EFFICIENT GENERALIZED MATRIX APPROXIMATIONS FOR BIOMARKER DISCOVERY AND VISUALIZATION IN GENE EXPRESSION DATA

Wenyuan Li, Yanxiong Peng, Hung-Chung Huang and Ying Liu*
Department of Computer Science, University of Texas at Dallas
Richardson, TX 75083, U.S.A.
**Email: ying.liu@utdallas.edu*

In most real-life gene expression data sets, there are often multiple sample classes with ordinals, which are categorized into the normal or diseased type. The traditional feature or attribute selection methods consider multiple classes equally without paying attention to the up/down regulation across the normal and diseased types of classes, while the specific gene selection methods particularly consider the differential expressions across the normal and diseased, but ignore the existence of multiple classes. In this paper, for improving the biomarker discovery, we propose to make the best use of these two aspects: the differential expressions (that can be viewed as the domain knowledge of gene expression data) and the multiple classes (that can be viewed as a kind of data set characteristic). Therefore, we simultaneously take into account these two aspects by employing the 1-rank generalized matrix approximations (GMA). Our results show that the consideration of both aspects can not only improve the accuracy of classifying the samples, but also provide a visualization method to effectively analyze the gene expression data on both genes and samples. Based on the GMA mechanism, we further propose an algorithm for obtaining the compact biomarker by reducing the redundancy.

1. INTRODUCTION

With the rapid advances of microarray technologies, massive amounts of gene expression data are generated in experiments. Analysis of these high-throughput data poses both opportunities and challenges to the biologists, statisticians, and computer scientists. One of the most important features in microarray data is the very high dimensionality with a small number of samples. There are over thousands of genes and at most several hundreds of samples in the data set. Such characteristic, which has never existed in any other type of data, has made the traditional data mining and analysis methods not effective, and therefore attracted the focus of recent research. Among these methods, a crucial approach is to select a small portion of informative genes for further analysis, such as disease classification and the discovery of structure of the genetic network [18]. Due to the drastic size difference of genes and samples, the step of gene selection is also the need of solving the well-known problem "curse of dimensionality" in statistics, data mining and machine learning [5].

However, quite different from the traditional feature selection in other data sets such as text [22], the final goal of gene selection is to discover the "biomarker", a minimal subset of genes that not only are differentially expressed across different sample classes, but also contains most relevant genes without redundancy. These two characteristics distinguish the task of discovering "biomarker" from the common feature selection tasks.

Recent gene selection methods fall into two categories: *filter methods* and *wrapper methods* [18]. The wrapper methods [3] are closely "embedded" in the classifier and thus are often time-consuming. On the other hand, the filter methods analyze the data by investigating their domain-specific targets: (1) differential expression across classes and (2) redundancies induced by the relevant genes. They are independent of the sample classification and are efficient in analyzing the functions of genes. Therefore, they attracted more focus of the studies in recent years.

The basic goals of these filter methods are to obtain a subset of genes with maximum relevance and minimum redundancy [9, 23, 4]. Most existing filter methods follow the methodologies of statistics [9] and information theory [4, 23, 18] to rank the genes and reduce the redundancy, such as t-like-statistics, mutual information or information gain based methods. These methods are computationally efficient. However, they select the biomarker by only considering the binary class labels, e.g., healthy/diseased,

*Corresponding author.

while the sample classes in the observed experiments are often ordinal with the gradually changing tendency [3]. For example, in the Lupus experiment (see Subsection 5.2), four classes of persons are considered. They are normal ones, relatives of patient, patients who show the early symptoms, and patients whose symptoms are complete. Also some gene expression experiments [a] consider classes of samples that are the composite of normal and disease ingredients with different scale, e.g., 1:4 or 3:4. In these gene expression data, although there are two types of classes, i.e., positive and negative, the labels of multiple classes show the ordinal scales according to the degree of their membership to the positive or negative type, e.g., 'normal', 'low-grade' tumor, 'intermediate-grade' tumor and 'high-grade' tumor [16]. However, when dealing with the data sets with such multiple classes and two types, most existing filtering methods e.g., information gain and t-statistics, combine all classes in positive type into a positive class and similarly combine all classes in negative type into a negative class, and then do the filtering process on the two combined classes. Such analysis may ignore the characteristics of the expression data within each single class, and therefore may lose the accuracy of discovering the biomarker with maximal relevance and minimal redundancy. On the other hand, most general feature selection methods, e.g., ReliefF [10], consider the multiple classes, but ignore the special characteristic of gene selection, up and down regulations. Therefore, they are not specific to the task of gene selection as well. There have been few works in the wrapper methods on investigating the biomarker on these data sets, such as Gaussian process model based method [3]. However, it is a wrapper method by using the leave-one-out error and forward selection and therefore is not efficient. Moreover, the original and intuitive objective of biomarker discovery is that the user can visually select the differentially expressed genes without redundancy. According to this objective, however, it is like a black-box screening the user out of the analysis process.

Therefore, in this paper, we propose a class of 1-rank Generalized Matrix Approximation (GMA) [b] filter method to simultaneously rank the genes and samples to identify the biomarker in the data sets with multiple classes. The GMA simultaneously takes into account the global between-class data distribution (differentially expression) and local within-class data distribution (collection of low or high values). As pointed out by Achlioptas and McSherry [1], through the low-rank matrix approximation, the *particular trends* or the *meaningful* dimensions of the high-dimensional data implicate that the overall structure inherent can be easily discovered. This is the second "blessing of dimensionality" stated by Donoho [5]. Latent semantic indexing [14] to understand text data, the success of HITS [11] and PageRank [13] algorithms to understand the huge WWW graph adjacency matrix, and recent greedy matrix approximation for machine learning [17] reveal this implication. Among these techniques, 1-rank matrix approximation is essential for analyzing the high-dimensional data [12, 11, 13]. One of the efficient techniques for getting the 1-rank matrix is to employ the discrete dynamical system to quickly converge to the local optima, which has been widely used and studied [12, 20, 7, 11].

We followed the framework of the resonance model introduced in our previous work of visually analyzing the high-dimensional data [12]. We generalized it as a novel discrete dynamical system, which is particularly designed for approximating the gene expression matrix with the multiple classes, i.e., `GMA-1`. In nature, it is a reinforcement mechanism simulating the resonance phenomenon. Due to the quick convergence and efficient matrix-vector multiplications, `GMA-1` is quite efficient. As a filter method, `GMA-1` provides the simultaneous ranking of genes and samples [c]. By rearranging the gene expression matrix with `GMA-1` rankings, we can visually observe the overall distribution (see Fig.4) of the values, where top genes are differentially expressed across

[a] In MAQC project, the description of the data sets are available at `http://www.fda.gov/nctr/science/centers/toxicoinformatics/maqc`.

[b] The 1-rank matrix is a matrix whose rank is 1. It is formally expressed as a multiplication $\mathbf{x}\mathbf{y}^T$ of two vectors $\mathbf{x}$ and $\mathbf{y}$. Therefore, a particular overall structure of the matrix can be approximated and observed by the tendency of $\mathbf{x}\mathbf{y}^T$. The GMA follows the framework of the traditional 1-rank matrix approximation in linear algebra and generalizes it by partitioning the matrix.

[c] The samples are ranked within each class.

classes and top samples are important to the class. Therefore, `GMA-1` can satisfy the biomarker discovery. Moreover, the sorted matrix according to the ranking of both genes and samples can be visually shown to the user for further analysis. Furthermore, if the user needs to refine the biomarker for obtaining the compact biomarker, `GMA-2` can be employed to remove redundant genes. We followed the idea of Jaeger *et al.* [9] by using the representative of the dense cluster in the gene correlation matrix of the biomarker to reduce the redundancy. As observed and proved in `GMA-2`, it is able to find clusters with fixed density. Different from a general clustering algorithm used by Jaeger, `GMA-2` is particularly customized for finding clusters with the fixed density. Therefore, `CBioMarker` combining `GMA-1` and `GMA-2` yields higher accuracy.

2. BASIC RESONANCE MODEL FOR APPROXIMATING MATRIX

In this section, we firstly introduce the resonance model for the purpose of revealing the terrain of the high-dimensional data set. Then the underlying rationale of the resonance model, i.e., the 1-rank matrix approximation, is explained by two theorems. Through the expatiation of the basic mechanism for approximating the matrix in this section, a generalized matrix approximation for the task of biomarker discovery shall be introduced in the next section.

2.1. Process of Resonance Model

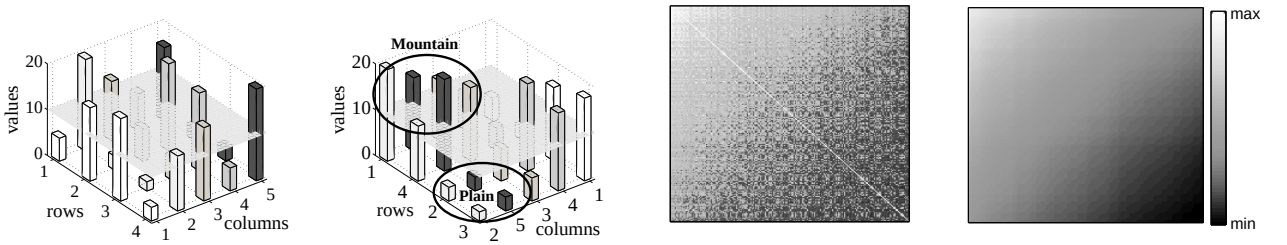
This resonance model has been introduced in [12] for visually analyzing the high-dimensional data set. The target is to rearrange the matrix for collecting the large values to the left-top corner of the sorted matrix (called "mountain"), while leaving the small values to the right-bottom corner (called "plains"). In this way, the data terrain (showing where the "mountains" and "plains" are) can be used to visually analyze the high-dimensional data sets. Fig.1(a) and (b) clearly indicates how this process can be used to visually analyze the matrix through the comparison before and after the rearranging process. In a real-world example, from a yeast gene correlation data [19] in Fig.1(c), we multiplied the ranking value vectors of rows and columns, we can get Fig.1(d). This phenomenon implies that, the resonance model indirectly does the work of the 1-rank matrix approximation by using the matrix in Fig.1(d) to approximate the real matrix in Fig.1(c). Through this matrix approximation process, the underlying dominant terrain and structure is revealed.

In nature, the resonance model is an iterative reinforcement learning process of the matrix. It simulates the resonance phenomenon by introducing a forcing object $\tilde{o}$, such that when an appropriate response function $\mathtt{r}$ is applied, $\tilde{o}$ will resonate to elicit those objects $\{o_i, \dots\} \subset \mathcal{O}$, whose "natural frequency" is similar to $\tilde{o}$. This "natural frequency" represents the *makeup* of both $\tilde{o}$ and the objects $\{o_i, \dots\}$ who resonated with $\tilde{o}$ when $\mathtt{r}$ was applied. Through the iterative reinforcement process, the "frequency" of the forcing object $\tilde{o}$ and the ranking values of the objects $o_i \in \mathcal{O}$ are updated and converged until $\tilde{o}$ is similar to those objects with the largest ranking values. In this way, the "frequency" vector $\tilde{\mathbf{o}}$ of $\tilde{o}$ and the ranking value vector $\mathbf{r}$ of the object set can approximate the matrix W by the matrix $\mathbf{r}\tilde{\mathbf{o}}^T$, denoted as $\mathbf{r}\tilde{\mathbf{o}}^T \approx W$.

In the context of the weighted bipartite graph $G = (\mathcal{O}, \mathcal{F}, E, W)$ and $W = (w_{ij})_{|\mathcal{O}|\times|\mathcal{F}|}$ [d] where $\mathcal{O}$ and $\mathcal{F}$ are two subsets of vertices, the static 'natural frequency' of $o_i \in \mathcal{O}$ is $\mathbf{o_i} = (w_{i1}, w_{i2}, \dots, w_{i|\mathcal{F}|})$. Likewise, the dynamic 'frequency' of the forcing object $\tilde{o}$ is defined as $\tilde{\mathbf{o}} = (\tilde{w_1}, \tilde{w_2}, \dots, \tilde{w_{|\mathcal{F}|}})$. The components of the graph G are clearly shown in Fig.2(a).

Simply put, if two objects of the same 'natural frequency' resonate, they should have a similar terrain. The evaluation of resonance strength between objects o_i and o_j is given by the response function $\mathtt{r}(\mathbf{o_i}, \mathbf{o_j}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. We defined this function abstractly to support different measures of resonance strength. For example, one existing measure to compare two terrains is the well-known *rearrangement inequality theorem*, where $\mathtt{I}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i y_i$ is maximized when the two positive sequences $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ are or-

[d] In the gene expression data, to make sure W is a non-negative matrix, we scale the values of W to the range of $[0, 1]$ by $\frac{w_{ij} - \min}{\max - \min}$, where min and max can be the minimum and maximum of each rows or of the whole matrix. In the rest of the paper, W is supposed to be a matrix whose range is in $[0, 1]$ if we do not mention.



(a) 3D matrix before sorting (b) 3D matrix after sorting (c) symmetric matrix S (d) approximation matrix $\mathbf{r}^*\tilde{\mathbf{o}}^{*T}$ of S

Fig. 1. Matrix approximation by the basic linear resonance model ($\mathtt{r}$=$\mathtt{c}$=$\mathtt{I}$): (a) and (b), a small example matrix with 4 rows and 5 columns to illustrate how the terrain, i.e., "mountains" and "plains" help analyzing the data in both rows and columns; (c) and (d) symmetric matrix sorted by $\mathbf{r}^*$ and $\tilde{\mathbf{o}}^*$.

dered in the same way (i.e., $x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n$ and $y_1 \geqslant y_2 \geqslant \cdots \geqslant y_n$) and is minimized when they are ordered in the opposite way (i.e., $x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n$ and $y_1 \leqslant y_2 \leqslant \cdots \leqslant y_n$).

Notice if two vectors maximizing $\mathbf{I}(\mathbf{x}, \mathbf{y})$ are put together to form $M = [\mathbf{x}; \mathbf{y}]$ (in MATLAB format), we obtain the terrain with the "mountain" in the left side and the "plain" in the right side. For example, the response function $\mathtt{I}$ is a suitable candidate to characterize the similarity of terrains of two objects. Likewise, $\mathtt{E}(\mathbf{x}, \mathbf{y}) = \exp(\sum_{i=1}^{n} x_i y_i)$ is also an effective response function with the function of magnifying the roles of "mountains".

To find the 'mountains' and 'plains', the forcing object $\tilde{o}$ evaluates the resonance strength of every objects o_i against itself to locate a 'best fit' based on the contour of its terrain. By running this iteratively, those objects that resonated with $\tilde{o}$ are discovered and placed together to form the 'mountains' within the 2-dimensional matrix W. In the same fashion, the 'plains' are discovered by combining those objects that resonated weakly with $\tilde{o}$. This iterative learning process between $\tilde{o}$ and G is outlined below.

Initialization Set up $\tilde{o}$ with a uniform distribution: $\tilde{\mathbf{o}} = (1, 1, \ldots, 1)$; normalize it as $\tilde{\mathbf{o}} = \mathtt{norm}(\tilde{\mathbf{o}})$[e]; then let $k = 0$; and record this as $\tilde{\mathbf{o}}^{(0)} = \tilde{\mathbf{o}}$.

Apply Response Function For each object $o_i \in \mathcal{O}$, compute the resonance strength $\mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_i})$; store the results in a vector $\mathbf{r} = \big(\mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_1}), \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_2}), \ldots, \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o}_{|\mathcal{O}|})\big)$; and then normalize it, i.e., $\mathbf{r} = \mathtt{norm}(\mathbf{r})$.

Adjust Forcing Object Using $\mathbf{r}$ from the previous step, adjust the terrain of $\tilde{o}$ for all $o_i \in \mathcal{O}$. To do this, we define the adjustment function $\mathtt{c}(\mathbf{r}, \mathbf{f_j}) : \mathbb{R}^{|\mathcal{O}|} \times \mathbb{R}^{|\mathcal{O}|} \to \mathbb{R}$, where the weights of the j-th frequency is given in $\mathbf{f_j} = (w_{1j}, w_{2j}, \ldots, w_{|\mathcal{O}|j})$. For each frequency f_j, $\tilde{w_j} = \mathtt{c}(\mathbf{r}, \mathbf{f_j})$ integrates the weights from $\mathbf{f_j}$ into $\tilde{o}$ by evaluating the resonance strength recorded in $\mathbf{r}$. Again, $\mathtt{c}$ is abstract, and can be materialized using the inner product $\mathtt{c}(\mathbf{r}, \mathbf{f_j}) = \mathbf{r} \bullet \mathbf{f_j} = \sum_i w_{ij} \cdot \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_i})$. Finally, we compute $\tilde{\mathbf{o}} = \mathtt{norm}(\tilde{\mathbf{o}})$ and record it as $\tilde{\mathbf{o}}^{(k+1)} = \tilde{\mathbf{o}}$.

Test Convergence Compare $\tilde{\mathbf{o}}^{(k+1)}$ against $\tilde{\mathbf{o}}^{(k)}$. If the result converges, go to the next step; else apply $\mathtt{r}$ on $\mathcal{O}$ again (i.e., forcing resonance), and then adjust $\tilde{o}$.

Matrix Rearrangement Sort the objects $o_i \in \mathcal{O}$ by the coordinates of $\mathbf{r}$ in descending order; and sort the frequencies $f_i \in \mathcal{F}$ by the coordinates of $\tilde{\mathbf{o}}$ in descending order.

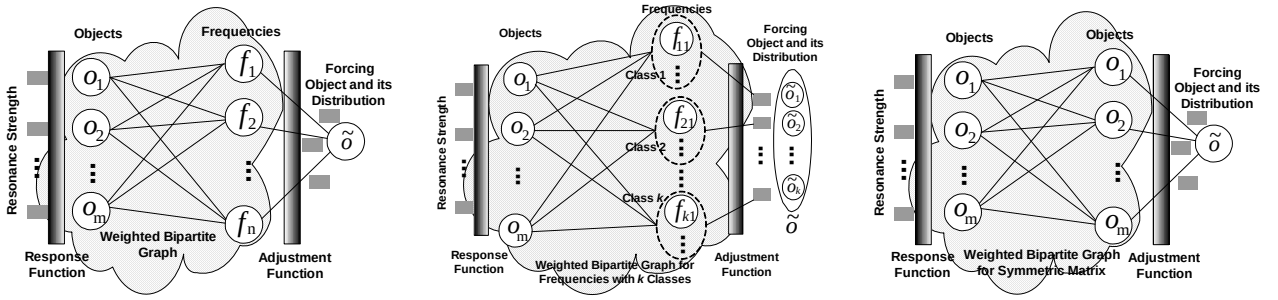
For clearly stating the whole process above, we further express it in the following formulas,

$$\mathbf{r}^{(k+1)} = \mathtt{norm}\big(\mathtt{r}(W\tilde{\mathbf{o}}^{(k)})\big) \quad (1)$$

$$\tilde{\mathbf{o}}^{(k+1)} = \mathtt{norm}\big(\mathtt{c}(W^T\mathbf{r}^{(k+1)})\big) \quad (2)$$

To illustrate how the matrix is sorted, let's take a look at a real-life example from a yeast gene expression data [19]. The symmetric gene correlation matrix is computed by Pearson correlation measure. After the resonance model, we obtained the converged $\mathbf{r}^*$

[e] $\mathtt{norm}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$, where $\|\mathbf{x}\|_2 = (\sum_{i=1}^{n} x_i^2)^{1/2}$ is 2-norm of vector $\mathbf{x} = (x_1, \ldots, x_n)$.



(a) basic resonance model (b) GMA-1: extended resonance model 1 (c) GMA-2: extended resonance model 2

Fig. 2. The resonance models of approximating the matrix for different purposes: (a) collecting the high values into the left-top corner; (b) simultaneously collecting high/low values into the left-top corners of k classes or submatrices W_i^- or W_i^+; (c) collecting the extremely high similarity/correlation values into the left-top corner to form a dense cluster.

and $\tilde{\mathbf{o}}^*$ with the decreasing order, and also sorted $o_i \in \mathcal{O}$ and $f_j \in \mathcal{F}$ accordingly. Certainly, the rows and columns of the matrix S are also rearranged with the same orders of o_i and f_j. The sorted S in this example is shown in Fig.1(c). We also draw its corresponding 1-rank approximation matrix $\mathbf{r}^*\tilde{\mathbf{o}}^{*T}$ in Fig.1(d). This example in Fig.1(c) and (d) illustrates two observations: (1) the function of the resonance model is to collect the large values in the left-top corner of the rearranged matrix and leave the small values to the right-bottom corner; (2) the underlying rationale is to employ the 1-rank matrix $\mathbf{r}^*\tilde{\mathbf{o}}^{*T}$ to approximate S. Actually, it is essential that the value distribution of $\mathbf{r}^*\tilde{\mathbf{o}}^{*T}$ determines how the values of the sorted S are distributed.

3. TWO GENERALIZED MATRIX APPROXIMATIONS BY EXTENDING RESONANCE MODEL FOR GENE SELECTION

In this section, we extend and generalize the basic mechanism of the resonance model in Section 2 for the purpose of the gene selection in two aspects. The first is to rank genes and samples for selecting those differentially expressed genes $\mathcal{G}=\{g_1, \ldots, g_k\}$. The second is to discover those very dense clusters in the correlation matrix computed from $\mathcal{G}$, and remove the redundant genes in $\mathcal{G}$ by only selecting one or two representative genes from each dense cluster. In the two steps, we particularly designed two extended resonance models. From the perspective of the matrix computation, they are two generalized matrix approximation methods based on the basic resonance model.

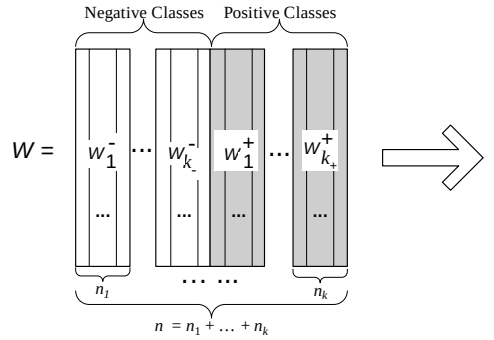
3.1. GMA-1 for Ranking Differentially Expressed Genes

Consider the general case of the gene expression data, suppose the data set consists of m genes and n samples with k classes, whose number of samples are $n_1, \ldots, n_k$ respectively and $n_1 + \ldots + n_k$=n. Without losing the generality, we suppose the first k_- classes are negative, the following k_+ classes are positive, and $k_- + k_+ = k$. Therefore, a general gene-sample matrix $W_{m\times n} = [\underbrace{W_i^-}_{1\leqslant i\leqslant k_-}, \underbrace{W_i^+}_{1\leqslant i\leqslant k_+}]$ is shown with submatrix blocks in Fig.3(a). Because the target of analyzing differentially expressed genes is to find up-regulated or down-regulated genes between negative and positive sample classes, the basic resonance model should be changed, from collecting high values to the left-top corner of W', to:

(1) A series of low values collections in each W_i^- into the left-top corner, and simultaneously a series of high values collections in each W_i^+ into the left-top corner.
(2) Controlling the differences of left-top corners between the negative classes W_i^- and W_i^+.

An example figure of such matrix approximation is illustrated in Fig.4. Therefore, to meet these two goals, we extended the basic resonance model, called GMA-1, according to this task as follows.

(1) Transformation of W: before doing the GMA-1, we need to transform the original gene-sample matrix W to W'. The structure of W is made of



(a) original matrix $W = [\ \underbrace{W_i^-}_{1\leqslant i\leqslant k_-}\ ,\ \underbrace{W_i^+}_{1\leqslant i\leqslant k_+}\ ]$

	up regulation	down regulation
$W_i'^- =$	$1 - W_i^-$	W_i^-
$W_i'^+ =$	W_i^+	$1 - W_i^+$
$W' =$	$[\underbrace{1 - W_i^-}_{1\leqslant i\leqslant k_-},\ \underbrace{W_i^+}_{1\leqslant i\leqslant k_+}\ ]$	$[\ \underbrace{W_i^-}_{1\leqslant i\leqslant k_-}\ ,\underbrace{1 - W_i^+}_{1\leqslant i\leqslant k_+}]$

(b) transformed matrix $W' = [\ \underbrace{W_i'^-}_{1\leqslant i\leqslant k_-}\ ,\ \underbrace{W_i'^+}_{1\leqslant i\leqslant k_+}\ ]$

Fig. 3. Transformation of the matrix W: the transformed matrix W' has the same structure of submatrix blocks as shown in (a), but with different submatrix $W_i'^-$ and $W_i'^+$ as listed in (b).

the submatrix blocks W_i^- and W_i^+ of negative classes and positive classes as shown in Fig.3(a). In the case of finding up-regulated and differentially expressed genes, since we need to collect the low values of W_i^- into the left-top corner, we need to reverse the values of W_i^- so that low values become high and vice versa. In other words, we do the transformation by $W_i'^- = 1 - W_i^-$. In this way, the result of collecting high values of $W_i'^-$ and $W_i'^+$ into their own left-top corners naturally lead to the result of collecting the low values of W_i^- into the left-top corners and the high values of W_i^+ into the left-top corners. This is an essential step to meet the first goal aforementioned. We can also use other reverse functions in stead of the simple $1 - x$ function used in Fig.3(b). Similarly, we can transform W by $W_i'^+ = 1 - W_i^+$ in the case of finding down-regulated and differentially expressed genes.

(2) The k partitions of the forcing object $\tilde{\mathbf{o}}$: an implicit requirement in the first goal is that the relative order of each class (submatrix $W_i'^-$ or $W_i'^+$) should be kept the same after doing GMA-1 and sorting W'. For example, after running our algorithm, it is required that all columns of the submatrix $W_2'^-$ must appear after all columns of $W_1'^-$, although we can change the order of columns or samples within $W_1'^-$ or $W_2'^-$. To satisfy this requirement, we partition the original forcing object's frequency vector $\tilde{\mathbf{o}}$ into k parts corresponding to k classes or submatrices. Specifically, $\tilde{\mathbf{o}} = (\tilde{\mathbf{o}}_\mathbf{1}; \ldots; \tilde{\mathbf{o}}_\mathbf{k})$ [f], where each $\tilde{\mathbf{o}}_\mathbf{i}$ corresponds to a sample class. In the process of GMA-1, we separately normalize each $\tilde{\mathbf{o}}_\mathbf{i}$ and then sum their resonance strength vectors together with the factor α to control the differentiation between the negative and positive classes.

(3) The factor α for controlling the differentiation between the negative and positive classes: the frequency vector of $\tilde{\mathbf{o}}$ is divided into $k = k_- + k_+$ parts, each of which is normalized independently. Therefore, we can control the differentiation between the negative and positive classes, by magnifying the resonance strengths $\mathbf{r}_i^+ = \mathtt{norm}(W_i'^+\tilde{\mathbf{o}}_i)$ of k_+ positive classes, or minifying the frequency subvectors $\mathbf{r}_i^- = \mathtt{norm}(W_i'^-\tilde{\mathbf{o}}_\mathbf{i})$ of k_- negative classes. In formal,

$$\mathbf{r} = \mathtt{norm}\Big(\underbrace{\mathbf{r}_1^- + \ldots + \mathbf{r}_{k_-}^-}_{k_- \text{ negative classes}} \underbrace{+ \alpha\mathbf{r}_1^+ + \ldots + \alpha\mathbf{r}_{k_+}^+}_{k_+ \text{ positive classes}}\Big) \tag{3}$$

where $\alpha \geqslant 1$ and α as a scaling factor is multiplied with the normalized positive classes' resonance strength vectors. With the increasing of α, the proportions of positive classes in the resonance strength vector $\mathbf{r}$ will increase and thus result in the increasingly large differences in the top-left corners between positive and negative classes. In this way, the user can tune α to get a suitable differential contrast of two types of classes.

[f] The concatenation of $k = k_- + k_+$ vectors is expressed in MATLAB format.

To summarize the above changes of the resonance model, we draw the architecture of the GMA-1 in Fig.2(b) and express its process in the following formulas:

$$\begin{aligned}
\mathbf{r}_i^{-(k+1)} &= \mathtt{norm}\big(W_i'^{-}\tilde{\mathbf{o}}_i^{-(k)}\big), \qquad i=1,\ldots,k^- \\
\mathbf{r}_i^{+(k+1)} &= \mathtt{norm}\big(W_i'^{+}\tilde{\mathbf{o}}_i^{+(k)}\big), \qquad i=1,\ldots,k^+ \\
\mathbf{r}^{(k+1)} &= \mathtt{norm}\big(\textstyle\sum_{i=1}^{k^-}\mathbf{r}_i^{-(k+1)} + \alpha\sum_{i=1}^{k^+}\mathbf{r}_i^{+(k+1)}\big) \\
\tilde{\mathbf{o}}_i^{-(k+1)} &= \mathtt{norm}\big((W_i'^{-})^T\mathbf{r}^{(k+1)}\big),\ i=1,\ldots,k^- \\
\tilde{\mathbf{o}}_i^{+(k+1)} &= \mathtt{norm}\big((W_i'^{+})^T\mathbf{r}^{(k+1)}\big),\ i=1,\ldots,k^+
\end{aligned} \tag{4}$$

Algorithm 3.1 (`GMA-1`): Biomarker Discovery.

Input: (1) $W_{m\times n}$, expression matrix from m genes set G and n samples set S;
(2) $(n_1,\ldots,n_k)^T$, sizes of the k sample classes with the submatrix structure as in Fig.3(a).
(3) $(k_-,k_+)^T$, numbers of negative and positive classes.
(4) *regulation option*, down or up;
(5) α, differentiation factor.

Output: (1) $(g_1,\ldots,g_m)$, ranking sequence of m genes;
(2) $(s_1,\ldots,s_n)$, ranking sequence of n samples.

1: preprocess W so that the values of W in [0,1] as following the steps in Subsection 2.1.
2: transform W to W' according to formulas in Fig. 3(b) with the knowledge of the matrix structure given by $(n_1,\ldots,n_k)^T$, and $(k_-,k_+)^T$ and *regulation option*.
3: iteratively run equations in Eqn.(4) to obtain the converged $\mathbf{r}^*$ and $\tilde{\mathbf{o}}_i^*$ $(i=1,2,\ldots,k)$.
4: sort $\mathbf{r}^*$ in decreasing order to get the ranking gene sequence $(g_1,\ldots,g_m)$, and sort each of $\tilde{\mathbf{o}}_1^*,\ldots,\tilde{\mathbf{o}}_k^*$ in decreasing order to get the sorted sample sequence {*comment: Because the positions of all sample classes in W' keep not changing as shown in Fig.3(a), each sorting of $\tilde{\mathbf{o}}_i^*$ can only change the order of samples within the i-th sample class W_i'.*}.

where $\mathbf{r}_i, \mathbf{r}_i^+, \mathbf{r}_i^- \in \mathbb{R}^{m\times 1}$ and $\tilde{\mathbf{o}}_i^- \in \mathbb{R}^{n_i^-\times 1}$, $\tilde{\mathbf{o}}_i^+ \in \mathbb{R}^{n_i^+\times 1}$. Comparing Eqn.(1) and (2) with Eqn.(4), besides using the linear functions $\mathtt{r} = \mathtt{c} = \mathtt{I}$, we partitioned the matrix W' to k submatrix blocks and divided the frequency vector $\tilde{\mathbf{o}}$ into k subvectors. Therefore, two equations in the basic resonance model are expanded to the $(2k+1)$ equations in GMA-1. We also formally summarize it as Algorithm 3.1 `GMA-1` for the biomarker discovery. A real-life example of the overall process in Algorithm `GMA-1` is visually shown in Fig.4.

In practice, `GMA-1` can quickly converge. Considering that `GMA-1` is a generalized resonance model by partitioning the matrix into k submatrices, its computational complexity is the same as the resonance model on the whole matrix, i.e., $\mathbf{O}(mn)$.

3.2. GMA-2 for Reducing Redundancy by Finding Dense Clusters

It has been recognized that the top-ranked genes may not be the minimum subset of genes for biomarker and classification [9, 4, 23], because there are correlations among the top-ranked genes, which induces the problem of reducing "redundancy" from the top-ranked gene subsets. One of the effective strategies is to take into account the gene-to-gene correlation and remove redundant genes through pairwise correlation analysis among genes [9, 4, 21]. In this section, we proposed to use the GMA-2, an special instance of the basic resonance model to reduce the redundancy of the top-ranked genes selected by GMA-1. The GMA-2 is a clustering method to find the high-density clusters. Then we can simply select one or more representative genes from each cluster and therefore reduce the redundancy. The underlying rationale is "*members of a very homogeneous and dense cluster are highly correlated and with more redundancy; while a heterogeneous and loose cluster means bigger variety in genes*". Although similar work has been done by Jaeger *et al.* [9], the authors used the fuzzy clustering algorithm which is not a suitable algorithm to control the density of the clusters. Comparing with the fuzzy clustering algorithm, the GMA-2 can not only find clusters with different densities, but also provide the membership degree for a cluster for each gene.

Given a pairwise correlation or similarity matrix of a set of genes [g], the GMA-2 outputs the largest cluster with the fixed density. To find more clusters with the fixed density, the GMA-2 can be iteratively run on the remaining matrix by removing rows and columns of the genes in clusters already found. Unlike the GMA-1 which is a generalization of the basic resonance model, the GMA-2 is actually a special instance of the basic resonance model. Observing Fig.1(c) and (d), the linear basic resonance model is

[g] In our context, this set of genes are the top-ranked m' genes selected by the GMA-1.

able to collect the high values of a symmetric matrix to the left-top corner of the sorted matrix. This means that it can approximate a high-density cluster. Therefore, we customized the basic resonance model to find the dense cluster by setting the response and adjustment functions to be $\mathtt{I}$ or $\mathbf{E}$. When $\mathtt{r} = \mathtt{c} = \mathtt{I}$, we called this linear resonance model as RML; and when $\mathtt{r} = \mathtt{c} = \mathtt{E}$, this non-linear resonance model is called RME. The overall architecture of RML and RME is illustrated in Fig.2(c). With these settings and $S = S^T$, two equations in the basic resonance model (i.e., Eqn.(1) and (2)) can be combined together by removing $\tilde{\mathbf{o}}$, and therefore RML and RME can be represented by Eqn.(5) and Eqn.(6) respectively as follows,

$$\mathbf{r}^{(k+1)} = \mathtt{norm}\big(S\mathbf{r}^{(k)}\big) \tag{5}$$

$$\mathbf{r}^{(k+1)} = \mathtt{norm}\big(\mathtt{E}(S\mathbf{r}^{(k)})\big) \tag{6}$$

A theoretical analysis is given in the following to show how RML works.

Given a nonnegative gene correlation matrix $S = (s_{ij})_{n\times n} \in \mathbb{R}^{n\times n}$, a nonnegative membership vector $\mathbf{x} = (x_1, \ldots, x_n)^T \in \{0,1\}^{n\times 1}$ is supposed to indicate the membership degree of each gene belonging to the dense and largest cluster, when the values of $\mathbf{x}$ are 0 or 1, $D(\mathbf{x})$ in Eqn.(7) means the density of a cluster formed by those genes whose corresponding x_i is 1.

$$D(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}x_ix_j = \mathbf{x}^T S\mathbf{x} \tag{7}$$

However, there are extensive studies on the problem of finding the densest subgraph [h] which is known as the NP-hard problem [6]. A typical strategy in approximation algorithms is to relax the integer constraints (i.e., $\mathbf{x}$ take the binary values 0 or 1) in $\mathbf{x}$ to the continuous real numbers, e.g., $\mathbf{x} \in [0,1]^{n\times 1}$ and normalize it as $||\mathbf{x}||_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = 1$. In this way, the membership degree $\mathbf{x}$ changes from the binary number to the continuous number. According to the matrix computation theory [8], we have the following theorem,

Theorem 3.1 (Rayleigh-Ritz). *Let $S \in \mathbb{R}^{n\times n}$ be a real symmetric matrix and $\lambda_{max}(S)$ be the largest eigenvalue of S, then we have,*

$$\lambda_{max}(S) = \max_{\mathbf{x}\in\mathbb{R}^n} \frac{\mathbf{x}^T S\mathbf{x}}{||\mathbf{x}||_2} = \max_{||\mathbf{x}||_2=1} \mathbf{x}^T S\mathbf{x} \tag{8}$$

and the eigenvector $\mathbf{x}^*$ corresponding to $\lambda_{max}(S)$ is the solution on which the maximum is attained.

Theorem 3.1 indicates that the first eigenvector $\mathbf{x}^*$ of S is the solution of $D(\mathbf{x})$ and therefore reveals a dense cluster. According to the linear algebra, the iterative running of Eqn.(5) in RML will lead to the convergence of $\mathbf{r}$ to the first eigenvector of S, i.e., $\mathbf{r}^* = \mathbf{x}^*$. Therefore, the RML can reveal the dense cluster. In practice, we found that the non-linear resonance model RME works better than the linear RML by using the exponential function to magnify the roles of high values in the dense cluster. Hence, based on RME, the GMA-2 is formally stated in Algorithm 3.2,

Algorithm 3.2 (`GMA-2`): Find a δ-Dense Cluster

Input: (1) $S_{n\times n}$, a non-negative gene correlation matrix from a set of n genes G;
(2) δ, a fixed density threshold.

Output: $G' = \{g_1, \ldots, g_k\} \in G$, a sequence of k genes which forms a dense cluster.

1: run RME on S to get the converged $\mathbf{r}^*$.
2: sort $\mathbf{r}^*$ in decreasing order and get the sequence of genes $(g_1, \ldots, g_n)$ according to this order. Then set the subset $G_2 = \{g_1, g_2\}$ and $k = 2$.
3: **while** $D(S(G_k)) \leqslant \delta$ **do**
4: $\quad k = k + 1$.
5: $\quad$ set $G_k = \{g_1, \ldots, g_k\}$ as the top k genes.
6: **end while**
7: **if** there is no k satisfying $D(S(G_k)) \leqslant \delta$ **then**
8: $\quad$ **return** $\emptyset$.
9: **end if**
10: **return** G_{k-1}.

4. ALGORITHM FOR COMPACT BIOMARKER DISCOVERY

In some cases, the user is more interested in the biomarker with the minimal genes that can classify the samples. Therefore, in this section, we discover the compact biomarker by combining `GMA-1` and `GMA-2`. We outlined it in Algorithm 4.1, `CBioMarker`.

[h]Considering a nonnegative symmetric matrix is the adjacency matrix of an undirected weighted graph, a dense cluster becomes the dense subgraph in this graph. Therefore, these two problems are equivalent.

Similar to that of the basic resonance model, the computational complexity of GMA-2 is $\mathbf{O}(n^2)$. Therefore, the computational complexity of CBioMarker is at most $\mathbf{O}(mn + hm'^2)$ if considering the size of S in each iteration is always m' (but in fact, S' is always smaller than that of the previous iteration after removing the gene dense cluster already found.), where h is the iteration number in Algorithm CBioMarker depending on the number of dense clusters found in S'. Therefore, Algorithm 4.1 CBioMarker is efficient as well. Our empirical result on the large Leukemia data set with the size 12582×72 in subsection 5.1 shows that it took about 3 seconds in MATLAB environment and Pentium IV PC with 512MB RAM.

Algorithm 4.1 (CBioMarker): Outline of Compact Biomarker Discovery with GMA-1 and GMA-2

Input: (1) $W_{m\times n}$, a gene expression matrix from a set of m genes G;
(2) $(n_1, \ldots, n_k)^T$, sizes of the k sample classes with the submatrix structure as shown in Fig.3(a).
(3) $(k_-, k_+)^T$, numbers of negative and positive classes.
(4) δ, a fixed density threshold of the cluster.

Output: $G' = \{g_1, \ldots, g_q\} \in G$, a subset of q genes which forms a biomarker.

1: run GMA-1 on W' to get the gene ranking sequence $(g_1, \ldots, g_m)$.
2: select the first m' genes from the ranking gene sequence $(g_1, \ldots, g_m)$ and compute its correlation matrix S.
3: set $G' = \{\}$ and $S' = S$.
4: **repeat**
5: run GMA-2 on S' with δ and get the highly correlated gene cluster sequence G''.
6: **if** G'' is not empty **then**
7: select the first representative gene g_1 and add it to G', i.e., $G' = \{G', g_1\}$. {comment: the number of representative genes selected depends on δ. If δ is high, then one representative gene is enough; otherwise, select several more.}
8: **end if**
9: set $G' = G' - G''$ and $S' = S(G')$.
10: **until** G'' is empty {comment: it indicates there are no δ-dense clusters any more.}
11: add the rest of genes that are not clustered and found by GMA-2 to G'.
12: **return** G'.

5. EMPIRICAL STUDY

In this section, we conducted the experiments on two data sets and compared our method with three most popular filter methods, T-statistics (T), Information Gain (IG) and ReliefF [10]. We firstly used the GMA-1 [i], T and IG to rank the genes and compared them over different feature sizes, k=2,4,10,20,50,100,200,500,1000. Each resulting feature subset was used to train an SVM classifier [j] with the linear kernel function. Because of the small number of samples, the Leave-One-Out Cross Validation (LOOCV), a popular performance validation procedure adopted by many researchers, was performed to assess the classification performance. Then for obtaining a minimum biomarker, we ran the CBioMarker to get the compact biomarker and similarly used LOOCV accuracy to evaluate it.

5.1. Leukemia Data

We used the Leukemia gene expression data [2], where besides the classes "ALL" and "AML", a new class "MLL" of samples is identified. It contains 12,582 genes and 72 samples with these 3 sample classes. Therefore, we performed three experiments to test our method by using one class versus the rest of classes as positive versus negative: (1) ALL versus MLL&AML, (2) MLL versus ALL&AML and (3) AML versus ALL&MLL. In each experiment, the gene expression matrix partition for our method is $W = [W_1^+, W_1^-, W_2^-]$ with one positive and two negative classes. In all three experiments, α was set to 2 for GMA-1. The results are shown in Table 1, 2 and 3. As shown in the three tables, our method GMA-1 outperforms the other methods in,

- High Accuracy: in all three experiments, GMA-1 maintains very high accuracies in different k. In the experiment "MLL versus ALL&AML", where the class MLL is hard to distinguish, GMA-1 can still obtain high accuracy even when k is very small.
- Compact biomarker: observing the accuracies of three methods from the small k to the large,

[i] Because GMA-1 can rank genes in terms of up and down regulation respectively, in this experiment of comparing k top-ranking genes, we selected $0.5k$ top-ranking genes in up regulation and $0.5k$ top-ranking genes in down regulation to form k top-ranking genes given by GMA-1.
[j] The SVM*light* was used.

GMA-1 is able to quickly obtain the high accuracy in the very small k, while the methods T and IG require larger k to arrive at the same accuracy (the numbers in bold in three tables show the minimum k each method requires to get the highest accuracy). This means that GMA-1 outperforms the other methods in terms of discovering the compact or minimal biomarker. For example, in Table 1, the top 2 ranking genes are found by GMA-1 and their accuracy is 100%, while the accuracy of the other two methods' top 2 ranking genes are less than 80%. Similar cases also appear in Table 2 and 3.

- Stability: not only do the small amount of selected genes have the higher accuracies than the other methods, but also the the large subset of selected genes maintain the high accuracy. This is a stable property with k increasing, and may be interesting to the biologists when they try to analyze more relevant genes contributing to the diseases. In contrast, the method T is not stable, especially in Table 2 when the samples are hard to distinguish.

Table 1. LOOCV accuracy rate (%) of ALL versus MLL&AML.

k=	*2*	*4*	*10*	*20*	*50*	*100*	*200*	*500*	*1000*
T	79.2	86.1	91.7	93.1	98.6	98.6	98.6	**100**	100
IG	76.4	80.6	95.8	**98.6**	98.6	98.6	98.6	98.6	98.6
RliefF	63.9	86.1	95.8	95.8	98.6	98.6	**100**	98.6	98.6
GMA-1	**100**	100	98.6	100	100	100	100	100	100

Table 2. LOOCV accuracy rate (%) of MLL versus ALL&AML.

k=	*2*	*4*	*10*	*20*	*50*	*100*	*200*	*500*	*1000*
T	69.4	65.2	81.9	80.6	84.7	86.1	**93.1**	90.3	87.5
IG	72.2	88.9	88.9	88.9	**98.6**	98.6	97.2	98.6	97.2
RliefF	72.2	88.9	95.8	94.4	94.4	94.4	97.2	**98.6**	98.6
GMA-1	86.0	88.9	97.2	98.6	**100**	97.2	98.6	98.6	98.6
CBioMarker: find 4 genes with 93.1%									

An important factor, which enables GMA-1 to perform well, is that the matrix approximation has the global searching ability to take into account the value distribution of the whole matrix and multiple classes in macroview way. This is different from the way of individually considering genes, samples, or gene-to-gene. We then intended to obtain the minimal biomarker while keeping a relatively high accuracy (e.g., the accuracy is greater than 90%). There is no need to find the compact biomarker in the experiments except "MLL versus ALL&AML", because GMA-1 already found 2 genes with the accuracy greater than 90%. Therefore, we performed the algorithm CBiomarker with $\delta = 0.7$ for GMA-2 for the second experiment. As shown in Table 2, we found 4 genes with the accuracy 93.1%. This result is better than any other method in Table 2 when $k = 4$.

Table 3. LOOCV accuracy rate (%) of AML versus ALL&MLL.

k=	*2*	*4*	*10*	*20*	*50*	*100*	*200*	*500*	*1000*
T	66.7	77.8	97.2	98.6	**100**	98.6	97.2	97.2	97.2
IG	79.2	76.4	87.5	93.1	**97.2**	97.2	97.2	97.2	97.2
RliefF	86.1	84.7	95.8	94.4	97.2	97.2	97.2	**98.6**	97.2
GMA-1	90.3	91.7	**97.2**	97.2	97.2	97.2	97.2	97.2	97.2

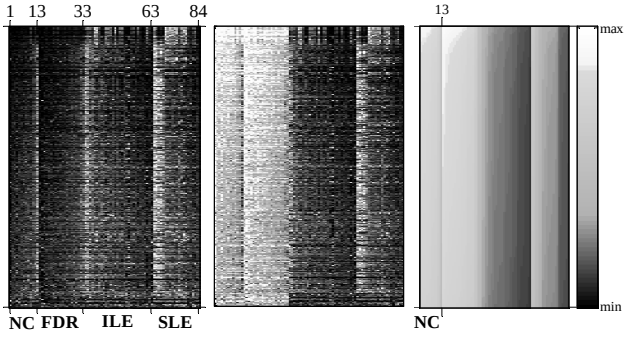
To test if the biomarker found by our methods is effective or not, for instance, we checked two genes found by GMA-1 in Table 1 with Entrez Gene in NCBI Website (http://www.ncbi.nlm.nih.gov/entrez). Two genes are MME which is underexpressed and LGALS1 which is overexpressed [k]. By investigating the result of Armstrong *et al.* [2], these two genes were also ranked as the first genes in the underexpressed and overexpressed genes respectively. MME is a common acute lymphocytic leukemia antigen that is an important cell surface marker in the diagnosis of human acute lymphocytic leukemia (ALL); while LGALS1 was also reported to be highly correlated with ALL [15].

5.2. Lupus Data

In this experiment, we used the unpublished data set taken from the Lupus gene expression experiments of Microarray core facility in UT SouthWestern Medical Center. We demonstrate the visualization ability of our method for facilitating the user to analyze both the genes and samples simultaneously. This data set contains 1,022 genes and 84

[k]The GeneBank No. of MME is J03779 and the GeneBank No. of LGALS1 is AI535946.

samples with 4 sample classes: "NC" (Normal Control), "FDR" (First-Degree Relative), "ILE" (Incomplete Lupus Erythematosus) and "SLE" (Systematic Lupus Erythematosus). Among these classes, "NC" and "FDR" are from the normal persons while "ILE" and "SLE" are from patients.



(a) sorted W (b) sorted W' (c) sorted $\mathbf{r}^*\tilde{\mathbf{o}}^{*T}$

Fig. 4. Visualization of the sorted matrix W, sorted transformation matrix $W' = [1-W_{\mathrm{NC}}, 1-W_{\mathrm{FDR}}, W_{\mathrm{ILE}}, W_{\mathrm{SLE}}]$, and sorted approximation matrix $\mathbf{r}^*\tilde{\mathbf{o}}^{*T} \approx W'$, where $\tilde{\mathbf{o}}^*$ is the concatenation of k vectors: $\tilde{\mathbf{o}}^*=(\tilde{\mathbf{o}}_1^*;\ldots;\tilde{\mathbf{o}}_k^*)$.

We performed `GMA-1` with $\alpha = 5$ on the data. The sorted matrix W with up regulation setting (see Fig.3(b)) is visualized by the grey scale image in Fig.4(a). From this redistribution of the whole matrix, the dominant tendency within each class can be clearly observed. While the most differentially expressed genes (or rows) are placed in the top of W, the low values of the first two classes "NC" and "FDR" are collected to the left-top corner of each submatrix W_{NC} and W_{FDR}, and the high values of the first two classes "ILE" and "SLE" are collected to the left-top corner of each submatrix W_{ILE} and W_{SLE}. In this way, the data within-class distributions and the between-class distribution are fully considered. To illustrate the process of `GMA-1`, we also drew the grey scale image of the transformed matrix W' for up regulation and the final approximation matrix $\mathbf{r}^*\tilde{\mathbf{o}}^{*T}$ given by the converged resonance strength vector $\mathbf{r}^*$ and the frequency distribution vector $\tilde{\mathbf{o}}$.

By observing the grey scale image of approximation matrix $\mathbf{r}^*\tilde{\mathbf{o}}^{*T}$ in Fig.4(c), we found that the outlier samples of each class are put in the rightmost place of the corresponding class submatrix. For example, the colors of the rightmost sample (the 13-th column) in the class "NC" are significantly different from the colors of all other left samples, which indicates that this sample may be an outlier of the class "NC". This can also be observed in Fig.4(a) of the original sorted gene expression matrix. After analyzing this visualization, besides obtaining the top-ranking relevant genes, the user can also draw the conclusion that some normal persons may be early-stage, undetected patients. Similar cases occur in the other classes as well.

6. CONCLUSIONS

In this work, we have introduced a novel perspective of the matrix approximation for filtering the genes in the multiple-class data sets. It comprehensively considers the global between-class data distribution and local within-class data distribution, and therefore improves the accuracy of the biomarker discovery. Meanwhile, it provides an overall tendency of the whole matrix for visualizing and analyzing the data. Experiments on gene expression data have demonstrated its efficiency and effectiveness of both biomarker discovery and visualization.

ACKNOWLEDGMENT

The authors would like to thank Dr. Quan Li from Microarray core facility in UT SouthWestern Medical Center for sharing his unpublished data set with us.

References

1. Achlioptas D, McSherry F. Fast computation of low rank matrix approximations. In *Proc. of STOC* 2001.
2. Armstrong SA, *et. al.* . MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics* 2002; **30(1):** 41–47.
3. Chu W, Ghahramani Z, Falciani F, Wild D. Biomarker discovery in microarray gene expression data with gaussian processes. *Bioinformatics* 2005; **21(16):** 3385–3393.
4. Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. In *Proc. of CSB* 2003.
5. Donoho DL. High-dimensional data analysis: The curses and blessings of dimensionality. In *Math Challenges of the 21st Century* 2000.
6. Feige U, Kortsarz G, Peleg D. The dense k-subgraph problem. *Algorithmica* 2001; **29(3):** 410–421.

7. Gibson D, Kleinberg J, Raghavan P. Clustering categorical data: An approach based on dynamical systems. *VLDB Journal* 2000; **8(3-4):** 222–236.
8. Golub G, Loan CV. *Matrix Computations*. The Johns Hopkins University Press, 1996.
9. Jaeger J, Sengupta R, Ruzzo WL. Improved gene selection for classification of microarrays. In *Proc. of PSB* 2003.
10. Kira K, Rendell L. A practical approach to feature selection. In *Proc. of ICML* 1992.
11. Kleinberg J. Authoritative sources in a hyperlinked environment. *J. of the ACM* 1999; **46(5):** 604–632.
12. Li W, Ong KL, Ng WK. Visual terrain analysis of high-dimensional datasets. In *Proc. of PKDD* 2005.
13. Page L, Brin S, Motwani R, Winograd T. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project. 1998.
14. Papadimitriou CH, Tamaki H, Raghavan P, Vempala S. Latent semantic indexing: A probabilistic analysis. In *Proc. of PODS* 1998.
15. Rozovskaia T, *et. al.* . Expression profiles of acute lymphoblastic and myeloblastic leukemias with all-1 rearrangements. *Proc. of National Academy of Sciences USA* 2003; **100(13):** 7853–7858.
16. Singh D, *et. al.* . Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* 2002; **1:** 203–209.
17. Smola AJ, Schölkopf B. Sparse greedy matrix approximation for machine learning. In *Proc. of ICML* 2000.
18. Tabus I, Astola J. Gene feature selection. Technical report. http://www.cs.tut.fi/∼tabus/course/GSP/Chapter2GSPSR.pdf.
19. Tavazoie S, Hughes J, Campbell M, Cho R, Church G. Yeast micro data set, 2000.
20. Tsaparas P. Using non-linear dynamical systems for web searching and ranking. In *Proc. of PODS* 2004.
21. Xing E, Jordan M, Karp R. Feature selection for high-dimensional genomic microarray data. In *Proc. of ICML* 2001.
22. Yang Y, Pedersen JP. A comparative study on feature selection in text categorization. In *Proc. of ICML* 1997.
23. Yu L, Liu H. Redundancy based feature selection for microarray data. In *Proc. of SIGKDD* 2004; Seattle, Washington.