BMC Bioinformatics

BioMed Central

Research

Genetic weighted k-means algorithm for clustering large-scale gene expression data

Fang-Xiang Wu[1,2]

Address: [1]Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, SK, S7N 5A9, Canada and [2]Division of Biomedical Engineering, University of Saskatchewan, Saskatoon, SK, S7N 5A9, Canada

Email: Fang-Xiang Wu - faw341@mail.usask.ca

from Symposium of Computations in Bioinformatics and Bioscience (SCBB07)
Iowa City, Iowa, USA. 13–15 August 2007

Published: 28 May 2008

BMC Bioinformatics 2008, **9**(Suppl 6):S12 doi:10.1186/1471-2105-9-S6-S12

This article is available from: http://www.biomedcentral.com/1471-2105/9/S6/S12

Abstract

Background: The traditional (unweighted) k-means is one of the most popular clustering methods for analyzing gene expression data. However, it suffers three major shortcomings. It is sensitive to initial partitions, its result is prone to the local minima, and it is only applicable to data with spherical-shape clusters. The last shortcoming means that we must assume that gene expression data at the different conditions follow the independent distribution with the same variances. Nevertheless, this assumption is not true in practice.

Results: In this paper, we propose a genetic weighted K-means algorithm (denoted by GWKMA), which solves the first two problems and partially remedies the third one. GWKMA is a hybridization of a genetic algorithm (GA) and a weighted K-means algorithm (WKMA). In GWKMA, each individual is encoded by a partitioning table which uniquely determines a clustering, and three genetic operators (selection, crossover, mutation) and a WKM operator derived from WKMA are employed. The superiority of the GWKMA over the k-means is illustrated on a synthetic and two real-life gene expression datasets.

Conclusion: The proposed algorithm has general application to clustering large-scale biological data such as gene expression data and peptide mass spectral data.

Background

Clustering is defined as a process of partitioning a set of objects (patterns) into a set of disjoined groups (clusters). Its goal is to reduce the amount of data by categorizing or grouping similar data items together and obtain useful information. Clustering methods can be divided into two basic types: hierarchical and partitional clustering [1]. Within each type there exists a wealth of subtypes and different algorithms. Hierarchical clustering proceeds successively either by merging smaller clusters into larger ones (bottom-up), or by splitting larger clusters into smaller clusters (top-down). The hierarchical clustering methods differ in the rules used to decide which two small clusters are merged or which large cluster is split. The final result of the algorithm is a binary tree of clusters called a dendrogram, which shows how the clusters are related to each other. By cutting the dendrogram at a desired level, a clus-

tering of objects in a dataset into disjoint groups is obtained.

On the other hand, partitional clustering – k-means, for example – attempts to directly divide a dataset into a number of disjoint groups. All partitional clustering algorithms need as input the number of clusters and a cost (criterion) function to define the quality of a partition. The partitional clustering method aims at optimizing the cost function to minimize the dissimilarity of the objects within each cluster, while maximizing the dissimilarity of different clusters. In general, the partitional clustering algorithms are iterative and hill-climbing, and thus they are sensitive to the choice of the initial partition. Furthermore, since the associated cost functions are nonlinear and multimodal, usually these algorithms converge to a local minimum. The algorithms based on combinatorial optimization such as integer programming, dynamic programming and, branch-bound methods are too expensive since the number of partitions of n objects into k clusters is $O(k^n)$.

Genetic algorithms (GA) [2], inspired by natural evolution of genes, offer heuristic solutions to some optimization problems. The algorithm typically starts with a set of solutions (randomly generated) called the population and creates successive, new generations of the population by genetic operations such as natural selection, crossover, and mutation. Natural selection is performed based on the fitness (related to the cost function) of an individual. For an individual, the better its fitness, the more chances it has to survive in the next generation. Crossover is performed by certain crossover rule and mutation aims at changing an individual by a user-specified mutation probility. The intuition underlying the approach is that each new population will be better than the previous one. Actually it has been proved [3] that a canonical GA converges to the global optimum with probability 1.

A GA is highly dependent on the coding of the solutions (individuals). In the context of weighted k-means, a natural representation of a solution is a pair of variables (partitional string, cluster centroids). The partitional string describes for each object the index of cluster which it belongs to. The cluster centroids are representative objects of the clusters and their attributes are found by averaging the corresponding attributes among the objects in a particular cluster. These two variables depend on each other such that if one of them is given, the other one can be uniquely constructed. Since the cluster centroids generally are real numbers, it might be very difficult to encode them. On the other hand, a direct encoding for partitional strings is a simple problem.

Genetic algorithms have been previously considered for clustering problems [4-11]. Often genetic algorithms are not hybridized with k-means algorithms [5,6,9,11] and thus their rates of convergence were very slow. On the other hand, when GA are hybridized with k-means algorithms [7,8,10], the resultant algorithms inherit some drawbacks of unweighted k-means algorithms, for example, that the resultant clusters are spherical-shape. Further, if the inherent structure of the clusters in the data is not spherical shaped, such algorithms can not give the correct results

In this paper, we propose a genetic weighted k-means algorithm (GWKMA). This is a hybrid approach to combining a GA with the weighted k-means algorithm (WKMA) [12,13] and partially remedies drawbacks of other attempts [4-11]. The GWKMA encode the solutions by partitional strings and employs three genetic operations – natural selection, crossover and mutation – and one WKM operation derived from the weighted k-means algorithm (WKMA).

Methods

WKMA

In a general sense, a k-partitioning algorithm takes as input a set $D = \{x_1, x_2 \cup, x_n\}$ of n objects and an integer K, and outputs a partition of D into exactly K disjoint subsets $D_1, \cup, D_K$. Denote such a partition by Δ. Each of the subsets is a cluster, with objects in the same cluster being somehow more similar to each other than they are to all subjects in other different clusters. One way to make the determination of Δ into a well-defined problem is to define a cost function which measures the clustering quality of any partitions of a dataset.

In this paper, each attribute of an object (gene) is expressed as a real number and thus each object may be described by a real number row vector of dimension d, where d is the number of attributes of an object. Assume that all objects in the dataset have the same number of attributes, i.e. no missing data. Let $(x_i, i = 1, \cup, n)$ be a dataset of n objects. Let x_{ij} denote the jth attribute of object x_i. $X = (x_{ij})$ is called an attribute matrix of object set D. For the predefined number K of clusters, the cost function for a weighted k-means clustering technique may be defined by

$$J_G(\Delta) = \sum_{k=1}^{K} \sum_{x_i \in D_k} (x_i - \bar{m}_k) G (x_i - \bar{m}_k)' \qquad (1)$$

where

$$\bar{m}_k = \frac{1}{n_k} \sum_{x_i \in D_k} x_i \qquad (2)$$

n_k and m_k are the mean and the number of objects in D_k, respectively, and G is a weighted matrix which is a symmetrical positive. The objective of a weighted k-means algorithm is to find an optimal partition expressed by Δ^* and a symmetrical positive matrix G^* satisfying equation (3) such that

$$J_{G^*}(\Delta^*) = \min_{\Delta}\{J_{G^*}(\Delta)\} \quad (3)$$

Obviously, given a partition Δ, the value of $J_G(\Delta)$ change with the multiplication of a weighted matrix G. Therefore the weighted matrix must be normalized. In this study the determinant of G is set to be 1, i.e.

$$(\det(G)) = 1 \quad (4)$$

For fixed $G = I$ in equation (1), condition (4) is satisfied automatically, and equations (1) and (3) become the cost function and optimal objective of a traditional k-means algorithm, respectively.

For a fixed partition, we wish to determine G such that the cost function (1) is optimized under the normalization condition (4). To do that, we form the Lagrangian function

$$L(G, \lambda) = \sum_{k=1}^{K}\sum_{x_i \in D_k}(x_i - \bar{m}_k)G(x_i - \bar{m}_k)' - \lambda(\det(G)) - 1) \quad (5)$$

and calculate its derivatives with respect to G

$$\frac{\partial}{\partial G}L(G, \lambda) = \sum_{k=1}^{K}\sum_{x_i \in D_k}(x_i - \bar{m}_k)'(x_i - \bar{m}_k) - \lambda G^{-1}(\det(G)) \quad (6)$$

Equating the derivative to zero and using the auxiliary condition (4) lead to

$$W = \lambda G^{-1}(\det(G)) = \lambda G^{-1}$$

where $W = \sum_{k=1}^{K} W_k$ and $W_k = \sum_{x_i \in D_k}(x_i - \bar{m}_k)'(x_i - \bar{m}_k)$ is the within-group variance of cluster k ($(k = 1, \cup, K)$, and

$$\lambda = (\det(W))^{1/d}$$

Finally, we have

$$G = W^{-1}(\det(W))^{1/d} \quad (7)$$

Note that W is dependent on partition Δ. To avoid ambiguousness, denote W induced by Δ as $W(\Delta)$. Substituting (7) into (1) leads to $J(\Delta) = d(\det(W(\Delta)))^{1/d}$. As d is a constant for a given dataset, the cost function of a weighted k-mean clustering is reduced to

$$J(\Delta_o) = (\det(W(\Delta)))^{1/d} \quad (8)$$

Thus the objective of a weighted k-mean algorithm is simplified as finding an optimal partition expressed by Δ_o which minimizes

$$J(\Delta_o) = \min_{\Delta}(\det(W(\Delta)))^{1/d} \quad (9)$$

There are $O(k^n)$ different partitions of n objects into exactly k clusters [1]. It is impractical to using an exhaustive search for the solution to clustering a large-size gene expression dataset. To overcome this problem, a heuristic approach is usually considered. The basic idea in the heuristic approach is to randomly select an initial partition and then move objects between groups if such moves make J significantly smaller.

Now consider how the cost function J changes when an object x currently in cluster D_i is tentatively moved to a different cluster D_j. Let $\Delta = (D_1, \cup, D_k)$, $\Delta' = (D_1, \cup, D_i\backslash\{x\}, \cup D_k)$, and $\Delta'' = (D_1, \cup, D_i\backslash\{x\}, \cup, D_j \cup \{x\}, \cup, D_k)$ $(i \neq j)$. Obviously the condition for successfully moving x from D_i into D_j is

$$\det(W(\Delta'')) < \det(W(\Delta)) \quad (10)$$

From the definitions, it follows that

$$W(\Delta) = W(\Delta') + \frac{m_i}{m_i - 1}(x - \bar{x}_i)'(x - \bar{x}_i) \quad (11)$$

$$W(\Delta'') = W(\Delta') + \frac{m_j}{m_j + 1}(x - \bar{x}_j)'(x - \bar{x}_j) \quad (12)$$

Condition (10) is reduced to

$$\frac{m_j}{m_j + 1}(x - \bar{x}_j)[W(\Delta')]^{-1}(x - \bar{x}_j)' < \frac{m_i}{m_i - 1}(x - \bar{x}_i)[W(\Delta')]^{-1}(x - \bar{x}_i)' \quad (13)$$

since $\det(A + \beta\gamma'\gamma) = \det(A)(1 + \beta\gamma A^{-1}\gamma')$ for any $d \times d$ invertible matrix A, any d – dimensional row vector γ, and any number β.

If reassignment is profitable, the greatest decrease in the cost function is obtained by selecting the cluster for which

$\frac{m_j}{m_j+1}(x-\bar{x}_j)[W(\Delta')]^{-1}(x-\bar{x}_j)'$ is minimal. This leads to the iteratively optimal weighted k-means algorithm (WKMA) shown in Figure 1.

GWKMA

As the WKMA is sensitive to initial partitions and its result is prone to the local minima, this paper proposes a genetic weighted k-means algorithm (GWKMA), shown in Figure 2. The GWKMA is a hybridization of GA and WKMA, including the three genetic operators in general GA and a WKM operator derived from WKMA. In the following we specify in details the encoding, selection, crossover, mutation, and WKM operators.

Encoding

In the literature [5,6,8], solutions (individuals) are encoded by the centers of clusters. Note that the centers of clusters are real numbers for general cluster tasks and the encoding of the real number in GA algorithms is hard and may degrade the accuracy of the solutions.

We use a partitional string to express a solution to a clustering. A partitional string is an integer string over the set $\{1,\cup, K\}$, on which each position corresponds to an object and the number in a position represents the cluster to which the corresponding object is assigned. Thus, the search space consists of all integer strings s_Δ with length n over the set $\{1,\cup, K\}$. A population is expressed by a set of partitional strings representing its individuals (solutions),

Weighted K-means algorithm (WKMA)

$[\Delta_o, J(\Delta_o)] = WKMA(\Delta, X, K)$

Input: an initial partition, Δ; the attribute matrix, X; the number of clusters, K

Output: a best partition, Δ_0 and its cost function value, $J(\Delta_0)$

1. **Compute** $\bar{m}_k$ using formula (2) and $J(\Delta)$ using formula (8) for an initial partition Δ.
2. **repeat**
3. **for** $i = 1$ to n
4. $k \leftarrow$ the index of the cluster where x_i belongs
5. **if** $n_k \neq 1$ **then** compute

$$\rho_l = \begin{cases} \frac{m_l}{m_l+1}(x-\bar{x}_l)[W(\Delta')]^{-1}(x-\bar{x}_l)', & l \neq k \\ \frac{m_l}{m_l-1}(x-\bar{x}_l)[W(\Delta')]^{-1}(x-\bar{x}_l)', & l = k \end{cases}$$

6. **if** $\rho_j = \min_l \rho_l < \rho_k$ **then** move x_i to D_j, adjust Δ, and re-compute $J, \bar{m}_j, \bar{m}_k$
7. **end for**
8. **until** no significant change in J in n consecutive attempts
13. **return** Δ_0 and $J(\Delta_0)$

Figure 1
Iterative optimal k-means algorithm.

Genetic Weighted K-means Algorithm (GWKMA)

Input: Attribute Matrix, X; Number of clusters, K; Mutation probability, Pm; Population size, N; Number of generation, GEN.

Output: the resultant partition Δ_o and the value of its cost function $J(\Delta_o)$.

1. **Initialize** the population Δ^* with the size N

   /* Δ^* is a set of partition string of a population */

2. $[\Delta^*, J(\Delta^*)] = WKM(\Delta^*, X, K, N)$

3. Re-order individuals such that the first individual is the optimal one in population , and set $\Delta_o = \Delta_1$, $JE(0) = J(\Delta_o)$, and $g = 1$

3. **While** ($g \le GEN$)

4. $\tilde{\Delta}^* = Selection(\Delta^*, X, K, N)$

5. $\Delta^* = Crossover(\tilde{\Delta}^*, N)$

6. $\Delta^* = Mutation(\Delta^*, Pm, K, N)$

7. $[\Delta^*, J(\Delta^*)] = WKM(\Delta^*, X, K, N)$

8. Find the optimal individual in population Δ^*, and denote it by $\overline{\Delta}$

9. **If** $J(\Delta_o) > J(\overline{\Delta})$,

   then $\Delta_o = \overline{\Delta}$, and set $JE(g) = J(\overline{\Delta})$

   else $JE(g) = JE(g-1)$;

   /* where $JE(g)$ stands for the value of the cost function of the best partition up to the g th generation from the beginning of the algorithm */

10. Re-order individuals such that $\Delta_o = \Delta_1$;

11. $g = g + 1$;

12. **End while**

13. **Return** $JE(GEN) = J(\Delta_o)$ and the resultant partition Δ_o.

Figure 2
Genetic weighted k-means algorithm (GKMA).

denoted by $\tilde{\Delta}^*$ or Δ^*. One may set some additional conditions to refine the search space. For example, to avoid a singular clustering, one may impose the constraint that each element in the set $\{1,\cup, K\}$ appears at least once in s_Δ.

The advantage of encoding the centers of clusters is that the resultant clusters from GA clustering are convex. GWKMA encodes the solutions (individuals) by integer strings (partitional strings). This simplifies the encoding of GA and does not degrade the accuracy of the solutions. Since GWKMA includes the weighted k-means operator, the resultant clusters from GWKMA are still convex.

Selection operator

$\tilde{\Delta}^* = Selection(\Delta^*, X, K, N)$. For convenience of the manipulation, GWKMA always assigns the best individual found over time in the population to individual 1 and copies it to the next population. Operator $\tilde{\Delta}^* = Selection(\Delta^*, X, K, N)$ selects $(N\text{-}1)/2$ individuals from the previous population according to the probability distribution given by

$$P_s(s_{\Delta i}) = F(s_{\Delta i}) \Big/ \sum_{i=1}^{N} F(s_{\Delta i}) \qquad (14)$$

where N (odd positive integer) is the number of individuals in a population, $s_{\Delta i}$ is the partitional string of individual i, and $F(s_{\Delta i})$ represents the fitness value of individual i in the current population. Fitness here is defined as

$$F(S_\Delta) = TJ - J(s_\Delta) \qquad (15)$$

where $J(s_\Delta)$ is calculated by (8), and TJ is calculated by the following formula

$$TJ = \det(S) = \det\Big(\sum_{x\in D}(x-\bar{m})'(x-\bar{m})\Big) \qquad (16)$$

and where $\bar{m} = \frac{1}{n}\sum_{x\in D} x$. It is evident that $TJ \geq J(s_\Delta)$ for any s_Δ in the problem. Note that there are $(N\text{-}1)/2+1$ individuals in $\tilde{\Delta}^*$.

Crossover operator

$\Delta^* = Crossover(\tilde{\Delta}^*, N)$. The intention of the crossover operation is to create new (and hopefully better) individuals from two selected parent individuals. In GWKMA, of two parent individuals, one always is the first individual that is the optimal individual found over time, and another is one of the selected $(N\text{-}1)/2$ individuals from the parent population other than the first individual. In this paper, the crossover operator adopts the single-point crossover method for simplicity. Note that after the crossover operation population Δ^* has N individuals.

Mutation operator

$\Delta^* = Mutation(\Delta^*, Pm, K, N)$. Each position in a coding string is randomly selected with a user-set mutation probability Pm, and the number in the selected position is uniformly randomly replaced by another integer from the set $\{1,\cup K\}$. In other work [14], such a mutation depends on the distance of the corresponding object from the cluster centoids. Actually such a complex strategy is not necessary because the WKM operator will be used. To avoid any singular partition (containing an empty cluster), the mutation operator also randomly assigns one object to a cluster which is empty after all genetic operations.

WKM operator

$[\Delta^*, J(\Delta^*)] = WKM(\Delta^*, X, K, N)$: The WKM operator is obtained by calling WKMA for each individual s_Δ in population Δ^*. It is sufficient to run the repeat loop in AKMA for several times. Note that $J(\Delta^*)$ is an N-dimensional vector, each component of which corresponds to the value of the cost function of an individual in population Δ^*. In other works [7-9], several different k-means operators were employed, and their functions are similar to that of WKMA. However, those k-means algorithms are neither iteratively optimal nor weighted.

Evaluation

The term "evaluation of a clustering method" usually refers to the ability of a given method to recover true clusters in a dataset. There have been several attempts to evaluate a clustering method on theoretical grounds [14,15]. Since a clustering result can be considered as a partition of objects into a number of groups, for evaluating a clustering method it is necessary to define a measure of agreement between two partitions of the same dataset. In the clustering literature, measures of agreement between partitions are referred to as external indices. Several such indices have been described [15,16]. This paper adopts the adjusted Rand index (ARI).

Consider two partitions of N objects: the r-cluster partition $U = \{u_1,\cup u_r\}$ and the s-cluster partition $V = \{v_1,\cup, v_s\}$. One may construct a contingency table (Table 1), where entry n_{ij} is the number of objects that are both in clusters u_i and v_j, $i = 1,\cup, r$, $j = 1,\cup, s$. Let $n_{i.} = \sum_{j=1}^{s} n_{ij}$ and $n_{.j} = \sum_{i=1}^{r} n_{ij}$ denote the sum of row $i\{i = 1,\cup, r\}$ and the sum of column j $(j = 1,\cup, s)$ in the contingency table,

Table 1: Contingency table for two partitions of *n* objects

	v_1	v_2		v_5	Total
u_1	n_{11}	n_{12}	∪	n_{1s}	$n_{1.}$
u_2	n_{21}	n_{22}	∪	n_{2s}	$n_{2.}$
⋮	⋮	⋮		⋮	⋮
u_r	n_{r1}	n_{r2}	∪	n_{rs}	$n_{r.}$
Total	$n_{.1}$	$n_{.2}$	∪	$n_{.s}$	$n_{..} = n$

respectively, and let $Z = \sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij}^2$ and $V = \binom{N}{2} = N(N-1)/2$ (the number of pairs of N objects). Based on the contingency matrix of two partitions, the ARI is defined as [16,17]:

$$ARI = \frac{\sum_{i=1}^{r}\sum_{i=1}^{s}\binom{n_{ij}}{2} - \frac{1}{V}\sum_{i=1}^{r}\binom{n_{i.}}{2}\sum_{i=1}^{s}\binom{n_{.j}}{2}}{\frac{1}{2}\left[\sum_{i=1}^{r}\binom{n_{i.}}{2} + \sum_{i=1}^{s}\binom{n_{.j}}{2}\right] - \frac{1}{V}\sum_{i=1}^{r}\binom{n_{i.}}{2}\sum_{i=1}^{s}\binom{n_{.j}}{2}} \quad (17)$$

The ARI is an adjusted Rand index [18] in that its expected value is 1 when they matched perfect and 0 when the two partitions are selected at random. Accordingly, the large value of ARI indicates the two partitions are highly in agreement. To investigate the sensitivity of the partition clustering methods to initial partitions, the clustering method is run with numerous different initial partitions. Then the average ARI (AARI) of all pair-wise resultant clusterings is calculated. This AARI indicates the sensitivity of the clustering method to initial partitions. The larger the value of AARI, the more insensitive (better) the clustering method is to initial partitions.

To evaluate the quality of the clusters, we propose a measure of internal consistency based on the singular value decomposition (SVD) of each cluster. To define internal consistency, suppose we are given a partition of our $n \times m$ dataset X into K disjoint clusters, where m is the number of time points and n is the number of genes. For the jth cluster ($j = 1, \cup, K$) we have a matrix X_j of microarray measurements, where the rows are genes and the columns are time points, so that X_j is a $n_j \times m$ matrix, where n_j is the number of genes in the jth cluster. Using the SVD, we decompose $X_j = U_j S_j V_j^T$, where U_j and V_j are orthogonal matrices and S_j is a diagonal matrix whose entries describe the importance of the columns of U_j and V_j. The matrix $X_j V_j = U_j S_j$ contains the projections of the rows (genes) of X_j onto the basis V_j. The entries of S_j (singular values) give the relative importance of the rows of V_j. If the first entry of S_j is much larger than the second entry then we know that most of the information in the rows of X_j is captured by a single dimension. We thus define the internal consistency of the jth cluster to be the ratio of the first and second singular values in S_j. The internal consistency provides a measure of how well a single dimension can describe all genes. We can evaluate the quality of a clustering with K clusters by the average internal consistency (AICo) of the K clusters. The high value of the AICo indicates the good quality of the clusters

Results

This section uses a synthetic and two real-life gene expression datasets to investigate the performance of the GWKMA in terns of AARI and AICo, while compared with the widely used k-means.

Synthetic dataset (SYN)

A synthetic dataset is generated by the sine function modeling cyclic behaviour of genes employed by Yeung, et al. [19]. Let x_{ij} be the simulated expression level of gene i and time point j in the dataset and be modeled by $x_{ij} = \lambda_j * \phi(i, j)(1 + \alpha_{ij})$, where $\phi(i, j) = \sin(2\pi j/8 - w_{k(i)} + \varepsilon_{ij})$. λ_j is the amplitude control at time j, which is chosen according to the standard normal distribution. $\phi(i, j)$ models the cyclic behaviour of genes. Each cycle is assumed to span 12 time points. Different clusters are represented by different phase shifts, and $w_{k(i)}$ represents a phase shift for gene i in cluster k, which is chosen according to the uniform distribution on interval $[0, 2\pi]$. The random variable ε_{ij} represents the noise of gene synchronization, which is chosen according to the normal distribution with the mean of zero and the standard deviation of 0.3. α_{ij} represents the error of gene i at time j, which is chosen according to the normal distribution with the mean of zero and the standard deviation of 0.4. Using the model above, a synthetic dataset is generated consisting of expression levels of 600 genes at 12 time points. These 600 genes belong to six clusters, each of which contains 100 genes.

Two real-life datasets

The first real-life dataset is a subset of gene expression profiles over 11 time points collected during the process of bacterial cell division [20], and contain 431 gene expression profiles with the standard deviation greater than 0.5 and no missing data points, denoted by BAC in this paper. The second dataset is a subset of gene expression profiles over 7 time points collected during the developmental program of sporulation in budding [21], and contains

529 gene expression profiles with the standard deviation greater than 1.0, and no missing data points, denoted by SPO. These two original datasets are publicly available from the Stanford microarray database [22] at http://genome-www5.stanford.edu/.

In the experiments conducted in this study, the number of generations is set to be *GEN* = 15, the population size = 21, and the mutation probability *Pm* = 0.10. The Matlab™ software package was used to conduct our experiments. Both AARI and AICo are computed over a variety of the numbers of clusters and for a number of the running results of both GWKMA and the traditional k-means.

The AICos with the cluster numbers from 2 to 10 are calculated from the results of 5 runs of both the GWKMA (solid lines) and the k-means (dash lines), and are depicted in the upper panel of Figure 3. The values of AICo for the GWKMA are greater than 1.8 while those for the k-means are less than 1.6. This indicates that the quality of clustering from the GWKMA is higher than that from the k-means. The AARI with the cluster numbers from 2 to 10 are calculated form the results of 5 runs of both the GWKMA (solid lines) and the k-means (dash lines), and are depicted in the lower panel of Figure 3. The values of AARI for the GWKMA are greater than those for the k-means over all the clusterings except for the one with k = 8. This result means that the GWKMA is more insensitive to initial partitions than the k-means.

Figures 4 and 5 depict the comparisons of the GWKMA and the k-means in terms of the AICo (the upper panels) and AARI (the lower panels) for the two real-life gene expression datasets. Before the clustering, two datasets are normalized by shifting the median of each gene expression profile to zero. From Figures 4 and 5, the same results are obtained from the real-life gene expression data as those from the synthetic dataset. That is, the GWKMA is better than the k-means in terns of AARI and AICo.

The superior quality of clustering from the GWKMA can be explained as follows. The k-means method assumes that 1) all attributes (data at time points) of objects (genes) are independent and 2) the standard deviations of all attributes over all objects are equal.

In practice, these two assumptions are not true. For example, we calculate the sample covariance matrix of dataset SPO shown in the matrix S in Figure 6. The elements on the main diagonal of matrix S are not equal and instead range from 0.23 to 4.44. This indicates that assumption 2) for the k-means is invalid. Actually, in many data analysis cases, gene expression data is normalized such that the standard deviation of each attribute over all objects is 1. In this case assumption 2) for the k-means is valid. However, assumption 1) for the k-means is still invalid. For exam-

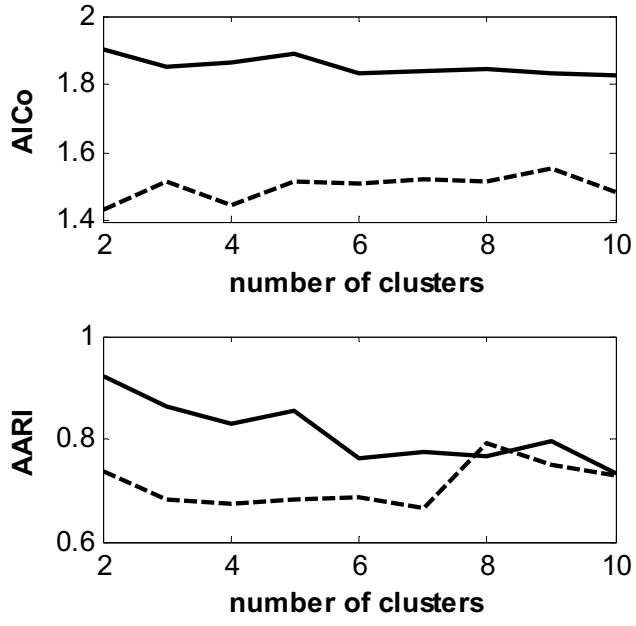

Figure 3
Comparison of the GWKMA (solid lines) to the k-means (dash lines) over a variety of the numbers of cluster on dataset SYN.

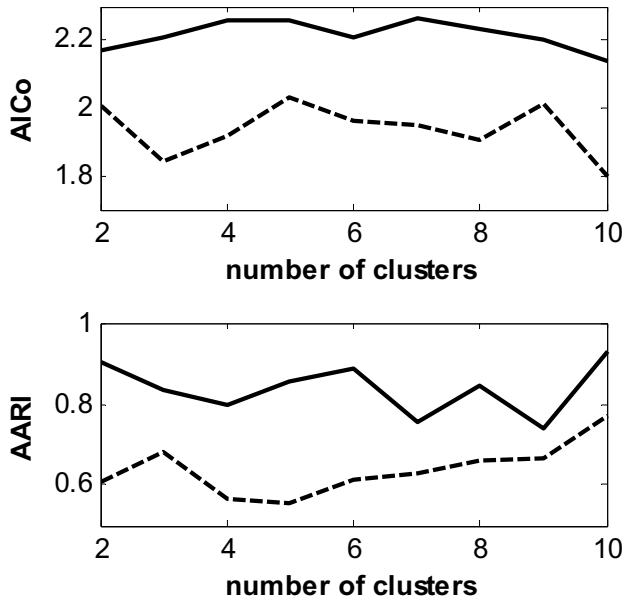

Figure 4
Comparison of the GWKMA (solid lines) to the k-means (dash lines) over a variety of the numbers of cluster on dataset BAC.

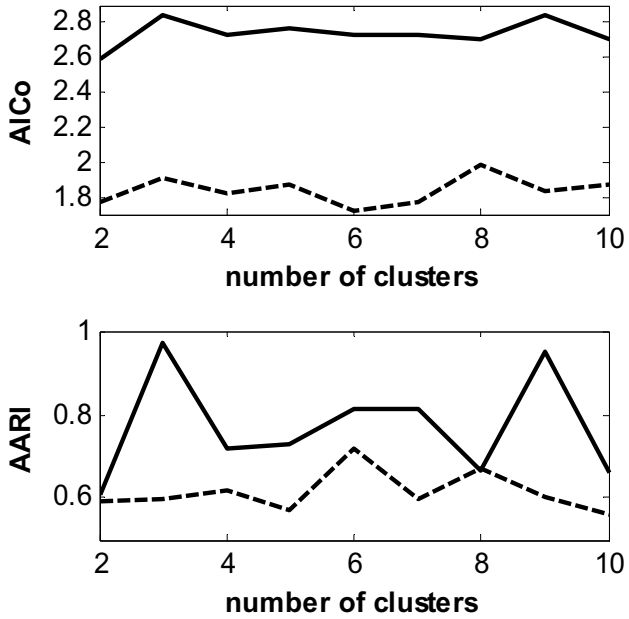

Figure 5
Comparison of the GWKMA (solid lines) to the k-means (dash lines) over a variety of the numbers of cluster on dataset SPO.

ple, in the matrix S in Figure 6 most off-diagonal elements are far from zero. This means most attributes in dataset SPO are correlated and thus not independent.

Conclusion

In this study, a genetic weighted k-means algorithm (GWKMA) is proposed which is a hybrid algorithm of the weighted k-means algorithm and a genetic algorithm. GWKMA was run on one synthetic and two real-life gene expression datasets. The results of the computational experiments show that the GWKMA performs better than the k-means in terms of the cluster quality (AARI) and the clustering sensitivity to initial partitions (AICo).

$$S = \begin{bmatrix} 4.44 & 2.73 & 2.23 & -0.27 & -1.91 & -1.05 & -1.24 \\ 2.73 & 3.92 & 2.64 & -0.17 & -1.59 & -1.29 & -1.61 \\ 2.23 & 2.64 & 2.35 & 0.00 & -1.28 & -1.01 & -1.36 \\ -0.27 & -0.17 & 0.00 & 0.23 & 0.13 & -0.01 & -0.11 \\ -1.91 & -1.59 & -1.28 & 0.13 & 1.63 & 1.12 & 1.28 \\ -1.05 & -1.29 & -1.01 & -0.01 & 1.12 & 1.20 & 1.26 \\ -1.24 & -1.61 & -1.36 & -0.11 & 1.28 & 1.26 & 1.95 \end{bmatrix}$$

Figure 6
The sample covariance matrix of dataset SPO.

In real-life datasets, the assumptions for the k-means are typically not satisfied. The weighted k-means does not needs the assumptions for the k-means. However, like the k-means, the weighted k-means is also sensitive to initial partitions. The proposed GWKMA possesses the merits of both genetic algorithm and the weighted k-mean algorithm, and thus overcomes the disadvantages of the k-means and the weighted k-means. In addition, the proposed algorithm is generic and could have applications to clustering large-scale biological data such as gene expression data and peptide mass spectral data.

Competing interests

The author declares that they have no competing interests.

Acknowledgements

This work is supported by Natural Sciences and Engineering Research Council of Canada (NSERC). The author benefited greatly from discussions with Drs W.J. Zhang and A.J. Kusalik at the early stage of this research.

This article has been published as part of *BMC Bioinformatics* Volume 9 Supplement 6, 2008: Symposium of Computations in Bioinformatics and Bioscience (SCBB07). The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/9?issue=S6.

References

1. Hartigan J: *Clustering Algorithms New York, Wiley Press*; 1975.
2. Reeves CR, Rowe JE: *Genetic Algorithms: Principles and Perspectives: A Guide To GA Theory Boston: Kluwer Academic Publishers*; 2003.
3. Rudolph G: **Convergence analysis of canonical genetic algorithms.** *IEEE Transactions on Neural Networks* 1994, **5:**86-101.
4. Krishna K, Murty MM: **Genetic K-means algorithm.** *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 1999, **29:**433-439.
5. Hall LO, Ozyurt IB, Bezdek JC: **Clustering with a genetically optimized approach.** *IEEE Transactions on Evolutionary Computation* 1999, **3:**103-112.
6. Maulik U, Bandyopadhyay S: **Genetic algorithm-based clustering technique.** *Pattern Recognition* 2000, **33:**1455-1456.
7. Franti P, Kivijärvi J, Kaukoranta T, Nevalainen O: **Genetic algorithms for large-scale clustering problems.** *The computer Journal* 1997, **40:**547-554.
8. Scheunders P: **A genetic c-means clustering algorithm applied to color image quantization.** *Pattern Recognition* 1997, **30:**859-866.
9. Al-Sultan KS, Khan MM: **Computational experience on four algorithms for the hard clustering problem.** *Pattern Recognition letters* 1996, **17:**295-308.
10. Wu FX, Zhang WJ, Kusalik AJ: **A genetic k-means clustering algorithm applied to gene expression data.** *Proceedings of the Sixteenth Canadian Conference on Artificial Intelligence, Halifax, Canada* 2003:520-526.
11. Tseng LY, Yang SB: **A genetic clustering algorithm for data with non-spherical-shape clusters.** *Pattern Recognition* 2000, **33:**1251-1259.
12. Duda RO, Hart PE, Stork DG: *Pattern Classification New York: Wiley Press*; 2001.
13. Spath H: *Cluster Analysis Algorithms for Data Reduction and Classification of Objects West Sussex: Ellis Horwood Limited*; 1975.
14. Estivill-Castro V: **Why so many clustering algorithms – A position paper.** *SIGKDD Explorations* 2002, **4:**65-73.
15. Theodoridis S, Koutroumbas K: *Pattern recognition San Diego: Academic Press*; 1999.
16. Dudoit S, Fridlyland J: **A prediction-based resampling method for estimating the number of clusters in a dataset.** *Genome Biol* 2002, **3(7):**RESEARCH0036.
17. Hubert L, Arabie P: **Comparing partitions.** *Journal of Classification* 1985, **2:**193-218.

18. Rand WM: **Objective criteria for the evaluation of clustering methods.** *Journal of the American statistical Association* 1971, **66:**864-850.
19. Yeung KY, Fraley C, Murua A, Raftery AE, Ruzzo WL: **Model-based clustering and data transformations for gene expression data.** *Bioinformatics* 2001, **17:**977-987.
20. Laub MT, McAdams HH, Feldblyum T, Fraser CM, Shapiro L: **Global analysis of the genetic network controlling a bacteria cell cycle.** *Science* 2000, **290:**2144-2148.
21. Chu S, DeRisi J, Eisen MB, Mulholland J, Botstein D, Brown PO, Herskowitz I: **The transcriptional program of sporulation in budding yeast.** *Science* 1998, **282:**699-705.
22. Sherlock G, Hernandez-Boussard T, Kasarskis A, Binkley G, Matese JC, Dwight SS, Kaloper M, Weng S, Jin H, Ball CA, Eisen MB, Spellman PT, Brown PO, Botstein D, Cherry JM: **The Stanford Microarray Database.** *Nucleic Acids Research* 2001, **29:**152-155.