

Mining Multiple Phenotype Structures Underlying Gene Expression Profiles

Chun Tang and Aidong Zhang
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260
{chuntang, azhang}@cse.buffalo.edu

ABSTRACT

DNA microarray technology is now widely used in basic biomedical research for mRNA expression profiling and are increasingly being used to explore patterns of gene expression in clinical research. Automatically detecting phenotype structures from gene expression profiles can provide deep insight into the nature of many diseases as well as lead in the development of new drugs. While most of the previous studies focus on only mining empirical phenotype structure which the experiment controls, it is also interesting to detect possible hidden phenotype structures underlying gene expression profiles.

Since the number of samples is usually limited, such data sets are very sparse in high-dimensional gene space. Furthermore, most of the genes of interest are buried in large amount of noise. Unsupervised phenotype structure discovery of such sparse high-dimensional data sets present interesting but challenging problems. In this paper, we propose the model of simultaneously mining both empirical and hidden phenotype structures from gene expression data. We demonstrate the effectiveness and efficiency of the proposed method on various real-world data sets.

Categories and Subject Descriptors

H.2.8 [Database Applications]: [Data Mining]

General Terms

Algorithms, Experimentation

Keywords

Phenotype, informative genes, array data, bioinformatics

1. INTRODUCTION

DNA microarray technology can be used to measure expression levels for thousands of genes in a single experiment, across different samples. The raw microarray data are transformed into gene expression matrices. Figure 1 shows an example. Usually, a row in a matrix represents a gene and a column represents a sample. The

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.
Copyright 2003 ACM 1-58113-723-0/03/0011 ...\$5.00.

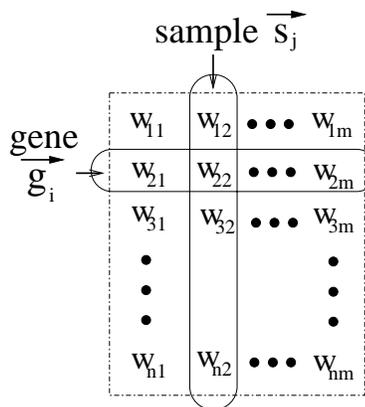


Figure 1: An example of a gene expression matrix.

numeric value in each cell characterizes the expression level of a specific gene in a particular sample.

Effective and efficient analysis techniques are demanding as gene expression data are accumulated rapidly. The gene expression matrix can be analyzed in two ways. On one hand, co-expressed genes can be grouped based on their expression patterns [8]. In such **gene-based analysis**, the genes are treated as the objects, while the samples are the attributes. On the other hand, the samples can be partitioned into homogeneous groups. Each group may correspond to a particular macroscopic phenotype, such as the present or absent clinical syndromes or cancer types [9]. Such **sample-based analysis** regards the samples as the objects and the genes as the attributes.

In typical array data sets, the volume of genes and the number of samples are very different, e.g., $10^1 \sim 10^2$ samples versus $10^3 \sim 10^4$ genes. Gene-based analysis and sample-based analysis therefore face very different challenges. In particular, due to the very high dimensionality in sample-based analysis, the techniques that are effective for gene-based analysis, such as MST [22], CAST [4], CLICK [18], OPTICS [12], and model-based clustering [25, 3], may not be adequate for analyzing samples.

In this paper, we will focus on the *sample-based analysis*. In particular, we are interested in mining *multiple phenotype structures*.

Within a gene expression matrix, there are usually several **empirical phenotypes** of samples controlled by the biological experiment, such as diseased samples, normal samples or drug treated samples. Figure 2 shows a gene expression matrix containing three empirical phenotypes of samples. Samples 1 ~ 3 belong to one phenotype, samples 4 ~ 7 to another phenotype while the remain-

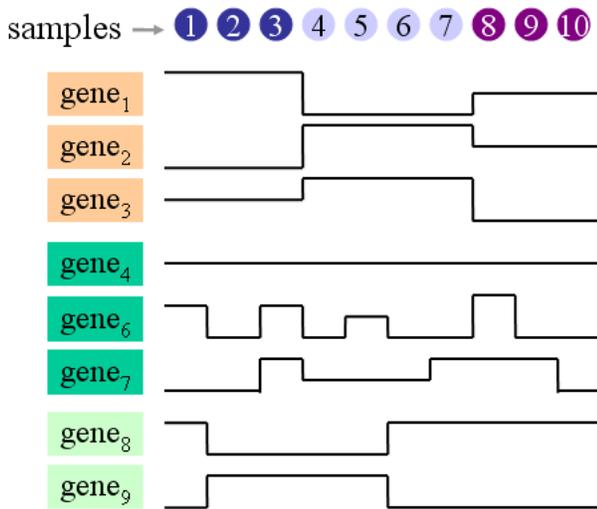


Figure 2: A simplified illustration of expression level distribution within a gene expression matrix.

ders belong to the third phenotype. For the sake of simplicity, the gene expression levels in the matrix are discretized into three-level ordinal values, i.e., either “high”, “intermediate” or “low”. Previous studies [9] have demonstrated that the empirical phenotypes of samples can be discriminated through a small subset of genes whose expression levels strongly correlate with the phenotype distinction. These genes are called *informative genes*. For example, in Figure 2, $gene_1 \sim gene_3$ are informative genes whose expression levels are low for one phenotype samples, intermediate for another phenotype and high for the third empirical phenotype.

Recent efforts in bioinformatics have studies methods for detecting the *empirical phenotypes* and finding the corresponding *informative genes* from the gene expression data based on either supervised [9, 5, 20] or unsupervised techniques [21, 19].

Although these methods are helpful, most of the genes collected may not necessarily manifest the empirical phenotype structure and there might be some **hidden phenotype structures** of other clinical interest buried in the data. For example, in Figure 2, although $gene_4 \sim gene_9$ cannot be used to distinguish different empirical phenotypes, $gene_8$ and $gene_9$ can distinguish a hidden phenotype structure: sample $\{2 \sim 5\}$ vs. samples $\{1, 6 \sim 10\}$.

Therefore, it is natural to ask “*Can we find both the empirical and the hidden phenotype structures automatically at the same time?*” Generally, a **phenotype structure** refers to: 1) a *exclusive and exhaustive partition of the samples* that samples of each group within the partition represent a unique phenotype; and 2) a *set of informative genes* manifesting this partition that each informative gene displays approximately invariant signals on samples of the same phenotype and highly differential signals for samples between different phenotypes. For example, in Figure 2, if no phenotype information are known in advance, can we correctly distinguish two phenotype structures of the samples: $\{\{1 \sim 3\}, \{4 \sim 7\}, \{8 \sim 10\}\}$ and $\{\{2 \sim 5\}, \{1, 6 \sim 10\}\}$ as well as output the corresponding informative gene sets?

Automatically mining phenotype structures is challenging. First, the values within data matrices are all real numbers such that there is usually no clear border between informative genes and noise genes. Second, there are many genes but a small number of sam-

ples. There is no existing technique to correctly detect phenotype structures from samples. Last, most of the genes collected may not necessarily be of interest. The experience shows that only less than 10% of all the genes involved in a gene expression matrix manifest the empirical phenotype structure [9] and the percentage of genes that manifest a hidden phenotype structure is even less. In other words, the gene expression matrix is very noisy.

In this paper, we tackle the problem of *mining multiple phenotype structures from gene expression data* by developing a novel unsupervised learning method. We claim the following contributions.

- We identify and formulate the problem of simultaneously mining both empirical phenotype structure and hidden phenotype structures from gene expression data.
- A set of statistic-based metrics of phenotype quality are proposed. They coordinate and compromise both the sample phenotype discovery and the informative gene selection.
- A iterative adjustment method is devised to find phenotype structures with high quality. This method dynamically manipulates the relationship between samples and genes while conducting an iterative adjustment to detect the phenotypes and informative genes.

The remainder of this paper is organized as follows. Section 2 reviews the related work. The phenotype quality measurements are proposed in Section 3, while the mining algorithm is developed in Section 4. Section 5 presents the experimental results and the conclusions in Section 6.

2. RELATED WORK

Our investigation is closely related to the research on both the sample-based analysis of microarray data and the clustering methods. We review some highly related work briefly in this section.

2.1 Unsupervised Empirical Phenotype Structure Detection

Recently, some methods have been proposed to find empirical phenotypes from samples assuming no sample class information is known in advance [16, 17]. In these approaches, samples are partitioned by conventional clustering methods, such as K-means, self-organizing maps (SOM), hierarchical clustering (HC), or graph based clustering. However, these traditional clustering techniques may not be effective to detect the phenotypes because the clustering is based on the full gene space that cannot handle the heavy noise well in the gene expression data. Although some approaches [21, 7, 14] reduce gene dimension or filter genes for clustering samples, the genes filtering processes are non-invertible and the deterministic filtering causes the samples to be grouped based on the local decisions. Furthermore, these approaches only focus on detecting the empirical phenotype structure, there is no efficient way to extend them to find the hidden phenotype structures.

2.2 Sub-space Clustering Methods

Sub-space clustering have been studied extensively [1, 6, 23] to find subsets of objects such that the objects appear as a cluster in a sub-space formed by a subset of the attributes. Although the sub-space clustering problem may appear similar to the phenotype structure detection problem at the first look, there are significant inherent differences between these two in the following aspects.

- In subspace clustering, the subsets of attributes for various subspace clusters are different and different subspace clusters can share some common attributes. However, in phenotype structure mining, we want to find a unique set of genes to

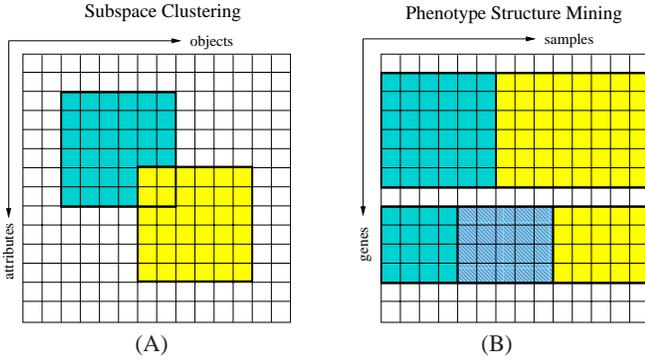


Figure 3: Differences of subspace clustering and phenotype structure mining. (A) shows a data matrix containing two subspace clusters. (B) shows a data matrix containing two phenotype structures. The upper structure contains two phenotypes while the lower one contains three phenotypes.

manifest each phenotype partition and exclusive sets of genes for different phenotype partitions.

- Two subspace clusters can share some common objects. Some objects may not belong to any subspace cluster. Nevertheless, in each phenotype structure, a sample must be in a phenotype and the phenotypes are exclusive.
- The pattern similarity measurements presented for subspace clustering (e.g., [23, 6]) can only detect whether genes rise and fall simultaneously within a subspace cluster. But in phenotype structure mining, the informative genes are required to show intra-consistent values. New metrics need to be defined to detect the informative genes.
- Subspace clustering algorithms only detect local correlated attributes and objects without considering dissimilarity between different clusters. To mine both phenotypes and informative genes, we want to get the genes which can differentiate all phenotypes for each phenotype structure.

3. PHENOTYPE QUALITY METRICS

A **candidate phenotype structure** contains a k -partition of samples and a subset of genes. In this section, we will introduce the metrics to measure the possibility for a single candidate phenotype structure to be an empirical or hidden phenotype structure. The mining of multiple phenotype structures will be discussed in the next section based on these metrics.

Each empirical or hidden phenotype structure should satisfy two requirements simultaneously. On one hand, the expression levels of each informative gene should have similar expressions over the samples of the same phenotype. On the other hand, the expression levels of each informative gene should display a clear dissimilarity between each pair of phenotypes in a certain phenotype structure. We introduce two separate statistical metrics: *intra-consistency* which measures the similarity of the gene expressions within each sample group of a candidate phenotype structure and *inter-divergency* which measures the dissimilarity between different sample groups on the candidate gene set. These metrics are combined to provide the *phenotype quality* metric, which is used for qualifying a phenotype structure.

Let $S = \{\vec{s}_1, \dots, \vec{s}_m\}$ be a set of samples and $G = \{\vec{g}_1, \dots, \vec{g}_n\}$ be a set of genes. The corresponding gene expression matrix can be represented as $M = \{w_{i,j} | 1 \leq i \leq n, 1 \leq j \leq m\}$, where $w_{i,j}$ is the expression level value of sample \vec{s}_j on gene \vec{g}_i . Usually

we have ($n \gg m$). The candidate phenotype structure includes a k -partition of samples $\{S_1, \dots, S_k\}$, where $S = \bigcup_{i=1}^k S_i$ and $S_i \cap S_j = \emptyset$ for ($1 \leq i < j \leq k$), and a set of genes $G' \subseteq G$.

3.1 Intra-consistency

Assume S' is one of the sample groups within the partition $\{S_1, \dots, S_k\}$ and $M_{S',G'} = \{w_{i,j} | \vec{g}_i \in G', \vec{s}_j \in S'\}$ is the corresponding sub-matrix with respect to S' and G' . The **variance** of each row in the sub-matrix is defined as:

$$Var(i, S') = \frac{\sum_{\vec{s}_j \in S'} (w_{i,j} - \bar{w}_{i,S'})^2}{|S'| - 1}, \quad (1)$$

where $\bar{w}_{i,S'} = \frac{1}{|S'|} \sum_{\vec{s}_j \in S'} w_{i,j}$. The *variance* of each row measures the variability of a given gene over all samples within the sub-matrix. A small variance value indicates that the gene has consistent values.

We can measure whether in a subset of genes, every gene has good consistency on a group of samples by the average of variance in the subset of genes. That is, we define the **intra-consistency** as:

$$\begin{aligned} Con(G', S') &= \frac{1}{|G'|} \sum_{\vec{g}_i \in G'} Var(i, S'), \\ &= \frac{1}{|G'| \cdot (|S'| - 1)} \sum_{\vec{g}_i \in G'} \sum_{\vec{s}_j \in S'} (w_{i,j} - \bar{w}_{i,S'})^2. \end{aligned} \quad (2)$$

3.2 Inter-divergency

We introduce the *inter-divergency* to quantize how a subset of genes can distinguish two phenotypes of samples.

The **inter-divergency** of a set of genes G' on two groups of samples (denoted as S_1 and S_2) is defined as

$$Div(G', S_1, S_2) = \frac{\sum_{\vec{g}_i \in G'} |\bar{w}_{i,S_1} - \bar{w}_{i,S_2}|}{|G'|}, \quad (3)$$

where \bar{w} is defined as in Equation 1. The measure is normalized by $|G'|$ to avoid the possible bias due to the volume of genes. The greater the inter-divergency, the better the genes differentiate the samples in different groups.

3.3 Phenotype Quality of a Candidate Phenotype Structure

We define the following quality measure Ω to quantize how good is the candidate phenotype structure.

$$\Omega = \frac{1}{\sum_{S_i, S_j (1 \leq i, j \leq k; i \neq j)} \frac{\sqrt{Con(G', S_i) + Con(G', S_j)}}{Div(G', S_i, S_j)}}. \quad (4)$$

Large value of *phenotype quality* is expected for qualifying an empirical or hidden phenotype structure.

4. MODEL AND STRATEGY

The **problem of mining multiple phenotype structures** can be described as Figure 4. Here we assume that there is no overlap between different informative gene sets because usually a gene cannot manifest different phenotype partitions. Since the gene expression data are noisy and complex, mining useful information buried with each data set is still an open problem and largely depends on domain knowledge. We assume the number of structures p and the number of phenotypes of each phenotype structure are user-specified.

It can be shown that finding the optimal phenotype structures

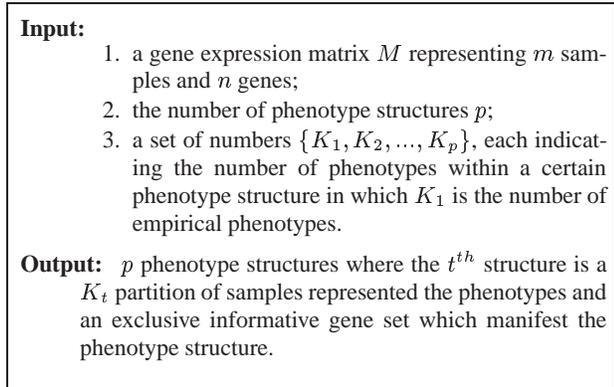


Figure 4: The problem description.

(i.e., the summation of phenotype qualities of all structures is maximized) is an NP-hard problem. As we may often have tens of thousands of genes and hundreds of samples, finding the optimal answer can be very costly. In this section, we will develop a heuristic searching method to approximate the best solution. It adopts the *simulated annealing technique* [11] and dynamically measures and manipulates the relationship between samples and genes while conducts an iterative adjustment of the candidate phenotype structures to approximate the best quality.

4.1 The Algorithm

The algorithm (shown in Figure 5) contains two phases: the candidate structure generation phase and the iterative adjustment phase. We first generate p candidate phenotype structures based on clustering techniques. In each iteration, the candidate sets will be iteratively adjusted towards the best summation of phenotype quality values. Details of the algorithm will be discussed in the following subsections.

4.1.1 Basic Elements

The algorithm maintain p candidate phenotype structures, each structure contains the following items ($1 \leq t \leq p$):

- a K_t partition of samples $\{S_1, S_2, \dots, S_{K_t}\}$ such that $S_i \cap S_j = \emptyset$ for $(1 \leq i < j \leq K_t)$ and $S = \bigcup_{i=1}^{K_t} S_i$.
- A set of genes $G_t \subseteq G$, which is a candidate set of the t^{th} informative gene set.
- The *phenotype quality* (Ω_t) of the t^{th} structure is calculated based on the partition $\{S_1, S_2, \dots, S_{K_t}\}$ on G_t .

The summation of phenotype qualities of all structures is regarded as the **quality** of the current candidate structures: $\Omega = \sum_{t=1}^p \Omega_t$.

An **adjustment** is an action to change the current candidate structures which is one of the following:

- For a gene $g_i \notin \forall G_t$, *insert* g_i into one of G_t . Thus there are p possible insertions;
- For a gene $g_i \in G_t$, *move* g_i to $G'_t (t' \neq t)$ or *remove* g_i from all candidate gene sets. Thus there are also p possible adjustments.
- For a sample s in partition group S' in t^{th} candidate structure, *move* s_i to partition group S'' where $S' \neq S''$. Thus, each sample has $K_t - 1$ possible adjustments for the t^{th} candidate structure, totally $\prod_{t=1}^p (K_t - 1)$ possible adjustments.

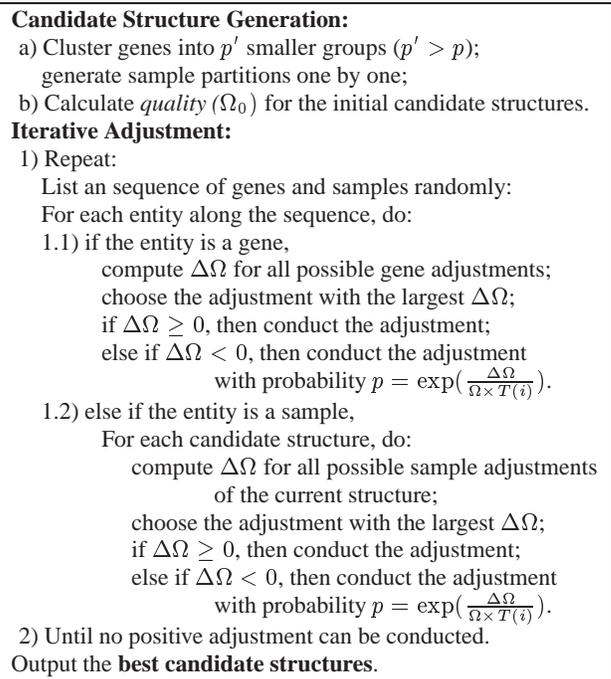


Figure 5: The algorithm.

To measure the effect of an adjustment, we calculate the **quality gain** of the adjustment as the change of the quality, i.e., $\Delta\Omega = \Omega' - \Omega$, where Ω and Ω' are the quality of the candidate structures before and after the adjustment, respectively.

The algorithm also records the **best candidate structures**, in which the highest quality so far is achieved.

4.1.2 Candidate Structure Generation

In this phase, we generate p candidate phenotype structures. One easy way is to randomly initialize p candidate structures, but the result will be unstable. Here we adopt a more deterministic, robust strategy for candidate structure generation.

The first step is to divide the whole genes into p' smaller groups where $p' > p$. The algorithm CAST (for cluster affinity search technique) [4] is applied to group genes and the *Pearson's Correlation Coefficient* is chosen to calculate the similarity matrix. CAST is a method specially designed for grouping gene expression data based on their pattern similarities. One advantage of CAST is that the number of groups does not need to be pre-specified. A threshold has to be set to approximate the size of each group. In biological applications, the experience shows that genes associated with similar functions always involve from several dozen to several hundred entities [9]. Thus, when grouping genes, the threshold should be set so that a majority of groups will contain several dozen to several hundred genes. Then we generate p' sub-matrices that each sub-matrix formed by one gene cluster and all samples.

The second step is to generate sample partitions one by one. First, we generate the candidate empirical phenotype partition. Our method is to cluster each sub-matrix on the sample dimension into K_1 group and calculate phenotype quality based on the cluster result for each sub-matrix. The sub-matrix with the highest phenotype quality is chosen that the sample partition becomes the candidate empirical phenotype partition and the genes within this sub-matrix become the corresponding informative genes. Then we generate the first candidate hidden phenotype partition which include

K_2 sample groups based on the rest sub-matrices. This process will be conducted for all hidden phenotype structures until all candidate structures are generated. We give the empirical phenotype structure the highest priority because the empirical structure is the most important information of a certain gene expression data.

Now, the goal becomes, given starting candidate structures, we try to apply a series of adjustments to reach the candidate structures such that the accumulated quality gain is maximized.

4.1.3 Iterative Adjustment

When we have a set of candidate structures, an immediate solution to the reach the goal is to heuristically move to better status by iteratively conducting adjustments. Hopefully, after a few rounds, we can be close to the optimal candidate structures. This heuristic moving carries the similar spirit as δ -cluster [23] and CLARANS [13].

In the iterative adjusting phase, during each iteration, genes and samples are examined one by one. For each gene, there are p possible adjustments, it can be either inserted into a candidate informative gene set or removed from all current candidate informative gene sets. The quality gain can be calculated for each possible adjustment and the adjustment with largest quality gain value is chosen. The adjustment will be conducted if $\Delta\Omega$ is positive. Otherwise, the adjustment will be conducted with a probability $p = e^{\frac{\Delta\Omega}{\Omega \times T(i)}}$, where $T(i)$ is a decreasing simulated annealing function [11] and i is the iteration number.

For each sample, since its partition membership is independent from different phenotype structures, we will examine the possible adjustment for each candidate structure, independently. For each candidate structure, there are $(K_t - 1)$ possible adjustments (i.e., the sample can be moved to one of the other $(K_t - 1)$ groups). We also calculate the quality gain, respectively. The adjustment with the largest quality gain will be chosen as the adjustment of the sample of a certain candidate structure. Similar to that of the genes, the adjustment will be conducted if $\Delta\Omega$ is positive. Otherwise, the adjustment will be conducted with a probability $p = e^{\frac{\Delta\Omega}{\Omega \times T(i)}}$.

This algorithm is sensitive to the order of genes and sample adjustments considered in each iteration. To give every gene or sample a fair chance, all possible adjustments are sorted randomly at the beginning of each iteration.

The probability function p has two components. The first part, $\frac{\Delta\Omega}{\Omega}$, considers the quality gain in proportion. The more Ω reduces, the less probability the adjustment will be performed. The second part, $T(i)$, is a decreasing simulated annealing function where i is the iteration number. When $T(i)$ is large, p will be close to 1 and the adjustment has high probability to be conducted. As the iteration goes on, $T(i)$ becomes smaller and thus the probability p also becomes less. In our implementation, we set $T(0) = 1$, and $T(i) = \frac{1}{1+i}$, which is a slow annealing function. The advantage of this annealing function is that the slow annealing is more effective to approach global optimal solution. But more iterations will be required.

As indicated in [11], a simulated annealing search can reach the global optimal solution as long as the annealing function is slow enough and there are sufficient number of iterations. The upper bound is the total number of possible solutions. However, a simulated annealing search may be practically infeasible to find the optimal solution. Thus, we set the termination criterion as *when-ever in an iteration, no positive adjustment is conducted*. Once the iteration stops, the best candidate structures will be output.

The time complexity of the heuristic algorithm is dominated by the iteration phase. The time to compute Ω at the beginning is in

$O(m \cdot n \cdot p)$. Here we ignore the effect introduced by K_t because K_t is usually very small. In each iteration, the time complexity depends on the calculation of Ω' for the possible adjustments. Since Equations 2, 3 and 4 are all accumulative, we can simplify the formula by only computing the changed part of the measurements. It can be proved that the time cost of computing Ω' is $O(m^2 \cdot p)$ for each gene, and $O(m \cdot n \cdot p)$ for each sample. There are n genes and m samples involved in each iteration. Therefore, the algorithm's time complexity is $O(n \cdot m^2 \cdot p \cdot l)$, where l is the number of iterations.

5. PERFORMANCE EVALUATION

In this section, we will report an extensive performance evaluation on the effectiveness and efficiency of the proposed two methods using various real-world gene expression data sets.

5.1 The Real-world Data Sets

The experiments are conducted on the following four gene expression data sets. The empirical phenotype structure controlled by the biological experiment of each data set is listed below.

The Leukemia Data Sets– The leukemia data sets are based on a collection of leukemia patient samples reported in [9]. It contains measurements corresponding to acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML) samples from bone marrow and peripheral blood. Two matrices are involved: one includes 38 samples (27 ALL vs. 11 AML, denoted as G1), and the other contains 34 samples (20 ALL vs. 14 AML, denoted as G2). Each sample is measured over 7129 genes.

The Multiple Sclerosis Data Sets– The multiple sclerosis (MS) dataset consists of array-derived gene expression profiles that were provided by our collaborators in the Department of Pharmaceutical Sciences and in the Department of Neurology. The data set contains two pair-wise group comparisons of interest. The first data subset, “MS vs. Controls”, contains array data from 15 MS samples and 15 age and sex-matched controls while the second subset is referred to as “MS-IFN” because it contains array data from 14 MS samples prior to and 24 hours after interferon- β (IFN) treatment. Each sample is measured over 4132 genes.

The colon Cancer Data Set– This data set consists of 22 normal and 40 tumor colon tissue samples. It was reported by Alon et al. in [2]. In this dataset, the number of genes is 2000.

The Hereditary Breast Cancer Data Set– This dataset is from Hedenfalk et al. [10]. They reported on a microarray experiment concerning the genetic basis of breast cancer. Tumors from 22 women were analyzed. Three types of samples are included in one data matrix: 7 of the women known to have the BRCA1 mutation, 8 known to have BRCA2, and 7 have no cancer being labeled “Sporadics”. Each sample is measured over 3226 genes.

The ground-truths of the empirical phenotype structure, which includes the information such as how many samples belong to each phenotype and the phenotype label for each sample, is used only to evaluate the experimental results.

5.2 Results

We first evaluate the empirical phenotype detection accuracy of the proposed method. Since there are ground-truths on the empirical phenotype structure, but no commonly accepted ground-truths on hidden phenotype structures, it is hard to compare the detection accuracy of hidden phenotype structures.

Data Set	MS_IFN	MS vs. Controls	Leukemia-G1	Leukemia-G2	Colon	Breast
Data size	4132 × 28	4132 × 30	7129 × 38	7129 × 34	2000 × 62	3226 × 22
K_1	2	2	2	2	2	3
SOM	0.4815	0.4920	0.6017	0.4920	0.4939	0.4112
SOM with PCA	0.5238	0.5402	0.5092	0.4920	0.4939	0.5844
CLUTO	0.4815	0.4828	0.5775	0.4866	0.4966	0.6364
δ -cluster	0.4894	0.4851	0.5007	0.4538	0.4796	0.4719
our method	0.8071	0.6137	0.9756	0.6868	0.6303	0.8248

Table 1: Rand Index value reached by applying different methods.

The *Rand Index* [15] between the ground-truth of the empirical phenotype structure P of the samples and the mining result of the empirical phenotype structure Q of an algorithm is adopted to evaluate the effectiveness of the algorithm. Let a represent the number of pairs of samples that are in the same cluster in P and in the same cluster in Q , b represent the number of pairs of samples that are in the same cluster in P but not in the same cluster in Q , c be the number of pairs of samples that are in the same cluster in Q but not in the same cluster in P , and d be the number of pairs of samples that are in different clusters in P and in different clusters in Q . The *Rand Index* is $RI = \frac{a+d}{a+b+c+d}$. The *Rand Index* lies between 0 and 1. Higher values of the *Rand Index* indicate better performance of the algorithm.

Table 1 provides the empirical phenotype detection results obtained by applying our model and some algorithms proposed previously [9, 14, 17, 23]. The clustering methods SOM, δ -cluster method, and our iterative adjustment approach are heuristic rather than deterministic, the results might be different in different executions. Thus for each approach, we run the experiments multiple times using different orders of objects or different parameters and calculate the average *Rand Index* values. Table 1 indicates that the method proposed in this paper consistently achieve better empirical phenotype detection results than the previously proposed methods. We analyze the results briefly as follows. In clustering methods such as self-organizing maps and graph-partitioning-based algorithm of CLUTO [17], objects are partitioned based on the full dimensional genes, the high percentage of irrelevant genes largely lower the performance. As indicated by [24], the principal components in PCA do not necessarily capture the class structure of the data. Therefore the methods assisted by PCA can not guarantee to improve the clustering results. The central idea of subspace clustering is different from our empirical phenotype detection. It is not surprising therefore that the δ -cluster algorithm is not effective in identifying the sample partition.

In the following, we evaluate the hidden phenotype structures identified by our approach. We do three sets of experiments with different numbers (10, 15 and 20) of hidden structures. Thus the total numbers of phenotype structures of three experiments are $p = 11$, $p = 16$ and $p = 21$. Since we have no pre-knowledge or ground-truth about how many phenotypes within each structure for each gene expression data set, we randomly generate a sequence of phenotype numbers ranged from 2 to 5 in each set of experiments. Using these sequences as $\{K_2, \dots, K_p\}$ to detect hidden phenotype structures from each data set. Figure 6 shows the phenotype quality of each phenotypes of each experiment. For example, Figure 6 (A) shows the phenotype qualities of 16 phenotype structures of each data set. The first point of each line indicates the empirical phenotype quality while the rest points indicate the hidden phenotype qualities. The numbers under x-axis show the number of phenotypes of each hidden structure. Figure 6 indicates that the proper number of phenotype structure of the gene expression data

is around 10 \sim 15. There are some trivial phenotype structure (quality value is rather small) when the p is bigger than 15.

Figure 7 shows two phenotype structures detected from Leukemia-G1 dataset by our method. In this figure, each column represents a sample, while each row corresponds to an informative gene. Different colors (grey degree in a black and white printout) in the matrix indicates the different expression levels. Figure 7 (A) shows the empirical phenotype structure. The first 27 samples belong to ALL group while the rest 11 samples belong to AML group. Among 45 informative genes, the top 19 genes distinguish ALL-AML phenotypes according to “on-off” pattern while the rest 26 genes follow “off-on” pattern. Figure 7 (B) shows one of the hidden phenotype structure in which the samples are divided into three groups. Each gene shows “low” for one class samples, “intermediate” for another class and “high” for the third class to distinguish three phenotypes. This hidden structure matches a subtype structure of the patients according to a previous study [9] that the first 19 samples are of B-cell ALL, the middle 8 samples are of T-cell ALL while the rest 11 samples belong to AML.

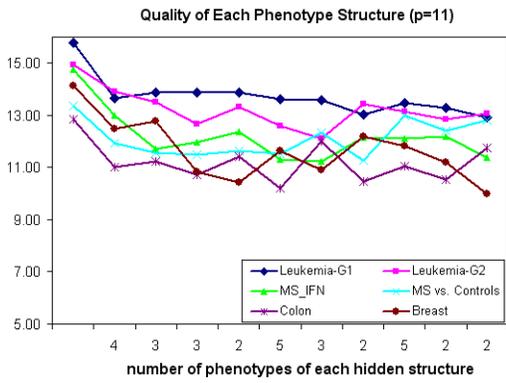
Data Size	# of iterations		running time (sec.)	
	mean	stand deviation	mean	stand deviation
4132 × 28	158	27.2	180	35.1
4132 × 30	168	29.5	195	37.8
7129 × 38	171	16.1	436	51.9
7129 × 34	198	35.9	458	101.2
2000 × 62	133	17.8	479	98.5
3226 × 22	157	22.2	167	35.6

Table 2: The mean value and standard deviation of the numbers of iterations and response time (in second) with respect to the matrix size.

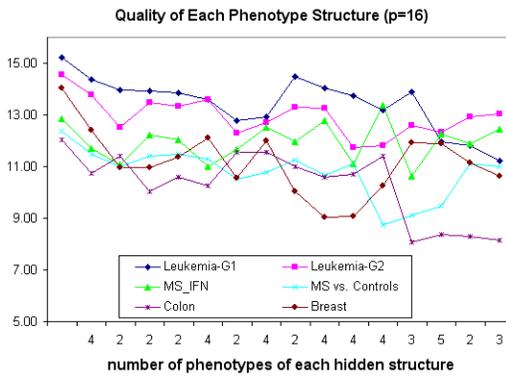
Table 2 reports the mean and standard deviation values of the numbers of iterations and the response time (in second) of the above gene expression data sets. The algorithm is executed 50 times with different parameters. The algorithm is implemented with MATLAB package and are executed on SUN Ultra 80 workstation with 450 MHz CPU and 256 MB main memory. The number of iterations are dominated by the simulate annealing function we used. We used a slow stimulate annealing function for effectiveness of the approaches. Since in reality, the number of genes in the human genome is about 30,000 \sim 50,000, efficiency is not a major concern.

6. CONCLUSIONS

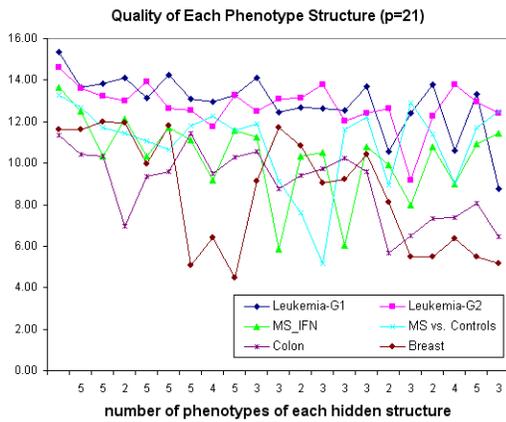
Effectively mining microarray data is an important bioinformatics research problem with broad applications. The previous studies focus on mining only empirical phenotype structure from the gene expression matrix. In this paper, we have proposed a novel approach to automatically detect both empirical phenotype structure



(A)

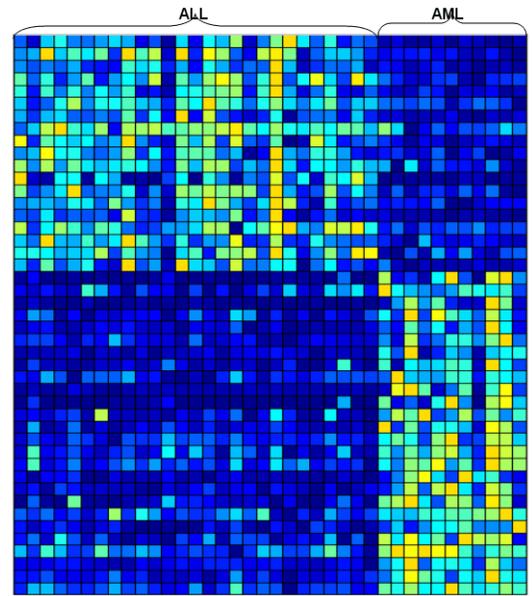


(B)

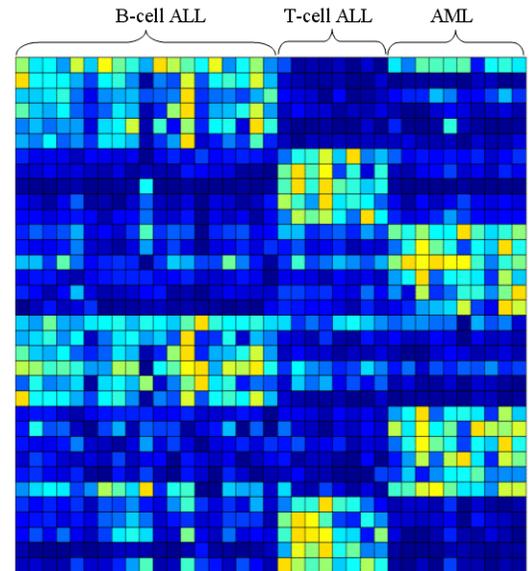


(C)

Figure 6: Phenotype qualities of each phenotype structures w.r.t. different numbers of phenotype structures.



(A)



(B)

Figure 7: Two of the phenotype structures detected from Leukemia-G1. (A) shows the empirical phenotype structure (45 informative genes). (B) shows a hidden phenotype structure with 3 groups of samples (34 informative genes).

and hidden phenotype structures from gene expression data simultaneously. A set of statistical measurements of phenotype quality are proposed to coordinate and compromise both the sample phenotype structure discovery and the informative gene mining.

We demonstrated the performance of the proposed approach by extensive experiments on various real-world gene expression data sets. The empirical evaluation shows that our approaches are effective and efficient on mining large real-world data sets.

7. REFERENCES

- [1] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.
- [2] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D. and Levine A.J. 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.
- [3] Barash Y. and Friedman N. Context-specific bayesian clustering for gene expression data. In *Proc. 5th Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 12–20. ACM Press, 2001.
- [4] Ben-Dor A., Shamir R. and Yakhini Z. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.
- [5] Brown M.P.S., Grundy W.N., Lin D., Cristianini N., Sugnet C.W., Furey T.S., Ares M.Jr. and Haussler D. Knowledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl. Acad. Sci.*, 97(1):262–267, January 2000.
- [6] Cheng Y., Church GM. Biclustering of expression data. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 8:93–103, 2000.
- [7] Ding, Chris. Analysis of gene expression profiles: class discovery and leaf ordering. In *Proc. of International Conference on Computational Molecular Biology (RECOMB)*, pages 127–136, Washington, DC., April 2002.
- [8] Eisen M.B., Spellman P.T., Brown P.O. and Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, Vol. 95:14863–14868, 1998.
- [9] Golub T.R., Slonim D.K. *et al.* Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, Vol. 286(15):531–537, October 1999.
- [10] Hedenfalk, I., Duggan, D., Chen, Y. D., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Kallioniemi, O. P., Wilfond, B., Borg, A., and Trent, J. Gene-expression profiles in hereditary breast cancer. *The New England Journal of Medicine*, 344(8):539–548, February 2001.
- [11] Kirkpatrick, S., Gelatt, C. D. Jr., and Vecchi, M. P. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [12] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jrg Sander. OPTICS: Ordering Points To Identify the Clustering Structure. *Sigmod*, pages 49–60, 1999.
- [13] Ng, Raymond T. and Han, Jiawei. Clarans: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):1003–1016, October 2002.
- [14] Peterson Leif E. Factor analysis of cluster-specific gene expression levels from cdna microarrays. *Computer Methods and Programs in Biomedicine*, 69(3):179–188, 2002.
- [15] Rand, W.M. Objective criteria for evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.
- [16] Rhodes, D.R., Miller, J.C., Haab, B.B., Furge, K.A. CIT: Identification of Differentially Expressed Clusters of Genes from Microarray Data. *Bioinformatics*, 18:205–206, 2001.
- [17] Schloegel, Kirk, Karypis, George. *CRPC Parallel Computing Handbook*, chapter Graph Partitioning For High Performance Scientific Simulations. Morgan Kaufmann, 2000.
- [18] Shamir R. and Sharan R. Click: A clustering algorithm for gene expression analysis. In *In Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*. AAAI Press., 2000.
- [19] Tang, Chun and Zhang, Aidong. An iterative strategy for pattern discovery in high-dimensional data sets. In *Proceeding of 11th International Conference on Information and Knowledge Management (CIKM 02)*, McLean, VA, November 4-9 2002.
- [20] Thomas J.G., Olson J.M., Tapscott S.J. and Zhao L.P. An Efficient and Robust Statistical Modeling Approach to Discover Differentially Expressed Genes Using Genomic Expression Profiles. *Genome Research*, 11(7):1227–1236, 2001.
- [21] Xing E.P. and Karp R.M. Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, Vol. 17(1):306–315, 2001.
- [22] Xu, Ying, Olman, Victor and Xu, Dong. Clustering gene expression data using a graph-theoretic approach: An application of minimum spanning trees. *Bioinformatics*, 18(4):536–545, 2002.
- [23] Yang, Jiong, Wang, Wei, Wang, Haixun and Yu, Philip S. δ -cluster: Capturing Subspace Correlation in a Large Data Set. In *Proceedings of 18th International Conference on Data Engineering (ICDE 2002)*, pages 517–528, 2002.
- [24] Yeung, Ka Yee and Ruzzo, Walter L. An empirical study on principal component analysis for clustering gene expression data. Technical Report UW-CSE-2000-11-03, Department of Computer Science & Engineering, University of Washington, 2000.
- [25] Yeung, K.Y., Fraley, C., Murua, A., Raftery, A.E., Ruzzo, W.L. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17:977–987, 2001.