Communications in Information and Systems
Volume 21, Number 1, 1–29, 2021

Fast random algorithms for manifold based optimization in reconstructing 3D chromosomal structures

Duan Chen*, Shaoyu Li, Xue Wang, and Kelin Xia

Inferring 3D structures of chromatins and chromosomes from experimental data is critical to understand biological functions of DNAs, and various computational algorithms have been developed in the past a few years. All algorithms are subject to the challenge of high computational cost if the number of loci in the target chromosome is large. In this paper, we tackle this difficulty and develop a set of fast algorithms for the manifold-based optimization (MBO) model, which is a popular method to reconstruct 3D chromosomal structures from Hi-C data. The proposed algorithms are based on random projection theory. We first approximate the column (row) space of the original data in a reduced dimension. Then interpolative decomposition technique is used to decompose the data matrix into a product of two matrices, each of which has a much smaller dimension comparing to the number of degree of freedom of the problem. With this low-rank approximation, all components in the gradient descent method of the optimization, including calculating gradient, line search, and solution updating, have the linear complexity, with respect to the total number of loci in the target chromosome. At last, a randomly perturbed gradient descent method is adopted so one can effectively escape saddle points of the non-convex optimization. In simulations, a synthetic simple helix and a simulated chromosomal structure are used to validate our algorithms, suggesting its highly enhanced efficiency and desired ability to recover structures from data subject to random lost and mild contamination of noises.

Keywords and phrases: Hi-C data, chromosomal structures, mathematical model, manifold based optimization, fast algorithms, randomized numerical linear algebra.

[*] Corresponding author.

1. Introduction

Genetic information of almost every living organism is encoded in deoxyribonucleic acid (DNA), which is critical for development and functions of the organism. DNAs organize into chromosomes and the collection of all chromosomes is called genome, existing in the nucleus of nearly all cells of alive eukaryotic organisms. Besides linear genetic information of DNA, it is crucial to understand and identify three dimensional (3D) structures of chromosomes (or their fundamental components, chromatins), because spatial organization of DNA essentially affects genome functions, such as transcription and its efficiency, spatial regulation, and genome interpretation [11, 21, 37, 39]. It is also meaningful from disease diagnosis and treatment, to drug design. 3D structures of chromatins and chromosomes can be inferred from experimentally obtained inter- and intra- chromosomal interactions. Chromosome conformation capture (3C) [10] with next-generation sequencing, including 5C [13], Hi-C [33, 44], TCC [29] and GCC [45] are able to quantitatively measure the number of interactions between genomic loci across large genomic regions or entire genomes. Processes of these techniques can be briefly summarized as the following [33, 40, 5, 22]. Pairs of chromosomal loci are first cross-linked, and then fragmented, with the size of restriction fragments determining the resolution of interaction mapping. After the next step of random ligation, interacting loci are quantified by amplifying ligated junction by PCR methods. High-throughput methods, such as Hi-C method, are able to quantify interactions between all possible pairs of genomic loci fragments simultaneously. Contact frequency, referring to the measured number of interactions between two loci in a population of cells, are typically presented as a matrix. The optimal 3D structures of chromosomes can be inferred from the data of frequency matrix, or Hi-C contact map, because it has (unknown) relation to the average *in vivo* 3D distance between loci and thus presumably reflects the average spatial organization of the corresponding chromosomes.

Recently, many computational algorithms have been developed to infer the coordinates of genomic loci in a chromosome from Hi-C contact matrices. A straightforward approach is the classic multidimensional scaling (MDS) [30]. In this method, the Hi-C data need to be first translated into a Euclidean distance matrix (EDM) that satisfying distance geometry properties such as triangle inequalities, then the EDM is linearly mapped to the Gram matrix. Eventually the coordinates are obtained by performing singular value decomposition (SVD) of the Gram matrix [12]. Forming the EDM in the first step is the most important process and a specific algorithm was proposed in [30]. Other model-based methods include inferring structures by maximum likelihood algorithms [42], Markov Chain Monte Carlo method [26, 46], and simulated annealing methods [48, 41, 3], etc. More generally, it can be formulated as a low-rank optimization problem for distance matrix completion [4, 3, 19, 52, 38, 28, 43]. The optimization-based methods are (almost) assumption-free, data-driven approaches that can handle nosiness, incompleteness, uncertainty of data in a systematic way. Thus, they are the major interests of our work. Regardless of methodologies, all numerical algorithms encounter challenge of great computational costs at high resolution Hi-C data. According to [44], recent experiments have been able to provide Hi-C data at resolution as high as 1-5 kilo-base pair (kbp) for several human lines. Comparing to the magnitude of billion-base pair of the whole chromosome, the number of loci in the data set, or the number of degree of freedom N, could easily reach the scale of $10^5 \sim 10^6$. At this scale, the computational cost and memory requirement are extremely high, if not prohibitive. For example, performing SVD of a N by N matrix is of complexity $O(N^3)$. In optimizations, it requires $O(N^2)$ complexity of matrix-vector multiplication in gradient calculation and line search, and this process is repeated in a large amount of overall iterations. On the other hand, due to the imperfectness of experiments, the contact frequency data is subject to sparseness, noisy nature, and experimental uncertainty. As consequence, fast and robust computational algorithms are indispensable and have priority to highly-accurate ones, to analyze Hi-C data at high resolution.

Challenges of high computational complexity can be tackled by random numerical linear algebra (RNLA) [18, 25]. Morden problems in applied mathematics (such as scientific computing in numerical partial differential equations (PDEs), integral equations (IEs), non-local interactions, or numerical linear algebra) and applied statistics (least-squares regression, quantile regression, machine learning) are usually associated with matrices at extra large scale. Using Monte Carlo methods, RNLA can provide computational algorithms for large-scale matrix operations with enhanced efficiency and controllable accuracy in high probability. Roughly speaking, there are two directions of RNLA. One is stochastic matrix approximation: comparing to its traditional deterministic counter parts such as SVD or QR, only *partial* information is extracted, from at most two passes, by some statistical strategies to approximate the original data matrix, such that fundamental matrix operations (e.g. matrix-vector multiplication) enjoy high efficiency (usually $O(N)$) and require less CPU memory. Specific algorithms include random matrix-vector multiplication [14], random SVD [15], random CUR decomposition [16, 17], or random interpolative decomposition (ID) [34, 35].

The other direction is subspace embedding: though random projection technique, the column (row) space of the original data matrix is embedded to a much smaller subspace with low distortion [51, 47]. As a result, either dimension or conditional number of the data matrix in regression problems is greatly reduced. In this approach, no matrix needs to be approximated but the overall iteration is speed-up. Various algorithms have been proposed for l_p regressions [9, 49, 7, 8, 6, 36, 50]. A broader review of RNLA can be found in [18, 25, 32].

The objective of this work is to develop high-efficient and reliable computational algorithms for optimization, to reconstruct 3D chromosomal structures from Hi-C data, using RNLA techniques. First, the problem is summarized as a manifold-based optimization (MBO) problem to the Hi-C data. In this work, we first assume a known *a prior* relation between loci distance and contact frequency as in other literatures, then the contact frequency matrix is converted to a "pseudo" distance matrix. Next, the data matrix is approximated as the product of two low dimension matrices. Thus, in the gradient descent method of solving the optimization problem, the computations of evaluating the objective function, gradient and step lengths all have complexity of $O(N)$ in *each iteration*. Additionally, a stochastic perturbation method is introduced to efficiently escape from the saddle points of the non-convex optimization. As analyzed in [23], approximation of the minimizer can be obtained in $O(poly(\log N))$ iterations. Therefore, the total computational complexity of our algorithm in time is $O(Npoly(\log N))$ and memory requirement is $O(N)$. Synthetic data and realistic Hi-C data will be used to verify the proposed algorithms.

The rest of the paper is organized as follows: Section 2 briefly reviews the manifold based optimization (MBO) used in Hi-C data analysis. In Section 3, a set of random algorithms, including random matrix low-rank approximation, the resulting gradient descent algorithm, and a saddle point escaping method, are introduced. Numerical results of the proposed fast algorithms in analyzing both simple synthetic and Hi-C data are presented in Section 4. The paper ends with conclusion in Section 5.

2. Manifold based optimization for chromosome structures

In this section we review the basic concepts of manifold based optimization (MBO) and how it is related to chromosome structure recovering. More detailed description of MBO can be found in [28, 38, 2, 43].

2.1. Manifold based optimization (MBO)

In this problem, the original or pre-processed experimental data is represented by a matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$, whose entries are measured as contact frequencies among loci of a chromosome. The size N of the so-called dissimilarity data depends on the experimental resolution (in terms of kilo-base-pair, or kbp). The fundamental assumption is that the contact frequency s_{ij} between loci i and j is related to their physical distance. Then the goal is to solve a matrix completion problem with a relation function $g : \mathbb{R} \to \mathbb{R}$, i.e., loosely speaking

$$\min_{\mathbf{D} \in \mathrm{EDM(N)}} \|\mathbf{D} - g(\mathbf{S})\|, \tag{1}$$

where the variable $\mathbf{D}$ is a $N \times N$ rigorous Euclidean distance matrix (EDM), whose entries are actually the distance-square of loci, i.e., d_{ij}^2. The matrix $g(\mathbf{S})$ is from entry-wise evaluation of $\mathbf{S}$, i.e., $g(\mathbf{S})_{ij} = g(s_{ij})$. A commonly used empirical choice [33] of the function g is

$$g(s_{ij}) = \begin{cases} s_{ij}^{-\alpha} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{2}$$

for some $0 < \alpha < 1$, while more recent work suggests treating the function g itself as unknown. The matrix norm $\|\cdot\|$ measures the difference between the data and prediction. For simplicity, the square of Frobenius norm $\|\cdot\|_F^2$ is used in this work. Note that in Eq. (1) the matrix $\mathbf{S}$ is subject to different levels of noise and missing entries.

The EDM $\mathbf{D}$ has a structure. Actually, consider the 3D structure of the target chromosome as $\mathbf{Y} \in \mathbb{R}^{N \times 3}$, with each row of $\mathbf{Y}$ being the (x, y, z) coordinates of the corresponding loci. Then its Gram matrix is defined as $\mathbf{G} = \mathbf{Y}\mathbf{Y}'$, where $\mathbf{Y}'$ is the transpose of $\mathbf{Y}$. Obviously it has rank of three and it has a relation with the EDM as

$$\mathbf{D} = \kappa(\mathbf{G}) = \mathrm{Diag}(\mathbf{G})\mathbf{1}' + \mathbf{1}\mathrm{Diag}(\mathbf{G})' - 2\mathbf{G}, \tag{3}$$

where $\mathrm{Diag}(\mathbf{G}) \in \mathbb{R}^{\mathrm{N}}$ is the column vector made of diagonal entries from $\mathbf{G}$ and $\mathbf{1} \in \mathbb{R}^N$ has all entries being one. With this relation, optimization problem (1) is on a low-dimensional space

$$\min_{\mathbf{G} \succeq 0} \|\kappa(\mathbf{G}) - g(\mathbf{S})\|_F^2, \qquad \text{s.t.} \qquad \mathrm{rank}(\mathbf{G}) = 3 \tag{4}$$

where $\mathbf{G} \succeq 0$ is positive semidefinite (PSD) matrix, from which the 3D structure $\mathbf{Y}$ will be recovered.

Directly solving Eq. (4) for the full matrix $\mathbf{G}$ is expensive and not necessary since the eventual goal is $\mathbf{Y}$. Indeed, solving $\mathbf{Y}$ can be investigated in the geometric framework of optimization on Riemannian manifolds of PSD matrices. A potential difficulty of solving $\mathbf{Y}$ is that $\mathbf{G} = \mathbf{Y}\mathbf{Y}'$ is invariant with respect to the transformation $\mathbf{Y} \mapsto \mathbf{Y}\boldsymbol{\Theta}$, where $\boldsymbol{\Theta} \in \mathcal{O}(3) = \{\boldsymbol{\Theta} \in \mathbb{R}^{3\times 3} : \boldsymbol{\Theta}'\boldsymbol{\Theta} = \boldsymbol{\Theta}\boldsymbol{\Theta}' = \mathbf{I}\}$. This property implies that the minima of the cost function Eq. (4) are not isolated. To address this theoretical issue, the problem is reformulated as an optimization problem on the quotient manifold defined as

$$\mathcal{M} \doteq S_+(3, N) \simeq \mathbb{R}_*^{n\times 3}/\mathcal{O}(3), \tag{5}$$

where $S_+(3, N) = \{\mathbf{X} \in \mathbb{R}^{N\times N} : \mathbf{X} = \mathbf{X}' \succeq 0, \operatorname{rank}(\mathbf{X}) = 3\}$ and $\mathbb{R}_*^{N\times 3} = \{\mathbf{Y} \in \mathbb{R}^{N\times 3} : \det(\mathbf{Y}'\mathbf{Y}) \neq 0\}$. The manifold $\mathcal{M}$ represents a set of equivalence classes of $\mathbf{Y}$ as $[\mathbf{Y}] = \{\mathbf{Y}\boldsymbol{\Theta} : \boldsymbol{\Theta} \in \mathcal{O}(3)\}$.

Based on this manifold, Eq. (4) can be reformulated as an unconstrained optimization with cost function

$$\min_{[\mathbf{Y}]\in\mathcal{M}} f([\mathbf{Y}]) = \min_{[\mathbf{Y}]\in\mathcal{M}} \|\kappa(\mathbf{Y}\mathbf{Y}') - \tilde{\mathbf{D}}\|_F^2. \tag{6}$$

where we denote $\tilde{\mathbf{D}} = g(\mathbf{S})$ for convenience. In this current work, we simply take Eq. (2) for the function g, as in [52, 33]. Note that matrix $\tilde{\mathbf{D}}$ from actual data is not a rigorous EDM; it is rather referred as *distance matrix data*.

It is important to note that the cost function f is defined on the manifold other than an Euclidean space [28]. Computational algorithms for Eq. (6) are established conceptually on the entire quotient space $\mathcal{M}$ but practically in $\mathbb{R}_*^{n\times 3}$. Both the first-order gradient descent algorithm and the second-order trust region method can be used to solve the MBO. But in practice, we will focus on the gradient descent method because the invariance of solution with respect to rotation greatly impacts the convergence properties of second-order method but is not harmful for the first-order methods [1, 2]. Indeed, the trust-region algorithm can only achieve superlinear convergence for (6) with carefully tuned parameters. Additionally, gradient descent method is more straightforward for the non-convex optimization as (6) at large scale, while the saddle-point escaping algorithms for trust-region methods still remain open. At last, it is convenient to construct fast algorithms for gradient descent methods. The last two issues will be further discussed in Section 3.

2.2. Gradient descent method on a manifold

The gradient of the cost function (at $[\mathbf{Y}]$) on a manifold $\mathcal{M}$ is a vector in its tangent space $T_{\mathbf{Y}}\mathcal{M}$. For any two vectors $\xi_{\mathbf{Y}}$ and $\eta_{\mathbf{Y}}$ in $T_{\mathbf{Y}}\mathcal{M}$, the Riemannian metric is

$$h_{\mathbf{Y}}(\xi_{\mathbf{Y}}, \eta_{\mathbf{Y}}) = \mathrm{Tr}(\xi'_{\mathbf{Y}}\eta_{\mathbf{Y}}), \tag{7}$$

where Tr is for the trace operator. The tangent space can be further decomposed to two orthogonal subspaces: one is the vertical space $\mathcal{V}_{\mathbf{Y}}\mathcal{M}$ and the other is the horizontal space $\mathcal{H}_{\mathbf{Y}}\mathcal{M}$. The former is tangent to the equivalence class $[\mathbf{Y}]$, i.e.

$$\mathcal{V}_{\mathbf{Y}}\mathcal{M} = \{\mathbf{Y}\boldsymbol{\Xi} : \boldsymbol{\Xi} \in \mathbb{R}^{3\times 3}, \boldsymbol{\Xi}' = -\boldsymbol{\Xi}\}, \tag{8}$$

while the latter is its orthogonal complement, i.e.

$$\mathcal{H}_{\mathbf{Y}}\mathcal{M} = \{\bar{\xi}_{\mathbf{Y}} \in \mathbb{R}^{n\times 3} : \bar{\xi}'_{\mathbf{Y}}\mathbf{Y} = \mathbf{Y}'\bar{\xi}_{\mathbf{Y}}\}, \tag{9}$$

Then the skew-symmetric matrix defines a projection of an arbitrary element $\xi \in \mathbb{R}^{N\times 3}$ onto the horizontal space $\mathcal{H}_{\mathbf{Y}}\mathcal{M}$ by $\Pi_{\mathcal{H}_{\mathbf{Y}}}(\xi) = \xi - \mathbf{Y}\boldsymbol{\Xi}$, and $\boldsymbol{\Xi}$ satisfies the Sylvester equation

$$\boldsymbol{\Xi}\mathbf{Y}'\mathbf{Y} + \mathbf{Y}'\mathbf{Y}\boldsymbol{\Xi} = \mathbf{Y}'\xi - \xi'\mathbf{Y}. \tag{10}$$

Then the gradient of the cost function on the manifold is the unique tangent vector in $T_{\mathbf{Y}}\mathcal{M}$ that is projected from the gradient of f with respected to $\mathbf{Y} \in \mathbb{R}^{N\times 3}$. In order to implement line search and update the search variable on the manifold, a local mapping from $T_{\mathbf{Y}}\mathcal{M}$ to the manifold, or a retraction is required. According to [28], a simple choice of retraction for the quotient manifold can be

$$\mathbf{R}_{\mathbf{Y}}(\bar{\xi}_{\mathbf{Y}}) = \mathbf{Y} + \bar{\xi}_{\mathbf{Y}}. \tag{11}$$

Computationally, the gradient $\mathrm{grad} f(\mathbf{Y})$ satisfies

$$h_{\mathbf{Y}}(\xi_{\mathbf{Y}}, \mathrm{grad} f(\mathbf{Y})) = Df(\mathbf{Y})[\xi_{\mathbf{Y}}], \quad \forall \xi_{\mathbf{Y}} \in T_{\mathbf{Y}}\mathcal{M}, \tag{12}$$

where the quantity $Df(\mathbf{Y})[\xi_{\mathbf{Y}}]$ is the directional derivative of f on the manifold in the direction of $\xi_{\mathbf{Y}}$, i.e.,

$$Df(\mathbf{Y})[\xi_{\mathbf{Y}}] = \lim_{t\to 0} \frac{f(\mathbf{Y} + t\xi_{\mathbf{Y}}) - f(\mathbf{Y})}{t}. \tag{13}$$

Then from the cost function in Eq. (6), the gradient is calculated as

$$\text{grad} f(\mathbf{Y}) = 2\kappa^*(\kappa(\mathbf{Y}\mathbf{Y}') - \tilde{\mathbf{D}})\mathbf{Y}, \tag{14}$$

where $\kappa^*(\mathbf{X})$ is the adjoint operator of κ and $\kappa^*(\mathbf{X}) = 2(\text{Diag}(\mathbf{X}\mathbf{1}) - \mathbf{X})$.

Given solution $\mathbf{Y}_i$ at the i-th step, using the retraction in (11), the gradient descent algorithm for (6) reads

$$\mathbf{Y}_{i+1} = \mathbf{Y}_i - 2\lambda_i\kappa^*(\kappa(\mathbf{Y}_i\mathbf{Y}'_i) - \tilde{\mathbf{D}})\mathbf{Y}_i. \tag{15}$$

The step size $\lambda_i > 0$ is determined by line search algorithms with the Armijo criterion, i.e.

$$f(\mathbf{Y}_i - \lambda_i \text{grad} f(\mathbf{Y}_i)) \leq f(\mathbf{Y}_i) - c_s\|\text{grad} f(\mathbf{Y}_i))\|_F^2 \tag{16}$$

for some parameter $0 < c_s < 1$.

3. Fast computational algorithms

Numerically solving the MBO encounters the following difficulties: (i) complexities of calculating all components, including cost function evaluation, computing gradient, and line search are $O(N^2)$. This could be extremely expensive for handling very high resolution Hi-C data. (ii) The MBO (6) is a non-convex optimization problem. Fortunately, it has been proved in [24] that all local minima are equivalent to the absolute minimum for this type of optimization, while special care needs to be taken to effectively escape saddle points in order to avoid high computational cost. In this section, we introduce fast computational algorithms to address these issues, including random low-rank approximation of matrix data and randomly perturbed saddle point escaping method.

3.1. Randomized low-rank matrix approximation

We are motivated by the special structures in the gradient method (14)-(15): according to the relation (3), the unknown EDM $\kappa(\mathbf{Y}\mathbf{Y}')$ in Eq. (6) is actually at most of rank 5. Since the data $\tilde{\mathbf{D}}$ is supposed to be "close" to the EDM, its rank is significantly smaller than its dimension. So we will try to represent the distance matrix data $\tilde{\mathbf{D}}$ in low-rank approximation, i.e., $\tilde{\mathbf{D}} \approx \mathbf{L}\mathbf{R}$, where $\mathbf{L} \in \mathbb{R}^{N\times k}$ and $\mathbf{R} \in \mathbb{R}^{k\times N}$. The parameter $k \ll N$ is called the numerical rank, which is usually larger than 5 since the data distance matrix is subject to missing entries or noises. With such an approximation, complexity of matrix operations, such as matrix-vector multiplication can be greatly reduced.

Traditional methods, such as truncated singular value decomposition (SVD) can accomplish this low-rank approximation. However, these deterministic methods could be even more expensive than the optimization problem itself when N is large. For example, it is well-known that the complexity of SVD is $O(N^3)$. Further, it is difficult to implement SVD or LU for Hi-C data since a large portion of entries of the distance matrix data are not available. To tackle these challenges, we proposed to use random projection method to obtain a low-rank approximation of the original data matrix more efficiently, without the need of all matrix entries.

Generally, to achieve a low-rank approximation of the matrix $\mathbf{A}$, a random matrix $\mathbf{\Omega} \in \mathbb{R}^{N\times(k+p)}$ is first multiplied to it to form a projection, i.e.,

$$\Pi_C \mathbf{A} = \mathbf{A}\mathbf{\Omega}, \tag{17}$$

where $k \ll N$ is the targeted rank and p (usually less than 10) is the oversampling parameter. A conventional choice of $\mathbf{\Omega}$ is an $N \times (k+p)$ normalized Gaussian matrix. In the second step, QR decomposition is applied to $\Pi_C\mathbf{A}$ to obtain the orthornomal vectors denoted by $\mathbf{Q} \in \mathbb{R}^{N\times(k+p)}$. Then a direct approximation is formed, $\mathbf{L} = \mathbf{Q}$, $\mathbf{R} = \mathbf{Q}'\mathbf{A}$, and $\mathbf{A} \approx \mathbf{LR}$.

However, this straightforward approximation is not quite efficient since computations for $\Pi_C\mathbf{A}$ and $\mathbf{R}$ are still of complexity $O(N^2)$. To solve the first issue, a very fast random projection method [31] can be applied, in which the random matrix $\mathbf{\Omega}$ is taken as

$$\Omega_{ij} = \begin{cases} -1 & \text{with probability } \frac{1}{2\sqrt{s}} \\ 0 & \text{with probability } 1 - \frac{1}{\sqrt{s}} \\ 1 & \text{with probability } \frac{1}{2\sqrt{s}} \end{cases}. \tag{18}$$

With such choice, only a small portion of the original data is needed since majority of $\mathbf{\Omega}$ entries are zeros by controlling the parameter s, which could be as large as N. Thus, the computational efficient is greatly enhanced despite little loss in accuracy.

The efficiency of random projection method can be further improved by combining with interpolative decomposition (ID) [34, 35]. In this approach, one does not need to compute $\mathbf{R} = \mathbf{Q}'\mathbf{A}$. Instead, an ID is implemented on the small matrix $\mathbf{Q}$, such that $\mathbf{Q} \approx \mathbf{XQ}(\mathbf{J}, :)$, where $\mathbf{X} \in \mathbb{R}^{N\times k}$ contains a $k \times k$ identity matrix, i.e., $\mathbf{X}(\mathbf{J}, :) = \mathbf{I}_k$ and other entries bounded by two, and $\mathbf{J}$ is a subset of row index of $\mathbf{Q}$ with size k. Hence there obtains a more efficient way to approximate the data, i.e.

$$\mathbf{A} \approx \mathbf{Q}\mathbf{Q}'\mathbf{A} \approx \mathbf{X}\mathbf{Q}(\mathbf{J},:)\mathbf{Q}'\mathbf{A} \approx \mathbf{X}\mathbf{A}(\mathbf{J},:). \tag{19}$$

Note that in the last approximation of Eq. (19), no matrix-vector multiplication is needed. One only needs to extract $\mathbf{J}$ entries of $\mathbf{A}$ and the ID is implemented on the small matrix $\mathbf{Q}$. The above description of low rank approximation is summarized in Algorithm 1.

Algorithm 1 Random Low-rank approximation of distance matrix

```
Input: Raw data: distance matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, parameter $k, p \ll N$.
Output: Matrices $\mathbf{L} \in \mathbb{R}^{N\times k}$ and $\mathbf{R} \in \mathbb{R}^{k\times N}$ such that $\mathbf{A} \approx \mathbf{LR}$.
  1. Random projection with oversampling: $\Pi_C\mathbf{A} = \mathbf{A}\mathbf{\Omega}$, where $\mathbf{\Omega} \in \mathbb{R}^{N\times(k+p)}$ being the conventional Gaussian matrix or the very fast projector in (18);
  2. QR decomposition of $\Pi_C\mathbf{A}$ to obtain the reduced approximation of orthonormal column basis $\mathbf{Q} \in \mathbb{R}^{N\times(k+p)}$ of $\mathbf{A}$;
  3. Apply ID to $\mathbf{Q}$ to obtain $\mathbf{X}$ and $\mathbf{J}$
  4. Return $\mathbf{L} = \mathbf{X}$ and $\mathbf{R} = \mathbf{A}(\mathbf{J},:)$
```

3.2. MBO with low-rank approximation

With such low-rank approximation of data $\tilde{\mathbf{D}} \approx \mathbf{LR}$, the MBO (6) can be implemented efficiently. For the gradient defined in (14), now we have approximation

$$\mathrm{grad} f(\mathbf{Y}) \approx 2\kappa^*(\kappa(\mathbf{Y}\mathbf{Y}') - \mathbf{LR})\mathbf{Y}. \tag{20}$$

It is obvious that computation of the right hand side of (20) is of complexity $O(k^2N)$. We only need to reformulate the computation of the cost-function:

Generally, for a matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$, let $\mathbf{A}^{(j)} \in \mathbb{R}^M$ and $\mathbf{A}_{(i)} \in \mathbb{R}^N$ denote its j-th column and i-th row, respectively. For any matrices $\mathbf{A} \in \mathbb{R}^{M\times N}$ and $\mathbf{B}^{N\times p}$, we have

$$\mathbf{AB} = \sum_{k=1}^{N} \mathbf{A}^{(k)}\mathbf{B}_{(k)} \tag{21}$$

Then if $\mathbf{A} \in \mathbb{R}^{M\times k}$, $\mathbf{C} \in \mathbb{R}^{k\times k}$ and $\mathbf{B} \in \mathbb{R}^{k\times N}$, the trace of $\mathbf{ACB}$ can be calculated as

$$\begin{aligned}\mathrm{Tr}(\mathbf{ACB}) &= \mathrm{Tr}\left(\sum_{i,j=1}^{k} c_{ij}\mathbf{A}^{(i)}\mathbf{B}_{(j)}\right) = \sum_{i,j=1}^{k} c_{ij}\mathrm{Tr}\left(\mathbf{A}^{(i)}\mathbf{B}_{(j)}\right) \\ &= \sum_{i,j=1}^{k} c_{ij}\langle \mathbf{A}^{(i)}, \mathbf{B}'_{(j)}\rangle, \qquad (22)\end{aligned}$$

where $\langle\cdot,\cdot\rangle$ represents the inner product of two column vectors. In this way, the cost function can be evaluated as

$$\begin{aligned}\|\kappa(\mathbf{YY}') - \mathbf{LR}\|_F^2 &= \mathrm{Tr}\left((\kappa(\mathbf{YY}') - \mathbf{LR})'(\kappa(\mathbf{YY}') - \mathbf{LR})\right) \\ &= \mathrm{Tr}\left(\kappa(\mathbf{YY}')'\kappa(\mathbf{YY}')\right) - 2\mathrm{Tr}\left(\mathbf{R}'\mathbf{L}'\kappa(\mathbf{YY}')\right) \\ &+ \mathrm{Tr}(\mathbf{R}'\mathbf{L}'\mathbf{LR}) \qquad (23)\end{aligned}$$

By definition (3) and denote $\mathbf{V} = \mathrm{Diag}(\mathbf{YY}') \in \mathbb{R}^{\mathrm{N}\times 1}$, the first term in the traces of Eq. (23) reads

$$\begin{aligned}\kappa(\mathbf{YY}')'\kappa(\mathbf{YY}') &= (\mathrm{Diag}(\mathbf{YY}')\mathbf{1}' + \mathbf{1}\mathrm{Diag}(\mathbf{YY}')' - 2\mathbf{YY}')^2 \\ &= 4\mathbf{Y}(\mathbf{Y}'\mathbf{Y})\mathbf{Y}' + \mathbf{V}(\mathbf{1}'\mathbf{V})\mathbf{1}' + \mathbf{1}(\mathbf{V}'\mathbf{1})\mathbf{V}' + 2\mathbf{V}(\mathbf{1}'\mathbf{1})\mathbf{V}' \\ &- 4\mathbf{V}(\mathbf{YY}')(\mathbf{1}'\mathbf{Y})\mathbf{Y}' - 4\mathbf{Y}(\mathbf{Y}'\mathbf{1})\mathbf{V}', \qquad (24)\end{aligned}$$

where we have used the fact $\mathrm{Tr}(\mathbf{AA}') = \mathrm{Tr}(\mathbf{A}'\mathbf{A})$. The second term in Eq. (23) is

$$\kappa(\mathbf{YY}')\mathbf{LR} = -2\mathbf{Y}(\mathbf{Y}'\mathbf{L})\mathbf{R} + \mathbf{V}(\mathbf{1}'\mathbf{L})\mathbf{R} + \mathbf{1}(\mathbf{V}'\mathbf{L})\mathbf{R}. \qquad (25)$$

Using Eqs. (24)-(25), computations of all traces in Eq. (23) can reduced to complexity of $O(k^2N)$ based on Eq. (22).

3.3. Error analysis

Low-rank approximation of matrix data can significantly improve the computational efficiency, but we need to know how different the resulting 3D structure from the one using original data. In this section we present some preliminary error analysis. Define

$$\mathbf{Y}_* = \arg\min_{[\mathbf{Y}]\in\mathcal{M}} \|\kappa(\mathbf{YY}') - \tilde{\mathbf{D}}\|_F^2 \qquad (26)$$

and

$$\tilde{\mathbf{Y}}_* = \arg\min_{[\mathbf{Y}]\in\mathcal{M}} \|\kappa(\mathbf{Y}\mathbf{Y}') - \mathbf{LR}\|_F^2 \tag{27}$$

as the recovered 3D structures from the original data and its low-rank approximation, respectively. Since the solution is invariant to rotation, it is more reasonable to check the difference between the corresponding EDMs defined by $\mathbf{Y}_*$ and $\tilde{\mathbf{Y}}_*$. Denote the EDM constructed by $\mathbf{Y}_*$ as $\mathcal{E}(\mathbf{Y}_*) = \kappa(\mathbf{Y}_*\mathbf{Y}_*')$ and $\mathcal{E}(\tilde{\mathbf{Y}}_*)$ is defined similarly. Then we have the following result:

Theorem 3.1. *Assume* **LR** *is the low rank approximation of the original data $\tilde{D}$ by Algorithm 1, $\mathbf{Y}_*$ and $\tilde{\mathbf{Y}}_*$ are the recovered structures via Eqs. (26)-(27). Then we have the estimate*

$$\begin{aligned}\mathbb{E}\left[\|\mathcal{E}(\tilde{\mathbf{Y}}_*) - \mathcal{E}(\mathbf{Y}_*)\|_F\right] &\le \sqrt{\sum_{i=6}^{N}\sigma_i^2(\tilde{\mathbf{D}})} + \sqrt{\mathbb{E}\left[\sum_{i=6}^{k}\sigma_i^2(\mathbf{LR})\right]} \\ &+ C(N,k,p)\sqrt{\sum_{i=k}^{N}\sigma_i^2(\tilde{\mathbf{D}})},\end{aligned} \tag{28}$$

where $C(N,k,p) = \left(1+\frac{k}{p-1}\right)^{1/2}\left(1+\sqrt{k+4k(N-k)}\right)$, and $\mathbb{E}[\cdot]$ represent the expectation value.

Proof. First we have

$$\begin{aligned}\|\mathcal{E}(\tilde{\mathbf{Y}}_*) - \mathcal{E}(\mathbf{Y}_*)\|_F &\le \|\kappa(\mathbf{Y}_*\mathbf{Y}_*') - \tilde{\mathbf{D}}\|_F + \|\kappa(\tilde{\mathbf{Y}}_*\tilde{\mathbf{Y}}_*') - \mathbf{LR}\|_F \\ &+ \|\tilde{\mathbf{D}} - \mathbf{LR}\|_F\end{aligned} \tag{29}$$

For the first two norms in (29), recall the fact that the rank of $\kappa(\mathbf{Y}_*\mathbf{Y}_*')$ is exactly 5 and $\mathbf{Y}_*$ is the minimizer of the MBO, then we have

$$\|\kappa(\mathbf{Y}_*\mathbf{Y}_*') - \tilde{\mathbf{D}}\|_F = \sqrt{\sum_{i=6}^{N}\sigma_i^2(\tilde{\mathbf{D}})}, \tag{30}$$

according to the Eckart-Young theorem [20], where $\sigma_i^2(\tilde{\mathbf{D}})$ is the singular value of $\tilde{\mathbf{D}}$. By the same argument,

$$\|\kappa(\tilde{\mathbf{Y}}_*\tilde{\mathbf{Y}}_*') - \mathbf{LR}\|_F = \sqrt{\sum_{i=6}^{k}\sigma_i^2(\mathbf{LR})}. \tag{31}$$

Estimations in (30)-(31) purely depend on the inherent qualities of the data. The last term in (29) is about the low-rank approximation error. According to Theorem 10.5 in [25], if the fundamental random projection is applied with the conventional choice of Gaussian matrix, one has

$$\mathbb{E}\left[\|\tilde{\mathbf{D}} - \mathbf{Q}\mathbf{Q}'\tilde{\mathbf{D}}\|_F\right] \leq \left(1 + \frac{k}{p-1}\right)^{1/2} \sqrt{\sum_{i=k}^{N} \sigma_i^2(\tilde{\mathbf{D}})}, \tag{32}$$

Further, if the ID algorithm is used, the efficiency of the matrix compressing is enhanced while the error bound in (32) will be enlarged by a factor. By an argument similar to Lemma 5.1 in [25] and notice that we use the Frobenius norm here, this factor is $1+\sqrt{k+4k(N-k)}$. Combining this fact and estimates (30)-(31), we obtain (28) and finish the proof. □

Remark 1. Future work needs to be done to improve the result of Theorem 3.1, such that the second term in (28) is also bounded by data $\tilde{D}$. However, the third term in (28) is the dominant one and it heavily depends on $\sigma_i(\tilde{\mathbf{D}}), i \geq k$ if $\tilde{\mathbf{D}}$ involves noises. In practice, the second term is negligible when k is taken small.

Remark 2. Although a sharp result on $\|\tilde{\mathbf{Y}}_* - \mathbf{Y}_*\|_F$ is not provided, Eq. (28) indicates that the accuracy of the fast algorithm depends on the quality of the distance matrix data $\tilde{\mathbf{D}}$, i.e., how "close" it is to a perfect EDM. In the extreme case that $\tilde{\mathbf{D}}$ is indeed an EDM, then $rank(\tilde{\mathbf{D}}) = 5$ and the fast algorithm will obtain the same structure as the original MBO does.

Remark 3. Generally, we want to prepare matrix data $\tilde{\mathbf{D}}$ such that its numerical rank as low as possible, in order to reduce the magnitude of the third term on the right-hand side of Eq. (28). Actually, this should also be the principle for the original MBO method, since $\tilde{\mathbf{D}}$ is assumed as an imperfect EDM and a structure is to be determined to match it as much as possible.

3.4. Escaping saddle points

It is well-known that one loses the convexity of the optimization problem when switching to the MBO defined by Eq. (6) from the original matrix completion formulation (1). For non-convex cases, the first-order stationary points (points that make gradient zero) could be global minima, local minima, saddle points or even local maxima. Obtaining the global minimum could be very difficult. However, as studied in [24], all local minima in (6) are actually global minima. Nevertheless, the gradient descent method is not able to distinguish saddle points from local minima, so even we have achieved barely changed gradient, we may get stuck at saddling points, asymptotically or for a sufficiently long time. Consequently, the algorithm is not efficient if no special treatment is applied. In [38], a dimension-by-dimension updating method was used to escape from the saddle point. But this algorithm requires computing eigenvalues of large gradient matrix of (1), so it is expensive when the problem size is large. Here we follow the method developed in [27], and propose to use the randomly perturbed gradient descent method. Before the algorithm is presented in Algorithm 2, the following definitions are introduced as in [27].

Definition 1. *A differentiable function $f(\cdot)$ is l-smooth if*

$$\|\mathrm{grad} f(\mathbf{Y}_1) - \mathrm{grad} f(\mathbf{Y}_2)\| \leq \ell \|\mathbf{Y}_1 - \mathbf{Y}_2\| \tag{33}$$

for some $l > 0$.

Definition 2. *A differentiable function $f(\cdot)$ is ρ-Hessian Lipschitz if*

$$\|\mathrm{Hess} f(\mathbf{Y}_1) - \mathrm{Hess} f(\mathbf{Y}_2)\| \leq \rho \|\mathbf{Y}_1 - \mathbf{Y}_2\| \tag{34}$$

for some $\rho > 0$.

Definition 3. *For a ρ-Hessian Lipschitz function $f(\cdot)$, one says $\mathbf{Y}^*$ is a ϵ-second-order stationary point if for some small $\epsilon > 0$*

$$\|\mathrm{grad} f(\mathbf{Y}^*)\| \leq \epsilon, \quad \textit{and} \quad \lambda_{\min}\left(\mathrm{Hess} f(\mathbf{Y}^*)\right) \leq -\sqrt{\rho\epsilon}, \tag{35}$$

where $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of the matrix.

According to [27], problem (6) is ℓ-smooth and ρ-Hessian Lipschitz, thus we propose to take the following Algorithm 2. The idea is that if the gradient of the cost function remains small (as $\|\mathrm{grad} f(\mathbf{Y}_i)\| \leq c_1\epsilon$) for a long time t_{thres}, then a small perturbation ξ_i is applied, which is uniformly sampled from a N-dimensional ball $\mathbb{B}_0(c_2\epsilon)$ with radius $c_2\epsilon$. Meanwhile, the solution before perturbation are recorded as $\mathbf{Y}_{t_{\mathrm{pert}}}$, as well as the moment of perturbation t_{pert}. The iteration continues until the cost function can not be significantly reduced comparing to the latest $\mathbf{Y}_{t_{\mathrm{pert}}}$, i.e., $f(\mathbf{Y}_i) > f(\mathbf{Y}_{t_{\mathrm{pert}}}) - c_3\sqrt{\epsilon^3}$. This algorithm is intuitive and easy to implemented, while its analysis is not trivial. The rigorous convergence analysis and explanation of parameters can be found in [27]. The major conclusion about the total iteration steps of the algorithm is included in Theorem 3.2.

Algorithm 2 Randomly perturbed Gradient Descent

```
Input: Initial guess $\mathbf{Y}_0$, error tolerance $\epsilon$, parameters $c_1$, $c_2$, $c_3$, $t_{\text{thres}}$, initial perturbation time $t_{\text{pert}} = -t_{\text{thres}} - 1$.
Output: $\epsilon$-second-order stationary point $\mathbf{Y}^*$
for $i = 0, 1, ...$ do
    Calculate gradient $\text{grad} f(\mathbf{Y}_i) = 2\kappa^*(\kappa(\mathbf{Y}_i\mathbf{Y}_i') - \mathbf{LR})\mathbf{Y}_i$;
    if $\|\text{grad} f(\mathbf{Y}_i)\| \le c_1\epsilon$ and $i - t_{\text{pert}} > t_{\text{thres}}$ then
        Update perturbation time $t_{\text{pert}} \leftarrow i$;
        Record solution before perturbation $\tilde{\mathbf{Y}}_{t_{\text{pert}}} \leftarrow \mathbf{Y}_{t_{\text{pert}}}$;
        Perturb the solution $\mathbf{Y}_i \leftarrow \tilde{\mathbf{Y}}_{t_{\text{pert}}} + \xi_i$, where $\xi_i$ uniformly $\sim \mathbb{B}_0(c_2\epsilon)$
    end if
    if $i = t_{\text{pert}} + t_{\text{thres}}$ and $f(\mathbf{Y}_i) > f(\mathbf{Y}_{t_{\text{pert}}}) - c_3\sqrt{\epsilon^3}$ then
        Return $\mathbf{Y}^* = \mathbf{Y}_{t_{\text{pert}}}$;
        Break
    end if
    Perform line search to get step size $\eta_i$
    Compute $\mathbf{Y}_{i+1} = \mathbf{Y}_i - 2\eta_i\kappa^*(\kappa(\mathbf{Y}_i\mathbf{Y}_i') - \mathbf{LR})\mathbf{Y}_i$
end for
```

Theorem 3.2 ([27]). *Assume the cost function is ℓ-smooth and ρ-Hessian Lipschitz. Then for any constant $\delta > 0$, $\epsilon \le \ell^2/\rho$, $\Delta f \ge f(\mathbf{Y}_0) - f(\mathbf{Y}^*)$, the randomly perturbed gradient descent method will output an ϵ-second order stationary point, with probability $1 - \delta$, and terminate in the step number N_{iter} for which*

$$N_{\text{iter}} = O\left(\frac{\ell(f(\mathbf{Y}_0) - f(\mathbf{Y}^*))}{\epsilon^2} \log^4\left(\frac{N\ell\Delta f}{\epsilon^2\delta}\right)\right) \tag{36}$$

Theorem 3.2 indicates that with the corresponding parameters, one can escape saddle points and arrive the ϵ-second order stationary points in $O(\log^4 N)$ steps with high probability.

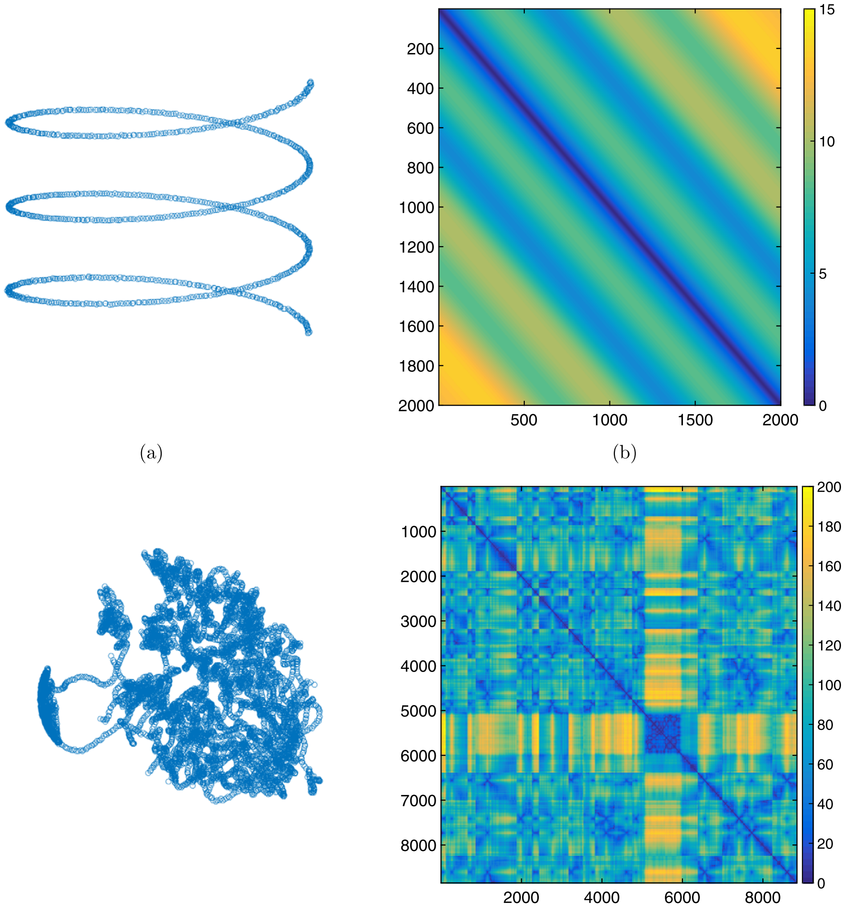
4. Numerical results

4.1. Data preparation

The efficiency and accuracy of the proposed fast MBO algorithms (FMBO) are demonstrated by a synthetic 3D helix data and a simulated chromosome structure of the yeast genome. The helix structure is calculated by

$$x(t) = 4\cos(3t); \quad y(t) = 4\sin 3t; \quad z(t) = 2t; \quad 0 \le t \le 2\pi, \tag{37}$$

while the latter is taken from [19]. The structures and their corresponding EDMs are shown in Fig. 1. It is easy to control the number of degree of freedom in the synthetic data for efficiency validation, while the chromosome structure is used to verify that our algorithm can handle complicated data.



(a) (b)

(c) (d)

Figure 1: 3D structures of simulations and the corresponding distance matrix data: (a)-(b) for a simple helix; (c)-(d) for a simulated chromosomal structure [19].

In order to simulate realistic situations, different levels of noises are artificially added in the EDMs in Fig. 1(b) and (d). To do this, we first generate an ideal EDM $\mathbf{D}$ by Eq. (3) with the synthetic helix or the known simulated chromosome structure. Then different levels of random noises are added to $\mathbf{D}$ in order to generate the actual distance matrix data $\tilde{\mathbf{D}}$. The "polluted" data is generated as $\tilde{D}_{ij} = (1+c\epsilon_{ij})D_{ij}$, where $0 < c < 1$ is called the noise level and ϵ_{ij} is a uniformly distributed random number in $[0, 1]$. We investigate the solution quality of the FMBO from two perspectives, the relative errors and Pearson's correlations between the numerical solution $\mathbf{Y}_*$ to the true structures $\mathbf{Y}$. Because of the earlier mentioned invariance of rotation, the relative error is defined as

$$\text{Err} = \frac{\|\mathbf{Y}_*\mathbf{Y}_*' - \mathbf{YY}'\|_{\text{F}}}{\|\mathbf{YY}'\|_{\text{F}}}, \tag{38}$$

and the Pearson's correlations has its classical definition on $\mathbf{Y}_*\mathbf{Y}_*'$.

The numerical rank k is a major parameter in Algorithm 1. So we first present how the low-rank approximations depend on the choices of k. For the data from simulated chromosome structures, mean relative errors defined in (38) are displayed in Table 1, with $c = 0.05$, $c = 0.25$, $c = 0.5$, and various values of numerical ranks. It can be concluded that the approximation errors do not heavily depend on k, but rather on the noise level, i.e. how close the data is close to a rank 5 EDM. These results verify the error bound in Theorem 3.1: approximation error majorly depends on $\sqrt{\sum_{i=k}^{N} \sigma_i^2(\tilde{\mathbf{D}})}$, which is larger if more noises are involved. On the other side, the coefficient $C(N, k, p)$ increases as k when $k < N/2$. Similar observations are obtained for the helix structure. Based on theses considerations, the parameters we take for the FMBO are $k = 5$, $p = 15$, $c_1 = 5$, $c_2 = 10$, $c_3 = 5$, $\epsilon = 1 \times 10^{-4}$, $t_{\text{thres}} = 200$ throughout the rest of simulations.

Table 1: Relative errors of low-rank approximation against k and noise level c

Noise level	c=0.05	c=0.25	c=0.5
$k = 5$	2.2%	6.2%	16.4%
$k = 10$	3.0%	6.7%	17.7%
$k = 20$	3.1%	8.2%	18.6%
$k = 40$	3.3%	9.1%	22.3%

4.2. Algorithm efficiency

Figure 2 displays the CPU time for the original MBO and the FMBO against the number of degree of freedom N. For better qualitative illustration, these relations are plotted in a log-log form. The red and green lines represent complexity relations for the original MBO and the FMBO, respectively. Theoretically, in each single step during iteration, the complexity is $O(N^2)$ for the MBO while $O(N)$ for the FMBO, but the iteration process makes the overall complexity higher. As illustrated in Theorem 3.2, the total iteration steps is $O(\log^4 N)$. For simplicity we estimate overall CPU time in terms of $O(N^\alpha)$, and investigate the number α by checking the slopes of lines that fit the log-log relation of the two groups of data against N. For the original MBO, the slope of the line fitting red dots is $\alpha = 2.5$, while the line fitting the green dots is estimated as $\alpha = 1.5$ for the FMBO. The two methods have the similar number $(O(N^{0.5}))$ of overall iteration steps, but the FMBO algorithm exhibits great efficiency $(O(N))$ over its original counter part $(O(N^2))$ in each step. The linear relation $O(N)$ is represented in a dashed black line of slope one for comparison. These results are for the synthetic helix data, where it is easy to control the number N.

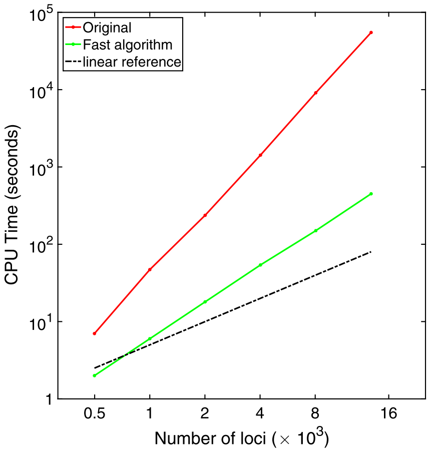
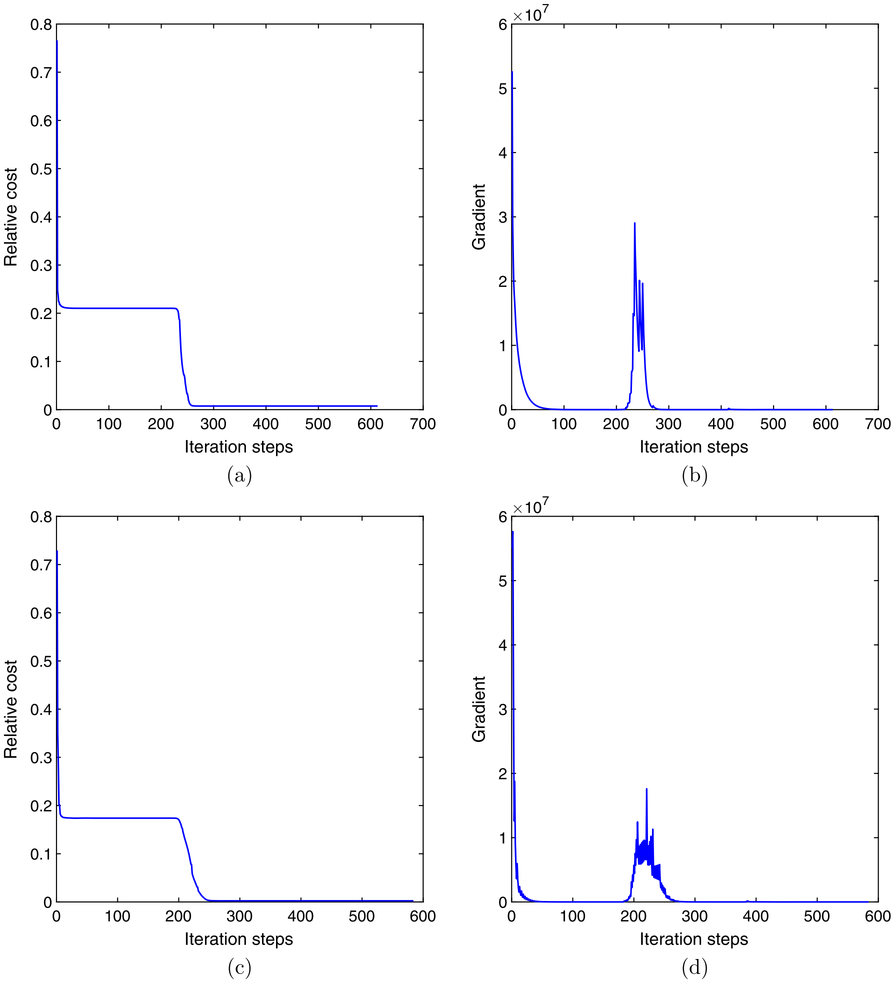


Figure 2: CPU time of algorithms (original optimization and fast optimization) agains number of loci N. The relation is rescaled as log-log and the linear relation $O(N)$ is plotted as dashed line for reference.

Figure 3 records the performance of gradient descent iteration of the MBO (top) and FMBO (bottom), respectively. Fig. 3(a) and (c) show the relative cost function $f([\mathbf{Y}])/\|\tilde{\mathbf{D}}\|_F^2$, and Fig. 3(b) and (d) represent the norm of the corresponding gradient during the iteration. With the same set of parameters, the original MBO and FMBO use similar number of steps to achieve the approximated minimizer. Further, it clearly shows that the random perturbation is necessary for these algorithms to escape from saddle points: in both cases, norms of gradient drop significantly within several iterations, and cost functions rapidly achieve equilibrium in the first 200 steps while remain fairly large. Random perturbations happen at the 241$^{\mathrm{st}}$ step in MBO and at the 205$^{\mathrm{th}}$ step in the FMBO, successfully helping the algorithm jump out of the equilibrium and obtain the actual minimum. These results of algorithm performances are for simulated chromosome data.



(a) (b)

(c) (d)

Figure 3: Relative value and gradients of cost functions during iteration processes. (a) Relative cost functions in the MBO; (b) Norm of gradient in the MBO; (c) Relative cost functions in the FMBO; (d) Norm of gradient in the FMBO.

4.3. Algorithm accuracy

Next we investigate the algorithm accuracy. According to analysis (28), both MBO and FMBO would perfectly recover the corresponding 3D structure (only subject to iteration error of optimization) in the ideal situation that the distance matrix data $\tilde{\mathbf{D}}$ were exactly an EDM. While in practice, the data $\tilde{\mathbf{D}}$ derived from experiment data is far from perfect, and it is subject to unknown noise and missing information. We want to examine (i) whether the efficient FMBO can actually recover the 3D structure accurately if the distance matrix data is noise free and (ii) the robustness of the FMBO at different levels of random noises.

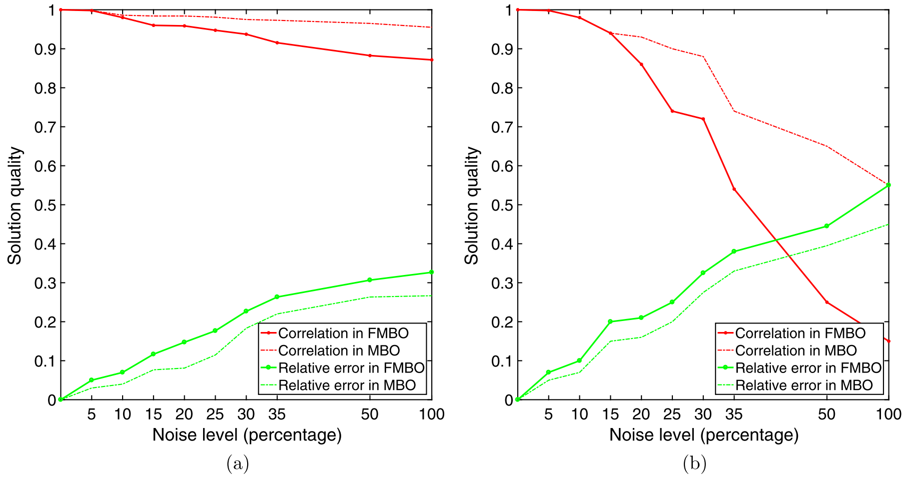


Figure 4: Solution accuracy against noise levels in distance matrix data. (a) For the simple helix data (b) For the simulated chromosomal data.

Figure 4 displays the solution quality against the noise level of data for the simple helix data (a) and the simulated chromosomal data (b). In both figures, red curves are Pearson correlation coefficients while the green ones are relative errors. The results for FMBO are in solid curves while dashed curves are for MBO. We conclude that both the original MBO and FMBO can recover the true structures if there is no noise in the distance matrix data, while the FMBO compromises solution quality comparing to the MBO as noise levels are increased, especially in the more complicated chromosomal structure.

Solutions from the synthetic helix data are visualized in Figure 5, with the distance matrix data being polluted by 5% (left), 25% (middle), and 50% (right) levels of noises. The first and second rows display the corresponding

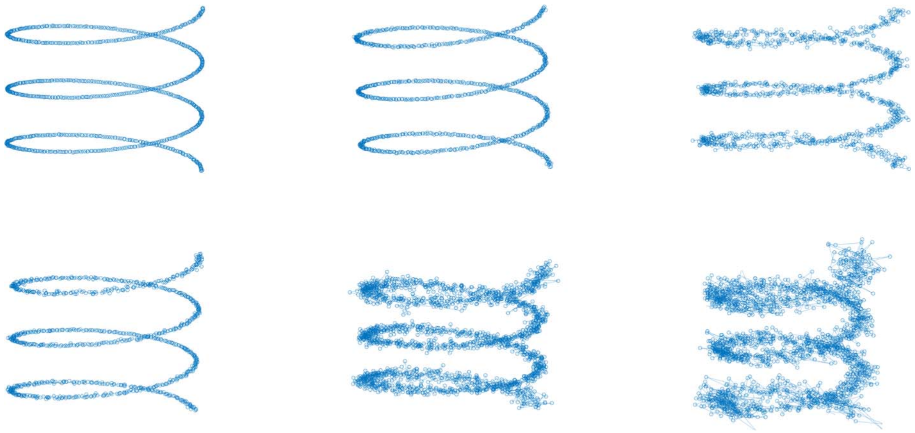


Figure 5: Numerical results for a synthetic helix structure recovered from distance matrices with 5% (left), 25% (middle) and 50% (right) of noises. First row: MBO results; Second row: RMBO results.

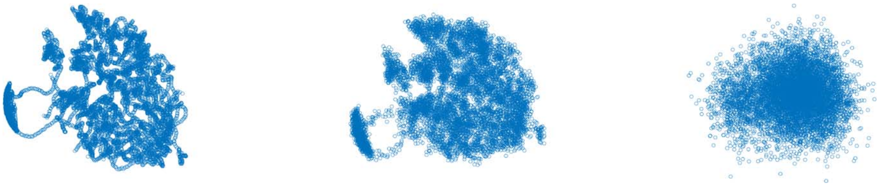


Figure 6: Numerical results of FMBO for a simulated chromosome structure recovered from distance matrices with 5% (left), 25% (middle) and 50% (right) of noise.

recovered helix structures from the MBO and FMBO, respectively. It can be observed that FMBO can successfully and efficiently recover the major characteristics of the structure, while the results are noisier than those generated by the original MBO. This phenomena can be understood by checking estimate (28): the solution accuracy or error bound mainly depends on the residual of the singular values of the distance matrix data $\tilde{\mathbf{D}}$ after truncation at k terms. The added noises are full-rank random matrices, so they make the overall distance matrices have slow decaying singular values. As results, high level noises lower solution accuracy or introduce larger magnitude of error bound in Eq. (28). Figure 6 shows the similar results for the simulated chromosome structures, which are recovered from distance matrices with 5% (left), 25% (middle) and 50% (right) of noise. Notice that when the noise level is high enough, the FMBO will totally miss the topological property of the original structure. These observations indicate that FMBO can greatly enhance algorithm efficiency, but it has more strict requirement for the quality (being close to an EDM) of the distance matrix data. Actually, for most existing structure-recovering algorithms, how to prepare the good distance matrix data from the original Hi-C contact map data is itself an important research topic.

In many cases, the experimental data of Hi-C contact map may be incomplete, while another advantage of the random projection method is that one only needs a small portion of original data if the random matrix is chosen as Eq. (18). For the simulated chromosomal structure case, we test the correlation of recovered 3D structure with different levels of missing data and the results are shown in Fig. 7. In these simulations, the noise level added to the EDM is 15%. For each level of data missing, the FMBO is implemented 20 times, then the variation and average of correlations are displayed. It is shown that from 5% to 50% data missing, the recovered 3D structures do not present significant differences from the true structure. So we conclude that the FMBO is robust to data missing in the structure prediction.

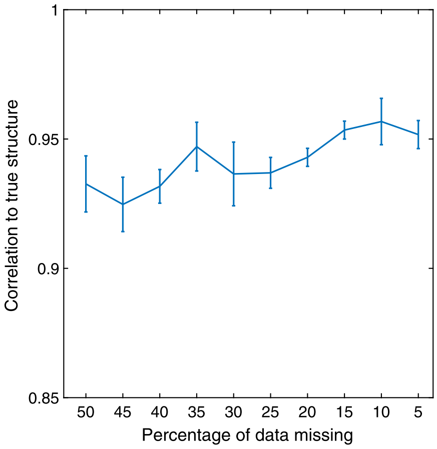


Figure 7: Ability of the RMBO to handle incomplete data.

5. Conclusion

Discovering three-dimensional structures of chromosomes is critical to understand their biological functions. Various models have blossomed over the past decades for analysis of 3D chromosome structures from Hi-C data.

Modeling methodologies include deterministic multidimensional scaling, optimization, Monte Carlo methods, or deep learning techniques. All models encounter the challenge of large volume of data at high resolutions. Furthermore, experimental Hi-C data are sparse, noisy, and subject to experimental uncertainty. The imprecision of the original data essentially limits the inherent solution quality, so it is profligate to pursue highly accurate results by expensive algorithms. Instead, fast algorithms are indispensable to enable various models to handle massive data, while balancing desired accuracy and efficiency.

In this work, we developed a set of randomized algorithms for the manifold-based optimization (MBO) model. The original MBO was verified to be useful in reconstructing 3D chromosomal structures, but could be extremely expensive when the total number of the loci N is large. It takes $O(N^2)$ complexity in *each* step in searching for the optimized structure, and the *total* steps needed are not clear due to the non-convexity of the objective function. Our fast algorithms for the MBO can achieve $O(N)$ complexity in each step and a total iteration step in the scale of $O(\log^4 N)$ was illustrated. The linear complexity in each step and polylog complexity in total steps greatly enhanced the efficiency of the MBO model. To achieve our goal, we used random projection theory to perform a low-rank approximation of the target dissimilarity data in the cost function. First, a random test matrix with small size was multiplied to the original data to approximate its column space, then interpolative decomposition was implemented to decompose the original data into a product of two matrices, each of which has a dimension much smaller than N. With such low-rank approximation of original data, and the inherent low-rank property of the distance matrix made of the unknown 3D structure, complexity of computations in calculating gradient of the cost function, line search, and cost function evaluation become $O(N)$. Another challenge is effectively escaping saddle points of the non-convex cost function, we adopted the randomly perturbed gradient descent method to address this difficulty, with a determined bound of total iteration numbers.

The developed algorithms were validated by two sets of data. One is a synthetic simple helix structure and the other is a simulated chromosomal structure from true Hi-C data. Our fast algorithm, termed as FMBO, exhibited greatly improved efficiency over the original MBO, while maintained satisfactory accuracy. 3D structures of the interested objects were able to be recovered by the data with different extents of missing entries and noises. It was also recognized that the FMBO algorithm is less resistant to high level of noise. In our simulations, the FMBO failed to recover the simulated chromosomal structure, if the contact frequency data include large noises.

A possible explanation of this phenomena is that the noises were "blended" in the useful information when doing the random projection for low-rank approximation. A thorough investigation of this characteristic and how to reduce its effects would be the potential future work. Another direction of future work could be establishing a complete data processing pipeline, such that the developed fast algorithms can be used on raw biological data.

References

[1] P-A Absil, Mariya Ishteva, Lieven De Lathauwer, and Sabine Van Huffel. A geometric Newton method for Oja's vector field. *Neural computation*, 21(5):1415–1433, 2009. MR2514671

[2] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. MR2364186

[3] Badri Adhikari, Tuan Trieu, and Jianlin Cheng. Chromosome3D: reconstructing three-dimensional chromosomal structures from Hi-C interaction frequency data using distance geometry simulated annealing. *BMC genomics*, 17(1):886, 2016.

[4] Davide Bau and Marc A Marti-Renom. Genome structure determination via 3C-based data integration by the Integrative Modeling Platform. *Methods*, 58(3):300–306, 2012.

[5] Jon-Matthew Belton and Job Dekker. Chromosome conformation capture (3c) in budding yeast. *Cold Spring Harbor Protocols*, 2015(6):pdb–prot085175, 2015.

[6] Kenneth L Clarkson, Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, Xiangrui Meng, and David P Woodruff. The fast Cauchy transform and faster robust linear regression. *SIAM Journal on Computing*, 45(3):763–810, 2016. MR3511781

[7] Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):54, 2017. MR3614862

[8] Michael B Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 278–287. SIAM, 2016. MR3478397

[9] Kenneth R Davidson and Stanislaw J Szarek. Local operator theory, random matrices and Banach spaces. *Handbook of the geometry of Banach spaces*, 1(317-366):131, 2001. MR1863696

[10] J Dekker, K Rippe, M Dekker, and N Kleckner. Capturing chromosome conformation science 2002 295. *N*, 5558:1306–1311.

[11] Job Dekker. Gene regulation in the third dimension. *Science*, 319(5871):1793–1794, 2008.

[12] Ivan Dokmanic, Reza Parhizkar, Juri Ranieri, and Martin Vetterli. Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.

[13] Josée Dostie and Job Dekker. Mapping networks of physical interactions between genomic elements using 5c technology. *Nature protocols*, 2(4):988, 2007.

[14] Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006. MR2231643

[15] Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on computing*, 36(1):158–183, 2006. MR2231644

[16] Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006. MR2231645

[17] Petros Drineas and Michael W Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(Dec):2153–2175, 2005. MR2249884

[18] Petros Drineas and Michael W Mahoney. RandNLA: randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

[19] Zhijun Duan, Mirela Andronescu, Kevin Schutz, Sean McIlwain, Yoo Jung Kim, Choli Lee, Jay Shendure, Stanley Fields, C Anthony Blau, and William S Noble. A three-dimensional model of the yeast genome. *Nature*, 465(7296):363, 2010.

[20] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[21] Peter Fraser and Wendy Bickmore. Nuclear organization of the genome and the potential for gene regulation. *Nature*, 447(7143):413, 2007.

[22] Alexey A Gavrilov, Arkadiy K Golov, and Sergey V Razin. Actual ligation frequencies in the chromosome conformation capture procedure. *PLoS One*, 8(3):e60403, 2013.

[23] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points-Online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

[24] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

[25] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011. MR2806637

[26] Ming Hu, Ke Deng, Zhaohui Qin, Jesse Dixon, Siddarth Selvaraj, Jennifer Fang, Bing Ren, and Jun S Liu. Bayesian inference of spatial organizations of chromosomes. *PLoS computational biology*, 9(1):e1002893, 2013.

[27] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1724–1732. JMLR.org, 2017.

[28] Michel Journée, Francis Bach, P-A Absil, and Rodolphe Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010. MR2678395

[29] Reza Kalhor, Harianto Tjong, Nimanthi Jayathilaka, Frank Alber, and Lin Chen. Genome architectures revealed by tethered chromosome conformation capture and population-based modeling. *Nature biotechnology*, 30(1):90, 2012.

[30] Annick Lesne, Julien Riposo, Paul Roger, Axel Cournac, and Julien Mozziconacci. 3D genome reconstruction from chromosomal contacts. *Nature methods*, 11(11):1141, 2014.

[31] Ping Li, Trevor J Hastie, and Kenneth W Church. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international*

conference on Knowledge discovery and data mining, pages 287–296. ACM, 2006.

[32] Edo Liberty, Franco Woolfe, Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007. MR2366406

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

[34] Per-Gunnar Martinsson, Vladimir Rokhlin, Yoel Shkolnisky, and Mark Tygert. ID: A software package for low-rank approximation of matrices via interpolative decompositions, Version 0.2, 2008.

[35] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30(1):47–68, 2011. MR2737933

[36] Xiangrui Meng and Michael W Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 91–100. ACM, 2013. MR3210770

[37] Adriana Miele and Job Dekker. Long-range chromosomal interactions and gene regulation. *Molecular biosystems*, 4(11):1046–1057, 2008.

[38] Bamdev Mishra, Gilles Meyer, and Rodolphe Sepulchre. Low-rank optimization for distance matrix completion. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4455–4460. IEEE, 2011. MR3064078

[39] Tom Misteli. Beyond the sequence: cellular organization of genome function. *Cell*, 128(4):787–800, 2007.

[40] Natalia Naumova, Emily M Smith, Ye Zhan, and Job Dekker. Analysis of long-range chromatin interactions using chromosome conformation capture. *Methods*, 58(3):192–203, 2012.

[41] Jackson Nowotny, Sharif Ahmed, Lingfei Xu, Oluwatosin Oluwadare, Hannah Chen, Noelan Hensley, Tuan Trieu, Renzhi Cao, and Jianlin Cheng. Iterative reconstruction of three-dimensional models of human

chromosomes from chromosomal contact data. *BMC bioinformatics*, 16(1):338, 2015.

[42] Oluwatosin Oluwadare, Yuxiang Zhang, and Jianlin Cheng. A maximum likelihood algorithm for reconstructing 3D structures of human chromosomes from chromosomal contact data. *BMC genomics*, 19(1):161, 2018.

[43] Jonas Paulsen, Odin Gramstad, and Philippe Collas. Manifold based optimization for single-cell 3D genome reconstruction. *PLoS computational biology*, 11(8):e1004396, 2015.

[44] Suhas SP Rao, Miriam H Huntley, Neva C Durand, Elena K Stamenova, Ivan D Bochkov, James T Robinson, Adrian L Sanborn, Ido Machol, Arina D Omer, Eric S Lander, et al. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680, 2014.

[45] CDM Rodley, F Bertels, B Jones, and JM O'sullivan. Global identification of yeast chromosome interactions using genome conformation capture. *Fungal Genetics and Biology*, 46(11):879–886, 2009.

[46] Mathieu Rousseau, James Fraser, Maria A Ferraiuolo, Josée Dostie, and Mathieu Blanchette. Three-dimensional modeling of chromatin structure from interaction frequency data using Markov Chain Monte Carlo sampling. *BMC bioinformatics*, 12(1):414, 2011.

[47] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 143–152. IEEE, 2006.

[48] Przemysław Szałaj, Zhonghui Tang, Paul Michalski, Michal J Pietal, Oscar J Luo, Michał Sadowski, Xingwang Li, Kamen Radew, Yijun Ruan, and Dariusz Plewczynski. An integrated 3-dimensional genome modeling engine for data-driven simulation of spatial genome organization. *Genome research*, 26(12):1697–1709, 2016.

[49] Joel A Tropp. Improved analysis of the subsampled randomized Hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011. MR2835584

[50] David Woodruff and Qin Zhang. Subspace embeddings and l_p-regression using exponential random variables. In *Conference on Learning Theory*, pages 546–567, 2013.

[51] David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014. MR3285427

[52] ZhiZhuo Zhang, Guoliang Li, Kim-Chuan Toh, and Wing-Kin Sung. 3D chromosome modeling with semi-definite programming and Hi-C data. *Journal of computational biology*, 20(11):831–846, 2013. MR3130294

Duan Chen
Department of Mathematics and Statistics
University of North Carolina at Charlotte
Charlotte, NC 28223
USA
E-mail address: dchen10@uncc.edu

Shaoyu Li
Department of Mathematics and Statistics
University of North Carolina at Charlotte
Charlotte, NC 28223
USA
E-mail address: sli23@uncc.edu

Xue Wang
Department of Health Science Research
Mayo Clinic
Jacksonville, FL
USA
E-mail address: wang.xue@mayo.edu

Kelin Xia
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
E-mail address: xiakelin@ntu.edu.sg

Received November 5, 2020