Pattern Analysis and Applications (2019) 22:649–667
https://doi.org/10.1007/s10044-017-0673-0

THEORETICAL ADVANCES

DK-means: a deterministic K-means clustering algorithm for gene expression analysis

R. Jothi[1] · Sraban Kumar Mohanty[2] · Aparajita Ojha[2]

Received: 3 March 2017 / Accepted: 8 December 2017 / Published online: 28 December 2017

Abstract

Clustering has been widely applied in interpreting the underlying patterns in microarray gene expression profiles, and many clustering algorithms have been devised for the same. K-means is one of the popular algorithms for gene data clustering due to its simplicity and computational efficiency. But, K-means algorithm is highly sensitive to the choice of initial cluster centers. Thus, the algorithm easily gets trapped with local optimum if the initial centers are chosen randomly. This paper proposes a deterministic initialization algorithm for K-means (DK-means) by exploring a set of probable centers through a constrained bi-partitioning approach. The proposed algorithm is compared with classical K-means with random initialization and improved K-means variants such as K-means++ and MinMax algorithms. It is also compared with three deterministic initialization methods. Experimental analysis on gene expression datasets demonstrates that DK-means achieves improved results in terms of faster and stable convergence, and better cluster quality as compared to other algorithms.

Keywords K-means clustering algorithm · Initial cluster centers · Gene expression clustering · Microarray data analysis

1 Introduction

Microarray technology has become popular in monitoring the expression of thousands of genes simultaneously. Due to tremendous increase in high-throughput data generated by the microarray experiments, analyzing and extracting useful information out of such high voluminous data has become one of the recent research problems in bioinformatics. In particular, grouping the related genes which exhibit similar expression patterns (co-expressed genes) is a fundamental task in microarray analysis as it helps in identifying the set of genes involved in the same biological process.

Clustering is an unsupervised learning technique that is used extensively in microarray analysis for identification of co-expressed genes [4, 20, 21, 23, 42]. Given a set of genes, a clustering algorithm divides them into a number of distinct clusters according to certain similarity measure. Each such cluster corresponds to a particular macroscopic phenotype, such as clinical syndromes or cancer types [20]. A good clustering algorithm should identify a set of clusters such that genes within a cluster possess high similarity as compared to the genes in different clusters.

There are numerous clustering algorithms for gene expression analysis, and these algorithms can be broadly categorized into hierarchical and partitional algorithms [20, 23]. Hierarchical algorithms generate a nested grouping of objects in the form of a dendrogram tree, which allows the user to cut the tree at any level to obtain desired cluster structure. These algorithms are very useful in identifying inherent nesting relationship in the genes, e.g., disease types and subtypes. But their quadratic runtime complexity poses a major problem for large datasets [20].

In contrast to hierarchical algorithms, partitional algorithms directly divide the given set of objects into k groups without imposing the hierarchical structure, where k is the number of clusters which is *a priori* [19, 42]. K-means is a well-known example of partitional clustering algorithm

✉ R. Jothi
r.jothi@sot.pdpu.ac.in

Sraban Kumar Mohanty
sraban@iiitdmj.ac.in

Aparajita Ojha
aojha@iiitdmj.ac.in

[1] Department of Computer Engineering, School of Technology, Pandit Deendayal Petroleum University, Gandhinagar, India

[2] Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Jabalpur, Madhya Pradesh, India

which obtains a k-partition of objects through an iterative refinement such that the sum of squared distances between the objects and their cluster centers is minimized. Starting with k randomly chosen initial centers, K-means assigns each object to a cluster having its nearest center. Then, cluster centers are recomputed using the current membership of the objects. Again the objects are reassigned to the nearest centers. This process repeats until there is no change in the cluster centers. Due to its simple implementation and high performance, K-means algorithm still remains as a popular method for gene expression analysis. References [11, 25–27, 30–32, 35, 37] evidence the extensive use of K-means and its variants in gene expression clustering.

In spite of its wide use, K-means algorithm has certain drawbacks. It is highly sensitive to the choice of initial centers. If the initial centers are randomly chosen, then the algorithm may converge to a local optimum which leads to unstable clustering results [19, 20, 38, 42]. Another major drawback of K-means is that it is not robust to noise and outliers. Several attempts have been made to overcome the drawbacks of K-means and a comparative study of which can be seen in [8]. It is worthwhile to note here that most of the cluster initialization methods run in quadratic time which may degrade the performance of K-means, especially when applied on high-throughput microarray datasets [8]. Moreover, a few initialization methods suffer from the problem of unstable clustering results [3].

This paper proposes an initialization algorithm for K-means named as deterministic K-means (DK-means). DK-means employs a two-step process for cluster center initialization. First, a set of probable centers are identified using recursive binary partitioning of the dataset. Then, the initial centers for K-means algorithm are determined by applying a minimum spanning tree-based clustering on the probable centers. Once the initial centers are identified, we use the iterative cluster assignment procedure as in K-means algorithm. We show that complexity of the proposed algorithm is $O(n \log n)$ time. The proposed algorithm is compared against K-means with random initialization and some of its variants such as K-means++ [3], MinMax K-means [40]. Three deterministic initialization algorithms such as Var-part [39], Histogram-based discriminate analysis [7] and density-based cluster representative identification for non-metric spaces (DBCRIMES) [4] approaches have also been considered for comparative study. Experimental analysis on both artificial and real gene expression datasets reveals that the proposed algorithm DK-means achieves reduced number of iterations, stable clustering results and improved cluster quality in terms of internal as well as external cluster validity measures as compared to other algorithms. Biological significance of the clusters obtained by the proposed algorithm is also investigated.

The rest of the paper is organized as follows. In Sect. 2, a brief description of K-means algorithm and an overview of existing cluster center initialization methods are presented. The proposed algorithm DK-means is described in Sect. 3. Complexity of the proposed algorithm is discussed in Sect. 4. Results of experimental validation are reported and discussed in Sect. 5, and finally, conclusion and future scope are given in Sect. 6.

2 Related work

2.1 K-means algorithm

Given a set of objects $X = \{x_1, x_2, \ldots, x_n\}$, the objective of K-means algorithm is to obtain a set of k clusters $S = \{S_1, S_2, \ldots S_k\}$ such that mean squared error (MSE) criterion is minimized. MSE is defined as follows [19].

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{k} \sum_{x_j \in S_i} d(x_j, \mu_i)^2. \tag{1}$$

where $d(.,.)$ represents the distance and μ_i is the center of cluster S_i. It can be seen that a complete enumeration of all possible clusterings to determine the global minimum of objective function given in Eq. (1) is NP-hard problem [8, 18]. Various approximate heuristics have been studied in the literature, and among these, Lloyd's algorithm is the most commonly used one [8], which is often referred to as K-means. Starting with k randomly chosen initial centers, the algorithm iteratively assigns the objects to the nearest centers. The algorithm stops when there is no change in the cluster centers. Steps involved in K-means are given in "Algorithm 1."

```
Algorithm 1  K-means Algorithm [19]
Input: Dataset $X$.
Output: $k$ clusters of $X$.
1. Randomly choose $k$ objects in $X$ as initial centers $\mu_i$,
   $1 \le i \le k$.
2. For each object $x_j$ in $X$, find the nearest center $\mu_i$ and
   assign object $x_j$ to cluster $S_i$.
3. Recompute centers based on assignment of objects.
4. Repeat steps 2 and 3 if there is change in centers; else
   stop.
```

Complexity of the K-means algorithm is $O(nkfI)$ time, where n is the number of objects in X, f is the number of features (dimensions) of each object x_i, k is the number of clusters and I is the number of iterations taken by the algorithm for convergence. Usually the parameters f, k and

I are much smaller than n, so complexity of the algorithm becomes as $O(n)$ [42].

Getting trapped with local optimum and sensitivity to noise and outliers are the major drawbacks of K-means algorithm [38]. Random selection of initial cluster centers may lead to local optimum results. If the initial centers for K-means partitions are chosen randomly, there are chances that two or more centers are located in close proximity. As a consequence, the objects are forced to be assigned to one of the nearest centers, leading to undesired cluster structure (See Fig. 1).

Although more sophisticated clustering algorithms have been proposed in the recent years, K-means algorithm remains as a popular choice for gene expression analysis due to its simple implementation and linear time complexity [18]. It has been shown that K-means algorithm has good exploitation capability which enables fast clustering of the gene expressions [23, 26, 38].

2.2 Initialization methods

Many cluster center initialization approaches have been proposed in the literature, some of which are discussed briefly here. Bradley and Fayyad [5] proposed a sampling-based seed selection algorithm for K-means. Several samples are chosen from the given dataset and K-means is run on each of the samples independently with random centers. Resulting centroids from each sample give a potential guess of the centers for the whole dataset. Global K-means algorithm proposed by Likas et al. employed an incremental approach to choose k cluster centers, one at a time [28]. Experimental results demonstrated that global K-means outperforms the classical K-means algorithm. However, the amount of computational overhead caused by the incremental procedure of the algorithm makes it slower as compared to classical K-means algorithm [8].

An enhanced version of K-means known as K-means++ was proposed by Arthur and Vassilvitskii [3]. K-means++ also applies incremental approach in choosing the centers one at a time. First center c_1 is chosen randomly from the dataset. Remaining centers c_i, $2 \leq i \leq k$, are chosen so that the distance between c_i and the previously chosen centers is maximum. Experimental results of K-means++ showed that it was faster than classical K-means. However, clustering results of K-means++ are not stable for different runs due to random choice of the first center.

Density-based initialization methods also exist in the literature. Khan and Ahmad [24] proposed cluster center initialization algorithm (CCIA) using density-based multi-scale data condensation (DBMSDC). CCIA is based on the fact that very similar points form the core of the cluster, and hence, identifying such core points using DBMSDC provides a basis for identifying actual cluster centers. However, this method is computationally expensive due to the overhead involved with density calculation.

Many researchers have proposed cluster initialization methods by considering the principal dimensions for splitting the dataset. (See, for example, [2, 10, 13, 39].) Ting and Jennifer [39] devised two divisive hierarchical approaches based on principal component analysis (PCA). The first method named as PCA-part method carries out hierarchical splitting along a hyperplane that passes through center of the dataset and orthogonal to the first eigenvector of the covariance matrix. The second method named as Var-part method proceeds similar to PCA-part method except that the splitting hyperplane is considered to be orthogonal to the axis which has the maximum variance. Let d_i be the dimension chosen for splitting by PCA-part (or Var-part) method. Once the points are projected to axis d_i, they are partitioned into two clusters based on the threshold which is computed from the mean point of axis d_i. After k-splits, this procedure results in k clusters, and

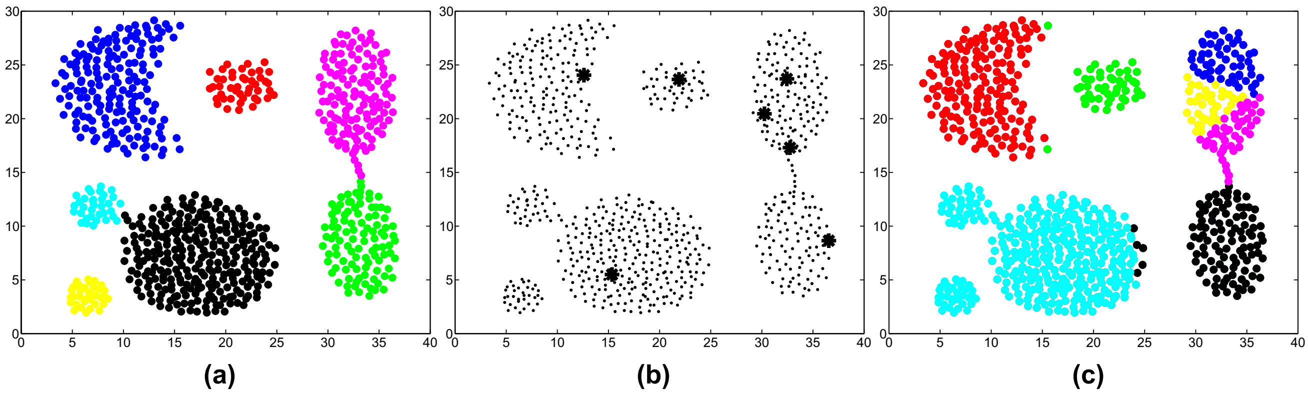


Fig. 1 K-means converging to a local optimum with random center initialization: **a** a given dataset with seven clusters. **b** Randomly chosen initial centers. **c** Result of K-means with centers chosen in **b**

the center of each of these clusters corresponds to actual centers for K-means.

Murat Erisoglu [13] proposed an initialization method by identifying the subspace of the dataset X along two main dimensions that best represent the spread of the dataset. First center is chosen as the data point with the longest distance from the centroid of X in the subspace. Remaining centers are chosen such that first center c_i has the maximum distance from the previously computed centers $c_1, \ldots, c_{i-1}$ as in K-means++ [3].

Tzortzis and Likas [40] have proposed a weighted version of K-means known as MinMax K-means. The MinMax algorithm assigns weights to the clusters in proportion to their variance and tries to control the emergence of clusters with large variance. Although MinMax attains high-quality partition, it incurs high computational cost due to the process employed for weight computation [40].

Numerous initialization methods have been studied in the past. Unfortunately many of these methods are not deterministic. Some researchers have worked on this aspect also. For example, references [4, 7, 12, 39] discuss on deterministic initialization methods. Celebi and Kingravi [7] proposed a Histogram-based deterministic initialization method, which is actually a modification of Var-part method of Ting and Jennifer [39]. As pointed out earlier, Var-part method recursively partitions the dataset based on an axis (dimension d_i) having larger variance and then mean point in axis d_i is considered to be the partitioning threshold. It is worthwhile to note here that the choice of mean point for thresholding may lead to poor results as observed in [7]. Celebi and Kingravi improved the results of Var-part method using Histogram modality as the partitioning threshold. Projection of the points on d_i is treated as a discrete probability distribution which is represented by a Histogram. Then the Histogram is divided into two clusters using between-class separability $BVar$ as the threshold; here, $BVar$ is taken as the Histogram modality. Next, a cluster having highest SSE is chosen for division using the above-said procedure. This is continued until required number of clusters is obtained and mean points of final clusters are taken to be the initial centers for K-means algorithm. The authors have reported that discriminant analysis using Histogram has improved the results as compared to Var-part method [7].

Filippo et al. [4] proposed an initialization algorithm, named as density-based cluster representative identification for non-metric spaces (DBCRIMES), that estimates the cluster centers using most representative patterns in an input dataset. The algorithm first constructs a probability density function $F(.)$ using parzen window, where $F(.)$ is an $n \times 1$ vector and each component F_i in $F(.)$ is the density value associated with the input pattern x_i. DBCRIMES has employed two approaches to extract most representative patterns from the estimated density function $F(.)$. The first one iteratively searches for elements with maximum value in the vector $F(.)$; once current maximum is chosen in a particular iteration, remaining values in $F(.)$ are radially decreased so that neighbors of current maximum are not given priority in the next iteration. The second approach ranks the data points according to their probability density values in $F(.)$ and selects the points having peak values, and they are located in the farthest proximity regions. Experimental analysis of DBCRIMES has shown that its results do not depend on the geometric shapes. However, quadratic complexity of DBCRIMES puts an overhead on the computational performance of K-means algorithm.

Spherical K-means clustering is a variant of K-means in which input data are considered to be a set of unit vectors that lie on the surface of unit hypersphere about the origin, and uses cosine similarity as its proximity measure [12]. Spherical K-means has gained popularity in document clustering. However, it inherits the problem of random initialization from standard K-means. Duwairi et al. [12] presented an initialization scheme for spherical K-means clustering and they demonstrated its use for document clustering. The algorithm employs an objective function based on the perturbation theory. Given input space is subdivided into a set of uniform intervals based on the weights present in the vectors. Then, the boundaries of these intervals suggest the set of initial means of clusters.

Various initialization approaches presented in the literature are generic in nature, and their effectiveness to high-dimensional datasets such as gene expression datasets is not known. Several attempts have been made to overcome the drawbacks of K-means for microarray analysis by integrating it with other computational paradigms such as genetic approach [25, 30, 31] and particle swarm optimization [11, 26, 27, 37]. These hybrid algorithms achieve improved results by integrating the enhanced exploring capability of evolutionary algorithms with the high performance of K-means.

Genetic algorithm has been used in conjunction with K-means (known as genetic K-means algorithm, GKA) to achieve robust clustering results [25]. Although GK-means algorithm converges to the global optimum, it involves high computational cost [23]. A fast version of GKA (FGKA) was proposed by Lu et al. [30], which suffered from initial center problem when the mutation probability is small. The author further extended FGKA using incremental cluster center selection approach known as incremental genetic k-means algorithm (IGKA) [31]. IGKA converges faster without the sensitivity to initial centers. It calculates the objective value total within-cluster variance (TWCV) and incrementally determine the cluster centers when the mutation probability is small after calculation of TWCV. Although IGKA finds the global optimum, the choice of genetic algorithm parameters (mutation probability rate, number of generations, size of the chromosome populations) influences the convergence of the algorithm.

An integration of particle swarm optimization (PSO) and K-means algorithm has proved to be efficient for solving clustering problem [11, 27, 37]. PK-means algorithm proposed by Du et al. [11] combines K-means with a variant of PSO known as particle-pair optimizer (PPO) to take advantage of the computational efficiency of K-means and the parallel search capability of PPO. Experimental results on yeast cell cycle, sporulation, lymphoma datasets have proved the efficiency of PK-means with respect to accuracy and less sensitivity to the initial randomly selected centers. A modified version of quantum-behaved particle swarm optimization (QPSO) algorithm, known as the multi-elitist QPSO (MEQPSO) model, was proposed by Sun et al. [37]. With one-step K-means operator, MEQPSO effectively accelerates the convergence speed of the algorithm. Recently, a cluster matching method is introduced by Lam et al. [27] to improve the performance of PSO-based K-means algorithm.

While there are a number of efficient methods to handle initial cluster center problem of K-means, significant increase in computation time is observed in using these methods, as compared to the use of classic K-means algorithm alone [3, 4, 11, 27, 28, 37, 39, 40]. Moreover, many of the existing initialization methods may become inappropriate for gene expression analysis due to the high dimensionality and highly interconnected nature of gene expression datasets. Also, clustering results of some of these methods turn out to be unstable. To overcome this problem, we propose a deterministic initialization method which obtains faster and stable convergence simultaneously.

3 Proposed deterministic K-means algorithm

The iterative procedure of K-means algorithm attempts to move an object x_j to a cluster S_i such that x_j is nearer to μ_i as compared to other centers μ_l, $i \neq l$. As membership of objects is decided based on closeness to a single object (center), it may not retrieve the actual shape of clusters [29]. With this motivation, the proposed deterministic K-means (DK-means) algorithm aims to capture m most likely objects such that the spread of each cluster in the dataset is well represented by a subset of these objects. We call these objects as *probable centers*.

DK-means algorithm involves a two-step process. In the first step, it identifies a set of probable centers by dividing the dataset X into m partitions, where $2 >> m >> k$. During the second step, the initial centers for K-means algorithm are determined by grouping the probable centers into k subsets, where center of each subset is considered as one of the k centers for K-means algorithm. A preliminary version of the work appears in a conference proceeding [22].

3.1 Recognizing probable centers

In order to recognize a set of probable centers, DK-means proposes a constrained bi-partitioning procedure. A general bi-partitioning recursively splits the dataset into a set of partitions according to certain objective function (e.g., minimize the intra-cluster variance) [9]. The resulting partitions are maintained in the form of binary tree. Bi-partitioning method starts from the root node that contains the given dataset X and splits the node into two subsets X_1 and X_2 based on certain partitioning criteria. The recursive splitting stops once required number of partitions are obtained. The proposed constrained bi-partitioning approach (CBA) follows a recursive splitting as in bi-partitioning, but it continues splitting as long as there are nodes to be split, whose size is greater than $\sqrt{n}$. Moreover, CBA chooses two farthest points from the node to be split in order to maximize inter-cluster variance.

CBA starts from the root node which contains all the objects in X. Let μ denote the center of X. Two centers μ_1 and μ_2 for bi-partitioning are chosen as follows.

$$\mu_1 = x : d(x_i, \mu) \leq d(x, \mu), \quad 1 \leq i \leq n, \tag{2}$$

$$\mu_2 = y : d(x_i, x) \leq d(y, x), \quad 1 \leq i \leq n. \tag{3}$$

The points μ_1 and μ_2 are the farthest pair of points in the node, choosing such points as centers maximize the separation between clusters. Once two centers are identified in the above manner, the node set X is split into two subsets X_1 and X_2 as follows.

$$X_1 = X_1 \cup \{x_i \in X \mid d(x_i, \mu_1) < d(x_i, \mu_2)\}, \tag{4}$$

$$X_2 = X_2 \cup \{x_i \in X \mid d(x_i, \mu_2) < d(x_i, \mu_1)\}. \tag{5}$$

The same splitting process is applied recursively on the subsets S_1 and S_2 as explained in the above manner. The CBA algorithm continues the splitting process as long as the size of the subset to be partitioned is greater than $\sqrt{n}$. Thus, the number of partitions m is not preset. Leaf nodes, the subsets which cannot be partitioned further, are stored in a set $S = \{S_1, S_2, \ldots, S_m\}$. Center of each partition S_i in S is considered to be a probable center y_i. Figure 2 shows an example of a dataset and its probable centers. It is obvious from the figure that each cluster is covered by a subset of probable centers.

3.2 Computation of initial centers for K-means

Let $Y = \{y_1, y_2, \ldots, y_m\}$ be the set of probable centers constituted from CBA. The next step is to identify k disjoint subsets of Y. The probable centers must be grouped such

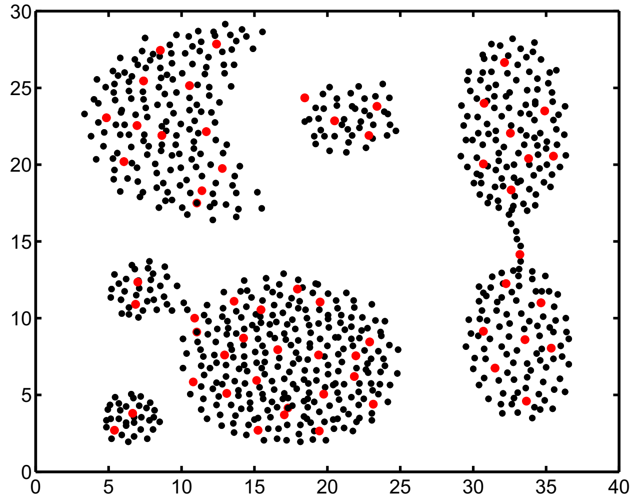


Fig. 2 Probable centers are marked in dots

that closeness between the centers within a subset is high as compared to closeness between the centers of different subsets. This in turn maximizes the inter-cluster separation of the final clusters produced by the K-means algorithm.

Algorithm 2 *Deterministic K-means (DK-means) Algorithm*

Input: Dataset X.
Output: k initial centers for K-means algorithm.

```
1  Let $S$ be the set of partitions and initialize $S = \phi$.
2  Initialize the node to be split $X' = X$.
3  Repeat
   3.1 if $|X'| > \sqrt{n}$
       3.1.1 Compute the center $\mu$ of the node $X'$.
       3.1.2 Choose two centers $x$ and $y$ using equations (2)
             and (3).
       3.1.3 Split the node set $X'$ into two subsets $S_1$ and $S_2$
             according to centers $x$ and $y$ using equations (4)
             and (5).
       3.1.4 Recursively apply splitting on $S_1$ and $S_2$.
   3.2 else $S = S \cup X'$
4  Until there is no node to split.
5  Identify a set of probable centers $Y = \{y_1, y_2, \ldots, y_{k'}\}$,
   where $y_i$ is the center of the subset $s_i$.
6  Build a set of best representative points ($R$) by choosing
   points closer to each probable center $y_i \in Y$.
7  Construct MST $T_1$ of $R$.
8  Identify and remove $k-1$ longest edges from $T_1$ to get $k$
   clusters.
9  Center from each of the $k$ clusters corresponds to actual
   center for K-means algorithm.
10 Apply K-means algorithm with identified centers.
```

Relative interconnectivity of the probable centers intuitively expresses neighboring nature of the subsets and a gap in the connectivity gives a clue on separation of the actual clusters in the dataset. In order to identify such gaps, minimum spanning tree (MST)-based representation of probable centers is employed, as the MST of a set of points can be used to reflect the similarity of the points with their neighborhood [43]. Simply removing $k-1$ longest edges from the MST results in k disjoint subsets of the nearest centers, such that each subset represents a cluster. This is illustrated in Fig. 3. Comparing the results of K-means with random initialization shown in Fig. 1c, the proposed initialization algorithm leads to relatively improved results as shown in Fig. 3d. As the probable centers are located at maximum distance apart, pruning longest edges from the center MST separate the noisy clusters from the actual clusters. Thus, the results are not affected by the presence of noise in the data.

As each $y_i \in Y$ may or may not belong to the dataset X, we choose a best representative point $r_i \in X$ from each S_i such that r_i is closest to y_i. Let $R = \{r_1, r_2, \ldots, r_m\}$ denotes the set of best representative points identified in the above manner. Prim's algorithm on R generates MST T_1, removing $k-1$ longest edges from T_1 yields k clusters of best representative points. Finally, the actual centers for K-means algorithm are computed from the center of k clusters. "Algorithm 2" describes the complete steps involved in proposed DK-means algorithm.

4 Complexity of proposed algorithm

Complexity of the proposed algorithm DK-means is analyzed as follows. Bi-means algorithm recursively partitions the dataset so that during every iteration, centers of the two partitions are maximum spaced apart. Any node in the Bi-means tree is divided recursively, if its size is greater than $\sqrt{n}$. Otherwise it becomes a leaf node, which corresponds to a partition S_i. Height of the Bi-means tree is $O(\log n)$. Thus, steps 1-4 of the DK-means algorithm take $O(n \log n)$ time to construct binary partitioning tree. Step 5 takes $O(n)$ time to identify the probable centers from each partition. Similarly $O(n)$ time is needed to find the best representative point set R in step 6. As size of the set R is $O(\sqrt{n})$, Prim's algorithm in step 7 takes $O(n)$ time. Obtaining k partitions from T_μ in step 8 takes $O(\sqrt{n})$ time. Step 9 and step 10 require $O(n)$ time to obtain final clusters of the dataset using K-means algorithm. Hence, the overall time complexity of DK-means algorithm is $O(n \log n)$. Although the proposed algorithm is $O(\log n)$ time slower than standard K-means algorithm, it takes very few iterations during cluster assignment phase which enables the proposed algorithm to run much faster than standard K-means.

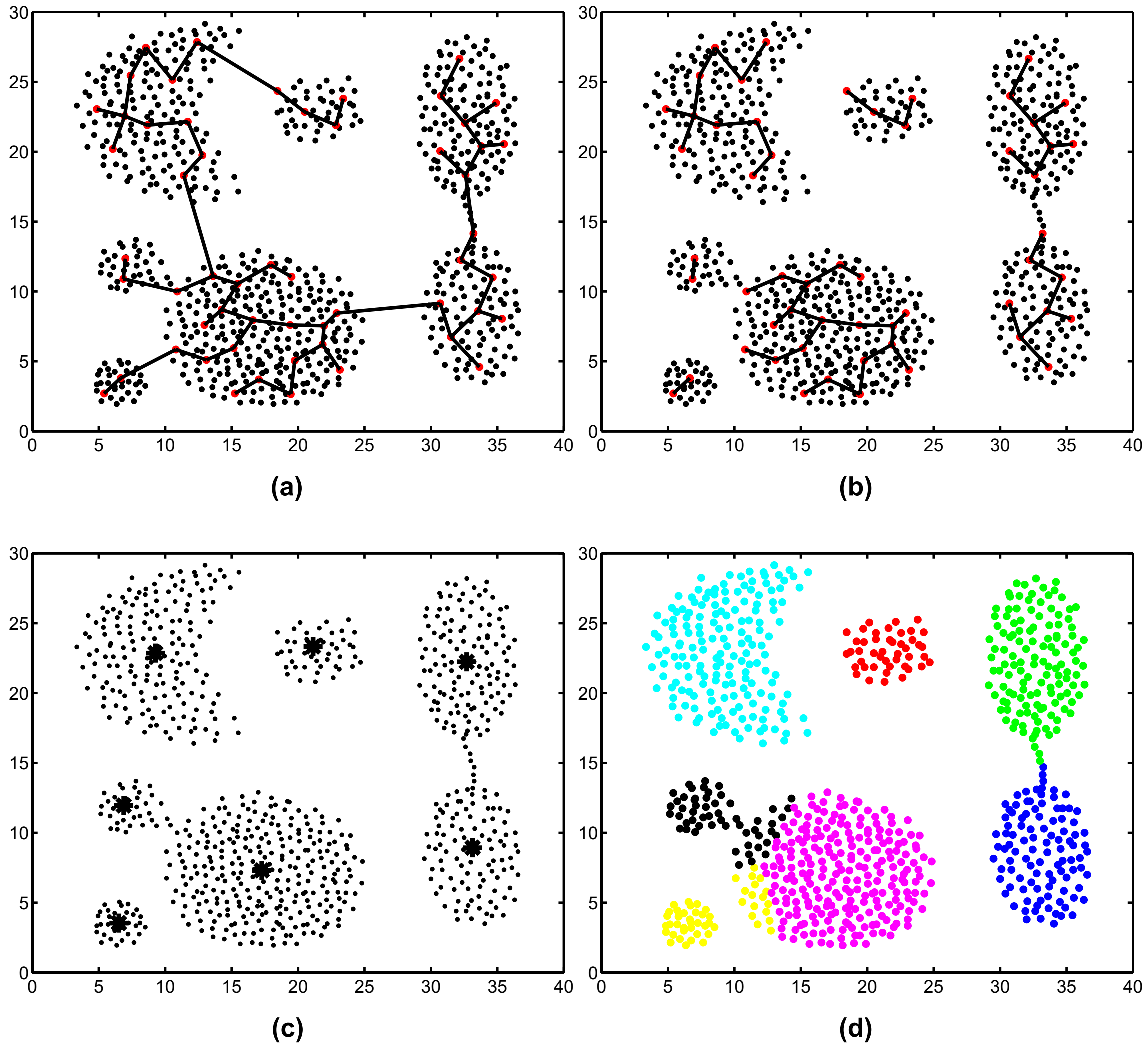


Fig. 3 MST-based partitioning of probable centers: **a** MST of probable centers. **b** k disjoint subsets of probable centers after removing $k-1$ longest edges. **c** Actual initial centers for K-means identified by proposed method. **d** Final clusters identified by K-means initialized with proposed algorithm

5 Results and discussion

5.1 Experimental setup

The proposed DK-means algorithm is compared with classical K-means (with random initialization), K-means variants such as K-means++ [3] and MinMax [40]. K-means++ applies a probabilistic approach in selecting the centers. It chooses the first center k_1 randomly from the dataset and other centers k_i, $2 \leq i \leq k$, are chosen such that distance between k_i and the previously chosen centers is maximum. K-means++ not only speeds up convergence of the clustering process but also yields better clustering results than K-means [8]. MinMax algorithm proposed by Tzortzis and Likas [40] tackles initialization problem by implementing a weighted version of the K-means by assigning weights to the clusters in proportion to their variance. Instead of minimizing the sum of squared error function, MinMax algorithm minimizes the maximum cluster variance by comparing relative variance of the clusters. In the beginning, all the clusters are given equal weights. Then, the weights are updated according to the cluster assignments through an iterative procedure. This weighing scheme controls the emergence of large variance clusters and attains high-quality partitions, irrespective of the initialization. K-means++ aims at selecting initial centers that cover the entire data space, thus providing better initialization to K-means [8, 40]. Similarly, MinMax algorithm has also proved to be effective in producing better initial centers by implementing a weighted version of the K-means. Thus, these two competing algorithms are chosen against which the proposed algorithm is compared.

Apart from the above-mentioned algorithms, the proposed algorithm is compared also with three deterministic initialization approaches, namely Var-part method of Ting and Jennifer [39], discriminant analysis using Histogram approach of Celebi and Kingravi [7] (referred as Histogram method) and DBCRIMES algorithm with representative identification by means of ranking criteria (DBCRIMES2) proposed by Filippo et al. [4]. These approaches have also shown to provide deterministic results, and hence, they are chosen for comparison with the proposed algorithm.

The evaluation is carried out based on the number of iterations (I), time required for K-means convergence and mean squared error (MSE) of clusters after the convergence. The quality of clusters obtained by different algorithms is compared using external quality measure, Adjusted Rand Index (ARI) and internal quality measure Silhouette Index (SI). ARI is an external cluster validity index that measures the degree of agreement between two partitions. Let P_1 be the ground truth partitions and let P_2 be the partitions predicted by a clustering algorithm. Let N_{11} be the number of pairs that are clustered together in both P_1 and P_2. Let N_{00} be the number of pairs that are in different clusters in both P_1 and P_2. Let N_{10} be the number of pairs that are in same cluster in P_1 and different clusters in P_2. Let N_{01} be the number of pairs that are in different clusters in P_1 and same cluster in P_2. Adjusted Rand Index (ARI) is defined as follows [16].

$$\text{ARI} = \frac{2(N_{11}N_{00} - N_{10}N_{01})}{(N_{11} + N_{10})(N_{10} + N_{01}) + (N_{11} + N_{01})(N_{01}N_{00})}$$

While the complete agreement between the two partitions P_1 and P_2 is indicated by ARI = 1, the complete disagreement between the partitions is observed when ARI = − 1.

Silhouette Index represents the degree of cohesion and separation of clusters obtained by a clustering algorithm. Let $C = \{C_1, C_2, \dots, C_k\}$ be the clustering solution obtained by an algorithm. Consider a point i which has been assigned to the cluster C_i. Let a_i denotes the average distance between the point i to the rest of the points in the same cluster C_i. Compute the average distance between the point i and the points in other clusters $C_j, C_i \neq C_j$, and let b_i denotes the minimum of average distances. For the ith point, Silhouette Index of a point is defined as follows [36].

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$$

An overall measure of goodness of a clustering is obtained by computing the average silhouette coefficient of all the points in the dataset. The Silhouette Index takes values in the range − 1 to 1, where the value 1 indicates all the points are appropriately clustered.

Both artificial and real datasets are considered for experimental study. All the algorithms are run for 10 times, and the mean values of the parameters I, MSE, ARI and SI are compared. All the algorithms have been implemented in MATLAB. Experiments are conducted on a computer with an Intel Core2 Duo Processor 2.83 GHz CPU and 4 GB memory.

5.2 Evaluation on artificial datasets

We consider four artificial datasets DS1, DS2, DS3 and DS4. These datasets correspond to Gaussain3, Gaussain5, Jain's toy and aggregation datasets, respectively. Details of the artificial datasets are given below.

DS1 This dataset is the Gaussain3 dataset used in [14]. It contains 60 instances of 600 features. It contains a partition of 3 clusters. The data simulate a pattern whereby a distinct set of 200 genes is up-regulated in one of the three clusters, and down-regulated in the remaining two.

DS2 This dataset is the Gaussain5 dataset used in [14]. It is a two-dimensional dataset of 500 points. It represents the union of observations from 5 bivariate Gaussians.

DS3 This dataset corresponds to Jain's toy dataset used in [18]. It is a two-dimensional dataset of 100 points. It represents two half-spirals shaped like a crescent, where each spiral corresponds to a cluster.

DS4 It is a two-dimensional dataset of 788 points representing the aggregation of 7 clusters as shown in Fig. 6a [15].

Table 1 shows the comparison of algorithms according to number of iterations (I) and mean squared error (MSE) on the datasets DS1, DS2, DS3 and DS4. For MinMax algorithm, first we run the MinMax weight learning iterative procedure to obtain the initial centers for K-means. Here, we measure both the number of iterations required for MinMax initialization and K-means convergence. The value mentioned in bracket denotes the number of iterations required for MinMax initialization. It is clearly visible from the results that the proposed algorithm DK-means performs quite better than the rest of the algorithms. It takes less number of iterations and also exhibits reduced MSE. The time reported for all the algorithms includes both the time spent on initialization and applying K-means with respective initial centers. Comparing the runtime, DK-means seems to converge faster than other algorithms on all the datasets. Due to repeated restarts during weight learning process, MinMax incurs high computational cost [40]. Similarly, DBCRIMES2 algorithm also exhibits slower speed due to quadratic time complexity of ranking-based initialization procedure. This is clearly reflected in the overall runtime of the MinMax as well as DBCRIMES2 algorithms.

Next, quality of clusters obtained by the different algorithms is compared on each of the artificial datasets. The clusters are evaluated using Adjusted Rand Index (ARI) and Silhouette Index (SI) measures. ARI and SI values of the clusters produced by the algorithms are shown in Tables 2 and 3, respectively. We have run all the algorithms for 10 runs and the mean values of ARI and SI scores are reported in the results. According to ARI and SI values, proposed algorithm DK-means achieves quite better results as compared to other algorithms. Although K-means++ chooses the centers in such a way that they do not collide with each other, random choice of the first center affects the subsequent centers. So the algorithm produces different results on different runs, thus suffering from unstable clusters as in random K-means. Similar reasoning applies for MinMax algorithm also, but the degree of randomness is less as compared to K-means++.

On DS1 dataset, both MinMax and DK-means produce better clustering results as compared to other algorithms. On DS2 dataset, DK-means attains maximum ARI value. Figure 4 illustrates the comparison of algorithms on DS2 dataset. Due to random center initialization, K-means is not able to completely separate the five clusters as shown in Fig. 4b. K-means++ could not identify the farthest points belonging to the same cluster as shown in Fig. 4c. Similarly, MinMax algorithm could not find a clear partition of the overlapping clusters. Var-Part, Histogram and DBCRIMES2 approaches have shown improvement over K-means and K-means++, although they could not identify exact cluster separation. Selecting more than one representative point for a cluster aids DK-means algorithm to identify the points of the same cluster which are farther, and thus, it is able to identify the expected clusters as shown in Fig. 4h.

Table 1 Comparison of algorithms according to the number of iterations for K-means convergence (*I*), time (in milliseconds) and mean squared error (MSE) on artificial datasets

Dataset	Parameter	K-means	K-means++	MinMax	Var-Part	Histogram	DBCRIMES2	DK-means
DS1	*I*	5	4	2 (52)	3	4	3	**2**
	Time	127.49	66.14	75.24	70.11	72.33	98.14	**64.66**
	MSE	4.79E+04	4.65E+04	4.62E+04	4.87E+04	4.50E+04	4.48E+04	**4.40E+04**
DS2	*I*	9	7	3 (91)	8	6	**4**	**4**
	Time	141.71	41.22	73.95	51.77	57.43	88.56	**28.53**
	MSE	1.4548	1.0314	0.6127	0.8343	0.8911	0.5666	**0.4829**
DS3	*I*	7	6	2 (52)	6	5	4	**3**
	Time	93.95	32.87	33.48	34.66	35.02	66.18	**28.07**
	MSE	10.8794	10.761	10.3178	11.4581	11.0877	12.6161	**10.3069**
DS4	*I*	18	17	12 (500)	13	10	12	**11**
	Time	143.33	41.35	260.26	79.65	80.14	101.23	**36.56**
	MSE	15.2122	15.6465	15.2418	16.0058	15.9847	14.8787	**14.6121**

The lowest value in each row is highlighted. (For MinMax algorithm, the value mentioned in bracket denotes the number of iterations required for MinMax initialization)

Table 2 Comparison of Adjusted Rand Index (ARI) values obtained by different algorithms on artificial datasets

Dataset	K-means	K-means++	MinMax	Var-Part	Histogram	DBCRIMES2	DK-means
DS1	0.4316	0.5923	**1.0000**	0.7878	0.8436	0.9858	**1.0000**
DS2	0.7173	0.7199	0.8939	0.8705	0.9144	0.9666	**1.0000**
DS3	0.6661	0.6331	0.7491	0.5366	0.6133	0.6848	**0.7699**
DS4	0.6478	0.7521	0.7527	0.4165	0.5001	0.6840	**0.7702**

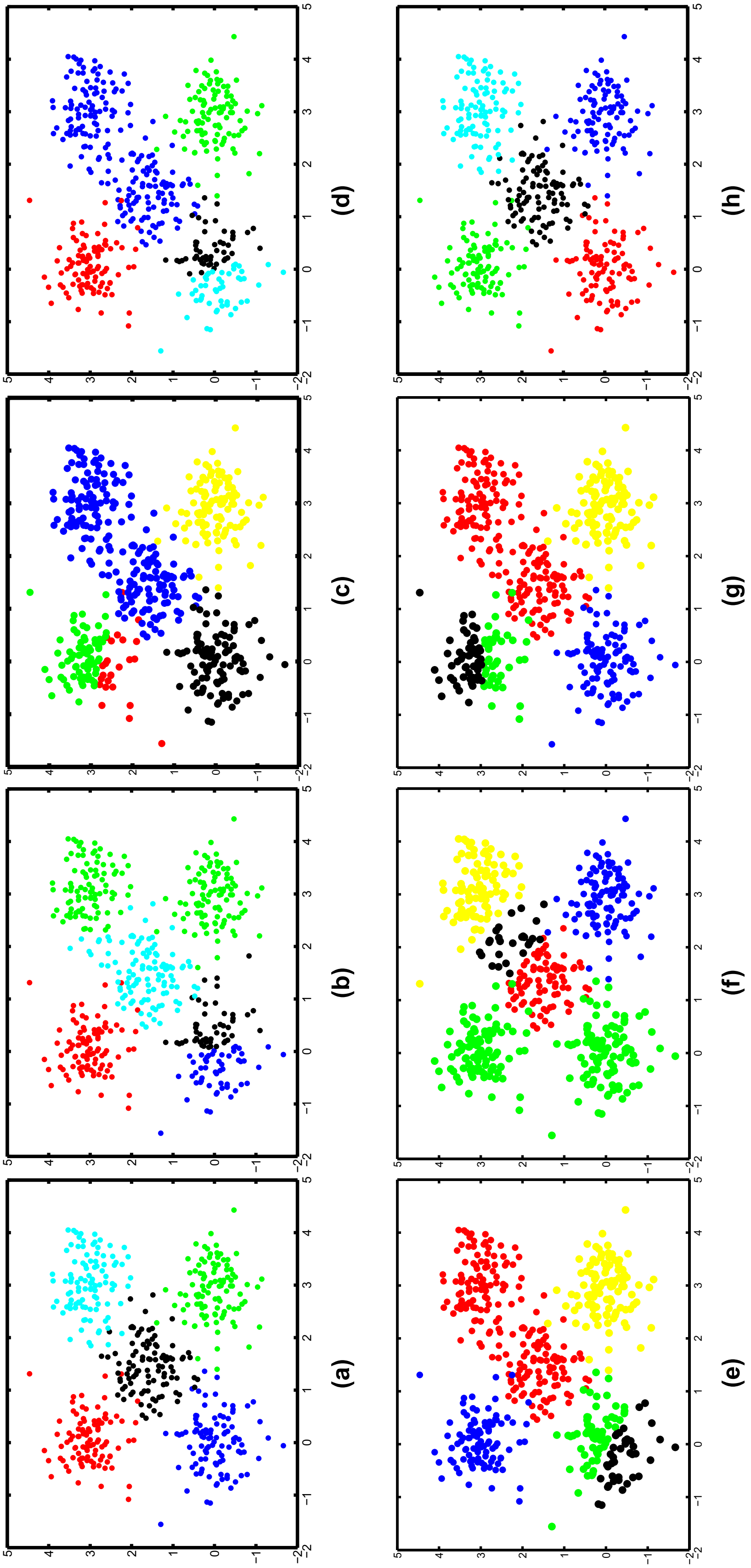
The highest value in each row is highlighted

Table 3 Comparison of Silhouette Index (SI) values obtained by different algorithms on artificial datasets

Dataset	K-means	K-means++	MinMax	Var-Part	Histogram	DBCRIMES2	DK-means
DS1	0.0455	0.0714	**0.1182**	0.1009	0.1174	0.9988	**0.1182**
DS2	0.6832	0.6811	0.8002	0.5888	0.6791	0.7405	**0.8653**
DS3	0.6311	0.6303	0.6361	0.5125	0.6187	0.6244	**0.6373**
DS4	0.6351	0.6722	0.6778	0.4297	0.5888	0.6700	**0.6801**

The highest value in each row is highlighted

Fig. 4 Comparison of algorithms on DS2 dataset: **a** Dataset with 5 Gaussian clusters. Clusters identified by **b** K-means with random initialization, **c** K-means++, **d** MinMax, **e** Var-Part, **f** Histogram, **g** DBCRIMES2 and **h** proposed DK-means algorithm



On DS3 dataset, performance of all the algorithms is almost similar as shown in Fig. 5. On DS4 dataset, DK-means performs fairly better than other algorithms as shown in Fig. 6. The touching clusters (shown in magenta and green color in Fig. 6a) are identified correctly by K-means++ and DK-means algorithms. But MinMax algorithm breaks one of the touching clusters due to the relative comparison of cluster variances. While the well-separated cluster (shown in blue color in Fig. 6a) is completely perceived by the proposed algorithm, the algorithms K-means and K-means++ are mixing up this cluster with other clusters. Although Var-part, Histogram and DBCRIMES2 algorithms compete our proposed algorithm DK-means in terms of stable clustering results, DK-means has shown improvement over these algorithms in terms of both cluster quality and stable results.

5.3 Evaluation on gene expression datasets

Performance of the proposed algorithm is also evaluated on ten real gene expression datasets and the details of them are given in Table 4. The BreastA, BreastB, DLBCLA, Novartis, ALB, Leukemia1 and LungA datasets are available at [6]. The datasets BreastA, BreastB, DLBCLA and Novartis are addressed in [17]. ALB, Leukemia and LungA datasets are addressed in [34]. Yeast cell cycle (yeast) can be downloaded from [41]. We use the class label given in [41] as the ground truth for yeast dataset. Lymphoma dataset was used in [1].

First we present the convergence parameters, namely number of iterations (I) and mean squared error (MSE) obtained by the different algorithms on the gene datasets. As dimensionality of the dataset increases, complexity of identifying best representative points for K-means initialization also increases. Table 5 shows the comparison of algorithms according to number of iterations and MSE on these datasets. Results indicate that K-means algorithm obtains slower convergence as well as higher MSE than the rest of the algorithms. It is also observed that K-means++ takes relatively less time in a few datasets such as BreastA and BreastB. As these datasets are high-dimensional, the proposed algorithm incurs a substantial amount of partition time which makes the algorithm little more expensive in terms of overall runtime as compared to K-means++. However, the proposed algorithm DK-means provides qualitative improvement over K-means++, without much compromise in the speed. MinMax algorithm runs much slower as compared to other algorithms due to the amount of time spent on MinMax initialization. The maximum number of iterations for MinMax initialization is set to 500 as given in [40]. It is also observed that MinMax initialization procedure did not converge in 500 iterations for Novartis, cGCM and yeast cell cycle datasets. So runtime of the MinMax algorithm is quite large on these datasets. Similarly, DBCRIMES2 also runs slower as compared to other algorithms. Both Var-part and Histogram algorithms have shown quite similar results on most of the datasets. As compared to other algorithms, the proposed algorithm DK-means attains faster convergence with reduced MSE on most of the datasets.

The clusters obtained on gene expression datasets by all the algorithms are validated using ARI and SI measures. The results are shown in Tables 6 and 7, respectively. As compared to random K-means and MinMax algorithms, DK-means obtains higher value of ARI on all the datasets as given in Table 6. Although K-means++ performs better than random K-means, its ARI value is lower than DK-means on all the datasets except BreastB dataset. On an average, DK-means gets better clustering results than other algorithms in terms of ARI. Comparing SI scores obtained by all the algorithms given in Table 7, it is observed that the proposed algorithm achieves improved clustering in terms of cluster cohesion and separation. Thus, DK-means achieves improved cluster quality in terms of both external and internal validity measures.

5.4 Deterministic analysis

In order to show that the proposed algorithm DK-means is deterministic in terms of convergence and cluster quality, the algorithm is executed for 10 times and the minimum, maximum and average number of iterations taken by the algorithm are recorded. Further, the quality measures MSE, ARI and SI are also computed on each of these runs. These parameters for K-means, K-means++ and MinMax algorithms are also computed. Table 8 reports minimum, maximum and average number of iterations taken by different algorithms for 10 independent runs. Table 8 shows that the average, minimum and maximum number of iterations taken by the proposed algorithm are the same, while these values are different in case of K-means, K-means++ and MinMax algorithms. These algorithms produce different clustering results based on their convergence.

While K-means starts with allocating cluster centers randomly and then looks for optimum solution, K-means++ starts with allocation of one cluster center randomly and then searches for other centers, given the first one. So both the algorithms use random initialization as a starting point and therefore provide different results on different runs. The proposed DK-means algorithm does not select centers randomly; rather, it uses the best representative points as initial centers. Hence, the results are deterministic both in terms of number of iterations and cluster quality.

The proposed algorithm is stable in terms of MSE for different runs of the algorithm. For the sake of illustration, the MSE plots by 10 independent runs of the algorithms K-means, K-means++, MinMax and DK-means on yeast cell cycle and lymphoma datasets are presented in Fig. 7. While

Fig. 5 Comparison of algorithms on DS2 dataset: **a** Dataset with 2 half-spirals. Clusters identified by **b** K-means with random initialization, **c** K-means++, **d** MinMax, **e** Var-Part, **f** Histogram, **g** DBCRIMES2 and **h** proposed DK-means algorithm

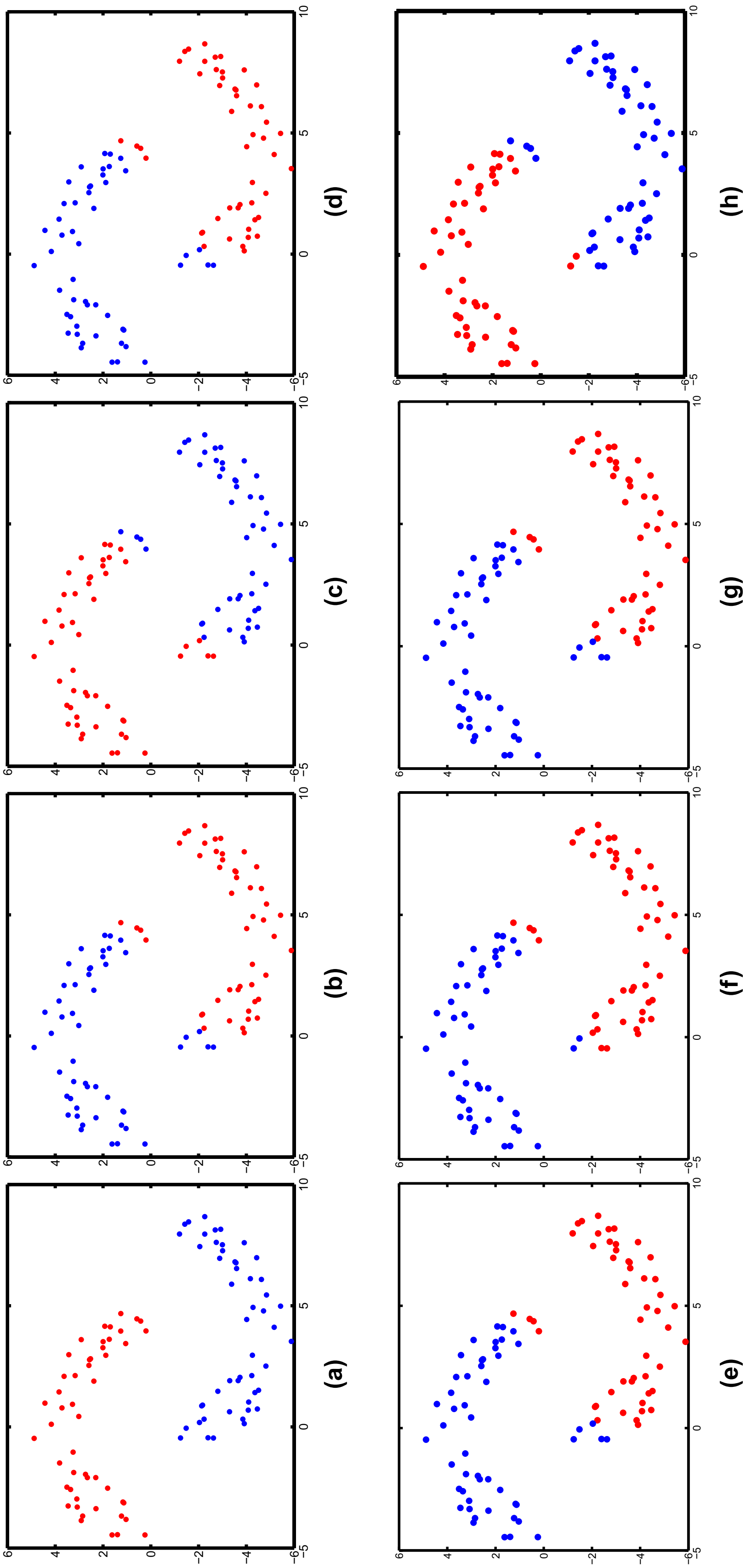


Fig. 6 Comparison of algorithms on DS4 dataset: **a** Dataset with 7 clusters. Clusters identified by **b** K-means with random initialization, **c** K-means++, **d** MinMax, **e** Var-Part, **f** Histogram, **g** DBCRIMES2 and **h** proposed DK-means algorithm (color figure online)

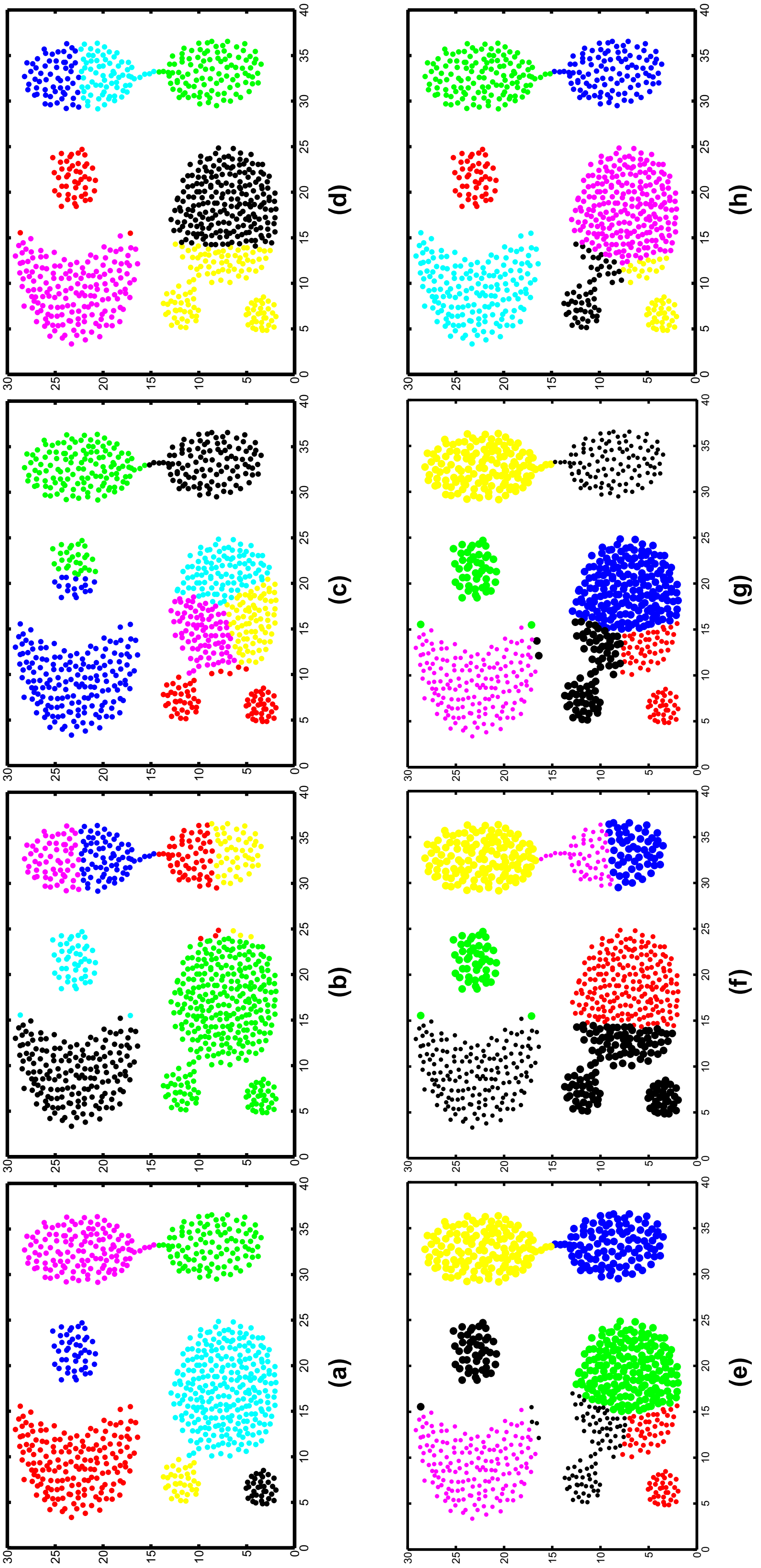


Table 4 Details of the datasets: No. of instances (n), No. of dimensions (d), No. of clusters (k)

Dataset	n	d	k
BreastA	98	1213	3
BreastB	49	1213	4
DLBCLA	141	661	3
Novartis	103	1000	4
ALB	38	1000	3
Leukemia	248	985	6
cGCM	90	630	13
LungA	214	197	4
Yeast	384	17	5
Lymphoma	80	100	3

other algorithms produce different value of MSE for each of the independent runs, the deterministic algorithms Var-part, Histogram, DBCRIMES2 and DK-means produce constant value of MSE. Similar analysis has been carried out on the rest of the datasets and the results are found to be consistent. Among deterministic approaches, DK-means has maintained deterministic nature as well as improved cluster quality.

5.5 Robustness against noise

In order to evaluate the robustness of the proposed algorithm against noise, we have carried out experiments on datasets with different levels of noise. The results are shown for artificial datasets DS2 and DS3 as representatives for almost all

Table 5 Comparison of algorithms according to Number of iterations for K-means convergence (I), Time (in milliseconds) and MSE on gene expression datasets

Dataset	Parameter	K-means	K-means++	MinMax	Var-Part	Histogram	DBCRIMES2	DK-means
BreastA	I	6	5	3 (58)	5	4	7	**3**
	Time	184.93	**92.54**	252.10	148.44	136.48	178.24	117.89
	MSE	88.32	84.05	87.36	90.17	84.36	82.31	**81.86**
BreastB	I	6	5	4 (56)	7	6	4	**3**
	Time	157.27	**48.07**	123.04	94.69	91.38	102.77	66.48
	MSE	1.4548	1.0314	0.6127	d1	d2	d3	**0.4829**
DLBCLA	I	9	8	6 (57)	7	7	**6**	**6**
	Time	467.59	221.42	243.66	d1	d2	d3	**185.91**
	MSE	8.23E+07	8.23E+07	8.23E+07	d1	d2	d3	**7.90E+07**
Novartis	I	11	13	8 (500)	14	12	12	**10**
	Time	164.33	230.41	403.26	147.75	152.81	184.01	**89.94**
	MSE	8.34E+08	5.83E+08	6.58E+08	7.02 + 08	6.94E+08	6.11E+08	**5.75E+08**
ALB	I	7	6	2 (53)	4	4	3	**2**
	Time	145.88	83.17	86.91	30.66	39.11	51.82	**21.92**
	MSE	8.83E+08	8.33E+08	8.21E+08	8.58E+08	8.36E+08	8.34E+08	**8.20E+08**
Leukemia	I	15	14	7 (85)	16	17	14	**13**
	Time	697.84	364.18	502.89	387.08	400.11	486.33	**270.07**
	MSE	7.63E+10	7.77E+10	7.50E+10	7.43E+10	7.39E+10	7.30E+10	**7.25E+10**
cGCM	I	12	9	3 (500)	11	8	5	**3**
	Time	352.21	255.51	653.40	211.25	220.44	289.65	**244.47**
	MSE	6.06E+08	5.45E+08	5.69E+08	6.01E+06	5.89E+06	5.77E+06	**5.43E+08**
LungA	I	22	20	6 (55)	17	11	12	**7**
	Time	157.90	71.52	94.6	87.44	74.69	100.44	**68.03**
	MSE	1.58E+07	**1.57E+07**	1.58E+07	1.58E+07	1.58E+07	1.58E+07	1.58E+07
Yeast	I	13	11	6 (500)	11	12	10	**8**
	Time	146.29	29.05	152.64	28.89	30.25	33.36	**22.44**
	MSE	6.3545	6.3774	6.3137	6.7125	6.8006	6.3444	**6.2736**
Lymphoma	I	7	6	4 (56)	8	5	5	**4**
	Time	140.81	30.56	47.12	50.06	49.88	68.19	**21.45**
	MSE	62.9683	62.7622	62.9682	63.4901	61.0057	61.8110	**60.7932**

The lowest value in each row is highlighted. (For MinMax algorithm, the value mentioned in bracket denotes the number of iterations required for MinMax initialization)

Table 6 Comparison of Adjusted Rand Index (ARI) values obtained by different algorithms on gene expression datasets

Dataset	K-means	K-means++	MinMax	Var-Part	Histogram	DBCRIMES2	DK-means
BreastA	0.3892	0.4888	0.4786	0.3141	0.4959	0.5141	**0.5969**
BreastB	0.0892	**0.2160**	0.1714	0.0653	0.0814	0.0766	0.0830
DLBCLA	0.1155	0.1068	0.1212	0.0944	0.1068	0.1270	**0.1325**
Novartis	0.1813	0.5546	0.4238	0.2473	0.4844	0.4991	**0.6010**
ALB	0.4159	0.5447	0.6987	0.4641	0.5133	0.6045	**0.9113**
Leukemia	0.6972	0.6779	0.6556	0.4555	0.5874	0.6166	**0.7437**
cGCM	0.2252	0.3153	0.3175	0.2188	0.3688	0.3502	**0.3968**
LungA	0.0504	0.1323	0.1741	0.0661	0.0845	0.1182	**0.1914**
Yeast	0.4247	0.4429	0.4247	0.3981	0.4446	0.5109	**0.5316**
Lymphoma	0.1872	0.1938	0.2451	0.1133	0.2045	0.2233	**0.2620**

The highest value in each row is highlighted

Table 7 Comparison of Silhouette Index (SI) values obtained by different algorithms on gene expression datasets

Dataset	K-means	K-means++	MinMax	Var-Part	Histogram	DBCRIMES2	DK-means
BreastA	0.1325	0.2296	0.1593	0.1745	0.2856	0.4287	**0.3412**
BreastB	0.1095	0.0718	0.2040	0.0514	0.0662	0.1860	**0.3845**
DLBCLA	0.2355	0.2492	0.1825	0.2043	0.2291	**0.2516**	0.2291
Novartis	0.3056	0.3789	0.4002	0.2006	0.3155	0.3688	**0.5058**
ALB	0.1197	0.4189	0.4318	0.2817	0.4366	0.5714	**0.4394**
Leukemia	0.2336	0.2496	0.3222	0.2688	0.2944	0.2845	**0.3370**
cGCM	0.3543	0.3321	0.1782	0.1467	0.2914	0.3044	**0.4059**
LungA	0.2321	0.3363	0.4360	0.1568	0.3901	0.4145	**0.4431**
Yeast	0.3714	0.3706	0.3714	0.2996	0.4145	0.4390	**0.5310**
Lymphoma	**0.3338**	0.2904	0.3186	0.2488	0.3008	0.2165	**0.3338**

The highest value in each row is highlighted

the datasets. Figure 8 illustrates the addition of different levels of noise to datasets DS2 and DS3. DK-means algorithm has shown performance improvement over K-means and K-means++ against varying levels of noise as shown in Fig. 9. As the probable centers of DK-means algorithm are located at maximum distance, pruning longest edges from the MST of probable centers separates the noise from the actual clusters. Thus, the results of DK-means are not affected by the presence of noise in the data. It is also observed that MinMax algorithm performs similar to DK-means.

5.6 Biological significance

The functional enrichment of a group of genes is represented by three independent structured, controlled vocabularies: molecular function, biological process and cellular component. The degree of functional enrichment is measured by the cumulative hypergeometric distribution. For a particular GO category, the probability p of producing k or more genes within a cluster of size x is calculated as [33]

$$p = 1 - \sum_{j=1}^{k-1} \frac{\binom{M}{j}\binom{N-M}{x-j}}{\binom{N}{x}},$$

where M and N denote the total number of genes within a category and that within the genome, respectively. By computing the p values, we can analyze the statistical significance of the genes in a cluster. If majority of genes in a cluster possess the same biological function, then p value of the obtained cluster will be equal to 0.

Biological significance of clusters obtained by the proposed algorithm DK-means is analyzed with the help of Gene Ontology (GO) Term Finder tool available at http://www.yeastgenome.org/cgi-bin/GO/goTermFinder.pl. The GO Term Finder searches for significant shared GO terms used to establish the biological relevance of a cluster. Biological enrichment of clusters obtained by the proposed algorithm is tested on all the ten gene expression datasets. It is found that clusters are biologically significant with respect to three most significant GO terms of all the clusters for each

Table 8 Deterministic analysis of the algorithms with respect to number of iterations (I)

Dataset	Iterations (I)	K-means	K-means++	MinMax	Var-Part	Histogram	DBCRIMES2	DK-means
BreastA	I_{MIN}	2	3	2	5	4	7	3
	I_{MAX}	8	8	5	5	4	7	3
	I_{AVG}	6	5	3	5	4	7	3
BreastB	I_{MIN}	4	2	3	7	6	4	3
	I_{MAX}	11	6	6	7	6	4	3
	I_{AVG}	6	3	4	7	6	4	3
DLBCLA	I_{MIN}	7	5	4	7	7	6	6
	I_{MAX}	14	17	6	7	7	6	6
	I_{AVG}	9	8	6	7	7	6	6
Novartis	I_{MIN}	4	3	5	14	12	12	10
	I_{MAX}	12	12	6	14	12	12	10
	I_{AVG}	9	7	6	14	12	12	10
ALB	I_{MIN}	4	3	2	4	4	3	2
	I_{MAX}	8	10	2	4	4	3	2
	I_{AVG}	7	6	2	4	4	3	2
Leukemia	I_{MIN}	8	11	4	16	17	14	13
	I_{MAX}	23	24	8	16	17	14	13
	I_{AVG}	15	14	7	16	17	14	13
cGCM	I_{MIN}	14	7	4	11	8	5	3
	I_{MAX}	16	11	6	11	8	5	3
	I_{AVG}	12	9	3	11	8	5	3
LungA	I_{MIN}	10	8	5	17	11	12	7
	I_{MAX}	8	29	7	17	11	12	7
	I_{AVG}	22	20	6	17	11	12	7
Yeast	I_{MIN}	10	6	4	11	12	10	4
	I_{MAX}	26	14	7	11	12	10	8
	I_{AVG}	13	11	6	11	12	10	8
Lymphoma	I_{MIN}	3	3	3	8	5	5	4
	I_{MAX}	14	11	5	8	5	5	4
	I_{AVG}	7	6	4	8	5	5	4

Here I_{MIN}, I_{MAX} and I_{AVG} denote the minimum, maximum and average number of iterations taken by the algorithms measured over 10 independent runs

dataset. For illustration purpose, yeast sporulation dataset [33] is considered here and the number of clusters is set to 6 as given in [33]. Biological relevance of each of the 6 clusters obtained by the DK-means algorithm is tested using GO Term Finder at 1% significance level. Table 9 reports three most significant GO terms along with their p values. All the clusters identified by DK-means have obtained p values less than 0.01, revealing that the clusters are biologically significant.

6 Conclusion

In this paper, DK-means algorithm is proposed, which is a variant of K-means with deterministic selection of initial centers. Efficiency of the proposed algorithm is demonstrated through experiments on both artificial as well as real gene expression datasets. The results prove that the proposed algorithm outperforms other existing algorithms with faster convergence rate, reduced MSE and improved cluster quality. While the clustering results of K-means

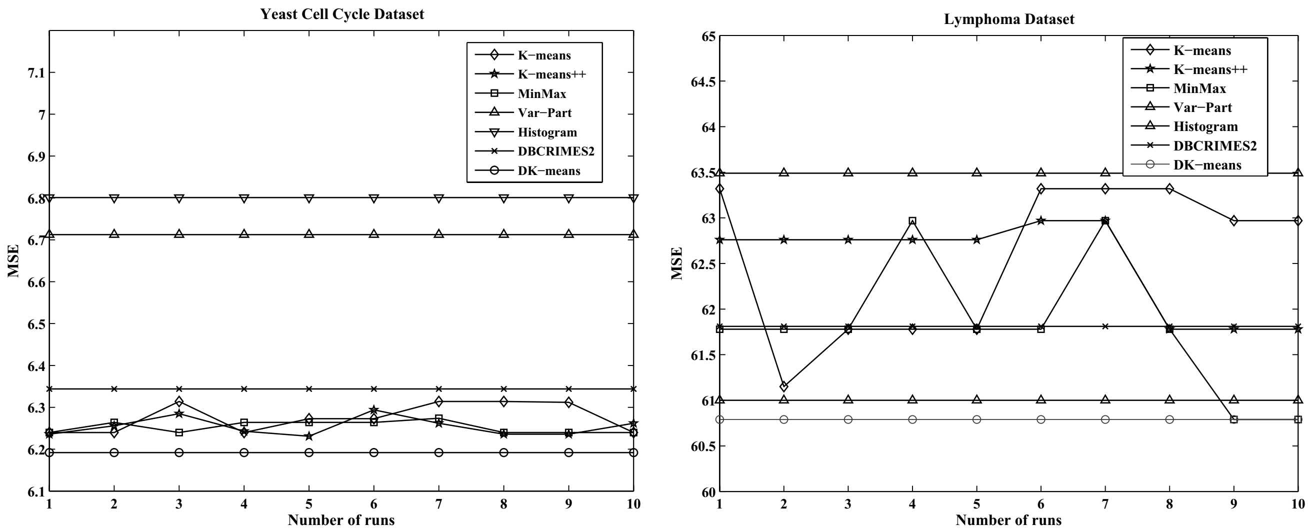


Fig. 7 MSE plots by 10 independent runs of the algorithms

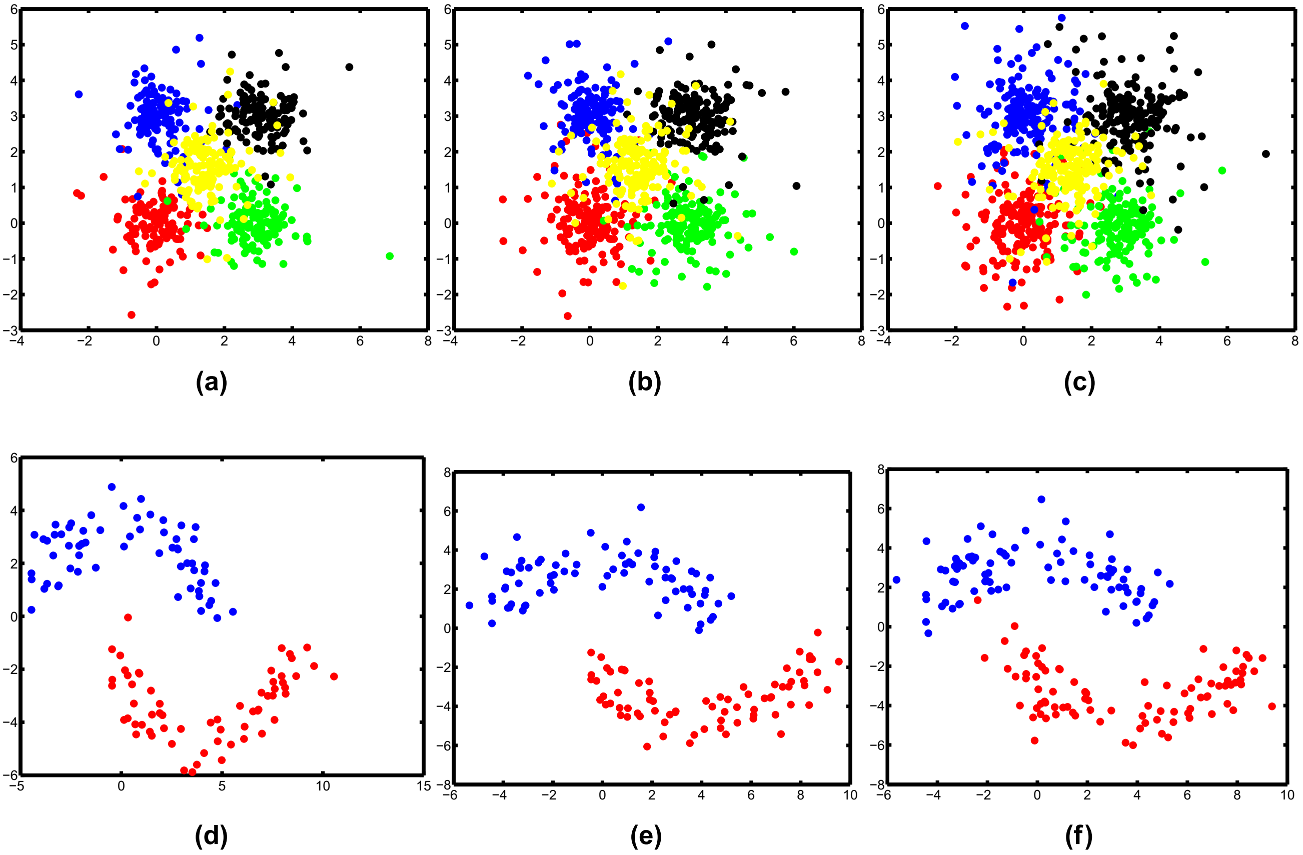


Fig. 8 Artificial datasets with different levels of noise. First row (second row) corresponds to DS2 (DS3) dataset. From left to right, columns correspond to low, medium and high noise levels

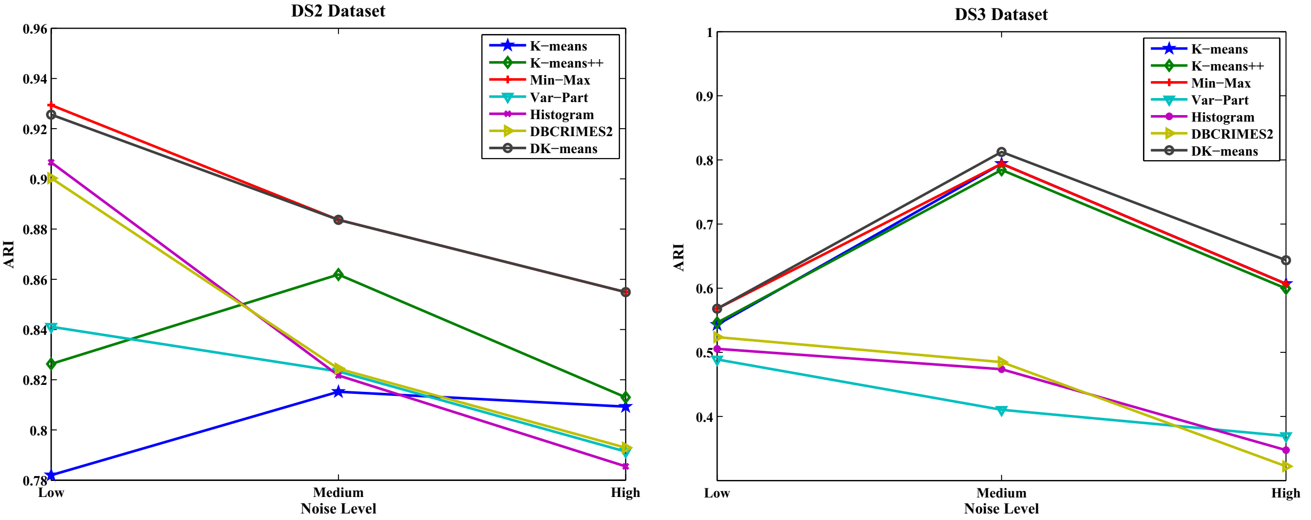


Fig. 9 Comparison of different algorithms with respect to robustness to noise

Table 9 Most significant GO terms and the corresponding *p* values for each of the 6 clusters of yeast sporulation data obtained by DK-means algorithm

Clusters	Three most significant GO terms	*p* value
Cluster 1	Glutamine family amino acid metabolic process GO:0009064	0.00038
	Carboxylic acid metabolic process GO:0019752	0.00174
	Oxoacid metabolic process GO:0043436	0.00174
Cluster 2	Cytoplasmic translation GO:0002181	2.55E−36
	Peptide biosynthetic process GO:0043043	5.56E−31
	Translation GO:0006412	5.56E−31
Cluster 3	Coenzyme metabolic process GO:0006732	3.04E−06
	Nucleotide metabolic process GO:0009117	6.19E−06
	Carbohydrate metabolic process GO:0005975	6.44E−06
Cluster 4	Ribonucleoprotein complex biogenesis GO:0022613	8.53E−12
	Ribosome biogenesis GO:0042254	8.53E−12
	ncRNA processing GO:0034470	2.21E−10
Cluster 5	Meiotic nuclear division GO:0007126	5.01E−15
	DNA recombination GO:0006310	8.63E−15
	Meiosis I GO:0007127	1.13E−14
Cluster 6	Anatomical structure development GO:0048856	5.87E−11
	Natomical structure morphogenesis GO:0009653	5.87E−11
	Cellular component morphogenesis GO:0032989	8.76E−11

algorithm with random initialization are unstable, DK-means algorithm produces deterministic results. Biological significance of the clusters produced by the proposed algorithm is also investigated, and the results indicate that clusters are biologically relevant.

References

1. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X et al (2000) Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403(6769):503–511
2. Alrabea A, Senthilkumar A, Al-Shalabi H, Bader A (2013) Enhancing k-means algorithm with initial cluster centers derived from data partitioning along the data axis with PCA. J Adv Comput Netw 1(2):137–142
3. Arthur D, Vassilvitskii S (2007) k-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, society for industrial and applied mathematics, pp 1027–1035
4. Bianchi FM, Livi L, Rizzi A (2016) Two density-based k-means initialization algorithms for non-metric data clustering. Pattern Anal Appl 19(3):745–763

5. Bradley PS, Fayyad UM (1998) Refining initial points for k-means clustering. In: Proceedings of 15th international conference on machine learning (ICML), vol 98. pp 91–99
6. Broad Institute Cancer Program Datasets (2016) http://broadinstitute.org/cgi-bin/cancer/
7. Celebi ME, Kingravi HA (2012) Deterministic initialization of the k-means algorithm using hierarchical clustering. Int J Pattern Recognit Artif Intell 26(07):1250,018–1–1250,018–25
8. Celebi ME, Kingravi HA, Vela PA (2013) A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl 40(1):200–210
9. Chavent M, Lechevallier Y, Briant O (2007) Divclus-t: a monothetic divisive hierarchical clustering method. Comput Stat Data Anal 52(2):687–701
10. Ding C, He X (2004) K-means clustering via principal component analysis. In: International conference on machine learning (ICML), ACM, pp 29–36
11. Du Z, Wang Y, Ji Z (2008) PK-means: a new algorithm for gene clustering. Comput Biol Chem 32(4):243–247
12. Duwairi R, Abu-Rahmeh M (2015) A novel approach for initializing the spherical k-means clustering algorithm. Simul Modell Pract Theory 54:49–63
13. Erisoglu M, Calis N, Sakallioglu S (2011) A new algorithm for initial cluster centers in k-means algorithm. Pattern Recognit Lett 32(14):1701–1705
14. Giancarlo R, Utro F (2011) Speeding up the consensus clustering methodology for microarray data analysis. Algorithms Mol Biol 6(1):1–13
15. Gionis A, Mannila H, Tsaparas P (2007) Clustering aggregation. ACM Trans Knowl Discov Data (TKDD) 1(1):1–30
16. Halkidi M, Batistakis Y, Vazirgiannis M (2001) On clustering validation techniques. J Intell Inf Syst 17(2):107–145
17. Hoshida Y, Brunet JP, Tamayo P, Golub TR, Mesirov JP (2007) Subclass mapping: identifying common subtypes in independent disease data sets. PloS ONE 2(11):e1195
18. Jain AK, Law MH (2005) Data clustering: a user's dilemma. Pattern Recognit Mach Intell 3776:1–10
19. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv (CSUR) 31(3):264–323
20. Jiang D, Tang C, Zhang A (2004) Cluster analysis for gene expression data: a survey. IEEE Trans Knowl Data Eng 16(11):1370–1386
21. Jothi R, Mohanty SK, Ojha A (2016a) Functional grouping of similar genes using eigenanalysis on minimum spanning tree based neighborhood graph. Comput Biol Med 71:135–148
22. Jothi R, Mohanty SK, Ojha A (2016b) On careful selection of initial centers for k-means algorithm. In: Proceedings of 3rd international conference on advanced computing, networking and informatics: ICACNI 2015, Vol 1, Springer India, New Delhi, pp 435–445
23. Kerr G, Ruskin HJ, Crane M, Doolan P (2008) Techniques for clustering gene expression data. Comput Biol Med 38(3):283–293
24. Khan SS, Ahmad A (2004) Cluster center initialization algorithm for k-means clustering. Pattern Recognit Lett 25(11):1293–1302
25. Krishna K, Murty MN (1999) Genetic k-means algorithm. IEEE Trans Syst Man Cybern Part B: Cybern 29(3):433–439
26. Lam YK, Tsang PW (2012) eXploratory k-means: a new simple and efficient algorithm for gene clustering. Appl Soft Comput 12(3):1149–1157
27. Lam YK, Tsang PWM, Leung CS (2013) Pso-based k-means clustering with enhanced cluster matching for gene expression data. Neural Comput Appl 22(7–8):1349–1355
28. Likas A, Vlassis N, Verbeek JJ (2003) The global k-means clustering algorithm. Pattern Recognit 36(2):451–461
29. Liu M, Jiang X, Kot AC (2009) A multi-prototype clustering algorithm. Pattern Recognit 42(5):689–698
30. Lu Y, Lu S, Fotouhi F, Deng Y, Brown SJ (2004a) FGKA: A fast genetic k-means clustering algorithm. In: Proceedings of the 2004 ACM symposium on Applied computing, ACM, pp 622–623
31. Lu Y, Lu S, Fotouhi F, Deng Y, Brown SJ (2004b) Incremental genetic k-means algorithm and its application in gene expression data analysis. BMC Bioinform 5(1):172–181
32. Martella F, Vichi M (2012) Clustering microarray data using model-based double k-means. J Appl Stat 39(9):1853–1869
33. Maulik U, Mukhopadhyay A, Bandyopadhyay S (2009) Combining pareto-optimal clusters using supervised learning for identifying co-expressed genes. BMC Bioinform 10(1):27–42
34. Monti S, Tamayo P, Mesirov J, Golub T (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Mach Learn 52(1):91–118
35. Nazeer K, Sebastian M, Kumar S (2013) A novel harmony search-k means hybrid algorithm for clustering gene expression data. Bioinformation 9(2):84–88
36. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
37. Sun J, Chen W, Fang W, Wun X, Xu W (2012) Gene expression data analysis with the clustering method based on an improved quantum-behaved particle swarm optimization. Eng Appl Artif Intell 25(2):376–391
38. Thalamuthu A, Mukhopadhyay I, Zheng X, Tseng GC (2006) Evaluation and comparison of gene clustering methods in microarray analysis. Bioinformatics 22(19):2405–2412
39. Ting S, Jennifer GD (2007) In search of deterministic methods for initializing k-means and gaussian mixture clustering. Intell Data Anal 11(4):319–338
40. Tzortzis G, Likas A (2014) The minmax k-means clustering algorithm. Pattern Recognit 47(7):2505–2516
41. Validating Clustering for Gene Expression Data (2012) http://faculty.washington.edu/kayee/cluster/
42. Xu R, Wunsch DC (2010) Clustering algorithms in biomedical research: a review. IEEE Rev Biomed Eng 3:120–154
43. Zahn CT (1971) Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans Comput 100(1):68–86