An Interactive Approach to Mining Gene Expression Data

Daxin Jiang, Jian Pei, *Member*, *IEEE Computer Society*, and Aidong Zhang, *Member*, *IEEE*

Abstract—Effective identification of coexpressed genes and coherent patterns in gene expression data is an important task in bioinformatics research and biomedical applications. Several clustering methods have recently been proposed to identify coexpressed genes that share similar coherent patterns. However, there is no objective standard for groups of coexpressed genes. The interpretation of co-expression heavily depends on domain knowledge. Furthermore, groups of coexpressed genes in gene expression data are often highly connected through a large number of "intermediate" genes. There may be no clear boundaries to separate clusters. Clustering gene expression data also faces the challenges of satisfying biological domain requirements and addressing the high connectivity of the data sets. In this paper, we propose an *interactive* framework for exploring coherent patterns in gene expression data. A novel *coherent pattern index* is proposed to give users highly confident indications of the existence of coherent patterns. To derive a coherent pattern index and facilitate clustering, we devise an *attraction tree* structure that summarizes the coherence information among genes in the data set. We present efficient and scalable algorithms for constructing attraction trees and coherent pattern indices from gene expression data sets. Our experimental results show that our approach is effective in mining gene expression data and is scalable for mining large data sets.

Index Terms—Bioinformatics, gene expression (microarray) data, clustering, interactive data mining.

1 Introduction

Microarray technology can simultaneously monitor the expression levels of thousands of genes during important biological processes and across collections of related samples. An important task of analyzing gene expression data is the detection of coexpressed genes and coherent gene expression patterns. A group of *coexpressed genes* exhibits a common expression pattern, while a *coherent gene expression pattern* (or, briefly, *coherent pattern*) characterizes the collective trend of the expression levels of a group of coexpressed genes. In other words, a coherent pattern is a "template," while the expression profiles of the corresponding coexpressed genes conform to the template with only small divergences.

For example, Iyer et al.'s data set [17] records the expression profiles of 517 human genes with respect to a 12-point time-series. In [17], Iyer et al. gave a list of 10 groups of coexpressed genes and the corresponding coherent gene expression patterns in the data set, which has been well accepted as the ground truth. In Fig. 1, we plot three groups of coexpressed genes and their corresponding coherent patterns from the ground truth. The top row shows the expression profiles of genes in each of the three groups. The profiles in each group appear to share a common trend shown in the bottom row, which is the point-wise median of the profiles. The error bars indicate the standard deviations.

- *D. Jiang and A. Zhang are with the Department of Computer Science and Engineering, State University of New York at Buffalo, 201 Bell Hall, Buffalo, NY 14260. E-mail: {djiang3, azhang}@cse.buffalo.edu.*
- *J. Pei is with the School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, BC VSA1S6 Canada. E-mail: jpei@cs.sfu.ca.*

Manuscript received 23 Mar. 2004; revised 8 Mar. 2005; accepted 24 Mar. 2005; published online 18 Aug. 2005.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0079-0304.

Why are clustering coexpressed genes and finding coherent patterns interesting and meaningful? As indicated by previous studies, coexpressed genes may belong to the same or similar functional categories and indicate co-regulated families [35], while coherent patterns may characterize important cellular processes and suggest the regulating mechanism in the cells [26].

To find coexpressed genes and identify coherent patterns, various clustering algorithms (e.g., [1], [4], [8], [18], [31], [32], [34], [35]) have been developed to partition a set of genes into clusters. Each cluster is considered as a group of coexpressed genes, and the corresponding coherent pattern can be simply the centroid of the cluster. Previous studies have confirmed that clustering algorithms are useful in identifying coexpressed gene groups and coherent patterns. However, the specific characteristics of gene expression data and special requirements arising from the domain of biology still pose challenges to the effective clustering of gene expression data.

1.1 Challenge 1: It Is Subtle to Unfold the Hierarchies of Coexpressed Genes and Coherent Patterns

A microarray data set typically contains multiple groups of coexpressed genes and their corresponding coherent patterns. As a general observation, *there is usually a hierarchy of coexpressed genes and coherent patterns in a typical gene expression data set*. For example, as shown in Fig. 2, a group of coexpressed genes S taken from Iyer's data set can be split into two subgroups S_1 and S_2, and S_2 can be further split into two subsubgroups S_{21} and S_{22}. The expression profiles of genes within each smaller subgroup become increasingly more uniform and the patterns more coherent when compared with the higher-level groups. Therefore, these groups of coexpressed genes form a hierarchy. At the upper levels of the hierarchy, large groups of genes

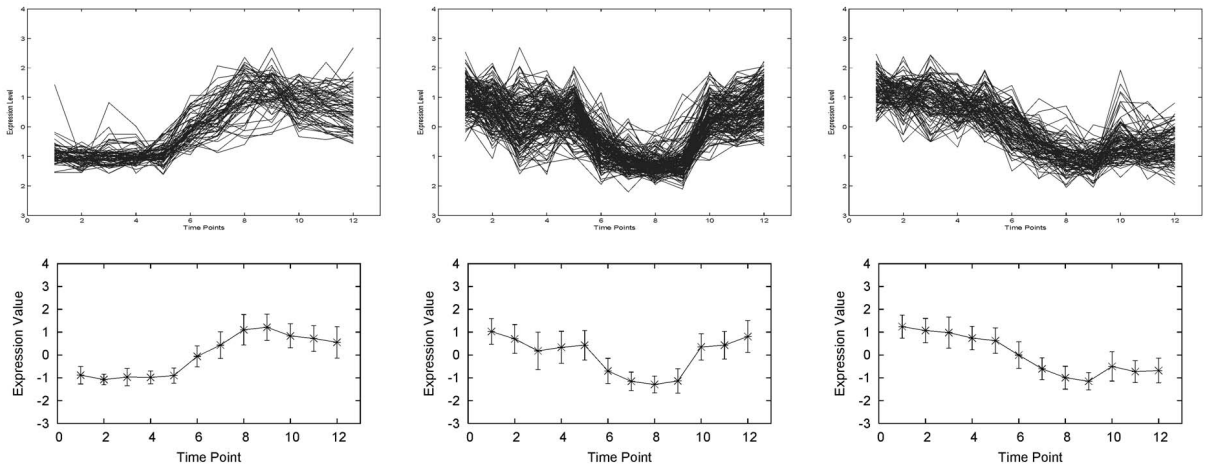


Fig. 1. Examples of coexpressed gene groups and corresponding coherent patterns.

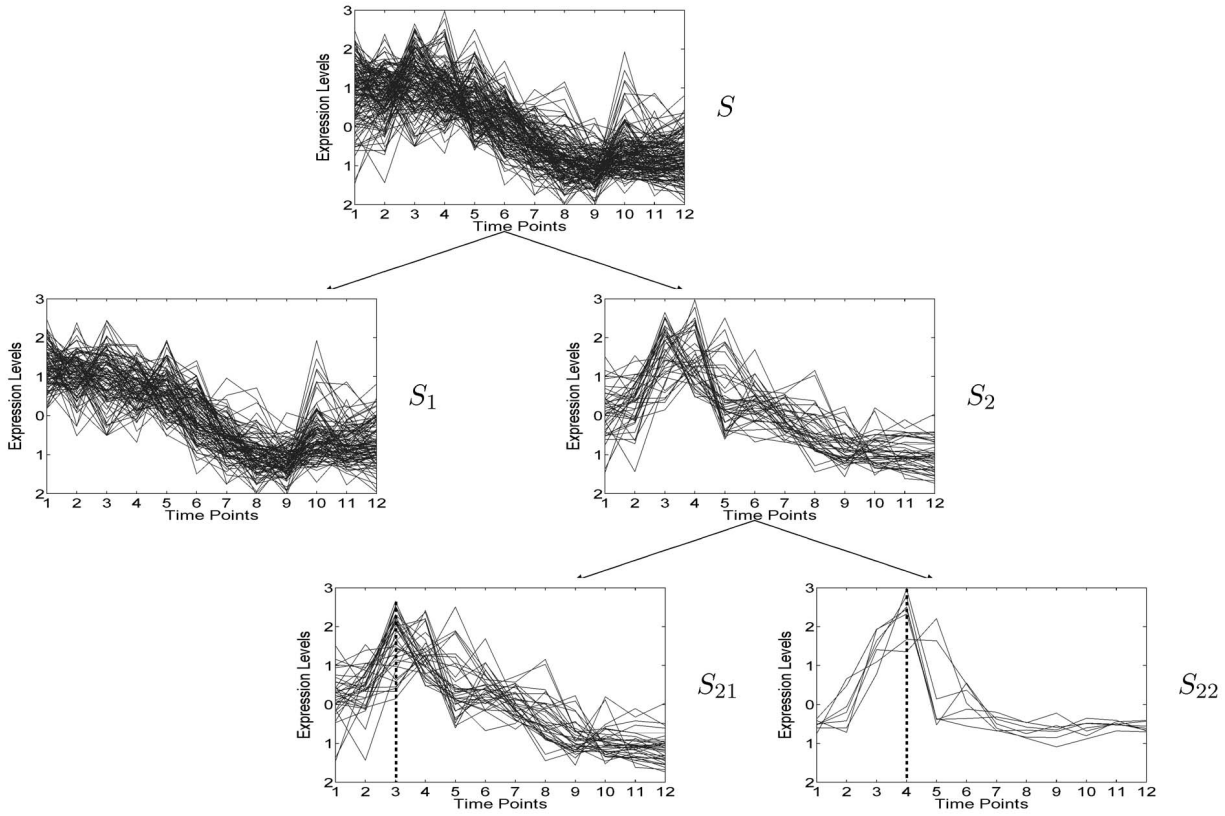


Fig. 2. The hierarchy of a coexpressed gene group.

generally conform to "rough" coherent expression patterns. At lower hierarchical levels, these larger groups are broken into smaller subgroups. Those smaller groups of coexpressed genes conform to "fine" coherent patterns; these patterns inherit some features from the "rough" patterns and add some distinct characteristics.

The apparent simplicity of this organization is complicated by *the lack of a rigorous definition or objective standard to unambiguously identify coexpressed gene groups*. The interpretation of co-expression often depends on the knowledge from domain experts. Typically, three situations may happen in the analysis of gene expression data:

- Biologists can often bring some prior knowledge to the analysis of a microarray data set. For example, some genes are known to be closely related in function, while some genes are known not to stay in the same cluster. If such prior knowledge is integrated into the clustering process, the mining results may be substantially improved.
- A microarray experiment often involves thousands of genes. However, only a small subset (perhaps several hundred) of those genes may play important roles in the underlying biological processes. In an initial examination, biologists may browse through the "rough" patterns in the data set. They may then choose several patterns of particular interest and

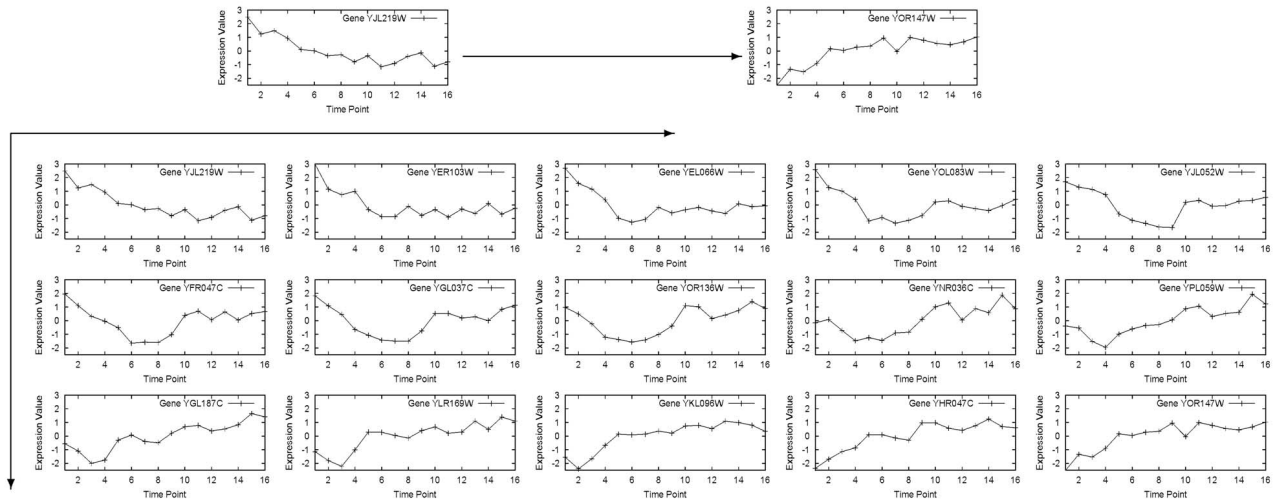


Fig. 3. The gradual change from one expression profile to a completely different profile.

decompose them into "finer" patterns in further analysis. In other words, biologists may have different requirements for the coherence of different parts of the data set.

- The domain knowledge of biologists is typically incomplete. That is, the functions of many genes in a data set are still unclear, and there could be various hypotheses regarding the functions of those genes. For example, in Fig. 2, the subset of genes S_2 can be split into two subsubsets S_{21} and S_{22}. The genes in S_{21} and S_{22} exhibit similar expression profiles. The critical difference is that the genes in S_{21} are up-regulated[1] at the third time point, while the expression levels of genes in S_{22} peak at the fourth time point. Two hypotheses could explain this phenomenon. It is possible that the genes in S_{22} are up-regulated by the genes in S_{21}.[2] If this is the case, it is meaningful to split S_2 into S_{21} and S_{22} may have similar functions, and it would be appropriate not to split S_2. Given such uncertainties, biologists would prefer an exploratory tool which can illustrate the possible options for partitioning the data set and assist in evaluating the range of hypotheses based on the underlying data structure.

Can we provide a flexible tool which allows biologists to interactively unfold the hierarchy of groups of coexpressed genes and derive the corresponding coherent patterns? Various users may want to explore the structure of a data set using a variety of criteria according to their research goals and background knowledge.

1.2 Challenge 2: It Is Difficult to Address the High Connectivity of Gene Expression Data Sets

In gene expression data, there are typically a large amount of genes which stay between the groups of coexpressed genes. These genes are called "intermediate" genes since they build "bridges" across different coexpressed gene groups. An example taken from yeast expression data ($CDC28$ [33]) is shown in Fig. 3. The two genes in the first row have very different expression profiles and, thus, cannot belong to the same coexpressed gene group. However, in the same data set, we can find a series of genes in which each gene is quite similar to its predecessor; such a series is illustrated in the lower rows of Fig. 3.

The biological role of "intermediate" genes can be different. On the one hand, some "intermediate" genes may participate in multiple cellular processes and, thus, should be classified into multiple clusters. On the other hand, the majority of "intermediate" genes may not involve in any biological processes of interest and, thus, do not belong to any clusters. In other words, these "intermediate" genes are simply noise. For example, in [7], only 416 out of 6,220 monitored transcripts were recognized as five cell-cycle regulated clusters, while the remaining 5,804 located around the clusters were considered as noise. Among the 416 cell-cycle regulated genes, 22 belong to multiple cell cycle phases. The large amount of "intermediate" genes pose a big challenge: *Gene expression data are often highly connected, and it is difficult to determine the borders between clusters*. Most existing methods make the decisions by force and may fall in one of the following two situations:

- The data set is decomposed into numerous small clusters. Some clusters will consist of groups of coexpressed genes, while many clusters will be made up of intermediate genes. Since there is no absolute standard, such as size or compactness, with which to rank the resulted clusters, it may require significant user effort to distinguish meaningful clusters from those trivial ones. This situation is illustrated in Fig. 4a.
- The data set is decomposed into several large clusters, each of which contains both coexpressed genes and many intermediate genes. However, the heavy representation of intermediate genes may lead to the skewing of cluster centroids. These "warped" centroids do not accurately represent the coherent patterns in the groups of coexpressed genes. This situation is exemplified in Fig. 4b.

1. A gene is called "up-regulated" when its expression level increases significantly.

2. The genes in S_{21} are up-regulated at the third time point. The product of those genes may in turn cause the up-regulation of the genes in S_{22} at the fourth time point.

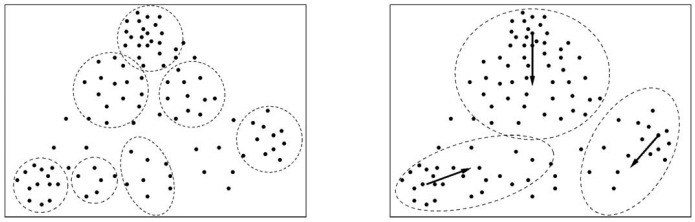


Fig. 4. Handling intermediate genes. (a) Deriving many (small clusters) and (b) deriving large clusters.

For both situations, the crisp borders forced between clusters do not allow a single gene to participate in multiple clusters.

In this paper, we examine the challenges of mining coherent patterns from gene expression data and make the following contributions:

- We propose a framework of *interactive exploration* for the analysis of microarray data. This approach supports exploration by users as guided by their domain knowledge and accommodates disparate user requirements for varying degrees of coherence in different parts of the data set.
- We develop a novel strategy for handling "intermediate" genes. A user first identifies coherent patterns. He/she can then determine the borders of groups of coexpressed genes on the basis of the distance between a gene and the coherent patterns. In particular, an "intermediate" gene is allowed to participate in more than one cluster.
- We design a *coherent pattern index* to give users ranked indications of the existence of coherent patterns. To derive a coherent pattern index, we adopt a density-based model to describe the coherence relationship between genes and devise an *attraction tree* structure to summarize the coherence information for the interactive exploration.
- We conduct an extensive performance study on both synthetic data sets and some real-world gene expression data sets to verify our design. The experimental results indicate that our approach is effective and scalable in mining gene expression data.

The remainder of the paper is organized as follows: In Section 2, we review related work. The attraction tree structure is introduced in Section 3. In Section 4, we present the interactive exploration of coherent patterns using the coherent pattern index. An extensive performance study is reported in Section 5. We discuss some related issues in Section 6 and conclude the paper in Section 7.

2 Related Work

Clustering is the process of grouping data objects into a set of disjoint *clusters*, so that objects within a cluster have high similarity, while objects in different clusters are dissimilar. To find coexpressed genes and discover coherent expression patterns, a number of clustering algorithms have been applied—some are adapted from the previous methods and the others are newly devised. These algorithms can be classified into three categories: *partition-based approaches*, *hierarchical approaches*, *density-based approaches*, and *pattern-based approaches*.

2.1 Partition-Based Approaches

The partition-based algorithms divide a data set into several mutually exclusive subsets based on certain clustering assumptions (e.g., there are k clusters in the data set) and optimization criteria (e.g., minimize the sum of distances between objects and their cluster centroids). We can further divide the partition-based methods into four subcategories: *the K-means algorithm and its derivatives* [15], [22], [25], [32], [35], *the Self-Organizing Map (SOM) and its extensions* [14], [20], [34], [36], *graph-based algorithms* [4], [13], [31], [39], and *model-based algorithms* [10], [12], [23], [41].

Although partition-based approaches have been shown useful in identifying coexpressed genes and coherent expression patterns, they may not be effective in addressing the two challenges discussed in Section 1. Many partition-based approaches (such as K-means, SOM, and model-based algorithms) require users to input the number of clusters, which is often unknown a priori. Additionally, a partition-based approach usually makes brute force decisions on the cluster borders and, thus, may fall into one of the two situations illustrated in Fig. 4.

2.2 Hierarchical Approaches

Hierarchical approaches organize objects into a hierarchy of nested clusters called a *dendrogram*. Depending on how the dendrogram is formed, hierarchical approaches can be further divided into *agglomerative methods* [3], [8], [27] and *divisive methods* [1], [14], [18]. Hierarchical approaches typically have two fundamental components: a strategy for merging or splitting nodes and a principle for cutting the dendrogram to derive clusters.

Most hierarchical approaches adopt a specific merge/split strategy to form the dendrogram. The strategy is intrinsic to the algorithm and, thus, determines the clustering results. For example, different agglomerative approaches adopt different measures for *cluster proximity*, such as single link, complete link, minimum-variance, etc. The divisive approaches, such as *SPC* [1], [5], *DHC* [18] and *SOTA* [14], are characterized by their splitting criteria. The diverse range of clustering algorithms suggests that a given set of data objects in high-dimensional space can be partitioned in multiple ways. For gene expression data, different partitions may correspond to various hypotheses regarding gene functions. Biologists may be interested in evaluating a range of the hypotheses and selecting the most appropriate one on the basis of their domain knowledge. However, most existing approaches generate the hierarchical structure in a deterministic manner, so that users are not exposed to the universe of possible options.

Another component of the hierarchical approaches is a method for cutting the dendrogram to derive clusters. As illustrated in Fig. 5a, users employing *TreeView*, a popular analysis tool, have to traverse the graphical dendrogram

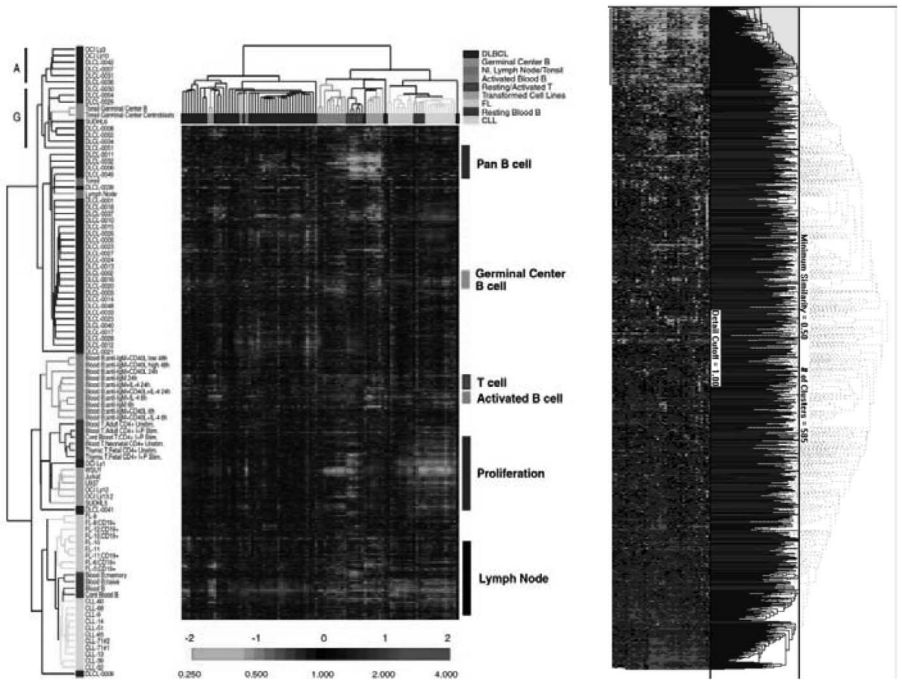


Fig. 5. Popular hierarchical tools. (a) Eisen's *TreeView* and (b) Seo et al.'s approach.

through visual inspection and derive the coexpressed genes. Although this gives users the flexibility of applying their domain knowledge, for a large data set with thousands of genes, a manual search on the graph is extremely ineffectively. Alternatively, Seo and Shneiderman [30] proposed the *minimum similarity bar* to cut the dendrogram and derive clusters (Fig. 5b). However, the minimum similarity bar is essentially a global parameter that restricts the minimum distance between derived clusters. It cannot adapt to various local structures within the data set, and the value of the bar is often difficult to determine.

2.3 Density-Based Approaches

The density-based approaches describe the distribution of a given data set by the "density" of data objects. The clustering process involves a search of the "dense areas" in the object space [9]. In [2], Ankerst et al. introduced the algorithm *OPTICS*, which does not generate clusters explicitly, but instead creates an ordering of the data objects and illustrates the cluster structure of the data set. However, when applied to a highly-connected data set, *OPTICS* may enter another dense area through "intermediate" data objects before traversing the current dense area thoroughly. Therefore, all genes following a single coherent pattern may not be accommodated consecutively in the order.

Another density-based approach, *DENCLUE* [16], measures object density from a global perspective. Data objects are assumed to "influence" each other, and the density of a data object is the sum of influence from all data objects in the data set. *DENCLUE* is robust to intermediate data objects. However, it outputs all clusters at the same level. Therefore, it cannot support an exploration of hierarchical cluster structures which exploits users' domain knowledge.

2.4 Pattern-Based Approaches

It is well-known in molecular biology that any cellular process may take place only in a subset of the attributes (samples or time points), and a gene may participate in multiple cellular processes. Recently, a series of *pattern-based clustering algorithms* have been proposed to capture coherence exhibited by a subset of genes on a subset of attributes.

In [6], Cheng and Church introduced the concept of *mean squared residue score* to measure the coherence between genes and attributes (either time series or samples). Given a set of genes and a set of conditions, a bicluster is a subset of genes coherent with a subset of attributes. A heuristic algorithm is also proposed. Yang et al. [40] have proposed a move-based, heuristic algorithm to find biclusters more efficiently. Both of these algorithms cannot be guaranteed to find the complete set of biclusters in a data set.

In [38], Wang et al. proposed a novel model of pattern-based cluster. A subset of objects O and a subset of attributes A form a pattern-based cluster (O, A) if, for any objects x, $y \in O$ and any attributes a, $b \in A$, the difference of changes of values on attributes a and b between objects x and y is smaller than a threshold δ. In a recent study [24], Pei et al. proposed mining nonredundant pattern-based clusters by an efficient algorithm *MaPle*. In addition, Liu and Wang [21] have proposed the concept of order-preserving clusters, which is a generalization of pattern-based clusters. In [42], the pattern-based approach was extended to mining gene-sample-time series microarray data.

3 The Attraction Tree

To enable the interactive exploration of coherent gene expression patterns, information regarding the coherence

among genes must be extracted and organized. In this section, we will introduce a density-based method for constructing an *attraction tree* structure. Once the attraction tree is built, the original data set will no longer need to be referenced. We will first describe the measure of the distance between two genes and the definition of the density of genes. The structure of the attraction tree will then be explicated.

3.1 The Distance Measure

Analysts of gene expression data are generally interested in the overall shapes of expression profiles rather than the absolute magnitudes. The commonly used Euclidean distance does not work well for scaling and shifting profiles [38]. Instead, *Pearson's correlation coefficient* is often used to measure the similarity between two expression profiles.

The definition of gene density starts with the measurement of the distance between two genes. For this purpose, Pearson's correlation, a similarity measure, has to be transformed into a distance measure. Given an object[3] O, we *normalize* the object to O' so that each O' has a mean of 0 and a variance of 1 over all attributes. The similarity and distance between the data objects are then defined respectively as

$$\begin{aligned} similarity(O_i, O_j) &= d_P(O'_i, O'_j) \text{ and} \\ distance(O_i, O_j) &= d_E(O'_i, O'_j), \end{aligned} \tag{1}$$

where $d_P(O'_i, O'_j)$ and $d_E(O'_i, O'_j)$ are the Person's correlation coefficient and the Euclidean distance between O'_i and O'_j, respectively. After normalization, the definitions of similarity and distance are consistent; i.e., given objects O_1, O_2, O_3, and O_4, $similarity(O_1, O_2) > similarity(O_3, O_4)$ implies $distance(O_1, O_2) < distance(O_3, O_4)$.

3.2 The Density Definition

The density of a data object O reflects the distribution of the other objects in O's neighborhood. The *radius-based* measure defines the density of O as the number of data objects within O's ε-neighborhood, where ε is a radius parameter specified by the user. As an alternative measure, the *k-nearest-neighbor density* (KNN) uses the distance between an object and its kth nearest neighbor; in this definition, a smaller distance indicates a higher density. However, these two approaches are sensitive to the global parameters ε and k, respectively, and it is usually difficult for users to specify appropriate parameter values.

A recently proposed method, *DENCLUE* [16], defines an *influence function* to describe the influence between two objects. The density of an object O is the sum of influences from all the objects in the data. *DENCLUE* avoids parameters such as ε and k. However, including the influence of distant data objects may corrupt the "real" cluster structure.

Basically, we want to measure the density of an object O as the sum of influences from objects within its own cluster, while ignoring the contribution of objects in other clusters. In [31], Shamir and Sharan use an *EM* algorithm to estimate the average pairwise similarity $\bar{S}$ between data objects within the same cluster. In this paper, we will use this method and, consequently, modify the density definition formulated by *DENCLUE* as

$$f(O_i, O_j) = e^{-\frac{distance(O_i, O_j)^2}{2\sigma^2}}, \tag{2}$$

$$density(O) = \sum_{O_j \in \mathcal{D}, similarity(O, O_j) \geq \bar{S}} f(O, O_j), \tag{3}$$

where $similarity(O, O_j)$ and $distance(O_i, O_j)$ are defined by (1). Objects O_i and O_j are called *neighbors* if

$$similarity(O_i, O_j) \geq \bar{S}.$$

We will address the determination of an appropriate value for parameter σ in Section 4.3.

3.3 Attraction Tree

Based on our density definition, a gene O_i is "influenced" by its neighbors. The direction of the *united influence* from all neighbors of O_i is an m-dimensional vector determined by

$$I(O_i)^{(d)} = \sum_{O_j \in \mathcal{D}, similarity(O_i, O_j) \geq \bar{S}} \frac{1}{f(O, O_j)} O_j^{(d)} \quad (1 \leq d \leq m), \tag{4}$$

where m is the number of attributes of object O_j and O_j^d is the dth attribute of O_j.

Intuitively, the direction of the united influence on object O_i indicates the dense region in O_i's neighborhood. If O_i moves toward the direction of $I(O_i)$, O_i is likely to reach an area with higher density. In particular, if an object O_i has a higher density than all of its neighbors, O_i is a *local maximum*. There are two special cases of a local maximum. In the first case, if an object O_i has no neighbors at all, O_i is a *noise object*. In the second case, if an object O_i has a higher density than any other object in the data set, O_i is the *global maximum*, denoted by O_{max}. We say a data object O_i is "attracted" by its *attractor* O_j (denoted by $O_i \rightarrow O_j$) according to the following definition:

$$Attractor(O_i) = \begin{cases} O_i & \text{if } O_i = O_{max} \\ \arg\max_{O_1 \in A_1} similarity(O_1, O_i) & \text{if } O_i \text{ is a Noise object} \\ \arg\max_{O_2 \in A_2} similarity(O_2, I(O_i)) & \text{if } O_i \text{ is not a local maximum} \\ \arg\max_{O_1 \in A_1} similarity(O_1, I(O_i)) & \text{otherwise.} \end{cases}$$

In the above definition, A_1 is the set of local maximums O_1 such that O_1 has a higher density than O_i, while A_2 is the set of O_i's neighbors O_2 such that O_2 has a higher density than O_i. The attraction from an object to another (i.e., $O_i \rightarrow O_j$) forms a partial order. Given any data object $O_i \neq O_{max}$, we can recursively trace the attractor of O_i until we reach O_{max}. Therefore, we can derive an *attraction tree* T where each node corresponds to an object O_i such that

$$Parent(O_i) = \begin{cases} nil & \text{if } O_i = O_{max} \\ Attractor(O_i) & \text{otherwise.} \end{cases}$$

We define the weight of each edge $e(O_i, O_j)$ as the similarity between O_i and O_j.

To locate the attractor of object O, all neighbors of O must be searched. However, for a large and highly-connected data set, this operation can be expensive. As an

3. Hereafter, we use the terms "objects" and "genes" interchangeably.

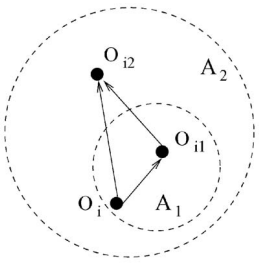


Fig. 6. The effect of K.

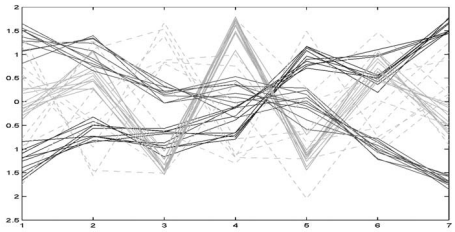


Fig. 7. A simplified synthetic gene expression data set.

approximation, we only keep the K nearest objects $O_j \in A(O)$ with respect to O. There are two extreme settings for K. When $K = 1$, the attractor of O becomes the nearest higher-density neighbor of O. At the other extreme, when $K = \infty$, the approximation is ignored, and all neighbors of O are exhaustively searched to find the attractor of O.

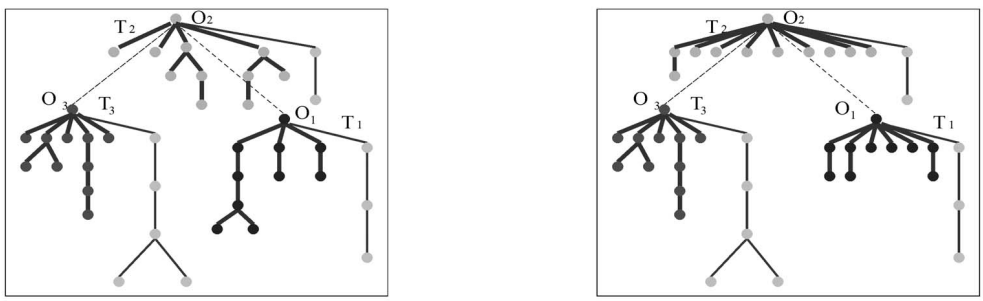
Intuitively, if K is set to a small number, we only need to search a relatively small neighborhood for the attractor. As a result, an object O_i may first be attracted to a "lower-level" local maximum O_{i1}, which covers a relatively small neighborhood A_1 (Fig. 6); while O_{i1} will then be attracted to a "upper-level" local maximum O_{i2}, which covers a larger neighborhood A_2. However, when we assign a large number of K, we will need to search a larger neighborhood for the attractor. In this situation, the object O_i is more likely to be directly attracted to the "upper-level" local maxima O_{i2}. That is, a smaller K favors a more detailed local structure, and, on average, a data object is attracted to the root of the attraction tree through more intermediate steps. In contrast, a large K "shortcuts" the attractor paths between a data object and the root and tends to delineate rough structures.

3.4 An Example

To illustrate the concept of the attraction tree, let us consider a synthetic data set $\mathcal{D}$ as represented in parallel coordinates in Fig. 7. This data set contains three coherent patterns, P_1, P_2, and P_3. We denote the groups of objects which conform to coherent pattern P_i as G_i and represent the objects conforming to P_1, P_2, and P_3 by the solid lines. There is also some noise in data set $\mathcal{D}$, and this is represented by the dot lines. Suppose O_1, O_2, and O_3 are the objects which have the locally maximal density in G_1, G_2, and G_3, respectively. In the resulting attraction tree, other objects in G_1, G_2, and G_3 will be attracted (directly or indirectly) to O_1, O_2, and O_3, respectively. Thus, O_1, O_2, and O_3 become the roots of attraction subtrees T_1, T_2, and T_3, respectively, where T_1, T_2, and T_3 contain all the objects of G_1, G_2, and G_3.

Fig. 8a is the attraction tree ($K = 1$) for $\mathcal{D}$. In this example, the density of O_2 is greater than that of both O_1 and O_3. Thus, O_2 has the globally maximal density and becomes the root of the tree. O_1 and O_3 are attracted to O_2 and become the roots of subtrees. Fig. 8b is the attraction tree for $\mathcal{D}$ with $K = \infty$, and it reveals a structure similar to that in Fig. 8a. However, objects in Fig. 8b tend to be attracted directly to upper-level attractors, giving that tree a flatter structure.

This example demonstrates two characteristics of the attraction tree structure. First, the attraction tree is *self-closed*. A group of objects conforming to the same coherent pattern forms a attraction subtree. Objects conforming to different coherent patterns are not mixed in the same attraction subtree. Second, the attraction tree is *robust to noise*. The root of each attraction subtree has the locally maximal density and represents the coherent pattern for that attraction subtree. Objects matching the coherent pattern stay connected with each other, while noise objects are connected either to the roots of the subtrees or to each other. A child O_j of a subtree root O_i must conform to the same coherent pattern as O_i if the edge (O_i, O_j) has a high weight. Otherwise, O_j must be a noise object. Even in a noisy environment, the density of noise will still be relatively lower than that of the coexpressed objects. Therefore, the attraction tree structure will remain stable, and the representatives of coherent patterns will not be perturbed by the presence of noise.



(a) (b)

Fig. 8. The attraction tree for the data set in Fig. 7. (a) $K = 1$ and (b) $K = \infty$.

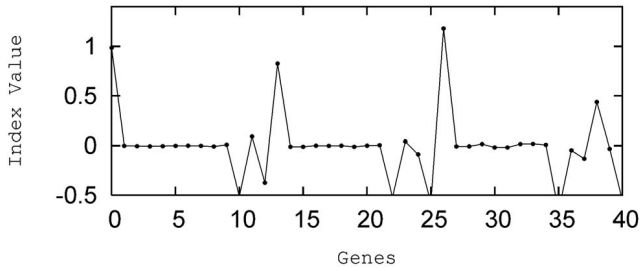


Fig. 9. The coherent pattern index graph for the data set in Fig. 7.

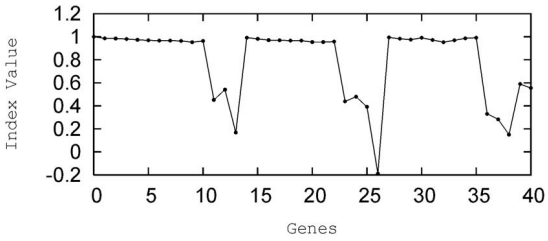


Fig. 11. The similarity curve for Fig. 7.

4 Interactive Exploration of Coherent Patterns

We now describe the application of the concepts defined in Section 3 to the interactive exploration of coherent patterns. Our goal is to plot a *coherent pattern index graph*, where the genes are ordered into an *index list* such that the genes sharing a coherent pattern stay close to each other in the list. Each gene is assigned a coherent pattern index value such that, if there is a consecutive sublist of genes sharing a coherent pattern, the first gene in the sublist has a significantly high index value and the following genes has a low index value. For example, the coherent pattern index graph for the synthetic data set in Fig. 7 is shown in Fig. 9. In the coherent index graph, a sharp pulse strongly indicates the existence of a coherent pattern. Such pluses can guide users in deriving coherent patterns and their corresponding coexpressed genes. Users can recursively examine the selected subsets of coexpressed genes as well as their subpatterns in depth.

4.1 Generating the Index List

Ordering genes into a list allows us to plot the genes and examine the probability of each to be a "leader" in a group of coexpressed genes in a two-dimensional space. An ordered *index list* can be generated based on the following observations:

1. In the attraction tree, the edges connecting a pair of objects O_1 and O_2 conforming to the same coherent pattern P have heavy weights (represented by thick lines in Fig. 8). Genes connected by those edges should remain in close proximity in the list.
2. The edges connecting a pair of intermediate (noise) objects O_1 and O_2 or connecting a pattern correlated object and an intermediate (noise) object have moderate weights (represented by thin lines in Fig. 8). Genes connected by those edges should also remain in proximity in the list but should be more separated than those addressed in case 1.
3. The edges connecting a pair of objects O_1 and O_2 conforming to different coherent patterns P_1 and P_2 have light weights (represented by dashed yellow lines in Fig. 8). Genes connected by those edges should be widely separated in the list.

On the basis of these observations, we have developed the gene-ordering algorithm shown in Fig. 10. This algorithm maintains a list, called $processedVertices$, to record the visiting order of the nodes in the attraction tree T. We start from the root of T. All the edges connecting the root with its children are put into a heap, where the edges are sorted in descending order of weight. We then iteratively extract the edge with the highest weight from the heap. At this point, the start vertex of the edge must have been processed since, otherwise, the edge could not have been put into the heap. We put the end vertex of the edge $currentVertex$ into the list $processedVertices$ and put all the edges connecting $currentVertex$ and its children into the $edgeHeap$. The loop continues until all of the edges in the tree have been visited. The resulting $processedVertices$ is the *index list* of the genes.

Fig. 11 shows the objects in the index list derived from the attraction tree in Fig. 8. For each object, the similarity value plotted in the figure is the similarity between the object and its parent in the attraction tree. The similarity

```
Proc ordering(AttractionTree root)
  processedVertices.add(root)
  for each child ch of root do edgeHeap.insert(edge(root, ch))
  while ( !edgeHeap.isEmpty() ) do
    currentEdge = edgeHeap.extract();   currentVertex = currentEdge.endVertex
    processedVertices.add(currentVertex)
    for each child ch of currentVertex do edgeheap.insert(edge(currentVertex, ch))
  end while
end Proc
```

Fig. 10. The gene-ordering algorithm.

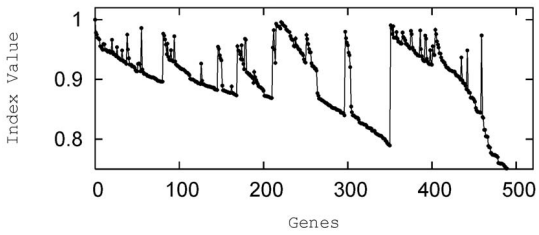


Fig. 12. The similarity curve for Iyer's data set.

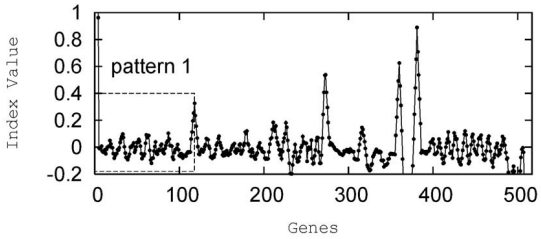


Fig. 13. The coherent pattern index graph for Iyer's data set.

curve can be divided into three-level terraces separated by two valleys. Each valley corresponds to the edge connecting different attraction subtrees. Since such edges have significantly lower weights than other edges, our search strategy does not allow the nodes in subtree T_2 to be visited before all the nodes in subtree T_1 have been visited. Similarly, the visit to subtree T_3 cannot start until the visit to subtree T_2 is finished.

While the similarity curve is informative, it is not always effective, especially with large data sets. For example, the similarity curve shown in Fig. 12 is messy. This is because the similarity curve cannot distinguish coexpressed genes from a chance pair of similar intermediate genes. To solve this problem, we design the coherent pattern index graph in the next section, where the beginning of a potential coherent pattern will be indicated clearly.

4.2 The Coherent Pattern Index and Its Graph

As previously noted, in the construction of an attraction tree and index list, coexpressed genes are located in subtrees and, thus, are arranged as neighbors in the index list. This structure becomes the genesis of the coherent pattern index. In an index list, we may observe a subsequence S of consecutive genes which are more coherent to their parents in the attraction tree than the genes preceding subsequence S. This configuration strongly suggests that S is the starting segment of a group of coexpressed genes.

The above observation leads us to focus on the recognition of *probes*, short subsequences of genes which appear at the beginning of a group of coexpressed genes. In a similarity curve, the similarity between a gene and its parent is plotted. For a gene g_i in an index list $g_1 \cdots g_n$, let $Sim(g_i)$ be g_i's similarity value in the similarity curve. $Sim(g_i) = 0$ if $(i < 1)$ or $(i > n)$. Let p be the minimum size of the probe. For each gene g_i in the index list $g_1 \cdots g_n$, we define the coherent pattern index $CPI(g_i)$ as follows:

$$CPI(g_i) = \sum_{j=1}^{p} Sim(g_{i+j}) - \sum_{j=0}^{p-1} Sim(g_{i-j}). \qquad (5)$$

Intuitively, a high coherent pattern index value indicates a strong potential that a given gene is the start of a group of coexpressed genes. The graph plotting the coherent pattern index values with respect to the index list is called the *coherent pattern index graph*. The valleys in the similarity curve correspond to the sharp pulses in the coherent pattern index graph. In particular, from the above definition, the first $(p-1)$ genes in the index list always generate the first sharp pulse. Fig. 13 is the coherent pattern index graph derived from Fig. 12 with $p = 5$. The coherent pattern index graph clearly indicates the existence of coherent patterns.

4.3 Drilling Down to Subgroups

Fig. 13 clearly indicates that there are five major coherent patterns in the data set. However, *can we further investigate the groups of coexpressed genes conforming to the coherent patterns and identify subgroups of coexpressed genes that conform to any subpatterns?*

Suppose a user accepts the five major coherent patterns reported by the system and clicks on the corresponding peaks in the coherent pattern index graph. The system will split the attraction tree T for the entire data set into five exclusive attraction subtrees. Each subtree corresponds to one coherent pattern, and the genes conforming to that coherent pattern are gathered in that subtree. The original data set is thus partitioned into five subsets.

The user may now select the first subset of genes $\mathcal{D}_1$ (indicated by the dashed box in Fig. 13) in order to move to a finer level of analysis. Fig. 14 shows the local coherent pattern index graph for the selected subset of genes. It should be noted that Fig. 14 is not simply a higher-resolution extract from Fig. 13. Rather, $\mathcal{D}_1$ is assembled from the attraction tree such that genes conforming to the coherent pattern are selected. The attraction tree, index list, and coherent pattern index graph are then generated, with only the genes in the selected subset considered. The user can specify local parameters (e.g., σ) for computing the influence and density in the subset of genes.

According to the influence function (2), a smaller σ will boost the relative influence of a gene on its neighborhood. A detailed discussion of the effect of σ on the influence calculation can be found in [16]. We use the standard deviation of the pairwise distance between genes to determine the value of σ. When the data set is split into smaller subsets, the standard deviation will decrease.

With the help of index graphs, users can recursively explore the coherent patterns until satisfying results are

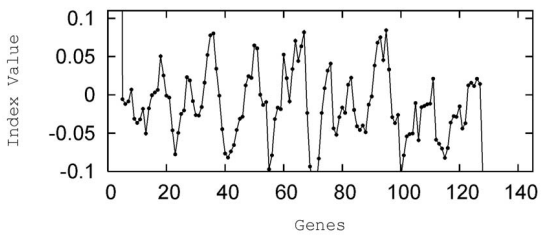


Fig. 14. The coherent pattern index graph for a subset of genes in Iyer's data set.

achieved. The coherent patterns and the corresponding coexpressed gene groups form a hierarchical tree T, where each node on T at level k contains a subset of genes G_i^k conforming to the same coherent pattern P_i^k. We select the expression profile of the root object of the attraction subtree with respect to G_i^k as the representative of the coherent pattern P_i^k.

As the final step of our approach, the following method is adopted to report groups of coexpressed genes from the identified coherent patterns in T. First, for each coherent expression pattern P_j, all the genes in the data set are ordered according to their similarity to P_j. Those genes with a similarity value greater than $\bar{S}$, the estimated average similarity between genes within the same cluster, are assigned to cluster G_j. Second, P_j is adjusted to the centroid of the genes in G_j. The above two steps repeat until the assignments of genes do not change. Finally, P_j and G_j are returned as a coherent expression pattern and the corresponding group of coexpressed genes. In particular, genes similar to more than one coherent pattern will participate in multiple clusters, while genes not similar to any coherent pattern are considered as noise and filtered out.

5 Performance Evaluation

We tested our approach with both synthetic data sets and real-world gene expression data sets. The prototype system, *GPX* (for *Gene Pattern eXplorer*), was implemented in Java. The experiments were conducted on a Sun workstation with a 440MHz CPU and 256 MB main memory.

In this section, we report the experimental results. In particular, we will compare the coherent patterns detected by our system with the results produced by six previous algorithms: two classical partition-based approaches, *K-means* and *SOM*, two graph-based approaches, *CAST* (Cluster Affinity Search Technique) [4] and *CLICK* (Cluster Identification via Connectivity Kernels) [31], a clustering algorithm newly developed for gene expression data, *ADAPT* (Adaptive quality-based clustering) [32], and a hierarchical approach *SOTA* (Self Organizing Tree Algorithm) [14].

We implemented the *K-means* and *SOM* algorithms and set the number of clusters as equal to the number of coherent patterns in the ground truth. *CAST* was implemented according to the algorithm described in [4]. The program was run with a wide range of settings for parameter t (the *affinity threshold*) and the result best matches the ground truth was selected. *CLICK* was downloaded from http://www.cs.tau.ac.il/rshamir/expander/expander.html. We accepted the default parameter setting in the software. *Adapt* has a Web interface at http://www.esat.kuleuven.ac.be/thijs/Work/Clustering.html. We set the minimum number of genes in a cluster as five and accepted the default value of 0.95 as the minimum probability of gene belonging to a given cluster. *SOTA* has a Web interface at http://gepas.bioinfo.cnio.es/cgi-bin/sotarray. We accepted the default parameter settings suggested by the Web site.

5.1 The Data Sets

The algorithms listed above were applied to both synthetic data sets and to two real-world gene expression data sets, Iyer's data set [17] and Spellman's data set [33].

Iyer et al. [17] monitored the expression levels of 8,600 distinct human genes during a 12-point time-series of serum stimulation. Those genes whose expression levels significantly changed during the time-series were selected for cluster analysis. Only 517 genes survived this significance test; other genes were filtered out. In other words, Iyer's data set contains 517 data objects with 12 attributes. In [17], the authors gave a list of 10 coexpressed gene groups and the corresponding coherent patterns. We adopted this as the ground truth for our experiments.

Spellman et al. [33] reported the genome-wide 6,220 mRNA transcript levels during the cell cycle of the budding yeast *S. cerevisiae* synchronized by three independent methods. From these data sets, we have selected the $cdc15$ time-series since it contains the largest number of cell cycles and the most coherent expression patterns. From the 6,200 genes monitored, 800 were found to be cell-cycle-dependent. The expression levels of those 800 peak at one of the following five phases: the early M/G_1 phase, the G_1 phase, the S phase, the G_2 phase, or the M phase. All of the cell-cycle correlated genes, together with their peaking phases, are listed at http://genome-www.stanford.edu/cellcycle/. From this set, we filtered out those genes which miss more than one-third of the measured expression values. The remaining 747 genes naturally form five coexpressed gene groups and conform to five coherent patterns. We used this data set to test the capabilities of our approach and other algorithms to detect these five cell-cycle correlated patterns.

We also generated synthetic data sets to test the effectiveness and efficiency of our algorithm. Two parameters were used to describe a cluster C_i; the minimum similarity δ_i between two data objects in the same cluster and the number N_i of data objects in the cluster. The constraints $\delta_i \geq 0.6$ and $N_i \geq 5$ were applied to generate clusters. Prior to the generation of a synthetic data set, users were asked to input N_c, the total number of clusters in the data set, and N_{noise}, the number of noise objects in the data set. Then, for each cluster C_i, the data generator randomly picks up a pair of valid δ_i and N_i and generates the objects in C_i. Finally, the data generator adds N_{noise} noise objects to the data set.

5.2 Effect of Parameters

The algorithms we have developed use two parameters. The first is K, the number of nearest neighbors used to construct the attraction tree. If a data object O and its attractor O_A belong to the same cluster in the ground truth, then the edge (O_A, O) on the attraction tree "correctly" discovers the coherence relationship between O and O_A. To examine the influence of K on the resulting attraction tree, we calculate the *correctness* as the percentage of "correctly" discovered edges on the attraction trees for the two real data sets, as shown in Fig. 15a. We can see that the *correctness* of the attraction trees is insensitive to the number of candidate attractors.

In fact, the selection of K's value depends on each user's requirements for the granularity of co-expression. As we

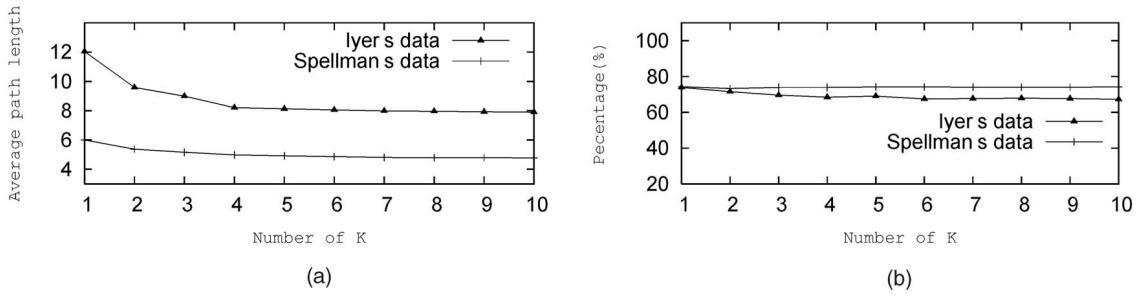


Fig. 15. The effect of the number of nearest neighbors K on attraction trees. (a) Correctness of attraction trees. (b) Average length of paths on attraction trees.

discussed in Section 3.3, a small K favors a detailed local structure and, on average, it takes more levels for a data object to be attracted to the root of the attraction tree. By contrast, a large K value "shortcuts" the attractor paths from a data object to the root and tends to delineate rough patterns. Fig. 15b illustrates the average length of attractor paths from each data object to the root of the attraction tree with respect to different Ks. For Iyer's data, users prefer clusters with small granularity; therefore, we set $K = 1$. For other data sets, we set $K = 10$ as the default value. Note that the "correctness" of the attraction tree is insensitive to K, so the mining results are robust with respect to different settings of K.

Another parameter in our algorithm is the minimum size of probe p, which is used to derive the pattern index graph from the object ordering. Fig. 16 illustrates the effect of p. We can see that a small value of p, such as 3, results in a detailed pattern index graph. A relatively large value of p, such as 15, summarizes the "rough" coherent patterns in the data set and hides the local details. In other words, different settings of p adjust only the "resolution" of the index graph, while the major pulses remain unchanged. In practice, users can set the value of p to be the minimum size of the clusters. Hereafter, we will set $p = 5$.

5.3 Comparing Pattern Index Graph with OPTICS

Fig. 17a shows the reachability-plots generated by *OPTICS* for Iyer's and Spellman's data sets. *OPTICS* has two parameters, the neighborhood radius ε and the minimum number of data objects $MinPts$. We tried a wide range of parameter values and selected those that best accorded with the ground truth. If we use a tighter setting, i.e., a smaller ε and/or a greater $MinPts$, many data objects will be considered as noise by *OPTICS*. A looser setting with an increased ε and/or a decreased $MinPts$ has little impact on the plots.

For Iyer's and Spellman's data sets, we can hardly tell the cluster structure from the corresponding reachability plots. We further examined the object ordering generated by *OPTICS* by comparing it with the ground truth. We found that each dent in the reachability plots corresponds to a small subset of some "real" cluster. Due to the high connectivity of the gene expression data, *OPTICS* often enters another cluster while the visiting of the current cluster has not yet been exhausted. Many data objects belonging to different clusters are thus mixed in the object ordering.

Fig. 17b illustrates the pattern index graphs generated by our method from the two real-world gene expression data sets. For both data sets, the index graphs show strong pulses. Each pulse corresponds to a major pattern in the

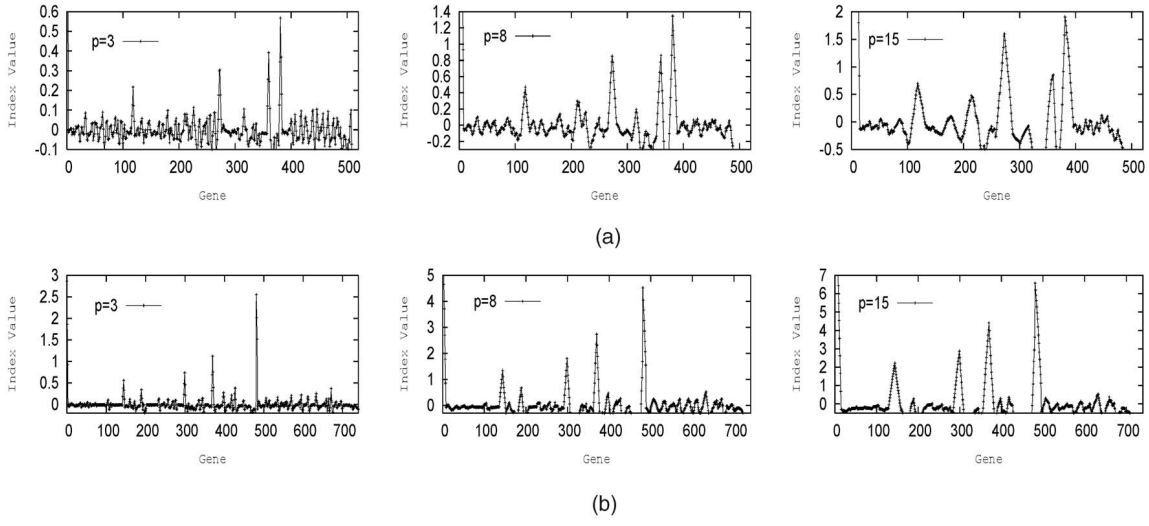


Fig. 16. The effect of minimum size of probe p on pattern index graphs. (a) Iyer's data. (b) Spellman's data.

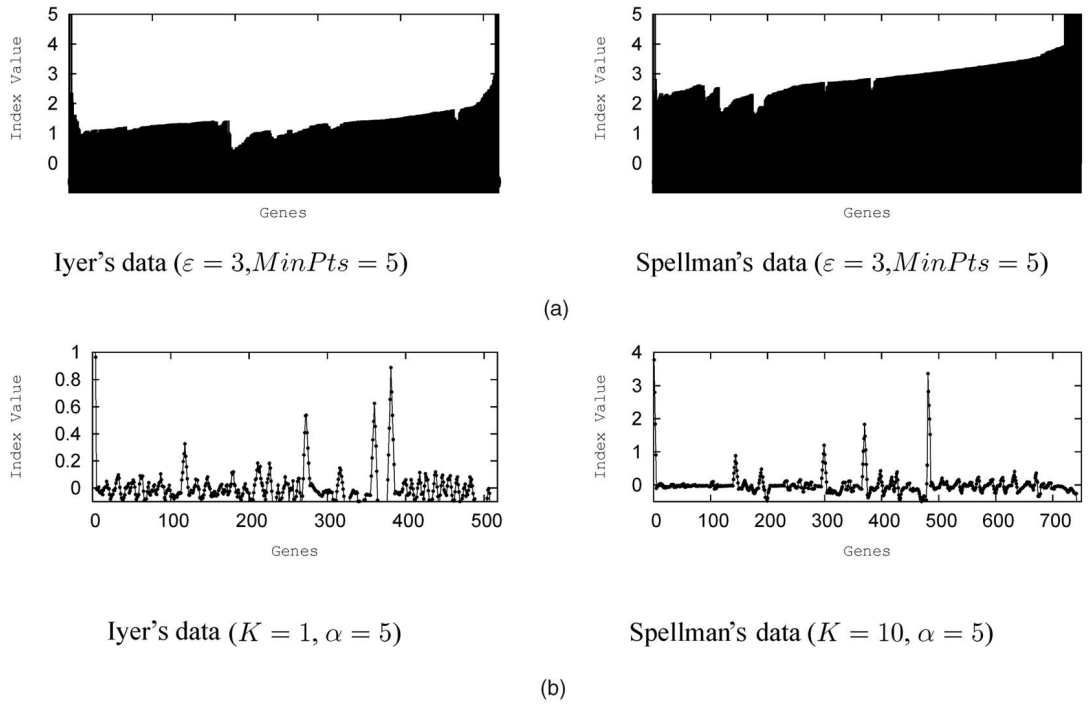


Fig. 17. Comparison of pattern index graph with *OPTICS*. (a) Object ordering generated by *OPTICS*. (b) Object ordering generated by pattern index graph.

ground truth, and the objects between two neighboring pulses tend to conform to a common pattern.

5.4 Comparison with Other Algorithms

We compared the coherent patterns identified by our approach with the ground truth and with the results produced by other methods. The ground-truth patterns provided in [17], [33] were used as the domain knowledge to guide the growth of the hierarchical tree of coexpressed genes and coherent patterns. That is, we split the nodes on the tree on the basis of the corresponding pattern index graphs until the tree "matches" the domain knowledge the best. We will introduce how to measure the quality of match in the following paragraph. The coherent patterns were then returned as described in Section 4.3. To make a fair comparison, for those algorithms with default parameter values, such as CLICK, ADAPT, and SOTA, we simply adopted the default values; for those without default parameter values, such as K-means, SOM, and CAST, we tuned the parameters as follows: For K-means and SOM, we set the parameter, i.e., the number of clusters K, as the number of coherent patterns in the ground truth; for CAST, we increased the parameter t with a small step (0.01) within its full range, i.e., from 0 to 1, and ran the algorithm repeatedly. The result which best "matches" the ground truth was picked.

Suppose $\{P_1, \ldots, P_n\}$ is the set of coherent patterns in the ground truth and $\{\tilde{P}_1, \ldots, \tilde{P}_m\}$ is the set of coherent patterns identified by a particular mining method. For each pattern P_i in the ground truth, we identified the pattern $\tilde{P}_j$ in the mining results which most closely resembles P_i, and called $\tilde{P}_j$ the "match" for P_i. Fig. 18 illustrates the hierarchy of coexpressed gene groups in Iyer's data. The pulses selected to split the (sub) data sets are marked in the corresponding coherent pattern index graphs. Fig. 19 lists the similarity between the ground-truth patterns for Iyer's data set and the corresponding "matches" identified by the various tested methods. We say a ground-truth pattern P_i is "accurately identified" if there exists a matching pattern $\tilde{P}_j$ such that the similarity between P_i and $\tilde{P}_j$ is greater than 0.9. The patterns accurately identified by each algorithm are indicated in a bold font. The numbers in parenthesis in the first row indicate the number of coherent patterns *returned* by each method. Not every pattern returned by one of these algorithms necessarily matches a ground-truth pattern.

A comparison of the patterns reported by each approach with the ground truth indicates that:

- Our method and *SOTA* were the only two approaches that accurately identified all 10 ground-truth patterns. However, *SOTA* overestimated the number of coherent patterns; it reported 50 coherent patterns with 80 percent false positives, while our system reported 10 patterns with zero false positives.
- Many of the methods failed to identify ground-truth patterns 5 and 9. These ground-truth patterns are shared by a small number of coexpressed genes and are similar to patterns 3 and 6, respectively. Therefore, most clustering algorithms merged them into patterns 3 and 6, respectively.
- All methods other than our approach and *CLICK* split the genes sharing ground-truth coherent pattern 2 into several smaller subsets. This is because the coherence of the genes in this group is much weaker than that of other clusters.

We then tested the full range of algorithms on Spellman's data set. Again, our system was the only one to accurately identify all the ground-truth patterns with a low false

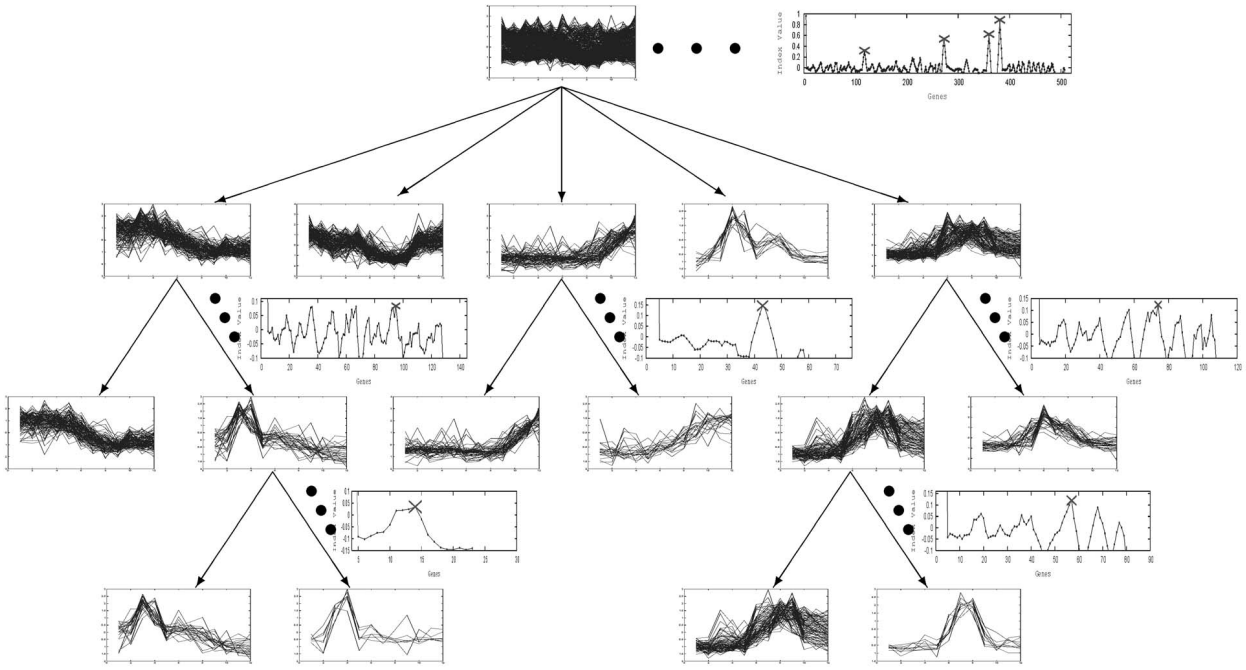


Fig. 18. The hierarchy of coexpressed gene groups in Iyer's data set.

Pattern	GPX (10)	Kmeans (10)	SOM (10)	ADAPT (11)	CLICK (7)	CAST (9)	SOTA (50)
1	**0.998**	**0.973**	**0.983**	**0.956**	0.884	**0.955**	**0.962**
2	**0.996**	**0.950**	**0.992**	**0.911**	**0.991**	0.887	**0.936**
3	**0.993**	**0.910**	0.872	**0.993**	**0.994**	**0.997**	**0.947**
4	**0.995**	**0.996**	**0.989**	**0.984**	0.883	**0.968**	**0.955**
5	**0.964**	0.882	0.716	0.868	0.886	0.855	**0.962**
6	**0.940**	**0.965**	0.764	**0.989**	**0.970**	**0.984**	**0.972**
7	**0.972**	0.880	0.892	**0.976**	**0.990**	0.719	**0.988**
8	**0.995**	**0.963**	**0.917**	**0.997**	**0.914**	**0.999**	**0.958**
9	**0.907**	**0.910**	0.848	0.824	0.844	0.800	**0.940**
10	**0.987**	**0.930**	**0.983**	**0.981**	**0.976**	**0.996**	**0.960**

Fig. 19. Coherent patterns discovered in Iyer's data set by different approaches.

positive rate (see Figs. 20 and 21). Ground-truth patterns 1 and 4 in this data set are difficult to identify. Pattern 1 corresponds to the genes peaking at the early $M/G1$ phase, which is an intermediate phase between the M (pattern 5) and $G1$ (pattern 2) phases. Some tested approaches assigned the genes following pattern 1 to either pattern 2 or pattern 5. Pattern 4 is a "weak" pattern which is conformed to by a small number of genes. Some approaches cannot effectively adapt to different cluster granularities and, thus, fail to identify pattern 4.

As discussed in Section 1, the interpretation of coherent patterns and coexpressed genes depends on the domain knowledge. Users may have different requirements for cluster granularity in different parts of the data set. Our system addresses this challenge by adopting an interactive approach and supporting flexible exploration incorporating the domain knowledge of users.

5.5 Efficiency and Scalability

We tested the efficiency and scalability of our method using synthetic data sets of various sizes. In fact, our algorithm proceeds in two steps: First, in the *preprocessing step*, we normalize the data objects and calculate the pairwise distance between data objects. In the *exploration step*, we construct the attraction tree structure and generate the pattern index graph to support interactive exploration. Figs. 22a and 22b illustrate the computation time for the preprocessing and exploration steps. As indicated there, our algorithm is scalable with respect to the number of genes, and the computation time is dominated by the preprocessing step.

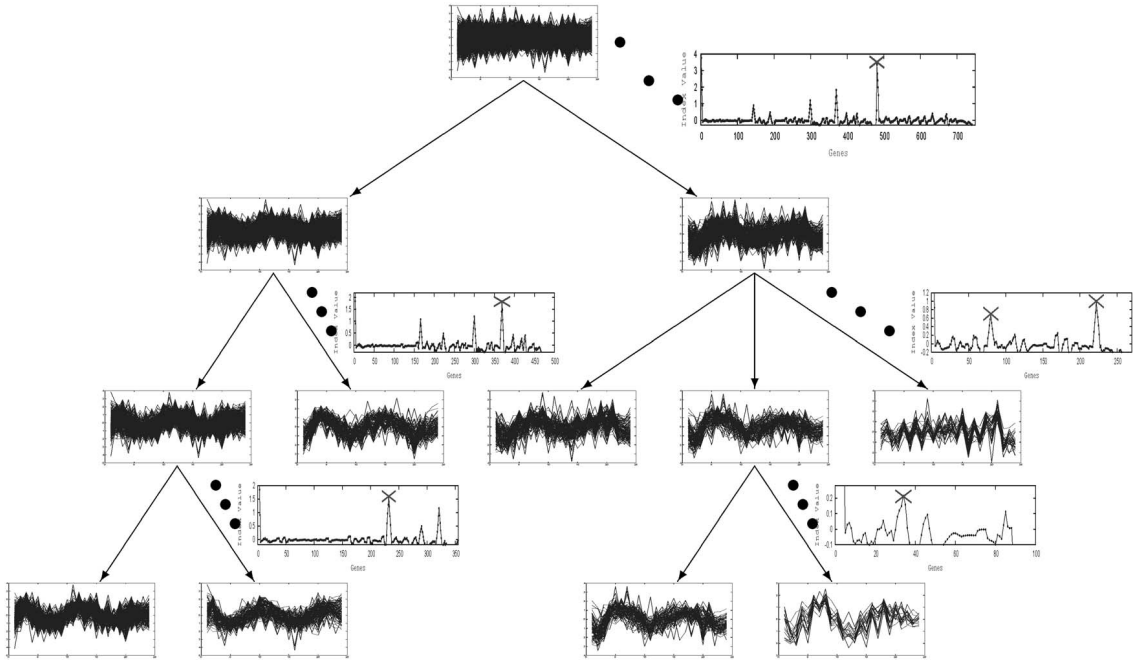


Fig. 20. The hierarchy of coexpressed gene groups in the Spellman's data set.

Pattern	GPX (7)	Kmeans (5)	SOM (5)	ADAPT (21)	CLICK (9)	CAST (19)	SOTA (99)
1	**0.901**	**0.928**	0.194	0.884	0.855	**0.900**	**0.938**
2	**0.970**	**0.976**	**0.972**	**0.972**	**0.978**	**0.970**	**0.968**
3	**0.980**	**0.950**	0.552	**0.953**	**0.970**	0.888	**0.940**
4	**0.901**	0.773	0.437	0.796	**0.984**	0.888	**0.961**
5	**0.945**	**0.965**	**0.964**	**0.962**	**0.978**	**0.956**	**0.956**

Fig. 21. Coherent patterns discovered in Spellman's data set by different approaches.

6 Discussion

6.1 Two Types of Gene Expression Data

Gene expression data come in two types. *Gene-time data* result from the monitoring of gene expression levels in identical samples during a time-series. Alternatively, the expression levels of genes may be collected from a set of samples (e.g., from healthy people and people with cancer) at a designated time point; this produces *gene-sample* data. In our density-based model, we treat the genes as data objects and assume that the attributes, either time-points or samples, are independent. Therefore, our method is applicable to both types of gene expression data. In this paper, we tested the effectiveness of our method only on gene-time data, such as Iyer's and Spellman's data sets, because ground truth information on coherent patterns and coexpressed is not currently available for gene-sample data set. However, our empirical study has shown that the two challenges presented in Section 1 apply to both types of gene expression data. Furthermore, the experimental results using synthetic data have indicated the general effectiveness of our method, especially when the data set

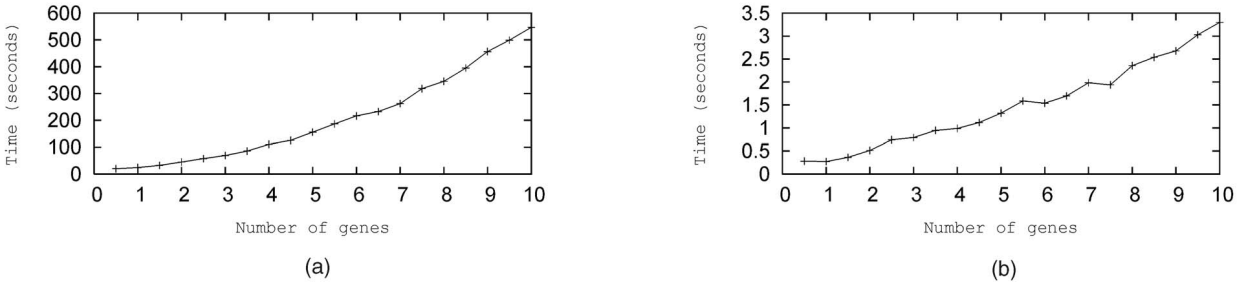


Fig. 22. The scalability of the two steps in our algorithm.

contains a large amount of noise. Therefore, it is likely that this density-based interactive approach will also be effective with gene-sample data.

6.2 Effectiveness versus Efficiency

As discussed in Section 5, the computation time required by our algorithm is dominated by the computation of the pairwise similarity between objects, which is $O(N^2)$. As a result, our method is not as efficient as some clustering algorithms with linear complexity. However, a gene expression data set typically contains several thousand of genes and less than 100 time-points or samples. Despite the rapid development of microarray technology, the natural limit on the total number of an organisms's genes (30,000 to 100,000) places a cap on the practical size of a gene expression data set. Compared with some conventional large-scaled transaction database or multimedia database, the size of a gene expression data set will not be too large. Therefore, biologists are much more concerned with the accuracy of the mining results than the efficiency of the mining process. From Fig. 22b, we can see that once the data set is preprocessed, the exploration step only takes linear time. That is, users can "drill down" and "roll up" the hierarchical tree of coherent patterns with quick feedback until a useful result has been achieved.

6.3 Integration of Domain Knowledge and Multiple Data Sources

Clustering is generally recognized as an "unsupervised" learning problem. However, biologists often have some prior knowledge about a gene expression data set. Additionally, recent technological advances have enabled the collection of various types of data at a genome-wide scale; examples include protein-protein interactions from the *yeast two-hybrid assay* [37] and *mass spectrometry* [11]. The integration of domain knowledge and multiple data sources may achieve more robust results [28], [29].

In this paper, we proposed an interactive approach, which takes the first step toward this integration of knowledge. However, the construction of the attraction tree and the generation of the pattern index graph are based purely on the expression profiles of genes. In the future, we plan to examine methods to further integrate the domain knowledge of users and other types of data sets to enhance the robustness and meaningfulness of the index graph.

7 Conclusion

Identifying coexpressed gene groups and discovering coherent patterns are two important tasks in mining gene expression data. In this paper, we have analyzed the challenges of clustering gene expression data and have proposed an interactive framework to help users explore coherent patterns based on their domain knowledge. Unlike many other clustering approaches, our method does not start by partitioning the data set into clusters of coexpressed genes. Instead, it prompts users with potential coherent patterns. Thus, our approach avoids arbitrary decisions regarding cluster borders and performs well in the environment with a large number of "intermediate" genes. Our empirical study has demonstrated that our method can identify most of the coherent patterns in a data set with higher accuracy than the state-of-the-art methods.

Acknowledgments

The research of Daxin Jiang and Aidong Zhang is partly supported by the US National Science Foundation (NSF) Grants DBI-0234895 and NIH Grant 1 P20 GM067650-01A1. The research of Jian Pei is partly supported by NSF Grant IIS-0308001, an NSERC Discovery Grant, a President's Research Grant, an Endowed Research Fellowship Award, and a startup grant in Simon Fraser University. All opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. A preliminary version of this paper appears as [19].

References

[1] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine, "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Array," *Proc. Nat'l Academy of Sciences USA*, vol. 96, no. 12, pp. 6745-6750, June 1999.

[2] M. Ankerst, M.M. Breunig, H.P. Kriegel, and J. Sander, "OPTICS: Ordering Points to Identify the Clustering Structure," *Proc. SIGMOD*, pp. 49-60, 1999.

[3] Z. Bar-Joseph, E.D. Demaine, D.K. Gifford, N. Srebro, A.M. Hamel, and T.S. Jaakkola, "K-ary Clustering with Optimal Leaf Ordering for Gene Expression Data," *Bioinformatics*, vol. 19, no. 9, pp. 1070-1078, 2003.

[4] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering Gene Expression Patterns," *J. Computational Biology*, vol. 6, nos. 3-4, pp. 281-297, 1999.

[5] M. Blatt, S. Wiseman, and E. Domany, "Super-Paramagnetic Clustering of Data," *Physical Rev. Letters*, vol. 76, 1996.

[6] Y. Cheng and G.M. Church, "Biclustering of Expression Data," *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology (ISMB)*, vol. 8, pp. 93-103, 2000.

[7] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart, and R.W. Davis, "A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle," *Molecular Cell*, vol. 2, no. 1, pp. 65-73, July 1998.

[8] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein, "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc. Nat'l Academy of Sciences USA*, vol. 95, no. 25, pp. 14863-14868, Dec. 1998.

[9] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining*, pp. 226-231, 1996.

[10] C. Fraley and A.E. Raftery, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis," *The Computer J.*, vol. 41, no. 8, pp. 578-588, 1998.

[11] A.C. Gavin et al., "Functional Organization of the Yeast Proteome by Systematic Analysis of Protein Complexes," *Nature*, vol. 415, no. 6868, pp. 123-124, Jan. 2002.

[12] D. Ghosh and A.M. Chinnaiyan, "Mixture Modelling of Gene Expression Data from Microarray Experiments," *Bioinformatics*, vol. 18, pp. 275-286, 2002.

[13] E. Hartuv and R. Shamir, "A Clustering Algorithm Based on Graph Connectivity," *Information Processing Letters*, vol. 76, nos. 4-6, pp. 175-181, 2000.

[14] J. Herrero, A. Valencia, and J. Dopazo, "A Hierarchical Unsupervised Growing Neural Network for Clustering Gene Expression Patterns," *Bioinformatics*, vol. 17, pp. 126-136, 2001.

[15] L.J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring Expression Data: Identification and Analysis of Coexpressed Genes," *Genome Research*, vol. 9, no. 11, pp. 1106-1115, 1999.

[16] A. Hinneburg and D.A. Keim, "An Efficient Approach to Clustering in Large Multimedia Database with Noise," *Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining,* 1998.
[17] V.R. Iyer et al., "The Transcriptional Program in the Response of Human Fibroblasts to Serum," *Science,* vol. 283, pp. 83-87, 1999.
[18] D. Jiang, J. Pei, and A. Zhang, "DHC: A Density-Based Hierarchical Clustering Method for Time Series Gene Expression Data," *Proc. Third IEEE Symp. Bio-Informatics and Bio-Engineering (BIBE '03),* 2003.
[19] D. Jiang, J. Pei, and A. Zhang, "Interactive Exploration of Coherent Patterns in Time-Series Gene Expression Data," *Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '03),* 2003.
[20] T. Kohonen, *Self-Organization and Associative Memory.* Berlin: Spring-Verlag, 1984.
[21] J. Liu and W. Wang, "OP-Cluster: Clustering by Tendency in High Dimensional Space," *Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03),* 2003.
[22] J.B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability,* pp. 281-297, Univ. of California, Berkeley, Univ. of California Press, Berkeley, 1967.
[23] G.J. McLachlan, R.W. Bean, and D. Peel, "A Mixture Model-Based Approach to the Clustering of Microarray Expression Data," *Bioinformatics,* vol. 18, pp. 413-422, 2002.
[24] J. Pei, X. Zhang, M. Cho, H. Wang, and P.S. Yu, "MaPle: A Fast Algorithm for Maximal Pattern-Based Clustering," *Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03),* 2003.
[25] P.A. Ralf-Herwig, C. Muller, C. Bull, H. Lehrach, and J. O'Brien, "Large-Scale Clustering of cDNA-Fingerprinting Data," *Genome Research,* vol. 9, pp. 1093-1105, 1999.
[26] M.F. Ramoni, P. Sebastiani, and I.S. Kohane, "Cluster Analysis of Gene Expression Dynamics," *Proc. Nat'l Academy of Science,* vol. 99, no. 14, pp. 9121-9126, July 2002.
[27] R. Šášik, T. Hwa, N. Iranfar, and W.F. Loomis, "Percolation Clustering: A Novel Algorithm Applied to the Clustering of Gene Expression Patterns in Dictyostelium Development," *Proc. Pacific Symp. Biocomputing,* pp. 335-347, 2001.
[28] E. Segal, H. Wang, and D. Koller, "Discovering Molecular Pathways from Protein Interaction and Gene Expression Data," *Bioinformatics,* vol. 19, pp. i264-i272, 2003.
[29] E. Segal, R. Yelensky, and D. Koller, "Genome-Wide Discovery of Transcriptional Modules from DNA Sequence and Gene Expression," *Bioinformatics,* vol. 19, pp. i273-i282, 2003.
[30] J. Seo and B. Shneiderman, "Interactively Exploring Hierarchical Clustering Results," *Computer,* vol. 35, no. 7, pp. 80-86, July 2002.
[31] R. Shamir and R. Sharan, "Click: A Clustering Algorithm for Gene Expression Analysis," *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology (ISMB '00),* 2000.
[32] F.D. Smet, J. Mathys, K. Marchal, G. Thijs, B.D. Moor, and Y. Moreau, "Adaptive Quality-Based Clustering of Gene Expression Profiles," *Bioinformatics,* vol. 18, pp. 735-746, 2002.
[33] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Bostein, and B. Futcher, "Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization," *Molecular Biology of the Cell,* vol. 9, pp. 3272-3297, 1998.
[34] P. Tamayo, D. Solni, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, and T.R. Golub, "Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation," *Proc. Nat'l Academy of Sciences USA,* vol. 96, no. 6, pp. 2907-2912, Mar. 1999.
[35] S. Tavazoie, D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church, "Systematic Determination of Genetic Network Architecture," *Nature Genetics,* pp. 281-285, 1999.
[36] S. Tomida, T. Hanai, H. Honda, and T. Kobayashi, "Analysis of Expression Profile Using Fuzzy Adaptive Resonance Theory," *Bioinformatics,* vol. 18, pp. 1073-1083, 2002.
[37] P. Uetz et al., "A Comprehensive Analysis of Protein-Protein Interactions in Saccharomyces Cerevisiae," *Nature,* vol. 403, no. 6770, pp. 601-603, Feb. 2000.
[38] H. Wang, W. Wang, J. Yang, and P.S. Yu, "Clustering by Pattern Similarity in Large Data Sets," *SIGMOD 2002, Proc. ACM SIGMOD Int'l Conf. Management of Data,* pp. 394-405, 2002.
[39] Y. Xu, V. Olman, and D. Xu, "Clustering Gene Expression Data Using a Graph-Theoretic Approach: An Application of Minimum Spanning Trees," *Bioinformatics,* vol. 18, pp. 536-545, 2002.
[40] J. Yang, W. Wang, H. Wang, and P.S. Yu, "δ-Cluster: Capturing Subspace Correlation in a Large Data Set," *Proc. 18th Int'l Conf. Data Eng. (ICDE 2002),* pp. 517-528, 2002.
[41] K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, and W.L. Ruzzo, "Model-Based Clustering and Data Transformations for Gene Expression Data," *Bioinformatics,* vol. 17, pp. 977-987, 2001.
[42] D. Jiang, J. Pei, and A. Zhang, "Mining Coherent Gene Clusters from Gene-Sample-Time Microarray Data," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '04),* 2004.

Daxin Jiang received the BS degree in computer science from the University of Science and Technology of China. He was an MS student at the Software Institute, Chinese Academy of Sciences, from 1998-2000. He is currently a PhD candidate in the Department of Computer Science and Engineering, State University of New York at Buffalo. His research interests include bioinformatics, data mining, machine learning, and information retrieval.

Jian Pei received the PhD degree in computing science from Simon Fraser University, Canada, in 2002. He is currently an assistant professor of computing science at Simon Fraser University, Canada. From 2002 to 2004, he was an assistant professor of computer science and engineering at the State University of New York at Buffalo. His research interests can be summarized as developing advanced data analysis techniques for emerging applications. Particularly, he is currently interested in various techniques of data mining, data warehousing, online analytical processing, and database systems, as well as their applications in bioinformatics. His current research is supported in part by the US National Science Foundation (NSF) and the Natural Sciences and Engineering Research Council (NSERC) of Canada. He has published more than 50 research papers in refereed journals, conferences, and workshops, has served on the program committees of more than 40 international conferences and workshops, and has been a reviewer for some leading academic journals. His research papers have received hundreds of citations. He is a member of the ACM, the ACM SIGMOD, the ACM SIGKDD, and the IEEE Computer Society.

Aidong Zhang received the PhD degree in computer science from Purdue University, West Lafayette, Indiana, in 1994. She was an assistant professor from 1994 to 1999, an associate professor from 1999 to 2002, and has been a professor since 2002 in the Department of Computer Science and Engineering at the State University of New York at Buffalo. Her research interests include databases, multimedia systems, content-based image retrieval, bioinformatics, and data mining. She is an author of more than 150 research publications in these areas. Dr. Zhang's research has been funded by the US National Science Foundation (NSF), NIMA, and NIH. Dr. Zhang serves on the editorial boards of the *International Journal of Bioinformatics Research and Applications*, *ACM Multimedia Systems*, the *International Journal of Multimedia Tools and Applications*, the *International Journal of Distributed and Parallel Databases*, and the *ACM SIGMOD DiSC* (Digital Symposium Collection). She was cochair of the technical program committee for ACM Multimedia 2001. She has also served on various conference program committees. Dr. Zhang is a recipient of the US NSF CAREER award and SUNY Chancellor's Research Recognition award. She is a member of the IEEE.