Interactive Exploration of Coherent Patterns in Time-series Gene Expression Data

Daxin Jiang Jian Pei Aidong Zhang
State University of New York at Buffalo
{djiang3,jianpei,azhang}@cse.buffalo.edu

ABSTRACT

Discovering coherent gene expression patterns in time-series gene expression data is an important task in bioinformatics research and biomedical applications. In this paper, we propose an *interactive exploration* framework for mining coherent expression patterns in time-series gene expression data. We develop a novel tool, *coherent pattern index graph*, to give users highly confident indications of the existences of coherent patterns. To derive a coherent pattern index graph, we devise an *attraction tree* structure to record the genes in the data set and summarize the information needed for the interactive exploration. We present fast and scalable algorithms to construct attraction trees and coherent pattern index graphs from gene expression data sets. We conduct an extensive performance study on some real data sets to verify our design. The experimental results strongly show that our approach is more effective than the state-of-the-art methods in mining real gene expression data, and is scalable in mining large data sets.

Categories and Subject Descriptors

I.5.3 [**Computing Methodologies**]: Pattern Recognition—*Clustering*

General Terms

Algorithms, Experimentation

Keywords

coherent patterns, gene expression data, bioinformatics

1. INTRODUCTION

DNA microarray technology enables simultaneously monitoring the expression levels for thousands of genes during important biological processes and across collections of related samples. An important task of analyzing the DNA microarray gene expression data is to find the co-expressed genes and the coherent gene expression patterns. A group of *co-expressed genes* are the ones with similar expression patterns, while a *coherent gene expression pattern* (or coherent pattern in short) characterizes the common trend of expression levels for a group of co-expressed genes. In other words, a coherent gene expression pattern is a "template", while the expression profiles of the corresponding co-expressed genes yield to the pattern with small divergence.

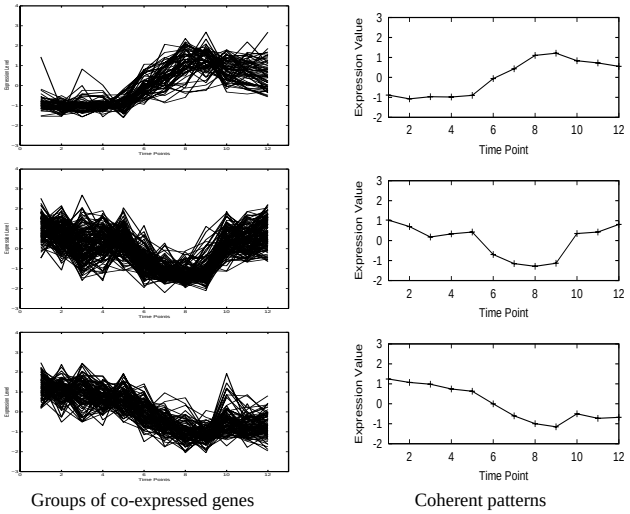

Figure 1: Examples of groups of co-expressed genes and corresponding coherent expression patterns

For example, the Iyer's data set [7] records the expression profiles of 517 human genes with respect to a 12-point time-series. In [4], Eisen et al. give a list of 10 groups of co-expressed genes and the corresponding coherent gene expression patterns in the Iyer's data set, which is well accepted as the *ground truth*.

In Figure 1, we demonstrate three groups of co-expressed genes in the ground truth and the corresponding coherent patterns. The left column elaborates three groups of co-expressed genes. It can be seen clearly from the figure that the co-expressed genes share the common trends in their expression profiles. The right column illustrates the coherent expression patterns corresponding to the groups of co-expressed genes.

Why do we want to find co-expressed genes and coherent gene expression patterns? In practice, co-expressed genes may have the same cellular functions and may be regulated by the same mechanism. Coherent gene expression patterns may suggest important cellular processes and characterize the regulating mechanism in the cells.

To find co-expressed genes and discover coherent expression patterns, various clustering algorithms have been developed, including some conventional methods, such as K-means [11], SOM (for Self Organizing Map) [10] and the hierarchical approaches [4, 1], as well as some newly proposed ones targeting at gene expression data, such as CAST [2], CLICK [9] and Adapt [5]. Generally, those

SIGKDD '03, August 24-27, 2003, Washington, DC, USA

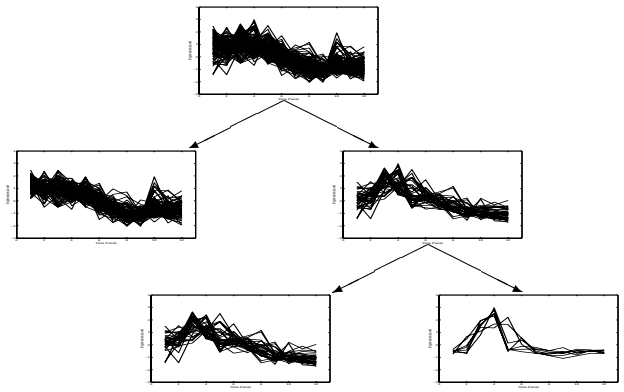

Figure 2: The hierarchy of a co-expressed gene group

clustering algorithms partition the set of genes into clusters. Each cluster is considered as a group of co-expressed genes and the coherent expression pattern can be simply the mean (the centroid) of the expression profiles of the genes in that cluster. While previous studies have confirmed that the clustering algorithms are useful to identify groups of co-expressed genes and discover coherent expression patterns, due to the distinct characteristics of time-series gene expression data and the special requirements from the biology domain, clustering gene expression data is still facing the following two unsettled challenges.

Challenge 1: It is subtle to unfold the hierarchies of co-expressed genes and coherent patterns

In a DNA microarray gene expression data set, there are usually multiple groups of co-expressed genes and the corresponding coherent patterns. One important and interesting feature is that *there is usually a hierarchy of co-expressed genes and coherent patterns in a typical gene expression data set*.

For example, Figure 2 shows a group of co-expressed genes in the Iyer's data set, which can be split into two subgroups, and one subgroup can be further split into two sub-subgroups. Genes in each subgroup have more uniform expression profiles, i.e., the pattern is more coherent. Therefore, the groups of co-expressed genes form a hierarchy. At the higher levels of the hierarchy, we can see large groups of genes approximately following some "rough" coherent expression patterns. At the lower levels of the hierarchy, the large groups of genes break into small subgroups. Those small subgroups of co-expressed genes follow some "finely" coherent expression patterns, which inherit some characteristics from the "rough" patterns, and add some distinct characteristics.

The subtlety here is that *there is no general and objective standard to identify co-expressed gene groups*. The interpretation of co-expressed genes and coherent patterns mainly depends on the domain knowledge. In some cases, a small difference in expression levels by a small group of genes at certain time instants may be of particular interest. In some other cases, the difference is considered insignificant at all based on some background knowledge. For example, in Figure 2, the left child of the root in the hierarchy tree contains more genes than its sibling. It can be further divided into subgroups by a brute force try. However, the ground truth (i.e., the partitioning theme and patterns justified by biologists) indicates that it should not be further split. In other words, in time-series gene expression data, the size and the granularity of groups of co-expressed genes as well as the corresponding coherent patterns may vary substantially. One challenge raises: *Can we provide a flexible tool for biologists so that they can interactively unfold the hierarchy of groups of co-expressed genes and derive the corresponding coherent patterns?* A user may want to explore the structure of the

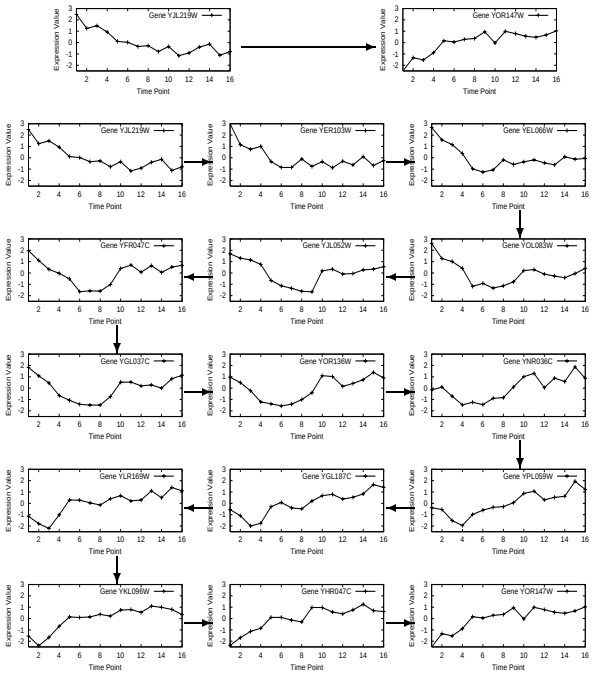

Figure 3: The gradual change from one expression profile to a complete different profile

data set and study the different parts of the data set with different criteria according to their domain background knowledge, which is hard to describe or to be integrated into the mining system.

Challenge 2: It is hard to handle the high connectivity of genes in the time-series gene expression data sets.

An interesting phenomenon in time-series gene expression data sets is that *groups of co-expressed genes may be highly connected by a large amount of "intermediate" genes*. For example, the two genes in the first row of Figure 3, taken from the real $CDC28$ data set, have very different expression profiles. However, we can find a series of intermediate genes between the two genes in the same data set, as shown in the figure, such that each two consecutive genes are quite similar to each other. These intermediate genes build a "bridge" between the two very different ones. Such "bridges" are common in the gene expression data sets.

The high connectivity in the gene expression data raises a challenge: *It is often hard to find the (clear) borders among the clusters*. Usually, the number of intermediate genes is much larger than the total number of co-expressed genes. Our empirical study indicates that, in many cases, the density of genes smoothly decreases from the core area to the border area. However, some border areas may also have high density. That is because the density of the cores of different clusters can be very different. For example, the density of cluster A's core area may be much higher than that of cluster B's core area. Therefore, the density of a border area of cluster A may be even higher than that of the core area of cluster B.

Many existing clustering methods may fall into one of the following hard situations when they mine the gene expression data: On the one hand, the data set is decomposed into numerous small clusters. Some clusters consist of groups of co-expressed genes, while most of the clusters consist of the intermediate genes. Since there is no absolute standard (e.g., size, compactness) to rank the resulted clusters, it may take a user a huge effort to examine which clusters are really the groups of co-expressed genes. On the other hand, the algorithm results in several large clusters. Each cluster

contains both the co-expressed genes and a large amount of intermediate genes. However, those intermediate genes may mislead the centroids of the clusters going astray. The centroids then no longer represent the true coherent patterns in the group of co-expressed genes.

In this paper, we study the problem of *mining coherent patterns from time-series gene expression data*, and make the following contributions. First, one important observation about the existing clustering methods is that almost all of them try to find the clusters (i.e., groups of co-expressed genes) first according to some global criteria, and then derive the coherent patterns based on the clusters. In this study, we propose a novel strategy of *interactive exploration of gene coherent patterns*: we interactively explore the hierarchy of coherent expression patterns and find the groups of co-expressed genes according to the coherent patterns. The users' background knowledge can be integrated by the interaction between the users and the system. Second, to implement our new strategy, we develop a novel interactive exploration tool, *coherent pattern index graph*, to give users highly confident indications of the existence of coherent patterns. Third, to derive a coherent pattern index graph, we need to extract the information about the relations between genes and their groups. We devise an *attraction tree* structure to record the genes in the data set which summarizes the information needed for the interactive exploration. Fourth, we present fast and scalable algorithms to construct the attraction tree and the coherent pattern index graph from the gene expression data set. Last, we conduct an extensive performance study on real data sets to verify our design. The experimental results strongly show that our approach is more effective than the state-of-the-art methods in mining real gene expression data, and is scalable in mining large data sets.

The remainder of the paper is organized as follows. In Section 2, we introduce the attraction tree structure and its construction algorithm. The interactive exploration of coherent patterns using coherent pattern index graph is presented in Section 3. An extensive performance study is reported in Section 4. The paper is concluded in Section 5.

2. THE ATTRACTION TREE

To enable the interactive exploration of the coherent gene expression patterns, it is important to extract and organize the information about the relation among genes and their groups. In this section we introduce a density-based method to construct an *attraction tree* structure.

We first describe how to measure the distance between two genes and how to define the density of genes. Then, we present the attraction tree structure.

One of the most commonly used distance measure is the *Euclidean distance*. However, as indicated in [12], for gene expression data, people are more interested in the overall shapes of expression profiles instead of the absolute magnitudes. The Euclidean distance does not work well for scaling and shifting profiles. Hence, the *Pearson's correlation coefficient* is often used to measure the similarity between two expression patterns [12].

To define the density of a gene, we have to tell the distance between two arbitrary genes. Thus, we need to transform the Pearson's correlation, a similarity measure, into a distance measure. Given an object[1] O, we transform the object to O' as follows. For each attribute d, let $O'_d = \frac{O_d - \eta_O}{\sigma_O}$, where η_O and σ_O are the mean and the standard deviation of all the attributes of O, respectively. Then, the similarity and distance between the data objects are defined as

$$similarity(O_i, O_j) = pearson(O'_i, O'_j), \quad (1)$$

and

$$distance(O_i, O_j) = euclidean(O'_i, O'_j). \quad (2)$$

The density of a data object O reflects the distribution of the other objects in O's neighborhood. A recently proposed method, Denclue [6], defines an *influence function* to describe the influence between two objects. For example, the *Gaussian influence function*, is defined as follows.

$$f(O_i, O_j) = e^{-\frac{d(O_i, O_j)^2}{2\sigma^2}} \quad (3)$$

where $d(O_i, O_j)$ is the distance between objects O_i and O_j and σ is a parameter. Given a data set $\mathcal{D}$, the density of an object O is the sum of influences from all the objects in the data set except for itself. That is,

$$density(O) = \sum_{O_j \in \mathcal{D}, O_j \neq O} f(O, O_j). \quad (4)$$

In this paper, we will follow the density definition by Denclue. We will address how to determine an appropriate value for parameter σ in Section 3.3.

An *attraction tree* structure records the information about genes that will be used later in the coherent pattern mining and clustering of co-expressed genes. Intuitively, to organize the groups of genes and their coherent patterns, we can treat the genes in the way that a gene with high density in a data set "*attracts*" the other genes with less density.

The attraction between two data objects O_i and O_j ($O_i \neq O_j$) is defined by the influence function (Equation 3). The attraction is said *from O_i to O_j* if $density(O_i) < density(O_j)$, denoted as $O_i \rightarrow O_j$. In the case that two objects are tie in density, we can artificially assign $O_i \rightarrow O_j$ for $(i < j)$. Thus, an object O is attracted by a set of objects $A(O)$ whose density are larger than that of O, where $A(O) = \{O_j | density(O_j) > density(O)\}$. We define the *attractor* of O as the object $O_j \in A(O)$ with the largest attraction to O, i.e.,

$$Attractor(O) = \arg \max_{O_j \in A(O)} f(O_j, O)$$

According to the influence function, the attractor of an object O is its closest neighbor with a higher density. The only exception is object O_{hd} whose density is the highest in the data set. We define the attractor of O_{hd} is O_{hd} itself.

The attraction from an object to another (i.e., $O_i \rightarrow O_j$) forms a partial order. Based on this partial order, we can derive an *attraction tree* T. Each node corresponds to an object O. The parent node of O is set to

$$Parent(O) = \begin{cases} nil & \text{if } Attractor(O) = O \\ attractor(O) & \text{otherwise} \end{cases}$$

We define the weight for each edge $e(O_i, O_j)$ on the attraction tree T as the similarity between O_i and O_j.

The attraction tree has the following two features. On one hand, the attraction tree is *self-closed*. A group of objects following the same coherent pattern forms a attraction subtree. Objects following different coherent patterns are not mixed in the same attraction subtree. On the other hand, the attraction tree is *robust to noises*. The root of each attraction subtree has the locally maximal density and represents the coherent pattern in this attraction subtree. Objects closely matching the coherent pattern stay at the high levels of the tree, while noises (or intermediate objects) stay at the low levels

[1] Hereafter, we use the terms "objects" and "genes" exchangeably.

of the tree. Even in an environment with a large amount of noises or intermediate objects, since the density of noises or intermediate objects are relatively lower than that of the co-expressed objects, the structure of the high levels of the attraction tree will not change and the representative of coherent patterns will not be deviated by the noises or the intermediate objects.

3. INTERACTIVE EXPLORATION OF CO-HERENT PATTERNS

The general idea of interactively exploring the coherent patterns is as follows. Technically, we want to plot the genes as well as its probability to be a "leader" in a group of co-expressed genes as a curve, so that a user can identify the coherent patterns as well as the corresponding genes easily. Roughly speaking, this can be achieved in the following four steps.

First, we order the genes into an *index list*, such that the genes sharing a coherent pattern stay close to each other in the list. Second, each gene carries 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. Third, a coherent pattern index graph can be plotted. The genes are laid out horizontally according to the index list, and the vertical axis is the coherent pattern index value. Within the coherent index graph, a sharp pulse may strongly indicate the existence of a coherent pattern. Last, a user may choose the pulse in the coherent pattern index graph and then the coherent pattern as well as the corresponding co-expressed genes can be derived. The user can recursively examine the selected groups of co-expressed genes as well as its sub-patterns in depth.

3.1 Generating the Index List

To plot the genes as well as its probability to be a "leader" in a group of co-expressed genes in a 2-dimensional space, we need to order the genes into a list. An ordering, *index list*, can be devised based on the following three observations. (1) In the attraction tree, the edges connecting a pair of objects O_1 and O_2 following the same coherent pattern P have heavy weights. Genes connected by those edges should stay close to each other in the list. (2) The edges connecting a pair of intermediate objects O_1 and O_2 or connecting a pattern correlated object and an intermediate object have moderate weights. Genes connected by those edges should stay close to each other in the list, too, but not as close as the ones in case 1. (3) The edges connecting a pair of objects O_1 and O_2 following different coherent patterns P_1 and P_2 have light weights. Genes connected by those edges should stay far away in the list.

Based on the above idea, we develop an algorithm to order the genes, as shown in Figure 4. In the algorithm, we maintain an FIFO list, called $processedVertex$, 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 the weight descending order. Then, we iteratively extract the edge with the highest weight from the heap. At this point, the start vertex of the edge must have been processed (Otherwise, the edge could not be put into the heap.) We put the end vertex of the edge $currentVertex$ into the list $processedVertex$ 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 $processedVertex$ is the *index list* of the data genes.

3.2 Coherent Pattern Index and the Graph

Suppose we check the genes one by one in the order of index list, if we find a consecutive subsequence S of genes such that the genes

```
Procedure ordering(AttractionTree root){
  // Initialize the processedVertex and the edgeHeap
   processedVertex.add(root)
   for each child ch of root do edgeHeap.insert(edge(root, ch))
  // Iteration
  while ( !edgeHeap.isEmpty() ) do {
      currentEdge = edgeHeap.extract()
      currentVertex = currentEdge.endVertex
      processedVertex.add(currentVertex)
      for each child ch of currentVertex
        edgeheap.insert(edge(currentVertex, ch)) }}
```

Figure 4: The algorithm ordering the genes.

in S are much more coherent to their parents in the attraction tree than the genes in the precedent subsequence of S do, then it may strongly suggest that S is the starting segment of a group of co-expressed genes. Remember that in the constructions of attraction tree and index list, co-expressed genes are located in subtrees and thus are arranged as neighbors in the index list. This is the intuition of the design of coherent pattern index.

Then, the problem becomes how to find those *probes*, the short subsequences of genes at the beginning of the groups of co-expressed genes. In the 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 probe as a parameter. 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 the gene is the starting one of a group of co-expressed 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 bring the first sharp pulse.

Figure 5 is the coherent pattern index graph for Iyer's data with $p = 5$. The coherent pattern index graph indicates the existence of coherent patterns clearly.

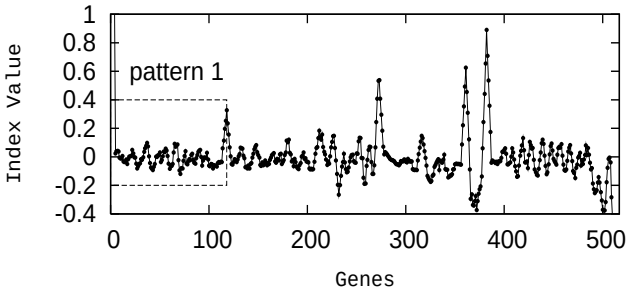

Figure 5: The coherent pattern index graph for the Iyer's data

3.3 Drilling down to Subgroups

From Figure 5, we can immediately tell that there are 5 major coherent patterns in the data set. However, *can we further investigate the groups of co-expressed genes following the coherent patterns and identify subgroups of co-expressed genes that follow any derivation patterns?*

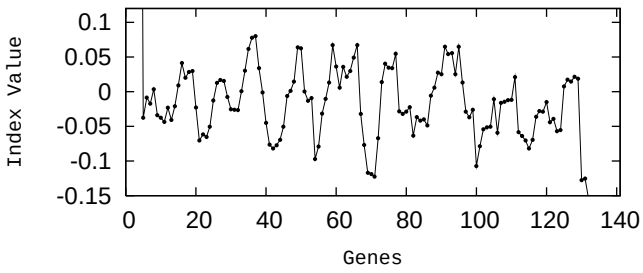

Figure 6: The coherent pattern index graph for a subset of genes in the Iyer's data set

Suppose a user accepts the 5 major coherent patterns reported by the system and click on the corresponding peaks in the coherent pattern index graph. The system will split the attraction tree T for the whole data into 5 exclusive attraction subtrees. Each subtree corresponds to one coherent pattern and the genes following that coherent pattern are gathered in that subtree. Therefore, the original data set is partitioned into 5 subsets accordingly.

Suppose a user now selects the first subset of genes $\mathcal{D}_1$ (as shown in Figure 5) and wants to zoom in $\mathcal{D}_1$. Figure 6 shows the local coherent pattern index graph for the selected subset of genes. Please note that Figure 6 is not simply extracted from Figure 5 with a higher resolution. Instead, we collect $\mathcal{D}_1$ from the attraction tree such that genes following the coherent pattern are selected. Then, the attraction tree, the index list and the coherent pattern index graph are generated, respectively. Only the genes in the selected subset are 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 (Equation 3), a smaller σ will boost the relative influence of a gene to its neighborhood. A detailed discussion on the effect of σ on the influence calculation can be found in [6]. We use the standard deviation of the pairwise distance between genes as σ. When the data set is split into smaller subsets, the standard deviation will decrease. To lower the computational cost, we use a small sample of the data set to approximate the standard deviation.

4. PERFORMANCE EVALUATION

We implemented and tested our approach on both real world gene expression data and synthetic data. The system is implemented in Java. The tests are conducted on a Sun Ultra 10 work station with a 440MHz CPU and 256 MB main memory.

We compared our method to some state-of-the-art methods, including Optics [8], CAST [2], CLICK [9] and Adapt [5]. We implemented Optics and CAST according to the related research papers. For CAST, we run the algorithm with various settings of parameter t (the affinity threshold) and chose the result which matches the ground truth best. CLICK is downloaded from http://www.cs.tau.ac.il/~rshamir/expander/expander.html. We used the default parameter set in the algorithm. Adapt has a web interface at http://www.esat.kuleuven.ac.be/~thijs/Work/Clustering.html. We set the minimal number of genes in a cluster to 5 and the minimal probability of a gene belonging to a cluster to the default value 0.95.

We conducted extensive tests on three typical real data sets: the Iyer's data set, the Cho's data set and the CDC28 data set.[2]

We developed an integrated environment, *GeneXplorer*, for interactive exploration of gene expression data. We apply GeneXplorer on all the three real data sets. For each data set, GeneXplorer identifies most of the coherent expression patterns with better quality than the other methods. Limited by space, here we only report the interactive exploration process on the Iyer's data set.

Figure 7 illustrates the exploration process. At the beginning, the coherent pattern index graph for the whole data set indicates five "major" coherent expression patterns. Suppose the user accepts the indication and asks the system to split the data set accordingly. Then the user can find some subsets already show some coherent patterns (e.g., the 2^{nd} and the 4^{th} subsets in the second row of the figure) clearly, while the others need to be further investigated.

The system generates the coherent pattern index graphs for the remaining subsets, respectively (i.e., the 1^{st}, the 3^{rd} and the 5^{th} subsets in the second row of the figure). In each of the coherent pattern index graphs for the subsets, there are multiple significant pulses.

How should we further split the data set and explore the finer patterns? Generally, the highest peak acts as a good signal. The user can ask the system to split the subset according to the highest peak in the graph. If the splitting result is not satisfying, the user can easily "roll back" to the previous level and choose the second highest peak to split the data set. In our experiment, we just assume the user chooses the highest peak in each subset and split the data set accordingly. The hierarchy extends to the third level. Such an interactive exploration can be conducted recursively, until the user is satisfied with the patterns and the groups of co-expressed genes.

We also compare the coherent expression patterns discovered by our GeneXploer with the ground truth and with the results from Adapt, CLICK and CAST. Suppose $\{P_1, \ldots, P_n\}$ is the set of coherent expression patterns in the ground truth and $\{\tilde{P_1}, \ldots, \tilde{P_m}\}$ is the set of coherent expression patterns discovered by one mining method. For each pattern P_i in the ground truth, we find the most similar pattern $\tilde{P_j}$ from the mining results, and call $\tilde{P_j}$ "match" the ground truth pattern P_i. In Figure 8, we list the similarity between the patterns in the ground truth and the matching patterns from various methods. We highlight with bold font the ground truth coherent expression patterns which are "matched" with similarity value above 0.9. The numbers in the parenthesis in the first row are the number of coherent patterns *returned* by each method.

The results clearly shows that GeneXplorer discovers most of the patterns in the ground truth in the Iyer's data set, and outperforms all other methods. In particular, Pattern 5 in the ground truth is *only* matched by GeneXplorer with a high similarity value. Fur-

[2]Iyer et al. monitored the expression levels of 8,600 distinct human genes during a 12 point time-series of serum stimulation [7]. Genes whose expression levels change significantly during the time-series were selected for cluster analysis. Only 517 genes survived after the significance test. In [4], the authors listed 10 co-expressed gene groups and the corresponding coherent patterns. We use the above list as the ground truth to test and compare the performance of our approach and other algorithms. Cho et al. [3] reported the genome-wide 6,220 mRNA transcript levels during the cell cycle (sampled at 16 time instants within roughly 2 cell cycles) of the budding yeast *S. cerevisiae* (CDC28 data set). Cell-cycle-dependent periodicity was found in 416 genes out of the 6,200 monitored ones. Among the 416 cell-cycle correlated genes, 386 genes have expression levels peaking at one of the following five phases: the early G_1 phase, the late G_1 phase, the S phase, the G_2 phase and the M phase, while the other 30 genes have expression levels peaking at multiple phases. All of the cell-cycle correlated genes, together with their peaking phases, are listed at http://171.65.26.52//yeast_cell_cycle/functional_categories.html. Therefore, the 386 genes naturally form five groups of co-expressed genes and follow five coherent expression patterns. We use this data set (Cho's data set) to test whether those 5 cell-cycle correlated patterns can be discovered by our approach and the other algorithms.

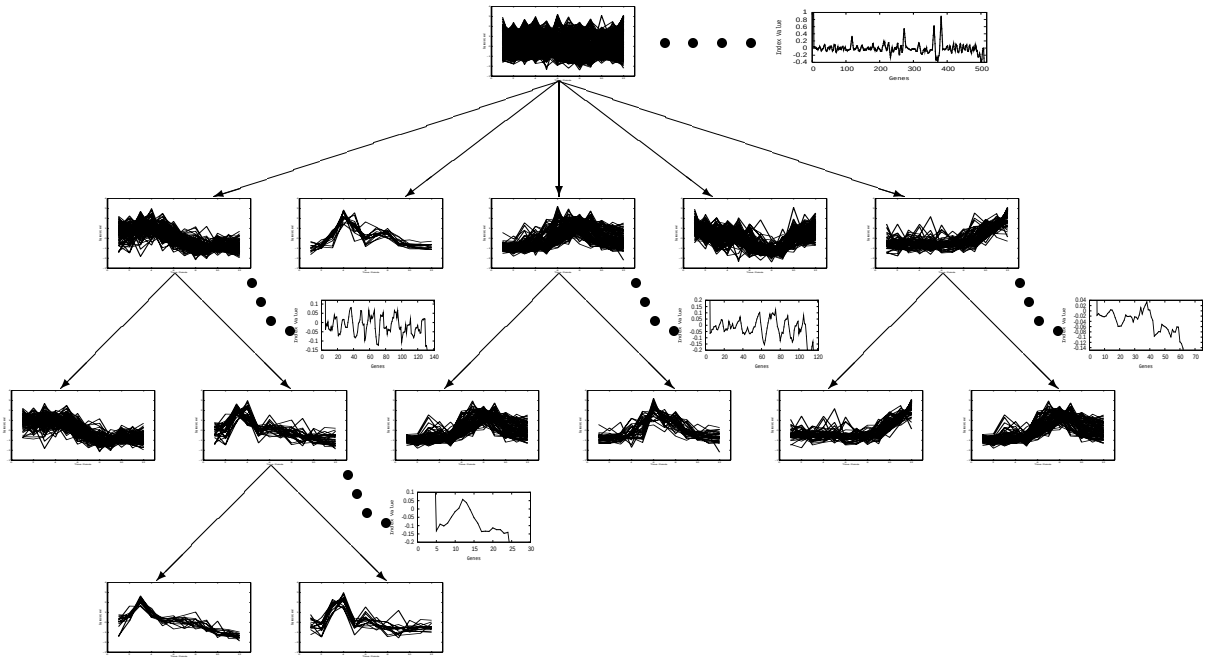

Figure 7: The hierarchy of co-expressed gene groups in the Iyer's data set

Pattern	GeneXplorer (9)	Adapt (11)	CLICK (7)	CAST (9)
1	**0.993**	**0.956**	0.884	**0.955**
2	**0.957**	**0.911**	**0.991**	0.887
3	**0.984**	**0.993**	**0.994**	**0.997**
4	**0.980**	**0.984**	0.883	**0.968**
5	**0.958**	0.855	0.868	0.855
6	**0.952**	**0.989**	**0.970**	**0.984**
7	**0.967**	**0.976**	**0.990**	0.719
8	**0.991**	**0.997**	**0.914**	**0.999**
9	0.702	0.824	0.844	0.800
10	**0.974**	**0.981**	**0.976**	**0.996**

Figure 8: Coherent patterns discovered in the Iyer's data set by different approaches

thermore, the only pattern in the ground truth (pattern 9) that GeneXplorer misses (matched with a low similarity value) cannot be matched with a high similarity value by any other method, either. The reason is that it is too similar to pattern 6 and thus hard to be distinguished.

We also tested the effect of the probe size and the scalability of our method, and compared it with Opitcs. The results show that our method is insensible to the setting of probe size, and is scalable w.r.t. the number of genes. The index list is also more effective than the reachability-plot by Optics. Limited by space, we omit the details here.

5. CONCLUSIONS

In this paper, we studied the problem of effectively mining coherent patterns from time-series gene expression data by proposing an interactive exploration framework. We developed a novel interactive exploration tool, coherent pattern index graph, to give users highly confident indications of the existence of coherent patterns. To derive a coherent pattern index graph, we devised an attraction tree structure to record the genes in the data set which summarizes the information needed for the interactive exploration. Our experimental study strongly shows that our approach is more effective than the state-of-the-art methods in mining real gene expression data, and is scalable in mining large data sets.

6. REFERENCES

[1] Alon U., et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. *Proc. Natl. Acad. Sci. USA*, Vol. 96(12):6745–6750, June 1999.

[2] Ben-Dor A., Shamir R. and Yakhini Z. Clustering gene expression patterns. *J. of Computational Biology*, 6(3/4):281–297, 1999.

[3] Cho, R. J., et al. A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell*, Vol. 2(1):65–73, July 1998.

[4] Eisen, Michael B., et al.. Cluster Analysis and Display of Genome-wide Expression Patterns. *Proc. Natl. Acad. Sci. USA*, 95(25):14863–14868, December 1998.

[5] Frank De Smet, et al.. Adaptive quality-based clustering of gene expression profiles. *Bioinformatics*, 18:735–746, 2002.

[6] Hinneburg, A., Keim, D.A. An efficient approach to clustering in large multimedia database with noise. *Proc. KDD'98*.

[7] Iyer V.R., et al. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.

[8] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jrg Sander. OPTICS: Ordering Points To Identify the Clustering Structure. *SIGMOD'99*, pages 49–60.

[9] R. Shamir and R. Sharan. Click: A clustering algorithm for gene expression analysis. In *In Proc. ISMB '00). AAAI Press.*

[10] Tamayo P., et al. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, Vol. 96(6):2907–2912, March 1999.

[11] Tavazoie, S., et al. Systematic determination of genetic network architecture. *Nature Genet*, pages 281–285, 1999.

[12] Wang, Haixun, Wang, Wei, Yang, Jiong and Yu, Philip S. Clustering by Pattern Similarity in Large Data Sets. In *SIGMOD'02*, pages 394–405.