An improved artificial bee colony algorithm for clustering cancer patients

Qiao Liu [a]

[a] School of Advanced Engineering, Beihang University, Beijing, 100191, PR China

Abstract

Clustering is a very important technology for data analysis in the research and application of data mining. The principle of cluster analysis is to classify a set of objects according to their similarity. It belongs to a kind of unsupervised classification due to the lack of priori knowledge. So far, many clustering algorithms have been proposed to solve different clustering problems.

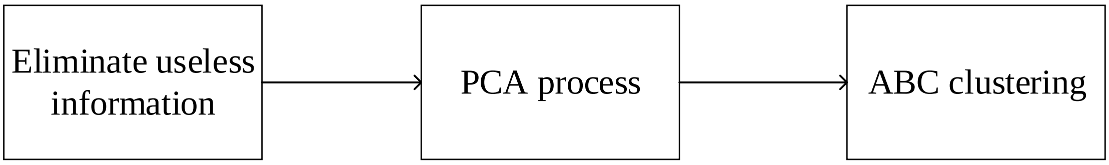
K-Means Clustering (KMC) algorithm is one of the classic method for cluster analysis. However, it is sensitive to the initial clustering centers and easy to be trapped by local optimum, we propose an improved Artificial Bee Colony (ABC) clustering algorithm based on K-Means to solve this clustering problem in the field of bioinformatics. Our algorithm integrates the improved ABC algorithm with the K-means iteration, which reduces the dependence on the initial clustering centers and increases the global search ability. The experiment results show that the clustering efficiency and performance has been significantly improved.

***Keywords**:* clustering analysis, artificial bee colony (ABC) algorithm, k-means

1. Introduction

In this clustering problem, a set of multi-platform cancer genomic data, including CNN, RPPA, mRNA, methylation, and miRNA of four kinds of cancer patients, is given, to simplify the problem, we only cluster one certain kind of cancer patients (LUSC) according to their relevant genomic data.

Our clustering method can be generally divided into three steps. First, we preprocess the data and eliminate the irrelevant data in order to reduce the dimensions. Second, we use the Principle Component Analysis (PCA) method to transform so many features into less uncorrelated and comprehensive features, which further reduces the dimensions of the problem. At last, an improved artificial bee colony (ABC) clustering algorithm integrated with k-means is employed to cluster cancer patients.



The flowchart of the algorithm

2. Assumptions

1. The data of each feature obeys the normal distribution.

2. There are some internal relationship among the features.

3. The smaller the information entropy of a feature is, the less influence it will have on clustering.

3. Clustering Method

3.1 Data Preprocessing

The genomic data of the LUSC patients can be expressed in the matrix X as below:

$$(\mathrm{X})_{n\times m} = ((\mathrm{X}^1)_{n\times m_1}, (\mathrm{X}^2)_{n\times m_2}, (\mathrm{X}^3)_{n\times m_3}, (\mathrm{X}^4)_{n\times m_4}),$$

where n is the number of LUSC patients and m is the total number of features, X^1, X^2, X^3, and X^4 respectively denote the data matrix of CNN, RPPA, mRNA, and miRNA. Each row means all the feature data of a patient while each colony means one kind of feature data of all the patients. So $\mathrm{m} = \sum_{i=1}^{4} \mathrm{m}_i$. We notice that the data contains huge features (m=21863) while the number of samples is relatively small (n=121). To solve the curse of dimensionality, we have to preprocess the data and reduce the huge dimensions.

To further investigate the data, we find that mRNA and miRNA contains huge features while several hundreds of the features contain useless data—they are all zero and make no difference to the cluster. Based on the assumption 1, each feature $X_i (1 \le i \le m)$ obeys the normal distribution $N(\mu_i, \sigma_i^2)$. We use the information entropy as the assessment index. We intent to eliminate the data with lower information entropy as it contains less information for clustering. However, the information entropy of X_i is proportional to the variance of X_i [1]. So we should eliminate the feature date with lower variance. Specifically, for the four types of the features, we calculate the maximum variance of each type, which is recorded as $\alpha \cdot \mathrm{var}_k (\mathrm{k} = 1,2,3,4)$. Then, the indicator α is introduced to separate the useful data and useless data. The feature data whose variance is lower than the $\alpha \cdot \mathrm{var}_k$ should be eliminated. Fig. 1 and Fig. 2 show the variance of the feature data before and after preprocessing. α is set to 0.5% in the experiment.

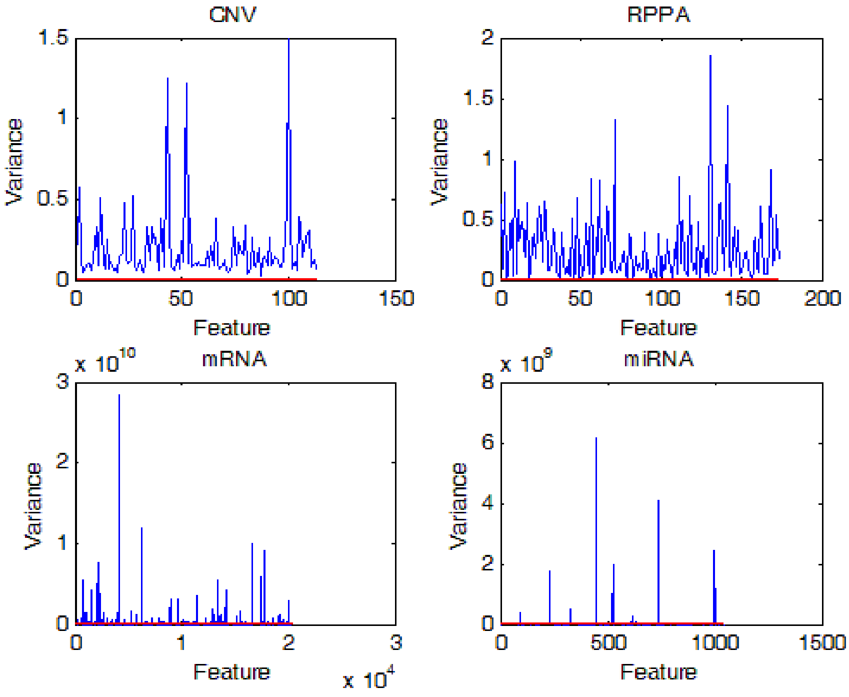


Fig. 1 The variance distribution before preprocessing. (The red line represents the threshold line)

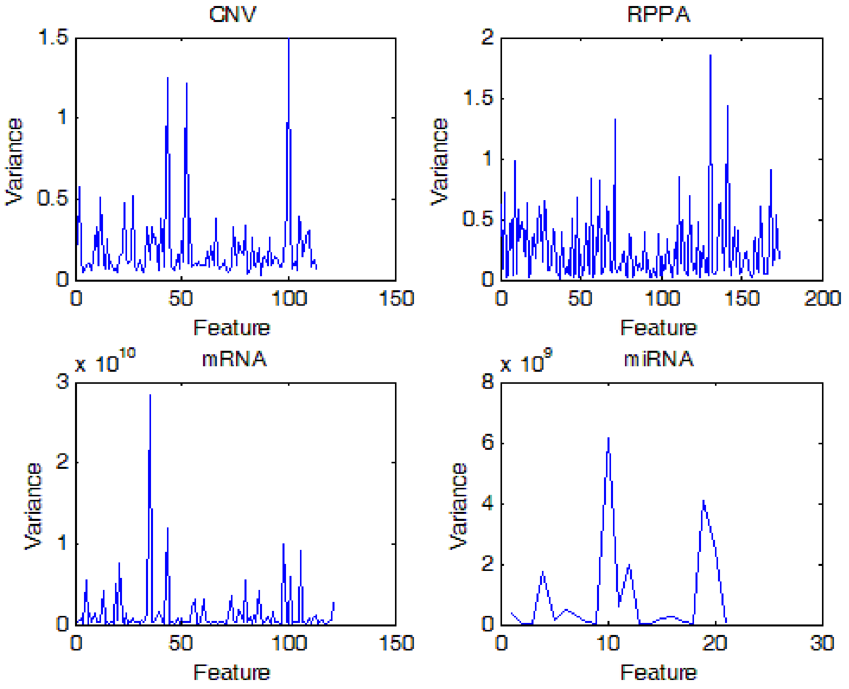


Fig. 2 The variance distribution after preprocessing

3.2 Principle Component Analysis (PCA)

After preprocessing the data, the dimension of the data is effectively reduced (from 21863 to 430), but it is still beyond our expectation. To further reduce the dimension and find the primary features that impact the clustering, the principle component analysis (PCA) method is employed in this scheme. Considering the difference between the four types of the feature. We conduct PCA method on each type of data respectively.

Take the CNN data for example, for the simplification, we standardize the original data by using the formula: $\frac{x_i - E(\mathrm{x}_i)}{\sqrt{D(\mathrm{x}_i)}}$, then, we store the standardized data in matrix X :

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m_1} \\ x_{21} & x_{22} & \cdots & x_{2m_1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm_1} \end{bmatrix} = (\mathrm{X}_1, \mathrm{X}_2, \cdots, \mathrm{X}_{m_1})$$

Next, we calculate the covariance matrix R, $R = (\mathrm{r}_{ij})_{m_1 \times m_1}$, r_{ij} is defined as below:

$$r_{ij} = \frac{\sum_{k=1}^{n} (\mathrm{x}_{ik} - \overline{\mathrm{x}}_i)(\mathrm{x}_{jk} - \overline{\mathrm{x}}_j)}{\sqrt{\sum_{k=1}^{n} (\mathrm{x}_{ik} - \overline{\mathrm{x}}_i)^2 \sum_{k=1}^{n} (\mathrm{x}_{kj} - \overline{\mathrm{x}}_j)^2}}$$

And then, we calculate the eigenvalue of R, the eigenvalues are as follows:

$$\lambda_1 \geq \lambda_2 \cdots \geq \lambda_{m_1} > 0 ,$$

The corresponding new features can be expressed as follows:

$$\begin{cases} Z_1 = a_{11} X_1 + a_{21} X_2 + \cdots + a_{m_1 1} X_{m_1} \\ Z_2 = a_{12} X_1 + a_{22} X_2 + \cdots + a_{m_1 2} X_{m_1} \\ \vdots \\ Z_{m_1} = a_{1m_1} X_1 + a_{2m_1} X_2 + \cdots + a_{m_1 m_1} X_{m_1} \end{cases} ,$$

where $a_i = \begin{pmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{m_1 i} \end{pmatrix}$ is the eigenvector of R. The bigger λ_i is, the more variability information the corresponding Z_i will contain. We select Z_i by the inequality as follows:

$$\frac{\sum_{i=1}^{p}\lambda_i}{\sum_{i=1}^{m_1}\lambda_i} = \frac{\sum_{i=1}^{p}\lambda_i}{n} \geq \beta ,$$

where β is the proportion of the variability information. β is usually fixed by 85%. The original data is replaced by the principle component $Z_1, Z_2, \cdots, Z_p$. Fig. 3 shows the principle component of four types of features.

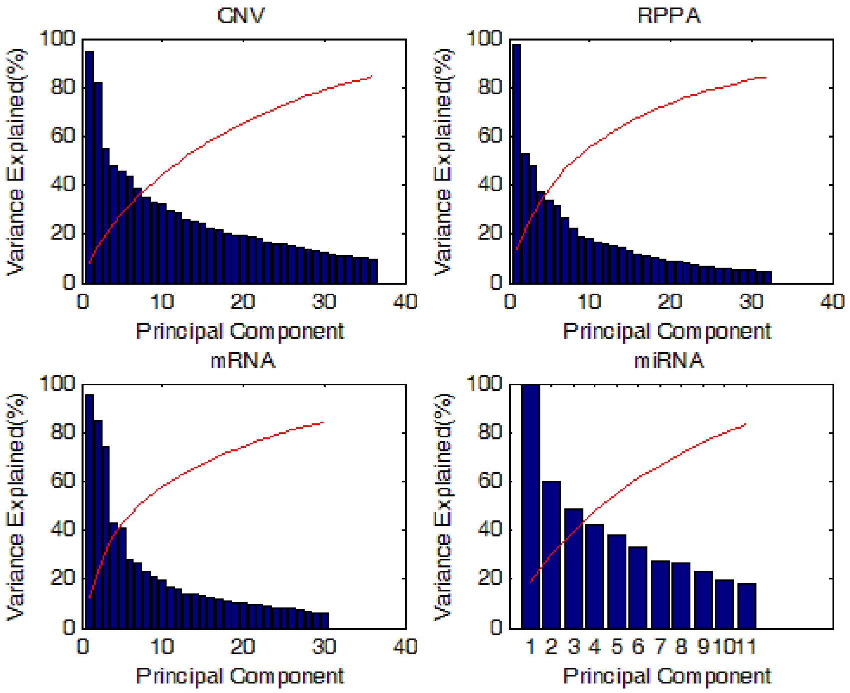


Fig. 3 The illustration of the PCA method. (The blue bars mean the eigenvalue of the corresponding Z_i, the red curves mean the accumulated proportion of the variability information.)

3.3 Clustering

After PCA process, on the one hand, we further reduce the dimension of the data (from 430 to 109), on the other hand, we find the principle component which effect the clustering most.

Then, we propose an improved ABC Clustering algorithm integrated with k-means to handle this problem.

3.3.1 The effective measurement of clustering

Let $S=\{s_1,s_2,\cdots,s_n\}$ denotes the data set after PCA process, each sample s_i has d features (linearly combined by original features), namely $s_i=\{s_{i1},s_{i2},\cdots,s_{id}\}$, our purpose is to classify the data set into K clusters. $C=\{C_1,C_2,\cdots,C_k\}$, the K clusters meet three conditions: $C_i\neq\varnothing$, $\bigcup_{i=1}^{k}C_i=S$, and $C_i\cap C_j=\varnothing(i\neq j)$. We try to cluster the data according to their Euclidian distances.

Before clustering, we should set up a criterion to judge the effectiveness of the clustering. The optimal clustering results should have small cohesion in every cluster and big separation among K clusters. So the CH (Calinski-Harabasz) index is introduced to evaluate the sort result[2].

Let W, B respectively denote the trace of the cohesion matrix and separation matrix:

$$tr(\mathrm{W})=\sum_{i=1}^{k}\sum_{s\in C_i}d^2(\mathrm{s},\mathrm{z}_i)\,,\qquad tr(\mathrm{B})=\sum_{i=1}^{k}n_i d^2(\mathrm{z}_i,\mathrm{z})\,,$$

where z_i is the center of the i*th* cluster, z is the center of all samples, n_i is the sample number of the i*th* cluster, and n is the number of all samples. Then, the CH index is defined as bellows:

$$CH=\frac{tr(\mathrm{B})/(\mathrm{K}-1)}{tr(\mathrm{W})/(\mathrm{n}-K)}$$

The higher the value of CH is, the better the effectiveness of the clustering will be, so our aim is to find the best clustering result with CH index as higher as possible. So the cluster problem is transformed into a multivariable optimization problem, which can be well solved by the improved intelligent swarm algorithm—artificial bee colony (ABC) algorithm.

3.3.2 ABC clustering

From the viewpoint of improved ABC algorithm, a location of a nectar source resembles a feasible solution to the concerned objective function (i.e., the CH index in this paper), and the nectar quantity resembles the corresponding function value. Since the search space (in which all the nectar sources exist) is not infinite, we assume that there are two boundaries on the solution vectors. In general, let $\mathbf{X}=(X^1,X^2,\cdots,X^{Dim})\in\mathbb{R}^{Dim}$ represent a feasible solution, let $\mathbf{Ub}=(U^1,U^2,\cdots,U^{Dim})\in\mathbb{R}^{Dim}$ denote the upper bound of all the solutions and let $\mathbf{Lb}=(L^1,L^2,\cdots,L^{Dim})\in\mathbb{R}^{Dim}$ denote the lower bound. Apparently, $\mathbf{Lb}\le\mathbf{X}\le\mathbf{Ub}$. In this problem, we use the minimum and maximum of all samples as the $\mathbf{Lb}$ and $\mathbf{Ub}$.

At the beginning of this improved ABC clustering algorithm, as many as SN unemployed bees are randomly initialized in the search space. The equation shows how the jth element of the ith employed bee's location $\mathbf{X}_i$ is generated:

$$X_i^j \leftarrow L^j + rand(0,1)\cdot(U^j - L^j),\\ i=1,2,...,\mathrm{SN},\ j=1,2,...,Dim,$$

where $rand(0,1)$ denotes a random number which ranges from 0 to 1 obeying uniform distribution. Thereafter, we choose K unemployed bees with high fitness out of SN directly as employed bees due to the significant influence of the initialization on the quality of the solution and the speed of convergence, the fitness is calculated as following two steps:

1. Clustering all samples $S=\{\mathrm{s}_1,\mathrm{s}_2,\cdots,\mathrm{s}_n\}$ (n=121) into SN clusters with k-means method, namely classify the sample into the corresponding cluster with shortest Euclidian distance between the sample and the SN cluster centers.

2. The fitness of each cluster is calculated in the following equation:

$$Dist_i = J_i / CN_i,\ i = 1,2,\cdots,N\ ,\ \ fitness_i = 1/(1+\mathrm{Dist}_i)\ ,$$

where CN_i denotes the number of samples belong to the i*th* cluster, $J_i = \sum_{xj \in C_i} d(\mathrm{x}_j, \mathrm{C}_i)$ denotes the sum distance between the center of the i*th* cluster to the internal cluster samples. $Dist_i$ represents the average internal cluster distance.

After we select K unemployed bees with the highest fitness as the initial cluster centers, the CH index is also calculated. Then an iterative optimization process gets started.

In each cycle of iteration, there are three phases to execute, namely the employed-bee phase, onlooker-bee phase and the re-start phase.

In the employed-bee phase, as many as K employed bees should find new positions in the search space to search. They generate new search positions through sharing locations with each other. Let's take the ith employed bee for example. Assume that currently it stays at $\mathbf{X}_i = (X_i^1, X_i^2, \cdots, X_i^{Dim})$. Denote $\mathbf{X}_i^* = (X_i^{*1}, X_i^{*2}, \cdots, X_i^{*Dim})$ as the to-be-computed search location for it. First, as many as $trial(i)$ out of all the Dim dimensions in $\mathbf{X}_i$ should be randomly selected. Here, $trial(i) \in [1,\ Dim]$ is an integral scalar that will be formally introduced soon. For each of the selected dimensions, employed bee i needs to "crossover and mutate" with another employed bee, which should be randomly chosen. The following equation shows how the ith employed bee takes usage of the kth employed bee's location to generate the new search direction in the jth dimension:

$$X_i^{*j} \leftarrow X_i^j + rand(-1,1) \cdot (X_k^j - X_i^j) \cdot \frac{trial(i)}{trial(i) + trial(k)},$$
$$k \in \{1, 2, \ldots, \mathrm{K}\},\ k \neq i.$$

When $\mathbf{X}^*$ is determined, a process of k-means cluster is carried out and the new index $CH(\mathbf{X}^*)$ is calculated. And then, $CH(\mathbf{X}^*)$ is compared to $CH(\mathbf{X})$. If $CH(\mathbf{X}^*) > CH(\mathbf{X})$, the ith employed bee will fly to the new location $\mathbf{X}^*$, i.e., $\mathbf{X}_i \leftarrow \mathbf{X}_i^*$. Also, $trial(i)$ should be reset to 1, i.e., $trial(i) \leftarrow 1$. Otherwise, when $CH^* \le CH$, the ith employed bee will remain at $\mathbf{X}_i$ but $trial(i)$ will add one, i.e., $trial(i)+1 \leftarrow trial(i)$ in that case. Through the principle how $trial(i)$ is calculated, it is not difficult to see that $trial(i)$ records the number of inefficient trial times of the ith employed bee.

When all the employed bees have updated their locations, the employed-bee phase in this current iteration is completed, which is followed by an onlooker-bee phase.

A roulette selection procedure guides those as many as (SN-K) onlooker bees to choose “qualified” employed bees to search around. A probability index P is calculated according to the following equations to show the search quality of all the employed bees.

$$P(i) = \frac{fitness(i)}{\sum_{j=1}^{K} fitness(j)},\ i = 1, 2, \ldots, \mathrm{K},$$

Let’s take the ith onlooker bee for example. In order to find an employed bee to follow around, a comparison is made between $rand(0,1)$ and $P(1)$. If $P(1) \ge rand(0,1)$, the ith onlooker bee will search around the 1st employed bee; otherwise, an additional comparison between another random number $rand(0,1)$ and $P(2)$ will be made. If all the $P(j),\ j \in \{1, 2, \ldots, \mathrm{K}\}$ happen to be smaller than $rand(0,1)$, such process is repeated until one $P(j)$ that satisfies $P(j) \ge rand(0,1)$ can be found. Then, the corresponding jth employed bee will be chosen to be searched around locally.

First, unlike that in the employed-bee phase, only one (randomly chosen) dimension of the onlooker bee’s solution vector is involved in a “crossover and mutation” process. The following equation shows how the ith onlooker bee takes usage of the mth employed bee's location to generate a new search location around the jth in the kth dimension:

$$Y_j^k \leftarrow X_j^k + rand(-1,1)\cdot(X_m^k - X_j^k)\cdot\frac{trial(j)}{trial(j)+trial(m)},$$
$$m\in\{1,2,\ldots,\mathrm{K}\},\ m\neq j.$$

When the location of the onlooker bee $\mathbf{Y}_i=\left(X_j^1,\ldots,X_j^{k-1},Y_j^k,X_j^{k+1},\ldots,X_j^{Dim}\right)$ is determined, a similar process of k-means cluster is carried out and a comparison between $CH(\mathbf{Y})$ and $CH(\mathbf{X})$ is made. If $CH(\mathbf{Y})>CH(\mathbf{X})$, the jth employed bee will abandon the current location $\mathbf{X}_j$ and fly to $\mathbf{Y}_i$, i.e., $\mathbf{X}_j\leftarrow\mathbf{Y}_i$. Also, $trial(j)$ is reset to 1. Otherwise, when $CH(\mathbf{Y})\leq CH(\mathbf{X})$, $trial(j)$ will add one. When all the onlooker bees have searched locally around their corresponding employed bee, the onlooker-bee phase is completed, which is followed by a re-start phase at the end of each iteration.

In the re-start phase, two things should be done. First, any $trial(i)$ that has exceeded Dim will be reset to Dim. Second, the average of $trial$ (i.e., $\frac{1}{\mathrm{K}}\sum_{i=1}^{K}trial(i)$) is compared to $\alpha_{odr}\cdot Dim$, where $\alpha_{odr}\in(0,1)$ is a user-specified scalar. If $\alpha_{odr}\cdot Dim<\frac{1}{K}\sum_{i=1}^{K}trial(i)$, the whole employed bee swarm is considered to be not working efficiently to a degree of α_{odr}. Then, $100\cdot\alpha_{odr}\%$ of all the employed bees will be fully re-initialized. Also, the corresponding $trial$ indices are reset to 1. On the other hand, if $\alpha_{odr}\cdot Dim\geq\frac{1}{K}\sum_{i=1}^{K}trial(i)$, nothing else will do and the current cycle of iteration is terminated directly.

When the iteration number reaches a pre-defined maximum cycle number MCN, the whole optimization process is accomplished.

3.3.2 ABC Cluster Results

All the simulations are carried out in a Matlab R2012a environment and executed on an Intel Core 2 CPU with 6 GB RAM running at 1.6GHz under Windows 8. The related parameters are list in the Table 1 below.

Table 1. Notations and Settings of User-Specific Parameters

Parameter	Description	Setting
MCN	maximal cycle number	10000
SN	swarm population	40
α_{odr}	overall degradation rate in this algorithm	0.9
Dim	dimension of the optimization position	109

In order to determine the best number of clusters, we set up a series of experiments with different K to evaluate the best clustering which has a relatively high CH index. Fig. 4 and Table. 2 show the convergence curve of the CH index and the clustering result with different K in the clustering experiments.

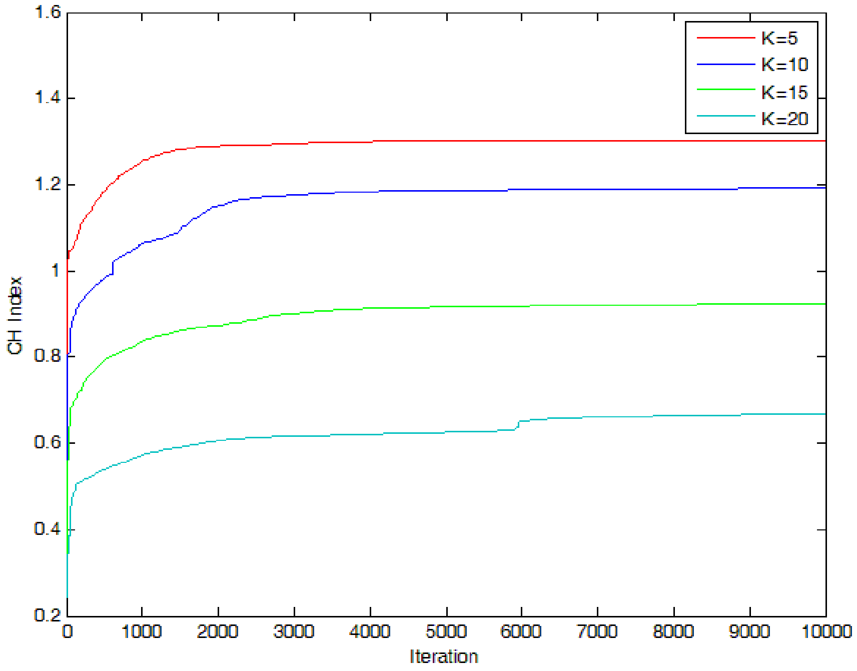


Fig. 4. The CH index curves with different clusters K (K=5, 10, 15, 20).

Table 2. Clustering result

K	Value of CH index	Clustering result (total=121)
5	1.3023	26, 23, 23, 25,24
10	1.1917	11, 12, 12, 13, 11, 12, 12, 13, 13, 12
15	0.9254	1, 10, 11, 11, 11, 10, 10, 10, 11, 10, 12, 1, 10, 2, 1
20	0.6683	1, 11, 12, 13, 7, 10, 11, 11, 11, 1, 4, 9, 4, 10, 1, 2, 1, 1, 1, 0

(The corresponding Patient No is attached in the appendix section for k=5)

In the experiment, we selected 4 different K (from 5 to 20) to simulate the clustering. The result shows that clustering with small K tents to have a relatively higher CH index. The selection of K will be discussed in detail in the next section.

4. Discussions

In this section, we will further analyze the simulation results that are shown in the preceding section.

4.1 The rationality of reducing dimension

In section 2 and 3, we reduced the dimension from 21863 to 109 by two steps. First, we eliminate the features with low variance. Although these features may have relation with the cancer, but they make little difference to the clustering. Second, in the PCA process, we transformed the original features into some new types. Although it may become harder for us to find the relation between the cancer and the original features, we can still find out the important original features according to the eigenvalues, moreover, the new types of features are independent, which is convenient for clustering using our proposed algorithm.

4.2 The parameters of the model

We use CH index to evaluate the effectiveness of the clustering result. CH is an effective index for it considers both cohesion in each cluster and separation among all the clusters. The value of CH index is directly related with the cluster number K.

As the number of patients is 121, so the range of K is from 1 to 121. However, from the clustering result in Table 2, we think we should not choose a big K, for the low CH index and there may be only one or zero patients in some clusters. When K is small (K=5 and 10), the clustering result is more balanced and there is no isolated patient. So, K=5 is the best choice among the experiments.

Besides, the total number of bees SN is fixed by 40 in the experiments. Although each employed bee represents a cluster center, the SN should be more than the number of the patients theoretically, we fixed it by 40 for reducing the computing load and there is little meaning for generating too many clusters.

4.3 The robustness of the model

In order to judge the robustness of our clustering algorithm, we fix the cluster number K and repeat the simulation for several times. As we know, for each simulation, the difference is the initial value of the cluster centers. As the iteration goes, the position of the cluster centers are updating and evolving. A robust clustering algorithm should have as little dependence on the initial values as possible. Table 3 shows the clustering result.

Table 3. Clustering result

NO	Value of CH index	Clustering result (K=5,total=121)
1	1.3023	26, 23, 23, 25, 24
2	1.2904	25, 24, 24, 24, 24
3	1.2713	26, 23, 23, 25, 24
4	1.2722	25, 24, 24, 24, 24
5	1.2822	25, 24, 23, 25, 24

For each simulation, the clustering result shows little difference which indicates the robustness of the clustering algorithm.

6. Conclusions

In this paper, we develop an improved ABC clustering algorithm based on the principle of k-means. K-means method is sensitive to the initial clustering centers and easy to be trapped by local optimum. So we integrate it with the improved intelligent swarm optimization algorithm. The main improvement in the algorithm is as follows:

First, the principle of “trial” is introduced to count the inefficient times of searching and control the search range. The employed bees with high “trial” tend to have a large searching range which can accelerate the convergence speed.

Second, the scalar $\alpha_{odr} \in (0,1)$ is introduced to evaluate the whole inefficiency of the employed bees, when it exceeds α_{odr} , some of the employed bees will be fully initialized, which will help to avoid the local optimum.

However, our algorithm still has some limitation. For example, the initial value of clustering centers are generated randomly which lacks enough rationality. Besides, when K is a big number, it may generate an empty cluster due to the isolation of the initial clustering center. Also, we admit that we may have not find a best K, but according to the experiment, the smaller the K is, the bigger the value of CH index will be.

7. Reference

[1] J. J. Song, Analysis of Statistical Information [M]. Tianjin: Naikai UP, 2005.

[2]T. Calinski, and J. Harabasz, A Dendrite Method for Dluster Analysis [J]. Communication in Statistics, 1974, 3(1): 1-27.

[3]J. Liao, Z. F. Hao, Data Mining and Mathematical Modeling[M]. Beijing: Defense Industry Press, 2010.

Appendix

Cluster No	Patient No
1	TCGA-66-2737,TCGA-21-1076,TCGA-22-5474,TCGA-22-5478,TCGA-43-2581,TCGA-60-2706,TCGA-22-5489,TCGA-66-2767,TCGA-60-2709,TCGA-34-2608,TCGA-43-5668,TCGA-66-2777,TCGA-66-2770,TCGA-33-6737,TCGA-60-2714,TCGA-60-2710,TCGA-60-2713,TCGA-60-2712,TCGA-56-5897,TCGA-18-4721,TCGA-66-2790,TCGA-22-4596,TCGA-22-1011,TCGA-39-5034,TCGA-39-5031,TCGA-39-5030
2	TCGA-22-4607,TCGA-85-6560,TCGA-37-4133,TCGA-37-4135,TCGA-43-2578 TCGA-22-5483,TCGA-39-5011,TCGA-34-5929,TCGA-37-3789,TCGA-37-5819 TCGA-60-2720,TCGA-33-6738,TCGA-21-1083,TCGA-21-5787,TCGA-60-2711 TCGA-66-2783,TCGA-66-2785,TCGA-37-4141,TCGA-66-2791,TCGA-39-5029 TCGA-66-2795,TCGA-39-5021,TCGA-39-5035
3	TCGA-66-2758,TCGA-21-1075,TCGA-21-1072,TCGA-34-5927,TCGA-22-5485 TCGA-33-4532,TCGA-22-5491,TCGA-22-5473,TCGA-66-2800,TCGA-66-2766 TCGA-77-6843,TCGA-77-6842,TCGA-21-1080,TCGA-33-4586,TCGA-33-4582 TCGA-34-2596,TCGA-60-2719,TCGA-66-2794,TCGA-39-5027,TCGA-22-4591 TCGA-66-2727,TCGA-43-3394,TCGA-39-5037
4	TCGA-22-0940,TCGA-37-3792,TCGA-43-6771,TCGA-21-1078,TCGA-22-5482 TCGA-60-2696,TCGA-34-5232,TCGA-90-6837,TCGA-34-5234,TCGA-60-2725 TCGA-22-5492,TCGA-22-5472,TCGA-22-5477,TCGA-77-6844,TCGA-66-2768 TCGA-21-5784,TCGA-94-7033,TCGA-66-2781,TCGA-66-2782,TCGA-66-2788 TCGA-39-5028,TCGA-39-5024,TCGA-34-5241,TCGA-46-6025,TCGA-56-6546
5	TCGA-85-6561,TCGA-85-6798,TCGA-43-6770,TCGA-21-1079,TCGA-22-5480 TCGA-60-2698,TCGA-39-5019,TCGA-34-5231,TCGA-34-5236,TCGA-34-5239 TCGA-60-2724,TCGA-60-2721,TCGA-22-5471,TCGA-22-5479,TCGA-77-6845 TCGA-34-7107,TCGA-60-2708,TCGA-21-5786,TCGA-21-5782,TCGA-33-4566 TCGA-33-4547,TCGA-56-6545,TCGA-46-6026