

Article

Density Peaks Clustering Algorithm Based on a Divergence Distance and Tissue—Like P System

Fuhua Ge and Xiyu Liu * 

Academy of Management Science, Business School, Shandong Normal University, Jinan 250014, China

* Correspondence: xyliu@sdnu.edu.cn

Abstract: Density Peaks Clustering (DPC) has recently received much attention in many fields by reason of its simplicity and efficiency. Nevertheless, empirical studies have shown that DPC has some shortfalls: (i) similarity measurement based on Euclidean distance is prone to misclassification. When dealing with clusters of non-uniform density, it is very difficult to identify true clustering centers in the decision graph; (ii) the clustering centers need to be manually selected; (iii) the chain reaction; an incorrectly assigned point will affect the clustering outcome. To settle the above limitations, we propose an improved density peaks clustering algorithm based on a divergence distance and tissue—like P system (TP-DSDPC in short). In the proposed algorithm, a novel distance measure is introduced to accurately estimate the local density and relative distance of each point. Then, clustering centers are automatically selected by the score value. A tissue—like P system carries out the entire algorithm process. In terms of the three evaluation metrics, the improved algorithm outperforms the other comparison algorithms using multiple synthetic and real-world datasets.

Keywords: density peaks clustering; divergence distance; clustering centers selection; tissue—like P system



Citation: Ge, F.; Liu, X. Density Peaks Clustering Algorithm Based on a Divergence Distance and Tissue—Like P System. *Appl. Sci.* **2023**, *13*, 2293. <https://doi.org/10.3390/app13042293>

Academic Editor: Luis Javier Garcia Villalba

Received: 17 January 2023

Revised: 4 February 2023

Accepted: 8 February 2023

Published: 10 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As an unsupervised machine learning method, clustering analysis separates the datasets into several groups or categories, so points in the same group are similar to each other, while points in different groups are dissimilar [1]. Currently, multiple fields have benefited from clustering analysis, such as data mining [2], image segmentation [3], pattern recognition [4], machine learning [5] and others. There have been thousands of clustering algorithms created for different applications and data characteristics, which can be primarily categorized into partition-based methods [6], hierarchy-based methods [7], grid-based methods [8], model-based methods [9] and density-based methods [10].

During the development of clustering methods, many classic algorithms have emerged. K-means [11] is a centroid-based method that is representative of partition-based algorithms. In K-means algorithms, the closest objects are clustered and classified by their distance from k points in the space. Cluster center values are successively updated to achieve the best clustering result from the iterative process. The algorithm has been extensively employed in several big data processing sectors because of its quick operation speeds and straightforward execution processes. Data objects are hierarchically decomposed using hierarchical methods. Hierarchical clustering can identify and process an unknown number of clusters. Hierarchical methods can be cataloged as cohesive or divisive methods. BIRCH [12] and Chameleon algorithms are two typical hierarchical clustering algorithms. The flaw with the hierarchical approach is that once a step (merge or split) is complete, it cannot be undone. Multi-resolution grids are used for clustering, such as STING [13], and there is no relationship between the amount of data objects processed and the processing time. All that matters is the size of each dimension. For model-based methods, a model is assumed for each cluster, and then a dataset is found that meets this model. The density

distribution of data points in space or another factor might be the basis for such a model. Model-based techniques typically relate to techniques based on neural network models and probability models. Gaussian Mixture Models [14] (GMM, Gaussian Mixture Models) serve as a representation for the former, and Self-Organizing Mapping Networks [15] (SOM, Self-Organized Maps) serve as a representation for the latter. A representative density-based clustering algorithm is called DBSCAN [16]. When compared to partitioned and hierarchical clustering algorithms, it can separate areas with adequate density into clusters, locate clusters of any shape in noisy spatial datasets, and define a cluster as the largest collection of densely linked points. However, DBSCAN has some shortcomings; the improper selection of parameters and MinPts will affect the overall clustering effect, and it cannot handle datasets with large density differences. To address the disadvantage of clustering analysis utilizing a set of global parameters, the OPTICS [17] algorithm is proposed. OPTICS does not intentionally generate dataset clusters, but outputs cluster ordering. HDBSCAN [18] is a clustering algorithm developed by Campello, Moulavi and Sander. It extends DBSCAN by converting DBSCAN into a hierarchical clustering algorithm, and then uses the technique of extracting planar clustering based on clustering stability. The biggest difference between HDBSCAN and traditional DBSCAN is that HDBSCAN can handle clustering problems with different densities. However, the processing effect of boundary points is not ideal.

Rodriguez and Laio recently introduced the Density Peaks Clustering method (DPC) [19] in Science, a novel density-based algorithm. DPC is based on the supposition that (1) the clustering center has a higher density than the neighboring points in its immediate vicinity; (2) the clustering center point is comparatively far from the places with greater densities. In addition, unlike other density-based clustering methods (including MeanShift [20] and DBSCAN), it does not require any iterative step, and assigns labels to non-central points in one step. Although DPC has the advantages of simple and fast calculations, it still has the following shortcomings. On the one hand, while processing data with an uneven density distribution, DPC finds it challenging to identify clustering centers in decision graphs comprised of density and distance. On the other hand, when a point is wrongly allocated during the assignment process, the other points connected to it are also impacted, which causes a chain reaction.

Many DPC variations have been suggested as solutions to the aforementioned problems. Three categories may be used to group together the enhancements of the original DPC. The first improvement aims at improving the calculation of local density. The major goal of the proposed kernels is to lessen the DPC's sensitivity to the parameter's value. As proof, Du et al. first incorporated K-nearest neighbors (KNN) into DPC to determine the local density of data points (KNN-DPC) [21]. Additionally, dimensional reduction is pre-processed using principal component analysis (PCA) when working with high-dimensional datasets to lessen computing complexity. Different from KNN-DPC, Liu et al. proposed shared-nearest-neighbor-based clustering to fast search for and find density peaks (SNN-DPC) [22]. SNN-DPC enhances local density computation by using novel definitions such as SNN similarity. In [23], the K nearest Shannon entropy-based density peaks clustering algorithm named SC was proposed. To determine the local density, K-nearest neighbors and entropy values are used, which takes into account the global and local structure of the dataset.

The second improvement aims to select clustering centers. Picking clustering centers may be improved in two ways: automatically selecting centers and boosting center identification in decision graphs. For instance, a single linkage density peaks clustering algorithm entitled DPSLC [24] was proposed to automatically detect all possible centers. The suggested approach leverages the neighborhood radius to choose a collection of potential density peaks that are far from their nearest higher-density spots. Xu et al. [25] proposed a method for clustering graph adaptive density peaks for automatic center selection. The clustering centers in this approach are automatically chosen according to the turning angle and graph connectivity.

The third improvement seeks to provide effective label assignment strategies. Lotfi, Abdulrahman [26] and colleagues suggested enhanced density peaks clustering (IDPC) that uses a two-step approach to find complicated clusters. Nevertheless, this method has a high sensitivity to the parameters. Recently, the local structure of the dataset has been considered by researchers of the recently introduced DPC-DLP [27] approach, while constructing their label propagation strategy. In this methodology, the border points are given lower density values than the core points, while the core points are given greater density values. By joining each point to its k-nearest neighbors, DPC-DLP builds a graph. This graph is utilized to create cluster backbones, and, after that, all data points receive the labels from the backbones. This method effectively classifies the points that are in the border regions and identifies clusters with complex shapes. The authors of [28] proposed a method called DGDPC. In order to create clusters, each set of points with a decay phenomenon is first assigned to a separate cluster. Then, two clusters are combined using the connection points. In [29], to efficiently handle data of complex forms or multi-manifold structures, DPC is integrated with the minimum spanning tree. Whereas this algorithm has a significant computing resource need and is unable to recognize twisted, folded, or curved clusters. Despite the proliferation of new approaches, there are still several drawbacks that have an impact on clustering outcomes.

Table 1 represents the main four properties of the various improvements of the density peaks clustering methods considering their evaluations.

Table 1. Comparison of various clustering algorithms.

Method	Year	Kernel	Clustering Centers Identification	Label Assignment Strategy	Chain Reaction
DPC	2014	Crisp, global structure	Decision graph	Non-iterative	✓
KNN-DPC	2016	Gaussian, local structure	Decision graph	Non-iterative	✓
IDPC	2016	Gaussian, local structure	Top score	Non-iterative Using a voting	×
DPC-DLP	2019	Gaussian, local structure	Top score	Iterative	×
DGDPC	2021	Gaussian, local structure	Automatically	Non-iterative	×
TP-DSDPC	2022	Gaussian, global structure	Top score automatically	Non-iterative	×

‘×’ refers to No, ‘✓’ means Yes.

As a new research content of biological computing, membrane computing (also known as the P system) draws inspiration from the organism’s own operating mechanisms and cooperation principles; that is, the structure of cells and tissues [30]. It is a model first proposed by Paun [31] in 1998. In membrane computing, there are mainly three types: a cell-like P system [32], a tissue—like P system [33] and a neural-like P system [34]. Every living cell is a running reaction unit of the P system, and each unit independently works. That is to say, each unit of the P system is independently calculated in parallel. The P system’s high level of parallelism means that it is frequently used with other algorithms to increase computing efficiency [35].

So far, the research content of the P system has primarily contained theoretical analysis [36] and application studies [37]. Numerous studies have also been conducted using clustering algorithms in conjunction with membrane computing [38]. In 2012, Cardona M et al. [39] first combined membrane computing with clustering algorithms. In 2016, Yan Huaning et al. [40] designed a new clustering algorithm by combining the PSO algorithm with the tissue—like P system. In 2020, Jiang Zhenni et al. [23] introduced a density peaks clustering algorithm based on the K-nearest Shannon entropy and tissue—like P system.

In 2021, Zhang Xiaoling et al. [41] proposed a spectral clustering method based on the coupling P system.

It is possible to improve the performance of DPC in some cases using the variants mentioned above. Nevertheless, as mentioned earlier, DPC still has shortcomings, such as the use of Euclidean distance, the manual selection of clustering centers and the chain reaction. To address the above drawbacks, we propose an improved density peaks clustering algorithm based on divergence distance and a tissue—like P system, and verify the clustering performance in this paper. From the perspective of clustering, the TP-DSDPC can be viewed as a novel variant of DPC. From the perspective of membrane computing, the TP-DSDPC is a new variant of TP systems that can process clustering problems. The following are the main contributions of our work:

- To eliminate the influence of Euclidean distance, a new distance metric—divergence distance—is proposed.
- To redefine the formula for calculating local density and relative distance, and automatically select the clustering centers according to the score value.
- The TP-DSDPC algorithm is integrated with a tissue—like P system, and optimizes the efficiency of the algorithm using P system’s computational parallelism in theory.
- The clustering performance of TP-DSDPC is simulated and verified on synthetic and UCI datasets.

The remainder of this study is structured as follows. Section 2 introduces the fundamental principles and steps of the DPC algorithm and relevant notions of the P system. Section 3 presents the TP-DSDPC algorithm at length. The performance of TP-DSDPC through experimental results and the analysis of datasets is demonstrated in Section 4. Conclusions are outlined in Section 5, along with suggestions for future investigations.

2. Related Work

In this section, we introduce the principle of density peak clustering and the tissue—like P system. For ease of understanding, we summarize major notations in Table 2.

Table 2. The main notations.

Notation	Description	Notation	Description
ρ_i	Density of point i	δ_i	The minimum distance from point i to point with higher density
d_{ij}	Euclidean distance between point i and point j	$Score_i$	Score value of point i
d_c	The cutoff distance	$cell_i$	The i th cell
$DV(x_i)$	Divergence of point i	R_i	Rule i
$DVdis(x_i, x_j)$	Divergence distance between point i and point j	$x(x_1, x_2, \dots, x_n)$	A dataset of n points

2.1. Density Peaks Clustering

As mentioned above, DPC is one of the most extensively used density-based clustering algorithms and is very adaptable. Two assumptions underlie DPC: clustering centers have the highest local density and are relatively far apart [42].

DPC mainly consists of [43]: firstly, local density ρ_i and relative distance δ_i of each point are calculated. It is then determined which points are the initial clustering centers, based on the decision graph. After that, the remaining points are allocated to their appropriate clustering centers.

In order to define the local density ρ_i of a data point, we use the following formula:

$$\rho_i = \sum_{j=1}^n \chi(d_{ij} - d_c) \tag{1}$$

where n represents the number of data points and d_{ij} represents the distance between the point i and the point j . d_c is the cutoff distance. χ refers to an indicator function, when $x < 0, \chi(x) = 1$, otherwise $\chi(x) = 0$.

Furthermore, the Gaussian kernel function can be used to calculate the local density of data points in addition to the abovementioned method. The specific formula is as follows:

$$\rho_i = \sum_{i \neq j} \exp \left\{ - \left(\frac{d_{ij}}{d_c} \right)^2 \right\} \tag{2}$$

The relative distance δ_i of a data point is calculated by the following formula:

$$\delta_i = \begin{cases} \max \delta & \rho_i = \max \rho \\ \min_{j, \rho_j > \rho_i} d_{ij} & \rho_i \neq \max \rho \end{cases} \tag{3}$$

Based on the above two variables, local density and relative distance, DPC constructs the decision graph with ρ as the abscissa and δ as the ordinate. The user needs to manually select clustering centers in the decision graph, or to select the clustering centers through the decision value γ , where $\gamma = \rho \times \delta$. After finding the clustering centers, those remaining points belong to a cluster larger than the local density of their nearest neighbors.

2.2. Tissue—Like P System

On the grounds of the cell-like P system, the tissue—like P system is proposed [44]. The cell-like P system only contains one cell, whereas the tissue—like P system contains multiple cells and the environment as well, and there are objects and rules in them. The movement of objects from cell to cell or cell to environment is carried out through rules in parallel execution [45]. The membrane structure of the tissue—like P system is demonstrated in Figure 1. The formal definition of the tissue—like P system of degree m ($m \geq 1$) is:

$$\Pi = O, \sigma_1, \dots, \sigma_n, syn, i_{out} \tag{4}$$

where

1. O is the alphabet, which contains all objects in the system.
2. syn are synapses that connect cells.
3. i_{out} represents the output cell of the system.
4. $\sigma_1, \dots, \sigma_n$ indicate n cells in the system, the detail definition is as follows: $\sigma_i = (Q_i, s_{i,0}, w_{i,0}, P_i), 1 \leq i \leq m$, Q_i refers to the collection of all states. $s_{i,0} \in Q_i$ refers to the initial state. $w_{i,0} \in O^*$ means the initial multiset of the object, when $w_{i,0} = \lambda$ there is no object in cell i . P_i stands for the rules of the system.

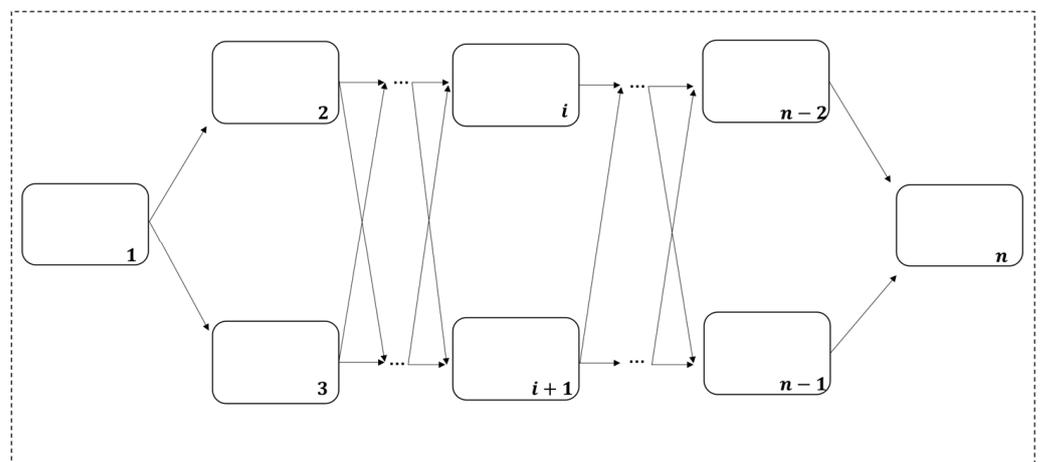


Figure 1. The basic membrane structure of the tissue—like P system.

3. The Proposed Method

An improved density peaks clustering algorithm based on divergence distance and the tissue—like P system is presented in this section. In the beginning, the divergence distance is introduced into the DPC algorithm, and clustering centers are automatically selected by score values. The whole algorithm is carried out within the structure of the tissue—like P system. Then, distinct rules and operations are respectively presented. Finally, algorithm complexity analysis is performed.

3.1. Divergence Distance

To address the influence of the above shortcomings, based on traditional Euclidean distance, the divergence distance is refined. Inspired by the importance ranking of data points, the definitions of divergence and divergence distance are shown as follows [46].

Definition 1. (Divergence). For a dataset D containing n points, where m points $(x_1, x_2, \dots, x_i, \dots, x_m)$ appear in the same circle. Point x_i can be represented by the Euclidean distance from point x_k ($1 \leq k \neq i \leq m < n$) to it. The divergence of x_i , denoted as $DV(x_i)$, is computed by:

$$DV(x_i) = \frac{E_u(x_1, x_i) + E_u(x_2, x_i) + \dots + E_u(x_m, x_i)}{\theta \times d} = \frac{\sum_{k=1, k \neq i}^m E_u(x_k, x_i)}{\theta \times d} \tag{5}$$

where $E_u(x_k, x_i)$ is the Euclidean distance between x_k and x_i ; the diameter of the circle where these points are located is d , the amount of accumulation is θ , $\theta = m - 1$. $E_u(x_k, x_i)$ ranges between 0 and d , so there is a range of 0 to 1 for $DV(x_i)$.

Definition 2. (Divergence distance). The divergence distance between point x_i and point x_j under the divergence defined in Definition 1 (tagged as $DVdis(x_i, x_j)$) is:

$$DVdis(x_i, x_j) = E_u(x_i, x_j) - e^{-(DV(x_i) - DV(x_j))^2} \tag{6}$$

According to Definition 2, there are two components to the divergence distance between points x_i and x_j . There are two parts to the distance between them: the Euclidean distance and the divergence difference.

3.2. Basic Principle of TP-DSDPC

In our algorithm, we first apply the divergence distance into the calculation formula of local density and relative distance, which is redefined as follows:

$$\rho_i = \sum_{i \neq j} \exp \left[-\frac{DVdis(x_i, x_j)}{d_c} \right]^2 \tag{7}$$

$$\delta_i = \begin{cases} \max \delta & \rho_i = \max \rho \\ \min_{j, \rho_j > \rho_i} DVdis(x_i, x_j) & \rho_i \neq \max \rho \end{cases} \tag{8}$$

Once the density and distance between each point have been calculated, the original DPC manually decides the clustering centers by drawing a decision graph, but this is not suitable for complex datasets and can easily cause incorrect clustering. Therefore, we automatically select the clustering centers by score value [26]. Here is the equation for calculating the score.

$$Score_i = \left(\frac{\rho_i}{\max(\rho)} \right)^2 \times \frac{\delta_i}{\max(\delta)} \tag{9}$$

According to this equation, points with high local density and distance are given a high value. By sorting the score value of each point in the dataset in descending order, c points are identified as clustering centers (c is the number of clusters).

As part of the assignment procedure of the DPC algorithm, a new version of the aggregation strategy is applied in order to resolve the chain reaction. The new version allocates the remaining points to nearest neighbors redefined with divergence distance (with higher density), which avoids cascades of misassignments.

3.3. The Initial Configuration of the Tissue—Like P System

To enhance the algorithm's computational efficiency, the tissue—like P system is combined with it. The initial configuration, which is shown in Figure 2 for each cell and rule, is as follows.

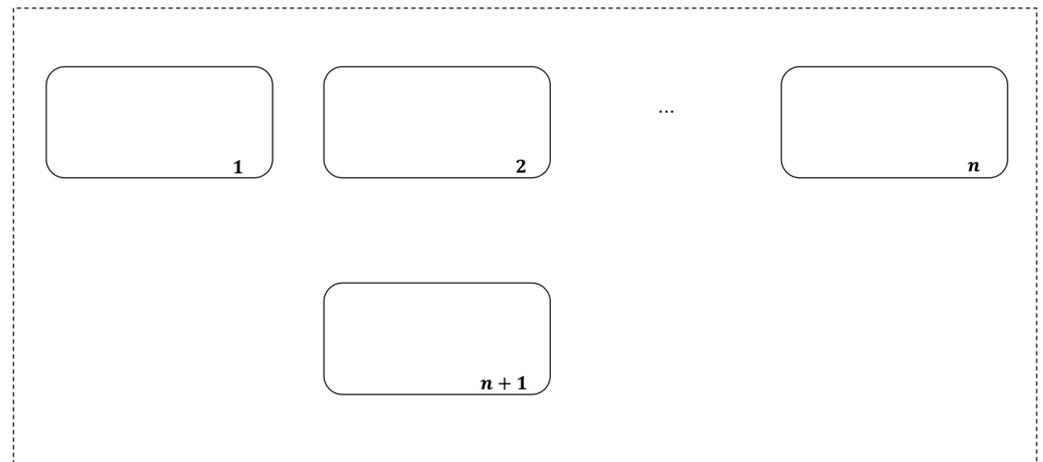


Figure 2. The initial configuration of the tissue—like P system.

- $cell_1, \dots, cell_n$: They represent n sample points in the dataset.
- $cell_{n+1}$: This is an empty cell.
- R_1 : Each point's local density and relative distance are respectively calculated by formulas (7) and (8) and input to $cell_1, \dots, cell_n$.
- R_2 : Obtain the divergence distance between each pair of data points by rule R_1 , build a distance matrix M and send to $cell_1, \dots, cell_n$.
- R_3 : The *Score* value of each point is calculated by formula (9) and input to $cell_1, \dots, cell_n$.
- R_4 : According to the *Score* value, the c clustering centers are determined, and the $cell_{n+1}$ is split into c new cells, which are marked as $cell_{n+1}, cell_{n+2}, \dots, cell_{n+c}$. Cluster count is c .
- R_5 : The cells where the clustering centers identified according to the *Score* value are located are fused with $cell_{n+1}, cell_{n+2}, \dots, cell_{n+c}$, respectively.
- R_6 : Allocation of the remaining points pursuant to matrix M , while the fusion of the intercellular membrane is performed.

3.4. The Process of TP-DSDPC Algorithm

The whole algorithm has seven main steps. The workflow chart of the TP-DSDPC algorithm is shown in Figure 3. Figure 4 displays the realization process of the tissue—like P system.

- Input: the dataset $x (x_1, x_2, \dots, x_n)$ and d_c
- Output: the clustering result
- Step 1: In $cell_1, \dots, cell_n$, we first apply rule R_1 to calculate the local density and relative distance of each data point at the same time.
- Step 2: The rule R_2 is applied to obtain the distance matrix M in $cell_1, \dots, cell_n$.
- Step 3: In $cell_1, \dots, cell_n$, the rule R_3 is used to calculate the *Score* value of each point as a backup for selecting the clustering centers.
- Step 4: According to the number of clusters, use rule R_4 to split $cell_{n+1}$ into c new cells.

- Step 5: Membrane fusion of clustering centers—apply the rule R_5 to fuse the c clustering centers with $cell_{n+1}, cell_{n+2}, \dots, cell_{n+c}$, respectively.
- Step 6: In this step, rule R_6 is used to allocate the remaining points, except clustering centers.
- Step 7 (Termination of calculation): The calculation is terminated when all of the points are allocated; the clustering result is output.

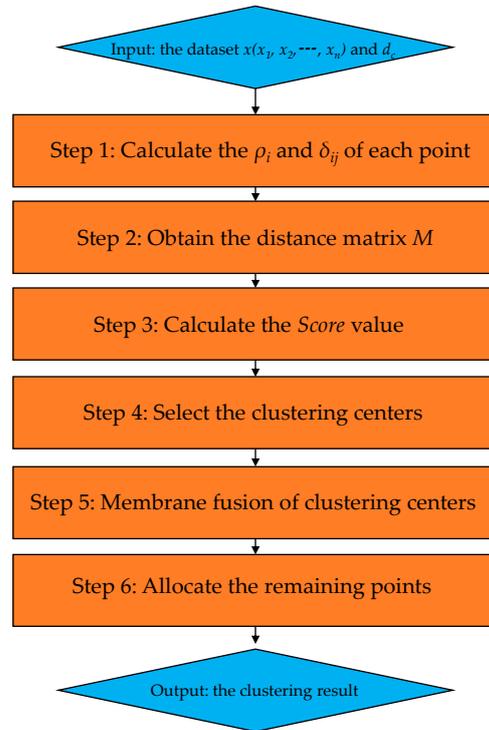


Figure 3. The workflow of the TP-DSDPC algorithm.

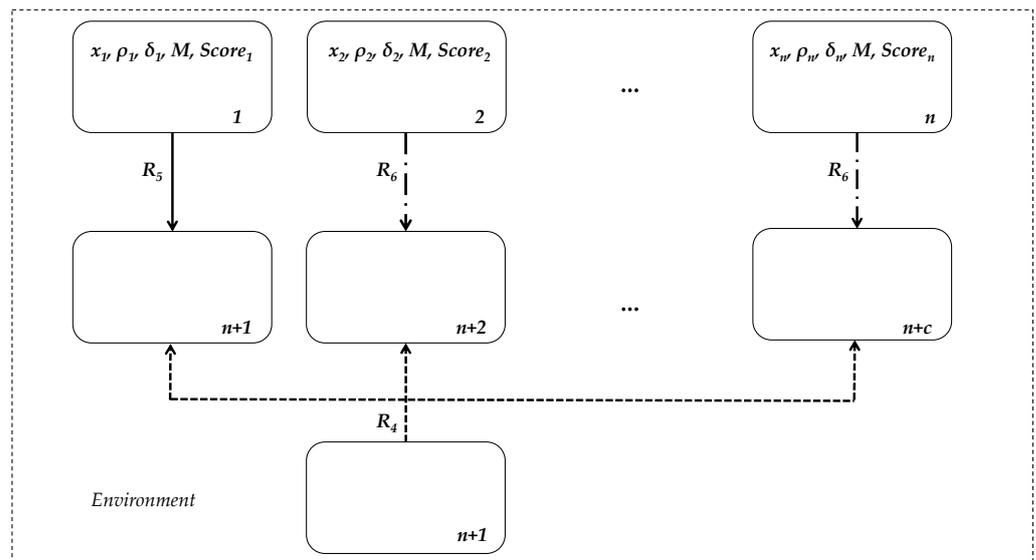


Figure 4. The realization process of the tissue-like P system.

3.5. Complexity Analysis

The sample size is set to n ; the dataset has c clusters. The time complexity of TP-DSDPC is composed of the following components: (i) the calculation of the divergence distance between all data points $O(n^2)$; (ii) density calculation for the points $O(n^2)$; (iii) calculating

the score value and selecting the highest scoring points $O(n \log n)$; (iv) each cluster is merged with its connect points $O(c^2)$; (v) all points are assigned to their anchor points' clusters $O(cn)$. Based on the preceding analysis, we can conclude that TP-DSDPC has a time complexity of $O(n^2)$, which is equivalent to that of DPC.

4. Experiment

In this section, to verify the performance of the proposed TP-DSDPC algorithm, we conduct experiments on both synthetic and real-world datasets. All of the experiments were organized in Matlab2016a on a laptop computer running Windows 10 with a 2.50 GHz CPU and 8G RAM. At the same time, we compared it with other clustering methods, including K-means [47], DBSCAN [48], DPC [19], KNN-DPC [21] and DGDPC [28]. Every method was repeated ten times with either the default parameter values given by the respective authors or tuned to achieve the best performance. After each run, the ACC, NMI and ARI indices of each algorithm were obtained, and the average result from several runs was used to determine final performance. The standard deviation (std) values are reported in brackets.

4.1. Evaluation Indicators

Usually, evaluation indicators are used to measure the quality of clustering results. We adopted three common clustering indicators: Accuracy (ACC) [49], Normalized Mutual Information (NMI) [50] and Adjusted Rand Index (ARI) [51]. It is believed that the clustering performance is better when ACC, NMI and ARI are higher. They were calculated as follows.

(1) Accuracy (ACC)

Based on the clustering label P and the true label T , accuracy represents how many samples have been correctly clustered compared to the total number of samples. There is a range of 0 to 1 for the ACC.

$$ACC = \frac{\sum_{i=1}^k \max |P_i \cap T_j|}{N}$$

(2) Normalized Mutual Information (NMI)

It uses information theory to measure the difference between the clustering partitions, and its range is in $[0, 1]$.

$$NMI(X, Y) = 2 \frac{I(X, Y)}{H(X) + H(Y)}$$

where $I(X, Y)$ is the mutual information between X and Y , and $H(X)$, $H(Y)$ are the entropy of random variables.

(3) Adjusted Rand Index (ARI)

The Rand Index indicates how closely the predicted and actual values match, and its range is $[0, 1]$. However, as RI does not guarantee that a random result is close to 0, we recommend using the Adjusted Rand Index, which has a higher degree of discrimination. Values are between $[-1, 1]$. Larger value indicates better consistency with reality.

$$RI = \frac{TP + TN}{TP + FP + TN + FN}$$

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]}$$

Decisions are represented as TP (true positive), FP (false positive), TN (true negative) and FN (false negative), respectively.

4.2. Experiments on Synthetic Datasets

In this subsection, the performances of K-means, DBSCAN, DPC, KNN-DPC, DGDPC and TP-DSDPC on nine synthetic datasets are reported. The nine synthetic datasets are listed in Table 3. Figure 5 displays the original data.

Table 3. The basic information of the nine synthetic datasets.

Dataset	Objects	Attributes	Clusters
Flame	240	2	2
Jain	373	2	2
Spiral	312	2	3
Threecircles	299	2	3
Smile	266	2	3
Fourlines	512	2	4
Aggregation	788	2	7
R15	600	2	15

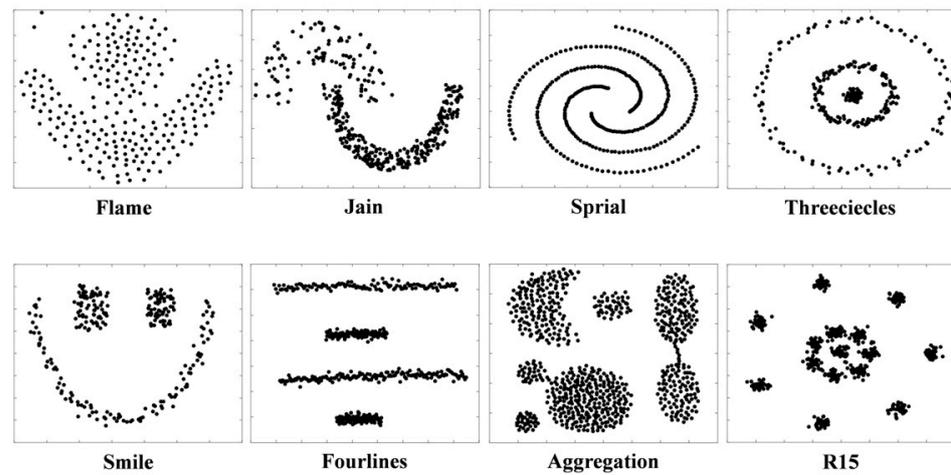


Figure 5. The original data of the nine synthetic datasets.

As shown in Figure 6, only the TP-DSDPC algorithm is completely effective for the Flame dataset. DPC and KNN-DPC cannot identify some noise points and boundary points. DGDPC does not recognize the junction points of two clusters. It can be seen from Figure 7 that TP-DSDPC is able to excellently cluster the Jain dataset. From the clustering results revealed in Figure 8, aside from K-means and the original DPC, others are processed well for clusters of the Spiral dataset.

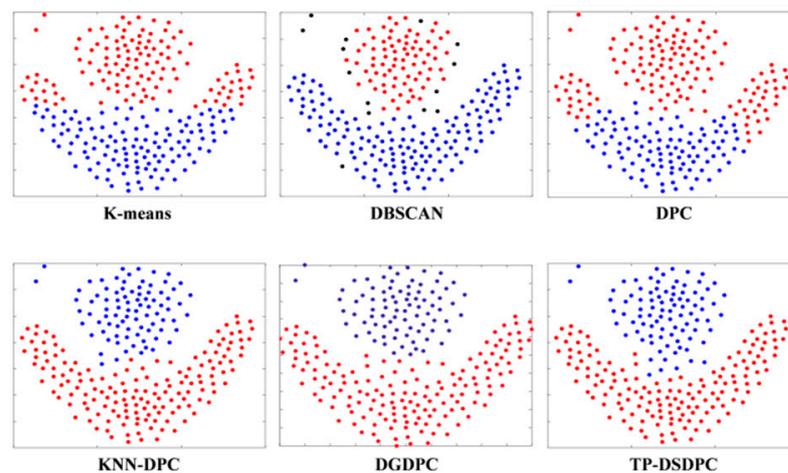


Figure 6. Clustering results on Flame.

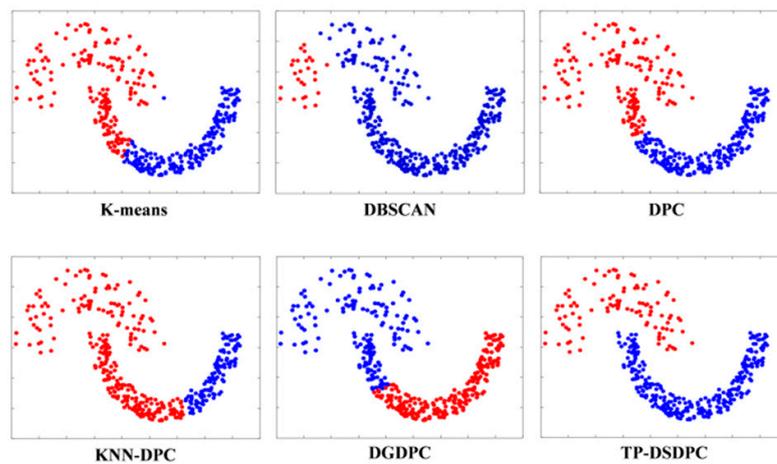


Figure 7. Clustering results on Jain.

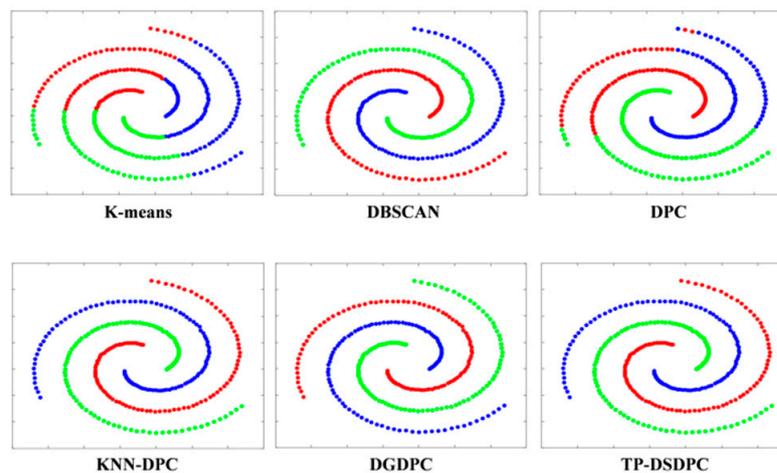


Figure 8. Clustering results on Spiral.

Only DBSCAN, DGDPC and TP-DSDPC had successful clustering results for the Three-circles dataset according to the clustering findings in Figure 9. This may be because other algorithms cannot recognize circular datasets. Figure 10 illustrates that most of the algorithms can perform well on the Smile dataset, except for K-means and DBSCAN. Figure 11 demonstrates that most algorithms can find proper clusters on the Fourlines dataset.

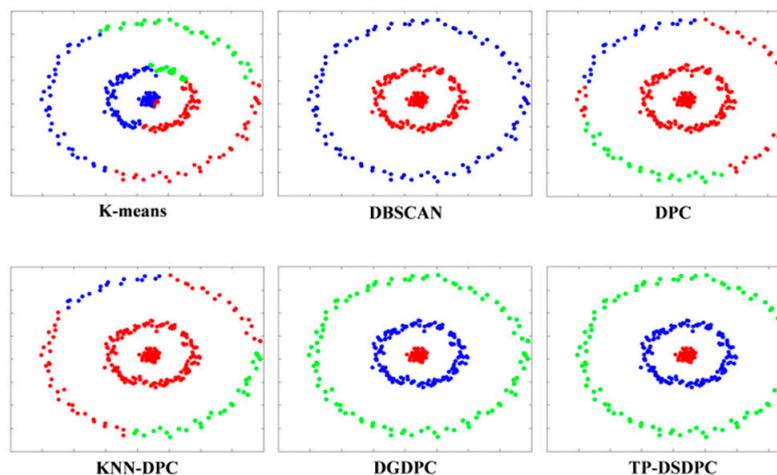


Figure 9. Clustering results on Threecircles.

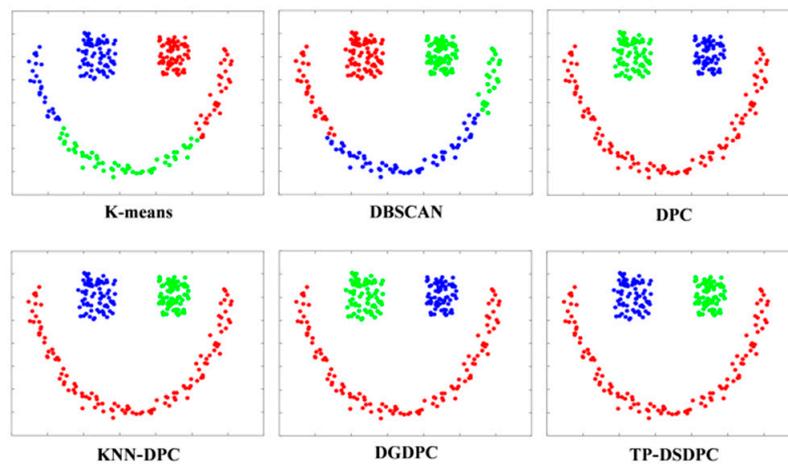


Figure 10. Clustering results on Smile.

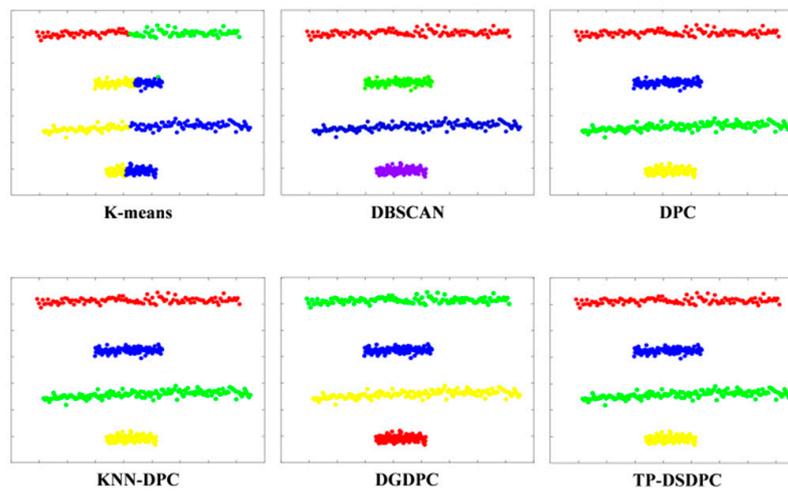


Figure 11. Clustering results on Fourlines.

The clustering results of the Aggregation dataset are displayed in Figure 12. The TP-DSDPC has the best clustering performance, while other algorithms are less efficient. As for the last R15 dataset, which is displayed in Figure 13, all of the algorithms can identify different clusters. However, K-means and DBSCAN have some clustering errors, because many normal points on the boundary of the clusters are also recognized as noise points.

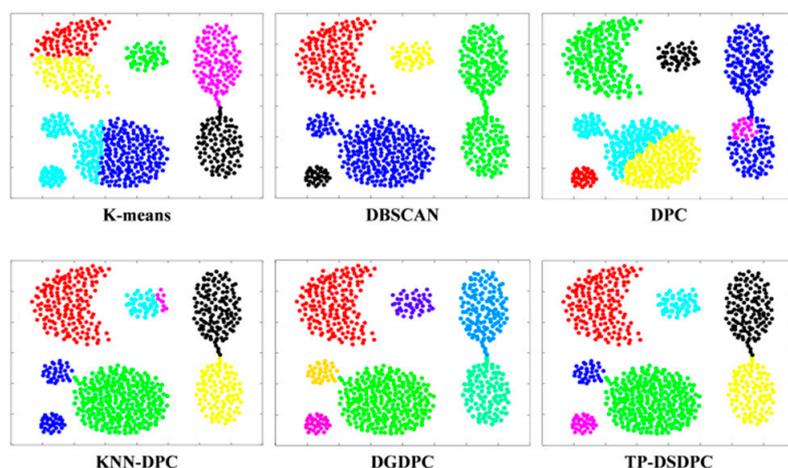


Figure 12. Clustering results on Aggregation.

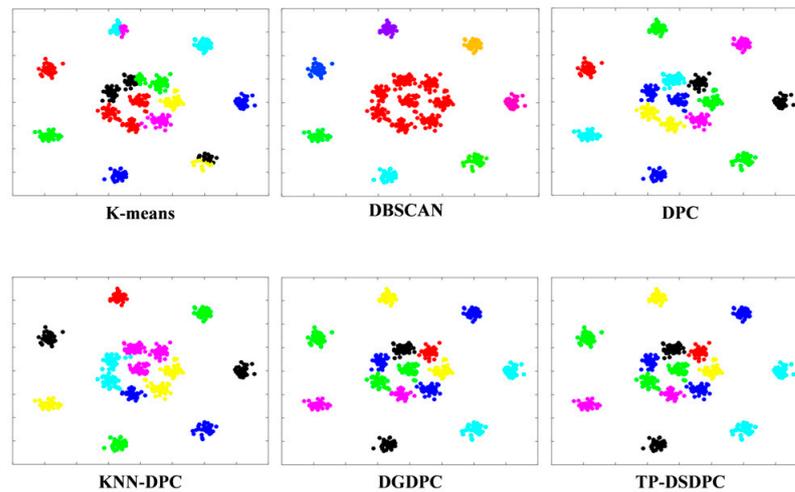


Figure 13. Clustering results on R15.

From Tables 4–6 (the bolded results are the best), the ACC, NMI and ARI of TP-DSDPC are separately the highest for all synthetic datasets, which shows its superiority on clustering datasets with complex structured clusters and non-uniform density.

Table 4. The ACC of the six algorithms on nine synthetic datasets.

Datasets	K-Means	DBSCAN	DPC	KNN-DPC	DGDPC	TP-DSDPC
Flame	0.6355	0.6749	0.8537	0.9613	0.9827	1.0000
	(0.0212)	(0.0253)	(0.0137)	(0.0158)	(0.0245)	(0.0000)
Jain	0.7019	0.7291	0.7955	0.5158	0.7025	1.0000
	(0.0479)	(0.0412)	(0.0392)	(0.0080)	(0.0136)	(0.0000)
Spiral	0.3308	1.0000	0.4473	1.0000	1.0000	1.0000
	(0.0277)	(0.0000)	(0.0330)	(0.0000)	(0.0000)	(0.0000)
Threecircles	0.3002	1.0000	0.3584	0.3002	1.0000	1.0000
	(0.0656)	(0.0000)	(0.0456)	(0.0235)	(0.0000)	(0.0000)
Smile	0.6359	0.6418	1.0000	1.0000	1.0000	1.0000
	(0.0330)	(0.0398)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
Fourlines	0.3085	1.0000	1.0000	1.0000	1.0000	1.0000
	(0.0448)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
Aggregation	0.5001	0.3349	0.7013	0.7758	1.0000	1.0000
	(0.0349)	(0.0999)	(0.0323)	(0.0255)	(0.0000)	(0.0000)
R15	0.7325	0.6718	0.9559	0.9684	0.9890	1.0000
	(0.0422)	(0.0547)	(0.0339)	(0.0157)	(0.0176)	(0.0000)

Table 5. The ACC of the six algorithms on nine synthetic datasets.

Datasets	K-Means	DBSCAN	DPC	KNN-DPC	DGDPC	TP-DSDPC
Flame	0.2038	0.3185	0.5602	0.7148	0.8411	1.0000
	(0.0133)	(0.0177)	(0.0142)	(0.0093)	(0.0387)	(0.0000)
Jain	0.2597	0.4829	0.5589	0.2216	0.3517	1.0000
	(0.0178)	(0.0120)	(0.0226)	(0.0061)	(0.0197)	(0.0000)
Spiral	0.1713	1.0000	0.2019	1.0000	1.0000	1.0000
	(0.0185)	(0.0000)	(0.0176)	(0.0000)	(0.0000)	(0.0000)
Threecircles	0.1569	0.9982	0.1753	0.1345	1.0000	1.0000
	(0.0149)	(0.0285)	(0.0119)	(0.0758)	(0.0000)	(0.0000)

Table 5. *Cont.*

Datasets	K-Means	DBSCAN	DPC	KNN-DPC	DGDPC	TP-DSDPC
Smile	0.5305 (0.0320)	0.5587 (0.0214)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Fourlines	0.1153 (0.0150)	0.9998 (0.2682)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Aggregation	0.3441 (0.0311)	0.2019 (0.0169)	0.4110 (0.0454)	0.4853 (0.0117)	1.0000 (0.0000)	1.0000 (0.0000)
R15	0.6259 (0.0543)	0.5487 (0.0227)	0.6449 (0.0186)	0.6571 (0.0209)	0.9448 (0.0197)	1.0000 (0.0000)

Table 6. The ARI of the six algorithms on nine synthetic datasets.

Datasets	K-Means	DBSCAN	DPC	KNN-DPC	DGDPC	TP-DSDPC
Flame	0.1904 (0.0281)	0.2718 (0.0110)	0.4205 (0.0112)	0.6251 (0.0246)	0.7619 (0.0415)	1.0000 (0.0000)
Jain	0.4759 (0.0297)	0.6014 (0.0157)	0.5813 (0.0046)	0.4206 (0.0096)	0.5381 (0.0429)	1.0000 (0.0000)
Spiral	0.1028 (0.0140)	1.0000 (0.0000)	0.2998 (0.0576)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Threecircles	0.1358 (0.0208)	0.9987 (0.0409)	0.2769 (0.0563)	0.1137 (0.0451)	1.0000 (0.0000)	1.0000 (0.0000)
Smile	0.6144 (0.0446)	0.6369 (0.0274)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Fourlines	0.2588 (0.0468)	0.9973 (0.0177)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Aggregation	0.1028 (0.0181)	0.0099 (0.0560)	0.3579 (0.0198)	0.4899 (0.0761)	1.0000 (0.0000)	1.0000 (0.0000)
R15	0.5108 (0.1235)	0.4236 (0.0361)	0.7003 (0.0285)	0.7140 (0.0419)	0.9047 (0.0526)	1.0000 (0.0000)

4.3. Experiments on Real-World Datasets

The UCI dataset is a widely recognized standard test dataset that is frequently used in clustering research. To further prove the performance of TP-DSDPC, we conducted experiments on six UCI datasets. Table 7 indicates the specific information of the datasets. The proposed algorithm is universally applicable to a wide variety of complex datasets, based on the results of the experiments.

Table 7. The basic information of the six UCI datasets.

Dataset	Objects	Attributes	Clusters
Iris	150	4	3
Ecoli	336	8	8
Zoo	101	18	7
Glass	214	10	6
Yeast	1484	9	10
Seeds	210	7	3

The results of ACC, NMI and ARI are separately presented in Tables 8–10. Bold is used to highlight the best outcomes.

Table 8. Results of ACC on UCI datasets.

Dataset	K-Means	DBSCAN	DPC	KNN-DPC	DGDPC	TP-DSDPC
Iris	0.6911 (0.0321)	0.6505 (0.0734)	0.7954 (0.0528)	0.8363 (0.0558)	0.8651 (0.0267)	0.8913 (0.0395)
Ecoli	0.6724 (0.0195)	0.5475 (0.0696)	0.4294 (0.0366)	0.4271 (0.0364)	0.6991 (0.0232)	0.6816 (0.0216)
Zoo	0.6817 (0.0363)	0.7018 (0.0451)	0.7529 (0.0583)	0.6803 (0.0398)	0.7284 (0.0345)	0.7748 (0.0264)
Glass	0.6458 (0.0976)	0.5154 (0.0381)	0.3943 (0.0176)	0.2625 (0.0422)	0.5692 (0.0463)	0.6958 (0.0205)
Yeast	0.2874 (0.0113)	0.3973 (0.0420)	0.3906 (0.0342)	0.4563 (0.0281)	0.4397 (0.0163)	0.6015 (0.0278)
Seeds	0.7525 (0.0272)	0.6373 (0.0267)	0.7735 (0.0314)	0.7964 (0.0161)	0.8569 (0.0232)	0.8711 (0.0380)

Table 9. Results of NMI on UCI datasets.

Dataset	K-Means	DBSCAN	DPC	KNN-DPC	DGDPC	TP-DSDPC
Iris	0.7437 (0.0229)	0.6955 (0.0343)	0.7346 (0.0247)	0.7978 (0.0164)	0.7809 (0.0200)	0.8103 (0.0231)
Ecoli	0.6444 (0.0401)	0.1421 (0.0183)	0.2289 (0.0223)	0.2181 (0.0348)	0.7094 (0.0402)	0.7285 (0.0199)
Zoo	0.7062 (0.0284)	0.7138 (0.0175)	0.7852 (0.0230)	0.7358 (0.0291)	0.7162 (0.0212)	0.7753 (0.0177)
Glass	0.7459 (0.0125)	0.1052 (0.0324)	0.4018 (0.0136)	0.4376 (0.0288)	0.6819 (0.0292)	0.8157 (0.0208)
Yeast	0.2436 (0.0167)	0.0295 (0.0329)	0.1224 (0.0287)	0.1375 (0.0205)	0.1226 (0.0197)	0.3649 (0.0231)
Seeds	0.5288 (0.0195)	0.3087 (0.0228)	0.5684 (0.0270)	0.6251 (0.0116)	0.7382 (0.0179)	0.7136 (0.0101)

Table 10. Results of ARI on UCI datasets.

Dataset	K-Means	DBSCAN	DPC	KNN-DPC	DGDPC	TP-DSDPC
Iris	0.6998 (0.0373)	0.5749 (0.0820)	0.6485 (0.0576)	0.7001 (0.0605)	0.7953 (0.0293)	0.7807 (0.0431)
Ecoli	0.5774 (0.0351)	0.3931 (0.0402)	0.2677 (0.0251)	0.2758 (0.0311)	0.6537 (0.0359)	0.7042 (0.0228)
Zoo	0.5636 (0.0408)	0.5796 (0.0520)	0.6441 (0.0513)	0.7015 (0.0629)	0.6561 (0.0436)	0.7549 (0.0373)
Glass	0.5459 (0.0148)	0.0137 (0.0463)	0.2297 (0.0198)	0.2017 (0.0480)	0.4307 (0.0530)	0.4473 (0.0347)
Yeast	0.1437 (0.0526)	0.0248 (0.0381)	0.0194 (0.0324)	0.0121 (0.0187)	0.0983 (0.0176)	0.3144 (0.0308)
Seeds	0.5886 (0.0250)	0.3319 (0.0310)	0.5992 (0.0399)	0.6129 (0.0167)	0.7764 (0.0256)	0.8013 (0.0354)

In order to more intuitively observe the ACC, NMI and ARI of six algorithms on these real-world datasets, histograms were applied to represent them, as can be seen separately from Figures 14–16.

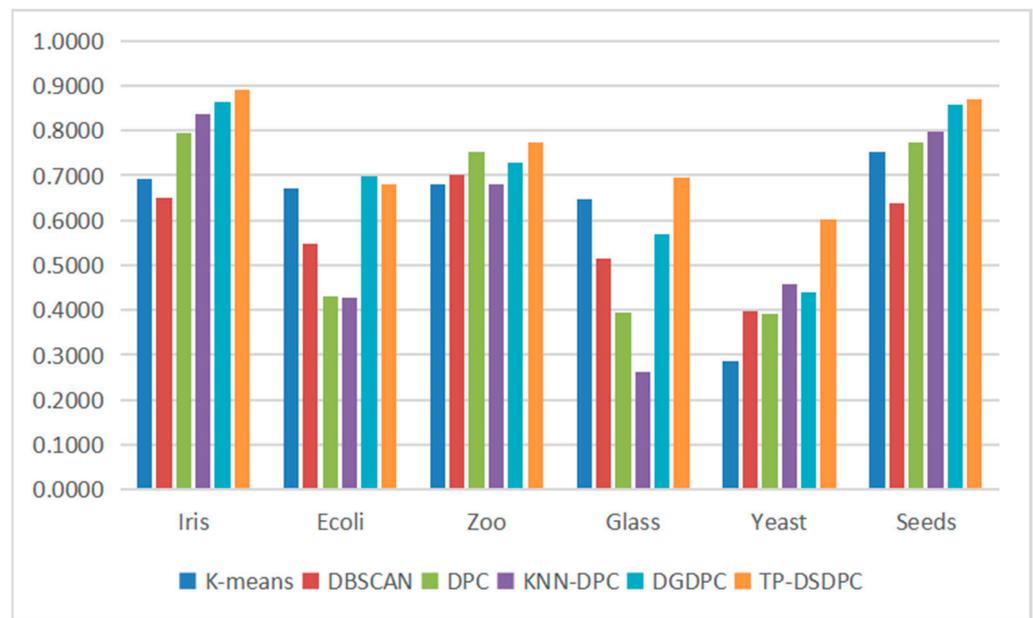


Figure 14. The ACC of the six algorithms on six real-world datasets.

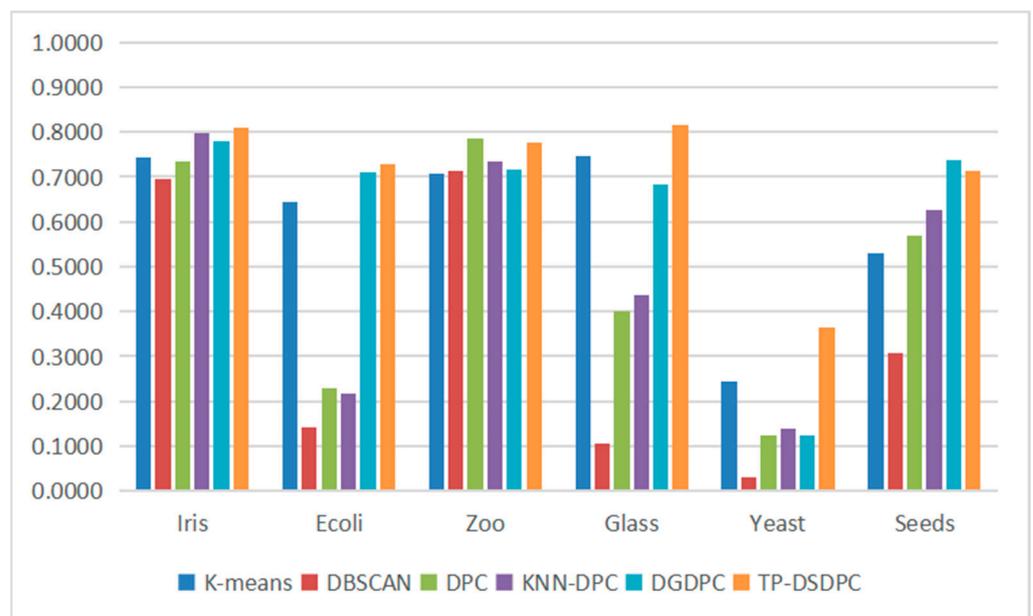


Figure 15. The NMI of the six algorithms on six real-world datasets.

Regarding the ACC, Figure 14 illustrates that the TP-DSDPC algorithm has the best results on the five real-world datasets, except for the Ecoli dataset. The results from the Ecoli dataset are also the second best. In terms of the NMI, Figure 15 demonstrates that the proposed algorithm achieved excellent results on most datasets. At the same time, it is only second to the DPC and DGDPC algorithms on the Zoo and Seeds datasets. In terms of the ARI, as displayed in Figure 16, the proposed algorithm obtained the best results on most datasets, except for the Iris and Glass datasets.

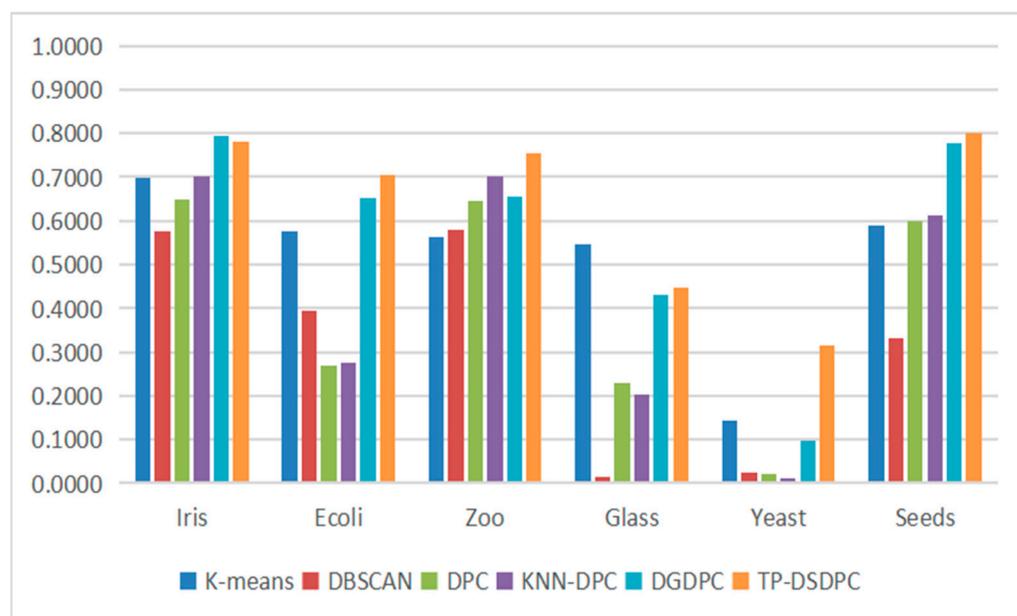


Figure 16. The ARI of the six algorithms on six real-world datasets.

4.4. Analysis of Experimental Results

As previously mentioned, the theory research and experiments on the synthetic and real-world datasets indicate that TP-DSDPC outperforms other algorithms. It obtained larger values of ACC, NMI and ARI than others. The TP-DSDPC algorithm can properly identify different types and scales of clusters. Moreover, it can be applied to more complicated situations. In this paper, all of the procedures of the algorithm were carried out in the structure of the tissue—like P system. Taking advantage of the extremely parallel computing characteristics of the tissue—like P system in membrane computing, it can improve efficiency to a certain extent in theory.

5. Conclusions

In this study, an improved density peaks clustering algorithm based on divergence distance and the tissue—like P system is proposed, termed TP-DSDPC. In the proposed algorithm, we introduce the divergence distance to relieve the impact of similarity measurements and the chain reaction. The main clustering process of TP-DSDPC is compatible with DPC. First of all, it calculates local density and updates relative distance by divergence distance. Then, it automatically selects the clustering centers by the score value from high to low. Lastly, the remaining points are clustered by the relative distance. Furthermore, the whole flow is implemented in the structure of the tissue—like P system and the efficiency of the algorithm is immensely promoted by the parallelism of the P system in theory. The proposed algorithm performs better on synthetic and real-world datasets, according to the experimental results.

In the years ahead, the proposed method will be expanded further so that it can address more clustering and optimization issues. Additionally, to improve the effectiveness and efficiency of TP-DSDPC, certain optimized methods should also be considered.

Author Contributions: Conceptualization, F.G. and X.L.; methodology, F.G. and X.L.; software, F.G.; validation, F.G.; Formal analysis, F.G.; Writing—original draft preparation, F.G.; writing—review and editing, F.G. and X.L.; supervision, F.G.; project administration, X.L.; Funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the National Natural Science Foundation of China (Nos. 62172262, 61876101, 61802234, and 61806114), China Postdoctoral Science Foundation Funded Project (2017M612339, 2018M642695), Natural Science Foundation of the Shandong Provincial (ZR2019QF007), China Post-doctoral Special Funding Project (2019T120607) and Youth Fund for Humanities and Social Sciences, Ministry of Education (19YJCZH244).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Contact the authors for the full datasets.

Conflicts of Interest: The authors of this paper declare no conflict of interest.

References

1. Li, F.X.; Zhou, M.; Li, S.; Yang, T.H. A New Density Peak Clustering Algorithm Based on Cluster Fusion Strategy. *IEEE Access* **2022**, *10*, 98034–98047. [\[CrossRef\]](#)
2. Wu, W.B.; Peng, M. A Data Mining Approach Combining K-Means Clustering with Bagging Neural Network for Short-Term Wind Power Forecasting. *IEEE Internet Things J.* **2017**, *4*, 979–986. [\[CrossRef\]](#)
3. Wang, Z.B.; Wang, E.; Zhu, Y. Image segmentation evaluation: A survey of methods. *Artif. Intell. Rev.* **2020**, *53*, 5637–5674. [\[CrossRef\]](#)
4. Zhang, X.Y.; Liu, C.L.; Suen, C.Y. Towards Robust Pattern Recognition: A Review. *Proc. IEEE* **2020**, *108*, 894–922. [\[CrossRef\]](#)
5. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [\[CrossRef\]](#)
6. Zhou, J.; Pedrycz, W.; Yue, X.D.; Gao, C.; Lai, Z.H.; Wan, J. Projected fuzzy C-means clustering with locality preservation. *Pattern Recognit.* **2021**, *113*, 107748. [\[CrossRef\]](#)
7. Bouguettaya, A.; Yu, Q.; Liu, X.M.; Zhou, X.M.; Song, A. Efficient agglomerative hierarchical clustering. *Expert Syst. Appl.* **2015**, *42*, 2785–2797. [\[CrossRef\]](#)
8. Guo, Y.N.; Yang, H.; Chen, M.R.; Go, D.W.; Cheng, S. Grid-based dynamic robust multi-objective brain storm optimization algorithm. *Soft Comput.* **2020**, *24*, 7395–7415. [\[CrossRef\]](#)
9. Xiang, S.J.; Yao, W.X. Semiparametric mixtures of regressions with single-index for model-based clustering. *Adv. Data. Anal. Classif.* **2020**, *14*, 261–292. [\[CrossRef\]](#)
10. Qin, H.C.; Li, R.H.; Wang, G.R.; Huang, X.; Yuan, Y.; Yu, J.X. Mining Stable Communities in Temporal Networks by Density-Based Clustering. *IEEE Trans. Big Data* **2022**, *8*, 671–684. [\[CrossRef\]](#)
11. Sinaga, K.P.; Yang, M.S. Unsupervised K-Means Clustering Algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [\[CrossRef\]](#)
12. Lorbeer, B.; Kosareva, A.; Deva, B.; Softic, D.; Ruppel, P.; Kupper, A. Variations on the Clustering Algorithm BIRCH. *Big Data Res.* **2018**, *11*, 44–53. [\[CrossRef\]](#)
13. Dat, N.D.; Phu, V.N.; Tran, V.T.N.; Chau, V.T.N.; Nguyen, T.A. STING Algorithm Used English Sentiment Classification in a Parallel Environment. *Int. J. Pattern Recogn.* **2017**, *31*, 1750021. [\[CrossRef\]](#)
14. Bai, Y.Z.; Chen, R.; Zhao, Y.; Wang, Y. Gaussian mixture model based adaptive control for uncertain nonlinear systems with complex state constraints. *Chin. J. Aeronaut.* **2022**, *35*, 361–373. [\[CrossRef\]](#)
15. Dong, B.; Weng, G.R.; Jin, R. Active contour model driven by Self Organizing Maps for image segmentation. *Expert Syst. Appl.* **2021**, *177*, 114948. [\[CrossRef\]](#)
16. Fu, H.P.; Li, H.; Dong, Y.Q.; Xu, F.; Chen, F.X. Segmenting Individual Tree from TLS Point Clouds Using Improved DBSCAN. *Forests* **2022**, *13*, 566. [\[CrossRef\]](#)
17. Tang, C.H.; Wang, H.; Wang, Z.W.; Zeng, X.K.; Yan, H.R.; Xiao, Y.J. An improved OPTICS clustering algorithm for discovering clusters with uneven densities. *Intell. Data Anal.* **2021**, *25*, 1453–1471. [\[CrossRef\]](#)
18. Stewart, G.; Al-Khassaweneh, M. An Implementation of the HDBSCAN* Clustering Algorithm. *Appl Sci-Basel.* **2022**, *12*, 2405. [\[CrossRef\]](#)
19. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [\[CrossRef\]](#)
20. Park, H. α -MeanShift ++: Improving MeanShift ++ for Image Segmentation. *IEEE Access* **2021**, *9*, 131430–131439. [\[CrossRef\]](#)
21. Du, M.J.; Ding, S.F.; Jia, H.J. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowl. Based Syst.* **2016**, *99*, 135–145. [\[CrossRef\]](#)
22. Liu, R.; Wang, H.; Yu, X.M. Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Inf. Sci.* **2018**, *450*, 200–226. [\[CrossRef\]](#)
23. Jiang, Z.; Liu, X.; Sun, M. A Density Peak Clustering Algorithm Based on the K-Nearest Shannon Entropy and Tissue-Like P System. *Math. Probl. Eng.* **2019**, *2019*, 1713801. [\[CrossRef\]](#)
24. Lin, J.L.; Kuo, J.C.; Chuang, H.W. Improving Density Peak Clustering by Automatic Peak Selection and Single Linkage Clustering. *Symmetry* **2020**, *12*, 1168. [\[CrossRef\]](#)
25. Xu, T.F.; Jiang, J.H. A Graph Adaptive Density Peaks Clustering algorithm for automatic centroid selection and effective aggregation. *Expert Syst. Appl.* **2022**, *195*, 116539. [\[CrossRef\]](#)

26. Lotfi, A.; Moradi, P.; Beigy, H. Density peaks clustering based on density backbone and fuzzy neighborhood. *Pattern Recognit.* **2020**, *107*, 107449. [[CrossRef](#)]
27. Seyedi, S.A.; Lotfi, A.; Moradi, P.; Qader, N.N. Dynamic graph-based label propagation for density peaks clustering. *Expert Syst. Appl.* **2018**, *115*, 314–328. [[CrossRef](#)]
28. Zhang, Z.Y.; Zhu, Q.S.; Zhu, F.; Li, J.N.; Cheng, D.D.; Liu, Y.; Luo, J.M. Density decay graph-based density peak clustering. *Knowl. Based Syst.* **2021**, *224*, 107075. [[CrossRef](#)]
29. Cheng, D.D.; Zhu, Q.S.; Huang, J.L.; Wu, Q.W.; Yang, L.J. Clustering with Local Density Peaks-Based Minimum Spanning Tree. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 374–387. [[CrossRef](#)]
30. Song, H.P.; Huang, Y.R.; Song, Q.; Han, T.; Xu, S.Y. Feature selection algorithm based on P systems. *Nat. Comput.* **2022**. [[CrossRef](#)]
31. Paun, G. Computing with membranes. *J. Comput. Syst. Sci.* **2000**, *61*, 108–143. [[CrossRef](#)]
32. Liu, Q.; Long, L.F.; Yang, Q.; Peng, H.; Wang, J.; Luo, X.H. LSTM-SNP: A long short-term memory model inspired from spiking neural P systems. *Knowl. Based Syst.* **2022**, *235*, 107656. [[CrossRef](#)]
33. Dong, J.P.; Zhang, G.X.; Luo, B.; Yang, Q.; Guo, D.Q.; Rong, H.N.; Zhu, M.; Zhou, K. A distributed adaptive optimization spiking neural P system for approximately solving combinatorial optimization problems. *Inf. Sci.* **2022**, *596*, 1–14. [[CrossRef](#)]
34. Yin, X.; Liu, X.Y.; Sun, M.H.; Ren, Q.Q. Novel Numerical Spiking Neural P Systems with a Variable Consumption Strategy. *Process.* **2021**, *9*, 549. [[CrossRef](#)]
35. Cai, Y.L.; Mi, S.H.; Yan, J.H.; Peng, H.; Luo, X.H.; Yang, Q.; Wang, J. An unsupervised segmentation method based on dynamic threshold neural P systems for color images. *Inf. Sci.* **2022**, *587*, 473–484. [[CrossRef](#)]
36. Chen, Y.H.; Chen, Y.; Zhang, G.X.; Paul, P.; Wu, T.B.; Zhang, X.H.; Rong, H.N.; Ma, X.M. A Survey of Learning Spiking Neural P Systems and A Novel Instance. *Int. J. Unconv. Comput.* **2021**, *16*, 173–200.
37. Jiang, Z.N.; Liu, X.Y. Novel coupled DP system for fuzzy C-means clustering and image segmentation. *Appl. Intell.* **2020**, *50*, 4378–4393. [[CrossRef](#)]
38. Zhang, G.X.; Gheorghe, M.; Pan, L.Q.; Perez-Jimenez, M.J. Evolutionary membrane computing: A comprehensive survey and new results. *Inf. Sci.* **2014**, *279*, 528–551. [[CrossRef](#)]
39. Cardona, M.; Colomer, M.A.; Zaragoza, A.; Perez-Jimenez, M.J. Hierarchical clustering with membrane computing. *Comput. Inform.* **2008**, *27*, 497–513.
40. Peng, H.; Luo, X.H.; Gao, Z.S.; Wang, J.; Pei, Z. A novel clustering algorithm inspired by membrane computing. *Sci. World J.* **2015**, *2015*, 929471. [[CrossRef](#)]
41. Zhang, X.L.; Liu, X.Y. Noises Cutting and Natural Neighbors Spectral Clustering Based on Coupling P System. *Process.* **2021**, *9*, 439. [[CrossRef](#)]
42. Wang, S.L.; Li, Q.; Zhao, C.F.; Zhu, X.Q.; Yuan, H.N.; Dai, T.R. Extreme clustering—A clustering method via density extreme points. *Inf. Sci.* **2020**, *542*, 24–39. [[CrossRef](#)]
43. Bian, Z.K.; Chung, F.L.; Wang, S.T. Fuzzy Density Peaks Clustering. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 1725–1738. [[CrossRef](#)]
44. Zhao, Y.Z.; Zhang, W.N.; Sun, M.H.; Liu, X.Y. An Improved Consensus Clustering Algorithm Based on Cell-Like P Systems with Multi-Catalysts. *IEEE Access* **2020**, *8*, 154502–154517. [[CrossRef](#)]
45. Liu, X.Y.; Zhao, Y.Z.; Sun, W.X. Tissue P Systems with Cooperating Rules. *Chin. J. Electron.* **2018**, *27*, 324–333. [[CrossRef](#)]
46. Yang, Y.Q.; Cai, J.H.; Yang, H.F.; Zhao, X.J. Density clustering with divergence distance and automatic center selection. *Inf. Sci.* **2022**, *596*, 414–438. [[CrossRef](#)]
47. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematical Statistics and Probability*; University of California: Los Angeles, LA, USA, 1967.
48. Ester, M.; Kriegel, H.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. KDD.* **1996**, *96*, 226–231.
49. Guo, W.J.; Wang, W.H.; Zhao, S.P.; Niu, Y.L.; Zhang, Z.Y.; Liu, X.G. Density Peak Clustering with connectivity estimation. *Knowl. Based Syst.* **2022**, *243*, 108501. [[CrossRef](#)]
50. Tao, X.M.; Guo, W.J.; Ren, C.; Li, Q.; He, Q.; Liu, R.; Zou, J.R. Density peak clustering using global and local consistency adjustable manifold distance. *Inf. Sci.* **2021**, *577*, 769–804. [[CrossRef](#)]
51. Cheng, D.D.; Zhang, S.L.; Huang, J.L. Dense members of local cores-based density peaks clustering algorithm. *Knowl. Based Syst.* **2020**, *193*, 105454. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.