frontiers | Frontiers in Genetics

TYPE Methods
PUBLISHED 08 March 2023
DOI 10.3389/fgene.2023.1110899

Check for updates

OPEN ACCESS

EDITED BY
Lin Hou,
Tsinghua University, China

REVIEWED BY
Nayang Shan,
Capital University of Economics and Business, China
Jianyong Sun,
Xi'an Jiaotong University, China

*CORRESPONDENCE
Lin Wan,
✉ lwan@amss.ac.cn
Liang Ma,
✉ maliang@ioz.ac.cn

†These authors have contributed equally to this work and share first authorship

SPECIALTY SECTION
This article was submitted to Statistical Genetics and Methodology, a section of the journal Frontiers in Genetics

RECEIVED 29 November 2022
ACCEPTED 13 February 2023
PUBLISHED 08 March 2023

CITATION
Chen Z, Zhang B, Gong F, Wan L and Ma L (2023), RobustTree: An adaptive, robust PCA algorithm for embedded tree structure recovery from single-cell sequencing data.
Front. Genet. 14:1110899.
doi: 10.3389/fgene.2023.1110899

RobustTree: An adaptive, robust PCA algorithm for embedded tree structure recovery from single-cell sequencing data

Ziwei Chen[1†], Bingwei Zhang[2,3†], Fuzhou Gong[2,3], Lin Wan[2,3*] and Liang Ma[3,4*]

[1]Department of Systems Biology, Columbia University Irving Medical Center, New York, NY, United States, [2]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, [3]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China, [4]Institute of Zoology, Chinese Academy of Sciences, Beijing, China

Robust Principal Component Analysis (RPCA) offers a powerful tool for recovering a low-rank matrix from highly corrupted data, with growing applications in computational biology. Biological processes commonly form intrinsic hierarchical structures, such as tree structures of cell development trajectories and tumor evolutionary history. The rapid development of single-cell sequencing (SCS) technology calls for the recovery of embedded tree structures from noisy and heterogeneous SCS data. In this study, we propose RobustTree, a unified framework to reconstruct the inherent topological structure underlying high-dimensional data with noise. By extending RPCA to handle tree structure optimization, RobustTree leverages data denoising, clustering, and tree structure reconstruction. It solves the tree optimization problem with an adaptive parameter selection scheme that we proposed. In addition to recovering real datasets, RobustTree can reconstruct continuous topological structure and discrete-state topological structure of underlying SCS data. We apply RobustTree on multiple synthetic and real datasets and demonstrate its high accuracy and robustness when analyzing high-noise SCS data with embedded complex structures. The code is available at https://github.com/ucasdp/RobustTree.

KEYWORDS

single-cell sequencing, robust principal component analysis, data denoising, clustering, tree structure reconstruction

1 Introduction

Cell fate decisions and tumorigenesis are complex biological processes that experience many state transitions, such as cell differentiation and somatic cell evolution. The rapid development of single-cell sequencing (SCS) technology makes it possible to unveil the dynamics of such biological processes. However, the limited genetic content of a single cell and the stochastic nature of sequencing techniques can result in high rates of gene dropout and various sequencing errors, leading to noisy SCS data (Gawad et al., 2016; Chen et al., 2022; Wen et al., 2022). It poses additional challenges to reconstruct the potential hierarchical topological structure or dynamics of cells from such noisy high-dimensional SCS data.

In recent years, recovering intrinsic structure from high-dimensional data has become a central topic in the data science and machine learning community. Pioneering works generally seek to perform data dimensionality reduction or associate data with certain structured objects (Mao et al., 2016). For example, principal curve (Hastie and Stuetzle, 1989) and its successors (Tibshirani, 1992; Bishop et al., 1998; Kégl et al., 2000; Smola et al., 2001; Sandilya and Kulkarni, 2002; Olivas et al., 2009), fits/maps an infinitely differentiable curve with a finite length to pass through the middle of data. However, these methods cannot handle self-intersection data. To address this gap, principal graph method uses a collection of piecewise smooth curves to approximate the data structures with self-intersection (Kégl and Krzyzak, 2002). Topological data analysis, which performs graph representation of high-dimensional datasets, provides another way to handle self-intersection structure (Carlsson, 2009). Ge et al. (2011) develops an efficient topological data analysis algorithm, which uses Reeb graphs (Dey and Wang, 2011) to extract a one-dimensional skeleton from unorganized data. Another tool, Mapper (Singh et al., 2007), builds simplicial complexes to preserve certain topological structures from the original dataset, and has been applied on single-cell data to reconstruct the dynamical structures of cell states (Rizvi et al., 2017). In addition, Mao et al. (2016) proposes a principal graph and structure learning framework based on reversed graph embedding (RGE) to capture the local information of the underlying graph structure. RGE is subsequently equipped by Monocle2 (Qiu et al., 2017) for single-cell trajectory inference.

The high noise in SCS data, which is brought by either technological or/and experimental issues, could possibly affect downstream clustering or distort the reconstruction of intrinsic structures. A number of computational methods have been proposed to retrieve lost and corrupted information from SCS data (Chen Z. et al., 2020; Patruno et al., 2021). Among them, methods based on matrix decomposition are computationally more efficient, especially extensions of robust principle component analysis (RPCA) algorithms. RPCA is an efficient low-rank matrix decomposition method for recovering low-dimensional subspace from corrupted data (Lin et al., 2010; Candes et al., 2011; Hsu et al., 2011; Vidal et al., 2016), which has been applied to denoise either DNA or RNA profiles of SCS data (Chen C. et al., 2020; Chen Z. et al., 2020; Su et al., 2022). Since the assayed cells often come from a few states in cell development or clones in tumor, and cells of same state or clone have similar or identical expression or genomic profile. In addition, noise in observed SCS data generally introduced by technologies can be random and sparse (Chen Z. et al., 2020; Su et al., 2022). Therefore, SCS data well suits the low-rank plus sparse matrices assumptions of RPCA.

In this study, we propose a unified framework, termed RobustTree, to reconstruct the inherent topological structure underlying high-dimensional data with high noise. The framework involves a matrix decomposition process that recovers the latent data points in a low-rank space. And these data points are used directly to reconstruct a tree to represent the inherent single-cell evolutionary trajectory. We also introduce a discriminative and compact feature representation for clustering problems with an assumption that the cluster centers should be close to each other when connected on the learned tree structure, otherwise they should be distant (Mao et al., 2015). More specifically, the optimization objective function of the framework consists of the following three basic components, including 1) denoising using robust principal component analysis (RPCA) and extended RPCA method; 2) performing data clustering with a soft assignment strategy; 3) reconstructing the minimum spanning tree (MST) among cluster centers as the potential topological structure. RobustTree leverages data denoising, clustering, and tree structure reconstruction and solves the tree optimization problem with an adaptive parameter selection scheme. Based on adaptive trade-off parameters, RobustTree not only can reconstruct continuous topological structure, e.g., cell development trajectory based on single-cell RNA sequencing gene expression data, but also display discrete-state topological structure, e.g., tumor evolution history based on single-cell DNA sequencing genetic variant data, including single-cell single nucleotide variation (scSNV) data and single-cell copy number alteration (scCNA) data. By using multiple simulated and real datasets, we demonstrate that RobustTree is accurate and robust on high-noise data with complex structures.

2 Materials and methods

Let $X_{M\times N} \in \mathcal{X}$ represent an observed noisy SCS data matrix where the rows represent data points, such as cells, and columns represent features, such as genes or mutations. We consider recovering a latent low-rank matrix $A_{M\times N}$ corresponding to each $X_{M\times N}$. In the recovered low-dimensional space, the same, or similar, data points are aggregated into K clusters, and we reconstruct a tree-like structure $\mathcal{T}$ at the cluster level to represent the true topology of the data.

In the following, we introduce three components of the proposed framework, including 1) RPCA and extended RPCA algorithms (Section 2.1), which are used to recover low-rank subspace from data matrix with corrupted and/or missing entries, 2) data clustering (Section 2.2) and 3) the minimum spanning tree (MST) optimization problem (Section 2.3). Finally, we describe our RobustTree framework, which is a low-rank matrix recovery framework coupled with tree structure optimization (Section 2.4).

2.1 RPCA and extended RPCA

2.1.1 RPCA

The celebrated dimensionality reduction method, PCA (Vidal et al., 2005), which assumes that noise follows a Gaussian distribution, is unrobust to in-sample outliers. As a consequence, the robust PCA (RPCA) (Lin et al., 2010; Candes et al., 2011; Hsu et al., 2011; Vidal et al., 2016) emerges to recover the potential low-rank matrix from data with sharp and sparse noise.

Assume that the observed data matrix $X_{M\times N}$ is generated by the sum of two matrices $X = A_0 + E_0$, where A_0 is a low-rank matrix, and E_0 represents the intra-sample outliers affected by random sparse noise, then the RPCA problem can be formulated as:

$$\min_{A,E} \operatorname{rank}(A) + \lambda \|E\|_0, \quad \text{s.t.} \quad A + E = X, \tag{1}$$

where $\|\cdot\|_0$ denotes 0-norm of the matrix (i.e., the number of non-zero entries in the matrix), and λ is a trade-off parameter. However, solving Problem Eq. 1 is generally NP-hard (Vidal et al., 2005; Chen Z. et al., 2020). Therefore, in order to reduce the above computation burden, Problem Eq. 1 can be convexly relaxed as

$$\min_{A,E} \|A\|_* + \lambda\|E\|_1, \quad \text{s.t.} \quad A + E = X, \tag{2}$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ represent the nuclear norm and the ℓ_1 norm of the matrix, respectively. We refer to Problem Eq. 2 as a relaxed version of RPCA, which can be efficiently solved by augmented Lagrange multipliers (ALM) (Lin et al., 2010).

2.1.2 Extended RPCA

In practice, dropout/missing events can occur frequently in the observed matrix, except for corrupted entries. In order to model the missing entries, we first define a linear mapping $\mathcal{P}_\Phi(\cdot)$: $\mathbb{R}^{M\times N} \rightarrow \mathbb{R}^{M\times N}$, which maps the missing entries to 0 and keeps the observed entries, i.e., $[\mathcal{P}_\Phi(X)]_{i,j} = X_{i,j}$ if $(i, j) \in \Phi$, and $[\mathcal{P}_\Phi(X)]_{i,j} = 0$ otherwise. Then, RPCA problem can be extended to the following version (Wright et al., 2013; Shang et al., 2014; Vidal et al., 2016; Chen Z. et al., 2020):

$$\min_{A,E} \|A\|_* + \lambda\|E\|_1, \quad \text{s.t.} \quad \mathcal{P}_\Phi(A + E) = \mathcal{P}_\Phi(X), \tag{3}$$

which aims to decompose X into a low-rank matrix A and sparse component E based only on the observed data $P_\Phi(X)$. The optimization Problem Eq. 3 is shown to be equivalent to solving the following constrained optimization problem (Shang et al., 2014):

$$\min_{A,E} \|A\|_* + \lambda\|\mathcal{P}_\Phi(E)\|_1 \quad \text{s.t.} \quad A + E = X, \tag{4}$$

which can also be solved by the ALM algorithm (Shang et al., 2014; Chen Z. et al., 2020). It is worth noting that when Φ is the index set of all entries in X, Problem Eq. 4 can be transformed into Problem Eq. 2; that is, Problem Eq. 2 can be regarded as a special case of Problem Eq. 4. Thus, we can solve these problems in a unified form provided by Problem Eq. 4.

2.2 Data clustering

The second component of our proposed framework is clustering of the recovered low-rank matrix. Since RPCA-based recovery of low-rank matrices may not fully guarantee an error-free state, we cannot obtain cluster centers by simply merging rows with the same features in the low-rank matrix A (Chen Z. et al., 2020). Assume there exist K clusters in the data points. Then we denote C_k as the kth cluster centroid of A ($k \in \{1, \ldots, K\}$). We minimize the following quantization error (Smola et al., 2001) to get the optimal cluster centroids:

$$\sum_{i=1}^{M} \min_{k=1,\ldots,K} \|A_i - C_k\|_2^2. \tag{5}$$

When $K < M$, we introduce an indicator matrix $\Delta \in [0,1]^{M\times K}$, where the (i, k)th element $\delta_{i,k} = 1$ indicates that data point A_i is assigned to the kth cluster, and $\delta_{i,k} = 0$ otherwise. Then, we get the equivalent optimization objective as follows:

$$\sum_{i=1}^{M}\sum_{k=1}^{K} \delta_{i,k}\|A_i - C_k\|_2^2, \tag{6}$$

where $\sum_{k=1}^{K}\delta_{i,k} = 1$ and $\delta_{i,k} \in \{0, 1\}$ for $\forall i \in \{1, \ldots, M\}$. This is the same optimization objective as K-means clustering. However, when K is relatively large, K-means with minimization of optimization Problem Eq. 6 might generate empty clusters. To avoid this, we introduce a right stochastic matrix R, where $\sum_{k=1}^{K} r_{i,k} = 1, \forall i = 1, \ldots, M$. When the obtained R is an integer solution, this variant is equivalent to the above representation with the indicator matrix Δ. Subsequently, we follow (Mao et al., 2016) and employ the following soft assignment strategy by adding negative entropy regularization, as:

$$\sum_{i=1}^{M}\sum_{k=1}^{K} r_{i,k}\left(\|A_i - C_k\|_2^2 + \sigma \log r_{i,k}\right), \tag{7}$$

where $\sigma > 0$ is a regularization parameter. When $R \in [0,1]^{M\times K}$ is a left stochastic matrix with each column summing to one, optimization Problem Eq. 7 is equivalent to mean shift clustering, and when $R \in [0,1]^{M\times K}$ is a right stochastic matrix with each row summing to one, optimization Problem Eq. 7 is equivalent to the Gaussian mixture model with uniform weights (Mao et al., 2016).

2.3 Minimum spanning tree (MST)

MST is a common graphical representation applied in the reconstruction of dynamic biological processes, such as cell developmental trajectory reconstruction and tumor evolutionary history recovery (Gawad et al., 2014; Yuan et al., 2015; Ross and Markowetz, 2016; Qiu et al., 2017; Chen et al., 2019; Chen Z. et al., 2020). MST characterizes lineage tracing path between different cell states or tumor clones and can explicitly reflect the process of cell development or the progression of subclones (Chen Z. et al., 2020).

We follow the MST optimization scheme proposed by Mao et al. (2015). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a connected undirected graph with weight W, where $\mathcal{V} = \{1, \ldots, K\}$ is a set of vertices, $\mathcal{E}$ is a set of edges and the entry $w_{i,j}$ in W represents the weight associated with the edge $(V_i, V_j) \in \mathcal{E}, \forall i, j \in \mathcal{V}$. Then we define a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_\mathcal{T})$ on the graph $\mathcal{G}$ that connects all vertices with minimum total weight, where $\mathcal{E}_\mathcal{T}$ contains the edge set of tree $\mathcal{T}$. In order to represent and learn a tree, we consider $\{b_{i,j}\}$ as binary variables; that is,

$$b_{i,j} = \begin{cases} 1 & if\ (V_i, V_j) \in \mathcal{E}_\mathcal{T}; \\ 0 & otherwise. \end{cases}$$

Let $B = [b_{i,j}] \in \{0,1\}^{K\times K}$; Then the integer linear programming formulation of MST can be represented as follows (Mao et al., 2015):

$$\begin{aligned} &\min_{B\in\mathcal{B}} \sum_{i,j} b_{i,j}w_{i,j}, \\ &s.t. \mathcal{B} = \left\{B \in \{0,1\}^{K\times K}\right\} \bigcap \mathcal{B}', \\ &\mathcal{B}' = \left\{B = B^T\right\} \bigcap \left\{\frac{1}{2}\sum_{i,j} b_{i,j} = |\mathcal{V}| - 1\right\} \\ &\bigcap \left\{\frac{1}{2}\sum_{V_i\in\mathcal{S},V_j\in\mathcal{S}} b_{i,j} = |\mathcal{S}| - 1\right\}, \forall \mathcal{S} \subseteq \mathcal{V}. \end{aligned} \tag{8}$$

Worth noting that there are three constraints applied to set $\mathcal{B}'$, including 1) connection symmetry as an undirected graph; 2) restriction of spanning trees containing $|\mathcal{V}| - 1$ edges; and 3) acyclic and connectivity of a spanning tree. Instead of solving such an integer programming problem directly, which is very difficult, we can relax the problem as

$$\min_{B\in\mathcal{B}} \sum_{i,j} b_{i,j} w_{i,j}, \tag{9}$$

where $b_{i,j} \geq 0$, the set of linear constraints is given by $\mathcal{B} = \{B \geq 0\} \bigcap \mathcal{B}'$ (Mao et al., 2015). Then Problem Eq. 9 can be solved by Kruskal's algorithm (Kruskal, 1956; Cormen et al., 2022).

2.4 Low-rank matrix recovery coupled with tree structure optimization

Based on the three building blocks described in Sections 2.1, Sections 2.2, and Sections 2.3, we are ready to formulate our unified framework to learn the latent topological structure hidden underneath noisy SCS data. We use the alternating direction multiplier method (ADMM) to solve the proposed optimization objective function by simultaneously recovering real data points and learning a tree-like structure with guaranteed convergence.

2.4.1 RobustTree framework

Given observed input SCS data $X_{M\times N}$, our goal is to reveal the underlying tree structure that generates X. Since the observed data may be corrupted by noise, it is not appropriate to recover the underlying topology directly from the observation matrix. Instead, to unveil the underlying structure, we assume that a latent low-rank matrix $A_{M\times N}$ can be recovered from $X_{M\times N}$, and we focus on learning a tree structure $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ on the cluster level of A. Let $C_{K\times N}$ indicate the cluster centers of data points in A, and $B_{K\times K}$ represent the adjacency matrix of vertices $\mathcal{V}$ in graph $\mathcal{T}$; then, the optimization problem with respect to variables {A, E, C, B, R} is formulated as:

$$\begin{gathered}\min_{A,E,B,C,R} \|A\|_* + \lambda\|\mathcal{P}_\Phi(E)\|_1 + \frac{\theta}{2}\sum_{k,k'} b_{k,k'}\|C_k - C_{k'}\|_F^2 \\ + \gamma\left[\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\|A_i - C_k\|^2 + \sigma\Omega(R)\right] \\ s.t. A + E = X, B \in \mathcal{B}, \sum_{k=1}^{K} r_{i,k} = 1, r_{i,k} \geq 0, \forall i, \forall k,\end{gathered} \tag{10}$$

where λ, θ, γ and σ are trade-off parameters, $r_{i,k}$ indicates the (i, k)th entry in matrix $R \in \mathbb{R}^{M\times K}$, $\Omega(R) = \sum_{i=1}^{M}\sum_{k=1}^{K} r_{i,k}\log r_{i,k}$ represents negative entropy regularization, and the definition of $\mathcal{B}$ is detailed in optimization Problem Eq. 8.

2.4.2 RobustTree framework optimization algorithm

We solve optimization Problem Eq. 10 by alternating direction multiplier method (ADMM). We divide variables into two disjoint groups as {A, E, C} and {B, R}, and then solve each subproblem iteratively until convergence is achieved. We show details of the two following subproblems below.

- Fix {B, R} and update {A, C, E}

When fixing {B, R}, Problem Eq. 10 can be simplified into the following subproblem:

$$\begin{gathered}\min_{A,E,C} \|A\|_* + \lambda\|\mathcal{P}_\Phi(E)\|_1 + \frac{\theta}{2}\sum_{k,k'} b_{k,k'}\|C_k - C_{k'}\|_F^2 + \gamma\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\|A_i - C_k\|^2 \\ s.t. A + E = X.\end{gathered} \tag{11}$$

The corresponding augmented Lagrange function is:

$$\begin{aligned}\mathcal{L}(A,C,E) = \|A\|_* + \lambda\|\mathcal{P}_\Phi(E)\|_1 + < \Lambda, X - A - E > + \frac{\mu}{2}\|X - A - E\|_F^2 \\ + \frac{\theta}{2}\sum_{k,k'} b_{k,k'}\|C_k - C_{k'}\|_F^2 + \gamma\sum_i\sum_k r_{i,k}\left(\|A_i - C_k\|^2\right).\end{aligned} \tag{12}$$

Optimization Problem Eq. 12 can be transformed into the following form after some matrix manipulations:

$$\begin{aligned}&\mathcal{L}(A,C,E)= \\ &\quad + \theta\mathrm{trace}\left(C^TLC\right) + \gamma\left[\mathrm{trace}\left(A^TA\right) - 2\mathrm{trace}\left(R^TAC^T\right) + \mathrm{trace}\left(C^T\Gamma C\right)\right] \\ &\quad \text{where } L = \mathrm{diag}(B\mathbf{1}_K) - B,\ \Gamma = \mathrm{diag}\left(\mathbf{1}^TR\right).\end{aligned} \tag{13}$$

Updating C.

Let the partial derivative of $\mathcal{L}(A,C,E)$ with respect to C be zero, i.e., $\partial\mathcal{L}(A,C,E)/\partial C = \theta LC - \gamma R^TA + \gamma\Gamma C = 0$; then, we can obtain an analytical solution of C given by

$$C_A = \left(\frac{\theta}{\gamma}L + \Gamma\right)^{-1} R^TA.$$

Substituting C_A into $\mathcal{L}(A,C,E)$, we have

$$\begin{aligned}\mathcal{L}(A,C_A,E) = \|A\|_* + \lambda\|\mathcal{P}_\Phi(E)\|_1 + < \Lambda, X - A - E > + \frac{\mu}{2}\|X - A - E\|_F^2 \\ + \gamma\left[\mathrm{trace}\left(A^TA\right) - \mathrm{trace}\left(A^TR\left(\frac{\theta}{\gamma}L + \Gamma\right)^{-1} R^TA\right)\right].\end{aligned} \tag{14}$$

Updating E.

By retaining items related only to E in Problem Eq. 14, we have

$$\mathcal{L}'(E) = \lambda\|\mathcal{P}_\Phi(E)\|_1 + < \Lambda, X - A - E > + \frac{\mu}{2}\|X - A - E\|_F^2. \tag{15}$$

Solution E of Problem Eq. 15 can be written as (Shang et al., 2014; Chen Z. et al., 2020):

$$\begin{aligned}[E_{k+1}]_\Phi &= \left[\mathcal{S}_{\lambda\mu_k^{-1}}\left(X - A_k + \mu_k^{-1}\Lambda_k\right)\right]_\Phi \\ [E_{k+1}]_{\Phi^C} &= [X - A_k + \Lambda_k]_{\Phi^C}.\end{aligned} \tag{16}$$

Updating A.

After removing items unrelated to A in Problem Eq. 14, we have

$$\begin{aligned}\mathcal{L}'(A) = \|A\|_* + < \Lambda, X - A - E > + \frac{\mu}{2}\|X - A - E\|_F^2 \\ + \gamma\left[\mathrm{trace}\left(A^TA\right) - \mathrm{trace}\left(A^TR\left(\frac{\theta}{\gamma}L + \Gamma\right)^{-1} R^TA\right)\right].\end{aligned} \tag{17}$$

Then Problem Eq. 17 is equivalent to

$$\begin{aligned}\mathcal{L}'(A) = \|A\|_* + < \Lambda, X - A - E_{k+1} > + \frac{\mu}{2}\|X - A - E_{k+1}\|_F^2 \\ + \gamma\left[\|A\|_F^2 - \|S^TA\|_F^2\right],\end{aligned} \tag{18}$$

where $S = R\left(\frac{\theta}{\gamma}L + \Gamma\right)^{-1/2}$. We apply a proximity gradient algorithm to solve Problem Eq. 18. Let

$$g(A) = \|A\|_*,$$

$$f(A) = -<\Lambda, A> + \frac{\mu}{2}\|X - A - E\|_F^2 + \gamma\left(\|A\|_F^2 - \|S^T A\|_F^2\right),$$

then $\mathcal{L}'(A) = g(A) + f(A)$. Obviously, $g(A)$: $\mathbb{R}^{m\times n} \to (-\infty, +\infty)$ is a convex function, and $f(A)$: $\mathbb{R}^{m\times n} \to (-\infty, +\infty)$ is a smooth convex function. Thus, $\min_A \mathcal{L}'(A)$ can be solved by a proximity gradient algorithm. Assume that the gradient of $f(A)$ is Lipschitz continuous and that its constant is L_f, i.e.,

$$\|\nabla f(X) - \nabla f(Y)\| \leq L_f \|X - Y\|, \forall X, Y \in \mathbb{R}^{m\times n}.$$

Owing to

$$\begin{aligned}\nabla f(A) &= -\Lambda + \mu(A - (X - E)) + 2\gamma A - 2\gamma SS^T A\\ &= -\Lambda + \mu(A - (X - E)) + 2\gamma\left(I - SS^T\right)A,\end{aligned}$$

we have

$$\begin{aligned}\|\nabla f(X) - \nabla f(Y)\| &= \|\mu(X - Y) + 2\gamma\left(I - SS^T\right)(X - Y)\|\\ &= \|\left[(\mu + 2\gamma)I - 2\gamma SS^T\right](X - Y)\|\\ &\leq \|X - Y\| \ \|(\mu + 2\gamma)I - 2\gamma SS^T\|.\end{aligned}$$

Then, we can set $L_f = \|(\mu + 2\gamma)I - 2\gamma SS^T\|$. Considering the quadratic approximation function of $\mathcal{L}'(A)$ at a given point A_k:

$$\mathcal{L}'(A, A_k) = f(A_k) + <\nabla f(A_k), A - A_k> + \frac{L_f}{2}\|A - A_k\|_F^2 + g(A), \tag{19}$$

since formula (19) is a strong convex function, a primal solution exists for $\mathcal{L}'(A, A_k)$, i.e.,

$$\begin{aligned}\arg\min_A \ \mathcal{L}'(A, A_k) &= \arg\min_A \ f(A_k) + <\nabla f(A_k), A - A_k> + \frac{L_f}{2}\|A - A_k\|_F^2 + g(A)\\ &= \arg\min_A \ g(A) + \frac{L_f}{2}\|A - \left(A_k - \frac{1}{L_f}\nabla f(A_k)\right)\|_F^2\\ &= \mathrm{prox}_{g/L_f}\left(A_k - \frac{1}{L_f}\nabla f(A_k)\right).\end{aligned}$$

Let $G_k = A_k - \frac{1}{L_f}\nabla f(A_k) = A_k - \frac{1}{L_f}(-\Lambda + \mu(A_k - (X - E_{k+1})) + 2\gamma(I - SS^T)A_k)$ and perform the singular value decomposition on G_k with $G_k = U_k \Sigma V_k^T$, $\Sigma = \mathrm{diag}\{\sigma_i\}_{1\leq i\leq r}$. Then, $\forall 1/L_f > 0$, $\mathrm{prox}_{g/L_f}(G_k) = U_k \Sigma_{g/L_f} V_k^T, \Sigma_{g/L_f} = \mathrm{diag}(\{\sigma_i - \frac{1}{L_f}\}_+), \{\cdot\}_+ = \max\{0, \cdot\}$. Thus, the iterative form of the proximity gradient algorithm at the current point A_k is as follows:

$$A_{k+1} = \mathrm{prox}_{g/L_f}(G_k).$$

Then, we obtain pseudocode for the subproblems of solving A, E, C with fixed B, R, as shown in Algorithm 1.

```
fixing {$B$, $R$}
while not converged do
  $[E]_\Phi = [\mathcal{S}_{\lambda\mu^{-1}}(X - A + \mu^{-1}\Lambda)]_\Phi$
  $[E]_{\Phi^C} = [X - A + \Lambda]_{\Phi^C}$
  while not converged do
    $L = \mathrm{diag}(B\mathbf{1}_K) - B, \ \Gamma = \mathrm{diag}(\mathbf{1}^T R)$
    $S = R(\frac{\theta}{\gamma}L + \Gamma)^{-1/2}$
    $L_f = \|(\mu + 2\gamma)I - 2\gamma SS^T\|$
    $G = A - \frac{1}{L_f}(-\Lambda + \mu(A - (X - E)) + 2\gamma(I - SS^T)A)$
    $A = \mathrm{prox}_{g/L\ f}(G)$,
    where $\mathrm{prox}_{g/L_f}(G) = U\Sigma_{g/L_f}V^T, \ \Sigma_{g/L_f} = \mathrm{diag}(\{\sigma_i - \frac{1}{L_f}\}_+)$,
    $\{\cdot\}_+ = \max\{0, \cdot\}$
  $C = (\frac{\theta}{\gamma}L + \Gamma)^{-1}R^T A$
```

Algorithm 1. The Algorithm of solution for *A*, *E*, *C* with fixed *B*, *R*

- Fix $\{A, C, E\}$ and update $\{B, R\}$

Given $\{A, C, E\}$, Problem Eq. 10 with respect to B and R is a jointly convex optimization problem, which can be solved independently.

Updating *R*.

When fixing $\{A, E, C\}$, the optimization function related to R can be written as follows:

$$\mathcal{L}'(r_i, \alpha) = \sum_k r_{i,k}\left(\|A_i - C_k\|^2 + \sigma\log(r_{i,k})\right) + \alpha\left(\sum_k r_{i,k} - 1\right). \tag{20}$$

The KKT condition is $\|A_i - C_k\|^2 + \sigma(1 + \log(r_{i,k})) + \alpha = 0$ and $\sum_k r_{i,k} = 1, r_{i,k} \geq 0, \forall k \in \{1, \ldots, K\}$. Then we have the analytic solution of R given by $r_{i,k} = \exp(\|A_i - C_k\|^2/\sigma - (1 + \alpha/\sigma))$. Owing to $\sum_k r_{i,k} = 1$, we can get $\exp(1 + \alpha/\sigma) = \sum_{k=1}^K \exp(-\|A_i - C_k\|^2/\sigma)$. Then we can rewrite $r_{i,k}$ as

$$r_{i,k} = \frac{\exp(-\|A_i - C_k\|^2/\sigma)}{\sum_{k=1}^K \exp(-\|A_i - C_k\|^2/\sigma)}. \tag{21}$$

Updating *B*.

The term associated with B in optimization function (10) is to find the minimum spanning tree among cluster centers in C, which can be solved *via* Kruskal's algorithm (Kruskal, 1956; Cormen et al., 2022). Then we can sort out the pseudocode for the subproblem of solving B, R with fixed A, C, E as shown in Algorithm 2.

```
fixing {$A$, $C$, $E$}
$d_{k,k'} = \|C_k - C_{k'}\|^2$, $\forall k$, $\forall k'$
Obtain $B$ by solving optimization Problem Eq. 9 via
Kruskal's algorithm
Compute $R$ with each element as $r_{i,k} = \frac{\exp(-\|A_i - C_k\|^2/\sigma)}{\sum_{k=1}^K \exp(-\|A_i - C_k\|^2/\sigma)}$
```

Algorithm 2. The Algorithm of solution for *B*, *R* with fixed *A*, *C*, *E*

Finally, combining the two subproblems above, we formulate the complete pseudocode of exact RobustTree algorithm in Algorithm 3. Fortunately, as it turns out, in the kth iteration of B and R, we do not have to solve the subproblem $(A_{k+1}^\star, E_{k+1}{}^\star, C_{k+1}^\star) = \mathcal{L}(A, C, E, B_k, R_k)$ exactly, corresponding to line 8-17 of Algorithm 3. Rather, when solving this subproblem, updating A_k, C_k and E_k once is sufficient for them to converge to the optimal solution of RobustTree problem. This leads to an inexact RobustTree algorithm (see Algorithm 4).

```
Input: $X$, $\lambda$, $\theta$, $\gamma$, and $\sigma$
Output: $A$, $C$, $E$, $B$, $R$
Initialize $A$ by $ZF(X)$, $K$, $C$
While not converged do
  $d_{k,k'} = \|C_k - C_{k'}\|^2$, $\forall k$, $\forall k'$
  Obtain $B$ by solving Problem Eq. 9 via Kruskal's
  algorithm
  Compute $R$ with each element as $r_{i,k} = \frac{\exp(-\|A_i - C_k\|^2/\sigma)}{\sum_{k=1}^K \exp(-\|A_i - C_k\|^2/\sigma)}$
  While not converged do
    $[E]_\Phi = [\mathcal{S}_{\lambda\mu^{-1}}(X - A + \mu^{-1}\Lambda)]_\Phi$
    $[E]_{\Phi^C} = [X - A + \Lambda]_{\Phi^C}$
    While not converged do
      $L = \mathrm{diag}(B\mathbf{1}_K) - B, \ \Gamma = \mathrm{diag}(\mathbf{1}^T R)$
      $S = R(\frac{\theta}{\gamma}L + \Gamma)^{-1/2}$
      $L_f = \|(\mu + 2\gamma)I - 2\gamma SS^T\|$
      $G = A - \frac{1}{L_f}(-\Lambda + \mu(A - (X - E)) + 2\gamma(I - SS^T)A)$
```

```
        $A = \mathrm{prox}_{g/L_f}(G)$,
        where $\mathrm{prox}_{g/L_f}(G) = U\Sigma_{g/L_f}V^T$, $\Sigma_{g/L_f} = \mathrm{diag}(\{\sigma_i - \frac{1}{L_f}\}_+)$,
        $\{\cdot\}_+ = \max\{0, \cdot\}$
    $C = (\frac{\theta}{\gamma} L + \Gamma)^{-1} R^T A$
```

Algorithm 3. The exact RobustTree algorithm

```
Input: $X$, $\lambda$, $\theta$, $\gamma$, and $\sigma$
Output: $A$, $C$, $E$, $B$, $R$
Initialize $A$ by $ZF(X)$, $K$, $C$
while not converged do
  $d_{k,k'} = \|C_k - C_{k'}\|^2$, $\forall k$, $\forall k'$
  Obtain $B$ by solving Eq. 9 via Kruskal's algorithm
  Compute $R$ with each element as $r_{i,k} = \frac{\exp(-\|A_i - C_k\|^2/\sigma)}{\sum_{k=1}^{K}\exp(-\|A_i - C_k\|^2/\sigma)}$
  $[E]_\Phi = [\mathcal{S}_{\lambda\mu^{-1}}(X - A + \mu^{-1}\Lambda)]_\Phi$
  $[E]_{\Phi^C} = [X - A + \Lambda]_{\Phi^C}$
  $L = \mathrm{diag}(B\mathbf{1}_K) - B$, $\Gamma = \mathrm{diag}(\mathbf{1}^T R)$
  $S = R(\frac{\theta}{\gamma}L + \Gamma)^{-1/2}$
  $L_f = \|(\mu + 2\gamma)I - 2\gamma SS^T\|$
  $G = A - \frac{1}{L_f}(-\Lambda + \mu(A - (X - E)) + 2\gamma(I - SS^T)A)$
  $A = \mathrm{prox}_{g/L_f}(G)$,
  where $\mathrm{prox}_{g/L_f}(G) = U\Sigma_{g/L_f}V^T$, $\Sigma_{g/L_f} = \mathrm{diag}(\{\sigma_i - \frac{1}{L_f}\}_+)$, $\{\cdot\}_+ = \max\{0, \cdot\}$
  $C = (\frac{\theta}{\gamma} L + \Gamma)^{-1} R^T A$
```

Algorithm 4. The inexact RobustTree algorithm

2.4.3 Convergence analysis

Since optimization Problem Eq. 10 is non-convex, many local optimal solutions are possible. We perform theoretical convergence analysis as shown in Theorem 1.

Theorem 1. Let $\{B_l, R_l, A_l, C_l, E_l\}$ *be the solution of Problem* (10) *in the* lth *iteration, and let* $\mathcal{L}_l = \mathcal{L}(B_l, R_l, A_l, C_l, E_l)$ *be the corresponding objective function value; then we have:*

1. $\{\mathcal{L}_l\}$ *monotonically decreasing and*
2. *Sequences* $\{B_l, R_l, A_l, C_l, E_l\}$ *and* $\{\mathcal{L}_l\}$ *converging.*

Proof. Let $\{B_l, R_l, A_l, C_l, E_l\}$ be the solution obtained in the lth iteration. By Algorithm 3, at the $(l + 1)$th iteration, we have

$$\begin{aligned}\mathcal{L}(B_l, R_l, A_l, E_l, C_l) &\geq \mathcal{L}(B_{l+1}, R_l, A_l, E_l, C_l) \geq \mathcal{L}(B_{l+1}, R_{l+1}, A_l, E_l, C_l)\\ &\geq \mathcal{L}(B_{l+1}, R_{l+1}, A_{l+1}, E_{l+1}, C_l) \geq \mathcal{L}(B_{l+1}, R_{l+1}, A_{l+1}, E_{l+1}, C_{l+1}).\end{aligned}$$

Then, sequence $\{\mathcal{L}_l\}$ is monotonically decreasing. In addition, since $\mathcal{L}(B, R, A, C, E)$ is lower-bounded by $-\gamma\sigma M \log K$, $\mathcal{L}^*$ exists such that $\{\mathcal{L}_l\}$ converges to $\mathcal{L}^*$ according to the Monotonic Convergence Theorem. Then, we prove that sequence $\{B_l, R_l, A_l, C_l, E_l\}$ converges. Owing to the compactness of feasible sets B and R, the sequence $\{B_l, R_l\}$ converges to $\{B^*, R^*\}$ as $l \to \infty$. Based on the ADAL algorithm (Shang et al., 2014), $\{A_l, E_l\}$ converges to $\{A^*, E^*\}$. Since $C = (\frac{\theta}{\gamma}L + \Gamma)^{-1}R^T A$, $\{C_l\}$ converges to $C^* = (\frac{\theta}{\gamma}L^* + \Gamma^*)^{-1}R^{*T}A^*$, where $L^* = \mathrm{diag}(B^*\mathbf{1})—B^*$, $\Gamma^* = \mathrm{diag}(\mathbf{1}^T R^*)$.

2.4.4 Adaptive parameter selection

We denote missing rate as s in the observed input SCS data. Then we select the hyper-parameters in Problem Eq. 10 as follows:

$$\lambda = \frac{1 + 3s}{\sqrt{\max(M, N)}}, \theta = \frac{\sqrt{M}}{N\sqrt{N}}, \gamma = \frac{M}{N\sqrt{N}}, \sigma = \frac{\mathrm{var}(\mathrm{X})}{\max(M, N)}, \quad (22)$$

where the selection of λ refers to Chen Z. et al. (2020). And we choose the other parameters to coordinate the value of each single item in Problem 10 with a similar magnitude during the optimization process.

We initialize K as following:

$$K = \begin{cases} M & if\ M \leq 500; \\ \frac{M}{5} & if\ 500 < M \leq 1000; \\ \frac{M}{50} & if\ M > 1000. \end{cases} \quad (23)$$

Each cell i is assigned to cluster k, which has the maximum value $r_{i,j}$, $\forall j \in \{1, \ldots, K\}$, i.e., $k = \arg\max_{j\in\{1,\ldots,K\}} r_{i,j}$, and finally remove the repeated cluster centers to get the final cluster centers. With the above parameter settings, RobustTree can be applied to the reconstruction of continuous trajectory and discrete-state topological structure.

2.5 Evaluations

To evaluate cluster assignment and data recovery performance, we adopt the following measurements, including 1) adjusted rand index (ARI) (Rand, 1971; Qiu et al., 2017; Chen et al., 2019); 2) the error rate of the recovered matrixes to the ground truth (Chen Z. et al., 2020); 3) the percentage of missing entries imputed correctly (Miura et al., 2018; Chen Z. et al., 2020) and 4) the false positives and false negatives (FPs + FNs) ratios of output genotype matrix to input genotype matrix for scSNV data (Miura et al., 2018; Chen Z. et al., 2020).

3 Results

3.1 RobustTree reconstructs continuous trajectories on noisy simulation data with high accuracy

To demonstrate that RobustTree can preserve the global structure and handle high-noise data with continuous topology, we apply RobustTree to 6 simulated datasets with continuous trajectories. The original data are taken from Mao et al. (2016), which contain 200 (Spiral), 100 (circle), 300 (Three-cluster), 300 (Tree), 100 (Distorted S-shape), and 200 (Two moons) data points, respectively. To test the robustness of RobustTree to noise, we add 1 to 4 sharp noise points (points in the red circle in the fourth row of Figure 1) to each datum. We compare the results of RobustTree with l_1 Graph and Spanning Tree, which are two algorithms performing principal graph and structure learning based on inverse graph embedding (Mao et al., 2016), as well as RPCA/RobustClone (Chen Z. et al., 2020).

Although l_1 Graph and Spanning Tree show better effectiveness and stability than the Polygonal Line method (Kégl et al., 2000), SCMS (Ozertem and Erdogmus, 2011), and Mapper (Singh et al., 2007) algorithms in the results of Figure 5 in

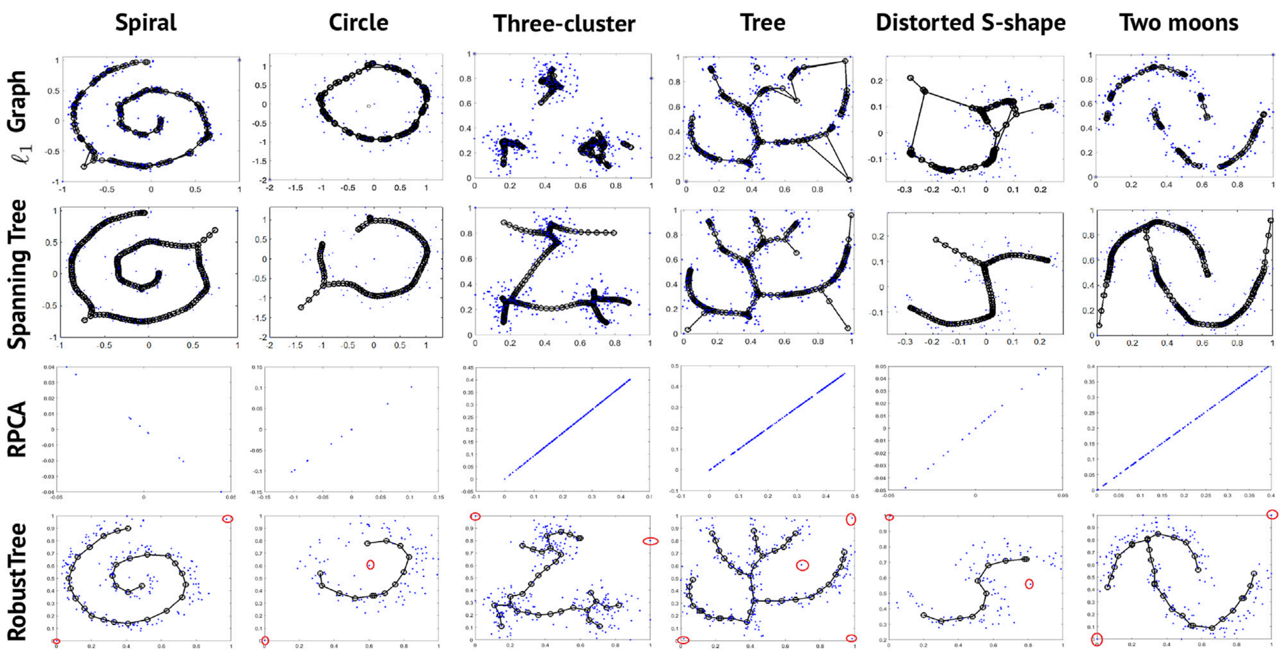

FIGURE 1
Results of 4 algorithms on 6 synthetic datasets with continuous topological structure. The red circle points in the fourth row are the artificially added noise points.

Mao et al. (2016), they are unstable to sharp noise. As shown in Figure 1, l_1 Graph and Spanning Tree will generate redundant bifurcations or edges to connect noise points with parameters tuned for the original dataset by Mao et al. (2016), highlighting the obviously ineffective identification or removal of noise. When we directly apply RPCA, which is the first step in RobustClone algorithm, to this synthetic dataset, it tends to optimize the continuous topological structure into a straight line. This is probably due to the fact that the original two dimensional data matrices do not have the relatively low-rank property required by the RPCA method.

In contrast, since RobustTree optimizes denoising, clustering and tree reconstruction in a unified framework, it shows stronger robustness than other methods. As shown in Figure 1, RobustTree effectively extracts the sharp noise into *E*, clusters the recovered latent low-rank matrix, and reconstructs the intrinsic continuous trajectory with multiple types of data, including structures with linear or simple bifurcations.

3.2 RobustTree reconstructs continuous multi-branch trajectory effectively

We perform RobustTree on simulated PHATE data to demonstrate its ability to handle data with continuous multi-branch development structures. The original data contain 1440 single cells and 60 genes (Moon et al., 2019), which imply an embedded continuous tree structure with 10 uniform branches to model a system where development along a given branch corresponds to increased expression of several genes (Figure 2A) (Chen et al., 2019; Moon et al., 2019).

RobustTree identifies 20 clusters and accurately reconstructs continuous trajectory with multi-branch on PHATE data, which contains three bifurcating events and one trifurcating event (Figure 2B). Figure 2C displays the distribution of clusters over branches. Clusters identified by RobustTree are almost exactly divided into a certain branch, except for clusters 2, 5, 10, and 18, which are located at branching points. The ARI between the branches identified by RobustTree and truth branch assignment is 0.6765. Since clusters at branching point contain cells from different branches, when excluding these clusters in ARI computation, we can achieve 0.9383.

We compare the RobustTree to Monocle2 (Qiu et al., 2017), a method that resolves complex single-cell trajectories using RGE, on this dataset. Monocle2 identifies 7 cell states, denoted as S1–S7, (Figures 2D, E, F), where real branches 1, 2 and 3, real branches 7, 8 and real branches 9, 10, are merged into S1, S3, and S7, respectively, leading to 6 main Monocle2 branches (Figures 2D, E, F). The ARI between the cell states identified by Monocle2 and the truth branch assignment is 0.4427.

3.3 RobustTree recovers discrete cell evolutionary history accurately on simulation data

Tumor evolution has been a subject with longstanding discussion (Nowell, 1976; Chen Z. et al., 2020). In tumors, cells form subpopulations (subclones) with nearly or completely identical genetic compositions, usually making the number of subclones much smaller than the number of cells or

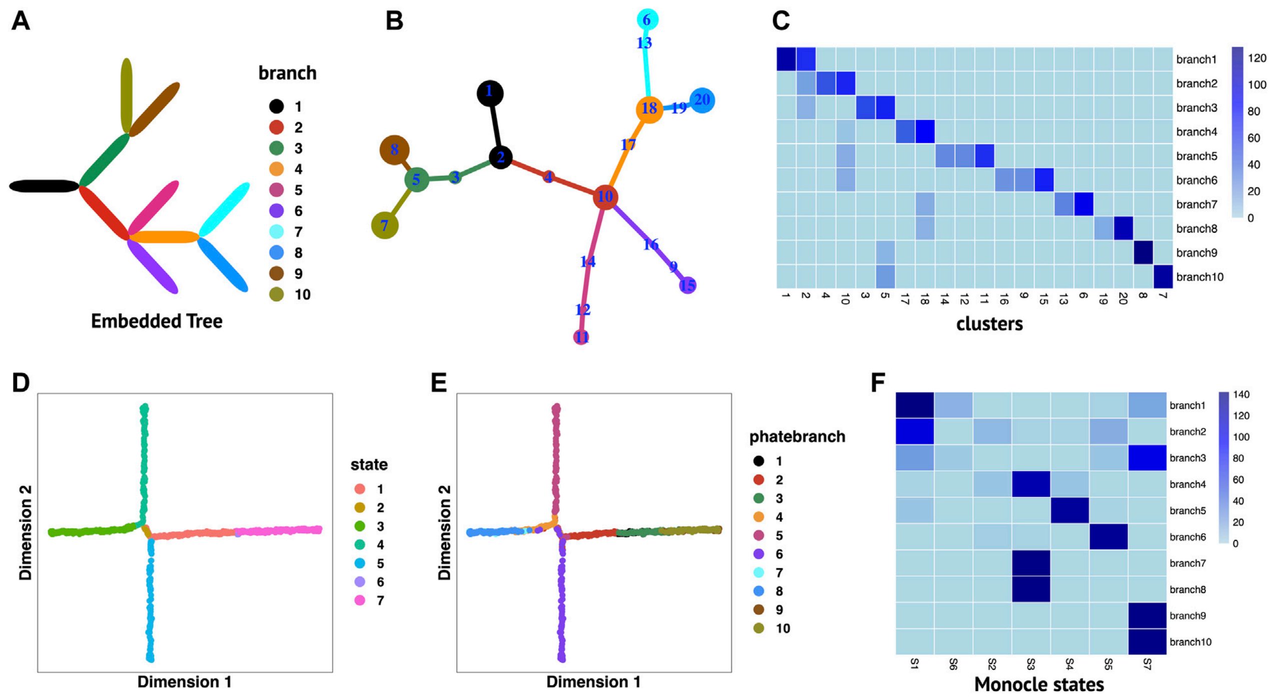

FIGURE 2
RobustTree reconstructs continuous multi-branch trajectory on PHATE data. **(A)** The real embedded tree structure of simulated PHATE data. **(B)** Tree trajectory reconstructed by RobustTree and visualized by R package igraph. The size of the cluster is proportional to the number of cells it contains, and the branch length is proportional to the distance between connected clusters. **(C)** Heatmap shows the percentage of cells in cluster (x-axis) distributed into real branch (y-axis). **(D)** Monocle2 reduces dimension on PHATE data. **(E)** Truth branch assignment on 2D embedding of Monocle2. **(F)** Heatmap shows the percentage of cells in Monocle states (x-axis) distributed into real branch (y-axis).

the number of mutational sites. In practice, the observed single-cell data are often incorporated with random noise caused by technical errors, including sequencing errors and dropout events. Accordingly, an important topic in tumor single-cell data analysis involves recovering subclonal genotypes and reconstructing the evolutionary history of subclones from corrupted data. In this study, we can also perform RobustTree on tumor single-cell DNA sequencing data to study the above problems.

We first apply RobustTree to simulated scSNV data, which contain 1000 cells and 300 mutation sites, along with a sequencing error rate of 30% and a dropout rate of 20% (Chen Z. et al., 2020) (Figure 3A). There are 5 subclones along the real cell evolution tree, containing 193, 235, 93, 241, and 238 cells, respectively (Supplementary Figure S1). We use RobustTree to recover real cell genotypes and reconstruct the subclone evolutionary tree, and compare the performance of RobustTree to a state-of-the-art method, SCG, on the dataset.

RobustTree shows more accuracy than SCG on this simulation dataset. Specifically, for subclone identification, RobustTree identifies 5 subclone assignments on the tree, where the subclone assignment is exactly the same as the true subclone assignment (Figure 3C), that is, the ARI between the subclones identified by RobustTree and the true clone assignment is 1. However, SCG identifies 4 clusters, containing 278, 93, 391, and 238 cells, respectively. The ARI between the SCG clusters and the real clone assignment is 0.7245. For genotype recovery, RobustTree recovers the true genotype matrix with 100% accuracy (Figure 3B) with error rate and FPs + FNs(output/input) as 0, missing imputed correctly rate as 1. And the recovered tree structure (Figure 3D) exactly coincides with the real evolutionary history (Supplementary Figure S1). In contrast, the FPs + FNs(output/input) and missing imputed correctly rate are 0.2484, and 0.9603, respectively, leading to the total error rate of 3.77% in SCG results.

3.4 RobustTree recovers scSNV genotype and infers subclonal tree on high-grade serous ovarian cancer data

We apply RobustTree on the single-cell high-grade serous ovarian cancer (abbreviated as HGSOC) data (McPherson et al., 2016; Roth et al., 2016; Chen Z. et al., 2020), which contain 420 cells and 43 selected SNV sites with a missing rate of 10.7% (Figure 4A). RobustTree efficiently recovers the real genotype by imputing the missing data and correcting the noisy entries (Figure 4B) and identifies 7 subclones on the reconstructed MST, which contain 40, 87, 0, 92, 18, 95, and 88 cells, respectively (Figure 4C). Since subclone1 does not contain any mutations, it is assigned as the root subclone (Figure 4C).

Along the phylogenetic trees reconstructed along RobustTree, heterozygous mutations first occur at loci 42 and

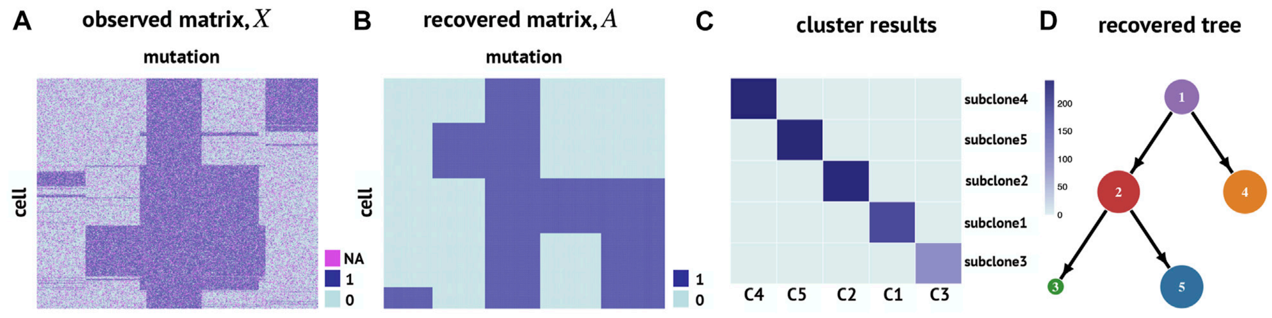

FIGURE 3
RobustTree reconstructs the tumor evolutionary tree on simulated single-cell DNA sequencing data. **(A)** Observed cell genotype matrix. **(B)** Recovered cell genotype matrix by RobustTree. **(C)** Heatmap shows the percentage of cells in each subclone (y-axis) distributed into real subclones (x-axis). **(D)** Subclone evolutionary tree reconstructed by RobustTree.

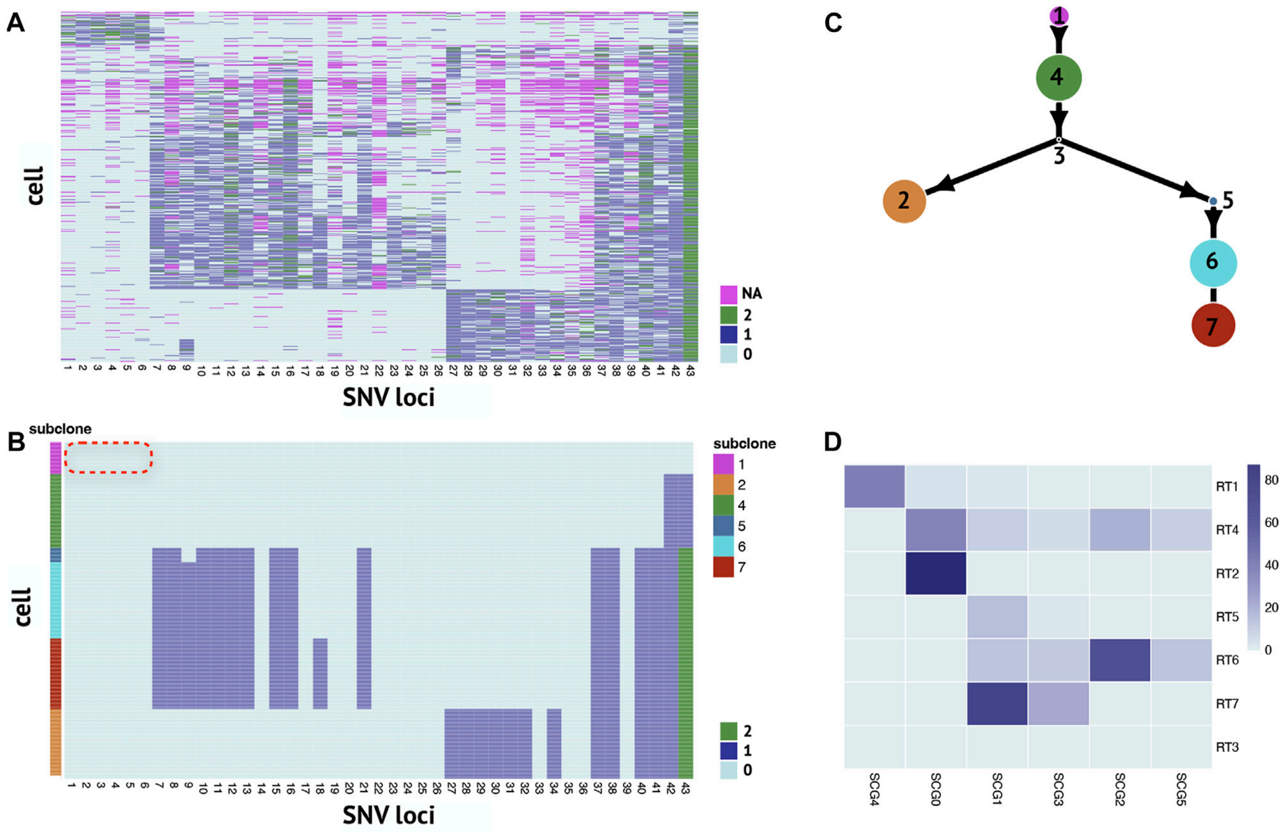

FIGURE 4
RobustTree reconstructs the tumor evolutionary tree on single-cell high-grade serous ovarian cancer data. **(A)** Observed noisy SNV genotype matrix. **(B)** Recovered SNV genotype matrix by RobustTree. **(C)** Subclone evolutionary tree reconstructed by RobustTree. **(D)** Heatmap shows the percentage of cells in each subclone (y-axis) distributed into SCG subclones (x-axis).

43 in subclone4. Followed by homozygous mutation at locus 43 and heterozygous mutations at loci 37, 38, 40 and 41, all descendant subclones inherit these mutations. In addition, based on the observable ancestor subclone4, subclone2 accumulates mutations at loci 27-34, on the other branch, mutations mostly occur at loci 7–21.

We compare RobustTree to SCG (Roth et al., 2016) on this dataset. SCG identifies 6 clusters, where one main branch in SCG results, SCG0, contains cells from both subclone2 and subclone4 on RobustTree, and another main branch consisting of clusters SCG1, SCG3, SCG2, SCG5 corresponds to the branch comprised of subclones 5, 6, 7 on RobustTree. The cells of root

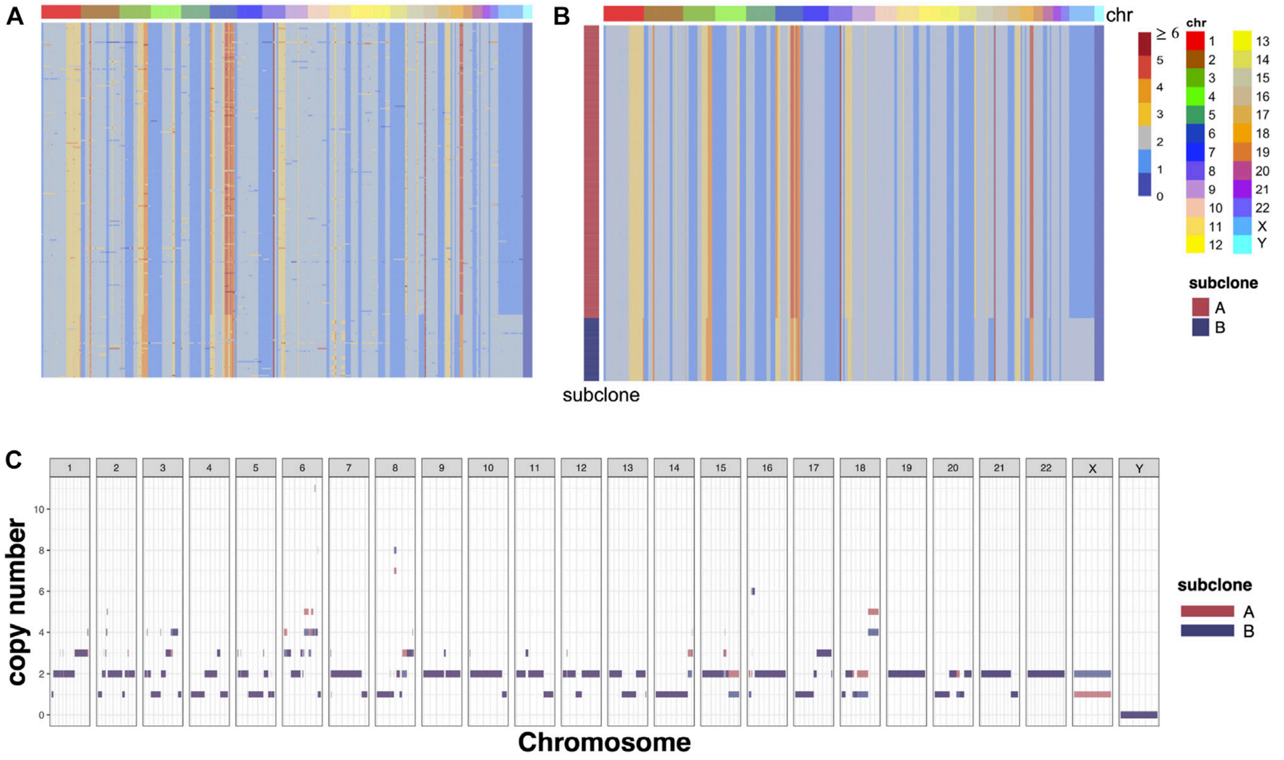

FIGURE 5
RobustTree reconstructs the tumor evolutionary tree on SA501X3F data. **(A)** Observed noisy CNA genotype matrix. **(B)** Recovered CNA genotype matrix by RobustTree. **(C)** Genotypes of subclones recovered by RobustTree.

cluster SCG4 dominate subclones1 of RobustTree, which can be interpreted as normal subclone (Figure 4D). However, SCG recovers some precancerous mutations in the normal cluster. In general, these precancerous mutations are expected to be carried in subsequent subclones, but they are completely absent in all progeny subclones (Figure 3 in Roth et al. (2016)). In contrast, RobustTree and RobustClone identify these mutations as false positive issues and recover the genotype without these mutations (Figure 4B), which seems more reasonable.

3.5 RobustTree recovers scCNA genotype on SA501X3F data

RobustTree can also detect copy number heterogeneity and identify subclones in scCNA data. To demonstrate this, the RobustTree algorithm was applied to the cell copy number profile from primary triple-negative breast cancer (TNBC) xenograft passages, denoted as SA501X3F data (Zahn et al., 2017; Campbell et al., 2019). The data consist of the copy number states with 260 single cells and 20,651 genomic bins, as shown in Figure 5A. By leveraging data denoising, clustering, and tree structure reconstruction, RobustTree identifies two subclones (Figure 5B), containing 214 cells (subclone A) and 46 cells (subclone B), respectively. RobustTree recovers true cell genotypes (Figure 5B) and subclonal genotypes (Figure 5C), where the difference between the genotypes of the two subclones lies in the large fragment variation on the X chromosome, and small fragment variants on the chromosomes 6, 8, 15, and 18 (Figure 5BC).

This result is completely consistent with the result of RobustClone, that is, the ARI value between RobustTree and RobustClone classification is 1. And the cell assignment of subcloneA is also completely consistent with the major subclone identified in Zahn et al. (2017); Campbell et al. (2019). In the results of Zahn et al. (2017), there are two subclones derived from the subcloneB of RobustTree, which contain 28 and 18 cells, respectively. Since Zahn et al. (2017) identifies clones without explicitly correction for noise, there exists some uncertainty of assignment between these two minor subclones (Campbell et al., 2019). Therefore, classification results with two subclones are more robust (Chen Z. et al., 2020).

4 Conclusion

Computational methods based on SCS data to reconstruct inherent structure can provide important insight into the understanding of cell development and tumor progression. In this study, we propose a unified framework, RobustTree, which can recover corrupted entries and reconstruct the intrinsic structure underlying data. By coupling RPCA with tree structure optimization, RobustTree can leverage data denoising, clustering and tree structure reconstruction, as well as solve the tree optimization problem using adaptive parameter selections. By comparing to some other state-of-the-art methods, experimental results demonstrate the effectiveness of RobustTree on various types of datasets with different topological structures, including continuous cellular complex development trajectory and discrete cell evolutionary tree.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://liwang8.people.uic.edu/PAMI2016-PGSL.html; https://github.com/KrishnaswamyLab/PHATE; https://github.com/ucasdp/RobustClone; https://static-content.springer.com/esm/art%3A10.1038%2Fnmeth.3867/MediaObjects/41592_2016_BFnmeth3867_MOESM262_ESM.xls; https://zenodo.org/record/2363826#.Y_Wn1OzMKdY.

Author contributions

ZC, BZ, FG, LM, and LW conceived the model. ZC and BZ conducted the experiments. ZC, BZ, LM, and LW analyzed the results. ZC, BZ, FG, LM, and LW wrote and reviewed the manuscript.

Funding

This work is supported by the National Key R&D Program of China under Grant 2019YFA0709501, NSFC grants (Nos.11971459, 12071466), NCMIS of CAS, and LSC of CAS.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2023.1110899/full#supplementary-material

References

Bishop, C. M., Svensén, M., and Williams, C. K. (1998). Gtm: The generative topographic mapping. *Neural Comput.* 10, 215–234. doi:10.1162/089976698300017953

Campbell, K. R., Steif, A., Laks, E., Zahn, H., Lai, D., McPherson, A., et al. (2019). clonealign: statistical integration of independent single-cell rna and dna sequencing data from human cancers. *Genome Biol.* 20, 54–12. doi:10.1186/s13059-019-1645-z

Candes, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *J. AMC* 58, 1–37. doi:10.1145/1970392.1970395

Carlsson, G. (2009). Topology and data. *Bull. Am. Math. Soc.* 46, 255–308. doi:10.1090/s0273-0979-09-01249-x

Chen, C., Wu, C., Wu, L., Wang, X., Deng, M., and Xi, R. (2020). scRMD: imputation for single cell RNA-seq data via robust matrix decomposition. *Bioinformatics* 36, 3156–3161. doi:10.1093/bioinformatics/btaa139

Chen, Z., An, S., Bai, X., Gong, F., Ma, L., and Wan, L. (2019). DensityPath: An algorithm to visualize and reconstruct cell state-transition path on density landscape for single-cell RNA sequencing data. *Bioinformatics* 35, 2593–2601. doi:10.1093/bioinformatics/bty1009

Chen, Z., Gong, F., Wan, L., and Ma, L. (2022). BiTSC2: Bayesian inference of tumor clonal tree by joint analysis of single-cell SNV and CNA data. *Briefings Bioinforma.* 23, bbac092. doi:10.1093/bib/bbac092

Chen, Z., Gong, F., Wan, L., and Ma, L. (2020). RobustClone: A robust PCA method for tumor clone and evolution inference from single-cell sequencing data. *Bioinformatics* 36, 3299–3306. doi:10.1093/bioinformatics/btaa172

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2022). *Introduction to algorithms*. Massachusetts: MIT press.

Dey, T. K., and Wang, Y. (2011). "Reeb graphs: Approximation and persistence," in Proceedings of the twenty-seventh annual symposium on Computational geometry, Paris, France, June 13-15, 2011, 226–235.

Gawad, C., Koh, W., and Quake, S. R. (2014). Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics. *Proc. Natl. Acad. Sci.* 111, 17947–17952. doi:10.1073/pnas.1420822111

Gawad, C., Koh, W., and Quake, S. R. (2016). Single-cell genome sequencing: Current state of the science. *Nat. Rev. Genet.* 17, 175–188. doi:10.1038/nrg.2015.16

Ge, X., Safa, I., Belkin, M., and Wang, Y. (2011). "Data skeletonization via reeb graphs," in *Advances in neural information processing systems* (Massachusetts: Massachusetts Institute of Technology Press), 24.

Hastie, T., and Stuetzle, W. (1989). Principal curves. *J. Am. Stat. Assoc.* 84, 502–516. doi:10.1080/01621459.1989.10478797

Hsu, D., Kakade, S. M., and Zhang, T. (2011). Robust matrix decomposition with sparse corruptions. *IEEE Trans. Inf. Theory* 57, 7221–7234. doi:10.1109/tit.2011.2158250

Kégl, B., Krzyzak, A., Linder, T., and Zeger, K. (2000). Learning and design of principal curves. *IEEE Trans. Pattern Analysis Mach. Intell.* 22, 281–297. doi:10.1109/34.841759

Kégl, B., and Krzyzak, A. (2002). Piecewise linear skeletonization using principal curves. *IEEE Trans. Pattern Analysis Mach. Intell.* 24, 59–74. doi:10.1109/34.982884

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.* 7, 48–50. doi:10.1090/s0002-9939-1956-0078686-7

Lin, Z., Chen, M., and Ma, Y. (2010). The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055.

Mao, Q., Wang, L., Goodison, S., and Sun, Y. (2015). "Dimensionality reduction via graph structure learning," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. Editor L. Cao (New York: Association for Computing Machinery), 765–774.

Mao, Q., Wang, L., Tsang, I. W., and Sun, Y. (2016). Principal graph and structure learning based on reversed graph embedding. *IEEE Trans. Pattern Analysis Mach. Intell.* 39, 2227–2241. doi:10.1109/tpami.2016.2635657

McPherson, A., Roth, A., Laks, E., Masud, T., Bashashati, A., Zhang, A. W., et al. (2016). Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer. *Nat. Genet.* 48, 758–767. doi:10.1038/ng.3573

Miura, S., Huuki, L. A., Buturla, T., Vu, T., Gomez, K., and Kumar, S. (2018). Computational enhancement of single-cell sequences for inferring tumor evolution. *Bioinformatics* 34, i917–i926. doi:10.1093/bioinformatics/bty571

Moon, K. R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D. B., Chen, W. S., et al. (2019). Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.* 37, 1482–1492. doi:10.1038/s41587-019-0336-3

Nowell, P. C. (1976). The clonal evolution of tumor cell populations: Acquired genetic lability permits stepwise selection of variant sublines and underlies tumor progression. *Science* 194, 23–28. doi:10.1126/science.959840

Olivas, E. S., Guerrero, J. D. M., Martinez-Sober, M., Magdalena-Benedito, J. R., Serrano, L., Jos, A., et al. (2009). *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. Pennsylvania: IGI global.

Ozertem, U., and Erdogmus, D. (2011). Locally defined principal curves and surfaces. *J. Mach. Learn. Res.* 12, 1249–1286.

Patruno, L., Maspero, D., Craighero, F., Angaroni, F., Antoniotti, M., and Graudenzi, A. (2021). A review of computational strategies for denoising and imputation of single-cell transcriptomic data. *Briefings Bioinforma.* 22, bbaa222. doi:10.1093/bib/bbaa222

Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H. A., et al. (2017). Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods* 14, 979–982. doi:10.1038/nmeth.4402

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* 66, 846–850. doi:10.1080/01621459.1971.10482356

Rizvi, A. H., Camara, P. G., Kandror, E. K., Roberts, T. J., Schieren, I., Maniatis, T., et al. (2017). Single-cell topological rna-seq analysis reveals insights into cellular differentiation and development. *Nat. Biotechnol.* 35, 551–560. doi:10.1038/nbt.3854

Ross, E. M., and Markowetz, F. (2016). Onconem: Inferring tumor evolution from single-cell sequencing data. *Genome Biol.* 17, 69–14. doi:10.1186/s13059-016-0929-9

Roth, A., McPherson, A., Laks, E., Biele, J., Yap, D., Wan, A., et al. (2016). Clonal genotype and population structure inference from single-cell tumor sequencing. *Nat. Methods* 13, 573–576. doi:10.1038/nmeth.3867

Sandilya, S., and Kulkarni, S. R. (2002). Principal curves with bounded turn. *IEEE Trans. Inf. Theory* 48, 2789–2793. doi:10.1109/tit.2002.802614

Shang, F., Liu, Y., Cheng, J., and Cheng, H. (2014). "Robust principal component analysis with missing data," in Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, November 3 - 7, 2014, 1149–1158.

Singh, G., Mémoli, F., and Carlsson, G. E. (2007). "Topological methods for the analysis of high dimensional data sets and 3d object recognition," in Eurographics Symposium on Point-Based Graphics, Mario Botsch, ETH Zurich, 2-3 Sept 2007.

Smola, A., Mika, S., Schölkopf, B., and Williamson, R. (2001). *Regularized principal manifolds*. Massachusetts: MIT Press.

Su, Y., Wang, F., Zhang, S., Liang, Y., Wong, K.-C., and Li, X. (2022). scWMC: weighted matrix completion-based imputation of scRNA-seq data via prior subspace information. *Bioinformatics* 38, 4537–4545. doi:10.1093/bioinformatics/btac570

Tibshirani, R. (1992). Principal curves revisited. *Statistics Comput.* 2, 183–190. doi:10.1007/bf01889678

Vidal, R., Ma, Y., and Sastry, S. (2016). *Generalized principal component analysis*. Berlin: Springer.

Vidal, R., Ma, Y., and Sastry, S. (2005). Generalized principal component analysis (gpca). *IEEE Trans. Pattern Analysis Mach. Intell.* 27, 1945–1959. doi:10.1109/TPAMI.2005.244

Wen, L., Li, G., Huang, T., Geng, W., Pei, H., Yang, J., et al. (2022). Single cell technologies: From research to application. *Innovation* 3, 100342. doi:10.1016/j.xinn.2022.100342

Wright, J., Ganesh, A., Min, K., and Ma, Y. (2013). Compressive principal component pursuit. *Inf. Inference A J. IMA* 2, 32–68. doi:10.1093/imaiai/iat002

Yuan, K., Sakoparnig, T., Markowetz, F., and Beerenwinkel, N. (2015). Bitphylogeny: A probabilistic framework for reconstructing intra-tumor phylogenies. *Genome Biol.* 16, 36–16. doi:10.1186/s13059-015-0592-6

Zahn, H., Steif, A., Laks, E., Eirew, P., VanInsberghe, M., Shah, S. P., et al. (2017). Scalable whole-genome single-cell library preparation without preamplification. *Nat. Methods* 14, 167–173. doi:10.1038/nmeth.4140