference, while our problem requires the concurrent inference of both the TAD domains (denoted as D_p) and the sample's linear genome (represented by the path p) directly from G. We show that this increased complexity results in NP-completeness of Problem 1. This hardness results motivates the development of practical heuristics for the problem.

Theorem 1. *Problem 1 is NP-complete.*

Proof. We prove Theorem 1 by reducing from the Path Avoiding Forbidden Pairs (PAFP) problem, which has been confirmed to be NP-complete even in directed acyclic graphs [13,22].

*Problem 2 (*Path avoiding forbidden pairs [22]*).* Given a graph $G = (V, E)$ with two fixed vertices $s, t \in V$ and a set of pairs of vertices $F \subset V \times V$, find a path from s to t that contains at most one vertex from each pair in F, or recognize that such path does not exist.

Gabow et al. [13] prove that the path avoiding forbidden pairs problem (PAFP), introduced in Problem 2, is NP-complete in directed acyclic graphs. We now reduce PAFP on DAGs to Problem 1. Suppose we are given an instance of PAFP with a DAG $G = (V, E)$, a source node s, and a sink node t. We define a symmetric cost function c on G, such that:

$$c(v, v') = c(v', v) = \begin{cases} 0 & \text{if } (v, v') \in F \\ 1 & \text{otherwise.} \end{cases}$$

We convert G to a new DAG $G' = (V', E')$ such that every path from the source node to the sink node in the new graph has the same length (same number of nodes). We use Breadth-First-Search (BFS), starting from s to generate the new graph. We first create G' that only has a source node $\bar{s}$, i.e. $V' = \{\bar{s}\}$ and $E' = \emptyset$. Let $V_0 = \{s\}$. Given the node set V_i, via BFS on G we create a new node set V_{i+1} which are all child nodes of the nodes in V_i. For each node $v \in V_{i+1}$, we

add a corresponding node $\bar{v}^{i+1}$ in G'. For each node pair (v', v) such that $v' \in V_i$ and $v \in V_{i+1}$, we add an edge from $\bar{v'}^i$ to $\bar{v}^{i+1}$ in G' if v' is a parent node of v in G. If $\bar{t}^i$ is already added in G', we add a node $\bar{t}^{i+1}$ in G' and add an edge from $\bar{t}^i$ to $\bar{t}^{i+1}$. If additionally t is in V_{i+1}, meaning that $\bar{t}^{i+1}$ is already in G', we only add an edge from $\bar{t}^i$ to $\bar{t}^{i+1}$ without adding the node $\bar{t}^{i+1}$ again. This procedure is iteratively conducted until $V_{i+1} = \{t\}$ or $V_{i+1} = \emptyset$. Figure 1 shows an example of constructing G' (Fig. 1(b)) from the original graph G (Fig. 1(a)). Since $|V_i| = O(|V|)$, we have that $|V'| = O(|V|^3)$. Therefore, the construction of G' can be accomplished within polynomial time. In addition, it is easy to see that G' is a DAG. Let $\bar{t}^n$ be the node in G' that corresponds to the sink node in G, which was added during the final iteration of the procedure. The set of s-t paths in G has a one-to-one correspondence with the set of paths from $\bar{s}$ and $\bar{t}^n$ in G'. Moreover, all the paths from $\bar{s}$ to $\bar{t}^n$ in G' maintain equal lengths $n+1$. We define a symmetric cost function c' on G', such that:

$$c'(\bar{v'}^i, \bar{v}^j) = c'(\bar{v}^j, \bar{v'}^i) = c(v, v').$$

Therefore, the instance of PAFP is a yes-instance if only if there exists a $\bar{s} - \bar{t}^n$ path in G' such that the cost c' of any node pair in this path is 1. The DAG G', combined with the source node $\bar{s}$, the sink node $\bar{t}^n$, the cost function c', the value $\gamma = 0$ and the function $\mu(l) \equiv 0$, becomes an instance of Problem 1.

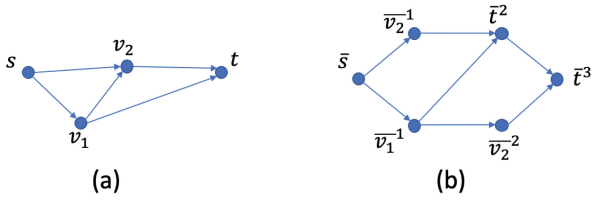

Fig. 1. An example of converting G (a) to G' (b) in the proof of Theorem 1.

We now prove that the instance of PAFP is a yes-instance if and only if there exists a solution of Problem 1 with objective value $\frac{(n+1)(n+2)}{2}$, hence Problem 1 is NP-complete.

Since $\gamma = 0$ and $\mu(l) \equiv 0$, the objective of Problem 1 becomes:

$$\max_{p \in P_{\bar{s}\bar{t}^n}} \max_{D_p} \sum_{[a_i^p, b_i^p] \in D_p} \sum_{\substack{v_i, v_j \in [a_i^p, b_i^p] \\ 1 \le i \le j \le |[a_i^p, b_i^p]|}} c'(v_i, v_j).$$

Since the cost function is non-negative, for any given $\bar{s} - \bar{t}^n$ path, choosing the whole path as one domain leads to the maximal:

$$\max_{p \in P_{\bar{s}\bar{t}^n}} \max_{D_p} \sum_{[a_i^p, b_i^p] \in D_p} \sum_{\substack{v_i, v_j \in [a_i^p, b_i^p] \\ 1 \le i \le j \le |[a_i^p, b_i^p]|}} c'(v_i, v_j) \le \max_{p \in P_{\bar{s}\bar{t}^n}} \sum_{v_i, v_j \in p, 1 \le i \le j \le |p|} c'(v_i, v_j).$$

Moreover, since c' is less than or equal to 1, and each $\bar{s} - \bar{t}^n$ path has the same length $n+1$, we have:

$$\max_{p \in P_{\bar{s}\bar{t}^n}} \sum_{v_i, v_j \in p, 1 \le i \le j \le |p|} c'(v_i, v_j) \le \frac{(n+1)(n+2)}{2}.$$

Therefore, there exists a solution of Problem 1 with objective value $\frac{(n+1)(n+2)}{2}$ if and only if there exists a $\bar{s} - \bar{t}^n$ path in G' such that the cost of any node pair in this path is 1, if and only if the instance of PAFP is a yes-instance. □

2.3 Computation of the μ Function

Filippova et al. [10] demonstrated a method for efficiently pre-computing μ on the linear reference genome. Nevertheless, as we discuss in Shen et al. [37], calculating μ in the context of genome graphs poses a more complex challenge. Consequently, we propose a different strategy to estimate μ. Generally, samples from normal cell types bear a greater resemblance to the linear reference genome compared to those from cancer samples. Hence, we select Hi-C data from a normal sample, process it using the linear reference genome, and calculate its μ function using the same approach as Filippova et al. [10]. This function is denoted as μ_0. It is evident that the sequencing depth of the Hi-C data can influence the value of the μ function. Therefore, when analyzing new Hi-C data, we estimate its μ function as follows:

$$\hat{\mu}(l) = \mu_0(l) \frac{N_{new}}{N_{old}}. \tag{3}$$

Here, N_{old} refers to the total count of Hi-C read pairs from the original normal sample, while N_{new} indicates the total count of Hi-C read pairs from new Hi-C data.

2.4 Graph-Based Dynamic Programming Algorithm

We use a dynamic program, computed in the topological ordering of the nodes, to solve the Problem 1:

$$OPT(l) = \max_{k, P_{kl} \neq \emptyset} \left\{ \max_{v \in PA(k)} OPT(v) + q(k, l) \right\}, \tag{4}$$

where $OPT(l)$ is the optimal solution for objective (1), applied to the subgraph induced by node l along with all nodes with a topological order less than that of l within G. P_{kl} denotes the collection of all paths from node k to node l in G, and $PA(k)$ is the set of parent nodes of k. $\max_{v \in PA(k)} OPT(v) = 0$ if k has no parent node. q is a function that maximizes over all paths between k and l:

$$q(k, l) = \max_{p \in P_{kl}} f(p). \tag{5}$$

We use a standard backtracking strategy, shown in Shen et al. [37], to reconstruct the optimal path p_{opt} from the dynamic program. The reconstructed path p_{opt} is taken to be the inferred genome. We prove that $OPT(t)$ is indeed the optimal solution for Problem 1.

Proposition 1. $OPT(t) = \max_{p \in P_{st}} \max_{D_p} \sum_{[a_i^p, b_i^p] \in D_p} f([a_i^p, b_i^p])$.

The proof is in Shen et al. [37]. However, this does not provide a polynomial time algorithm. As we show in Shen et al. [37], computing the function q in (5) is NP-complete. Moreover, the NP-completeness of computing q is not attributable to the particular definition of function f as outlined in (2). We show in Shen et al. [37] that computing q remains NP-complete under various definitions of f. Therefore, it is hard to solve the dynamic programming objective shown in (4), which is consistent with the hardness conclusions in Sect. 2.2. This provides a focus for developing heuristics.

2.5 Heuristics for Computing q

We propose a novel heuristic algorithm, detailed in Algorithm 1, to compute the function $q(k, l)$ for any node pair (k, l). The central principle behind this algorithm is that a node situated between nodes k and l (in topological order) that has more interactions with other nodes is more likely to be a part of the path connecting k to l that maximizes the function f. Consequently, we sort the nodes in descending order based on their cumulative interactions with other nodes (line 6 of Algorithm 1) and progressively add nodes from the highest to the lowest interactions until a k-l path is established.

Algorithm 1. $q(k, l)$ computation

1: **Input** k,l, genome directed acyclic graph $G = (V, E)$, nodes list T that contains all nodes in G sorted by their topological order, contact matrix M, reachable matrix M_r, the function f
2: **if** $M_r[k, l] = 0$ **then**
3: **return**
4: **end if**
5: $V_{sub} \leftarrow \{v \in V \mid T.index(v) \geq T.index(k)$ and $T.index(v) \leq T.index(l)\}$
6: Sort vertices v in V_{sub} according to the sum $\sum_{v' \in V_{sub}} M[v, v']$, arranging them from the highest to the lowest value to form V'_{sub}.
7: $p \leftarrow \{k, l\}$, $q \leftarrow -\infty$
8: $edges \leftarrow is\_edge(k, l)$
9: **for** $v \in V'_{sub}$ **do**
10: $p, edges \leftarrow insert(p, M_r, T, edges, v)$
11: **if** $edges = |p| - 1$ **then**
12: $q \leftarrow f(p)$
13: **return** q, p
14: **end if**
15: **end for**

Specifically, we employ the following functions and data structures within Algorithm 1 to enhance the algorithm's efficiency:

- reachable matrix M_r, where $M_r[i, j] = 1$ if there exists a path from node i to node j in the directed acyclic genome graph G, otherwise $M_r[i, j] = 0$.
- $is\_edge(k, l)$, which returns 1 if there is an edge from k to l in G, otherwise it returns 0.
- $insert(p, M_r, T, edges, v)$, of which the pseudo-code is provided in Algorithm 2. This function contains the following steps:
  - Given a node set p which encompasses all nodes already incorporated and are topologically sorted, the function determines whether there exists a path in G that includes all nodes in $p \cup \{v\}$. Such a path may include additional nodes that are not in $p \cup \{v\}$. This step can be efficiently achieved with the help of M_r and a balanced tree structure such as AVL tree [11], of which the details are introduced in the proof of Theorem 2.
  - If the aforementioned path exists, the node v is then inserted into p according to the topological ordering (function *update* in line 7 of Algorithm 2).
  - The function also updates an integer variable *edges* (line 8 of Algorithm 2), which keeps track of how many neighboring nodes in p have edges in G.

The *insert* function yields a revised node set p and an updated value for *edges* (line 10 of Algorithm 1). A legitimate path in graph G is formed by the nodes in p if and only if the condition $edges = |p| - 1$ is met (line 11 of Algorithm 1). Once a path is established, we compute the function value $f(p)$ and use it as the value of q (line 12 of Algorithm 1). We have the following result on the time complexity of Algorithm 1:

Theorem 2. *The total time complexity for Algorithm 1 and the dynamic program using the heuristic Algorithm 1 are respectively* $\mathcal{O}(|V|^2)$ *and* $\mathcal{O}(|V|^4)$*, where* $|V|$ *is the number of nodes in the graph.*

The proof is in Shen et al. [37]. In practice, the time complexity is still too high for long chromosomes. To address this, as detailed in Sect. 2.7, we implement additional practical strategies to further decrease the algorithm's time complexity.

2.6 Accuracy of the Heuristic Algorithm

Let $\hat{p}$ represent the path from node k to node l as predicted by Algorithm 1, and let p_{gt} denote the "ground truth" path, defined as $p_{gt} = \arg\max_{p \in P_{kl}} f(p)$. In an ideal scenario, a heuristic algorithm would ensure that, for any specified DAG G and any interaction distribution present on G, the value $f(\hat{p})$ closely approximates $f(p_{gt})$. However, as we demonstrate in Shen et al. [37], it is possible to create an example where the discrepancy between $f(\hat{p})$ and $f(p_{gt})$ can be infinitely large, indicating that our heuristic algorithm does not offer a bounded approximation in the worst-case scenario.

Algorithm 2. *insert* function

Input node set p in which nodes are topologically sorted, matrix M_r, nodes list T that contains all nodes in G sorted by their topological order, integer variable $edges$, node v
Find two adjacent nodes v_1 and v_2 in p such that $T.index(v_1) \leq T.index(v) \leq T.index(v_2)$.
if $v_1 = v$ or $v_2 = v$ **then**
    return p,$edges$
end if
if $M_r[v_1, v] = 1$ and $M_r[v, v_2] = 1$ **then**
    $p \leftarrow update(p, v)$
    $edges \leftarrow edges + is\_edge(v_1, v) + is\_edge(v, v_2) - is\_edge(v_1, v_2)$
end if
return p,$edges$

However, within the scope of Hi-C analysis, the distribution of interactions on a genome graph is not arbitrary. Conceptually, each interaction, represented by a pair of nodes, stems from two primary sources: (a) The "ground truth" source. Both nodes of the interactions from this source lie on the ground truth path p_{gt}. Interactions from this source are informative when constructing $\hat{p}$. (b) The "noise" source, which accounts for interactions arising due to various systematic biases such as sequencing errors, mapping errors, etc. In this scenario, the interactions are not necessarily confined to the path p_{gt}. Under mild assumptions, we propose a theoretical framework that more accurately reflects the real-world Hi-C situation, and we demonstrate that with high probability the output path $\hat{p}$ is equivalent to p_{gt}, as long as the number of mapped read pairs is $\Omega(|p_{gt}| \log |V|)$. Although the number of total nodes $|V|$ in the graph can be large, the required number of read pairs for a successful inference is only proportional to the logarithm of it. The details of this theoretical analysis can be found in Shen et al. [37]. We observe that in practice, this criterion regarding the number of read pairs is readily met. For instance, in our experiments, the graph has approximately 5×10^5 nodes, and the total number of mapped read pairs is around 3×10^8. This result provides some theoretical justification for the choice of the heuristic in Algorithm 1.

2.7 Practical Improvements to Efficiency and Accuracy

In practice, we introduce two modifications to our heuristic algorithm to enhance its accuracy and speed. First, given that the size of TADs typically does not exceed 3 Mb [3], we implement an additional heuristic adjustment to the dynamic program. When calculating the function q, we restrict our consideration to paths where the combined length of the DNA sequences on the nodes is under 3 Mb. That is, we replace P_{kl} in Eq. (4) and (5) with $\bar{P}_{kl}$, the collection of paths from k to l that are no more than 3 Mb. Let L denote the largest length of the paths in $\bar{P}_{kl}$, where length here refers to the number of nodes; generally,

$L \sim \frac{3Mb}{k_{bin}} \ll |V|$. Now the time complexity of the dynamic program when using the heuristic algorithm for q becomes $\mathcal{O}(|E||V| + L^4)$, where $\mathcal{O}(|E||V|)$ comes from computing the reachable matrix M_r.

Second, our empirical observations suggest that for most node pairs (k, l), computing q using Algorithm 1 is quite effective. Nonetheless, this method might not adequately capture the signals of large deletions. To mitigate this, we have refined Algorithm 1, as detailed in Algorithm 3. In this adjustment, for each node pair (k, l), we initially execute a node-weighted shortest path algorithm (where each node's weight is determined by the length of its corresponding DNA sequence) to identify a path p_{base} and calculate its score (lines 5–6 of Algorithm 3). Subsequently, Algorithm 1 is applied; if the path p derived from Algorithm 1 surpasses the score of p_{base}, p is returned, otherwise p_{base} is the selected path.

The shortest path algorithm for directed graphs with nonnegative weights has a time complexity of $\mathcal{O}(|E| + |V| \log(|V|))$. Consequently, the overall time complexity for Algorithm 3 to estimate q remains $\mathcal{O}(|V|^2)$ (or $\mathcal{O}(|L|^2)$ if we use the heuristic above), equating to that of Algorithm 1. Additionally, given that the path generated by Algorithm 3 will always yield a higher score compared to that from Algorithm 1, all the theoretical results in Sect. 2.6 are applicable to Algorithm 3 as well.

Algorithm 3. $q(k, l)$ computation v.2

1: **Input** k,l, genome directed acyclic graph $G = (V, E)$, nodes list T that contains all nodes in G sorted by their topological order, contact matrix M, reachable matrix M_r, the function f
2: **if** $M_r[k, l] = 0$ **then**
3: **return**
4: **end if**
5: $p_{base} \leftarrow$ shortest path from k to l
6: $q_{base} \leftarrow f(p_{base})$
7: $q, p \leftarrow$ Algorithm 1
8: **if** $q > q_{base}$ **then**
9: **return** q, p
10: **else**
11: **return** q_{base}, p_{base}
12: **end if**

3 Experimental Results

3.1 Construction of a Genome Graph with Hi-C Reads Mapped

We construct the genome graph with structural variations from the K-562 cancer cell line reported by Zhou et al. [43] and the linear reference genome GRCh37,

against which the SVs were called. We primarily use the VG toolkit [15] to incorporate simple variants and further process the variant file and the resulting graph so that the final genome graph is a directed acyclic graph. Details on the construction of the genome graph can be found in Shen et al. [37].

We apply the graph-based Hi-C pipeline [36] to process the raw Hi-C reads of the K-562 cancer cell line from Rao et al. [31] (accession number: SRR1658693). Subsequently, we employed our graph-based heuristic dynamic programming algorithm to infer the sample genome. Finally, all raw Hi-C reads were remapped to the newly inferred linear genome to generate chromosome-specific contact matrices with bin size 10 kb. Detailed descriptions of our algorithmic implementations, including hyper-parameter configurations, are provided in Shen et al. [37].

3.2 Graph Hi-C Workflow Improves TAD Identification

Table 1. The comparisons of three metrics reflecting CTCF or SMC3 enrichments near TAD boundaries across different genomes. Linear reference: linear reference genome; Reconstruction: genome inferred by our algorithm. TADs are called by Armatus with hyper-parameter $\gamma_{Ar} = 0.5$. Hi-C sample: SRR1659693.

		Average peak	Boundary tagged ratio	Fold change
CTCF	Linear reference	0.172	0.346	0.019
	Reconstruction	**0.202**	**0.387**	**0.144**
SMC3	Linear reference	0.091	0.194	0.044
	Reconstruction	**0.115**	**0.237**	**0.249**

We assess the quality of the new contact matrices from the inferred genome by their ability to exhibit biologically sound TAD structures. We use Armatus [10] to identify TADs from these matrices. The detected TADs are then compared with those identified from contact matrices created from the linear reference using HiC-Pro. We evaluate the quality of TADs against the enrichment of regulatory elements CTCF and SMC in K-562 cell lines around detected TAD boundaries.

We measure the enrichment of CTCF and SMC3 around TAD boundaries with three metrics: average peak, boundary tagged ratio, and fold change (Table 1). Average peak measures the average occurrence frequency of peaks within 30 kb range centered on TAD boundaries. Boundary tagged ratio measures the frequency of TAD boundaries that are enriched for regulatory elements. Fold change measures the change of enrichment of regulatory elements between regions around and far away from TAD boundaries. Since the TAD boundaries identified using our method are situated along a path within the genome graph, we use Graph Peak Caller [18] for calling and comparing CTCF and SMC3 peaks

on the graphs. Further details on graph peak calling and these metrics can be found in Shen et al. [37].

Both CTCF and SMC3 peaks are more concentrated around TAD boundaries identified based on the inferred genomes than linear reference. Figure 2 graphically presents these peak signals around TAD boundaries, clearly showing that the signals from the inferred linear genome are more pronounced than those from the linear reference. Table 1 shows that, relative to the linear reference, there is a higher enrichment of CTCF and SMC3 signals near the TAD boundaries identified from the new contact matrices.

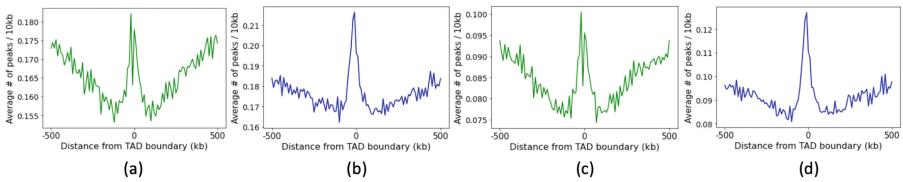

Fig. 2. (a), (b) CTCF peak signals around TAD boundaries from the linear reference genome (a) and the inferred linear genome (b). (c), (d) SMC3 peak signals around TAD boundaries from the linear reference genome (c) and the inferred linear genome (d). TADs are called by Armatus with hyper-parameter $\gamma_{Ar} = 0.5$.

4 Discussion

In this study, we establish a novel connection between Hi-C analysis and genome graphs and explore a novel application domain in pan-genomics. We develop the first algorithm that leverages genome graphs for inferring genome sequences from Hi-C reads. The experimental results demonstrate that the genomes inferred via our algorithm facilitate the creation of superior Hi-C contact matrices compared to those derived using a linear reference. These promising outcomes highlight the ability of genome graphs to enhance Hi-C analysis, especially for cancer samples that contain large-scale structural variations.

There are several avenues for future research stemming from this work. First, our dynamic programming algorithm, despite its reliance on heuristics, is not exceptionally fast. For instance, processing chromosome 1 with our algorithm requires around two days, even with some parallelism techniques applied. Accelerating our algorithm could be a fruitful area of exploration. Second, currently the normalization method for Hi-C data mapped onto graphs is lacking, which is crucial for correcting inherent biases. As a result, to ensure equitable comparisons, all contact matrices presented in the experimental section are unnormalized. Developing new methodologies for normalizing graph-based Hi-C data could be a vital and intriguing direction for future research. Third, our current approach is applicable only to DAGs. This limitation prevents us from testing

these methods on more complex non-directed acyclic graphs, such as the human pangenome graphs created by Liao et al. [24]. Therefore, adapting our methodology for use with general graphs represents a significant and necessary direction for future research. Additionally, given that our algorithm is applicable not only to cancer cell lines, it would be interesting to test it on more cell types, particularly normal ones, to evaluate its performance. Finally, while there has been research like that by Wang et al. [42] focusing on identifying structural variations from Hi-C data and rearranging contact matrices accordingly, we choose not to benchmark our method against theirs. This is because our primary aim in this work is to introduce the use of genome graphs in Hi-C analysis for the first time, while the method of Wang et al., although it can create improved Hi-C contact matrices, is not able to be used on genome graphs. In the future, it would be interesting to explore the integration of these two approaches, potentially leading to even more substantial improvements in Hi-C analysis.

Acknowledgments. This work was supported in part by the US National Science Foundation [DBI-1937540, III-2232121], the US National Institutes of Health [R01HG01-2470] and by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program.

Disclosure of Interests. C.K. is a co-founder of Ocean Genomics, Inc.

References

1. Ameur, A.: Goodbye reference, hello genome graphs. Nat. Biotechnol. **37**(8), 866–868 (2019)
2. Ay, F., Bailey, T.L., Noble, W.S.: Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts. Genome Res. **24**(6), 999–1011 (2014)
3. Bonev, B., Cavalli, G.: Organization and function of the 3D genome. Nat. Rev. Genet. **17**(11), 661–678 (2016)
4. Chin, C.S., et al.: Multiscale analysis of pangenomes enables improved representation of genomic diversity for repetitive and clinically relevant genes. Nat. Methods, 1–9 (2023)
5. De Laat, W., Duboule, D.: Topology of mammalian developmental enhancers and their regulatory landscapes. Nature **502**(7472), 499–506 (2013)
6. Dekker, J., Rippe, K., Dekker, M., Kleckner, N.: Capturing chromosome conformation. Science **295**(5558), 1306–1311 (2002)
7. Dixon, J.R., et al.: Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature **485**(7398), 376–380 (2012)
8. Ebler, J., et al.: Pangenome-based genome inference allows efficient and accurate genotyping across a wide spectrum of variant classes. Nat. Genet. **54**(4), 518–525 (2022)
9. Eggertsson, H.P., et al.: GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs. Nat. Commun. **10**(1), 5402 (2019)
10. Filippova, D., Patro, R., Duggal, G., Kingsford, C.: Identification of alternative topological domains in chromatin. Algorithms Mol. Biol. **9**, 1–11 (2014)
11. Foster, C.C.: A generalization of AVL trees. Commun. ACM **16**(8), 513–517 (1973)

12. Fraser, P., Bickmore, W.: Nuclear organization of the genome and the potential for gene regulation. Nature **447**(7143), 413–417 (2007)
13. Gabow, H.N., Maheshwari, S.N., Osterweil, L.J.: On two problems in the generation of program test paths. IEEE Trans. Softw. Eng. **3**, 227–231 (1976)
14. Garrison, E., et al.: Building pangenome graphs. bioRxiv, 2023–04 (2023)
15. Garrison, E., et al.: Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nat. Biotechnol. **36**(9), 875–879 (2018)
16. Gong, Y., Li, Y., Liu, X., Ma, Y., Jiang, L.: A review of the pangenome: how it affects our understanding of genomic variation, selection and breeding in domestic animals? J. Anim. Sci. Biotechnol. **14**(1), 1–19 (2023)
17. Grewal, S.I., Moazed, D.: Heterochromatin and epigenetic control of gene expression. Science **301**(5634), 798–802 (2003)
18. Grytten, I., Rand, K.D., Nederbragt, A.J., Storvik, G.O., Glad, I.K., Sandve, G.K.: Graph peak caller: calling ChIP-seq peaks on graph-based reference genomes. PLoS Comput. Biol. **15**(2), e1006731 (2019)
19. Hadi, K., et al.: Distinct classes of complex structural variation uncovered across thousands of cancer genome graphs. Cell **183**(1), 197–210 (2020)
20. Hickey, G., et al.: Pangenome graph construction from genome alignments with Minigraph-Cactus. Nat. Biotechnol., 1–11 (2023)
21. Kim, D., Paggi, J.M., Park, C., Bennett, C., Salzberg, S.L.: Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nat. Biotechnol. **37**(8), 907–915 (2019)
22. Kolman, P., Pangrác, O.: On the complexity of paths avoiding forbidden pairs. Discret. Appl. Math. **157**(13), 2871–2876 (2009)
23. Li, A., et al.: Decoding topologically associating domains with ultra-low resolution Hi-C data by graph structural entropy. Nat. Commun. **9**(1), 3265 (2018)
24. Liao, W.W., et al.: A draft human pangenome reference. Nature **617**(7960), 312–324 (2023)
25. Lieberman-Aiden, E., et al.: Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science **326**(5950), 289–293 (2009)
26. Nora, E.P., et al.: Spatial partitioning of the regulatory landscape of the X-inactivation centre. Nature **485**(7398), 381–385 (2012)
27. Pandey, P., Gao, Y., Kingsford, C.: VariantStore: an index for large-scale genomic variant search. Genome Biol. **22**(1), 1–25 (2021)
28. Pope, B.D., et al.: Topologically associating domains are stable units of replication-timing regulation. Nature **515**(7527), 402–405 (2014)
29. Qiu, Y., Kingsford, C.: Constructing small genome graphs via string compression. Bioinformatics **37**(Supplement_1), i205–i213 (2021)
30. Rakocevic, G., et al.: Fast and accurate genomic analyses using genome graphs. Nat. Genet. **51**(2), 354–362 (2019)
31. Rao, S.S., et al.: A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell **159**(7), 1665–1680 (2014)
32. Rennie, S., Dalby, M., van Duin, L., Andersson, R.: Transcriptional decomposition reveals active chromatin architectures and cell specific regulatory interactions. Nat. Commun. **9**(1), 487 (2018)
33. Roayaei Ardakany, A., Gezer, H.T., Lonardi, S., Ay, F.: Mustache: multi-scale detection of chromatin loops from Hi-C and Micro-C maps using scale-space representation. Genome Biol. **21**, 1–17 (2020)
34. Schöpflin, R., et al.: Integration of Hi-C with short and long-read genome sequencing reveals the structure of germline rearranged genomes. Nat. Commun. **13**(1), 6470 (2022)

35. Servant, N., et al.: HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. Genome Biol. **16**(1), 1–11 (2015)
36. Shen, Y., Yu, L., Qiu, Y., Zhang, T., Kingsford, C.: Improving Hi-C contact matrices using genome graphs. bioRxiv, 2023–11 (2023)
37. Shen, Y., Yu, L., Qiu, Y., Zhang, T., Kingsford, C.: Technical report: graph-based genome inference from Hi-C data (2023). https://github.com/Kingsford-Group/graphhic/blob/main/technicalreport.pdf
38. Sirén, J., et al.: Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. Science **374**(6574), abg8871 (2021)
39. Wang, S., et al.: HiNT: a computational method for detecting copy number variations and translocations from Hi-C data. Genome Biol. **21**, 1–15 (2020)
40. Wang, T., et al.: The human pangenome project: a global resource to map genomic diversity. Nature **604**(7906), 437–446 (2022)
41. Wang, X.T., Cui, W., Peng, C.: HiTAD: detecting the structural and functional hierarchies of topologically associating domains from chromatin interactions. Nucleic Acids Res. **45**(19), e163–e163 (2017)
42. Wang, X., et al.: Genome-wide detection of enhancer-hijacking events from chromatin interaction data in rearranged genomes. Nat. Methods **18**(6), 661–668 (2021)
43. Zhou, B., et al.: Comprehensive, integrated, and phased whole-genome analysis of the primary ENCODE cell line K562. Genome Res. **29**(3), 472–484 (2019)