

Gene Trajectory Clustering with a Hybrid Genetic Algorithm and Expectation Maximization Method

Zeke S. H. Chan and Nikola Kasabov, *Senior Member, IEEE*

Knowledge Engineering and Discovery Research Institute (KEDRI), Auckland University of Technology,
Auckland, New Zealand

Email: shchan@aut.ac.nz, nkasabov@aut.ac.nz

Abstract— Clustering time course gene expression data (gene trajectories) is an important step towards solving the complex problem of gene regulatory network (GRN) modeling and discovery as it significantly reduces the dimensionality of the gene space required for analysis. This paper introduces a novel method that hybridizes Genetic Algorithm (GA) and Expectation Maximization algorithms (EM) for clustering with the mixtures of Multiple Linear Regression models (MLRs). The proposed method is applied to cluster gene expression time course data into smaller number of classes based on their trajectory similarities. Its performance and application as a generic clustering method to other complex problems are discussed.

Index Terms— Gene trajectory clustering, Microarray data; Genetic Algorithm, Expectation Maximization, Gene regulatory networks.

I. INTRODUCTION

DNA microarray provides a snapshot of the expression level of thousands of genes in a particular cell type. The technology has enhanced our understanding of the complicated biological systems at the molecular level and has been particularly useful in medical applications. By comparing the expression of these genes in normal and diseased tissues for example, we can identify the set of genes that may be causing the disease. For example, in [1] microarray gene expression data taken from Diffuse large B-cell lymphoma tissues is used to identify a set of genes that can be used to predict the outcome of treatment of this disease for individual patients. The significance of this technology is shown also by its substantial market growth at 150% per annum over from 1997 to 2000 [2].

However, the problem of predicting the behavior of tumor or other type of cells requires more than analyzing static expression values of the genes from the RNA. As genes may vary in their expression level over time manifesting the dynamics of the cell processes, by measuring the expression of all genes over time, we can make a step towards finding some relationships between the genes and inferring Gene Regulatory Networks (GRN) that govern the underlying interaction between the genes.

In a single cell, the DNA, the RNA and the protein molecules interact in a continuous way during the process of the RNA transcription from DNA (genotype), and the subsequent RNA to protein (phenotype) translation [3]. A single gene interacts with many other genes in this process, inhibiting, directly or indirectly, the expression of some of them, and promoting others at the same time. This interaction can be represented as a GRN (see Fig. 1). A significant challenge for information scientists and biologists is to create computational models of GRN from both dynamic data (e.g. gene expression data of thousands of genes over time, and also from protein data) and from static data (e.g. DNA), under different external inputs (diseases, drugs, etc.). A large amount of both static and dynamic gene expression data is available from public domain databases. Collecting both static and time course gene expression data from up to 30,000 genes is now a common practice in many biological, medical and pharmaceutical laboratories.

Several generic information methods for modeling and for the discovery of GRN have been investigated so far [4]. Among them are: statistical methods, that include correlation techniques, linear regression, Bayesian networks, hidden Markov models; neural networks and evolving connectionist systems; evolutionary computation, and genetic algorithms in particular; directed graphs; Petri nets; ordinary and partial differential equations [5-7].

An important problem with inferring GRN is the large problem dimension (thousands of genes) relative to the small number of observations (several to tens of time points). For this reason, many clustering algorithms [8-13] are developed to reduce the problem dimension. After filtering out genes that are considered inactive or contain missing values, these algorithms cluster the remaining useful genes into a small, manageable number of representative groups (genes that cluster together are expected to be co-regulated and have similar functions), which are then used to infer the GRN that represents the relationship between them [14]. Clustering is thus an important tool for analyzing GRN. A simple example of such network is given in Fig. 1.

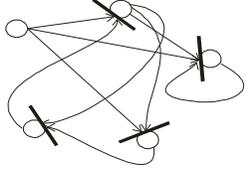


FIG. 1. A SIMPLIFIED GENE REGULATORY NETWORK GRN, WHERE THE NODES REPRESENT EITHER GENE CLUSTERS, SINGLE GENES OR PROTEINS. THE LINKS REPRESENT THE RELATIONSHIP BETWEEN THE CLUSTERS IN CONSECUTIVE TIME MOMENTS.

Because of the large problem dimension, the search space is highly multi-modal. Clustering methods that use local-learning algorithms such as K-means algorithm and Expectation Maximization algorithm (EM) are susceptible to finding locally optimal solutions and are unable to reach the global optimum. Moreover, local learning algorithms reach different local optima in each run (from different initial solutions), yielding inconsistent performance. In this paper we propose a framework that hybridizes Evolutionary Computation, in particular Genetic Algorithm (GA), with a local-learning algorithm to perform optimal clustering of time-course gene expression data to alleviate this problem. GA is a robust, global optimization algorithm capable of finding global optimum in a multi-modal search space, yet it is inefficient in local-hill climbing. The hybrid algorithm combines the strengths of GA and the local-learning algorithm (in our case, EM) by using the former to select subset of data as initial cluster centers and the later to perform fast local optimization to achieve the final centers from these initial centers. In this way, the optimality of the final centers returned by the local-search algorithm can be used as the objective function for GA, which searches for the globally optimal subset of data as initial cluster centers. In other words, rather than beginning the local optimization from data points that are randomly chosen, the hybrid algorithm begins the local optimization from data points that are globally optimal, therefore increasing the consistency of the final clustering solution.

As a demonstration of this framework, we hybridize GA and EM for optimal clustering with mixture of Multiple Linear Regression Models (MLRs). In the experiment, the hybrid algorithm is applied to the human fibroblasts time-course data that requires clustering of 512 useful genes and shows superior performance over using EM alone. Results show that although the hybrid algorithm requires higher computational cost, it performs consistently better in clustering accuracies. The hybridization framework is feasible to incorporate any local optimization methods to enhance the overall performance, for example, GA with the Scaled Conjugate Gradient method.

This paper is organized as follows. In Section II, literature on the problem of gene trajectory clustering, genetic algorithm and its application to clustering is reviewed. In Section III, the proposed hybrid algorithm is described in detail. Experiment is performed in Section IV, in which the

hybrid algorithm is applied to cluster the human fibroblasts data. Finally, conclusions and discussions are made in Section V.

II. BACKGROUND

A. Clustering of Gene Expression time-course data

The clustering of gene expression time-course data or gene trajectories is central to GRN analysis. It involves extracting a small number representative gene series from thousands of gene series. The clustering algorithms have been mainly non-model-based methods such as K-means clustering [15], Tree-based clustering [15], hierarchical clustering [16], graph-based CAST algorithm [11] and self-organizing map [10]. In recent years, model-based methods such as Autoregressive models [17], B-splines [8], cubic-splines [13] and mixture of Multiple Linear Regression Models (MLRs) [12, 18] are gaining popularity. The advantages suggested for the later are threefold: first, they can handle missing and irregularly spaced measurements; second, they can adjust model-smoothness and third, their out-of-sample predictive power can be evaluated. In this work, gene clustering is performed with the mixture of MLRs proposed in [12, 18]. All these methods use local optimization method for clustering, which are ineffective when the search space is highly multi-modal. For this reason, we employ Evolutionary Computation and in particular, Genetic Algorithm (GA) to overcome these local optima.

B. Evolutionary Computation and Genetic Algorithm

Evolutionary Computation (EC) [19] represents a general class of global optimization algorithms that imitate the evolutionary process explained by Darwinian Natural Selection. Its branches include Evolutionary Strategies, Genetic Algorithm and Genetic Programming.. EC searches with a set of multiple points and requires only the evaluation of the function value of the points. It is therefore robust for complex, black problems that lack derivative information and contain multi-modality – the problems that defeat classical, derivative-based search methods. However, the evaluation of multiple search points causes EC to be computationally intensive. Therefore, although some earliest forms of EC have existed since the mid 1960s, EC received research attention only after the availability of high-speed computers in the last two decade.

GA was proposed by Holland in 1975 [20]. It performs optimization in the binary domain to imitate the evolutionary process that occurs at the genetic level. Each solution is treated as a *chromosome* (strand of DNA code that carries genetic information) and each binary bit on the chromosome a *gene*. The *genetic or evolutionary operators* – *crossover*, *mutation* and *selection* - are direct imitations of their genetic counterparts. Although GA's operation seems heuristic, convergence to a high quality optimum (in the global sense) is theoretically supported by the Schema theorem developed

by Holland. For this reason, GA is one of the most studied forms of EC today and is widely applied to different Engineering problems.

The basic operations of GA iterates between crossover, mutation and selection. Each iteration is called a *generation*. Each solution is coded as a binary string. First, a set of initial solutions is created and become the first *parent* (for “reproducing” new solutions called *offspring*) population. The crossover operator imitates the mixing of genetic information during sexual reproduction. It swaps genes and/or segments at random positions between two parental strings to form two new strings, of which one becomes the offspring and the other abandoned to reduce statistical dependence between offspring. The crossover probability p_c is usually between 0.5 and 0.9 [21]. The mutation operator imitates the occasional changes of genetic information on a chromosome due to external influences. It inverts each gene on the new string with a rare (between 0.001 to 0.1) but fixed probability p_m [21]. The set of new solutions created by the crossover and mutation operator from the parent population are called the *offspring* population. The goodness of each solution is called its *fitness* and is evaluated by the optimization objective. The standard selection operator is the Roulette Wheel selection that assigns to each offspring solution a survival probability in proportion to its fitness. The survivors become the next generation parents. The process iterates until some termination criteria are met.

C. Application of Genetic Algorithm to Clustering/Feature selection

In the hybrid algorithm proposed, GA does not directly perform clustering but rather, complements it by selecting subset of data and use them as the initial cluster centers. This task is commonly known as feature (subset) selection, to which many successful GA applications have been reported [22-27]. In most of these approaches, each individual solution is encoded as an N -bit string where N is the total number of features. A “1” in the n th bit-position indicates that the n th feature is selected, whereas a “0” indicates otherwise. Conventional evolutionary operators of crossover and mutation are employed. The subset size may be restricted or self-determined, with the later requiring more function evaluations. Assuming that the optimal genes contribute to high fitness solution, this type of binary coding is consistent with the Schema theorem [28] that assures convergence towards the optimum. Our GA adapts this conventional solution representation and modeling.

III. THE HYBRID CLUSTERING ALGORITHM

The objective of the proposed hybrid clustering framework is to improve the exploration capability of local clustering algorithms in multi-modal search space with Genetic Algorithm (GA). As a demonstration, we apply the hybrid framework to gene clustering with a mixture of Multiple Linear Regression models (MLRs), which uses Expectation

Maximization algorithm (EM) as the local learning algorithm. The hybrid algorithm consists of two levels as depicted in Fig. 2. At the higher level, GA searches for the optimal subset of genes that act as initial cluster centers. At the lower level, the local learning method, in this case the Expectation Maximization algorithm (EM), performs local clustering from these initial centers. The objective is to combine the strength of GA and EM to produce a global yet efficient clustering algorithm.

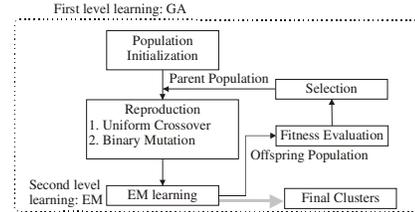


FIG. 2. THE HYBRID CLUSTERING ALGORITHM.

The objective of the GA at the top level is to select the optimal subset of genes as the initial cluster centers. Let n denote the total number of genes and G denote the number of clusters. Each solution is encoded as an n -bit string with the i th bit position corresponding to the i th gene for $i=[1, 2, \dots, n]$. A “1” in the i th bit position indicates that the i th gene is selected and a “0” indicates otherwise. Thus for selecting G optimal genes as the initial centers, all feasible solutions must have G number of “1”s and $(N-G)$ number of “0”s. The evolutionary operators include uniform crossover and mutation. Uniform crossover selects two random parents and at the crossover probability p_c , swaps the bits between the parents at the same bit position with 50% probability to create two new solutions. One of them is then taken as the offspring. It has the important function of passing on high fitness schema from the parents to the offspring. Uniform crossover has the advantage over traditional k -point crossover that it eliminates biases in the crossover search [29]. Mutation simply inverts each bit of the offspring at the mutation probability p_m to diversify to the search. It has the important function of preventing premature convergence. Finally, a repair operator is applied to the final offspring to ensure that the offspring is a feasible solution, i.e. it has G number of “1”s. If the solution has more than G number of “1”, the “1”s are randomly inverted, whereas if it has less than G number of “1”s, the “0”s are randomly inverted. The repair continues until the solution has exactly G number of “1”s. Fig. 3 summarizes the operation of these three evolutionary operators.

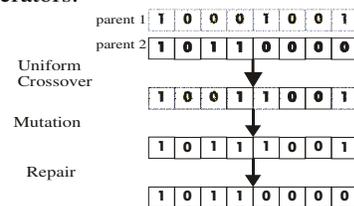


FIG. 3. THE OPERATION OF THE UNIFORM CROSSOVER, MUTATION AND REPAIR OPERATOR (FOR $G=3$ CLUSTERS).

The fitness function of each solution is measured as the maximum log likelihood of the EM-optimized model (described later). For selection method we use the elitist scheme of $(\mu+\lambda)$ that selects the best-fit μ solutions out of the joint pool of μ parents and λ offspring to be the next generation parents. The elitist scheme ensures that the best solutions always stay in the population and thus yield faster convergence. Since the search relies only on the population information and not on gradient information, it is less susceptible to getting trapped in local optima.

The clustering model is a mixture of G MLRs (one for each cluster), each of which represents a single gene trajectory cluster given by

$$\mathbf{Y}_i = S(\mu_k + \gamma_i) + \varepsilon_i \quad \gamma_i \sim N(0, \Gamma) \quad \varepsilon_i \sim N(0, R) \quad (1)$$

where $\mathbf{Y}_i = [y_{i,1}, y_{i,2}, \dots, y_{i,l}]$ is the i th gene trajectory of length l , S is the $l \times (p+1)$ regression matrix (or basis matrix) where p is the regression order, μ_k is the $(p+1)$ -vector of regression coefficients, γ_i and ε_i are uncorrelated Gaussian noises for the regression coefficients and the trajectory respectively. The regression matrix S can adapt Vandermonde, spline basis function or even time-series functions. Here we use the Vandermonde function given by

$$S = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^p \\ 1 & t_2 & t_2^2 & & t_2^p \\ \vdots & & & & \vdots \\ 1 & t_l & t_l^2 & \cdots & t_l^p \end{bmatrix} \quad (2)$$

where $\{t_1, t_2, \dots, t_l\}$ are the times at which the points in the trajectory are observed. Let $\mathbf{z}_i = (z_{i1}, \dots, z_{iG})$ be the cluster membership vector for the i th trajectory where $z_{ik}=1$ if the i th trajectory belongs to the k th cluster and 0 otherwise. The standard method for mixture model learning is to treat z_i as missing variables and apply Expectation Maximization (EM). See [18, 30] for implementation details. Here we only provide all the necessary iterative equations. EM performs parameter estimation through maximizing the complete data log likelihood $l(\cdot)$ given by

$$\begin{aligned} & l(\pi_k, \Gamma, \sigma^2, \mu_k) \\ &= \sum_{i=1}^n \sum_{k=1}^G z_{ik} \log(\pi_k) \\ & - \frac{1}{2} \sum_{i=1}^n [\log |\Gamma| + \gamma_i^T \Gamma^{-1} \gamma_i] \\ & - \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^G z_{ik} \left[n_i \log \sigma^2 + \frac{1}{\sigma^2} \|\mathbf{Y}_i - S_i(\mu_k + \gamma_i)\|^2 \right] \end{aligned} \quad (3)$$

The maximization is achieved by iterating (4)-(7),

$$\pi_k = \frac{1}{n} \sum_{i=1}^n \pi_{k/i} \quad (4)$$

$$\mu_k = \left(\sum_{i=1}^n \pi_{k/i} S_i^T S_i \right)^{-1} \left[\sum_{i=1}^n \pi_{k/i} S_i^T (\mathbf{Y}_i - S_i \hat{\gamma}_{ik}) \right] \quad (5)$$

$$\sigma^2 = \frac{1}{\sum_{i=1}^n n_i} \times \sum_{i=1}^n \sum_{k=1}^G \pi_{k/i} E[(\bar{\mathbf{Y}}_i - S_i \hat{\gamma}_{ik})^T (\bar{\mathbf{Y}}_i - S_i \mu_k - S_i \hat{\gamma}_{ik}) + S_i \hat{\Gamma}_{ik} S_i^T] \quad (6)$$

$$\Gamma = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^G \pi_{k/i} \hat{\Gamma}_{ik} \quad (7)$$

where

$$\hat{\Gamma}_{ik} = E[\gamma_i \gamma_i^T | \mathbf{Y}_i, z_{ik} = 1] = \left(\Gamma^{-1} + \frac{S_i^T S_i}{\sigma^2} \right)^{-1} \quad (8)$$

$$\hat{\gamma}_{ik} = E[\gamma_i | \mathbf{Y}_i, z_{ik} = 1] = (\sigma^2 \Gamma^{-1} + S_i^T S_i)^{-1} S_i^T (\mathbf{Y}_i - S_i \mu_k) \quad (9)$$

$$\pi_{k/i} = P(z_{ik} = 1 | \mathbf{Y}_i) = \frac{f(y | z_{ik} = 1) \pi_k}{\sum_{j=1}^G f(y | z_{ik} = 1) \pi_j} \quad (10)$$

$$f(y | z_{ik}) \equiv N(S \mu_k, \sigma^2 I + S \Gamma S^T) \quad (11)$$

π_k is the prior probability for the k th cluster. The initial regression coefficients μ'_k of the k th cluster are computed from G randomly sampled curves \mathbf{Y}_i

$$\mu'_k = (S^T S)^{-1} S^T \mathbf{Y}_i \quad (12)$$

The initial variance σ_i^2 has lesser effect to the final solution and is simply set to an arbitrarily small value (0.1 in our case).

The choice for initial regression coefficients makes significant difference to the quality of the final solution. It is because the search space is multi-modal and EM is a local optimizer, the initial values determine which local optimum the learning leads to. In standard EM, the initial centers are randomly chosen subset from the data or computed with K-means algorithm. The later may not be suitable in this application as we find that its objective leads to a different solution from that of the EM.

To improve the initial estimates, the hybrid algorithm uses GA to select the optimal subset of data as the initial cluster centers and calculate their regression coefficients using (12). In each generation, GA produces a set of offspring solutions, each of which represents a subset of genes used for computing the initial clusters. Each offspring is then EM-optimized. Using maximum-likelihood given in (3) as the fitness function, the best offspring are selected as next generation parents to reproduce new offspring. The evolutionary operators assure the accumulation of the optimal gene and hence the increase of the overall fitness.

IV. APPLICATION TO THE RESPONSE OF HUMAN FIBROBLASTS TO SERUM DATA SET

The experiment compares the performance between the proposed hybrid algorithm and the random-initialization EM in the application of gene trajectory clustering of the human fibroblasts to serum data set. The data set was reported in [31]. It recorded the time program at 12 time points of gene expression during the physiological response of fibroblasts to

serum using cDNA microarrays. The data consists of 8618 genes sampled at $\{0, 0.25, 0.5, 1, 2, 4, 6, 8, 12, 16, 20, 24\}$ hours, of which 517 genes are identified to respond significantly to the serum and are clustered. The log-normalized data of 517 genes is shown in Fig. 4.

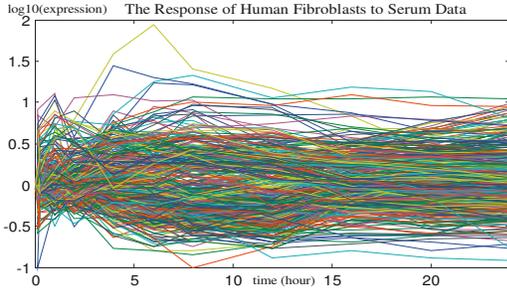


FIG. 4. THE HUMAN FIBROBLASTS TO SERUM DATA (517 GENES).

The order for the MLRs is chosen to be $P=6$ since the coefficients for higher orders are negligibly small ($<10e-4$). The clustering results obtained by the hybrid method and the random-initialization EM are visually similar and also to those obtained in [8, 31]. We must therefore use the model likelihood to measure the goodness of fit. Some example results from clustering the data into $G=7$ classes are shown in Fig. 5 and Fig. 6.

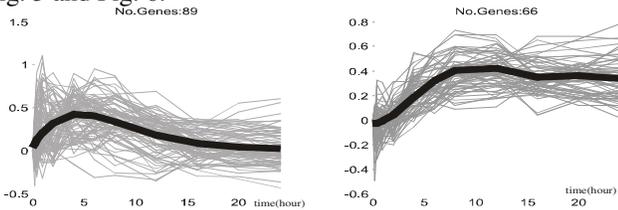


FIG. 5. TWO OF THE SEVEN CLUSTERS AND THEIR ASSOCIATED GENES.

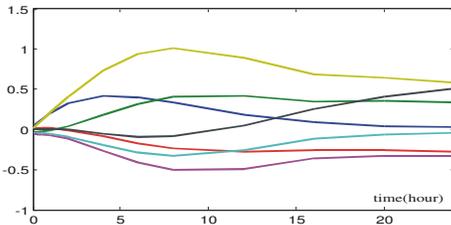


FIG. 6. ALL SEVEN CLUSTERS OF GENE TRAJECTORIES.

The settings for GA and EM are as follows. The binary coded solution is $N=517$ bits long with “1”s at the selected gene and “0”s at the rest. The population sizes are $\mu=10$ for the parents and $\lambda=20$ for the offspring, which are relatively small in order to limit computation time. The selection scheme is the elitist scheme $(\mu+\lambda)$ that always keeps the best individuals in the population to accelerate convergence. Uniform crossover is applied at a high crossover rate $p_c=0.9$, which is a common choice to facilitate transmission of optimal schema in the population. While binary mutation is often applied at the mutation rate $p_m=1/N$ that yields an average of one inversion per string, we set p_m to relative high

rate of $p_m=(G/N)$ to yield an average of G (G is the number of clusters) inversions per string to increase the diversity of the search. Each GA runs for 20 generations (obviously, the larger the number the generation, the higher the solution quality becomes. Here we use a relatively small number of generations for time saving and demonstration purpose only). For EM, the stopping criterion is when the maximum log likelihood increases by less than 1.

Performance is measured in maximum log likelihood (ML) obtained by the models over clustering the data into $\{5, 7, 9, 11, 13, 15\}$ classes. Conversion from ML to other performance measures like Akaike or Bayesian information criterion is straightforward. All results are averaged over 10 runs. They are tabulated in Table 1.

TABLE 1: MAXIMUM LOG LIKELIHOOD OF THE MLRS WITH DIFFERENT NUMBER OF CLUSTERS IDENTIFIED BY GA AND RANDOM INITIALIZATION.

No. Clusters	GA Initialization		Random Initialization	
	mean	std	mean	std
5	2200.50	0.08	2114.60	50.52
7	2590.20	0.09	2511.40	68.57
9	2898.50	0.03	2809.10	71.05
11	3086.20	0.74	3019.10	31.11
13	3246.70	2.46	3136.70	54.21
15	3350.40	1.69	3228.70	57.53

Results in Table 1 show that the hybrid algorithm scores higher ML with smaller standard deviation (higher consistency) than the random initialization EM over at all tested number of clusters. The improvement in ML increases as the number of clusters increases. This is explained by the increased number of gene combinations at higher number of clusters ($2e^{107}$ combinations at $G=5$ versus $3e^{1028}$ combinations at $G=15$), which introduces more multimodality and in turn amplifies the advantage of GA’s global optimization capability. A typical GA run is shown in Fig. 7. The fitness increases uni-directionally, which is the characteristic of the $(\mu+\lambda)$ selection scheme.

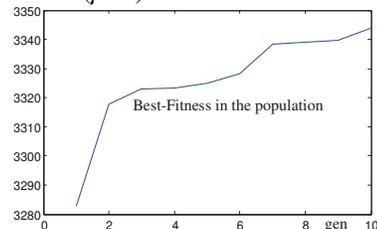


FIG. 7. EVOLUTION OF THE BEST-FITNESS FOUND IN THE POPULATION.

Note that GA’s superior performance incurs higher computational costs, requiring a total of $(\lambda \times \text{max.generations})$ EM evaluations. However, with the human fibroblast data that has 517 genes and 12 time points, each EM evaluation requires less than 10 seconds (running in Matlab on a

Pentium IV 2.4GHz) and hence GA poses little problem with computation time.

V. CONCLUSIONS

The paper introduces a novel clustering method based on the hybridization of GA and EM algorithms to mixture of MLRs and applies this method for clustering of gene expression time course data (trajectories) as a step towards the creation of gene regulatory networks to model the behavior of biological cells. The experimental results on gene expression time course data available on the public domain show the advantages of the hybrid GA-EM approach when compared with the standard approach of using random initialization EM algorithm only. The method can be applied on other complex problems of variable trajectory clustering, such as signal processing and financial data.

The proposed hybrid GA framework can be easily generalized to other clustering methods, for example, the K-means and the multivariate Normal mixture model. By applying GA to optimal selection of the subset of data as initial clusters, the solutions obtained by these algorithms can be more globally optimal.

ACKNOWLEDGMENTS

This research is supported by the KEDRI postdoctoral fellow research fund.

REFERENCES

- [1] M. A. Shipp, K. N. Ross, P. Tamayo, and A. P. Weng, "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Medicine*, vol. 8, pp. 68-74, 2002.
- [2] I. Strategic Directions International, "Microarray Technology: The Next Step in Genomic and Proteomic Analysis," Strategic Directions International, Inc., 2001.
- [3] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the boolean network model," presented at Pacific Symposium on Biocomputing, 1999.
- [4] H. de Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *Journal of Computational Biology*, vol. 9, pp. 67-102, 2002.
- [5] K. W. Kohn and D. S. Dimitrov, "Mathematical Models of Cell Cycles," *Computer Modeling and Simulation of Complex Biological Systems*, 1999.
- [6] J. R. Koza, W. Myrdlowec, G. Lanza, J. Yu, and M. A. Keane, "Reverse Engineering of Metabolic Pathways from Observed Data using Genetic Programming," presented at Pacific Symposium on Biocomputing, 2001.
- [7] N. Kasabov, *Evolving connectionist systems - methods and applications in bioinformatics, brain study and intelligent machines*. London-New York: Springer Verlag, 2002.
- [8] Y. Luan and H. Li, "Clustering of time-course gene expression data using a mixed-effects model with B-splines," *Bioinformatics*, vol. 19, pp. 474-482, 2003.
- [9] M. B. Eisen, P. T. Spellman, and P. O. Brown, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences USA*, vol. 14, pp. 14863-14868, 1995.
- [10] P. Tamayo, D. Slonim, and J. Mesirov, "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences USA*, vol. 96, pp. 2907-2912, 1999.
- [11] A. Ben-Dor and Z. Yakhini, "Clustering gene expression patterns," presented at Proceedings of the Third Annual International Conference on Computational Molecular Biology, Lyon, France, 1999.
- [12] I. Cadez, S. Gaffney, and P. Smyth, "A General Probabilistic Framework for Clustering Individuals," Department of Information and Computer Science, University of California, Irvine 00-09, 2000.
- [13] Z. Bar-Joseph, G. Gerber, D. Gifford, T. Jaakkola, and I. Simon., "A new approach to analyzing gene expression time series data," presented at Proceedings of The Sixth Annual International Conference on Research in Computational Molecular Biology, 2002.
- [14] N. Kasabov and D. Dimitrov, "A method for gene regulatory network modelling with the use of evolving connectionist systems," presented at ICONIP2002 - International Conference on Neuro-Information Processing, Singapore, 2002.
- [15] M. B. Eisen, "Softwares at <http://rana.lbl.gov/index.htm>," 2003.
- [16] P. T. Spellman, G. Sherlock, M. Q. Zhang, and V. R. Iyer, "Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization," *Molecular Biology of the Cell*, vol. 9, pp. 3273-3297, 1998.
- [17] M. F. Ramoni, S. P., and K. I. S., "Cluster analysis of gene expression dynamics," *Proceedings of the National Academy of Sciences USA*, vol. 99, pp. 9121-9126, 1999.
- [18] S. Gaffney and P. Smyth, "Curve Clustering with Random Effects Regression Mixtures," presented at Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, Florida, 2003.
- [19] T. Baeck, G. Rudolph, and H. Schwefel, "Evolutionary programming and evolution strategies: Similarities and differences," presented at the Second Annual Conference on Evolutionary Programming, San Diego CA, 1993.
- [20] J. H. Holland, *Adaptation in natural and artificial systems*: The University of Michigan Press, Ann Arbor, MI, 1975.
- [21] T. Baeck, D. B. Fogel, and Z. Michalewicz, *Evolutionary Computation I. Basic algorithm and operators*, vol. 1. Bristol: Institute of Physics Publishing, 2000.
- [22] J. Yang and V. Honavar, "Feature Subset Selection Using a Genetic Algorithm," *IEEE Intelligent Systems*, vol. 13, pp. 44-49, 1998.
- [23] E. Yom-Tov and G. F. Inbar, "Feature Selection for the Classification of Movements From Single Movement-Related Potentials," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, pp. 170-177, Sept 2002.
- [24] F. Z. Brill, D. E. Brown, and W. N. Martin, "Fast Genetic Selection of Features for Neural Network Classifiers," *IEEE Transactions on Neural Networks*, vol. 3, pp. 324-328, Mar 1992.
- [25] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, pp. 164-171, Jul 2000.
- [26] L. B. Jack and A. K. Nandi, "Genetic algorithms for feature selection in machine condition monitoring with vibration signals," *IEE Proceedings of Vision, Image and Signal Processing*, vol. 147, pp. 205-212, Jun 2000.
- [27] I. Inza, P. Larranaga, R. Etxebarria, and B. Sierra, "Feature Subset Selection by Bayesian network-based optimization," *Artificial Intelligence*, vol. 123, pp. 157-184, 2000.
- [28] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [29] L. J. Eshelman, R. A. Caruana, and J. D. Schaffer, "Biases in the Crossover Landscape," presented at the Third International Conference on Genetic Algorithms, 1989.
- [30] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood for incomplete data via the EM algorithm," *Journal of Statistics Society*, vol. B, pp. 1-38, 1977.
- [31] V. R. Iyer, M. B. Eisen, D. T. Ross, and G. Schuler, "The transcriptional program in the response of human fibroblasts to serum," *Science*, vol. 283, pp. 83-87, 1999.