BIOINFORMATICS

Vol. 17 no. 5 2001
Pages 405–414

Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters

Alexander V. Lukashin and Rainer Fuchs*

Biogen Inc., 14 Cambridge Center, Cambridge, MA 02142, USA

Received on August 17, 2000; revised on October 9, 2000; accepted on October 10, 2000

ABSTRACT

Motivation: Cluster analysis of genome-wide expression data from DNA microarray hybridization studies has proved to be a useful tool for identifying biologically relevant groupings of genes and samples. In the present paper, we focus on several important issues related to clustering algorithms that have not yet been fully studied.

Results: We describe a simple and robust algorithm for the clustering of temporal gene expression profiles that is based on the simulated annealing procedure. In general, this algorithm guarantees to eventually find the globally optimal distribution of genes over clusters. We introduce an iterative scheme that serves to evaluate quantitatively the optimal number of clusters for each specific data set. The scheme is based on standard approaches used in regular statistical tests. The basic idea is to organize the search of the optimal number of clusters simultaneously with the optimization of the distribution of genes over clusters. The efficiency of the proposed algorithm has been evaluated by means of a reverse engineering experiment, that is, a situation in which the correct distribution of genes over clusters is known *a priori*. The employment of this statistically rigorous test has shown that our algorithm places greater than 90% genes into correct clusters. Finally, the algorithm has been tested on real gene expression data (expression changes during yeast cell cycle) for which the fundamental patterns of gene expression and the assignment of genes to clusters are well understood from numerous previous studies.

Availability: The source code of the program implementing the algorithm is available upon request from the authors.

Contact: alex_lukashin@biogen.com

INTRODUCTION

Rapid advances in microarray technologies over the last several years have made it possible to simultaneously monitor the expression profiles of thousands of genes under various experimental conditions (for reviews see Lockhart and Winzeler, 2000; Young, 2000). The production of increasingly reliable and accessible expression data has stimulated the development of computational tools to interpret such data and to organize them efficiently in system-level conceptual schemes. Current methods for the analysis of gene expression data typically rely on the use of clustering algorithms applied to gene expression profiles (Hartigan, 1975; Jain and Dubes, 1988). The fundamental biological premise underlying these approaches is that genes that display similar expression patterns are co-regulated and may share a common function or contribute to a common pathway. Although this assumption may be overly simplistic and will not always be true, cluster analysis has been demonstrated to be of significant value for the exploration of gene expression data (Wen *et al.*, 1998; Eisen *et al.*, 1998; Spellman *et al.*, 1998; Tamayo *et al.*, 1999; Alon *et al.*, 1999; Perou *et al.*, 1999; Tavazoie *et al.*, 1999; Zweiger, 1999; White *et al.*, 1999; Brown *et al.*, 2000; Roberts *et al.*, 2000; Ross *et al.*, 2000).

In spite of the fact that a variety of different clustering algorithms is now available, a number of important questions remain to be addressed. For example, techniques based on hierarchical clustering (e.g. Eisen *et al.*, 1998; Alon *et al.*, 1999) have problems related to robustness, uniqueness, and optimality of linear ordering which complicates the interpretation of the resulting hierarchical relationships. On the other hand, algorithms that are based on the optimization of a given cost function (e.g. Tamayo *et al.*, 1999; Tavazoie *et al.*, 1999) cannot guarantee that the resulting solution corresponds to the global optimum rather than to a local one. A problem common to both of these popular clustering techniques is how to determine the optimal number of clusters. Hierarchical clustering approaches leave this problem to the observer who has to interpret tree topologies and identify branch points that segregate clusters of biological relevance. In optimization-based approaches, the number of clusters is introduced as a fixed, external parameter of the algorithm that is being used.

*To whom correspondence should be addressed.

In the present paper, we address some of the problems raised above. First, we propose a robust clustering algorithm based on the simulated annealing procedure (Kirkpatrick *et al.*, 1983; Aart and van Laarhoven, 1987). The advantage of this technique is its ability to cope with the local minima problem; it is guaranteed to eventually find the global optimum. Second, we describe a methodology that serves to identify quantitatively the optimal number of clusters for any specific data set. Finally, we present both statistical and biological validation of our approach.

METHODS

Clustering by simulated annealing

Let N be the number of time-course gene expression profiles with M time points each. Since we focus on the shapes of expression patterns rather than on absolute levels of expression, each profile is normalized such that the expression level varies between 0 and 1. Each ith profile is represented by an M-dimensional vector, $\{e_1^i, e_2^i, \ldots, e_M^i\}$, with component e_m^i corresponding to the normalized expression level of gene i at time point m $(0 \leqslant e_m^i \leqslant 1)$. The similarity metric we use is the Euclidean distance, d_{ij}, between vectors i and j:

$$d_{ij} = \left[\sum_{m=1}^{M} (e_m^i - e_m^j)^2 \right]^{1/2} \tag{1}$$

For a given number of clusters, K, we optimize the distribution of profiles over the clusters by minimizing the sum of distances d_{ij} *within clusters* using Equation (2)

$$E(K) = \frac{1}{K} \sum_{k=1}^{K} \left[\sum_{i \in Ck} \sum_{j \in Ck} d_{ij} \right] \tag{2}$$

where $i \in Ck$ stands for vector i that belongs to the cluster number k. To minimize the E-value in Equation (2) we apply the simulated annealing algorithm (Kirkpatrick *et al.*, 1983; Aart and van Laarhoven, 1987). Initially, the distribution of vectors over clusters is randomly assigned. At each iterative step, a randomly selected vector is taken out from its cluster and reassigned to another randomly chosen cluster. A new value E^{new} is calculated and compared with the previous value E^{old}. If E^{old} is larger than E^{new}, the new assignment of the vector is unconditionally accepted and used as the starting point for the next iteration. Otherwise, the new assignment is accepted with probability $\exp[-(E^{\text{new}} - E^{\text{old}})/T]$, where the parameter T can be interpreted as the 'temperature', if the E-value is treated as the 'energy' of the system. This algorithm guarantees that after a sufficient number of iterative steps the system obeys the Boltzmann distribution at a given temperature. Consequently, if the temperature T approaches zero slowly enough the system will reach the global minimum of the E function avoiding local minima. Routinely, we use an exponential cooling schedule $T_{n+1} = cT_n$, where n is the step number and the value $1 - c$ is positive and close to zero. We verified that the E-value and the corresponding optimal distribution of genes over clusters resulting from the simulated annealing procedure applied to minimization of the E function Equation (2) did not depend on the random number seed if $1 - c \leqslant 10^{-6}$ (data not shown).

Conceptual framework for determining the optimal number of clusters

Obviously, the optimal number of clusters depends primarily on the variation between profiles within a given data set. A measure of this variation is the distribution function $p(d)$ of Euclidean distances between the vectors that represent the profiles in the data set. Function $p(d)$ is normalized so that the integral over all d is equal to one. An example of such a function is shown in Figure 2A (see below for details). The wider the function $p(d)$ the more clusters are needed to obtain tight clusters with distinctive patterns of expression. However, beyond some number of clusters K the further increase of K is often meaningless: eventually, K may become so large and clusters so tight that the standard deviation within clusters is less than the experimental error.

To treat the problem of identifying the optimal number of clusters quantitatively we introduce a *cutoff distance* D and postulate that the assumption that vectors i and j belong to the same cluster is incorrect if $d_{ij} \geqslant D$. A closer look at the relationship between the distribution function $p(d)$ and the distance D may help to clarify the meaning of D. Suppose that we have only one cluster for all genes. Then for a given D the fraction of incorrect vector pairs $f(D, K = 1)$ is defined by the integral

$$f(D, K = 1) = \int_D^{\infty} p(x)\, \mathrm{d}x \tag{3}$$

that is, by the probability to find a pair of vectors with the distance between them equal to or greater than D.

Integral (3) yields the *upper boundary* (with respect to the number of clusters K) for the fraction of incorrect vector pairs $f(D, K)$. In general, for a given number of clusters K and for an optimal assignment of vectors to clusters, the probability of finding an incorrect vector pair can be estimated as the weighted average fraction of incorrect vector pairs:

$$f(D, K) = \frac{1}{K} \sum_{k=1}^{K} \frac{\text{number of incorrect vector pairs in cluster \#}k}{\text{total number of vector pairs in cluster \#}k} \tag{4}$$

This probability monotonously decreases with the increase of the number of clusters K.

The *lower boundary* for the function $f(D, K)$ can be set by a straightforward analogy with the pre-assignment of P-values in regular statistical tests. We define the lower boundary as the maximal allowed probability to find an incorrect vector pair in a cluster. Therefore, in our conceptual framework the optimal number of clusters is defined as the solution of Equation (5):

$$f(D, K) = P. \quad (5)$$

Given parameters D and P, we solve this equation by sequentially increasing the number of clusters K and repeating the minimization of function (2) for each value of K, until the fraction of incorrect vector pairs $f(D, K)$ reaches the P-value. Note, that the shorter the cutoff distance D the more clusters are needed to reach the same value for the fraction of incorrect vector pairs.

Of course, the result of this process, the number of clusters K, will depend on the particular values we choose for parameters D and P. The problem of identifying the optimal number of clusters has therefore now become a problem of optimizing those two parameters.

The relationship between parameters D and P—reverse engineering experiment

Parameter P represents the fraction of allowed false positives, and we assign it arbitrarily within a reasonable interval. Routinely, we use $P = 0.055$. Once parameter P is fixed, the optimal value of the cutoff distance D can be derived as follows. Suppose we know the optimal number of clusters K^{opt} for a given data set *a priori*. Then the value of D can be determined by solving the equation $f(D, K^{\text{opt}}) = P$. To this end we utilized the following reverse engineering procedure. First, we randomly generated 24 seed patterns of expression with 10 time points each. Second, each pattern was transformed into a cluster by splitting the pattern into individual profiles (from 10 to 200 profiles per cluster with the total number of profiles equal to 2000; for the specific number of profiles in clusters see Table 1). This splitting step is a random procedure but controlled in such a way that the weighted average standard deviation from the seed patterns within clusters does not exceed a pre-assigned value SD. In this experiment, we used $SD = 0.15$, a value that reflects the typical variation observed in published expression profiling experiments. Figure 1 shows the clusters produced by this approach. The Euclidean distances between profiles are depicted in Figure 2A. Because of the way this particular data set was constructed we know that the optimal number of clusters is $K^{\text{opt}} = 24$. Parameter D can thus be determined by finding a solution for the equation $f(D, K^{\text{opt}} = 24) = 0.055$. To do this, we applied the simulated annealing algorithm to our data set and calculated the optimal distribution of profiles for different values of K (the running time for one set of parameters K and D is approximately 1 min on a standard SGI workstation). For each resulting distribution we then calculated the fraction of incorrect vector pairs $f(D, K)$ over various distances D using Equation (4). The results are shown in Figure 2B. This graph demonstrates how the fraction of incorrect vector pairs decreases with the number of clusters for different cutoff distances D. It is apparent that for the chosen P-value of 0.055 the known optimal number of clusters $K = 24$ is found at an optimal cutoff distance of $D = 1.10$. We verified that the optimal cutoff distance D did not depend on a particular choice for the optimal number of clusters (for a given number of time points): we repeated the above procedure for $K^{\text{opt}} = 10$ and $K^{\text{opt}} = 34$ and obtained values of $D = 1.10 \pm 0.02$ (data not shown).

Obviously, parameter D depends on the number of time points. Therefore, we address the question: having determined D for our reverse engineering data set, how can D be established for other data sets with different numbers of data points? To this end, we consider the normalized dis-

Table 1. Comparison of the expected distribution of profiles over clusters with the calculated distribution

Expected		Calculated		
Cluster number	Number of profiles	Number of profiles	Missed	Added
1	10	16	2	8
2	15	19	0	4
3	20	52	20	52
4	25	31	1	7
5	30	33	0	3
6	35	40	0	5
7	40	40	0	0
8	50	50	0	0
9	55	52	4	1
10	60	59	1	0
11	70	56	16	2
12	75	91	15	31
13	80	87	0	7
14	85	80	5	0
15	90	99	1	10
16	95	100	10	15[a]
17	105	77	39	11
18	120	122	7	9
19	130	129	2	1
20	140	143	3	6
21	145	151	0	6
22	160	155	6	1
23	165	123	53	11
24	200	195	6	1
Sum	2000	2000	191	191

[a] All from cluster #3.

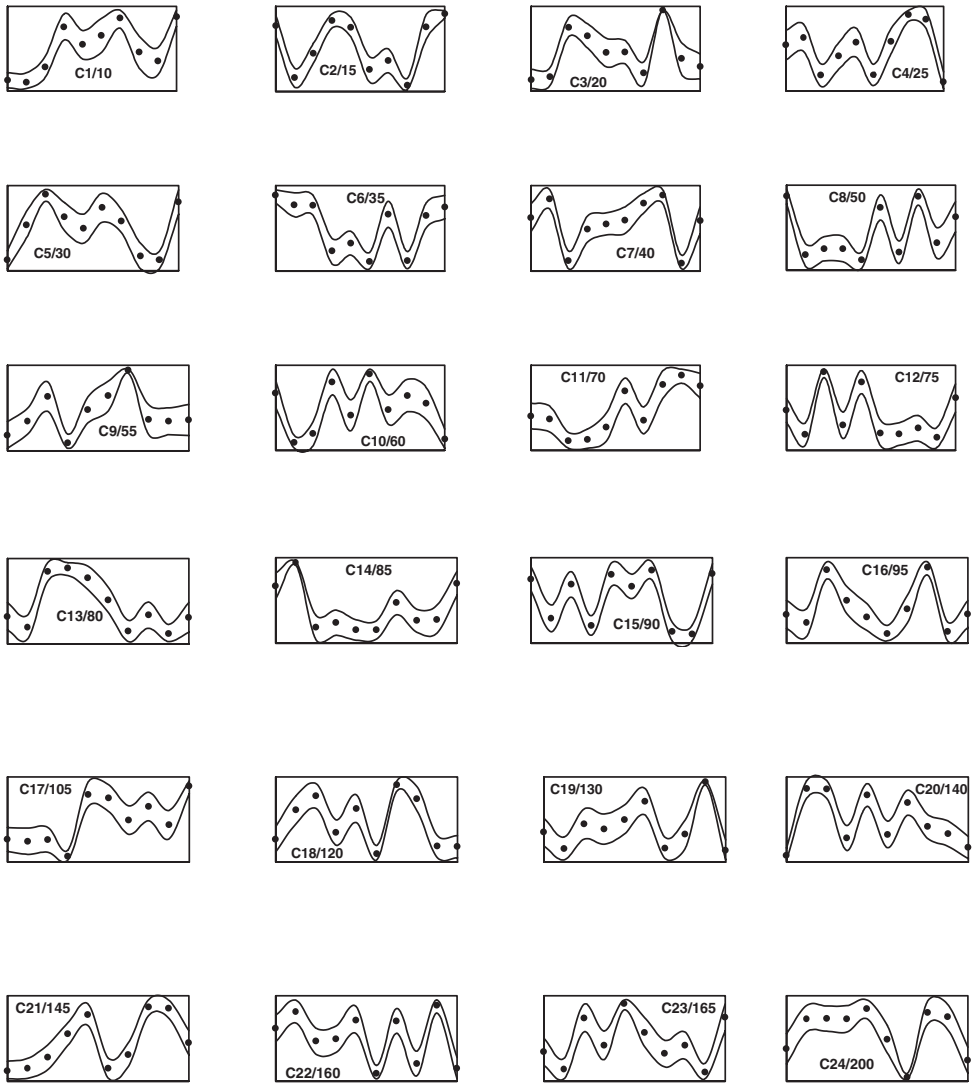

Fig. 1. Shapes of 2000 randomly generated temporal patterns of gene expression grouped in 24 clusters. The horizontal axis on each template represents time in linear scale. The vertical axis is the expression level ranging from 0 to 1. Each cluster is represented by the average pattern for profiles in the cluster (filled circles). Smooth curves indicate the standard deviation of average expression. Ck/n stands for 'cluster #k contains n individual profiles'. The weighted average standard deviation SD for this data set is equal to 0.15.

tribution function of distances between profiles with randomly shuffled data points. The probability Q that two randomly generated profiles will have a distance between them of less than or equal to D is defined by Equation (6)

$$Q(D) = \int_o^D g(x)\,\mathrm{d}x \qquad (6)$$

where $g(x)$ represents the normalized distribution function of distances between profiles with randomly shuffled time points. $Q(D)$ is a measure of the probability of finding two profiles randomly clustered together in a given data set. The dashed curve in Figure 2A shows the distribution function of randomized profiles for our reverse engineering data set and yields $Q(D) = 0.05$ for our choice of P and the corresponding optimized cutoff value $D = 1.10$. We can now derive D for a new data set as the solution of Equation (6), where the function $g(x)$ is the result of the randomization of the new data set. The optimal number of clusters K can then be determined as described above.

RESULTS AND DISCUSSION

We validated the clustering approach described here in two ways. The efficiency of the algorithm was evaluated

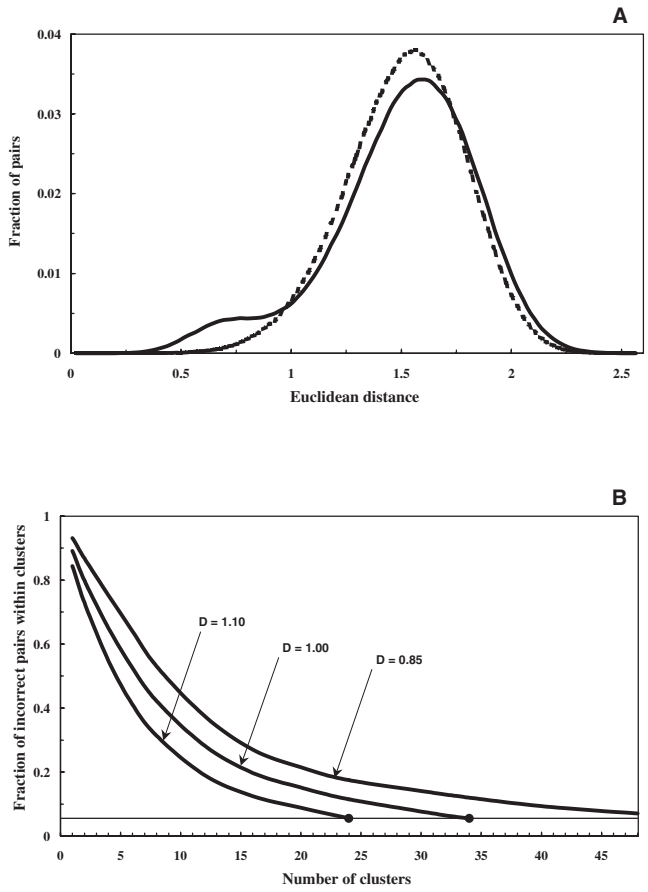

Fig. 2. Reverse engineering experiment. (A) Solid curve represents the normalized distribution function of distances between the 2000 profiles generated to form 24 clusters as described in the text (see also Figure 1). This function is the result of the overlay of two curves, one that corresponds to distances between profiles within clusters (the left shoulder), and another that corresponds to distances between profiles belonging to different clusters (the right peak). For comparison, the dashed curve shows the distribution function for the same set with shuffled time points (the shuffling destroys similarity between profiles within a cluster, and the left maximum disappears). (B) The fraction of incorrect pairs (Equation 4) as a function of the number of clusters for the optimized distribution of profiles over clusters. The thick lines correspond to three different values of the cutoff distance D. The thin horizontal line corresponds to a P-value of 0.055.

in a statistically rigorous way by means of a reverse engineering experiment in which the correct solution was known *a priori*. The biological relevance of the clustering produced by our algorithm was illustrated by its application to a gene expression data set (Cho *et al.*, 1998) for which correct clustering was recognized previously by means of a variety of different approaches including visual inspection.

Reverse engineering experiment

The similarity metric (1) and the cost function (2) we use for the clustering represent only one possibility among a variety of other approaches. For example, a Bayesian model could be used instead of Euclidean distances or a term that maximizes distances between clusters could be added to the cost function (2). Our specific choices were motivated by the simplicity of the expressions (1) and (2), the fact that the simulated annealing algorithm generates robust output, and the corresponding fast speed of its software implementation. Of course, this particular choice may affect the results, and a statistically rigorous test for the efficiency of the algorithm is desirable. As a rule, different clustering algorithms are tested on real gene expression data in situations in which the 'correct' solution is unknown, and the quality of clustering is assessed by the biological relevance of the results. In the present paper, we evaluate the efficiency of our algorithm by means of a reverse engineering experiment, that is, by utilizing an approach in which the correct solution is known *a priori*.

The details of constructing a data set for our reverse engineering experiment are described in the Methods. Figure 1, along with the list of the distribution of particular profiles over clusters, represents the *expected* solution. Our goal was to retrieve this solution from the individual profiles only, without any knowledge about the correct assignment of profiles to clusters. We clustered all profiles by minimizing the cost function (2) for the number of clusters $K = 24$ which is the optimal number of clusters as described above. Figure 3 presents the resulting clustering for the set of parameters ($P = 0.055$; $D = 1.10$; $K = 24$).

A visual comparison of Figures 1 (the expected clustering) and 3 (the calculated clustering) shows only two significant changes: (i) the standard deviation of cluster #1 is markedly larger for the calculated clustering, and (ii) cluster #3 disappeared (15 members of this cluster migrated to cluster #16), and instead of it a new cluster appeared, which we placed arbitrarily into the position of the former cluster #3. Table 1 provides detail information about the re-distribution of profiles over clusters. Our algorithm distributes 1809 profiles out of 2000 as expected, demonstrating an efficiency of the algorithm at a level of 90% correct answers. Note that this is a lower boundary of its efficiency because not every re-distribution of profiles between clusters may actually be wrong. For example, the initial clusters #3 and #16 (Figure 1) are quite similar, and the fact that our clustering procedure transfers 15 members of cluster #3 into cluster #16 is thus not surprising. Of course, we verified that for the trivial case of standard deviation $SD = 0$ (all profiles within a given cluster are identical) our algorithm is able to find the exact solution (data not shown).

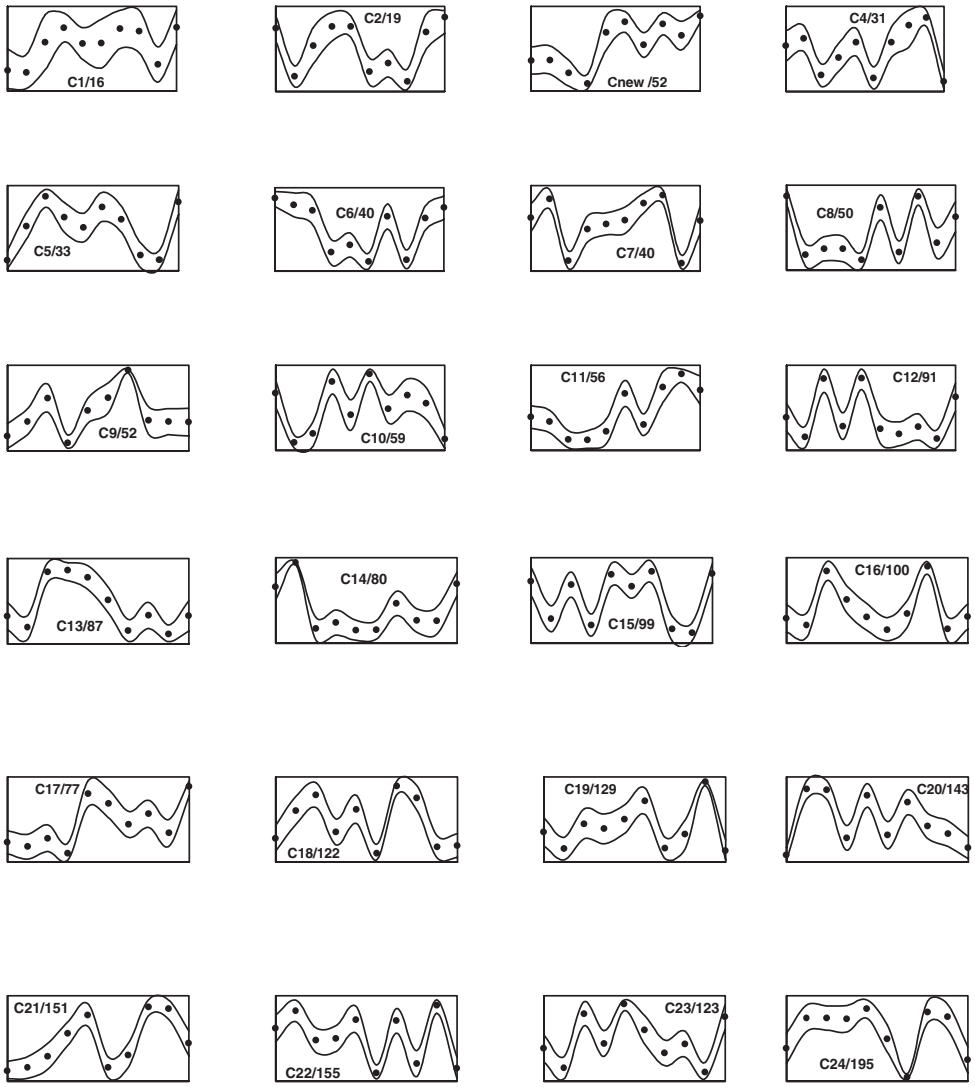

Fig. 3. The results of the reverse engineering experiment. The same 2000 temporal profiles whose fundamental patterns are shown in Figure 1 were clustered by means of the simulated annealing algorithm without utilizing any previous knowledge. The shapes of the patterns presented in this figure should be compared with Figure 1. To make the comparison easier we re-assigned cluster numbers for calculated clusters such that if a calculated cluster and one of the expected clusters are similar to each other they would be in the same topographical position in both figures.

Biological validation: yeast cell cycle

The yeast cell cycle data set provided by Cho *et al.* (1998) has established itself as a *de facto* standard for the assessment of newly developed clustering algorithms. This set contains time-course expression profiles for more than 6000 genes, with 17 time points for each gene taken at 10-min intervals covering nearly two yeast cell cycles (160 min). This data set is very attractive because a large number of genes contained in it are biologically characterized and have been assigned to different phases of the cell cycle. Our goal here was to demonstrate that our algorithm is able to extract biologically relevant fundamental patterns of expression, such as cell-cycle periodicity, without any *a priori* knowledge.

The raw expression profiles were downloaded from http://genomics.stanford.edu. A variation filter was used to eliminate those genes whose expression levels were relatively low and genes that did not show significant changes during the time-course. Specifically, the following conditions had to be satisfied for a gene to be retained in the data set: (i) an absolute value of expression at all 17 time points of equal to or greater than 100 (in units

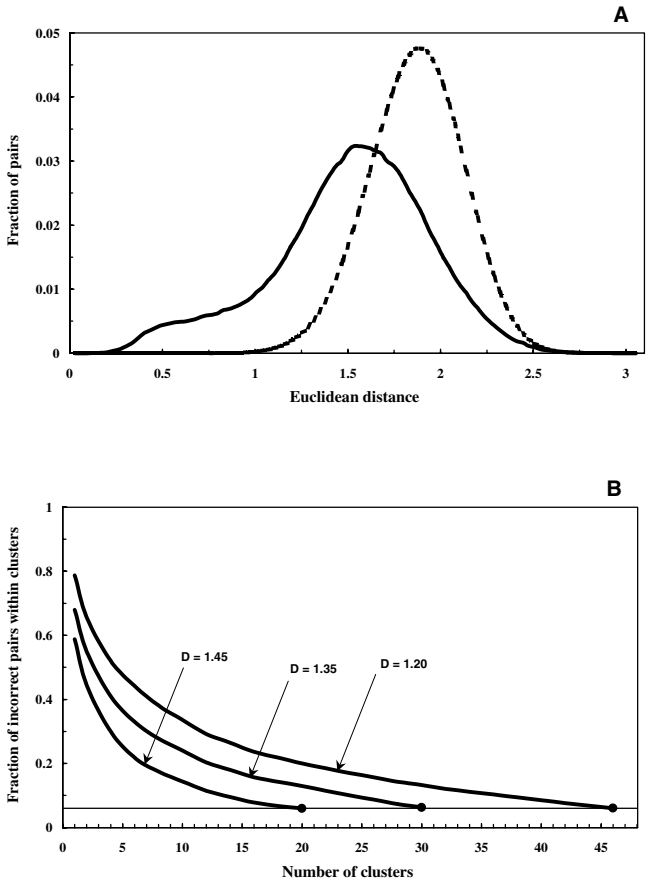

Fig. 4. Yeast data set. (A) Solid curve shows the normalized distribution of Euclidean distances between the 1306 gene expression profiles that passed our variation filter. As in Figure 2A, for comparison, the dashed curve represents the distribution function for profiles with randomly shuffled time points. (B) The fraction of incorrect pairs (Equation 4) as a function of the number of clusters for the optimized distribution of profiles over clusters. The thick lines correspond to three different values of the cutoff distance D. The thin horizontal line corresponds to a P-value of 0.055.

used in the downloaded file); (ii) at least a 2.5-fold change in expression level during the time-course. The profiles for the 1306 genes that passed the variation filter were normalized such that expression level for each gene varied between 0 and 1. After transforming the profiles into vectors the Euclidean distances (1) between all vectors were calculated. Figure 4A shows the distribution function of those distances. Next we used our simulated annealing approach to generate clusters of expression profiles and applied the conceptual framework described above to identify the optimal number of clusters for this data set. We calculated the fraction of incorrect vector pairs as described above for different numbers of clusters K and cutoff distances D. Figure 4B depicts the dependence of the fraction of incorrect vector pairs (4) on three different values of the cutoff distance D (compare with Figure 2B). Those specific values were chosen to relate the statistical and biological validation of our algorithm in a quantitative manner: the distances displayed in Figure 4B yield the same fraction $Q(D)$ of distances $d_{ij} \leqslant D$ between vectors with shuffled components for the cell cycle data set as the corresponding distances from the reverse engineering experiment shown in Figure 2B. For example, comparing the dashed curves in Figures 2A and 4A indicates that $Q(D) = 0.05$ for $D = 1.10$ in the reverse engineering experiment and $D = 1.45$ for the cell cycle data set. Having determined by applying our conceptual framework that an accepted fraction of false positives $P = 0.055$ corresponds to $Q(D) = 0.05$, we can now derive the optimal cutoff distance $D = 1.45$ from Figure 4A and the optimal number of clusters $K = 20$ from Figure 4B. Figure 5 shows the 20 fundamental patterns of gene expression during two yeast cell cycles that our simulated annealing clustering algorithm identified. Note that these clusters are distinctive (there is no strong visual similarity between patterns) and that some of them exhibit vividly the periodic behavior (for example, clusters #3, 4, 12, 18) that is to be expected from this data set. It is useful to compare 20 clusters shown in Figure 5 of the present paper with clusters shown in Figure 2a of the paper by Tamayo *et al.* (1999), where practically the same set of profiles were grouped into 30 clusters. It is seen that 30 clusters in this case overestimate the optimal number of clusters. For example, clusters #24, 28 and 29 (Tamayo *et al.*, 1999, Figure 2a), demonstrate the same shape and definitely could be combined into one group. Our algorithm generates only one cluster of this shape (Figure 5, cluster #18).

The next step of our analysis was the use of known biological information to verify that our algorithm is indeed able to extract patterns that correspond to different phases of the yeast cell cycle. To this end, we used the list of biologically characterized genes together with their assignment to particular cell cycle phases from Table 1 of the paper by Cho *et al.* (1998). Table 2 of the present paper shows how the 111 genes from Cho's list that passed our variation filter are distributed between the clusters depicted in Figure 5. These genes are found in six clusters (Table 2). In Figure 6 five clusters are presented that obviously correspond to five cell cycle phases: early G_1, late G_1, S, G_2 and M. Note that some genes that were assigned to the same phase in the Cho paper (Cho *et al.*, 1998) are placed in different clusters by our algorithm (Table 2). We verified that these genes indeed have significantly different patterns of expression (data not shown).

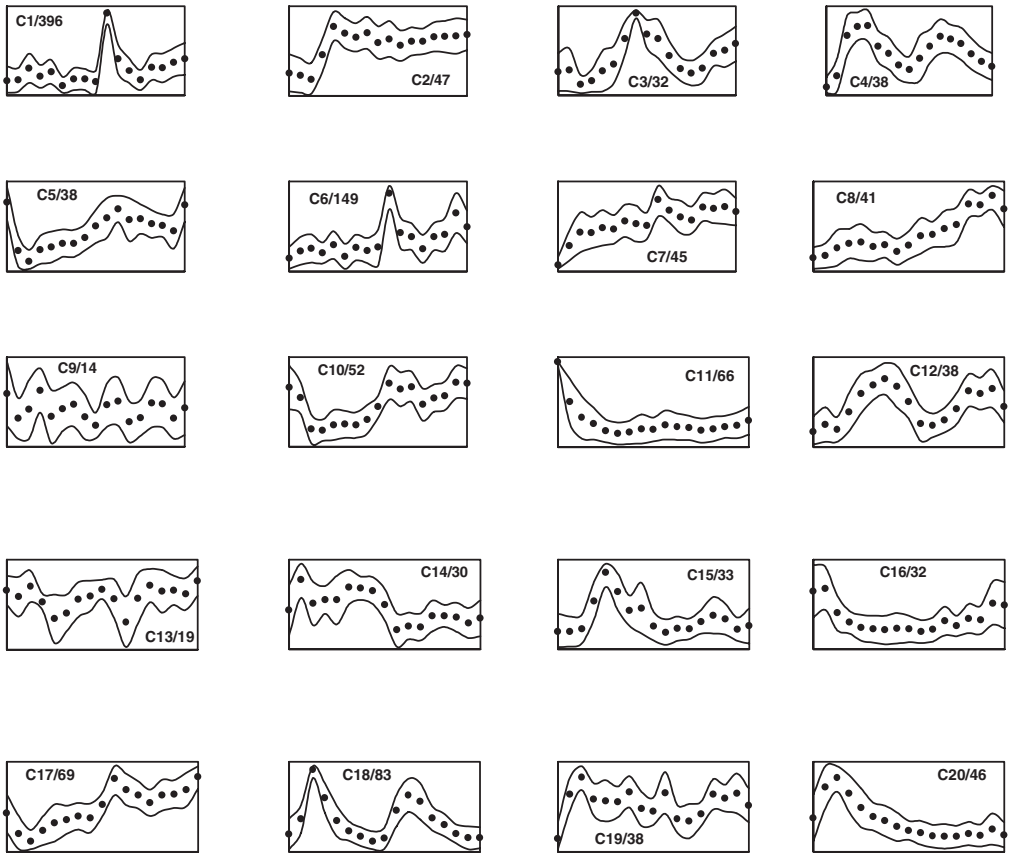

Fig. 5. The 1306 yeast genes that passed the variation filter were grouped into 20 clusters. The horizontal axis on each template represents time ranging from 0 to 160 min. The vertical axis is the normalized expression level ranging from 0 to 1. Each cluster is represented by the average pattern for profiles in the cluster (filled circles). Smooth curves indicate the standard deviation of average expression. Ck/n stands for 'cluster #k contains n genes'.

Table 2. Distribution of biologically characterized genes (Cho *et al.*, 1998) over the clusters shown in Figure 6

Cluster #	1	3	4	5	12	18
Early G_1 (22/32)	15	2	–	5	–	–
Late G_1 (43/83)	1	–	4	–	–	38
S-phase (18/46)	–	–	14	–	4	–
G_2-phase (13/33)	–	–	4	–	9	–
M-phase (15/34)	1	10	–	–	4	–

After the name of each cell cycle phase, the number of genes that passed our variation filter is shown relative to the total number of genes belonging to this category listed in Table 1 of Cho *et al.* (1998). The total number of genes in each cluster is depicted in Figure 5. Only those clusters are shown which contain at least one biologically characterized gene that passed the variation filter.

CONCLUSION

In this paper we propose a simple and robust clustering algorithm aiming to find not only an optimal distribution of expression profiles over clusters but also to simultaneously identify the number of clusters that is optimal for a given data set. We have verified the efficiency of the algorithm by means of reverse engineering experiment and by analyzing real experimental data for which the biological relevance of the results can be recognized. The next crucial question to answer is: what constitutes the biological meaning behind the clustering? In other words, given a set of clusters having characteristic shapes of expression profiles, how can we extract information about interconnectivity and mutual regulation of genes that belong to different clusters? Several computational schemes that are trying to address this issue can be found in the current literature (see, for example Chen *et al.*, 1999; Tavazoie *et al.*, 1999; Weaver *et al.*, 1999). The input data for these techniques is a set of expression patterns grouped into clusters; the output is a regulatory network. Obviously, the quality and biological relevance of the resulting regulatory network depend strongly on the quality of clustering as well as the chosen number of clusters. We believe that the algo-

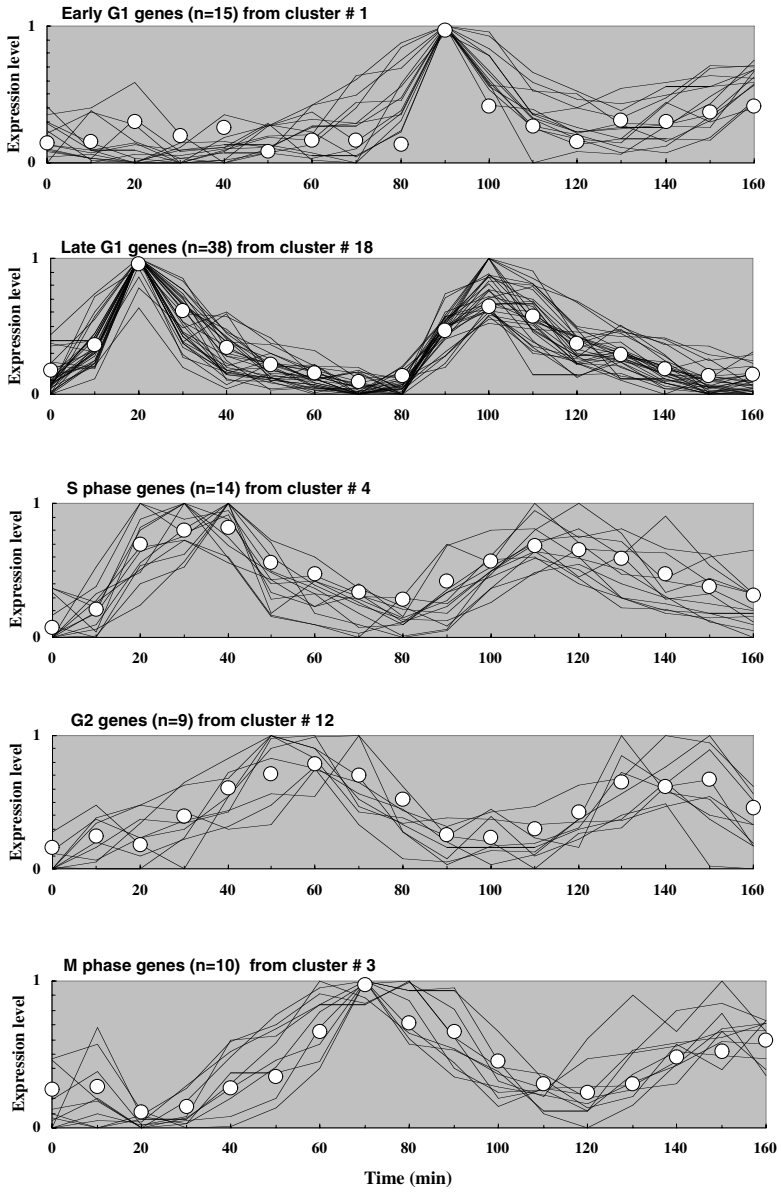

Fig. 6. Five fundamental patterns taken from Figure 5 that correspond to the five cell cycle phases. Expression levels are shown on the vertical axis and time points on the horizontal axis. On each template, open circles represent the average pattern for all profiles in the cluster. Solid lines represent individual expression profiles. The genes presented are only those that belong to this cluster *and* are biologically characterized and assigned to a specific cell cycle phase (Cho *et al.*, 1998). The cell cycle phases to which these genes were assigned are shown on the top of each template together with the number of genes that passed the variation filter (see Table 2). Note the periodic behavior of the fundamental patterns and consistency of cell cycle phase changes with sequential shifts of peak positions from early to late time.

rithm we presented here that controls both issues can be a helpful tool for the identification and modeling of biologically meaningful regulatory networks.

REFERENCES

Aart,E.H.L. and van Laarhoven,P.J.M. (1987) *Simulated Annealing: a Review of the Theory and Applications*. Kluwer, Dordrecht.

Alon,U., Barkai,N., Notterman,D.A., Gish,K., Ybarra,S., Mack,D. and Levine,A.J. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, **96**, 6745–6750.

Brown,M.P.S., Grundy,W.N., Lin,D., Cristianini,N., Sugnet,C.W., Furey,T.S., Ares,M. and Haussler,D. (2000) Knowledge-based analysis of microarray gene expression data by using support

vector machines. *Proc. Natl Acad. Sci. USA*, **97**, 262–267.

Chen,T., He,H.L. and Church,G.M. (1999) Modeling gene expression with differential equations. *Pacific Symposium on Biocomputing'99.* World Scientific, Singapore, pp. 29–40.

Cho,R.J., Campbell,M.J., Winzeler,E.A., Steinmetz,L., Conway,A., Wodicka,L., Wolfsberg,T.G., Gabrielian,A.E., Landsman,D., Lockhart,D.J. and Davis,R.W. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65–73.

Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14 863–14 868.

Hartigan,J. (1975) *Clustering Algorithms*. Wiley, New York.

Jain,A. and Dubes,R. (1988) *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NJ.

Kirkpatrick,S., Gelatt,C.D. and Vecchi,M.P. (1983) Optimization by simulated annealing. *Science*, **220**, 671–680.

Lockhart,D.J. and Winzeler,E.A. (2000) Genomics, gene expression and DNA arrays. *Nature*, **405**, 827–846.

Perou,C.M., Jeffrey,S.S., van de Rijn,M., Rees,C.A., Eisen,M.B., Ross,D.T., Pergamenschikov,A., Williams,C.F., Zhu,S.X., Lee,J.C.F., Lashkari,D., Shalon,D., Brown,P.O. and Botstein,D. (1999) Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl Acad. Sci. USA*, **96**, 9212–9217.

Roberts,C.J., Nelson,B., Marton,M.J., Stoughton,R., Meyer,M.R., Bennett,H.A., He,Y.D., Dai,H., Walker,W.L., Hughes,T.R., Tyers,M., Boone,C. and Friend,S.H. (2000) Signaling and circuitry of multiple MAPK pathways revealed by matrix and global gene expression profiles. *Science*, **287**, 873–880.

Ross,D.T., Scherf,U., Eisen,M.B., Perou,C.M., Rees,C., Spellman,P., Iyer,V., Jeffrey,S.S., van de Rijn,M., Waltham,M., Pergamenschikov,A., Lee,J.C.F., Lashkari,D., Shalon,D., Myers,T.G., Weinstein,J.N., Botstein,D. and Brown,P.O. (2000) Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genet.*, **24**, 227–235.

Spellman,P.T., Sherlock,G., Zhang,M.O., Iyer,V.R., Anders,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.

Tamayo,P., Slonim,D., Mesirov,J., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S. and Golub,T.R. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.

Tavazoie,S., Hughes,J.D., Campbell,M.J., Cho,R.J. and Church,G.M. (1999) Systematic determination of genetic network architecture. *Nature Genet.*, **22**, 281–285.

Weaver,D.C., Workman,C.T. and Stormo,G.D. (1999) Modeling regulatory networks with weight matrices. *Pacific Symposium on Biocomputing'99*. World Scientific, Singapore, pp. 112–123.

Wen,X., Fuhrman,S., Michaels,G.S., Carr,D.B., Smith,S., Barker,J.L. and Somogayi,R. (1998) Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl Acad. Sci. USA*, **95**, 334–339.

White,K.P., Rifkin,S.A., Hurban,P. and Hogness,D.S. (1999) Microarray analysis of *Drosophila* development during metamorphosis. *Science*, **286**, 2179–2184.

Young,R.A. (2000) Biomedical discovery with DNA arrays. *Cell*, **102**, 9–15.

Zweiger,G. (1999) Knowledge discovery in gene-expression-microarray data: mining the information output of the genome. *Trends Biotechnol.*, **17**, 429–436.