

RESEARCH ARTICLE

Reweighted multi-view clustering with tissue-like P system

Huijian Chen, Xiyu Liu *

Business School, Shandong Normal University, Jinan, China

* xyliu@sdnu.edu.cn



OPEN ACCESS

Citation: Chen H, Liu X (2023) Reweighted multi-view clustering with tissue-like P system. PLoS ONE 18(2): e0269878. <https://doi.org/10.1371/journal.pone.0269878>

Editor: Qichun Zhang, University of Bradford, UNITED KINGDOM

Received: March 6, 2022

Accepted: May 29, 2022

Published: February 10, 2023

Copyright: © 2023 Chen, Liu. 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: All relevant data are within the paper and its [Supporting information](#) files.

Funding: This research project is supported by National Natural Science Foundation of China (61876101,61802234,61806114,62172262), Social Science Fund Project of Shandong Province, China(16BGLJ06,11CGLJ22), Natural Science Fund Project of Shandong Province, China (ZR2019QF007), Postdoctoral Project, China (2017M612339,2018M642695), Humanities and Social Sciences Youth Fund of the Ministry of Education, China(19YJCZH244), Postdoctoral

Abstract

Multi-view clustering has received substantial research because of its ability to discover heterogeneous information in the data. The weight distribution of each view of data has always been difficult problem in multi-view clustering. In order to solve this problem and improve computational efficiency at the same time, in this paper, Reweighted multi-view clustering with tissue-like P system (RMVCP) algorithm is proposed. RMVCP performs a two-step operation on data. Firstly, each similarity matrix is constructed by self-representation method, and each view is fused to obtain a unified similarity matrix and the updated similarity matrix of each view. Subsequently, the updated similarity matrix of each view obtained in the first step is taken as the input, and then the view fusion operation is carried out to obtain the final similarity matrix. At the same time, Constrained Laplacian Rank (CLR) is applied to the final matrix, so that the clustering result is directly obtained without additional clustering steps. In addition, in order to improve the computational efficiency of the RMVCP algorithm, the algorithm is embedded in the framework of the tissue-like P system, and the computational efficiency can be improved through the computational parallelism of the tissue-like P system. Finally, experiments verify that the effectiveness of the RMVCP algorithm is better than existing state-of-the-art algorithms.

1. Introduction

Membrane computing [1–5], as a branch of natural computing, aims to abstract computational models from the structure and function of biological cells and from the collaboration of cell groups such as organs and tissues. Membrane computing has been developed so far, and it mainly includes three basic computing models: cell-like P system [6], tissue-like P system [7, 8] and neuro-like P system [9, 10]. In the process of calculation, each cell acts as an independent unit, and each unit runs independently without interfering with each other [11]. The entire membrane system runs in extremely parallel mode. The tissue-like P system consists of cells and environment containing objects and rules. The movement of objects from cell to cell or cell to environment is carried out through the rules in extremely parallel execution. The tissue-like P system can be combined with other algorithms to improve the computational efficiency of the algorithm thanks to the computational parallelism of the tissue-like P system.

Special Funding Project, China(2019T120607). Xiyu Liu conceptualized, revised the first draft and made the decision to publish.

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

Clustering [12–15] is a tool of machine learning and artificial intelligence, which divides a group of data points into corresponding clusters, so that the similarity of data points in clusters is high, and lower similarity between clusters. It is an unsupervised learning technique. A great deal of single-view clustering methods have been proposed, such as spectral clustering [16–18], graph clustering [19, 20], subspace clustering [21], k-means clustering [22] and so on.

With the deep research of clustering, the combination of clustering and deep learning methods and the application of clustering have been widely studied and achieved good clustering performance. Network clustering is related to many real applications, such as social community detection [23]; and disease module identification [24]. Wang et al. [25] proposed a single-cell clustering model based on denoising autoencoder and graph convolution network.

With the development of science and technology, more and more data are represented by multiple views, which are known as multi-view data. [26]. Compared with single-view clustering, multi-view clustering [27–32] has received extensive attention due to its better clustering performance. So far, a variety of multi-view clustering methods have been proposed. Multi-view clustering methods can be roughly divided into the following categories: multi-view k-means clustering [33], multi-view spectral clustering [34], multi-view subspace clustering [28, 30, 35], multi-view graph clustering [36, 37], multi-task multi-view clustering [38], etc. Multi-view subspace clustering and multi-view graph clustering have been widely studied owing to their satisfactory clustering performance. Self-representation model has achieved commendable progress in the study of single-view subspace clustering, which regards each data point as a linear combination of data. The subspace representation matrix S , which is also regarded as the similarity matrix, can be obtained as follows:

$$\min_S \| \mathbf{X} - \mathbf{X}S \|_F^2 + \alpha \| S \|_F^2 \quad (1)$$

where \mathbf{X} is the original data matrix. Guo et al. [39] extended the single-view self-representation model to multi-view clustering, which assumes that samples from different categories are embedded in independent subspaces. Therefore, the fused multi-view self-representation feature should be a block diagonal. The noise information in the data has always been the main factor affecting the clustering performance. In order to alleviate the impact of noise information on the clustering performance and make better use of the information of each view, scholars have proposed many methods. For example, Yin et al. [40] used a more direct and intuitive block diagonal regularization to preserve the underlying structure of each view, and at the same time introduced the Cauchy loss function to deal with noise information. The underlying public structure of multi-view data can be effectively retained by the derived consistency representation matrix, and is robust to noise information and data damage. In addition, the clustering performance will also be affected by the process of fusing the similarity matrix. Kang et al. [41] proposed a new multi-view clustering model in which the fusion graph approximates the original graph of each individual view but maintains an explicit cluster structure. The existing multi-view subspace clustering method still has a problem. After getting the similarity matrix of each view and the final uniform matrix, the second operation is implemented, that is, applying additional clustering algorithms (usually spectral clustering algorithms) to the uniform matrix, which will affect the clustering performance. Zhang et al. [42] proposed a Consensus One-step Multi-view Subspace Clustering model, which can solve the defect of poor clustering performance caused by the two-step operation.

Graph-based multi-view clustering method is one of the most popular multi-view clustering methods. In this method, the similarity matrix of each view is first constructed and merged into a unified matrix, and then an additional clustering algorithm or other methods are applied to the unified matrix to acquire clustering results. The construction of the similarity matrix of

each view is a very significant step, the reason is that the quality of the similarity matrix of each view has a great impact on the final clustering performance. Many scholars have proposed some methods for constructing similarity matrix, such as k-nearest neighbor algorithm(k-NN), Clustering with Adaptive Neighbors [12], etc. The construction of similarity matrix is affected by many factors, such as noise information and outliers, similarity metrics, etc. Huang et al. [43] proposed a new model that simultaneously performs multi-view clustering tasks and learns similar relationships in the kernel space. If there are c clusters, the target optimal graph can be directly divided into precise c connected components. In addition, the model can automatically assign appropriate weights to each view without additional parameters. The allocation of weights is an important topic in machine learning. For example, Liu et al. [44] proposed a new weight initialization method. Weight allocation in multi-view clustering is also significant, and the method in this paper will focus on the weight allocation of each view.

In the above-mentioned multi-view clustering algorithms, the weight distribution of each view and the weakening of noise data have not been effectively processed. Therefore, inspired by multi-view subspace clustering and graph-based multi-view clustering, in order to more effectively assign the weight of each view, Reweighted multi-view clustering with tissue-like P system (RMVCP) algorithm is proposed in this paper. RMVCP performs two fusion operations on each view. In the first fusion process, the self-representation matrix of each view is first constructed by the self-representation method, which can also be regarded as the similarity matrix of each view. Then assign appropriate weights to each view to fuse the similarity matrix of each view into a unified matrix. This operation is an iterative operation. Finally, the updated unified matrix and the updated similarity matrix of each view are generated. In the second fusion operation, the updated similarity matrix of each view generated in the first operation is used as input, and the appropriate weights are assigned to each view again to generate the final matrix. At the same time, Constrained Laplacian Rank (CLR) [45] is applied to the final matrix to directly generate clustering results without additional clustering steps (such as K-means). In addition, in order to improve the computational efficiency of the RMVCP algorithm, the RMVCP algorithm is integrated with the tissue-like P system. Fig 1 shows the RMVCP process without tissue-like P system. Fig 2 shows the RMVCP algorithm process in the framework of the tissue-like P system.

In summary, the contributions of our work are listed as follows:

- In order to assign weights to each view more reasonably, all views will be merged twice. The two fusion operations are iterative processes, which can assign more reasonable weights to each view.
- Constrained Laplacian Rank (CLR) are imposed on the unified matrix after the second fusion. Therefore, the clustering results can be directly output without applying additional clustering algorithms, avoiding the suboptimal solution of the existing two-step method.
- The RMVCP algorithm is integrated with the tissue-like P system, and use the computational parallelism of the tissue-like P system to improve the computational efficiency of the algorithm.
- The RMVCP algorithm integrates multiple processes into one framework. Experiments on several datasets prove that the clustering performance of our algorithm is better than other state-of-the-art algorithms.

The rest of this paper is organized as follows. The related research on multi-view clustering and the basic definition of the tissue-like P system are introduced in Section 2; In Section 3, RMVCP method is proposed; Comparative experiments were conducted to verify the

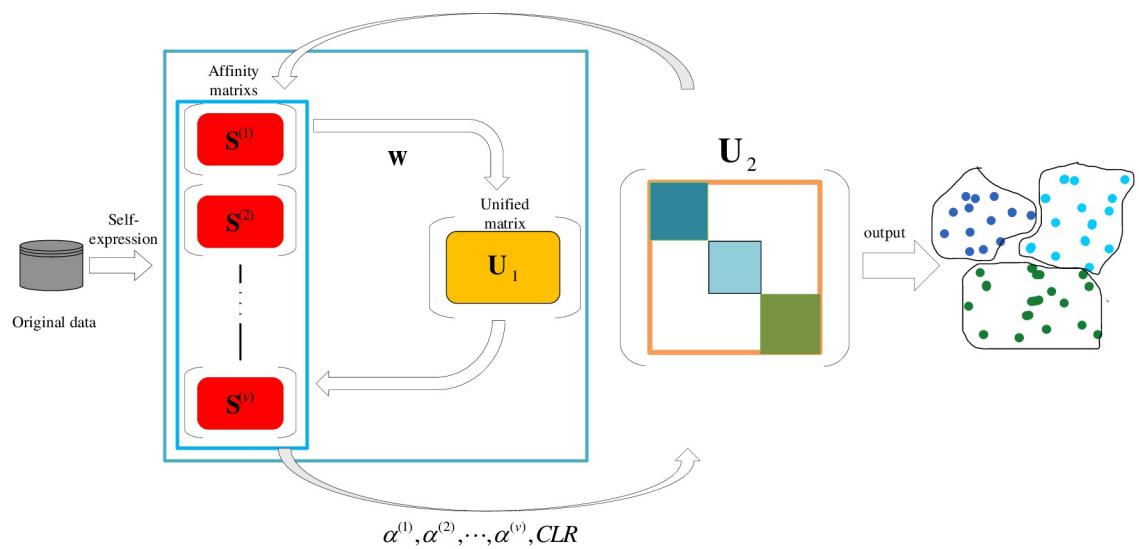


Fig 1. The RMVCP process without tissue-like P system.

<https://doi.org/10.1371/journal.pone.0269878.g001>

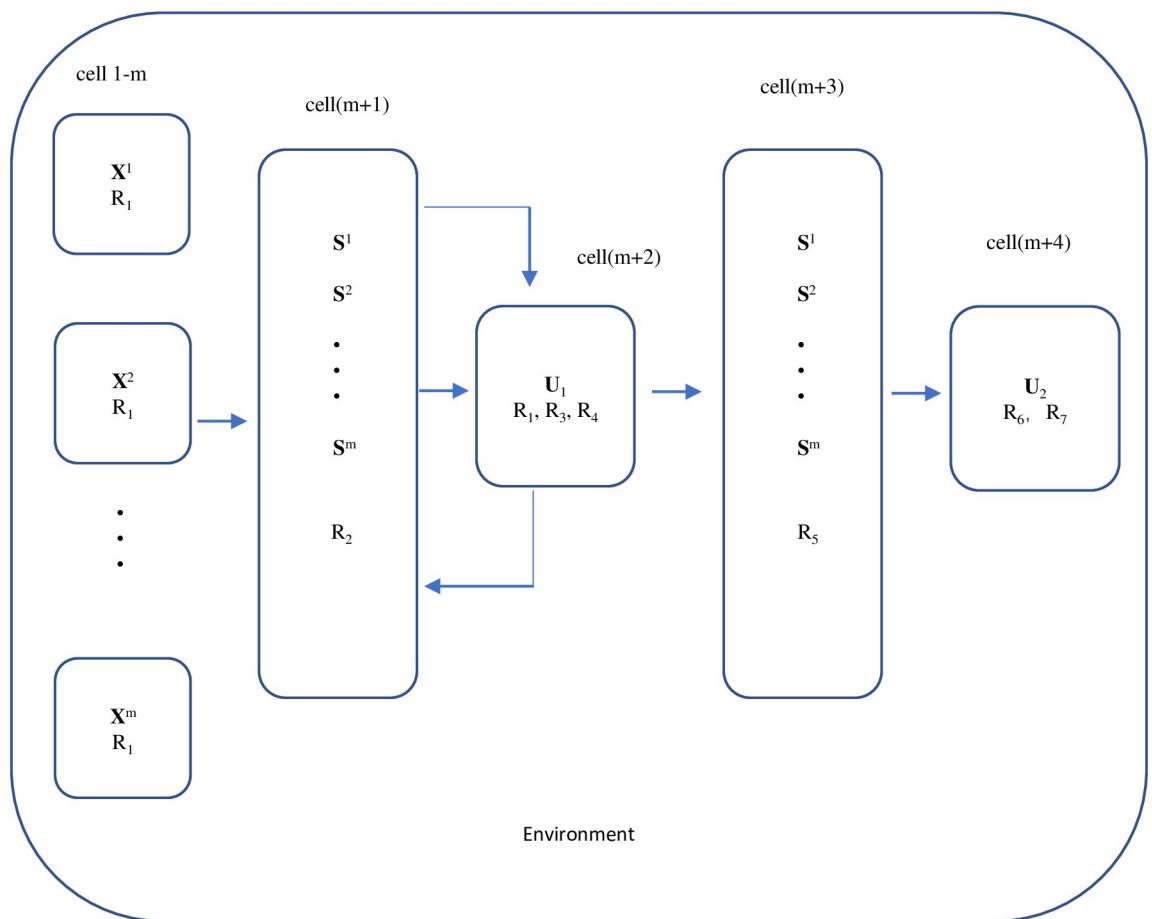


Fig 2. The RMVCP algorithm process in the framework of the tissue-like P system.

<https://doi.org/10.1371/journal.pone.0269878.g002>

effectiveness of the RMVCP algorithm in Section 4; At the end of the paper, we conclude in Section 5 and point out what we can do in the future.

2. Related work

2.1 Multi-view clustering

Currently, the most researched multi-view clustering methods are multi-view subspace clustering and graph-based multi-view clustering. Both multi-view subspace clustering and graph-based multi-view clustering have good clustering performance. Our RMVCP algorithm is also inspired by these two clustering methods. Multi-view subspace clustering uses multiple low-dimensional subspaces to represent high-dimensional data. Wang et al. [46] proposed Exclusivity-Consistency Regularity Multi-view Subspace Clustering (ECMSC). Many methods focus on the fusion of multiple views, without considering the direct consistency and difference information of the views. ECMSC considers a kind of exclusive information between views, so as to achieve information complementarity, which is helpful to improve the clustering performance. With the study of the potential representation of the data, Zhang et al. [47] proposed Latent Multi-view Subspace Clustering (LMSC). LMSC explores a latent representation of multi-view data, and then constructs a subspace representation from the latent representation. Zhang et al. integrated these two processes into an algorithm framework, while also reducing the impact of noise. High-dimensional data has always been a challenge for multi-view clustering. In order to cluster high-dimensional data more effectively, Wang et al. [48] proposed Multi-view Subspace Clustering with Intactness-Aware Similarity (MSC_IAS). MSC_IAS reduces the data dimension while preserving the data information, integrates it into a complete space, and constructs the similarity matrix. Then apply a clustering algorithm to the similarity matrix. This method can effectively process high-dimensional data. In order to more efficiently use the information across multiple views, Kang et al. [41] proposed Multi-graph Fusion for Multi-view Spectral Clustering (GFSC). GFSC can explore heterogeneous information between views, construct a similarity matrix with a self-representation method, and perform views fusion and spectral clustering at the same time. The noise information in the data greatly affects the clustering performance. In order to be able to reduce the noise information, Zhang et al. [42] proposed Consensus One-step Multi-view Subspace Clustering (COMVSC), COMVSC optimally integrates discriminative partition-level information, which can effectively reduce the impact of noise information. These state-of-the-art algorithms show good clustering performance, but the common defect is that only one fusion operation is performed on each view.

The graph-based multi-view clustering method first constructs the similarity matrix of each view, then merges each view into a unified matrix, and finally applies additional clustering algorithms or other methods to the unified matrix to obtain the clustering results. The construction of the similarity graph of each view is a very important step. Many scholars have proposed some methods for constructing similarity graphs, such as k-nearest neighbor algorithm (k-NN), Clustering with Adaptive Neighbors (CAN) [12], etc. On the other hand, the method of fusing each similarity graph is also very important. But similarly, the existing multi-view graph clustering method only merges each view once, so it does not achieve good clustering performance. Nie et al. [49] proposed the Parameter-Free Auto-Weighted Multiple Graph Learning (AMGL). AMGL solves the problem of multiple parameters in the fusion process, and automatically assigns the weight of each view on the basis of modifying the traditional spectral clustering method. Graph-based multi-view clustering methods need to apply additional clustering algorithms to obtain clustering results, and the two-step operation will affect the clustering performance. Nie et al. [50] proposed Self-weighted Multiview Clustering

(SwMC). SwMC automatically assigns weights to each view without prior knowledge. At the same time, the clustering results are directly obtained without additional clustering algorithms. In addition, the quality of the similarity graph is affected by noisy data, which in turn affects the clustering results. Nie et al. [51] proposed Multi-View Clustering and Semi-Supervised Classification with Adaptive Neighbours (MLAN). MLAN obtains the final graph for clustering by learning the local manifold structure to alleviate the noise problem. Wang et al. [37] proposed GMC: Graph-Based Multi-View Clustering (GMC). GMC jointly builds multiple view graphs and fusion graph, and automatically assign weights to each view. Obviously, these state-of-the-art multi-view graph clustering algorithms only perform one fusion operation.

2.2 Tissue-like P system

The tissue-like P system is similar to a graph structure. In the tissue-like P system, each cell and environment are equivalent to the nodes of the graph, and the communication channels between cell to cell and cell to environment are equivalent to the edge of the graph. The calculation process of the tissue-like P system is to perform calculation operations in cells through rules, and then apply certain rules to transfer objects between cells and cells or between cells and the environment through communication channels. The basic definition of the tissue-like P system is as follows:

$$\Pi = (O, K, \omega_1, \dots, \omega_m, E, ch, (s_{(i,j)})_{(i,j) \in ch}, (R_{(i,j)})_{(i,j) \in ch}, i_0) \quad (2)$$

- (1) O represents a finite multiset of objects;
- (2) K represents the states of the alphabet;
- (3) $\omega_i, 1 \leq i \leq m$ represents the finite multiset of objects in the initial state of cells $1, \dots, m$;
- (4) $E \subseteq O$ represents a copy of any number of symbolic objects in the environment;
- (5) $ch \subseteq \{(i, j) | i, j \in \{0, 1, \dots, m\}, i \neq j\}$ represents the communication channel between cells and cells and between cells and the environment;
- (6) $s_{(i,j)}$ is the initial state of the channel (i, j) ;
- (7) $R_{(i,j)}$ is a finite co/inverse transportation rule of the form $(s, x/y, s')$, where $s, s' \in K, x, y \in O^*$;
- (8) $i_0 \in \{1, \dots, m\}$ is the output cell.

3. Reweighted multi-view clustering with tissue-like P system (RMVCP)

Different from exploring the local information of data, the exploration of global information of data can better grasp the relationship between data points, which is the motivation for us to use the self-representation method to construct the similarity matrix of each view. Moreover, the quality of each view is uneven, and it is undesirable to treat each view equally. This prompted us to assign weight to each view. Nevertheless, twice weighting each view is necessary to improve clustering accuracy. In addition, the improved multi-view clustering algorithm is combined with tissue-like P system to improve the computational efficiency of the algorithm, which is due to the parallel computing ability of tissue-like P system. Therefore, we propose Reweighted multi-view clustering with tissue-like P system (RMVCP) in this paper, where RMVCP-1 refers to the first weight allocation process and RMVCP-2 is the second weight allocation for each view.

3.1 The first fusion process of the RMVCP model (RMVCP-1)

In the RMVCP-1 process, the similarity matrix of each view is constructed by a self-representation method [41]. Self-representation method treats each data point as a linear combination of the data itself. Given data matrix $\mathbf{X} \in R^{d \times n}$. The similarity matrix \mathbf{S} can be obtained by solving:

$$\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{X}\mathbf{S}\|_F^2 + \alpha \|\mathbf{S}\|_F^2 \quad s.t. \quad \mathbf{S} \geq 0 \quad (2)$$

where α is a trade-off parameter. Then we extend it to multi-view clustering:

$$\min_{\mathbf{S}^v} \sum_{v=1}^m \|\mathbf{X}^v - \mathbf{X}^v \mathbf{S}^v\|_F^2 + \alpha \|\mathbf{S}^v\|_F^2 \quad s.t. \quad \mathbf{S}^v \geq 0 \quad (3)$$

where m is the number of the views. The similarity matrix of each view obtained by this formula reflects different aspects of the original data. On the basis of this formula, the construction of the unified matrix \mathbf{U}_1 is as follows:

$$\mathbf{U}_1 = \frac{\sum_{v=1}^m \mathbf{S}^v}{m} \quad (4)$$

Obviously, the construction of the unified matrix is simply adding the similarity matrix of each view and dividing by the number of views without considering the weight of each view. This will lead to poor clustering performance. Therefore, we calculate the weight of each view in the process of graph fusion. The formula is as follows:

$$\min_{\mathbf{U}_1} \sum_{v=1}^m w_v \|\mathbf{S}^v - \mathbf{U}_1\|_F^2 \quad (5)$$

where w_v is the weight of each view. In this way, the unified matrix can better reflect the characteristics of the data with good views are given large weights and bad views are given small weights. The expression of weight is:

$$w_v = \frac{1}{2 \|\mathbf{S}^v - \mathbf{U}_1\|_F} \quad (6)$$

Then the goal formula is proposed by combining Eqs 3 and 5:

$$\begin{aligned} \min_{\mathbf{S}^v, \mathbf{U}_1, \mathbf{F}} & \sum_{v=1}^m \|\mathbf{X}^v - \mathbf{X}^v \mathbf{S}^v\|_F^2 + \alpha \|\mathbf{S}^v\|_F^2 + \beta w_v \|\mathbf{S}^v - \mathbf{U}_1\|_F^2 \\ & s.t. \quad \mathbf{S}^v \geq 0 \end{aligned} \quad (7)$$

By solving Eq 7, we can learn the similarity matrix of each view and the final unified matrix after weighting by an iterative algorithm. Finally, the unified matrix is fed to the spectral clustering algorithm.

For clustering, an ideal situation is that the number of connected components of the similarity matrix is equal to the number of clusters. When this situation is met, that is, the number of connected components of the similarity matrix is equal to k , the data point can be exactly divided into k clusters. So here we introduce Theorem 1 [52, 53].

Theorem 1. *The multiplicity of the eigenvalue 0 of the Laplacian matrix of the similarity matrix is equal to the number of connected components in the graph of the similarity matrix.*

According to Theorem 1, we know that when the number of eigenvalues 0 of the similarity matrix is k , the number of connected components is exactly k . From the Ky Fan's Theory [54], we get the final expression of Eq 7 as follows(The specific process is shown in the RMVCP-2 process):

$$\min_{\mathbf{S}^v, \mathbf{U}_1, \mathbf{F}} \sum_{v=1}^m \|\mathbf{X}^v - \mathbf{X}^v \mathbf{S}^v\|_F^2 + \alpha \|\mathbf{S}^v\|_F^2 + \beta w_v \|\mathbf{S}^v - \mathbf{U}_1\|_F^2 + \gamma \text{Tr}(\mathbf{F}_{\mathbf{U}_1}^T \mathbf{L}_{\mathbf{U}_1} \mathbf{F}_{\mathbf{U}_1}) \quad (8)$$

$$\text{s.t. } \mathbf{S}^v \geq 0, \mathbf{F}_{\mathbf{U}_1}^T \mathbf{F}_{\mathbf{U}_1} = \mathbf{I}$$

where $\mathbf{F}_{\mathbf{U}_1}$ is the spectral embedding matrix, $\mathbf{L}_{\mathbf{U}_1}$ is the Laplacian matrix of the unified matrix, and α, β, γ are regularization parameters.

Next, we optimize Eq 8:

We optimize each variable through an iterative method.

Updating \mathbf{S}_v when $\mathbf{F}_{\mathbf{U}_1}$ and \mathbf{U}_1 are fixed. So Eq 7 becomes:

$$\min_{\mathbf{S}^v} \sum_{v=1}^m \|\mathbf{X}^v - \mathbf{X}^v \mathbf{S}^v\|_F^2 + \alpha \|\mathbf{S}^v\|_F^2 + \beta w_v \|\mathbf{S}^v - \mathbf{U}_1\|_F^2 \quad (9)$$

It can be seen from Eq 9 that each view is independent. So, we only consider one view at a time. In order to update \mathbf{S}_v , we perform the derivative operation on Eq 9 to obtain:

$$(\text{Eq 9})^* = -2(\mathbf{X}^v)^T (\mathbf{X}^v - \mathbf{X}^v \mathbf{S}^v) + 2\alpha \mathbf{S}^v + 2\beta w_v (\mathbf{S}^v - \mathbf{U}_1) \quad (10)$$

Then we make Eq 10 equal to zero and we get:

$$\mathbf{S}^v = ((\mathbf{X}^v)^T \mathbf{X}^v + \alpha \mathbf{I} + \beta w_v \mathbf{I})^{-1} (\beta w_v \mathbf{U}_1 + (\mathbf{X}^v)^T \mathbf{X}^v) \quad (11)$$

where \mathbf{I} is the identity matrix.

Updating \mathbf{U}_1 when $\mathbf{F}_{\mathbf{U}_1}$ and \mathbf{S}_v are fixed. We obtain:

$$\min_{\mathbf{S}} \sum_{v=1}^m \beta w_v \|\mathbf{S}^v - \mathbf{U}_1\| + \gamma \text{Tr}(\mathbf{F}_{\mathbf{U}_1}^T \mathbf{L}_{\mathbf{U}_1} \mathbf{F}_{\mathbf{U}_1}) \quad (12)$$

After deriving [41], we can obtain:

$$\mathbf{U}_1(:, i) = \frac{\sum_v w_v \mathbf{S}^v(:, i) - \frac{\gamma \mathbf{q}_i}{4\beta}}{\sum_v w_v} \quad (13)$$

where $\mathbf{q}_i \in R^{n \times 1}$ with the j -th entry $q_{ij} = \|\mathbf{F}_{\mathbf{U}_1 i,:} - \mathbf{F}_{\mathbf{U}_1 j,:}\|^2$.

Updating $\mathbf{F}_{\mathbf{U}_1}$ when \mathbf{S}_v and \mathbf{U}_1 are fixed. We need to solve the following problems:

$$\min_{\mathbf{F}_{\mathbf{U}_1}} \text{Tr}(\mathbf{F}_{\mathbf{U}_1}^T \mathbf{L}_{\mathbf{U}_1} \mathbf{F}_{\mathbf{U}_1}) \quad \text{s.t. } \mathbf{F}_{\mathbf{U}_1}^T \mathbf{F}_{\mathbf{U}_1} = \mathbf{I} \quad (14)$$

$\mathbf{F}_{\mathbf{U}_1}$ is composed of the eigenvectors corresponding to the first k smallest eigenvalues of the Laplacian matrix. We terminate algorithm 1 when the number of iterations is greater than 200 or the relative change of \mathbf{U}_1 is less than 0.001. So far, we have calculated the similarity matrix of each view that will be input to the next algorithm after the first iteration. The RMVCP-1 process is summarized in Algorithm 1.

Algorithm 1 RMVCP-1

Input: Data matrices: $\mathbf{X}^1, \dots, \mathbf{X}^m$, parameters $\alpha > 0, \beta > 0, \gamma > 0$.

output: Similarity matrices: $\mathbf{S}^1, \dots, \mathbf{S}^m$, Unified matrix $\mathbf{U}_1, \mathbf{F}_{\mathbf{U}_1}$.

```

Initialize: Random matrices  $\mathbf{U}_1$  and  $\mathbf{F}_{\mathbf{U}_1}$ ,  $w_v = 1/m$ .
repeat
  1: Update  $\mathbf{S}_v$  by Eq 11 for each view.
  2: For each element  $S_{ij}^v = \max(S_{ij}^v, 0)$ .
  3: Update  $\mathbf{U}_1$  by Eq 13.
  4: Update  $\mathbf{F}_{\mathbf{U}_1}$  by Eq 14.
  5: Update  $w_v$  by Eq 6.
until stopping criterion is met.

```

3.2 The second fusion process of the RMVCP model (RMVCP-2)

RMVCP-1 continuously updates $\mathbf{S}^1, \dots, \mathbf{S}^m$ in an iterative manner until the algorithm converges. We perform another fusion process again, and constantly update $\mathbf{S}^1, \dots, \mathbf{S}^m$ so that the similarity matrix $\mathbf{S}^1, \dots, \mathbf{S}^m$ can better represent the characteristics of each view. We use the updated $\mathbf{S}^1, \dots, \mathbf{S}^m$ obtained from RMVCP-1 as the input of RMVCP-2. The objective function [50] is:

$$\min_{u_{2ij} \geq 0, u_{2i} \mathbf{1}_n = 1} \sum_{v=1}^m \alpha^{(v)} \|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F^2 \quad (15)$$

where \mathbf{U}_2 is the unified matrix, and $\alpha^{(v)}$ is the weight of the v -th view:

$$\alpha^{(v)} = \frac{1}{2 \|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F} \quad (16)$$

In this case, the unified matrix \mathbf{U}_2 is obtained after fusion, and additional clustering algorithms are applied to the unified matrix \mathbf{U}_2 , which will affect the final clustering performance. Here we introduce the Constrained Laplacian Rank (CLR) method to avoid additional clustering algorithms and directly output the clustering results.

It can be seen from Theorem 1 that when the rank of the Laplacian matrix of the unified matrix \mathbf{U}_2 is $\mathbf{L}_{\mathbf{U}_2} = n - c$, where c is the multiplicity of the eigenvalue 0 of $\mathbf{L}_{\mathbf{U}_2}$, the data points can be directly divided into c clusters. So, [Eq 15](#) becomes:

$$\min_{u_{2ij} \geq 0, u_{2i} \mathbf{1}_n = 1, \text{rank}(\mathbf{L}_{\mathbf{U}_2}) = n - c} \sum_{v=1}^m \alpha^{(v)} \|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F^2 \quad (17)$$

It is very difficult to solve [Eq 17](#). Let $\xi_i(\mathbf{L}_{\mathbf{U}_2})$ denote the i -th smallest eigenvalue of $\mathbf{L}_{\mathbf{U}_2}$. $\xi_i(\mathbf{L}_{\mathbf{U}_2}) \geq 0$ since $\mathbf{L}_{\mathbf{U}_2}$ is positive semi-definite. Then, $\text{rank}(\mathbf{L}_{\mathbf{U}_2}) = n - c$ can be achieved if $\sum_{i=1}^c \xi_i(\mathbf{L}_{\mathbf{U}_2}) = 0$. From the Ky Fan's Theory, we know:

$$\sum_{i=1}^c \xi_i(\mathbf{L}_{\mathbf{U}_2}) = \min_{\mathbf{F} \in R^{n \times c}, \mathbf{F}_{\mathbf{U}_2}^T \mathbf{F}_{\mathbf{U}_2} = \mathbf{I}} \text{Tr}(\mathbf{F}_{\mathbf{U}_2}^T \mathbf{L}_{\mathbf{U}_2} \mathbf{F}_{\mathbf{U}_2}) \quad (18)$$

So, [Eq 17](#) becomes:

$$\min_{\mathbf{U}_2, \mathbf{F}_{\mathbf{U}_2}} \sum_{v=1}^m \alpha^{(v)} \|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F^2 + 2\lambda \text{Tr}(\mathbf{F}_{\mathbf{U}_2}^T \mathbf{L}_{\mathbf{U}_2} \mathbf{F}_{\mathbf{U}_2}) \quad (19)$$

$$\text{s.t. } u_{2ij} \geq 0, u_i \mathbf{1}_n = 1, \mathbf{F}_{\mathbf{U}_2} \in R^{n \times c}, \mathbf{F}_{\mathbf{U}_2}^T \mathbf{F}_{\mathbf{U}_2} = \mathbf{I}$$

The optimization of this formula is as follows:

Updating $\mathbf{F}_{\mathbf{U}_2}$ when \mathbf{U}_2 , $\alpha^{(v)}$ is fixed. Eq 18 becomes:

$$\min_{\mathbf{F}_{\mathbf{U}_2} \in R^{n \times c}, \mathbf{F}_{\mathbf{U}_2}^T \mathbf{F}_{\mathbf{U}_2} = \mathbf{I}} \text{Tr}(\mathbf{F}_{\mathbf{U}_2}^T \mathbf{L}_{\mathbf{U}_2} \mathbf{F}_{\mathbf{U}_2}) \quad (20)$$

$\mathbf{F}_{\mathbf{U}_2}$ is formed by the c eigenvectors corresponding to the first c smallest eigenvalues of $\mathbf{L}_{\mathbf{U}_2}$.

Updating \mathbf{U}_2 when $\mathbf{F}_{\mathbf{U}_2}$, $\alpha^{(v)}$ is fixed. Eq 19 becomes:

$$\min_{u_{2ij} \geq 0, u_{2i} \mathbf{1}_n = 1} \sum_{v=1}^m \sum_{i,j=1}^n (\alpha^{(v)} u_{2ij} - s_{ij}^{(v)})^2 + \lambda \sum_{i,j=1}^n \|\mathbf{f}_{\mathbf{U}_2 i} - \mathbf{f}_{\mathbf{U}_2 j}\|_2^2 u_{2ij} \quad (21)$$

It is obvious that Eq 21 is independent for each i , so we consider each i separately:

$$\min_{u_{2ij} \geq 0, u_{2i} \mathbf{1}_n = 1} \sum_{j=1}^n \sum_{v=1}^m \alpha^{(v)} (u_{2ij} - s_{ij}^{(v)})^2 + \lambda \sum_{j=1}^n \|\mathbf{f}_{\mathbf{U}_2 i} - \mathbf{f}_{\mathbf{U}_2 j}\|_2^2 u_{2ij} \quad (22)$$

We denote $r_{ij} = \|\mathbf{f}_{\mathbf{U}_2 i} - \mathbf{f}_{\mathbf{U}_2 j}\|_2$ for the purpose of avoiding Eq 22 that are too complicated. So, Eq 22 can be written in vector form as:

$$\min_{u_{2i} \geq 0, u_{2i} \mathbf{1}_n = 1} \|\mathbf{u}_{2i} - \left(\sum_{v=1}^m \alpha^{(v)} \mathbf{s}_i^{(v)} - \frac{\lambda}{2} \mathbf{r}_i \right) / \sum_{v=1}^m \alpha^{(v)}\|_2^2 \quad (23)$$

The problem can be solved by an iterative algorithm. Algorithm 1 shows the process of the second fusion.

Algorithm 2 RMVCP-2

Input: $\mathbf{S}^1, \dots, \mathbf{S}^m \in R^{n \times n}$ obtained by algorithm 1, number of clusters c .
Output: $\mathbf{U}_2 \in R^{n \times n}$ with c connected components.

Initialize: the weight for each view $\alpha^{(v)} = \frac{1}{m}$, $\mathbf{F}_{\mathbf{U}_2}$ is composed of the eigenvectors corresponding to the first c smallest eigenvalues of $\mathbf{L}_{\mathbf{U}_2} = \mathbf{D}_{\mathbf{U}_2} - \frac{\mathbf{U}_2^T + \mathbf{U}_2}{2}$.

repeat

repeat

1: Calculate and update \mathbf{U}_2 by Eq 23.

2: Update $\mathbf{F}_{\mathbf{U}_2}$ by Eq 20

until converge Update $\alpha^{(v)}$ by Eq 16.

until converge

RMVCP takes the similarity matrix $\mathbf{S}^1, \dots, \mathbf{S}^m$ of each view generated by the iterative update process of Algorithm 1 as input into Algorithm 1, and can directly output the clustering results.

3.3 Initial configuration of the tissue-like P system

In this paper, in order to improve the computational efficiency of the RMVCP algorithm, we combine RMVCP with the tissue-like P system. We first set up the initial configuration of the tissue-like P system in this paper.

- cell i , $1 \leq i \leq m$: Multiset of objects $\omega_i = \mathbf{X}^i, w_i, \mathbf{F}_{\mathbf{U}_1}, \mathbf{U}_1$;
 R_1 : Rule R_1 uses Eq 11 to generate \mathbf{S}^v , $1 \leq v \leq m$ and send it to the cell $(m+1)$;
 R_{10} : Rule R_{10} sends copies of α, β, I in the environment to cell $m-1$.
- cell $(m+1)$: Multiset of objects $\omega_{m+1} = w_1, \dots, w_m, \mathbf{F}_{\mathbf{U}_1}$;
 R_2 : Rule R_2 uses Eq 13 to generate the updated \mathbf{U}_1 and send it to the cell $(m+2)$;
 R_{20} : Rule R_{20} sends copies of β in the environment to the cell $(m+1)$.

- cell (m+2): Multiset of objects $\omega_{m+2} = X^1, \dots, X^m, w_1, \dots, w_m$;
 R_1 : Rule R_1 uses Eq 11 to generate the updated S^v , $1 \leq v \leq m$ and send it to the cell (m+1);
 R_{10} : Rule R_{10} sends copies of α, β, I in the environment to cell m-1.
 R_3 : Rule R_3 uses Eq 6 to generate the weight w_1, \dots, w_m for each view;
 R_4 : Rule R_4 uses Eq 14 to generate the updated object F_{U_1} ;
 R_{40} : Rule R_{40} sends the S^v , $1 \leq v \leq m$ in the cell (m+2) to the cell (m+3). Rule R_{40} can only be triggered when the relative change of U_1 is less than 0.001.
- cell (m+3): $\omega_{m+3} = \alpha^1, \dots, \alpha^m, F_{U_2}$;
 R_5 : Rule R_5 uses Eq 26 to generate U_2 and send it to the cell (m+4). At the same time, rule R_5 calculates the relative change of Eq 19.
- cell (m+4): R_6 : Rule R_6 uses Eq 20 to generate the updated object F_{U_2} ;
 R_7 : Rule R_7 uses Eq 16 to generate updated object $\alpha^1, \dots, \alpha^m$;
- Environment: $E = \alpha, \beta, \gamma, I$.

Fig 3 shows the initial configuration of the tissue-like P system.

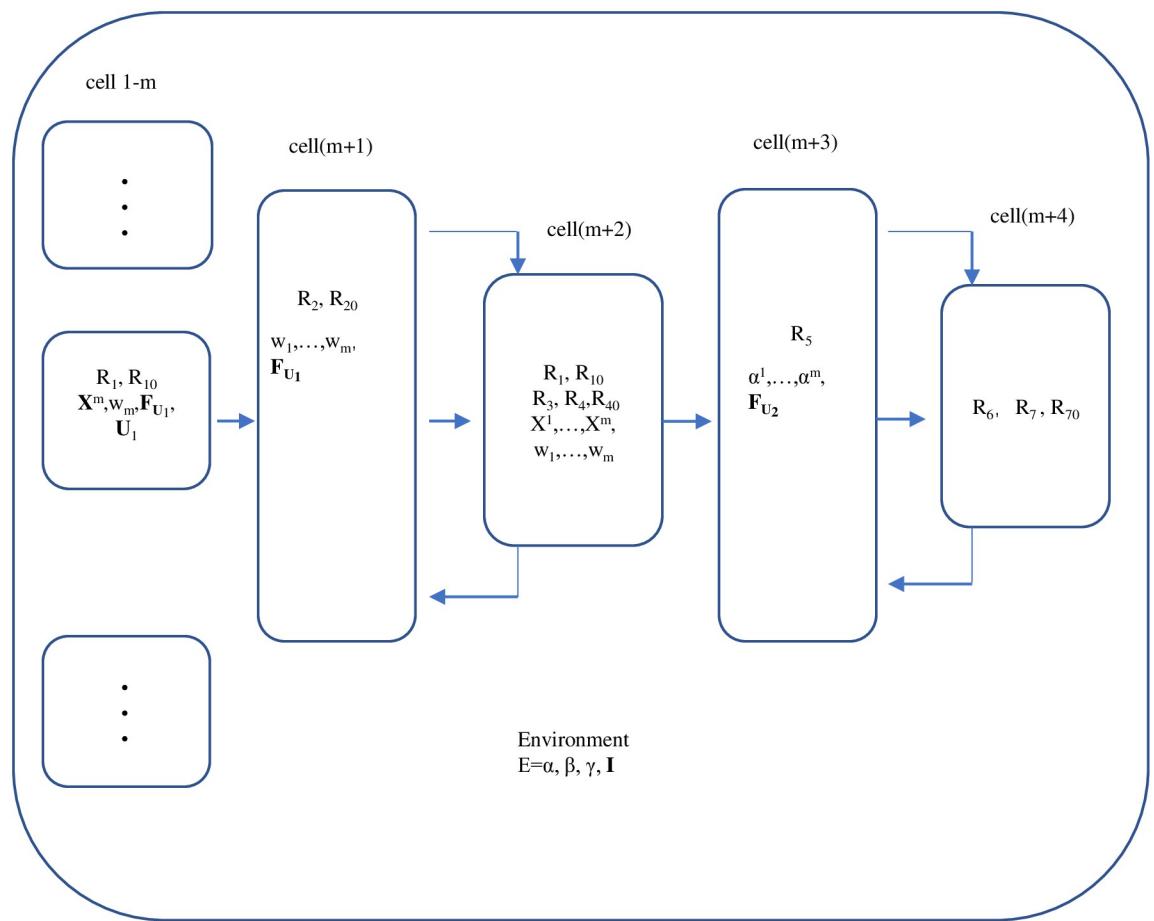


Fig 3. The initial configuration of the tissue-like P system.

<https://doi.org/10.1371/journal.pone.0269878.g003>

3.4 Computational process

- Step 1: In cells 1 to m, we first simultaneously apply the rule R_{10} to transfer the copies of $\alpha, \beta, \mathbf{I}$ in the environment to the membrane, and then apply the rule R_1 to generate $\mathbf{S}^v, 1 \leq v \leq m$ and send it to the cell (m+1).
- Step 2: In cell (m+1), the rule R_{20} is applied to transfer the copy of β in the environment to the membrane, and then the rule R_2 is applied to generate \mathbf{U}_1 and send it to cell (m+2).
- Step 3: In this step, first apply the rule R_3 to generate the updated weight w_1, \dots, w_m , and then the rule R_4 is applied to generate the updated $\mathbf{F}_{\mathbf{U}_1}$. Next the rule R_{10} is applied to transfer the copies of $\alpha, \beta, \mathbf{I}$ in the environment to the cell (m+2), and then apply the rule R_1 to produce the updated $\mathbf{S}^v, 1 \leq v \leq m$ and send it to the cell (m+1). Then apply the rules in the cell (m+1) again.

Steps 2 to 3 are a cyclic process.

- Step 4: When the conditions of triggering rule R_{40} are met, rule R_{40} is triggered, and the updated $\mathbf{S}^v, 1 \leq v \leq m$ is generated and sent to the cell (m+3). In the cell (m+3), after receiving the updated $\mathbf{S}^v, 1 \leq v \leq m$ from the cell (m+2), the rule R_5 is applied to generate \mathbf{U}_2 and send it to the cell (m+4).
- Step 5: When the cell (m+4) receives the \mathbf{U}_2 sent from the cell (m+3), the rules R_6 and R_7 are applied to generate updated $\mathbf{F}_{\mathbf{U}_2}$ and $\alpha^1, \dots, \alpha^m$, which are sent back to the cell (m+3). Then apply rule R_5 in the cell (m+3) again.

Steps 4 to 5 are a cyclic process.

- Step 6(Termination of calculation): The calculation is terminated when the relative change of Eq 18 does not exceed 10^{-8} . Then the updated \mathbf{U}_2 is output. The specific process is shown by Algorithm 3.

Algorithm 3 RMVCP

Input: Data matrices: $\mathbf{x}^1, \dots, \mathbf{x}^m$, parameters $\alpha > 0, \beta > 0, \gamma > 0, \mathbf{I}$.
output: Unified matrix $\mathbf{U}_1, \mathbf{F}_{\mathbf{U}_1}, \mathbf{U}_2 \in R^{n \times n}$ with c connected components, $\mathbf{F}_{\mathbf{U}_2}$.
Initialize: Random matrices \mathbf{U}_1 and $\mathbf{F}_{\mathbf{U}_1}$, $w_v = 1/m$, random $\alpha^v, 1 \leq v \leq m$.
1: cell 1-m: R_{10}, R_1 : Generate $\mathbf{S}^v, 1 \leq v \leq m$ and send it to the cell (m+1);
repeat
2: cell (m+1): R_{20}, R_2 : The updated \mathbf{U}_1 is generated and sent to the cell (m+2);
3: cell (m+2): R_3 : The updated w_1, \dots, w_m are generated and sent to the cell (m+1);
4: The updated $\mathbf{F}_{\mathbf{U}_1}$ is generated and sent to the cell (m+1);
 R_{10}, R_1 : The updated $\mathbf{S}^v, 1 \leq v \leq m$ are generated and sent to the cell (m+1).
until Rule R_{40} is triggered, and the updated $\mathbf{S}^v, 1 \leq v \leq m$ are generated and sent to the cell (m+3).
repeat
4: cell (m+3): R_5 : The updated \mathbf{U}_1 is generated and sent to the cell (m+4);
5: cell (m+4): $R_6: \mathbf{F}_{\mathbf{U}_2} \rightarrow$ cell (m+3);
 R_7 : The updated $\alpha^1, \dots, \alpha^m$ are generated and sent to the cell (m+3);
until The condition for termination of calculation is met. Output \mathbf{U}_1 .

3.5 Convergence analysis of RMVCP-2

In this section we prove the convergence of RMVCP-2.

Lemma 1. For any positive numbers a and b , inequality 23 holds:

$$a - \frac{a^2}{2b} \leq b - \frac{b^2}{2b} \quad (24)$$

Proof. We use $\tilde{\mathbf{U}}_2$ to represent the updated \mathbf{U}_2 after each iteration. After the first iteration of the loop, we get:

$$\tilde{\mathbf{U}}_2 = \underset{u_{2ij} \geq 0, u_{2i} \mathbf{1}_n = 1, \text{rank}(\mathbf{L}_{\mathbf{U}_2}) = n-c}{\text{argmin}} \sum_{v=1}^m \alpha^{(v)} \|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F^2 \quad (25)$$

Because of $\alpha^{(v)} = \frac{1}{2\|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F}$, we obtain:

$$\sum_{v=1}^m \frac{\|\tilde{\mathbf{U}}_2 - \mathbf{S}^{(v)}\|_F^2}{2\|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F} \leq \sum_{v=1}^m \frac{\|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F^2}{2\|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F} \quad (26)$$

From Lemma 1, we obtain:

$$\begin{aligned} & \sum_{v=1}^m \|\tilde{\mathbf{U}}_2 - \mathbf{S}^{(v)}\|_F - \sum_{v=1}^m \frac{\|\tilde{\mathbf{U}}_2 - \mathbf{S}^{(v)}\|_F^2}{2\|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F} \\ & \leq \sum_{v=1}^m \|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F - \sum_{v=1}^m \frac{\|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F^2}{2\|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F} \end{aligned} \quad (27)$$

After deduction, we obtain:

$$\sum_{v=1}^m \|\tilde{\mathbf{U}}_2 - \mathbf{S}^{(v)}\|_F \leq \sum_{v=1}^m \|\mathbf{U}_2 - \mathbf{S}^{(v)}\|_F \quad (28)$$

Therefore, it can be seen that the value of the objective function of each iteration will decrease, and finally meet the KKT condition of the objective function, and converge to the local optimal solution.

3.6 Complexity analysis

RMVCP-1:

The complexity of RMVCP-1 mainly comes from the update of $\mathbf{S}^{(v)}$ and $\mathbf{F}_{\mathbf{U}_1}$. When updating $\mathbf{S}^{(v)}$, it costs $O(n^3)$ due to matrix multiplication and matrix inversion. In the process of updating $\mathbf{F}_{\mathbf{U}_1}$, the operation of singular value decomposition takes $O(n^3)$.

RMVCP-2:

The complexity of RMVCP-2 mainly comes from the process of updating the weights $\alpha^{(v)}$ and $\mathbf{F}_{\mathbf{U}_2}$. The complexity of updating weight $\alpha^{(v)}$ is $O(mn^2)$. When updating $\mathbf{F}_{\mathbf{U}_2}$, it is necessary to calculate the eigenvector of the Laplacian matrix of \mathbf{U}_2 , so the complexity of updating $\mathbf{F}_{\mathbf{U}_2}$ is $O(cn^2)$.

4. Experiments

4.1 Datasets

In order to verify the clustering performance of our proposed RMVCP algorithm, we conduct comparative experiments on five public datasets. The five datasets include ORL, MSRC, HW, Yale, Wikipedia Article. The specific information of the five datasets is as follows:

- ORL: ORL [55] is an image dataset, which contains 400 images from 40 people. Each person has 10 different images. ORL has four views, namely GIST (512), LBP (59), HOG (864) and CENTRIST (254) (the dimensions of each view are in parentheses).
- MSRC: MSRC [56] is an image dataset. It contains 210 samples of 7 types. The 7 categories are bicycle, tree, car, airplane, building, cow, face. There are 30 images in each category. There are six views in MSRC, namely CENTRIST (1302), CMT (48), GIST (512), HOG (100), LBP (256), SIFT (210).
- HW: HW [57] is an image dataset. It contains 2000 images in 10 categories. These 10 categories respectively show one of the 10 numbers “0–9”. There are 200 images in each category. HW has 6 views, namely FAC (216), FOU (76), KAR (64), MOR (6), PIX (240), ZER (47).
- Yale: The Yale dataset [56] is an image dataset. It contains 165 samples in 15 categories. Each category shows a different person, and each person has 11 different states, wearing glasses and not wearing glasses, and so on. Yale has three views, namely Intensity (4096), LBP (3304), Gabor (6750).
- Wikipedia Article: Wikipedia Article [58] is a dataset composed of featured articles selected from Wikipedia. It contains 693 samples in 10 categories. There are two views in Wikipedia Article, with feature dimensions of 128 and 10 respectively.

The specific information of these five datasets is shown in [Table 1](#), where d1, d2, d3, d4, d5, d6 is the number of features in each view, n is the number of samples, and c is the number of clusters.

4.2 Comparison algorithms and evaluation indicators

In this paper, in order to prove the effectiveness of our proposed RMVCP algorithm, we compare the RMVCP algorithm with some other state-of-the-art algorithms. These algorithms include single-view spectral clustering (SC), connected feature methods (CF), Auto-weighted Multiple Graph Learning (AMGL) [49], One-step Multi-view Spectral Clustering (OMSC) [34], Multi-view Concept Clustering (MVCC) [59], Multi-View Clustering via Deep Matrix Factorization (MVC-DMF) [60], Deep Matrix Factorization based Solution (DMFClusts) [61], Binary Multi-view Clustering (BMVC) [62], Multi-graph Fusion for Multi-view Spectral Clustering (GFSC) [41], Multi-View Clustering in Latent Embedding Space (MCLES) [63], Multi-view clustering via deep concept factorization (MCDCF) [64].

Table 1. The specific details of the five datasets ORL, MSRC, HW, Yale, Wikipedia Article.

	d1	d2	d3	d4	d5	d6	n	c
ORL	512	59	864	254	-	-	400	40
MSRC	1302	48	512	100	256	210	210	7
HW	216	76	64	6	240	47	2000	10
Yale	4096	3304	6750	-	-	-	165	15
Wikipedia Article	128	10	-	-	-	-	693	10

<https://doi.org/10.1371/journal.pone.0269878.t001>

In order to verify the clustering performance of each algorithm, the evaluation criteria we adopt in this paper are **Accuracy (Acc)**, **Normalized Mutual Information (NMI)**, **Purity** [65]. The calculation methods of these performance metrics are as follows:

Acc is used to verify whether the obtained label is consistent with the real label provided by the data:

$$\text{Acc} = \frac{\sum_{i=1}^n \tau(z^i, \text{map}(o_i))}{N}$$

where z^i is the label after clustering, o_i is the true label, n is the total number of data points, τ is the indicator function.

NMI: First define A and B as two random variables, $H(A)$ and $H(B)$ are their corresponding entropy respectively, then use the following formula to calculate NMI:

$$\text{NMI}(A, B) = \frac{I(A, B)}{\sqrt{H(A)H(B)}}$$

where $I(A, B)$ represents the mutual information between A and B . The larger the value, the better the performance.

Purity: Purity is defined as the proportion of documents that are correctly clustered to the total documents. The formula is as follows:

$$\text{purity} = \frac{1}{N} \sum_{i=1}^k \max |b_i \cap g_j|$$

b_i represents the i -th cluster and g_j represents the classification that has the maximum count for cluster b_i .

4.3 Evaluation of experimental results

We conduct experiments on five datasets ORL, MSRC, HW, Yale and Wikipedia Article, and the compared algorithms are SC, CF, AMGL, OMSC, MVCC, MVC-DMF, DMFClusts, BMVC, GFSC, MCLES, MCDCF. Tables 2–4 respectively shows the comparison of the Acc, NMI, and Purity results of the RMVCP algorithm and other algorithms on the five data sets. The best results are highlighted in bold and the second-best results are underlined. Figs 4–6 shows the histogram comparison of the Acc, NMI, and Purity results of all algorithms on the five data sets.

- It can be seen from the experimental results that compared to multi-view clustering, the clustering performance of single-view clustering is worse than that of multi-view clustering. In these five data sets, the clustering performance of the spectral clustering algorithm for each view is not very satisfactory. From the Acc results, for ORL, MSRC, HW, Yale, Wikipedia Article data sets, the best spectral clustering results are 25.87%, 22.66%, 7.26%, 24.02%, 2.78% lower than RMVCP, respectively. This fully shows that the RMVCP algorithm is better than the single-view spectral clustering algorithm.
- For the feature connection method, all the features are connected together and single-view spectral clustering is performed on them. This method simply superimposes the features together, and the Acc results on the five data sets are 26.3%, 36.85%, 20.33%, 38.62%, 1.24% lower than the RMVCP algorithm respectively. This fully illustrates the importance of assigning weights to views.

Table 2. Comparison of algorithms on five datasets for Acc.

	ORL	MSRC	HW	Yale	Wikipedia Article
SC (1)	53.28	33.00	73.44	24.14	17.36
SC (2)	44.93	56.00	71.23	37.80	52.63
SC (3)	54.63	50.00	63.49	25.25	-
SC (4)	39.60	57.86	10.84	-	-
SC (5)	-	28.33	63.44	-	-
SC (6)	-	60.67	75.49	-	-
CF	54.20	46.48	62.42	23.20	54.17
AMGL	54.00	72.86	71.13	41.98	38.23
OMSC	39.00	31.43	12.35	16.15	22.08
MVCC	67.75	75.71	84.11	21.77	58.66
MVC-DMF	41.16	36.27	42.07	47.23	43.72
DMFClusts	41.75	46.71	50.93	15.37	23.13
BMVC	56.93	39.90	27.18	29.25	19.29
GFSC	63.68	56.43	70.48	55.88	49.61
MCLES	<u>76.25</u>	74.29	~ 1	6.67	54.11
MCDCF	70.25	<u>82.76</u>	~ 1	<u>59.94</u>	45.80
RMVCP	80.50	83.33	<u>82.75</u>	61.82	<u>55.41</u>

¹~ indicates that the running time of the algorithm exceeds one hour, and the following is the same.

<https://doi.org/10.1371/journal.pone.0269878.t002>

- Compared with these multi-view clustering algorithms, in general, the RMVCP algorithm is better than other multi-view clustering algorithms. From the Acc results, the MVCC algorithm is second only to the RMVCP algorithm, which shows that the multi-view conceptual clustering method has good clustering performance in the multi-view clustering, and the RMVCP algorithm is superior to the conceptual clustering method.

Table 3. Comparison of algorithms on five datasets for NMI.

	ORL	MSRC	HW	Yale	Wikipedia Article
SC (1)	74.14	21.46	77.62	18.12	5.33
SC (2)	67.28	46.17	72.12	37.62	49.81
SC (3)	74.31	43.45	63.05	20.64	-
SC (4)	57.15	58.23	1.06	-	-
SC (5)	-	15.14	67.03	-	-
SC (6)	-	52.19	<u>80.67</u>	-	-
CF	74.57	41.26	64.92	17.61	51.09
AMGL	75.91	73.20	78.69	43.61	26.60
OMSC	53.93	20.74	22.32	6.87	35.78
MVCC	84.35	65.25	78.81	11.92	<u>51.49</u>
MVC-DMF	60.81	19.55	46.48	45.51	44.75
DMFClusts	63.66	32.81	46.20	3.41	17.32
BMVC	73.91	21.23	13.96	21.75	7.84
GFSC	81.83	51.05	74.19	60.67	43.15
MCLES	<u>88.58</u>	70.40	~	0	46.90
MCDCF	85.30	<u>76.06</u>	~	<u>64.22</u>	34.82
RMVCP	89.07	84.25	87.62	64.42	51.58

<https://doi.org/10.1371/journal.pone.0269878.t003>

Table 4. Comparison of algorithms on five datasets for Purity.

	ORL	MSRC	HW	Yale	Wikipedia Article
SC (1)	61.63	35.95	77.93	25.08	19.51
SC (2)	50.55	58.00	72.51	41.35	56.46
SC (3)	60.80	54.14	66.15	26.88	-
SC (4)	44.15	63.57	11.13	-	-
SC (5)	-	30.76	66.73	-	-
SC (6)	-	64.19	78.99	-	-
CF	61.03	48.48	65.74	24.68	58.86
AMGL	62.50	78.57	74.88	44.08	41.00
OMSC	49.00	34.29	29.75	16.92	44.16
MVCC	72.25	75.71	85.00	23.09	61.31
MVC-DMF	43.38	39.51	52.88	50.99	51.00
DMFClusts	47.20	50.71	55.58	19.12	28.24
BMVC	59.40	40.10	28.46	29.95	21.31
GFSC	78.38	74.71	82.27	69.70	62.29
MCLES	81.25	79.52	~	6.67	56.13
MCDCF	75.73	82.76	~	61.39	48.60
RMVCP	85.00	83.81	86.90	61.82	59.88

<https://doi.org/10.1371/journal.pone.0269878.t004>

- The GFSC algorithm uses a self-representation method to generate the similarity matrix of each view, without performing the second fusion, and finally uses an additional spectral clustering step to generate the final clustering result. It can be seen from the results that the clustering performance of the GFSC algorithm is worse than that of the RMVCP algorithm, which indicates that the secondary fusion can allocate the weight of views more reasonably and reduce the influence of noise information. At the same time, in terms of accuracy, the performance of directly generating clustering results is better than using additional clustering steps.
- AMGL is a self-weighted graph learning method with good clustering performance, but only one weight assignment is performed in the clustering process. It can be seen from the results

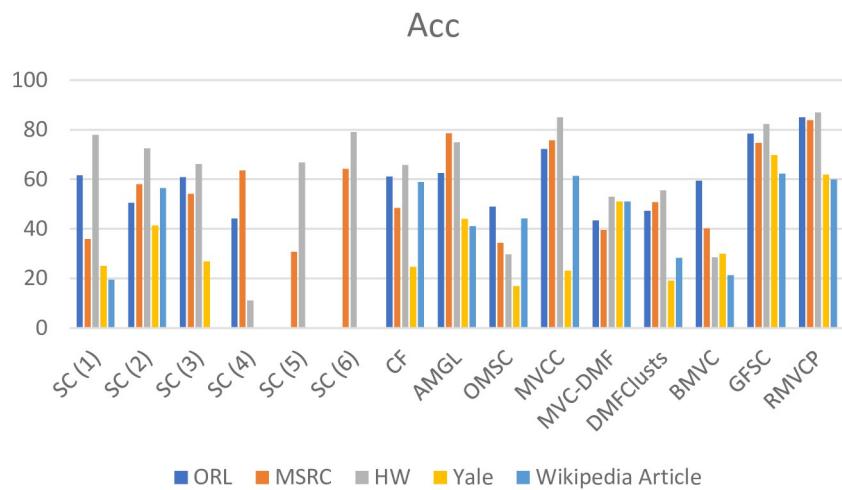


Fig 4. A histogram comparison of the Acc results of all algorithms on the five datasets.

<https://doi.org/10.1371/journal.pone.0269878.g004>

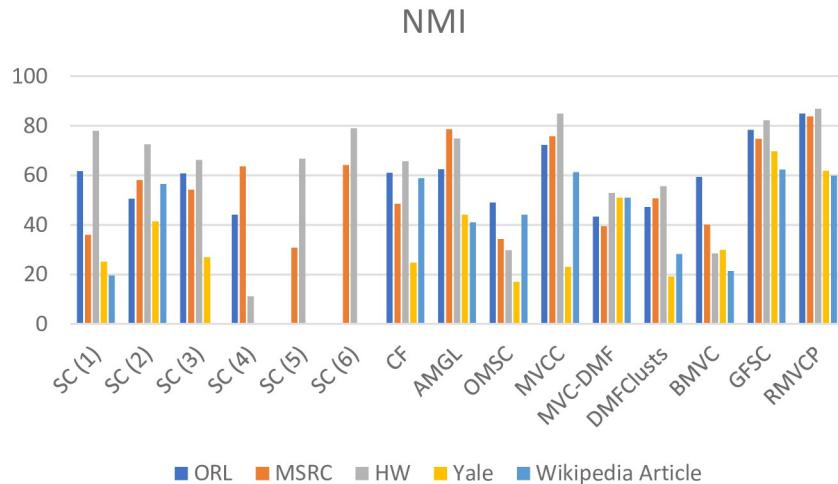


Fig 5. A histogram comparison of the NMI results of all algorithms on the five datasets.

<https://doi.org/10.1371/journal.pone.0269878.g005>

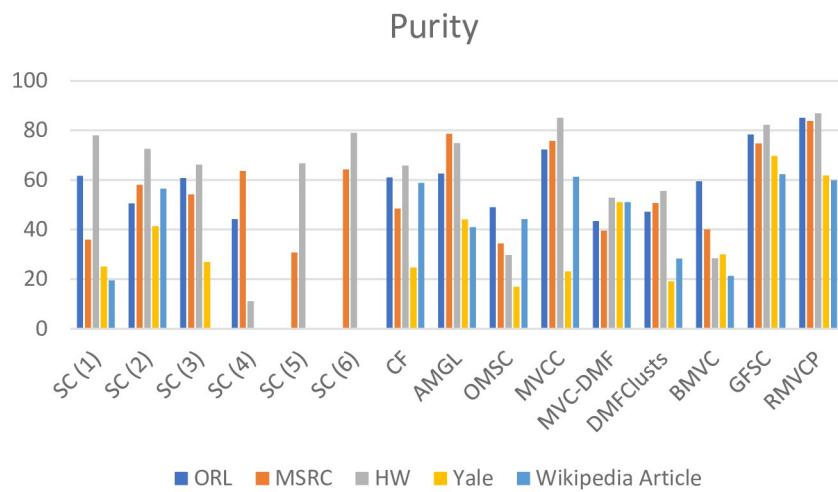


Fig 6. A histogram comparison of the Purity results of all algorithms on the five datasets.

<https://doi.org/10.1371/journal.pone.0269878.g006>

that the RMVCP algorithm is superior to the AMGL algorithm in all aspects, which illustrates the importance of the secondary distribution of weights.

- MCLES searches for the potential embedding space of data to explore the global information of data. Concept decomposition and deep learning are applied to multi-view clustering by MCDCF. The experimental results reveal that the running time of MCLES and MCDCF on the HW dataset with 2000 data points is more than an hour, which indicates that the two algorithms cannot deal with a slightly larger dataset, and RMVCP can deal with this kind of dataset.

4.4 Weight distribution analysis and convergence speed in the process of RMVCP-2

The weight distribution of each view plays a vital role in the clustering performance of multi-view clustering. The RMVCP algorithm will assign the weight of each view twice, which will

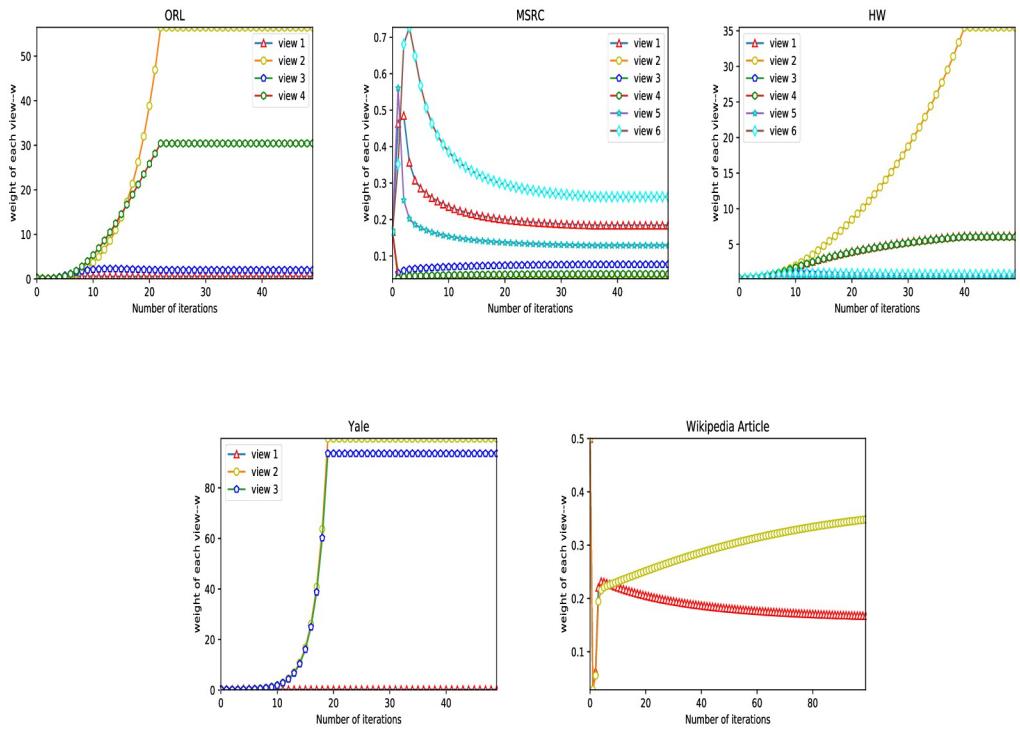


Fig 7. The change of the weight w of each view on the five datasets in the RMVCP-1 process.

<https://doi.org/10.1371/journal.pone.0269878.g007>

make up for the defect of unreasonable weight assignment that may be caused by assigning the weight once. Fig 7 shows the change of the weight w of each view on the five datasets in the RMVCP-1 process. Fig 8 demonstrates the change of the weight $\alpha^{(v)}$ of each view on the five datasets in the RMVCP-2 process.

In the RMVCP-1 process, most of the views of the four datasets ORL, MSRC, HW, and Wikipedia Article are assigned weights with higher discrimination. In the RMVCP-2 process, MSRC, HW, and Wikipedia Article also assign a higher discrimination weight to each view, which indicates that on these three data sets, it is not enough to assign weights once to them. A second weight assignment process is needed to achieve a more reasonable weight assignment. In Yale's two weight assignment processes, the weights of the three views are not much different, which shows that the quality of the three views may be similar. For the ORL data set, the weight distribution in the RMVCP-2 process is not much different, indicating that the data set may be easier to distinguish without the need for two weight distribution processes.

Fig 9 shows the change of the objective function value of RMVCP-2 on five data sets of ORL, MSRC, HW, Yale, and Wikipedia Article. Obviously, the convergence speed of the objective function on the five data sets is very fast. The four data sets of ORL, HW, Yale, and Wikipedia Article all converged within five iterations. The MSRC data set has a relatively slow convergence rate, converging around 15 iterations.

4.5 Visual analysis of unified matrices \mathbf{U}_1 and \mathbf{U}_2

To verify the effectiveness of double fusion for weight assignment, we visualize the \mathbf{U}_1 produced by the RMVCP-1 process and the \mathbf{U}_2 produced by the RMVCP-2 process, respectively. Since both \mathbf{U}_1 and \mathbf{U}_2 are subjected to the Constrained Laplacian Rank operation, the better the clustering performance, the clearer the block structure of \mathbf{U}_1 and \mathbf{U}_2 . Figs 10 and 11 shows

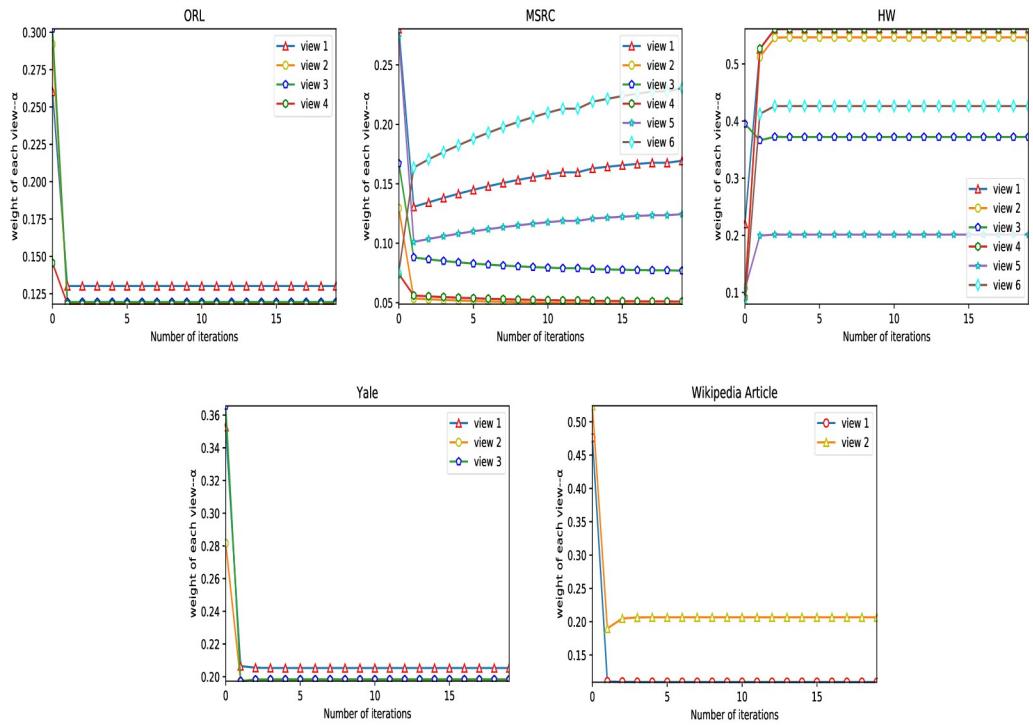


Fig 8. The change of the weight $\alpha^{(v)}$ of each view on the five datasets in the RMVCP-2 process.

<https://doi.org/10.1371/journal.pone.0269878.g008>

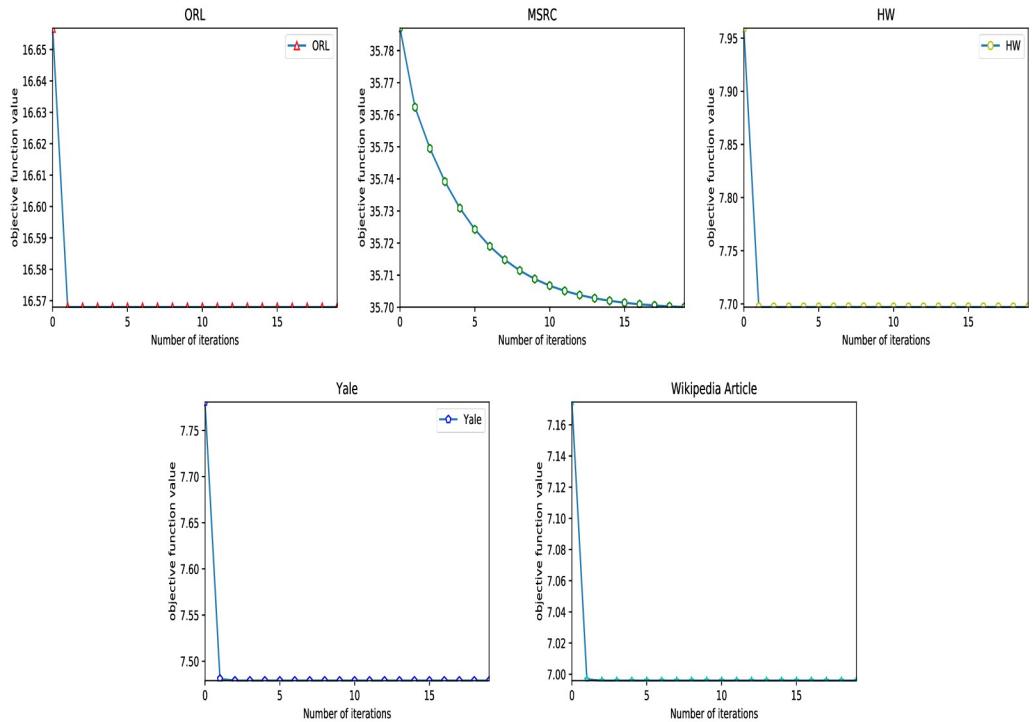


Fig 9. The change of the objective function value of RMVCP-2 on five data sets of ORL, MSRC, HW, Yale, and Wikipedia Article.

<https://doi.org/10.1371/journal.pone.0269878.g009>

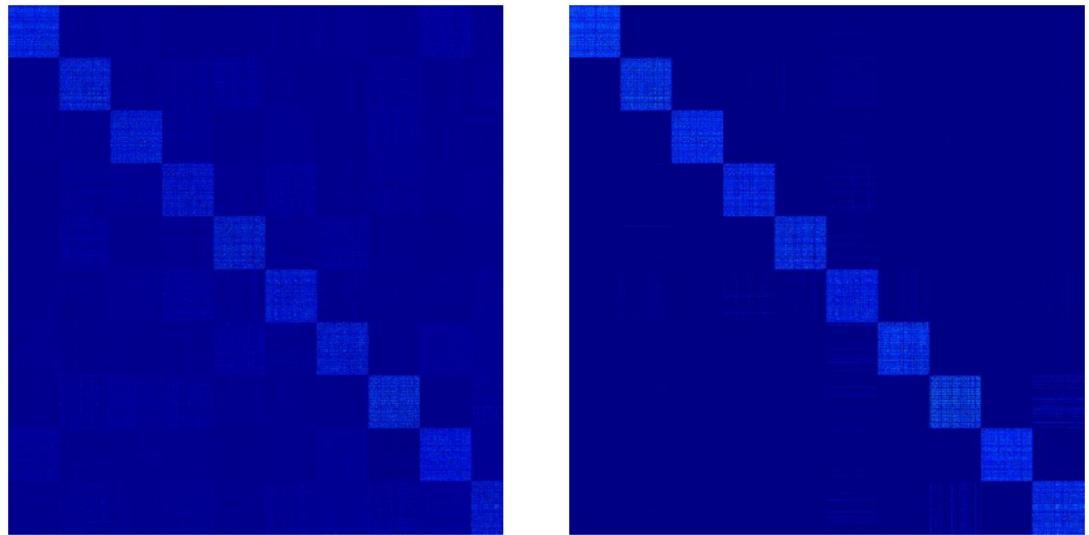


Fig 10. Visual analysis of unified matrices U_1 and U_2 on HW.

<https://doi.org/10.1371/journal.pone.0269878.g010>

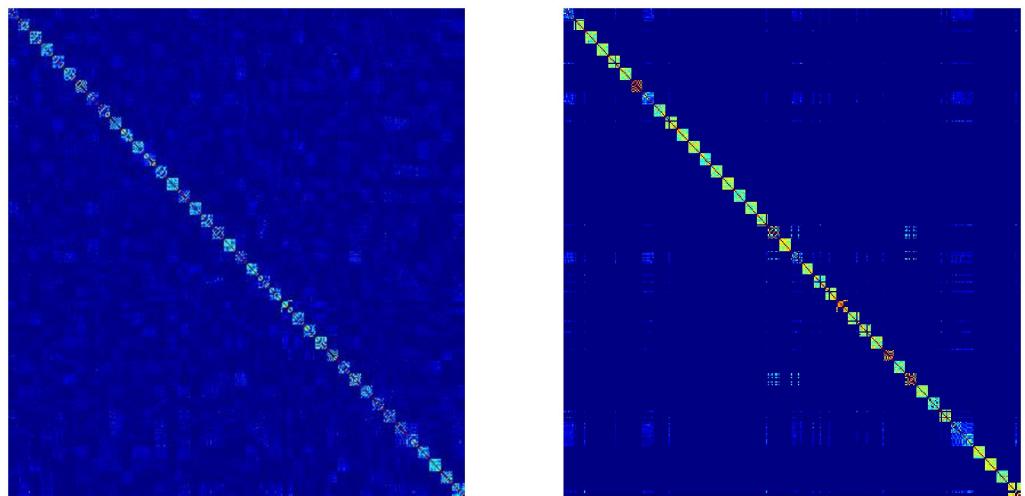


Fig 11. Visual analysis of unified matrices U_1 and U_2 on ORL.

<https://doi.org/10.1371/journal.pone.0269878.g011>

the block structure of U_1 and U_2 on HW and ORL. It is obvious from Figs 10 and 11 that U_2 has a clearer block structure and less noisy data than U_1 . This shows that RMVCP-2 is a necessary and effective step to improve clustering performance due to its better weight assignment of views and reduction of noisy data.

4.6 Parameter analysis

There are three hyperparameters in the RMVCP algorithm, namely α , β , and γ , all of which need to be set in advance before the experiment. The value of γ will not change the clustering results in the actual experiment. We set γ to 0.01 in the experiment. Fig 12 shows the influence of the changes of hyperparameters α , β , and γ on the results of Acc on the five datasets. After doing a lot of experiments, we found that the best hyperparameters α , β , and γ on ORL are set

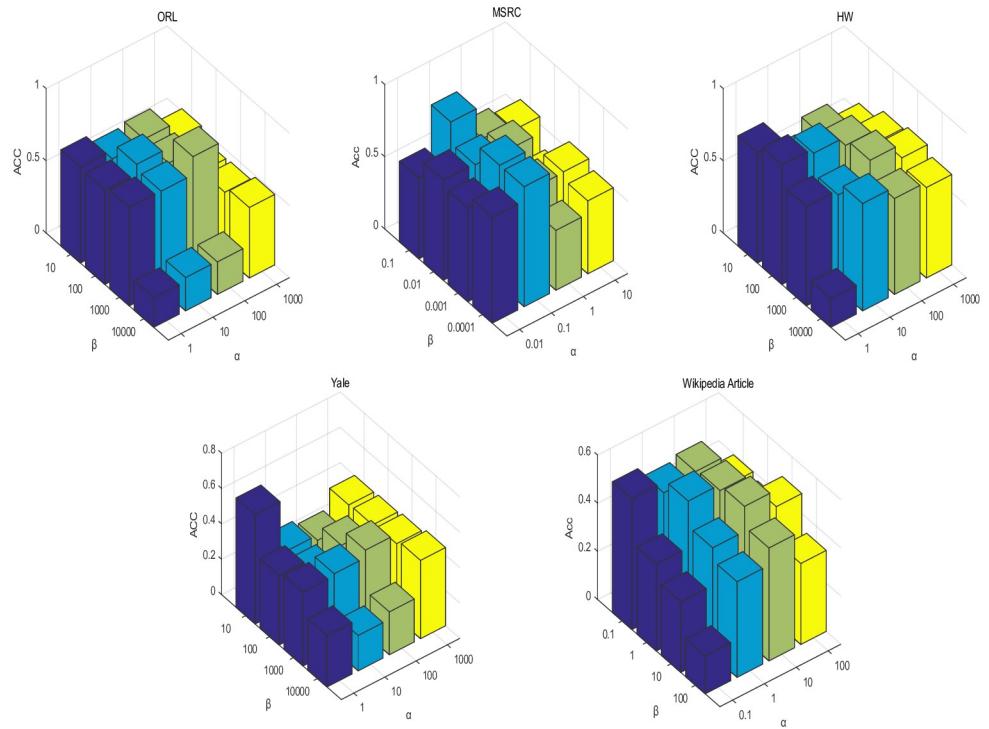


Fig 12. The influence of the changes of hyperparameters α , β , and γ on the results of Acc on the five datasets.

<https://doi.org/10.1371/journal.pone.0269878.g012>

to 100, 1000, and 0.01. On MSRC, the parameters are set to 0.1, 0.1, 0.1. The parameter setting on HW is 1, 100, 0.01. The parameter is set to 1, 10, 0.01 on Yale. The parameters on Wikipedia Article are set to 1, 1, 0.01.

5. Discussion

Extensive experiments have verified that the clustering performance of RMVCP algorithm is better than that of other state-of-the-art algorithms, indicating the effectiveness of twice weight allocation for each view and combination with tissue-like P system. The quality of each view is irregularity, and RMVCP performs two weight assignment operations on each view. The results and effects are revealed in the experiment to verify its effectiveness. In addition to the better accuracy of the RMVCP algorithm, RMVCP can also handle a slightly larger dataset rather than MCLES and MCDCF. However, three parameters need to be set in advance in RMVCP algorithm. It can be seen from parameter sensitivity experiment that RMVCP algorithm is sensitive to α and β on some datasets, which is the deficiency of RMVCP algorithm. Therefore, we will focus on the parameter problem of the algorithm in the future, and strive to reduce the number of parameters and weaken the influence of different parameter values on clustering performance.

6. Conclusion and future research

In this paper, in order to solve the problem of view weight distribution and noise reduction in multi-view clustering, Reweighted multi-view clustering with tissue-like P system (RMVCP) are proposed. Inspired by multi-view subspace clustering and graph-based multi-view clustering, RMVCP performs a two-step operation. In the first step (RMVCP-1), the self-representation method is used to construct the similarity matrix of each view, and then the fusion

operation is performed. In the second step (RMVCP-2), the updated similarity matrix of each view generated in the process of RMVCP-1 is used as input for the second fusion operation. Correspondingly, the weight of each view has been allocated more reasonably. At the same time, we combine the RMVCP algorithm with the tissue-like P system, and use the computational parallelism of the tissue-like P system to improve the computational efficiency of the RMVCP algorithm. In the future, we can use the idea of secondary fusion in some other state-of-the-art multi-view clustering algorithms, and at the same time, we can combine multiple models in membrane computing with clustering algorithms.

Supporting information

S1 File. 5 datasets are used in the experiment in this paper.
(ZIP)

Author Contributions

Conceptualization: Xiyu Liu.

Funding acquisition: Xiyu Liu.

Methodology: Huijian Chen.

Supervision: Xiyu Liu.

Validation: Huijian Chen.

Visualization: Huijian Chen.

Writing – original draft: Huijian Chen.

Writing – review & editing: Xiyu Liu.

References

1. Paun G. Computing with membranes. *J Comput Syst Sci*. 2000; 61(1):108–43. <https://doi.org/10.1006/jcss.1999.1693>
2. Alhazov A, Freund R, Ivanov S. When catalytic P systems with one catalyst can be computationally complete. *Journal of Membrane Computing*. 2021. <https://doi.org/10.1007/s41965-021-00079-x>
3. Ceterchi R, Zhang L, Subramanian KG, Zhang G. Hilbert words as arrays generated with P systems. *Journal of Membrane Computing*. 2021. <https://doi.org/10.1007/s41965-021-00078-y>
4. Zhang G, Pérez-Jiménez M, Riscos-Núez A, Verlan S, Gheorghe M. *Membrane Computing Models: Implementations*. Springer. 2021
5. Zhang G, Pérez-Jiménez M, Gheorghe M. *Real-life Applications with Membrane Computing*. Springer International Publishing. 2017
6. Alhazov A, Freund R, Ivanov S. P systems with limited number of objects. *Journal of Membrane Computing*. 2021; 3(1):1–9. <https://doi.org/10.1007/s41965-020-00068-6>
7. Díaz-Pernil D, Pérez-Jiménez M, Romero-Jiménez á. Efficient simulation of tissue-like P systems by transition cell-like P systems. *Natural Computing*. 2009; 8(4):797–806. <https://doi.org/10.1007/s11047-008-9102-z>
8. Aman B, Ciobanu G. Travelling salesman problem in tissue P systems with costs. *Journal of Membrane Computing*. 2021; 3(2):97–104. <https://doi.org/10.1007/s41965-021-00077-z>
9. Gheorghe M, Lepticaru R, Konur S, Niculescu IM, Adorna HN. Spiking neural P systems: matrix representation and formal verification. *Journal of Membrane Computing*. 2021; 3(2):133–48. <https://doi.org/10.1007/s41965-021-00075-1>
10. Zhang G, Rong H, Paul P, He Y, Neri F, P MJ. A Complete Arithmetic Calculator Constructed from Spiking Neural P Systems and its Application to Information Fusion. *International Journal of Neural Systems*. 2020; 31(01). <https://doi.org/10.1142/S0129065720500550>

11. Sosík P, Drastík J, Smolka V, Garzon M. From P systems to morphogenetic systems: an overview and open problems. *Journal of Membrane Computing*. 2020; 2(4):380–91. <https://doi.org/10.1007/s41965-020-00057-9>
12. Nie F, Wang X, Huang H. Clustering and projected clustering with adaptive neighbors. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*; New York, New York, USA: Association for Computing Machinery; 2014. p. 977–86.
13. Wang X, Huang DS. A novel density-Based clustering framework by using level set method. *IEEE Transactions on Knowledge and Data Engineering*. 2009; 21(11):1515–31. <https://doi.org/10.1109/TKDE.2009.21>
14. Ohi AQ, Mridha MF, Safir FB, Hamid MA, Monowar MM. AutoEmbedder: A semi-supervised DNN embedding system for clustering. *Knowl-Based Syst*. 2020. <https://doi.org/10.1016/j.knosys.2020.106190>
15. Nie F, Wu D, Wang R, Li X. Self-weighted clustering with adaptive neighbors. *IEEE Transactions on Neural Networks and Learning Systems*. 2020; PP(99):1–14. PMID: [32011264](#)
16. Huang D, Wang CD, Wu J, Lai J, Kwok CK. Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*. 2019; 32(6):1212–26. <https://doi.org/10.1109/TKDE.2019.2903410>
17. Affeldt S, Labiod L, Nadif M. Spectral clustering via ensemble deep autoencoder learning (SC-EDAE). *Pattern Recogn*. 2020; 108. <https://doi.org/10.1016/j.patcog.2020.107522>
18. Ye X, Zhao J, Chen Y, Guo LJ. Bayesian adversarial spectral clustering with unknown cluster number. *IEEE Transactions on Image Processing*. 2020; 29:8506–18. <https://doi.org/10.1109/TIP.2020.3016491> PMID: [32813658](#)
19. Shi D, Zhu L, Li Y, Li J, Nie X. Robust structured graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*. 2019; 31(11):4424–36. <https://doi.org/10.1109/TNNLS.2019.2955209>
20. Ren P, Xiao Y, Chang X, Prakash M, Chen X. Structured optimal graph-based clustering with flexible embedding. *IEEE Transactions on Neural Networks and Learning Systems*. 2019; 31(10):3801–13. <https://doi.org/10.1109/TNNLS.2019.2946329> PMID: [31722496](#)
21. Chun-Guang Li, Chong You, René Vidal. Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework. *IEEE Transactions on Image Processing*. 2017; 26(6):2988–3001. <https://doi.org/10.1109/TIP.2017.2691557>
22. Khan IK, Luo Z, Huang JZ, Shahzad W. Variable weighting in fuzzy k-means clustering to determine the number of clusters. *IEEE Transactions on Knowledge and Data Engineering*. 2019; 32(9):1838–53. <https://doi.org/10.1109/TKDE.2019.2911582>
23. Su Y, Liu C, Niu Y, Cheng F, Zhang X. A Community Structure Enhancement-Based Community Detection Algorithm for Complex Networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2019; 51(5):2833–46. <https://doi.org/10.1109/TSMC.2019.2917215>
24. Tian Y, Su X, Su Y, Zhang X. EMODMI: A Multi-Objective Optimization Based Method to Identify Disease Modules. *IEEE Transactions on Emerging Topics in Computational Intelligence*. 2020; 5(4):570–82. <https://doi.org/10.1109/TETCI.2020.3014923>
25. Wang H, Zhao J, Su Y, Zheng C. scCDG: A Method based on DAE and GCN for scRNA-seq data Analysis. *IEEE/ACM transactions on computational biology and bioinformatics*. 2021. <https://doi.org/10.1109/TCBB.2021.3126641> PMID: [34752401](#)
26. Liu BY, Huang L, Wang CD, Lai JH, Yu P. Multi-view consensus proximity learning for clustering. *IEEE Transactions on Knowledge and Data Engineering*. 2020. <https://doi.org/10.1109/TKDE.2020.3025759>
27. Yu X, Liu H, Wu Y, Zhang CM. Fine-grained similarity fusion for multi-view spectral clustering q. *Inform Sciences*. 2021; 568:350–68. <https://doi.org/10.1016/j.ins.2021.03.059>
28. Zheng Q, Zhu J, Ma Y, Li Z, Tian Z. Multi-view subspace clustering networks with local and global graph information. *Neurocomputing*. 2021; 449:15–23. <https://doi.org/10.1016/j.neucom.2021.03.115>
29. Wu D, Hu Z, Nie F, Wang R, Yang H, Li X. Multi-view clustering with interactive mechanism. *Neurocomputing*. 2021; 449:378–88. <https://doi.org/10.1016/j.neucom.2021.03.065>
30. Zhang G, Zhou Y, He X, Wang C, Huang D. One-step kernel multi-view subspace clustering. *Knowl-Based Syst*. 2020; 189. <https://doi.org/10.1016/j.knosys.2019.105126>
31. Zhao Q, Zong L, Zhang X, Liu X, Yu H. Multi-view clustering via clusterwise weights learning. *Knowl-Based Syst*. 2020; 193. <https://doi.org/10.1016/j.knosys.2019.105459>
32. Qian Y, Yin X, Kong J, Wang J, Gao W. Low-rank graph optimization for multi-view dimensionality reduction. *Plos One*. 2019; 14(12). <https://doi.org/10.1371/journal.pone.0225987> PMID: [31851696](#)
33. Yang M, Sinaga K. A feature-reduction multi-view k-means clustering algorithm. *IEEE Access*. 2019; 7:114472–86. <https://doi.org/10.1109/ACCESS.2019.2934179>

34. Zhu X, Zhang S, Hu R, He W, Lei C, Zhu P. One-step multi-view spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*. 2018; 31(10):2022–34. <https://doi.org/10.1109/TKDE.2018.2873378>
35. Zhuge W, Hou C, Jiao Y, Yue J, Tao H, Yi D. Robust auto-weighted multi-view subspace clustering with common subspace representation matrix. *Plos One*. 2017; 12(5). <https://doi.org/10.1371/journal.pone.0176769> PMID: 28542234
36. Shi S, Nie F, Wang R, Li X. Multi-view clustering via nonnegative and orthogonal graph reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*. 2021;1–14. <https://doi.org/10.1109/TNNLS.2021.3108050> PMID: 34288875
37. Wang H, Yang Y, Liu B. GMC: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*. 2019; 32(6):1116–29. <https://doi.org/10.1109/TKDE.2019.2903810>
38. Zhang X, Zhang X, Liu H, Liu X. Multi-task multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*. 2016; 28(12):3324–38. <https://doi.org/10.1109/TKDE.2016.2603983>
39. Guo J, Yin W, Sun Y, Hu Y. Multi-view subspace clustering with block diagonal representation. *IEEE Access*. 2019; 7:84829–38. <https://doi.org/10.1109/ACCESS.2019.2923614>
40. Yin M, Liu W, Li MS, Jin TS, Ji RR. Cauchy loss induced block diagonal representation for robust multi-view subspace clustering. *Neurocomputing*. 2021; 427:84–95. <https://doi.org/10.1016/j.neucom.2020.11.017>
41. Kang Z, Shi G, Huang S, Chen W, Pu X, Zhou JT, et al. Multi-graph fusion for multi-view spectral clustering. *Knowl-Based Syst*. 2019; 189.
42. Zhang P, Liu X, Xiong J, Zhou S, Cai Z. Consensus one-step multi-view subspace clustering. *IEEE Transactions on Knowledge and Data Engineering*. 2020; 31.
43. Liu S, Ding C, Jiang F, Wang Y, Yin B. Auto-weighted multi-view learning for semi-Supervised graph clustering. *Neurocomputing*. 2019; 362:19–32. <https://doi.org/10.1016/j.neucom.2019.07.011>
44. Liu J, Liu Y, Zhang Q. A weight initialization method based on neural network with asymmetric activation function. *Neurocomputing*. 2022; 483:171–82. <https://doi.org/10.1016/j.neucom.2022.01.088>
45. Nie F, Wang X, Jordan MI, Huang H. The Constrained Laplacian Rank Algorithm for graph-based clustering. AAAI Press. 2016.
46. Wang X, Guo X, Zhen L, Zhang C, Li SZ. Exclusivity-consistency regularized multi-view subspace clustering. *Computer Vision and Pattern Recognition*; 2017.
47. Zhang C, Hu Q, Fu H, Zhu P, Cao X. Latent multi-view subspace clustering. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017.
48. Wang X, Lei Z, Guo X, Zhang C, Shi H, Li S. Multi-view subspace clustering with intactness-aware similarity. *Pattern Recogn*. 2019; 88:50–63. <https://doi.org/10.1016/j.patcog.2018.09.009>
49. Nie F, Jing L, Li X. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. AAAI Press. 2016.
50. Nie F, Jing L, Li X. Self-weighted multiview clustering with multiple graphs. Twenty-Sixth International Joint Conference on Artificial Intelligence; 2017.
51. Nie F, Cai G, Li X. Multi-view clustering and semi-supervised classification with adaptive neighbours. AAAI press. 2017.
52. Newman MW, Libraty N, On O, On KA. The Laplacian spectrum of graphs. *graph theory, combinations and applications*. 1991; 18(7):871–98.
53. Chung F. Spectral graph theory. *Regional Conference Series in Math*. Cbms Amermathsoc. 1997.
54. Kevin Fan. On a theorem of weyl concerning eigenvalues of linear transformations I. *P Natl Acad Sci USA*. 1949; 35 11.
55. Nie F, Tian L, Wang R, Li X. Multiview semi-supervised learning model for image classification. *IEEE Transactions on Knowledge and Data Engineering*. 2019; 32(12):2389–400. <https://doi.org/10.1109/TKDE.2019.2920985>
56. Xu J, Han J, Nie F, Li X. Multi-view scaling support vector machines for classification and feature selection. *IEEE Transactions on Knowledge and Data Engineering*. 2020; 32(7):1419–30. <https://doi.org/10.1109/TKDE.2019.2904256>
57. Yang M, Deng C, Nie F. Adaptive-weighting discriminative regression for multi-view classification. *Pattern Recogn*. 2019; 88:236–45. <https://doi.org/10.1016/j.patcog.2018.11.015>
58. Yang Y, Wang H. Multi-view clustering: A survey. *Big Data Mining and Analytics*. 2018.
59. Hao W, Yan Y, Li T. Multi-view clustering via concept factorization with local manifold regularization. *IEEE International Conference on Data Mining (ICDM2016)*; 2017.
60. Zhao H, Ding Z, Yun F. Multi-view clustering via deep matrix factorization. AAAI Press. 2017:2921–7.

61. Wei S, Wang J, Yu G, Carlotta, Zhang X. Multi-View multiple clusterings using deep matrix factorization. AAAI Press. 2019; 34:6348–55. <https://doi.org/10.1609/aaai.v34i04.6104>
62. Zhang Z, Liu L, Shen F, Shen HT, Shao L. Binary multi-view clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018; 41(7):1774–82. <https://doi.org/10.1109/TPAMI.2018.2847335> PMID: 29994652
63. Chen M, Huang L, Wang C, and Huang D. Multi-view Clustering in Latent Embedding Space. Proceedings of the AAAI Conference on Artificial Intelligence. 2020.
64. Chang S, Hu J, Li TR, Wang H, and Peng B. Multi-view clustering via deep concept factorization. Knowledge-Based Systems. 2021; 217:106807. <https://doi.org/10.1016/j.knosys.2021.106807>
65. Chong P, Zhao K, Cai S, Qiang C. Integrate and conquer: double-sided two-dimensional k -means via integrating of projection and manifold construction. Acm T Intel Syst Tec. 2018; 9(5):1–25. <https://doi.org/10.1145/3156667>