

RESEARCH ARTICLE

Identifying Multi-Dimensional Co-Clusters in Tensors Based on Hyperplane Detection in Singular Vector Spaces

Hongya Zhao¹*, Debby D. Wang^{2,3}*, Long Chen²*, Xinyu Liu², Hong Yan²

1 Industrial Center, Shenzhen Polytechnic, Shenzhen, China, **2** Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, **3** Caritas Institute of Higher Education, New Territories, Hong Kong

* These authors contributed equally to this work.

* ee.chenlong@my.cityu.edu.hk



CrossMark
click for updates

OPEN ACCESS

Citation: Zhao H, Wang DD, Chen L, Liu X, Yan H (2016) Identifying Multi-Dimensional Co-Clusters in Tensors Based on Hyperplane Detection in Singular Vector Spaces. PLoS ONE 11(9): e0162293. doi:10.1371/journal.pone.0162293

Editor: Fabio Rapallo, Università degli Studi del Piemonte Orientale Amedeo Avogadro, ITALY

Received: February 28, 2016

Accepted: August 19, 2016

Published: September 6, 2016

Copyright: © 2016 Zhao et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: Data are available as a Supporting Information file ([S1 Dataset](#)) in MATLAB MAT-File Format. Additionally, the source link of each dataset used has been added to the manuscript.

Funding: This work is supported by the Hong Kong Research Grants Council (Project CityU 11214814) and the National Natural Science Funds of China (Project No. 31100958).

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Co-clustering, often called biclustering for two-dimensional data, has found many applications, such as gene expression data analysis and text mining. Nowadays, a variety of multi-dimensional arrays (tensors) frequently occur in data analysis tasks, and co-clustering techniques play a key role in dealing with such datasets. Co-clusters represent coherent patterns and exhibit important properties along all the modes. Development of robust co-clustering techniques is important for the detection and analysis of these patterns. In this paper, a co-clustering method based on hyperplane detection in singular vector spaces (**HDSVS**) is proposed. Specifically in this method, higher-order singular value decomposition (HOSVD) transforms a tensor into a core part and a singular vector matrix along each mode, whose row vectors can be clustered by a linear grouping algorithm (LGA). Meanwhile, hyperplanar patterns are extracted and successfully supported the identification of multi-dimensional co-clusters. To validate **HDSVS**, a number of synthetic and biological tensors were adopted. The synthetic tensors attested a favorable performance of this algorithm on noisy or overlapped data. Experiments with gene expression data and lineage data of embryonic cells further verified the reliability of **HDSVS** to practical problems. Moreover, the detected co-clusters are well consistent with important genetic pathways and gene ontology annotations. Finally, a series of comparisons between **HDSVS** and state-of-the-art methods on synthetic tensors and a yeast gene expression tensor were implemented, verifying the robust and stable performance of our method.

Introduction

Clustering analysis has become a fundamental tool in statistics, machine learning and signal processing [1]. A number of clustering algorithms have been developed, with the general idea of seeking groups among different objects in a full feature space. However, this process has several limitations, such as the adoption of a global-feature similarity among objects and the

selection of a representative for each group. In contrast, many applications detect sub-matrices, which manifest coherent patterns among the rows or columns, in the object-feature matrix. For example, identification of genes that are co-expressed under certain conditions in microarray experiments and text mining of document groups that are characterized by word groups are two representatives of this type of study [2, 3]. To assist this sub-matrix analysis, a series of efficient approaches, such as sub-dimensional clustering, linear grouping and co-clustering, have been developed [2, 4–10].

Inspired by the concept of *direct clustering* [11], biclustering (co-clustering) plays an important role in the analysis of gene expression data [2]. This technique simultaneously clusters the rows (genes) and columns (experimental conditions) of the gene-condition matrix. Consequently, a subset of rows exhibiting significant coherence within a subset of columns in the matrix can be extracted. These coherent rows and columns are accordingly regarded as a bicluster, which corresponds to a specific coherent pattern. Commonly-studied biclusters in gene expression data present patterns with *constant value*, *coherent value*, and *coherent evolutions* [12]. Practically, biclustering is quite challenging, especially for large-scale data sets. Comprehensive reviews on this topic can be found in [12–17].

Cheng and Church developed an efficient node-detection algorithm (*CC*) to find valuable submatrices in yeast or human expression data, based on mean squared residue scores [2]. Later, co-clustering in a document-word matrix was novelly transferred into a bipartite graph partitioning problem by Dhillon [18], and a spectral algorithm (*BSGP*) was proposed to give a reasonable partitioning solution. Bergmann *et al.* defined a transcription module in gene expression data by iteratively searching for co-clusters until a threshold was reached (*ISA*) [19]. *ISA* was also reported to perform well when applied to large-scale data. Subsequently, a simple binary reference model was provided for comparing and validating biclustering methods [20], and meanwhile a fast divide-and-conquer algorithm (*BiMax*: <http://www.tik.ee.ethz.ch/sop/bimax>) was proposed. As another classic co-clustering method, *FABIA* (*Factor Analysis for Bicluster Acquisition*) is a multiplicative model depending on linear dependencies between gene expression and experimental conditions [21].

Nowadays, multi-dimensional arrays (tensors), such as color images ([*row, column, color*]) [22, 23] and microarray data ([*gene, condition, time*]) [24, 25], frequently occur in clustering-related studies, demanding effective techniques that can deal with such data sets and identify useful co-clusters in them [26–30]. Banerjee *et al.* proposed a tensor co-clustering method by describing all the known relations between the different entity classes with a relation graph model [27]. Another method to detect co-clusters in tensors is based on multilinear decomposition with sparse latent factors [29]. In our work, the multi-dimensional co-clustering of a high-order tensor is accomplished by the conjunction of higher-order singular value decomposition (HOSVD) [23] and linear grouping algorithm (LGA) [5, 31]. Huang *et al.* has also employed HOSVD [23], together with the K-Means clustering, in their co-cluster method. However, the K-Means algorithm could only form clusters around “object” centers in the singular vector spaces, which is mainly related to constant biclusters. On the contrary, LGA could find linear structures (lines, planes and hyperplanes) in the singular vector spaces. These linear structures correspond to other types of co-clusters (constant-row/column, additive and multiplicative co-clusters) in addition to constant ones in the original data. Firstly in our method, generalized from the SVD of a matrix, a truncated HOSVD is implemented on an Nth-order tensor, resulting in a core tensor and a series of singular vector matrices along each mode. Secondly, LGA is subsequently applied to reveal the linear patterns embedded in singular vector matrices, with biclusters along each mode detected. Finally, through combining the detected linear structures of all the modes and defining a scoring function, we can successfully identify significant co-clusters in the tensor. To validate our method and compare it with existing ones,

multiple synthetic and biological tensors are used, and the detected significant co-clusters are analyzed according to genetic pathways and gene ontology (**GO**) annotations [32] to attest the biological significance of these co-clusters [33].

Methods

Notations and Preliminaries

An N th-order (N -mode) tensor $\mathcal{A} = \{a_{i_1 i_2 \dots i_N}, i_k = 1, \dots, I_k\}$ can be defined as a multi-dimensional array, where N is the number of dimensions [23]. Here we adopt a boldface and Euler-script letter \mathcal{A} to denote a tensor, with its entry notated as $a_{i_1 i_2 \dots i_N}$. Accordingly, a column vector can be denoted using a boldface and lowercase letter, e.g. \mathbf{a} , with its i th entry notated as a_i ; and a matrix is denoted by a boldface and uppercase letter, e.g. \mathbf{A} , with its entry in the i th row and j th column denoted as a_{ij} .

Further, the i th row and j th column of \mathbf{A} can be notated as \mathbf{a}_i and \mathbf{a}_j , respectively. Additionally, fibers and slices of \mathcal{A} can be defined. For example, the column-, row- and tube-fibers of a 3-mode tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$ are denoted as \mathbf{a}_{jk} , \mathbf{a}_{ik} and \mathbf{a}_{ij} , respectively. The horizontal, lateral and frontal slices of this tensor are notated as $\mathbf{A}_{i::}$, $\mathbf{A}_{::j}$, and $\mathbf{A}_{::k}$, respectively.

The process of transforming a tensor into a 2D matrix is called unfolding or flattening. The mode- n unfolded matrix $\mathbf{A}_{(n)}$ of a tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$ is a matrix of size $I_n \times (\prod_{k \neq n} I_k)$, with its mode- n fibers reduced to the columns [34].

Unfolded matrices play an important role in the product of a tensor and a matrix. Similar to the product of two 2D matrices, the mode- n product of a tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$ with a matrix $\mathbf{U} \in \mathbf{R}^{J \times I_n}$ can be denoted as $\mathcal{A} \times_n \mathbf{U}$, which is a tensor (size $I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N$) composed of the following entries,

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 \dots i_{n-1} j i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 \dots i_{n-1} i_n i_{n+1} \dots i_N} u_{ji_n} \quad (1)$$

This can also be described by a matrix product in Eq (2), using the unfolded matrices.

$$\mathbf{y} = \mathcal{A} \times_n \mathbf{U} \Leftrightarrow \mathbf{Y}_{(n)} = \mathbf{U} \mathbf{A}_{(n)} \quad (2)$$

Biclustering in Matrices

Biclusters in Matrices. In a data matrix \mathbf{A} , a bicluster is a sub-matrix of \mathbf{A} and represents a coherent pattern [12]. Specifically, we notate a bicluster as \mathbf{A}_{IJ} , where $\mathbf{I} = \{i_1, i_2, \dots, i_s\}$ stands for a subset of rows and $\mathbf{J} = \{j_1, j_2, \dots, j_t\}$ is a subset of columns. Based on \mathbf{A}_{IJ} , we can further define several types of generally-discussed biclusters:

1. constant biclusters, i.e. $\{a_{ij} = \mu \mid i \in \mathbf{I}, j \in \mathbf{J}\}$;
2. constant-row or constant-column biclusters, i.e. $\{\mathbf{a}_{ij}^T = \mu_i \mathbf{1}_J \mid i \in \mathbf{I}\}$ or $\{\mathbf{a}_{ij} = \mu_j \mathbf{1}_I \mid j \in \mathbf{J}\}$ where $\mathbf{1}_J$ or $\mathbf{1}_I$ is a column vector of ones;
3. additive-row or additive-column biclusters, i.e. $\{\mathbf{a}_{ij}^T = \mu_i \mathbf{1}_J + \mathbf{a}_{kj}^T \mid i, k \in \mathbf{I}\}$ or $\{\mathbf{a}_{ij} = \mu_j \mathbf{1}_I + \mathbf{a}_{ik} \mid j, k \in \mathbf{J}\}$;
4. multiplicative-row or multiplicative-column biclusters, i.e. $\{\mathbf{a}_{ij}^T = \mu_i \mathbf{a}_{kj}^T \mid i, k \in \mathbf{I}\}$ or $\{\mathbf{a}_{ij} = \mu_j \mathbf{a}_{ik} \mid j, k \in \mathbf{J}\}$.

Let us consider the analysis of gene expression data (*[gene, experimental condition]*) as an example. Our aim is to identify gene groups with similar behaviors or functions, such as a group of genes that are highly correlated under a group of experimental conditions [2, 12, 35]. In this regard, a constant bicluster means that a group of genes have the same expression level under a group of conditions, thus exhibiting certain kinds of homogeneity [35]. Similarly, constant-row, constant-column, additive and multiplicative biclusters can also reveal genes with related behaviours or functions, which can be coordinately investigated and targeted in gene regulation [6, 12]. Specifically, a constant-row bicluster means that each gene in a group has the same expression level under all conditions in a group, but different genes may have different expression levels. A constant-column bicluster means that all genes in a group have the same expression level under each condition in a group, but a gene may have different expression levels for different conditions. In an additive bicluster, expression levels of all genes in a group under one condition is higher or lower by a constant than those under another condition. In a multiplicative bicluster, expression levels of all genes in a group under one condition is a multiple of those under another condition.

Most biclustering techniques permute the original matrix and optimize a scoring function. Commonly-used scoring functions include the sum of squares [11] in Eq (3) and the mean squared residue score [2] in Eq (4),

$$SSQ = \sum_{i \in I, j \in J} (a_{ij} - \bar{A}_{IJ})^2 \quad (3)$$

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - \bar{a}_{ij} - \bar{a}_{Ij} + \bar{A}_{IJ})^2 \quad (4)$$

where \bar{A}_{IJ} is the mean of sub-matrix A_{IJ} , and \bar{a}_{ij} and \bar{a}_{Ij} are the means of row a_{ij} and column a_{ij} , respectively. A_{IJ} is defined as a δ -bicluster if $H(I, J) \leq \delta$, where $\delta (\delta > 0)$ is a pre-specified residue score threshold value.

Unfortunately, $H(I, J)$ can be used only to detect biclusters of types (a), (b) and (c), but not type (d). Therefore, a more general scoring function was proposed [36] for a thorough bicluster-search. This function $S(I, J)$, as expressed in Eq (5), is derived from Pearson's correlation:

$$S(I, J) = \min_{i \in I, j \in J} (S_{ij}, S_{ji}) \quad (5)$$

where $S_{ij} = 1 - \frac{1}{|J|-1} \sum_{j \neq k \in J} |\rho(a_{ij}, a_{ik})|$ and $S_{ji} = 1 - \frac{1}{|I|-1} \sum_{i \neq k \in I} |\rho(a_{ij}, a_{kj})|$. $\rho(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$ is Pearson's correlation between two vectors x and y . Normally, a lower $S(I, J)$ value represents a stronger coherence among the involved rows or columns [36]. Similarly as above, we can define a δ -bicluster if $S(I, J) \leq \delta$, where $\delta > 0$.

Biclustering Based on Singular Value Decomposition (SVD) and Clustering in Singular Vector Matrices. In biclustering analysis, SVD-based methods play an important role and have been broadly applied to detect significant biclusters. Representative methods include sparse SVD (SSVD), regularized SVD (RSVD), robust regularized SVD (RobRSVD), nonnegative matrix factorization (NMF) and nonsmooth-NMF (nsNMF) [36–39].

Let A be a matrix of size $N \times M$, and its SVD can be defined as follows,

$$A = U\Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (6)$$

where r is the rank of \mathbf{A} , $\mathbf{U} = [\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_r]$ is an $N \times r$ matrix of left orthonormal singular vectors, $\mathbf{V} = [\mathbf{v}_1 \mathbf{v}_2 \dots \mathbf{v}_r]$ is an $M \times r$ matrix of right orthonormal singular vectors, and $\Sigma = \text{diag}(\sigma_1 \sigma_2 \dots \sigma_r)$ is an $r \times r$ diagonal matrix with positive singular values ($\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$).

In Eq (6), $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$ is a rank-one matrix that is called an SVD layer. Normally, the SVD layers corresponding to large σ_k values are regarded as effective signals, while the rest are considered as noise [36]. Based on effective SVD layers, a rank- l ($l < r$) approximation of \mathbf{A} can be derived by minimizing the squared Frobenius norm:

$$\begin{aligned}\mathbf{A} \approx \mathbf{A}^{(l)} &= \sum_{i=1}^l \sigma_i \mathbf{u}_i \mathbf{v}_i^T \\ &= \arg \min_{\text{rank}(\mathbf{A}^*)=l} \| \mathbf{A} - \mathbf{A}^* \|_F^2 \\ &= \arg \min_{\text{rank}(\mathbf{A}^*)=l} \text{tr}\{ (\mathbf{A} - \mathbf{A}^*)(\mathbf{A} - \mathbf{A}^*)^T \}\end{aligned}\quad (7)$$

In order to develop SVD-based techniques for co-cluster analysis of high-order tensors, let us first study the properties of biclusters in 2D singular vector spaces. For simplicity, merely coherent patterns along columns are considered here.

Proposition 1: If \mathbf{A}_c is a bicluster with size $n \times d$, then $\text{rank}(\mathbf{A}_c) \leq 2$.

Proof: First, we can rewrite \mathbf{A}_c as follows,

$$\mathbf{A}_c = [\mathbf{a}_{:,1}^c \ \mathbf{a}_{:,2}^c \ \dots \ \mathbf{a}_{:,d}^c] \quad (8)$$

If \mathbf{A}_c corresponds to a constant or constant-column bicluster, then $\text{rank}(\mathbf{A}_c) \leq 1$. Otherwise, if \mathbf{A}_c corresponds to a multiplicative-column bicluster, then \mathbf{A}_c can be formulated as Eq (9),

$$\mathbf{A}_c = [\mathbf{a}_{:,1}^c \ k_2 \mathbf{a}_{:,1}^c \ \dots \ k_d \mathbf{a}_{:,1}^c] \quad (9)$$

As well, here the rank of \mathbf{A}_c is no more than one ($\text{rank}(\mathbf{A}_c) \leq 1$), since $k_2 \mathbf{a}_{:,1}^c, \dots, k_d \mathbf{a}_{:,1}^c$ are linearly dependent on $\mathbf{a}_{:,1}^c$.

At last, if \mathbf{A}_c corresponds to an additive bicluster, then the following equation can be derived,

$$\mathbf{A}_c = [\mathbf{a}_{:,1}^c \ \mathbf{a}_{:,1}^c + b_2 \mathbf{1}_n \ \dots \ \mathbf{a}_{:,1}^c + b_d \mathbf{1}_n] \quad (10)$$

Here, at most two vectors, such as $\mathbf{a}_{:,1}^c$ and $\mathbf{a}_{:,1}^c + b_2 \mathbf{1}_n$, are linearly independent. Therefore, $\text{rank}(\mathbf{A}_c) \leq 2$.

Proposition 2: Assume \mathbf{A}_c with size $s \times t$ is a bicluster. If $\mathbf{A}_c = \mathbf{U}_c \Sigma_c \mathbf{V}_c^T$, where \mathbf{U}_c and \mathbf{V}_c are the left and right singular vector matrices respectively of \mathbf{A}_c and $\Sigma_c = \text{diag}(\sigma_1^c, \sigma_2^c, \dots, \sigma_s^c)$ contains the singular values of \mathbf{A}_c with $\sigma_1^c \geq \sigma_2^c \geq \dots \geq \sigma_s^c$, then each column of \mathbf{A}_c can be represented as a linear combination of first two columns, \mathbf{u}_1^c and \mathbf{u}_2^c , of \mathbf{U}_c , and each row of \mathbf{A}_c can be represented as a linear combination of first two columns, \mathbf{v}_1^c and \mathbf{v}_2^c , of \mathbf{V}_c .

Proof: According to **Proposition 1**, the rank of \mathbf{A}_c is at most 2. Because $\sigma_m^c = 0$ for $m > 2$, $\mathbf{A}_c = \mathbf{U}_c \Sigma_c \mathbf{V}_c^T$ can be rewritten as

$$\begin{aligned} \mathbf{A}_c &= \mathbf{U}_c \Sigma_c \mathbf{V}_c^T \\ &= [\mathbf{u}_1^c \quad \mathbf{u}_2^c] \begin{bmatrix} \sigma_1^c & 0 \\ 0 & \sigma_2^c \end{bmatrix} \begin{bmatrix} (\mathbf{v}_1^c)^T \\ (\mathbf{v}_2^c)^T \end{bmatrix} \\ &= [\mathbf{u}_1^c \quad \mathbf{u}_2^c] \begin{bmatrix} \sigma_1^c & 0 \\ 0 & \sigma_2^c \end{bmatrix} \begin{bmatrix} v_{11}^c & v_{12}^c & \cdots & v_{1d}^c \\ v_{21}^c & v_{22}^c & \cdots & v_{2d}^c \end{bmatrix} \\ &= [\sigma_1^c v_{11}^c \mathbf{u}_1^c + \sigma_2^c v_{21}^c \mathbf{u}_2^c \quad \sigma_1^c v_{12}^c \mathbf{u}_1^c + \sigma_2^c v_{22}^c \mathbf{u}_2^c \quad \cdots \quad \sigma_1^c v_{1d}^c \mathbf{u}_1^c + \sigma_2^c v_{2d}^c \mathbf{u}_2^c] \end{aligned} \quad (11)$$

Let $\alpha_{j1} = \sigma_1^c v_{1j}^c$ and $\alpha_{j2} = \sigma_2^c v_{2j}^c$, then the j -th column of \mathbf{A}_c is

$$\mathbf{a}_j^c = \alpha_{j1} \mathbf{u}_1^c + \alpha_{j2} \mathbf{u}_2^c \quad (12)$$

That is,

$$\left\{ \begin{array}{l} \alpha_{j1} u_{11}^c + \alpha_{j2} u_{12}^c = a_{1j}^c \\ \alpha_{j1} u_{21}^c + \alpha_{j2} u_{22}^c = a_{2j}^c \\ \vdots \\ \alpha_{j1} u_{s1}^c + \alpha_{j2} u_{s2}^c = a_{sj}^c \end{array} \right. \quad (13)$$

Thus, each column of \mathbf{A}_c can be represented as a linear combination of first two columns, \mathbf{u}_1^c and \mathbf{u}_2^c , of \mathbf{U}_c . Geometrically, Eqs (12) and (13) mean that points (u_{m1}^c, u_{m2}^c) ($m = 1, 2, \dots, s$) are distributed on a line.

Similarly, we can obtain

$$\mathbf{a}_i^c = \beta_{i1} \mathbf{v}_1^c + \beta_{i2} \mathbf{v}_2^c \quad (14)$$

and

$$\left\{ \begin{array}{l} \beta_{i1} v_{11}^c + \beta_{i2} v_{12}^c = a_{i1}^c \\ \beta_{i1} v_{21}^c + \beta_{i2} v_{22}^c = a_{i2}^c \\ \vdots \\ \beta_{i1} v_{t1}^c + \beta_{i2} v_{t2}^c = a_{it}^c \end{array} \right. \quad (15)$$

Thus, each row of \mathbf{A}_c can be represented as a linear combination of first two columns, \mathbf{v}_1^c and \mathbf{v}_2^c , of \mathbf{V}_c . Geometrically, Eqs (14) and (15) mean that points (v_{m1}^c, v_{m2}^c) ($m = 1, 2, \dots, t$) are distributed on a line.

Proposition 3: Assume \mathbf{A}_{c1} and \mathbf{A}_{c2} with sizes $s_1 \times t$ and $s_2 \times t$ are two different biclusters,

where $s_1 + s_2 = s$. Let $\mathbf{A}_c = \begin{bmatrix} \mathbf{A}_{c1} \\ \mathbf{A}_{c2} \end{bmatrix}$. If $\mathbf{A}_c = \mathbf{U}_c \Sigma_c \mathbf{V}_c^T$, where \mathbf{U}_c and \mathbf{V}_c are the left and right singular vectors matrices respectively of \mathbf{A}_c and $\Sigma_c = \text{diag}(\sigma_1^c, \sigma_2^c, \dots, \sigma_s^c)$ contains the singular values of \mathbf{A}_c with $\sigma_1^c \geq \sigma_2^c \geq \dots \geq \sigma_s^c$, then each column of \mathbf{A}_{c1} can be represented as a linear combination of s_1 rows of first four columns, $\mathbf{u}_1^{c1}, \mathbf{u}_2^{c1}, \mathbf{u}_3^{c1}, \mathbf{u}_4^{c1}$, of \mathbf{U}_c , each column of \mathbf{A}_{c2} can be represented as another linear combination of rows $s_1 + 1$ to $s_1 + s_2$ of first four columns, \mathbf{u}_1^{c2} ,

$\mathbf{u}_2^c, \mathbf{u}_3^c, \mathbf{u}_4^c$, of \mathbf{U}_c , and each row of \mathbf{A}_c can be represented as a linear combination of first four columns, $\mathbf{v}_1^c, \mathbf{v}_2^c, \mathbf{v}_3^c, \mathbf{v}_4^c$, of \mathbf{V}_c .

Proof: According to **Proposition 1**, the rank of \mathbf{A}_{c1} is at most 2, and so is the rank of \mathbf{A}_{c2} .

Thus, $\text{rank}(\mathbf{A}_c) \leq 4$. Let $\mathbf{U}_c = \begin{bmatrix} \mathbf{U}_{c1} \\ \mathbf{U}_{c2} \end{bmatrix}$, where \mathbf{U}_{c1} represents the first s_1 rows and \mathbf{U}_{c2} the remaining s_2 rows of \mathbf{U}_c . Because $\sigma_m^c = 0$ for $m > 4$, $\mathbf{A}_c = \mathbf{U}_c \Sigma_c \mathbf{V}_c^T$ can be rewritten as

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{A}_{c1} \\ \mathbf{A}_{c2} \end{bmatrix} = \begin{bmatrix} [\mathbf{u}_1^{c1} \quad \mathbf{u}_2^{c1} \quad \mathbf{u}_3^{c1} \quad \mathbf{u}_4^{c1}] \\ [\mathbf{u}_1^{c2} \quad \mathbf{u}_2^{c2} \quad \mathbf{u}_3^{c2} \quad \mathbf{u}_4^{c2}] \end{bmatrix} \begin{bmatrix} \sigma_1^c & & & \\ & \sigma_2^c & & \\ & & \sigma_3^c & \\ & & & \sigma_4^c \end{bmatrix} \begin{bmatrix} (\mathbf{v}_1^c)^T \\ (\mathbf{v}_2^c)^T \\ (\mathbf{v}_3^c)^T \\ (\mathbf{v}_4^c)^T \end{bmatrix} \quad (16)$$

where \mathbf{u}_m^{c1} ($m = 1, 2, 3, 4$) are the first four columns of \mathbf{U}_{c1} and \mathbf{u}_m^{c2} ($m = 1, 2, 3, 4$) are the first four columns of \mathbf{U}_{c2} . Similar to the proof for **Proposition 2**, we can obtain

$$\mathbf{a}_j^{c1} = a_{j1}^{c1} \mathbf{u}_1^{c1} + a_{j2}^{c1} \mathbf{u}_2^{c1} + a_{j3}^{c1} \mathbf{u}_3^{c1} + a_{j4}^{c1} \mathbf{u}_4^{c1} \quad (17)$$

$$\mathbf{a}_j^{c2} = a_{j1}^{c2} \mathbf{u}_1^{c2} + a_{j2}^{c2} \mathbf{u}_2^{c2} + a_{j3}^{c2} \mathbf{u}_3^{c2} + a_{j4}^{c2} \mathbf{u}_4^{c2} \quad (18)$$

where \mathbf{a}_j^{c1} represents the j -th column of \mathbf{A}_{c1} or the first s_1 elements in the j -th column of \mathbf{A}_c , and \mathbf{a}_j^{c2} represents the j -th column of \mathbf{A}_{c2} or the remaining s_2 elements in the j -th column of \mathbf{A}_c .

Geometrically, Eq.(17) means that points $(u_{m1}^c, u_{m2}^c, u_{m3}^c, u_{m4}^c)$ ($m = 1, 2, 3, \dots, s_1$) are distributed on a hyperplane, and that points $(u_{m1}^c, u_{m2}^c, u_{m3}^c, u_{m4}^c)$ ($m = s_1 + 1, s_1 + 2, s_1 + 3, \dots, s_1 + s_2$) are distributed on another hyperplane. Similarly, we can show that the points $(v_{m1}^c, v_{m2}^c, v_{m3}^c, v_{m4}^c)$ ($m = 1, 2, \dots, t$) are also distributed on a hyperplane.

In practical applications, biclusters are embedded in a large matrix with irrelevant elements. Additionally, the biclusters themselves can have noise. Three examples are shown below:

XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
XAXXXXAXXX	XAXAXXAXXX	XAXXXAXXXA
XXXXXXXXXX	XAXAXXAXXX	XXXXXXXXXX
XXXXXXXXXX	XXXXXXXXXX	XAXXXAXXXA
XAXXXXAXXX	XAXAXXAXXX	XAXXXAXXXA
XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
XXXXXXXXXX	XBXBXXBXXX	XXBXBBXXXX
XAXXXXAXXX	XXXXXXXXXX	XXBXBBXXXX
XXXXXXXXXX	XBXBXXBXXX	XXXXXXXXXX
XXXXXXXXXX	XBXBXXBXXX	XXBXBBXXXX
Matrix1		Matrix2
Matrix3		

In these matrices, “X” represents entries of irrelevant elements, and “A” and “B” represent the entries of two biclusters A and B respectively. The actual values at the location marked by “X” can be different, and they are background noise. The values at the locations marked by “A” and “B” should form bicluster patterns as described in Section “Biclusters in Matrices”.

Due to noise, the rank of Matrix 1 can be greater than 2. We can remove small singular values using a method to be discussed in Section “HOSVD”. Assume that we still retain 2 singular values, then Eqs (12) to (15) are only approximations for rows 2, 5 and 8. Remember that our task in biclustering is to find these row indices (and column indices 2, 3 and 7). That is, we do not know beforehand rows 2, 5 and 8 contain a bicluster and we need to identify them. Assume that we take SVD of Matrix 1, then from the left singular vector matrix U, points (u_{m1}, u_{m2}) ($m = 2, 5, 8$) should form a line approximately according to Eqs (12) and (13), while points (u_{m1}, u_{m2}) ($m \neq 2, 5, 8$) will be distributed randomly and do not satisfy these equations. Our task now is to detect the line from all points (u_{m1}, u_{m2}) ($1 \leq m \leq 10$). Once the line is detected, we can then determine which points are on the line. The row indices of these points correspond to the locations of the bicluster. Similarly, we can identify relevant columns of the bicluster by detecting lines using the right singular vector matrix V. To identify biclusters in Matrices 2 and 3, we need to find hyperplanes in singular vector spaces. The lines and hyperplanes can be detected using the linear grouping algorithm (LGA) to be discussed in Section “Linear Grouping Algorithm (LGA)”.

Identification of Co-clusters in High-order Tensors

Co-clusters in High-order Tensors. The hyperplane model in singular vector spaces can be extended to the analysis of higher-order tensor data. For example, a co-cluster (represented by a sub-tensor $\mathcal{A}_{IJK} \in \mathbf{R}^{I \times J \times K}$) in a 3-mode (3D) tensor $\mathcal{A}_{I_1 I_2 I_3} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$, where $\{\mathbf{I} \in \mathbf{I}_1, \mathbf{J} \in \mathbf{I}_2, \mathbf{K} \in \mathbf{I}_3\}$, can be similarly defined as in the 2D case. This definition involves the pre-defined fibers and slices in a 3D tensor. Now we use constant co-clusters as examples,

1. $\mathcal{A}_{IJK} \in \mathbf{R}^{I \times J \times K}$ corresponds to a constant co-cluster if $\{a_{ijk} = \mu \mid i \in \mathbf{I}, j \in \mathbf{J}, k \in \mathbf{K}\}$;
2. a constant-column-fiber co-cluster can be expressed as $\{\mathbf{a}_{Ijk} = \mu_{jk} \mathbf{1}_I \mid j \in \mathbf{J}, k \in \mathbf{K}\}$;
3. a constant-horizontal-slice co-cluster can be defined as $\{\mathbf{A}_{j\mathbf{K}} = \mu_i \mathbf{1}_{j\mathbf{K}} \mid i \in \mathbf{I}\}$;

Here $\mathbf{1}_I$ or $\mathbf{1}_{j\mathbf{K}}$ is a vector or matrix of ones. Accordingly, additive and multiplicative co-clusters can be also defined.

To evaluate the significance of co-clusters in high-order tensors, a scoring function generalized from that for matrices in Eq (5) can be used. Let $S_{I_1 \dots I_n \dots I_N}$ represent the average of $S_{I_1 i_n}, \dots, S_{I_{n-1} i_n}, S_{i_n I_{n+1}}, \dots, S_{i_n I_N}$, then we can define the scoring function as

$$\begin{aligned} & S(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_N) \\ &= \min_{i_1 \in \mathbf{I}_1, i_2 \in \mathbf{I}_2, \dots, i_N \in \mathbf{I}_N} (S_{I_1 I_2 \dots I_N}, S_{I_1 I_2 \dots I_N}, S_{I_1 I_2 \dots I_N}) \end{aligned} \quad (19)$$

Similarly, a lower score indicates a higher significance. A δ -co-cluster can be further defined if $S(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_N) \leq \delta$ ($\delta > 0$). In particular, we name such a co-cluster in a 3D tensor as a δ -tricluseter.

HOSVD. Similar to SVD, HOSVD that decomposes a tensor into a core tensor and a singular vector matrix along each mode is employed in our method, to extract co-clusters in high-order tensors [23, 40, 41].

The HOSVD of an N-mode tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$ can be expressed as follows,

$$\mathcal{A} = \mathcal{T} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)} \times_5 \mathbf{U}^{(5)} \times_6 \mathbf{U}^{(6)} \quad (20)$$

where $\mathbf{U}^{(n)} \in \mathbf{R}^{I_n \times r_n}$ ($n = 1, \dots, N$) is a factor (singular vector) matrix, and $\mathcal{T} \in \mathbf{R}^{r_1 \times r_2 \times \dots \times r_N}$ is the core tensor. The SVD of a matrix \mathbf{A} in Eq (6) can be formulated in a similar format with $\mathcal{T} = \Sigma \in \mathbf{R}^{I_1 \times I_2}$, as a special case of Eq (20).

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T = \Sigma \times_1 \mathbf{U} \times_2 \mathbf{V} \quad (21)$$

On the other hand, HOSVD can be expressed as a matrix format, using the unfolded matrices along each mode,

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)}\Sigma^{(n)}(\mathbf{V}^{(n)})^T, \quad n = 1, \dots, N \quad (22)$$

That is, we obtain a matrix $\mathbf{A}_{(n)}$ of size $I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N$ for mode n , and $\mathbf{U}^{(n)}$ is the left singular vector matrix of $\mathbf{A}_{(n)}$. When the tensor \mathcal{A} is unfolded to a matrix $\mathbf{A}_{(n)}$, a co-cluster in \mathcal{A} will be unfolded to a bicluster in $\mathbf{A}_{(n)}$. From $\mathbf{U}^{(n)}$, we can find the row indices of $\mathbf{A}_{(n)}$ that contain a bicluster, according to **Propositions 1 to 3**. These row indices correspond to the locations in the tensor along mode n . By combining the row indices of biclusters in all unfolded matrices $\mathbf{A}_{(n)}$ ($n = 1, \dots, N$), we can then find the co-cluster in an N -dimensional space or N -mode tensor \mathcal{A} . Now the major task is to find hyperplanes for each singular vector matrix $\mathbf{U}^{(n)}$. That is, the problem of detecting co-clusters in a multi-dimensional space has been effectively converted to the detection of biclusters in singular vector spaces.

For a tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$, the n -rank of \mathcal{A} ($\text{rank}_n(\mathcal{A})$) is defined as the column rank of $\mathbf{A}_{(n)}$. Let $r_n = \text{rank}_n(\mathcal{A})$, ($n = 1, \dots, N$), then \mathcal{A} has the rank of (r_1, r_2, \dots, r_N) . Inspired by the rank- l ($l < r$) SVD approximation of matrix \mathbf{A} in Eq (7), the singular-value truncation also can be used to reveal effective signals and reduce noises in a tensor [23]. Accordingly, a portion of singular values $\Sigma^{(n)}$ in Eq (22) will be adopted. Thus, we can define a truncated HOSVD depending on the matrix format as follows, *for a tensor \mathcal{A} , its decomposition of rank- $(\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_N)$, with $\tilde{r}_n < r_n = \text{rank}_n(\mathcal{A})$ for at least one mode (n), is called a truncated HOSVD* [40]. This concept for a 3-mode tensor is shown in Fig 1.

The optimal rank $(\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_N)$ of a tensor can be determined from a compromise between accuracy and number of singular values used. Because we use the unfolded matrices to

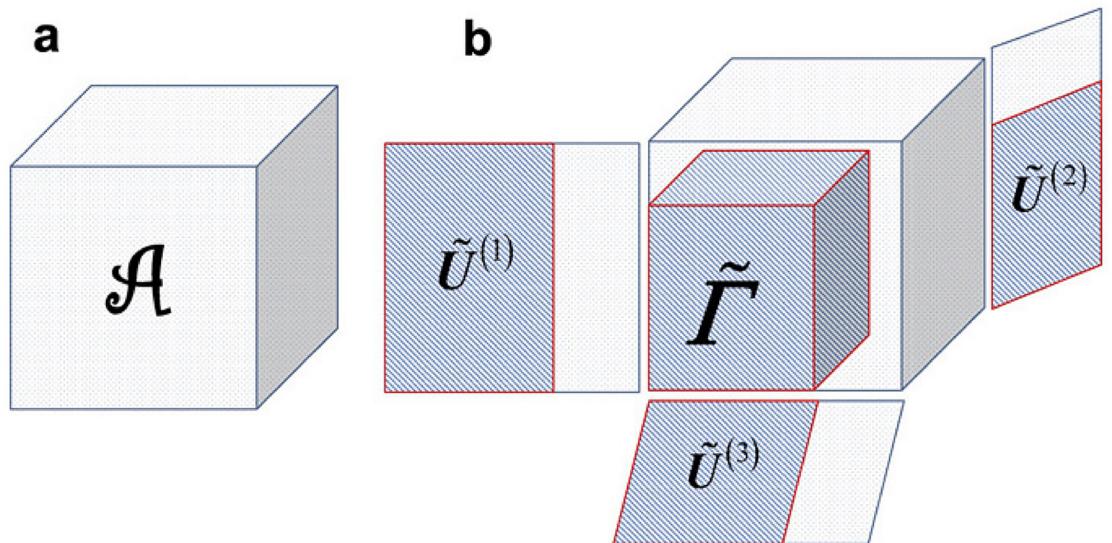


Fig 1. The truncated HOSVD of a 3-mode tensor. Part a shows the original tensor \mathcal{A} , and Part b displays the concept of a truncated HOSVD of \mathcal{A} .

doi:10.1371/journal.pone.0162293.g001

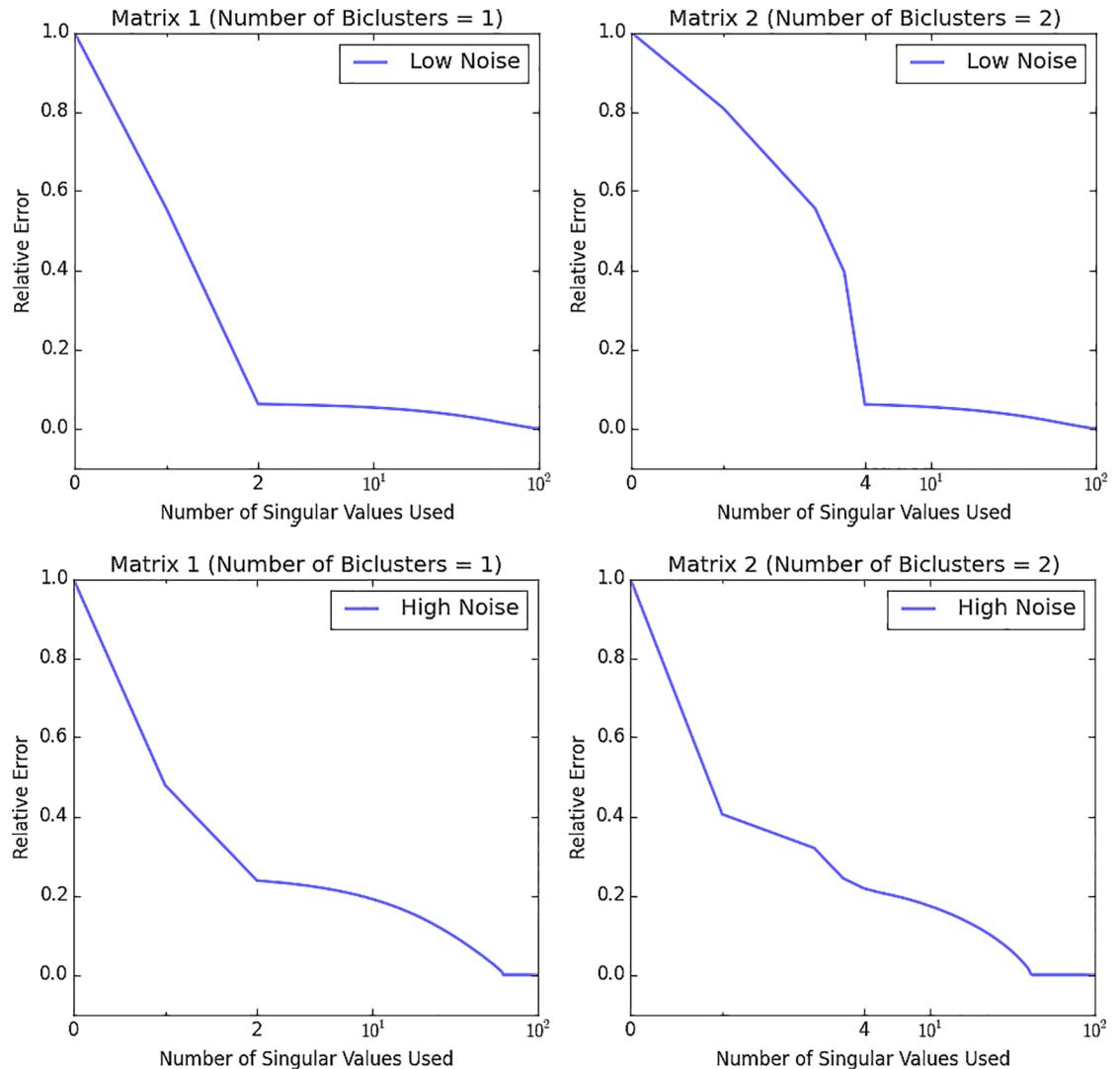


Fig 2. Derivation of the optimal rank. According to the relative error curve with different number of singular values used, the optimal rank of the Matrix 1 and Matrix 2 are 2 and 4, respectively.

doi:10.1371/journal.pone.0162293.g002

determine the row indices of co-clusters, we find the optimal rank \tilde{r}_n of each unfolded matrix $\mathbf{A}_{(n)}$ and the optimal rank of the tensor is then $(\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_N)$. The accuracy of a matrix $\tilde{\mathbf{A}}$ is defined using the Frebenius norm as $acc = \frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2}{\|\mathbf{A}\|_F^2}$, where $\tilde{\mathbf{A}}$ is reconstructed from SVD after truncation of mall singular values. The relative error is defined as $1 - acc$. Fig 2 presents the relative error curves for two examples \mathbf{A} and $\tilde{\mathbf{A}}$ with different numbers of singular values used. The two 2D synthetic matrices of size 100×100 , which are similar to Matrix 1 and Matrix 2 discussed below **Proposition 3**, are embedded with one and two additive biclusters of size 10×10 , respectively. Each additive bicluster is produced from a seed column of random numbers distributed in a normal distribution $N(0, \zeta^2)$ ($\zeta = 6$ for low nosie level and $\zeta = 1.25$ for high noise level) and each of the other 9 columns is produced by adding to the seed column a random number with the standard normal distribution. The background of each matrix also contains random numbers with the standard normal distribution. As shown in Fig 2, the corner

point corresponds to the optimal rank (2 and 4) of the two 2D synthetic matrices. As noise increases, the relative error curve becomes smooth and the corner point disappears. Through many simulation experiments, we find that a threshold at 20% relative error can be used to find the optimal rank value.

Linear Grouping Algorithm (LGA). As discussed in Section “Biclustering Based on Singular Value Decomposition (SVD) and Clustering in Singular Vector Matrices”, bicluster searching in a 2D matrix \mathbf{A} can be transformed into a hyperplane detection problem in the singular vector matrices produced by SVD, which can be generalized to a high-order tensor \mathcal{A} . In detail, a co-cluster in \mathcal{A} can be represented by the biclusters along each mode, and a series of linear relations among the HOSVD-generated singular vectors can be accordingly inferred as below,

$$\left\{ \begin{array}{l} \alpha_{11}\mathbf{u}_1^{(1)c} + \alpha_{12}\mathbf{u}_2^{(1)c} + \dots + \alpha_{1\tilde{r}_1}\mathbf{u}_{\tilde{r}_1}^{(1)c} + \beta_1 = \mathbf{0} \\ \alpha_{21}\mathbf{u}_1^{(2)c} + \alpha_{22}\mathbf{u}_2^{(2)c} + \dots + \alpha_{2\tilde{r}_2}\mathbf{u}_{\tilde{r}_2}^{(2)c} + \beta_2 = \mathbf{0} \\ \vdots \\ \alpha_{N1}\mathbf{u}_1^{(N)c} + \alpha_{N2}\mathbf{u}_2^{(N)c} + \dots + \alpha_{N\tilde{r}_N}\mathbf{u}_{\tilde{r}_N}^{(N)c} + \beta_N = \mathbf{0} \end{array} \right. \quad (23)$$

where the co-cluster is decomposed as $\mathcal{A}_c = \mathcal{T}_c \times_1 \tilde{\mathbf{U}}^{(1)c} \times_2 \tilde{\mathbf{U}}^{(2)c} \dots \times_N \tilde{\mathbf{U}}^{(N)c}$.

[Eq \(23\)](#) represents a group of hyperplanar (linear) relations in a multi-dimensional space, which can be named *hyperplanar co-clusters*. Importantly, similar to the 2D case, these hyperplanar relations shed light on the multi-dimensional co-cluster identification in tensors [5, 13, 36, 37, 39].

Specifically in our work, the *linear grouping algorithm* (LGA) was adopted to reveal the linear relations embedded in the singular vector matrices, which were generated by a truncated HOSVD. This model follows the linear patterns among the involved vectors [5]. Originally, LGA clustered data points by fitting a mixture of linear regression models [42], and later an evolved model based on an orthogonal regression approach was proposed, which provided favorable performances in applications with outliers [5]. Recently, this model has been improved to a robust linear clustering method [43]. A simple procedure of LGA can be described as **Algorithm 1**.

Algorithm 1 Procedure of LGA for a group of vectors $\mathbf{U}_{I_n \times \tilde{r}_n}$

procedure LGA ($\mathbf{U}_{I_n \times \tilde{r}_n} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{I_n}]^T, K$)

Step 1: Scale the variables through dividing them by the standard deviation,

$$\tilde{\mathbf{U}}_{I_n \times \tilde{r}_n} = [\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_{I_n}]^T,$$

where
$$\left\{ \begin{array}{l} \tilde{\mathbf{u}}_i = \frac{\mathbf{u}_i}{s_i} \\ s_i = \sqrt{\frac{1}{d-1}(\mathbf{u}_i - \bar{\mathbf{u}}_i)(\mathbf{u}_i - \bar{\mathbf{u}}_i)^T} \\ i = 1, 2, \dots, I_n \end{array} \right.$$

Step 2: Select K random sub-samples of size \tilde{r}_n ,

$$\mathbf{G}^0 = \{\mathbf{g}_1^0, \mathbf{g}_2^0, \dots, \mathbf{g}_K^0\}$$

$$\text{where } \left\{ \begin{array}{l} \mathbf{g}_k^0 = [\tilde{\mathbf{u}}_{k1}^0, \tilde{\mathbf{u}}_{k2}^0, \dots, \tilde{\mathbf{u}}_{k\tilde{r}_n}^0] \\ \tilde{\mathbf{u}}_{ki}^0 \in \tilde{\mathbf{U}}_{I_n \times \tilde{r}_n} \end{array} \right.$$

Step 3: Loop

for $j = 0, 1, \dots, J$ **do**

```

1. Initialize  $K$  orthogonal regression hyperplanes  $\mathbf{H}^j = \{\mathbf{h}_1^j, \mathbf{h}_2^j, \dots, \mathbf{h}_K^j\}$  by
   fitting the samples in  $\mathbf{G}^j = \{\mathbf{g}_1^j, \mathbf{g}_2^j, \dots, \mathbf{g}_K^j\}$ , resulting in
    $\mathbf{h}_k^j = \{(\hat{\alpha}_k, \hat{\beta}_k) \mid \hat{\alpha}_k^T \tilde{\mathbf{u}}_{ki}^j - \hat{\beta}_k = 0, \tilde{\mathbf{u}}_{ki}^j \in \mathbf{g}_k^j, \|\hat{\alpha}_k\| = 1, i = 1, 2, \dots, \tilde{r}_n\} \quad (k = 1, 2, \dots, K).$ 
2. Compute the distance between each hyperplane  $\mathbf{h}_k^j$  and each sample  $\tilde{\mathbf{u}}_i$ ,
    $d_{ik}^j = \text{distance}(\tilde{\mathbf{u}}_i, \mathbf{h}_k^j) = |\hat{\alpha}_k^T \tilde{\mathbf{u}}_i - \hat{\beta}_k|,$  where  $\begin{cases} k = 1, 2, \dots, K \\ i = 1, 2, \dots, I_n \end{cases}.$ 
3. Form  $K$  groups for  $\tilde{\mathbf{u}}_i$ ,  $\mathbf{G}^{j+1} = \{\mathbf{g}_1^{j+1}, \mathbf{g}_2^{j+1}, \dots, \mathbf{g}_K^{j+1}\}$ , where  $\tilde{\mathbf{u}}_i \in \mathbf{g}_k^{j+1}$  if
    $k = \text{argmin}_k(d_{ik}^j).$ 
4. Compute the evaluation function for each iteration,  $D^j = \sum_{k=1}^K \sum_{\tilde{\mathbf{u}}_i \in \mathbf{g}_k^{j+1}} d_{ik}^j.$ 
end for  $D^j$  reaches the minimum  $D^{opt}.$ 
Return  $\mathbf{G}^{opt}$ 
end procedure

```

The purpose of applying the LGA algorithm is to find the row indices of each co-cluster along each mode of a tensor from the corresponding unfolded matrix, as discussed above. This is done by detecting hyperplanes formed by some but not all points $(u_{m1}, u_{m2}, \dots, u_{md})$ ($m = 1, 2, \dots, I_n$) in the left singular vector space along each mode. Initially, we choose random points to compute the coefficients of hyperplanes. Then we assign each point to the closest hyperplane. This process is repeated to improve the result. The procedure is similar to the K-means algorithms. The difference is that we deal with hyperplanes while the K-means algorithm involves cluster centers only. We consider a point is on a hyperplane if its distance to the hyperplane is smaller than a pre-specified threshold, which is determined experimentally. From a set of points on the same hyperplane, we can then find the row indices of a corresponding co-cluster. All co-clusters detected from the hyperplanes are subject to an evaluation and elimination procedure discussed below.

Multi-dimensional Co-clustering in Tensors Based on HOSVD and LGA. HOSVD decomposes an N -mode tensor \mathcal{A} into a core tensor and singular vector matrices $\mathbf{U}^{(n)}$ along all modes. Hyperplanes embedded in each truncated singular vector matrix were revealed by LGA, based on the natural linear patterns among the vectors. Through combining the products along each mode, the high-order co-clusters in this tensor were successfully identified. To further filter such co-clusters, those having a score (Eq (20)) smaller than or equal to δ , namely $S(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_N) \leq \delta$ ($\delta > 0$), were extracted and regarded as more significant. A co-cluster is eliminated if it is a part of or identical to another co-cluster. The block diagram of our co-clustering method based on hyperplane detection in singular vector spaces (**HDSVS**) is shown in Fig 3.

Experiment Results

To verify **HDSVS**, several data sets, including multiple synthetic and biological tensors, were used in the experiments. Two synthetic tensors were constructed with increased noise and overlapped degree, to evaluate the effects of noise and overlapping complexity to the co-cluster identification. Two biological tensors were from *gene expression data from 12 multiple sclerosis patients under an IFN-β therapy* [44, 45] and *spatial/temporal lineage tracing data of embryonic cells in a crowd of Caenorhabditis elegans* [34, 46], and were used to test the performance of our method in practical problems. **HDSVS** can be successfully applied to co-clustering in high-order tensors. However, because most existing methods are designed for 2D data, we conducted comparisons of **HDSVS** with other methods using only matrices or second-order tensors in Section “Experiment Comparisons with Other Methods Using 2D Synthetic Data and 2D Yeast Gene Expression Data”. Specifically, 2D synthetic tensors generated based on well-

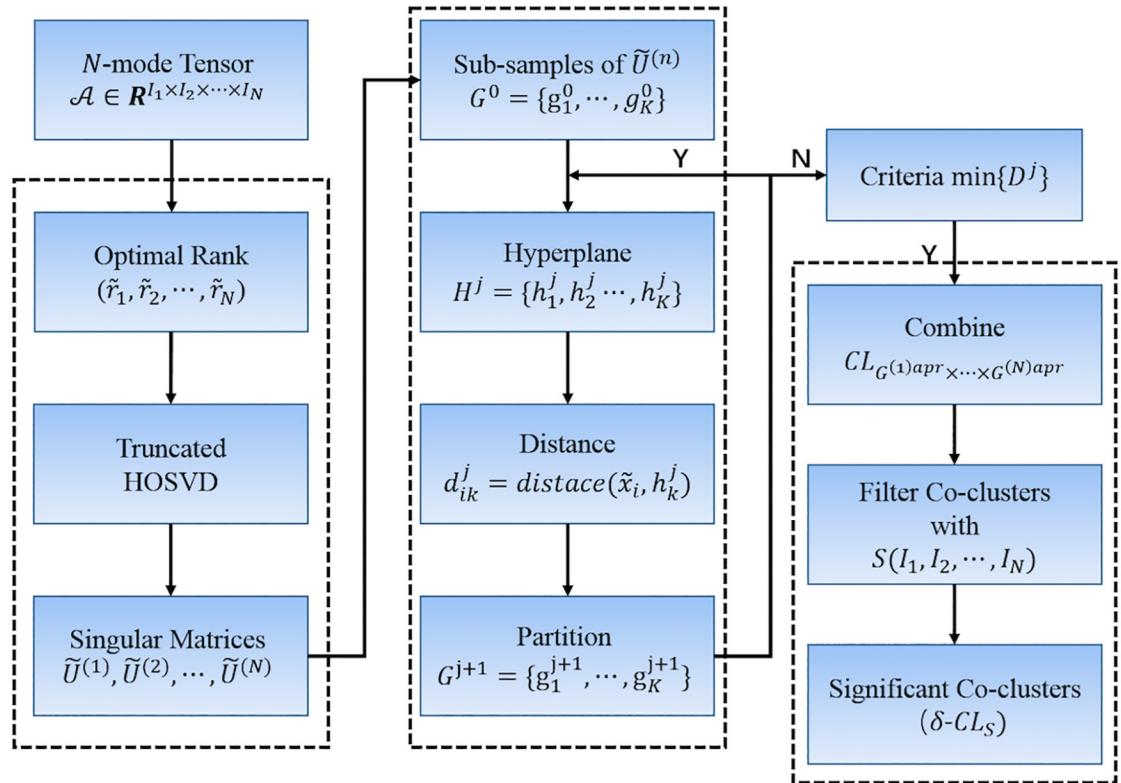


Fig 3. Block diagram of our co-clustering method based on hyperplane detection in singular vector spaces (HDSVS). In the left-hand side, the flow for a truncated HOSVD is shown. The LGA module is presented in the middle. The ranking procedure based on a scoring function, for revealing significant co-clusters (δ -CLs) in a tensor, is listed in the right-hand side.

doi:10.1371/journal.pone.0162293.g003

published principles [47] and a 2D yeast gene expression tensor [20] were adopted to compare the performance and robustness of our method with those of existing methods.

Evaluation of Noise and Overlapping Effects in Co-cluster Identification Using Synthetic Tensors

A matching scoring, generated by the Jaccard coefficient [14], was first defined to evaluate the agreement between a detected co-cluster and the true one. Let $CL_1 = (G_1^1, G_2^1, \dots, G_N^1)$ and $CL_2 = (G_1^2, G_2^2, \dots, G_N^2)$ be two co-clusters in a tensor $A \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, where $G_i^j \subseteq I_i$ ($i = 1, 2, \dots, N; j = 1, 2$) is a subset of the i th dimension of A . The matching score can be expressed as follows,

$$MS(CL_1, CL_2) = \max_{CL_1} \max_{CL_2} \frac{\sum_{i=1}^N |G_i^1 \cap G_i^2|}{\sum_{i=1}^N |G_i^1 \cup G_i^2|} \quad (24)$$

Further, we denote a true co-cluster as CL_{true} and a detected one as δ -CL, then a larger value of $MS(CL_{true}, \delta\text{-CL})$ represents a better detection. Based on such matching scores, effects of noise and overlapping complexity on the co-cluster identification will be discussed, using the two synthetic tensors.

In the first case, four types of CL_{true} ($10 \times 10 \times 10$), constant, constant-column-fiber, additive-fiber and multiplicative-fiber co-clusters, were embedded into a 3D tensor \mathcal{A}_1 ($100 \times 100 \times 100$), whose background was generated based on the standard normal distribution. The four types of co-clusters were generated as follows:

1. constant co-cluster, i.e. $\{a_{ijk} = 2 \mid i \in \mathbf{I}, j \in \mathbf{J}, k \in \mathbf{K}\}$;
2. constant-column-fiber co-cluster, i.e. $\{\mathbf{a}_{ijk} = \mu_{jk} \mathbf{1}_\mathbf{I} \mid j \in \mathbf{J}, k \in \mathbf{K}\}$, where μ_{jk} was randomly selected from $U(-2, 2)$;
3. additive-fiber co-cluster, i.e. $\{\mathbf{a}_{ijk} = \mu_{jk} \mathbf{1}_\mathbf{I} + \mathbf{a}_{\mathbf{I}(1)} \mid j \in \mathbf{J}, k \in \mathbf{K}\}$, where $\mathbf{a}_{\mathbf{I}(1)}$ is the first fiber of the co-cluster, μ_{jk} and each value of $\mathbf{a}_{\mathbf{I}(1)}$ were randomly selected from $U(-2, 2)$;
4. multiplicative-fiber co-cluster, i.e. $\{\mathbf{a}_{ijk} = \mu_{jk} \mathbf{a}_{\mathbf{I}(1)} \mid j \in \mathbf{J}, k \in \mathbf{K}\}$, where $\mathbf{a}_{\mathbf{I}(1)}$ is the first fiber of the co-cluster, where μ_{jk} and each value of $\mathbf{a}_{\mathbf{I}(1)}$ were randomly selected from $U(-2, 2)$;

Then Gaussian white noise with different signal-to-noise ratios (SNRs) was generated to degrade CL_{true} . The proposed **HDSVS** algorithm and PARAFAC with sparse latent factors [29] was then applied to the noisy tensors, after which $MS(CL_{true}, \delta\text{-}CL)$ was calculated. The experiment was performed 100 times for each method and the matching scores from all experiments were averaged to obtain the final score for comparison. The matching scores corresponding to various SNRs are summarized in Fig 4a. For the constant co-cluster, the matching score of **HDSVS** and PARAFAC are 1 for all SNRs. PARAFAC performs better than **HDSVS** when the SNR is low for the constant-column-fiber ($SNR \leq 15$) and the multiplicative-fiber ($SNR \leq 5$) co-cluster. However, as the SNR increases, **HDSVS** has higher matching scores than PARAFAC for the constant-column-fiber ($SNR \geq 20$) and the multiplicative-fiber ($SNR \geq 10$) co-cluster. HDSVS is much better than PARAFAC for additive-fiber co-clusters. As discussed above, constant, constant-column and multiplicative biclusters have rank 1, while additive biclusters have rank 2. Our proposed HDSVS algorithm is especially effective for additive co-clusters because the hyperplane model fits their linear structures well. The PARAFAC based co-clustering method relies on the K-means clustering formulation and can only work well with co-clusters that can be represented by their centers, corresponding to structures of rank 1.

Likewise, a 3D tensor \mathcal{A}_2 ($100 \times 100 \times 100$) with two overlapping co-clusters were generated. The experiment was also performed 100 times. Each time, the types of two overlapping co-clusters were chosen randomly. For simplicity, merely the overlapped cubic patterns were considered in the evaluation, and thus the overlapping degree v can be defined as the size of overlapped cubes in each dimension ($0 \leq v \leq 9$). As shown in Fig 4b, both methods have reasonably good performance for detecting overlapping co-clusters. However, for all overlapping degrees, HDSVS performs consistently better than PARAFAC.

Co-cluster Identification in Gene Expression Data from Sclerosis Patients under an IFN- β Therapy

A 3D tensor generated from the gene expression data of multiple sclerosis patients, who accepted a treatment of IFN- β injection, is discussed here. Twelve patients with western European ancestry, including eight females and four males (average age of 36.4), were recruited in this study. Fifteen milliliters of EDTA blood sample (at peripheral venous) were drawn from each patient, respectively at the baseline day and 2 days, 1 month, 1 year and 2 years after the initiation of an IFN- β therapy (<http://link.springer.com/article/10.1007/s12035-013-8463-1/fulltext.html>) [44, 45].

In detail, the gene expression data can be represented by a 3D tensor (gene \times patient \times time) of size $18862 \times 12 \times 5$. Considering the IFN- β therapy, we only kept the therapy-related genetic

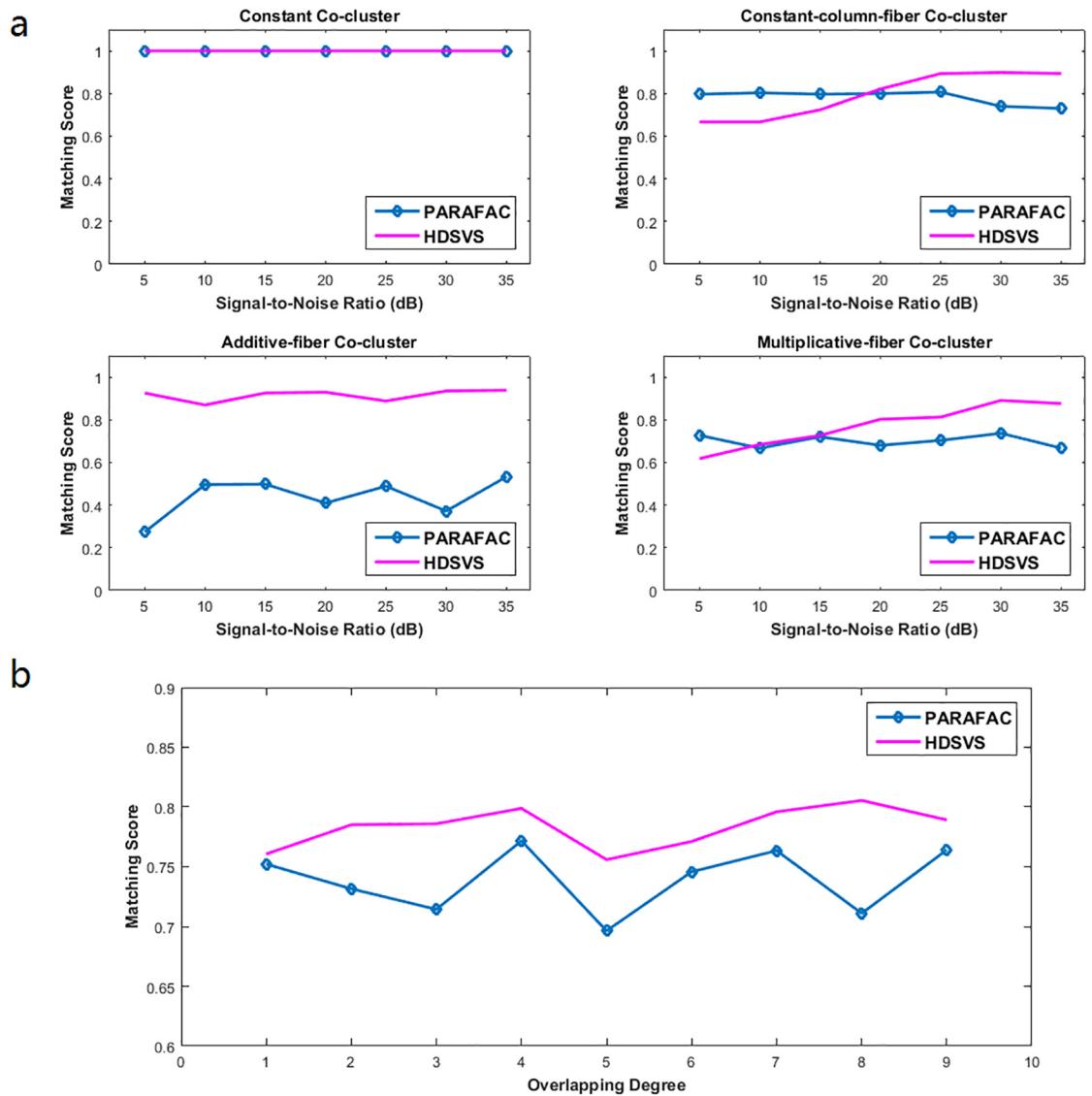


Fig 4. Effects of noise and overlapping complexity on co-cluster identification in two synthetic tensors. (a) Matching scores between true co-clusters and the detected ones, with different SNRs. (b) Matching scores between two overlapping co-clusters and the true ones, with various overlapping degrees.

doi:10.1371/journal.pone.0162293.g004

pathways (56 genes), leading to a simplified tensor of size $56 \times 12 \times 5$. Regarding each gene-patient matrix (time is fixed) as a layer, Fig 5 shows the heat maps for these five layers.

HDSVS was applied on this simplified tensor \mathcal{A} . Its core tensor \mathcal{T} and singular vector matrices $\mathbf{U}^{(n)}$ ($n = 1, 2, 3$) can be extracted by HOSVD, while considering noises, a truncated HOSVD $\tilde{\mathcal{A}}$ was implemented ($\tilde{\mathcal{T}}$ and $\tilde{\mathbf{U}}^{(n)}$). Specifically, the optimal rank for the truncated HOSVD is (2, 4, 4), which was derived based on the method discussed in Section “HOSVD” and Fig 2. Once the truncated HOSVD was obtained, LGA was used for the bicluster-detection along each mode ($\tilde{\mathbf{U}}^{(n)}$). For example, the 56 points in $\tilde{\mathbf{U}}^{(1)}$ can be linearly grouped into two patterns, and similar patterns can be derived for $\tilde{\mathbf{U}}^{(2)}$ and $\tilde{\mathbf{U}}^{(3)}$. Such linear patterns, at the first two dimensions of each $\tilde{\mathbf{U}}^{(n)}$ (or $\mathbf{U}^{(n)}$), are shown in Fig 6a to 6c, respectively. As a supplementary study, the first three dimensions of each $\tilde{\mathbf{U}}^{(n)}$ (or $\mathbf{U}^{(n)}$) are separately plotted in Fig 7a to 7c. Further,

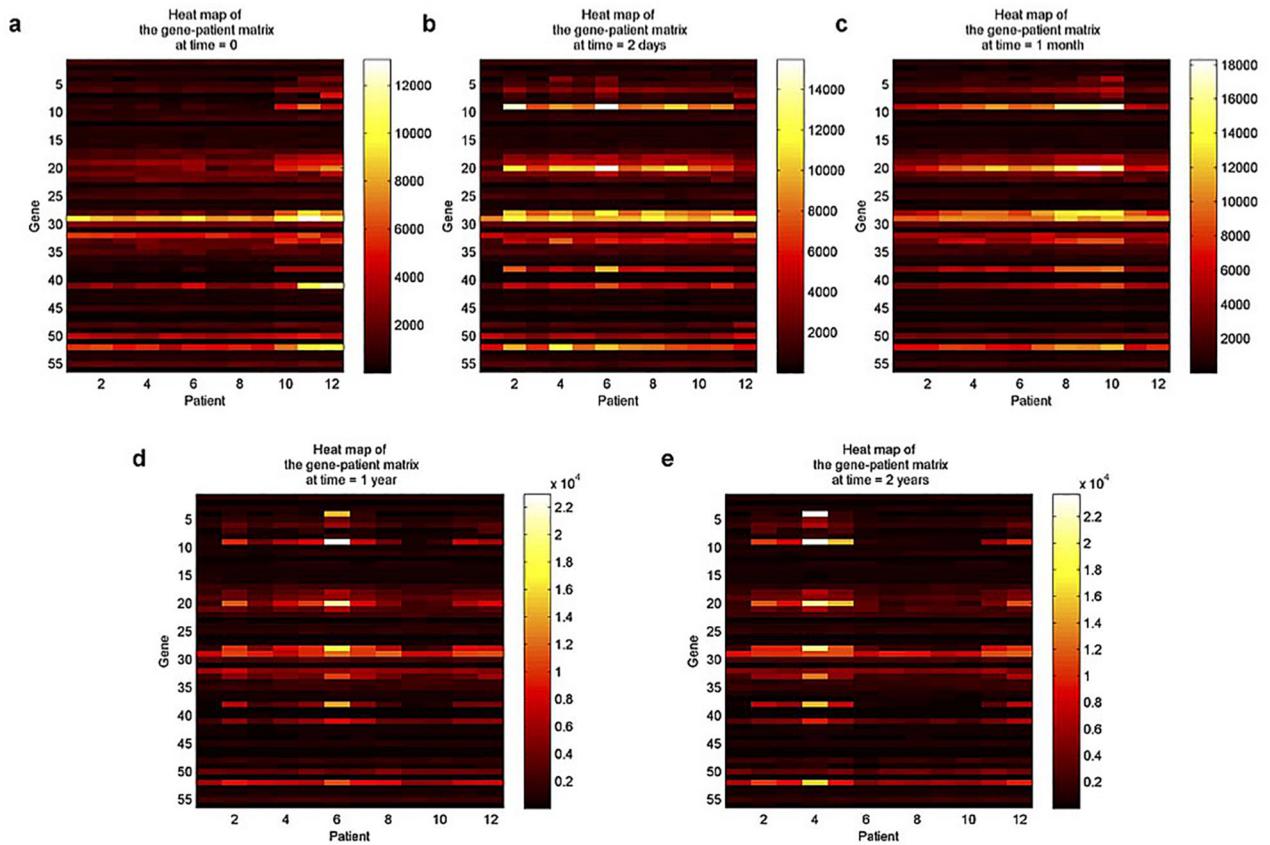


Fig 5. Heat maps for genepatient matrices at fixed time points. (a) to (e), Scenarios for genepatient heat maps corresponding to the baseline day, and 2 days, 1 month, 1 year and 2 years after the initiation of an IFN- β therapy.

doi:10.1371/journal.pone.0162293.g005

the linear or planar patterns of these points in the 3D space are displayed in Fig 7d to 7f. The planar structures consistently validate the grouping or clustering capability of HDSVS.

As shown in Fig 6, the 56 genes are divided into two linear groups denoted as E^i ($i = 1, 2$), the 12 patients correspond to three groups P^j ($j = 1, 2, 3$), and the 5 time points represent two

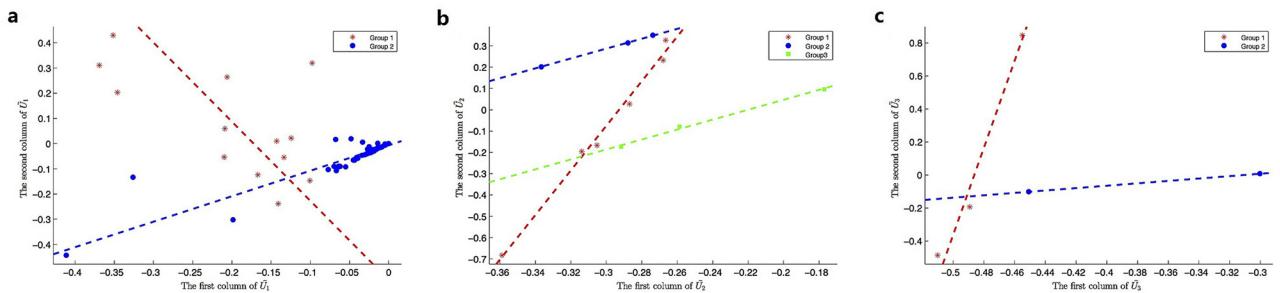


Fig 6. The linear patterns embedded in the 2D singular-vector space. (a) to (c), The linear groups along the directions of first two singular vectors of $U^{(n)}$ ($n = 1, 2, 3$), respectively.

doi:10.1371/journal.pone.0162293.g006

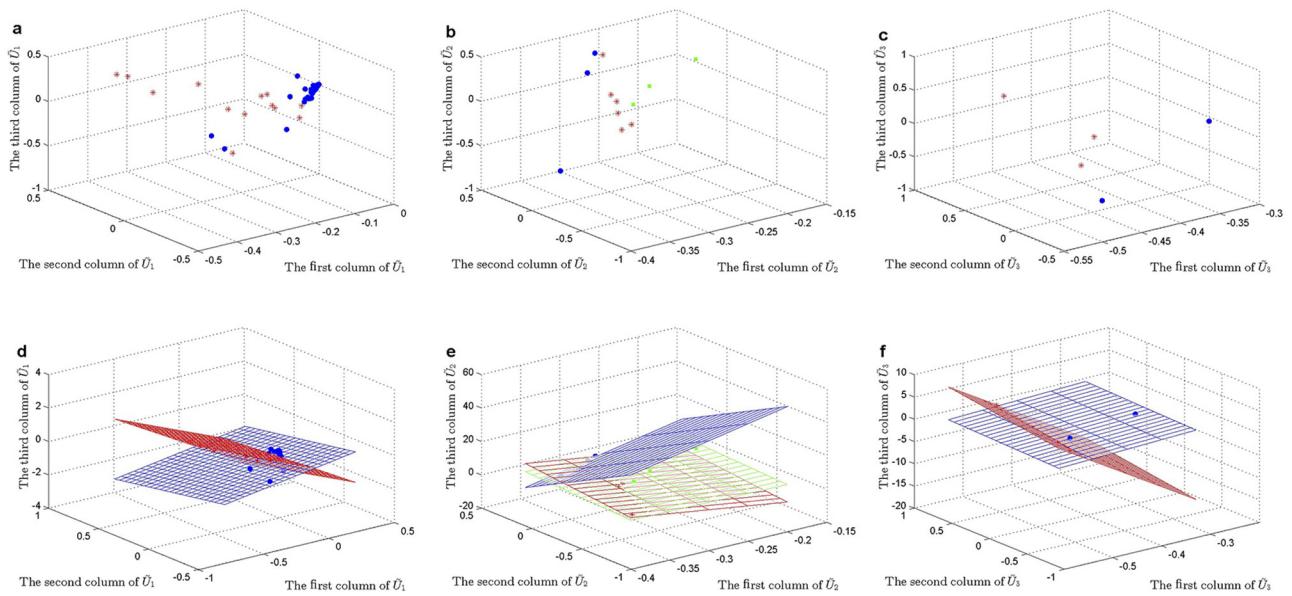


Fig 7. Linear patterns embedded in the 3D singular-vector space of the genexpatientxtime tensor. (a) to (c), The scatter plots along the directions of first three singular vectors of $\mathbf{U}^{(n)}$ ($n = 1, 2, 3$), respectively. (d) to (f), Linear or planar patterns of the 3D-points in (a) to (c).

doi:10.1371/journal.pone.0162293.g007

groups \mathbf{T}^k ($k = 1, 2$). Accordingly, we combine these indexes to build a co-cluster as follows,

$$\begin{aligned} CL_{ijk} &= \{(\mathbf{G}_1^i, \mathbf{G}_2^j, \mathbf{G}_3^k)\} \\ &= \{(\mathbf{E}^i, \mathbf{P}^j, \mathbf{T}^k) \mid i = 1, 2; j = 1, 2, 3; k = 1, 2\} \end{aligned} \quad (25)$$

To refine the findings, significant δ -CLs (Eq (19)) were extracted. CL_{121} , including 13 genes, 6 patients, and 2 time points, finally stood out with $\delta = 0.156$.

To profile this co-cluster, we observed it along each two modes and now present the heat maps in Fig 8. Fig 8a and 8b display the genexpatient matrices at the baseline time and 1 year after the IFN- β therapy. In Fig 8c through Fig 8h, the genetime matrices for the 6 involved patients are shown. Here we can detect the similar patterns among the first four patients (Fig 8c to 8f), roughly defining a constant-lateral-slice or additive-lateral-slice co-cluster. Additionally, the patientxtime matrices for the 13 genes in this co-cluster are presented in Fig 8i to 8u. Interestingly, an L-shape is revealed in the majority of these figures, and this similarly leads to the formation of an additive-horizontal-slice or multiplicative-horizontal-slice co-cluster. All these results have attested the reliability of our algorithm.

The biological significance of CL_{121} was further analyzed according to genetic pathways and GO annotations [33]. Specifically, important contribution of the 13 genes (CXCL10, EIF2AK2, IFIT1, IRF7, IRF9, ISG15, ISG20, MX1, NFKB1, OAS1, RSAD2, STAT1, TLR8) to the biological processes in GO has long been elucidated (Table 1). As reported in [48, 49], the *immune system process* (GO: 0002376) was consistently annotated, verifying the enrichment of its three sub-terms, *immune effector process*, *defense response to virus* and *response to virus* in our detected co-cluster.

Moreover, the genetic pathways of the 13 genes were analyzed (Table 2). These pathways were mostly inferred from previous studies [44, 45]. Interestingly, the enrichment of *bone remodeling pathway* is shown in our analysis, with a small p -value of 2.3E-4 in Table 2. This finding leads to sufficient biological evidences of correlating *bone remodeling* with the IFN- β treatment for sclerosis patients.

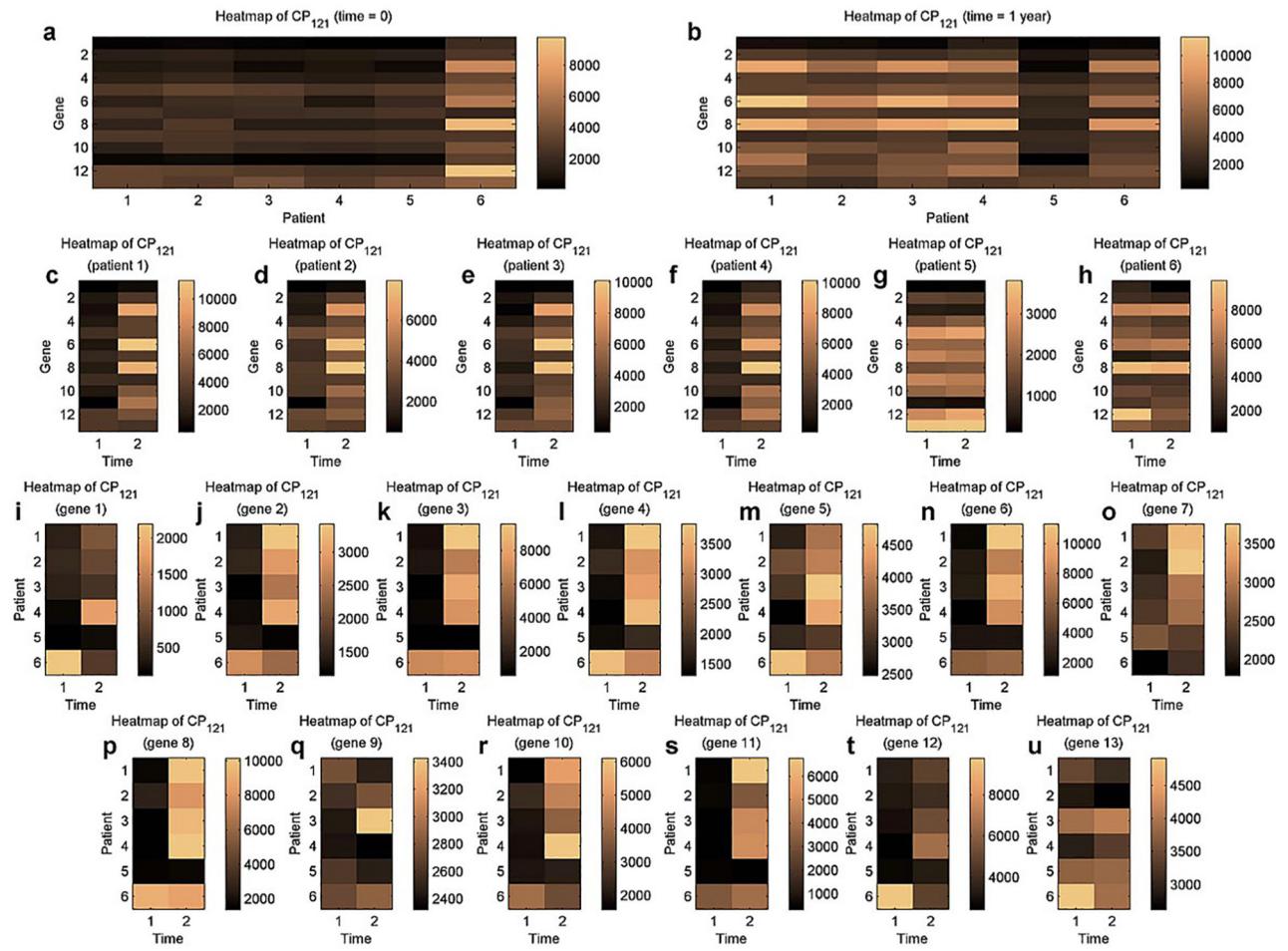


Fig 8. Heat maps (along each two modes) of the significant co-cluster CL_{121} in the genexpatientxtime tensor. (a) and (b), Heat maps of the genexpatient matrix at the baseline time and at time = 1 year. (c) to (h), Heat maps of the genextime matrices for the 6 patients. (i) to (u), Heat maps of the patientxtime matrices for the 13 genes.

doi:10.1371/journal.pone.0162293.g008

Common characteristics of the patients [48], such as a shorter disease duration, a lower EDSS score and an easier relapse, in CL_{121} were further revealed, leading to an effective profile of their disease progression. Overall, our algorithm can be beneficial to personalized therapy design and new drug discovery in the treatments of sclerosis patients.

Table 1. Biological processes regulated by the 13 annotated genes in the significant co-cluster CL_{121} (identified in the genexpatientxtime tensor).

GO term	Description	P-value	Enrichment (N, B, n, b)
GO:0002252	Immune effector process	8.81E-5	2.07 (56,25,13,12)
GO:0051607	Defense response to virus	1.17E-4	2.26 (56,21,13,11)
GO:0045069	Regulation of viral genome replication	2.12E-4	3.35 (56,9,13,7)
GO:0045071	Negative regulation of viral genome replication	2.12E-4	3.35 (56,9,13,7)
GO:0009615	Response to virus	2.77E-4	1.91 (56,27,13,12)
GO:0048525	Negative regulation of viral process	6.28E-4	3.02 (56,10,13,7)
GO:0050792	Regulation of viral process	6.28E-4	3.02 (56,10,13,7)

doi:10.1371/journal.pone.0162293.t001

Table 2. Genetic pathways of the 13 annotated genes in the significant co-cluster CL_{121} (identified in the genexpatientxtime tensor).

Pathway annotated	Term	P-value	Benjamini adjusted p-value
BIOCARTA	Bone remodelling	2.3E-4	1.0E-2
BIOCARTA	IFN- α signaling pathway	1.9E-2	3.5E-1
BIOCARTA	Double stranded RNA induced gene expression	1.9E-2	3.5E-1
BIOCARTA	Inactivation of Gsk3 by AKT causes accumulation of b-catenin in alveolar macrophages	5.9E-2	6.0E-1
BIOCARTA	Toll-like Receptor Pathway	7.5E-2	5.9E-1
KEGG PATHWAY	Toll-like receptor signaling pathway	2.1E-6	4.3E-5
KEGG PATHWAY	RLG-I-like receptor signaling pathway	5.1E-5	5.1E-4
KEGG PATHWAY	Cytosolic DNA-sensing pathway	1.7E-3	1.1E-2
KEGG PATHWAY	Chemokine signaling pathway	1.8E-2	8.8E-2
KEGG PATHWAY	Pancreatic cancer	8.2E-2	2.9E-1

doi:10.1371/journal.pone.0162293.t002

Co-clustering of Embryonic Cell Cycles in the Lineages of *C. elegans*

Recently, the techniques of live-cell imaging microscopy and fluorescent tagging have developed rapidly. These techniques have been broadly used in observing gene expression, nuclei movement and nuclei division, during the embryogenesis of a single cell [34, 38, 46]. An example of the lineage tracing of *C. elegans* can be found in Fig 9.

The length of a cell cycle in an organism is important in the tracing of an embryonic cell lineage. For different organisms and cells, the length varies significantly. Depending on well-reported protocols in [46], the cell cycle lengths of ~300 *C. elegans* embryos were evaluated by perturbing their 1219 genes in [49] (<http://phenics.icts.hkbu.edu.hk/>). For simplicity, the 8-cell stage in the AB branch was regarded as our founder-cell stage [34], and 14 descendants of each founder cell were studied. As a summary, a genexdescendantxfounder tensor of size $1219 \times 14 \times 8$ was constructed. Importantly, identification of co-clusters in this tensor can lead to the derivation of cell fates in the *C. elegans* lineage.

Similar to the preceding section, the optimal rank of (2, 5, 8) was derived for the truncated HOSVD. Linear patterns (LGA) in $\tilde{U}^{(n)}$ were separately displayed in Fig 10a to 10c, where the first two column vectors were used as representatives. Meanwhile, the first three dimensions of each $\tilde{U}^{(n)}$ (Fig 11a to 11c) and their planar patterns in the 3D space are separately displayed in Fig 11d to 11f.

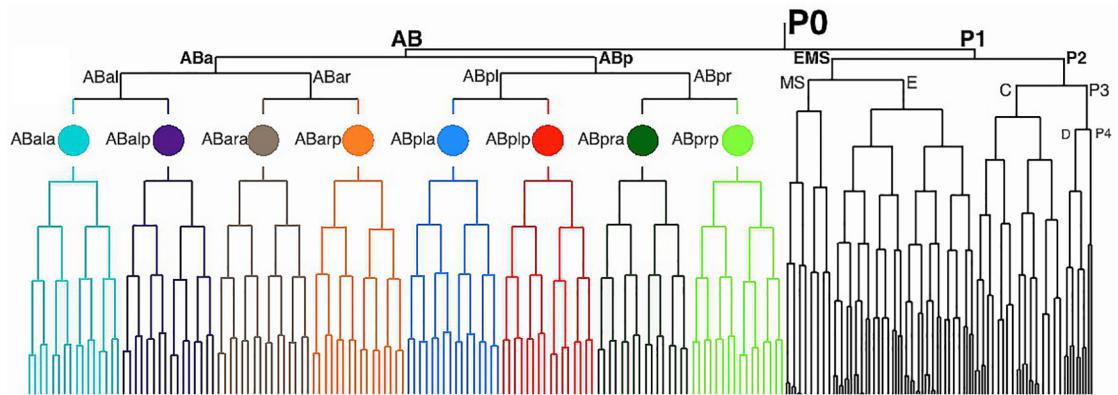


Fig 9. Lineage tracing of *C. elegans* at the 200-cell stage.

doi:10.1371/journal.pone.0162293.g009

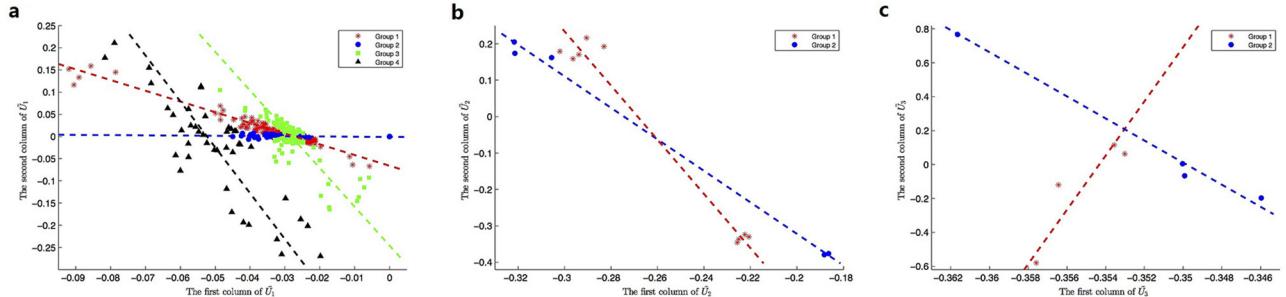


Fig 10. The linear patterns embedded in the 2D singular-vector space. (a) to (c), The linear groups along the directions of first two singular vectors of $\mathbf{U}^{(n)}$ ($n = 1, 2, 3$), respectively.

doi:10.1371/journal.pone.0162293.g010

Specifically, 4, 2 and 2 clusters in the three modes were detected and are shown in Table 3. Here, the two groups of founder cells (F) are consistently traced to their mother cells (ABal/ABpl and ABar/ABpr). Furthermore, all the cells in \mathbf{D}^2 are terminal cells, and only three terminal cells (*aaa, *paa and *pap) are distributed to \mathbf{D}^1 . Compared to ancestors, terminal cells at the same stage were reasonably grouped together, as earlier cell cycles had evolved to different lengths after multiple cell divisions.

A significant δ -CL (CL_{122}), including 42 genes (\mathbf{E}^1), 5 terminal cells (\mathbf{D}^2), and 4 founder cells (\mathbf{F}^2), was detected with $\delta = 0.0702$. The descendant×founder matrices for the corresponding 42 genes are displayed in Fig 12, as a series of heat maps. Likewise, similar patterns (two horizontal black lines) can be identified in the majority of these maps, defining a specific additive or multiplicative co-cluster type. The biological functions of these 42 annotated genes were further explored, using the GO terms and KEGG pathways [33]. Results are listed in Table 4, where the four annotated functional categories have long been emphasized in earlier studies of lineage tracing of *C. elegans* [34, 46, 49]. These functional categories may be importantly correlated to the descendant cells of ABar and ABpr branches (\mathbf{F}^2).

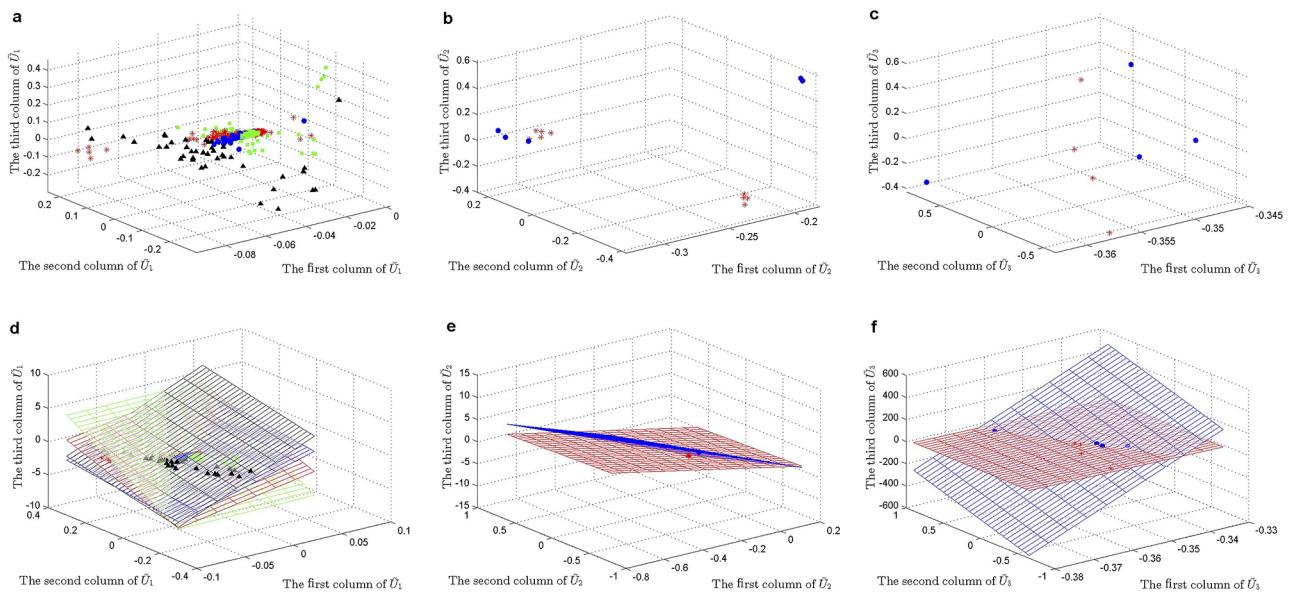


Fig 11. Linear patterns embedded in the 3D singular-vector space for the lineage tracing data of *C. elegans*. (a) to (c), The scatter plots along the directions of first three singular vectors of $\mathbf{U}^{(n)}$ ($n = 1, 2, 3$), respectively. (d) to (f), Linear or planar patterns of the 3D-points in (a) to (c).

doi:10.1371/journal.pone.0162293.g011

Table 3. The linear groups in different modes of the genexdescendantxfounder tensor (“*” denotes a founder cell).

Features modes		The corresponding linear groups
Perturbed genes (E)	E ¹	The number of genes in the group is 42
	E ²	The number of genes in the group is 729
	E ³	The number of genes in the group is 379
	E ⁴	The number of genes in the group is 69
descendant cells (D)	D ¹	“*a” “*aa” “*aaz” “*ap” “*p” “*pa” “*paa” “*pap” “*pp”
	D ²	“*aap” “*apa” “*app” “*ppa” “*ppp”
founder cells (F)	F ¹	“ABala” “ABalp” “ABpla” “ABplp”
	F ²	“ABara” “ABarp” “ABpra” “ABprp”

doi:10.1371/journal.pone.0162293.t003

Another significant δ -CL (CL_{321}) with $\delta = 0.0853$ include 379 genes (E³), 5 terminal cells (D²), and 4 founder cells (F¹). Cell-fate changes have been well demonstrated with perturbed genes such as mex-5, gsk-3, skr-2 and cdc-25.1 [34], and now they are detected in CL_{321} as well. As a great number (379) of genes were involved, multiple functional categories were annotated with



Fig 12. Heat maps. Heat maps of the descendantxfounder matrices for the 42 genes involved in the significant co-cluster CL_{122} (identified in the genexdescendantxfounder tensor).

doi:10.1371/journal.pone.0162293.g012

Table 4. Annotated functional categories revealed by the significant co-cluster CL_{122} (identified in the genexdescendantxfounder tensor).

Annotated functional categories	Term	P-value	Benjamini adjusted p-value
GO:0009792	Embryonic development ending in birth or egg hatching	8.0E-4	1.1E-1
GO:0006260	DNA replication	4.1E-3	2.6E-1
GO:0006259	DNA metabolic process	4.4E-2	8.9E-1
KEGG PATHWAY	Mismatch repair	2.3E-2	3.2E-1
KEGG PATHWAY	DNA replication	4.4E-2	3.0E-1

doi:10.1371/journal.pone.0162293.t004

smaller p -values. The top sub-terms in **GO** pathway are *embryonic development ending in birth or egg hatching* (GO: 0009792), *post-embryonic development* (GO:0009791) and *nematode larval development* (GO: 0002119), each with a p -value less than 7.0E-30. The top three sub-terms in **KEGG** pathway are *spliceosome*, *RNA degradation* and *Wnt signaling*, as revealed by CL_{321} . Notably, it successfully captures the well-acknowledged module—*Wnt signaling pathway*.

Experiment Comparisons with Other Methods Using 2D Synthetic Data and 2D Yeast Gene Expression Data

In order to test the performance and robustness of **HDSVS**, we implemented a series of comparison experiments. Matrices of 2D tensor data were used because most existing methods can process 2D data only and cannot be generalized to higher-order tensors easily, and multiple state-of-the-art biclustering methods were adopted for such comparisons. These methods include **ISA** [19], **CC** [2], **FABIA** [21], **BSGP** [18], **SMR** [29] and **BiMax** [20]. Specifically in our comparison experiments, **MTBA** (*MATLAB Toolbox for Biclustering Analysis*) was used as an algorithm-suite.

First, 2D synthetic data were generated based on principles in [47], resulting in matrices (500×200) with single-type biclusters (50×50). The background values are generated by a normal distribution $N(0, 1)$. Comprehensively, constant, constant-row/column, additive and multiplicative biclusters were considered and generated as follows:

1. constant biclusters, i.e. $\{a_{ij} = 2 \mid i \in I, j \in J\}$;
2. constant-row or constant-column biclusters, i.e. $\{\mathbf{a}_{ij}^T = \mu_i \mathbf{1}_J \mid i \in I\}$ or $\{\mathbf{a}_{ij} = \mu_j \mathbf{1}_I \mid j \in J\}$ where $\mathbf{1}_J^T$ or $\mathbf{1}_I$ is a column vector of ones, μ_i and μ_j are drawn from a normal distribution $N(0, 1)$;
3. additive-row or additive-column biclusters, i.e. $\{\mathbf{a}_{ij}^T = \mu_i \mathbf{1}_J + \mathbf{a}_{(1)j}^T \mid i, k \in I\}$ or $\{\mathbf{a}_{ij} = \mu_j \mathbf{1}_I + \mathbf{a}_{i(1)}^T \mid j, k \in J\}$ where $\mathbf{a}_{(1)j}$ ($\mathbf{a}_{i(1)}$) is the first row (column) of the biclusters, μ_i , μ_j and each value of $\mathbf{a}_{(1)j}$ ($\mathbf{a}_{i(1)}$) are drawn from a normal distribution $N(0, 1)$;
4. multiplicative-row or multiplicative-column biclusters, i.e. $\{\mathbf{a}_{ij}^T = \mu_i \mathbf{a}_{(1)j}^T \mid i, k \in I\}$ or $\{\mathbf{a}_{ij} = \mu_j \mathbf{a}_{i(1)} \mid j, k \in J\}$ where $\mathbf{a}_{(1)j}$ ($\mathbf{a}_{i(1)}$) is the first row (column) of the biclusters, μ_i , μ_j and each value of $\mathbf{a}_{(1)j}$ ($\mathbf{a}_{i(1)}$) are drawn from a normal distribution $N(0, 1)$.

For each scenario involving a specific bicluster type, the matching scores of the detected biclusters and the true ones under Gaussian white noise with different SNRs (dB) were derived for each method. These SNR-MS curves of the compared methods and our algorithm are displayed in Fig 13, where parts a, b, c and d show the case concerning constant, constant-row/column, additive and multiplicative biclusters, respectively. Depending on such SNR-MS curves, the robustness of each method can be reasonably measured. As shown in Fig 13, **HDSVS** performs well and stably under different SNRs. Especially, it outperforms others when additive biclusters are involved. This validates the robustness and generalization capability of our method.

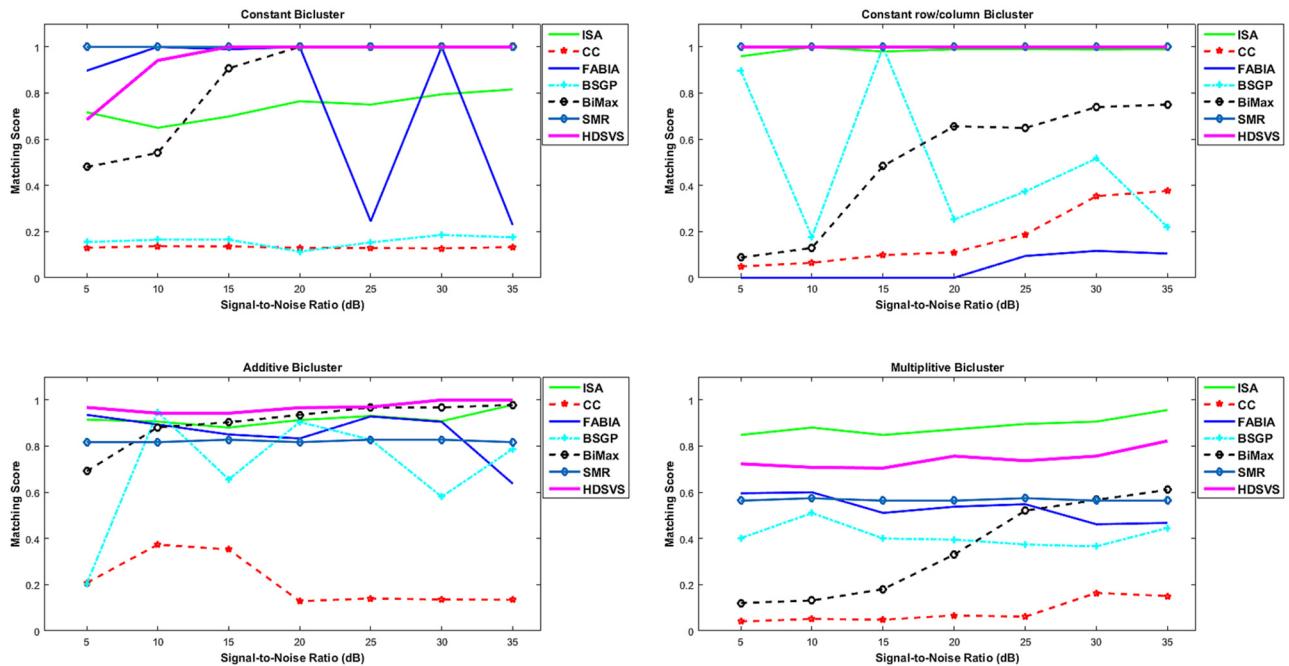


Fig 13. Robustness test and performance comparison for HDSVS and a series of other biclustering methods (ISA, CC, FABIA, BSGP, SMR and BiMax). (a) Signal-to-noise ratios (SNRs) vs. matching scores (MSs) for different biclustering methods, to search for constant biclusters. (b) SNR-MS curves for different biclustering methods, to search for constant-row/column biclusters. (c) SNR-MS curves for searching for additive biclusters. (d) SNR-MS curves for searching for multiplicative biclusters.

doi:10.1371/journal.pone.0162293.g013

To further evaluate the statistical significance of the results generated by different methods, a biological 2D tensor, namely gene expression data of yeast cells towards different stress conditions, was employed in our comparison experiments [20]. The original microarray data (<http://www.tik.ee.ethz.ch/sop/bimax/>) contains 2993 genes and 173 stress conditions, and have been normalized using mean centering. A set of Perl modules for accessing GO information and evaluationg the collective annotation of a gene group to GO terms was developed in [32], based on which the statistical significance of each annotation can be calculated. Using such modules, we carried out the GO enrichment significance test for our method and each of the comparison methods. The results from different thresholds are presented in Fig 14, where **HDSVS** and **SMR** outperforms other methods in this significance test. However, **HDSVS** (59 seconds) is about 12 times faster than **SMR** (786 seconds, when $K = 2$).

Conclusion

In this paper, we proposed a co-clustering method based on hyperplane detection in singular vector spaces (**HDSVS**), to identify co-clusters in high-order tensors. Based on linear structures of co-cluster patterns, this algorithm successfully extracted significant co-clusters (δ -CLs). Specifically, linear patterns embedded in the singular vector matrix along each mode, produced by a truncated HOSVD, were the key to co-cluster identification. These linear structures revealed by LGA showed a favorable performance in capturing the significant patterns. **HDSVS** was validated by multiple synthetic and biological tensors.

It is worth noting that, the performance of **HDSVS** was investigated with respect to different noise levels and overlapped degrees in tensors. Our method showed a robust performance on noisy tensors, due to the selection of singular vectors by the truncated HOSVD. Meanwhile, the

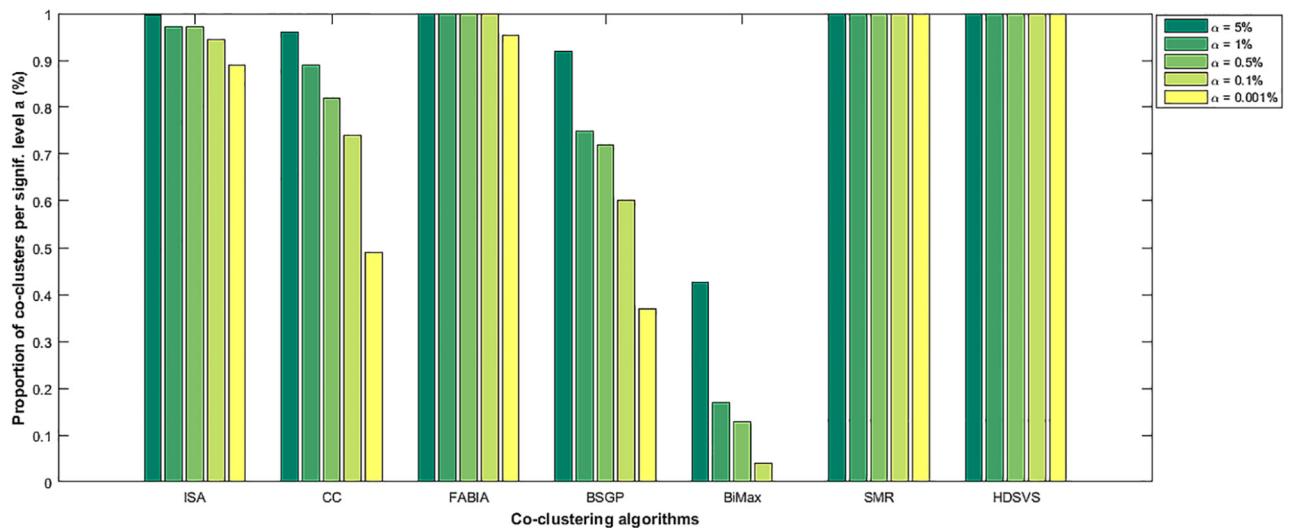


Fig 14. GO enrichment significance tests. Significance test results (under different thresholds) for **HDSVS** and existing biclustering methods (**ISA**, **CC**, **FABIA**, **BSGP**, **SMR** and **BiMax**), based on a 2D yeast gene expression tensor.

doi:10.1371/journal.pone.0162293.g014

applications of **HDSVS** to two biological tensors, namely the genexpatient \times time tensor and the genexdescendant \times founder tensor, validated its reliability in dealing with real-world applications. Especially, the genes in the detected co-clusters were significantly enriched in biologically-verified pathways and **GO** terms. In addition, comparisons between **HDSVS** and other popular methods on 2D synthetic data and 2D yeast gene expression data further showed the robustness and stability of **HDSVS**. The experiment results show that **HDSVS** is an efficient method for co-cluster identification in high-order tensors. In this paper, we have used HOSVD for tensor decomposition. We can also consider the use of several other decomposition methods. For example, the dominant multidimensional subspace in tensor data can be found using higher order orthogonal iteration of tensors (HOOI) [50]. As discussed above, PARAFAC with sparse latent factors [29] has good performance for detecting co-clusters of rank 1 with low SNR. These decomposition methods can be explored in the future to improve the performance of HDSVS.

Supporting Information

S1 Dataset. Yeast gene expression data, *C. elegans* cell cycle data and sclerosis patients gene expression data.

(MAT)

Acknowledgments

This work is supported by the Hong Kong Research Grants Council (Project CityU 11214814) and the National Natural Science Funds of China (Project No. 31100958). We would like to thank Dr. Zhongying Zhao for providing the *C. elegans* lineage data.

Author Contributions

Conceptualization: HY.

Data curation: LC.

Formal analysis: DW HY.

Funding acquisition: HY.

Investigation: HZ.

Methodology: HZ HY.

Project administration: HY.

Resources: HY.

Software: HZ DW LC XL.

Supervision: HY.

Validation: DW.

Visualization: XL.

Writing – original draft: HZ DW LC.

Writing – review & editing: LC HY.

References

1. Xu R, Wunsch D, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*. 2005; 16(3):645–678. doi: [10.1109/TNN.2005.845141](https://doi.org/10.1109/TNN.2005.845141)
2. Cheng Y, Church GM. Bioclustering of expression data. In: Ismb. vol. 8; 2000. p. 93–103.
3. Dhillon IS, Mallela S, Kumar R. A divisive information theoretic feature clustering algorithm for text classification. *The Journal of Machine Learning Research*. 2003; 3:1265–1287.
4. Lam BS, Yan H. Subdimension-based similarity measure for DNA microarray data clustering. *Physical Review E*. 2006; 74(4):041906. doi: [10.1103/PhysRevE.74.041906](https://doi.org/10.1103/PhysRevE.74.041906)
5. Van Aelst S, Wang XS, Zamar RH, Zhu R. Linear grouping using orthogonal regression. *Computational Statistics & Data Analysis*. 2006; 50(5):1287–1312. doi: [10.1016/j.csda.2004.11.011](https://doi.org/10.1016/j.csda.2004.11.011)
6. Gan X, Liew AW, Yan H. Discovering biclusters in gene expression data based on high-dimensional linear geometries. *BMC bioinformatics*. 2008; 9(1):209. doi: [10.1186/1471-2105-9-209](https://doi.org/10.1186/1471-2105-9-209) PMID: [18433477](#)
7. Zhao H, Liew AWC, Xie X, Yan H. A new geometric bioclustering algorithm based on the Hough transform for analysis of large-scale microarray data. *Journal of Theoretical Biology*. 2008; 251(2):264–274. doi: [10.1016/j.jtbi.2007.11.030](https://doi.org/10.1016/j.jtbi.2007.11.030) PMID: [18199458](#)
8. Zhao H, Chan KL, Cheng LM, Yan H. A probabilistic relaxation labeling framework for reducing the noise effect in geometric bioclustering of gene expression data. *Pattern Recognition*. 2009; 42(11):2578–2588. doi: [10.1016/j.patcog.2009.03.016](https://doi.org/10.1016/j.patcog.2009.03.016)
9. Wang DZ, Yan H. A graph spectrum based geometric bioclustering algorithm. *Journal of theoretical biology*. 2013; 317:200–211. doi: [10.1016/j.jtbi.2012.10.012](https://doi.org/10.1016/j.jtbi.2012.10.012) PMID: [23079285](#)
10. Chen HC, Zou W, Tien YJ, Chen JJ. Identification of bicluster regions in a binary matrix and its applications. *PLOS ONE*. 2013; 8(8):e71680. doi: [10.1371/journal.pone.0071680](https://doi.org/10.1371/journal.pone.0071680) PMID: [23940779](#)
11. Hartigan JA. Direct clustering of a data matrix. *Journal of the american statistical association*. 1972; 67(337):123–129. doi: [10.1080/01621459.1972.10481214](https://doi.org/10.1080/01621459.1972.10481214)
12. Madeira SC, Oliveira AL. Bioclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*. 2004; 1(1):24–45. doi: [10.1109/TCBB.2004.2](https://doi.org/10.1109/TCBB.2004.2)
13. Busygin S, Prokopyev O, Pardalos PM. Bioclustering in data mining. *Computers & Operations Research*. 2008; 35(9):2964–2987. doi: [10.1016/j.cor.2007.01.005](https://doi.org/10.1016/j.cor.2007.01.005)
14. Zhao H, Wee-Chung Liew A, Z Wang D, Yan H. Bioclustering analysis for pattern discovery: current techniques, comparative studies and applications. *Current Bioinformatics*. 2012; 7(1):43–55. doi: [10.2174/157489312799304413](https://doi.org/10.2174/157489312799304413)
15. An J, Liew AWC, Nelson CC. Seed-based bioclustering of gene expression data. *PLOS ONE*. 2012; 7(8):e42431. doi: [10.1371/journal.pone.0042431](https://doi.org/10.1371/journal.pone.0042431) PMID: [22879981](#)

16. Pontes B, Giraldez R, Aguilar-Ruiz JS. Quality measures for gene expression biclusters. PLOS ONE. 2015; 10(3):e0115497. doi: [10.1371/journal.pone.0115497](https://doi.org/10.1371/journal.pone.0115497) PMID: [25763839](#)
17. Oghabian A, Kilpinen S, Hautaniemi S, Czeizler E. Biclustering methods: biological relevance and application in gene expression analysis. PLOS ONE. 2014; 9(3):e90801. doi: [10.1371/journal.pone.0090801](https://doi.org/10.1371/journal.pone.0090801) PMID: [24651574](#)
18. Dhillon IS. Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2001. p. 269–274.
19. Bergmann S, Ihmels J, Barkai N. Iterative signature algorithm for the analysis of large-scale gene expression data. Physical review E. 2003; 67(3):031902. doi: [10.1103/PhysRevE.67.031902](https://doi.org/10.1103/PhysRevE.67.031902)
20. Prelić A, Bleuler S, Zimmermann P, Wille A, Bühlmann P, Gruissem W, et al. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics. 2006; 22(9):1122–1129. doi: [10.1093/bioinformatics/btl060](https://doi.org/10.1093/bioinformatics/btl060) PMID: [16500941](#)
21. Hochreiter S, Bodenhofer U, Heusel M, Mayr A, Mitterrecker A, Kasim A, et al. FABIA: factor analysis for bicluster acquisition. Bioinformatics. 2010; 26(12):1520–1527. doi: [10.1093/bioinformatics/btq227](https://doi.org/10.1093/bioinformatics/btq227) PMID: [20418340](#)
22. Comon P. Tensors: a brief introduction. IEEE Signal Processing Magazine. 2014; 31(3):44–53. doi: [10.1109/MSP.2014.2298533](https://doi.org/10.1109/MSP.2014.2298533)
23. Kolda TG, Bader BW. Tensor decompositions and applications. SIAM review. 2009; 51(3):455–500. doi: [10.1137/07070111X](https://doi.org/10.1137/07070111X)
24. Omberg L, Golub GH, Alter O. A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. Proceedings of the National Academy of Sciences. 2007; 104(47):18371–18376. doi: [10.1073/pnas.0709146104](https://doi.org/10.1073/pnas.0709146104)
25. Ponnappalli SP, Saunders MA, Van Loan CF, Alter O. A higher-order generalized singular value decomposition for comparison of global mRNA expression from multiple organisms. PLOS ONE. 2011; 6(12):e28072. doi: [10.1371/journal.pone.0028072](https://doi.org/10.1371/journal.pone.0028072) PMID: [22216090](#)
26. Zhao L, Zaki MJ. Tricluster: an effective algorithm for mining coherent clusters in 3d microarray data. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM; 2005. p. 694–705.
27. Banerjee A, Basu S, Merugu S. Multi-way Clustering on Relation Graphs. In: SDM. vol. 7. SIAM; 2007. p. 225–334.
28. Huang H, Ding C, Luo D, Li T. Simultaneous tensor subspace selection and clustering: the equivalence of high order svd and k-means clustering. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data mining. ACM; 2008. p. 327–335.
29. Papalexakis EE, Sidiropoulos N, Bro R. From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors. Signal Processing, IEEE Transactions on. 2013; 61(2):493–506. doi: [10.1109/TSP.2012.2225052](https://doi.org/10.1109/TSP.2012.2225052)
30. Wu T, Benson AR, Gleich DF. General Tensor Spectral Co-clustering for Higher-Order Data. arXiv preprint arXiv:160300395. 2016;.
31. García-Escudero LA, Gordaliza A, San Martin R, Van Aelst S, Zamar R. Robust linear clustering. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2009; 71(1):301–318. doi: [10.1111/j.1467-9868.2008.00682.x](https://doi.org/10.1111/j.1467-9868.2008.00682.x)
32. Boyle EI, Weng S, Gollub J, Jin H, Botstein D, Cherry JM, et al. GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. Bioinformatics. 2004; 20(18):3710–3715. doi: [10.1093/bioinformatics/bth456](https://doi.org/10.1093/bioinformatics/bth456) PMID: [15297299](#)
33. Huang DW, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nature protocols. 2009; 4(1):44–57. doi: [10.1038/nprot.2008.211](https://doi.org/10.1038/nprot.2008.211)
34. Du Z, Santella A, He F, Tiangson M, Bao Z. De novo inference of systems-level mechanistic models of development from live-imaging-based phenotype analysis. Cell. 2014; 156(1):359–372. doi: [10.1016/j.cell.2013.11.046](https://doi.org/10.1016/j.cell.2013.11.046) PMID: [24439388](#)
35. Cheng KO, Law NF, Siu WC, Liew A. Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization. BMC bioinformatics. 2008; 9(1):1. doi: [10.1186/1471-2105-9-210](https://doi.org/10.1186/1471-2105-9-210)
36. Yang WH, Dai DQ, Yan H. Finding correlated biclusters from gene expression data. Knowledge and Data Engineering, IEEE Transactions on. 2011; 23(4):568–584. doi: [10.1109/TKDE.2010.150](https://doi.org/10.1109/TKDE.2010.150)
37. Kluger Y, Basri R, Chang JT, Gerstein M. Spectral biclustering of microarray data: coclustering genes and conditions. Genome research. 2003; 13(4):703–716. doi: [10.1101/gr.648603](https://doi.org/10.1101/gr.648603) PMID: [12671006](#)

38. Pascual-Montano A, Carazo JM, Kochi K, Lehmann D, Pascual-Marqui RD. Nonsmooth nonnegative matrix factorization (nsNMF). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 2006; 28(3):403–415. doi: [10.1109/TPAMI.2006.60](https://doi.org/10.1109/TPAMI.2006.60)
39. Sill M, Kaiser S, Benner A, Kopp-Schneider A. Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics*. 2011; 27(15):2089–2097. doi: [10.1093/bioinformatics/btr322](https://doi.org/10.1093/bioinformatics/btr322) PMID: [21636597](#)
40. De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*. 2000; 21(4):1253–1278. doi: [10.1137/S0895479896305696](https://doi.org/10.1137/S0895479896305696)
41. Weiland S, Van Belzen F. Singular value decompositions and low rank approximations of tensors. *Signal Processing, IEEE Transactions on*. 2010; 58(3):1171–1182. doi: [10.1109/TSP.2009.2034308](https://doi.org/10.1109/TSP.2009.2034308)
42. Lenstra AK, Lenstra J, Kan AR, Wansbeek T. Two lines least squares. *North-Holland Mathematics Studies*. 1982; 66:201–211. doi: [10.1016/S0304-0208\(08\)72452-6](https://doi.org/10.1016/S0304-0208(08)72452-6)
43. Pison G, Van Aelst S, Zamar RH. A robust linear grouping algorithm. In: *Compstat 2006-Proceedings in Computational Statistics*. Springer; 2006. p. 43–53.
44. Hecker M, Hartmann C, Kandulski O, Paap BK, Koczan D, Thiesen HJ, et al. Interferon-beta therapy in multiple sclerosis: the short-term and long-term effects on the patients' individual gene expression in peripheral blood. *Molecular neurobiology*. 2013; 48(3):737–756. doi: [10.1007/s12035-013-8463-1](https://doi.org/10.1007/s12035-013-8463-1) PMID: [23636981](#)
45. Hundeshagen A, Hecker M, Paap BK, Angerstein C, Kandulski O, Fatum C, et al. Elevated type I interferon-like activity in a subset of multiple sclerosis patients: molecular basis and clinical relevance. *J Neuroinflammation*. 2012; 9(1):140. doi: [10.1186/1742-2094-9-140](https://doi.org/10.1186/1742-2094-9-140) PMID: [22727118](#)
46. Bao Z, Murray JL, Boyle T, Ooi SL, Sandel MJ, Waterston RH. Automated cell lineage tracing in *Caenorhabditis elegans*. *Proceedings of the National Academy of Sciences of the United States of America*. 2006; 103(8):2707–2712. doi: [10.1073/pnas.0511111103](https://doi.org/10.1073/pnas.0511111103) PMID: [16477039](#)
47. Eren K, Deveci M, Küçüktunç O, Çatalyürek ÜV. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics*. 2013; 14(3):279–292. doi: [10.1093/bib/bbs032](https://doi.org/10.1093/bib/bbs032) PMID: [22772837](#)
48. Moore JL, Du Z, Bao Z. Systematic quantification of developmental phenotypes at single-cell resolution during embryogenesis. *Development*. 2013; 140(15):3266–3274. doi: [10.1242/dev.096040](https://doi.org/10.1242/dev.096040) PMID: [23861063](#)
49. Shao J, He K, Wang H, Ho WS, Ren X, An X, et al. Collaborative regulation of development but independent control of metabolism by two epidermis-specific transcription factors in *Caenorhabditis elegans*. *Journal of Biological Chemistry*. 2013; 288(46):33411–33426. doi: [10.1074/jbc.M113.487975](https://doi.org/10.1074/jbc.M113.487975) PMID: [24097988](#)
50. De Lathauwer L, De Moor B, Vandewalle J. On the best rank-1 and rank-(r 1, r 2, ..., rn) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*. 2000; 21(4):1324–1342. doi: [10.1137/S0895479898346995](https://doi.org/10.1137/S0895479898346995)