Interactive Semisupervised Learning for Microarray Analysis

Yijuan Lu, Qi Tian, Feng Liu, Maribel Sanchez, and Yufeng Wang

Abstract—Microarray technology has generated vast amounts of gene expression data with distinct patterns. Based on the premise that genes of correlated functions tend to exhibit similar expression patterns, various machine learning methods have been applied to capture these specific patterns in microarray data. However, the discrepancy between the rich expression profiles and the limited knowledge of gene functions has been a major hurdle to the understanding of cellular networks. To bridge this gap so as to properly comprehend and interpret expression data, we introduce Relevance Feedback to microarray analysis and propose an interactive learning framework to incorporate the expert knowledge into the decision module. In order to find a good learning method and solve two intrinsic problems in microarray data, high dimensionality and small sample size, we also propose a semisupervised learning algorithm: Kernel Discriminant-EM (KDEM). This algorithm efficiently utilizes a large set of unlabeled data to compensate for the insufficiency of a small set of labeled data and it extends the linear algorithm in Discriminant-EM (DEM) to a kernel algorithm to handle nonlinearly separable data in a lower dimensional space. The Relevance Feedback technique and KDEM together construct an efficient and effective interactive semisupervised learning framework for microarray analysis. Extensive experiments on the yeast cell cycle regulation data set and *Plasmodium falciparum* red blood cell cycle data set show the promise of this approach.

Index Terms—Relevance Feedback, semisupervised learning, Kernel DEM, microarray analysis.

1 Introduction

High throughput microarray technology provides tempo-spatial specific expression profiles for thousands of genes simultaneously. Genes that are involved in correlated functions tend to yield similar expression patterns in microarray hybridization experiments. Analyzing these data and learning their expression patterns can therefore reveal the functional association of genes. This raises an important question as to what extent functional information can be revealed from mining expression data. Obviously, there is a gap between the low-level expression data and the high-level functionality: The ultimate goal of microarray analysis is to establish a well-characterized function map of the entire genome, while machine-based analysis can only search for genes that have similar or correlated patterns of expression by data processing. To bridge this gap to facilitate comprehension and interpretation of microarray expression data, we introduce *Relevance Feedback* to microarray analysis.

Relevance Feedback was initially developed in document retrieval [1] and widely applied in content-based image retrieval (CBIR) [2], [3]. The basic idea is to get a human in the loop. At first, computer processing provides initial retrieval results. Users are then asked to evaluate the current retrieval results according to degrees that are relevant or irrelevant to the request. The system then applies the user's feedback to update the training examples to improve performance for the next round. This learning process can be applied iteratively if the user desires. Relevance Feedback algorithms have been shown to provide dramatic performance boosts in image retrieval systems [3]. Although successful in multimedia informational retrieval (e.g., text, image, video), Relevance Feedback has rarely been used in the field of bioinformatics. In this paper, we propose an interactive learning framework based on Relevance Feedback and construct a real-time demonstration system with this learning framework for gene classification and retrieval.

To build an effective learning framework, we must find an efficient learning method that can construct a robust classifier and accurately recognize patterns. To date, many supervised machine learning methods show good performance in gene classification, including Fisher Linear Discriminant Analysis [4], K Nearest Neighbors (KNN) [5], Decision Tree, Multilayer Perceptron [6], and Support Vector Machines (SVM) [7]. In spite of the progress made by these learning methods, two problems still plague efforts to analyze high throughput microarray data: 1) the high dimensionality and 2) the relatively small sample size.

The dimension of the genomic data is usually very high (typically from tens to hundreds) so that machine learning is afflicted by the *curse of dimensionality* as the search space grows exponentially with the dimension. Despite the widely held view that high throughput approaches are overwhelming us with data, the mere fact is that, much of the time, *high dimensionality* obscures the salient details of the data. Moreover, *small sample size* precludes the development of solidly supported conclusions. Pure machine learning methods such as SVM cannot give stable or

- *Y. Lu and Q. Tian are with the Department of Computer Science, University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249-1644. E-mail: {lyijuan, qitian}@cs.utsa.edu.*
- *F. Liu is with the Department of Pharmacology, University of Texas Health Science Center at San Antonio, 7703 Floyd Curl Drive, San Antonio, TX 78229. E-mail: liuf@uthscsa.edu.*
- *M. Sanchez and Y. Wang are with the Department of Biology, University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249. E-mail: maribel.sanchez@gmail.edu, yufeng.wang@utsa.edu.*

Manuscript received 2 Mar. 2006; revised 31 Aug. 2006; accepted 12 Oct. 2006; published online 12 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI0041-0306.
Digital Object Identifier no. 10.1109/TCBB.2007.070206.

meaningful results with a small sample size [3]. Therefore, an approach that is relatively unaffected by these problems will allow us to get more useful results from these data.

Discriminant-EM (DEM) [8] is a *semisupervised* learning algorithm proposed for such a purpose. DEM solves the small sample size problem by taking a hybrid of labeled and unlabeled data to train the classifier. It assumes that only a fraction of the data is labeled with "ground truth," but still takes advantage of the entire data set to generate a good classifier. This learning paradigm can be viewed as an integration of supervised learning and unsupervised learning. Related work on semisupervised learning can be referenced in [9], [10], [11], [12]. In addition, DEM solves the high-dimensionality problem by linear discriminant analysis. It tries to find a mapping such that the data are clustered in the reduced feature space in which the probabilistic structure can be simplified and captured by simpler model assumptions, e.g., Gaussian or Gaussian mixtures. However, since the discriminating step is linear, it is difficult for DEM to handle nonlinearly separable data.

In this paper, we extend the linear algorithm in DEM to use a nonlinear kernel and produce a generalized Kernel Discriminant-EM algorithm (KDEM). KDEM transforms the original data space, X, to a higher dimensional kernel "*feature space*," F, then projects the transformed data to a lower dimensional discriminating subspace such that nonlinear discriminating features can be identified, allowing for a better classification in a nonlinear feature subspace.

Moreover, we combine Relevance Feedback and KDEM together and construct an efficient and effective semisupervised learning framework for microarray analysis. Extensive experiments on the yeast cell cycle regulation data set and *Plasmodium falciparum* red blood cell cycle data set show the promising performance of this approach.

The rest of the paper is organized as follows: In Section 2, we illustrate the kernel DEM algorithm in detail. In Section 3, Relevance Feedback is introduced and discussed. In Section 4, we apply KDEM to gene classification and implement an efficient interactive system with Relevance Feedback for gene classification and retrieval. Finally, conclusions and future work are discussed in Section 5.

2 Kernel Discriminant-EM Algorithm

2.1 Linear Discriminant Analysis

Multiple Discriminant Analysis (MDA) is a traditional linear multiclass discriminant analysis that helps to find a direction, W, for efficient discrimination. After projecting W onto this direction, data can be well separated in the reduced feature space.

Fig. 1 shows a simple example of projecting data from two dimensions onto a line. Of course, if projecting data onto an arbitrary line, it usually produces a confused mixture of samples from all classes and, thus, produces poor recognition performance. However, by moving the line around, we might be able to find an orientation for which the projected samples are well separated. This is exactly the goal of classical discriminant analysis [13].

MDA finds the optimal W and separates samples by attempting to maximize the separability of class centers

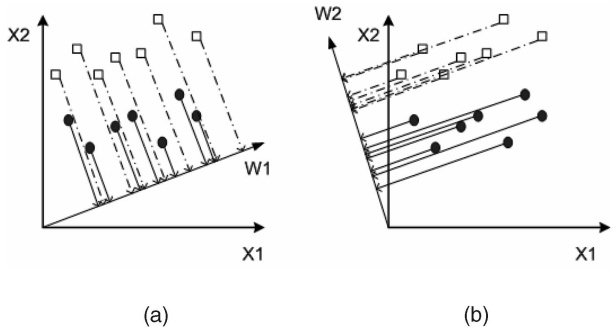

Fig. 1. The same sets of samples are projected onto two different lines in the direction marked W. W_1 is an arbitrary line. W_2 calculated by MDA (on the right) shows greater separation between the square and circle projected points.

(between-class variance, S_B) and minimize the variance of the samples within the same class (within-class variance, S_W). Therefore, the goal is to maximize the ratio of (1):

$$W = \arg\max_W = \frac{|W^T S_B W|}{|W^T S_W W|}, \tag{1}$$

$$S_B = \sum_{j=1}^{C} N_j \cdot (\boldsymbol{m}_j - \boldsymbol{m})(\boldsymbol{m}_j - \boldsymbol{m})^T, \tag{2}$$

$$S_W = \sum_{j=1}^{C} \sum_{i=1}^{N_j} (\boldsymbol{x}_i^{(j)} - \boldsymbol{m}_j)(\boldsymbol{x}_i^{(j)} - \boldsymbol{m}_j)^T. \tag{3}$$

Here, W denotes the transformation matrix, which contains weight vectors of a linear feature extractor, i.e., for a sample $\boldsymbol{x}$, the feature is given by the projections $(W^T \boldsymbol{x})$. Between-class variance, S_B, measures the separability of class centers and within-class variance, S_W, measures the separability of class centers and samples within that class. C is the number of classes, N_j is the number of the samples of the jth class, $\boldsymbol{x}_i^{(j)}$ is the ith sample from the jth class, $\boldsymbol{m}_j$ is mean vector of the jth class, and $\boldsymbol{m}$ is the grand mean of all examples. It should be noted that transformation matrix $W = [w_1, w_2, \cdots w_{d2}]$ maps the original d_1-dimensional data space X to a d_2-dimensional space Δ $(d_2 \leq C - 1)$.

It is obvious that the discrimination step in MDA is linear. If the components of the data distribution are mixed up, it is very unlikely to find a good linear mapping. Hence, MDA has an obvious drawback in handling data that are not linearly separable.

2.2 Kernel Discriminant Analysis

To take into account nonlinearity in the data, we proposed a kernel-based approach. The original MDA algorithm is applied in a kernel feature space F. Via a nonlinear mapping $\phi : \boldsymbol{x} \rightarrow \phi(\boldsymbol{x})$, the data $\boldsymbol{x} \in R^N$ is mapped into a potentially much higher dimensional feature space F, where a simple classification is to be found [14].

This idea can be easily understood with the famous nonlinearly separable data example-XOR (Fig. 2). In the original two-dimensional input space, a rather complicated nonlinear decision surface is necessary to separate the

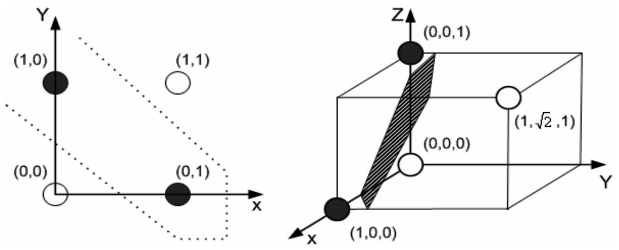

Fig. 2. Nonlinearly separable XOR example: In the original input space, this construction corresponds to a nonlinear decision boundary. Using the second-order monomials $x_1^2, \sqrt{2}x_1x_2, x_2^2$ as features, a separation in feature space can be found using a linear hyperplane.

classes, whereas, in a feature space of second-order monomials $\phi : (x_1, x_2)^T \rightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T$, all one needs for separation is a linear hyperplane [14].

It should be noted that if Δ is the average distance between features of objects in the d-dimensional space, the distance between objects is of the order $\Delta * \sqrt{d}$ (euclidean distance definition). Therefore, in order to separate data well (i.e., enlarge the distance between objects), the number of components in $\phi(\boldsymbol{x}) : d$ is necessarily very large or even infinite. However, this mapping is too expensive and will not be carried out explicitly.

Fortunately, for certain feature spaces and their corresponding mapping, there is a highly effective trick for computing scalar products in features spaces using kernel functions [14]. Let us come back to the XOR example. Here, the computation of a scalar product between two feature space vectors can be readily reformulated in terms of a kernel function k:

$$\begin{aligned}(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y})) &= \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)\left(y_1^2, \sqrt{2}y_1y_2, y_1^2\right)^T \\ &= \left((x_1, x_2)(y_1, y_2)^T\right)^2 = (\boldsymbol{x} \cdot \boldsymbol{y})^2 = k(\boldsymbol{x}, \boldsymbol{y}).\end{aligned} \tag{4}$$

This finding generalizes: For $x, y \in R^N$ and $d \in N$, the kernel function $k(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y})^d$ computes a scalar product in the space of all products of d vector entries (monomials) of $\boldsymbol{x}$ and $\boldsymbol{y}$ [14]. This is the same idea adopted by the support vector machines [15], kernel PCA [16], and invariant feature extractions [17].

Using the trick of rewriting the MDA formulae with only dot products of the form $\phi_i^T\phi_j$, the reproducing kernel matrix can be substituted into the formulation and the solution, eliminating the need for direct nonlinear transformation. With superscript ϕ denoting quantities in the new space, we have the objective function of kernel MDA in the following form:

$$W_{opt} = \arg\ \max_W \frac{|W^T S_B^\phi W|}{|W^T S_W^\phi W|}, \tag{5}$$

$$S_B^\phi = \sum_{j=1}^{C} N_j \cdot (\boldsymbol{m}_j^\phi - \boldsymbol{m}^\phi)(\boldsymbol{m}_j^\phi - \boldsymbol{m}^\phi)^T, \tag{6}$$

$$\mathrm{S}_W^\phi = \sum_{j=1}^{C}\sum_{i=1}^{N_j} (\phi(\boldsymbol{x}_i^{(j)}) - \boldsymbol{m}_j^\phi)(\phi(\boldsymbol{x}_i^{(j)}) - \boldsymbol{m}_j^\phi)^T, \tag{7}$$

with S_B^ϕ and S_W^ϕ being between-class and within-class scatter matrices, $\boldsymbol{m}^\phi = \frac{1}{N}\sum_{k=1}^{N}\phi(\boldsymbol{x}_k)$, $\boldsymbol{m}_j^\phi = \frac{1}{N_j}\sum_{k=1}^{N_j}\phi(\boldsymbol{x}_k)$, where $j = 1, \ldots, C$, and N is the total number of samples.

In general, there is no other way to express the solution $W_{opt} \in F$, either because the dimension of F is too high or because we do not know the actual feature space connected to a certain kernel. However, we know [18], [19] that any column of the solution, W_{opt}, must lie in the span of all training samples in F, i.e., $\boldsymbol{w}_i \in F$. Thus, for some expansion coefficients $\vec{\alpha} = [\alpha_1, \cdots, \alpha_N]^T$,

$$\boldsymbol{w}_i = \sum_{k=1}^{N} \alpha_k \phi(\boldsymbol{x}_k) = \Phi\vec{\alpha}\ \ i = 1, \cdots, N, \tag{8}$$

where $\Phi = [\phi(\boldsymbol{x}_1), \cdots, \phi(\boldsymbol{x}_N)]$. We can therefore project a data point $\boldsymbol{x}_k$ onto one coordinate of the linear subspace of F as follows (we will drop the subscript on $\boldsymbol{w}_j$ in the ensuing):

$$\boldsymbol{w}^T\phi(\boldsymbol{x}_k) = \vec{\alpha}^T\Phi^T\phi(\boldsymbol{x}_k) = \vec{\alpha}^T\begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_k) \\ \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_k)\end{bmatrix}, \tag{9}$$

where we have rewritten dot products, $\phi(\boldsymbol{x})^T\phi(\boldsymbol{y})$ with kernel notation $k(\boldsymbol{x}, \boldsymbol{y})$. Similarly, we can project each of the class means onto an axis of the subspace of feature space F using only products:

$$\begin{aligned}\boldsymbol{w}^T\boldsymbol{m}_j^\phi &= \vec{\alpha}^T\frac{1}{N_j}\sum_{k=1}^{N_j}\begin{bmatrix}\phi(\boldsymbol{x}_1)^T\phi(\boldsymbol{x}_k) \\ \vdots \\ \phi(\boldsymbol{x}_N)^T\phi(\boldsymbol{x}_k)\end{bmatrix} \\ &= \vec{\alpha}^T\begin{bmatrix}\frac{1}{N_j}\sum_{k=1}^{N_j}k(\boldsymbol{x}_1, \boldsymbol{x}_k) \\ \vdots \\ \frac{1}{N_j}\sum_{k=1}^{N_j}k(\boldsymbol{x}_N, \boldsymbol{x}_k)\end{bmatrix} = \vec{\alpha}^T\boldsymbol{\mu}_j.\end{aligned} \tag{10}$$

It follows that

$$\boldsymbol{w}^T S_B\boldsymbol{w} = \vec{\alpha}^T K_B\vec{\alpha} \text{ and } \boldsymbol{w}^T S_W\boldsymbol{w} = \vec{\alpha}^T K_W\vec{\alpha}, \tag{11}$$

where $K_B = \sum_{j=1}^{C} N_j(\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T$ and

$$K_W = \sum_{j=1}^{C}\sum_{k=1}^{N_j}(\zeta_k - \boldsymbol{\mu}_j)(\zeta_k - \boldsymbol{\mu}_j)^T.$$

The goal of kernel multiple discriminant analysis (KMDA) is to find

$$A_{opt} = \arg\ \max_A \frac{|A^T K_B A|}{|A^T K_W A|}, \tag{12}$$

where $A = [\vec{\alpha}_1, \cdots, \vec{\alpha}_{C-1}]$, C is the total number of classes, N is the number of training samples, and K_B and K_W are $N \times N$ matrices which require only kernel computations on the training samples [18].

Now, we can solve for $\vec{\alpha}$s′, the projection of a new pattern z onto w as given by (10). Similarly, algorithms using different matrices for S_B and S_W in (1) are easily obtained along the same lines.

2.3 Discriminant-EM

The DEM algorithm [8] is a semisupervised learning algorithm that was proposed within the transductive learning framework and has been used in content-based image retrieval (CBIR) with Relevance Feedback.

DEM alleviates the small sample size problem by compensating for a small set of labeled data L with a large set of unlabeled data U. Considering these unlabeled data to contain information about the joint distribution over features, DEM uses the Expectation-Maximization (EM) approach to predict the parameters of probabilistic models of whole data distributions with unlabeled data and assign class labels with labeled data.

$$y_i = \arg\ \max_{j=1,\ldots,C} p\big(\boldsymbol{y}_j|\boldsymbol{x}_i, L, U : \forall \boldsymbol{x}_i \in U\big), \qquad (13)$$

where C is the number of classes and y_i is the class label for $\boldsymbol{x}_i$.

The implicit assumption is that labeled and unlabeled data are from the same probabilistic distribution. However, when this assumption is not valid, incorporating unlabeled data could decrease the classification performance [20]. Most of the time this assumption is considered nearly valid for MDA [13] in the reduced feature dimension space.

By combining MDA with the EM framework, DEM learns a classifier simultaneously by inserting a multiclass linear discriminating step in the standard EM iteration loop. Besides, DEM supplies MDA with enough labeled data and applies semisupervised learning techniques in a lower dimensional space projected by discriminant analysis.

A scenario of DEM is as follows: L is the labeled data set, U is the unlabeled data set, and $D = L \cup U$ represents the whole data set. At first, project D to a lower dimensional space Δ by MDA and learn the weak parameters Θ of data distribution with labeled set L. Then, the DEM algorithm iterates over these three steps, Expectation-Discrimination-Maximization, until a stopping criterion is satisfied.

- Expectation: Give each unlabeled sample its probabilistic label l_j and classification confidence w_j based on parameters Θ and the Gaussian mixtures model. After this step, a new weighted data set $D' = L \cup \{\boldsymbol{x}_j, l_j, w_j : \forall \boldsymbol{x}_j \in U\}$ has been obtained.
- Discrimination: Project D' to a new subspace by linear discriminant analysis and produce a new data set

$$\hat{D} = \{W^T\boldsymbol{x}_j, y_j : \forall \boldsymbol{x}_j \in L\} \cup \{W^T\boldsymbol{x}_j, l_j, w_j : \forall \boldsymbol{x}_j \in U\}.$$

- Maximization: Maximize a posteriori probability on $\hat{D}$ and estimate the parameters Θ of the probabilistic models given by the Bayesian classifier.

2.4 Kernel Discriminant-EM

Considering it is difficult for MDA to handle nonlinearly separable data, we apply KMDA in DEM and generalize DEM to Kernel DEM (KDEM) in which, instead of a simple linear transformation to project the data into discriminant subspaces, the data is first projected nonlinearly into a high-dimensional feature space, F, where the data are linearly separated better.

The nonlinear mapping $\phi(\cdot)$ is implicitly determined by the kernel function, which must be determined in advance. The transformation from the original data space $\boldsymbol{X}$ to the discriminating space Δ, which is a linear subspace of the feature space F, is given by $\boldsymbol{w}^T\phi(\cdot)$ implicitly or $\mathbf{A}^T\zeta$ explicitly [18]. A low-dimensional generative model is used to capture the transformed data in Δ.

$$p(\boldsymbol{y}|\Theta) = \sum_{j=1}^{C} p(\boldsymbol{w}^T\phi(\boldsymbol{x})|c_j;\theta_j)p(c_j|\theta_j). \qquad (14)$$

Empirical observations suggest that the transformed data $\boldsymbol{y}$ approximates Gaussian mixtures in Δ. In our current implementation, we use low-order Gaussian mixtures to model the transformed data in Δ. KDEM can be initialized by selecting all labeled data as kernel vectors and by training a weak classifier based on only labeled samples.

Then, the three steps of KDEM are iterated until an appropriate convergence criterion is satisfied:

- E-step: Set $\hat{Z}^{(k+1)} = E[Z|D;\hat{\Theta}^{(k)}]$.
- D-step: Set $A_{opt}^{k+1} = \arg\ \max_A \frac{|A^T K_B A|}{|A^T K_W A|}$ and project a data point $\boldsymbol{x}$ to a linear subspace of feature space F.
- M-Step: Set $\hat{\Theta}^{(k+1)} = \arg\ \max_\Theta p(\Theta|D;\hat{Z}^{(k+1)})$.

The same notation is used in [8]. The E-step gives probabilistic labels to unlabeled data, which are then used by the D-step to separate the data. As mentioned above, this assumes that the class distribution is moderately smooth.

In real application of KDEM, we encounter one problem: While we could avoid working explicitly in the extremely high or infinite-dimensional space F, we are now facing a problem in N variables, a number which, in many practical applications, would not allow the storage or manipulation of $N \times N$ matrices on a computer anymore. Furthermore, solving an eigen-problem of this size is very time consuming $(O(N^3))$. To maximize (12), we need to solve an $N \times N$ eigen or mathematical programming problem, which might be intractable for large N. Approximate solutions could be obtained by sampling representative subsets of the training data $\{\boldsymbol{x}_k|k = 1,\cdots,M, M << N\}$ and using $\xi_k = \big[k(\boldsymbol{x}_1,\boldsymbol{x}_k),\cdots,k\big(\boldsymbol{x}_M,\boldsymbol{x}_k\big)\big]^T$ to take the place of ξ_k. Here, the representative training data are called kernel vectors.

3 Relevance Feedback

3.1 Human in the Loop

Initially developed for document retrieval [1], Relevance Feedback was transformed and introduced into content-based multimedia retrieval, mainly content-based image retrieval (CBIR), during the early to mid 1990s [21], [22], [23]. Interestingly, it appears to have attracted more attention in the image field than the text field—a variety of solutions were proposed within a short period and it remains an active research topic.

A challenge in content-based image retrieval is the *semantic gap* between the high-level semantics in a human mind and the low-level computed features (such as color, texture, and shape). Users seek semantic similarity (e.g., airplane and bird are very similar in terms of low-level features such as shape), but the machine can only measure

similarity by feature processing. To bridge the gap between low-level features and high-level semantics, Relevance Feedback with human in the loop was introduced.

Similar problems exist in microarray analysis. The central dogma in molecular biology, as it pertains to genome biology, is that *understanding gene expression will explain cell function and cell pathology*. However, there is a gap between the low-level expression data and its high-level functionality. In the analysis and interpretation of microarray data, people are interested in finding genes according to their function. Consequently, computationally, we can only search for genes that have similar or correlated patterns of expression. Hence, two main problems (gaps) in this challenge persist. First, given recent developments in measuring expression levels (in particular in microarray technology), how can we infer gene expression patterns from expression data? Second, how can we go from expression pattern to function? In other words, how can we define the role of each gene (or sequence of genes) in terms of biological function and subsequently understand how the genome functions as a whole.

To bridge this gap so as to properly comprehend and interpret expression data produced by microarray technology, it is necessary to have a human or specialist in the loop, which asks for interaction between human and machine. The user gives feedback to tell the machine how relevant the current retrieval results are to his/her request. Then, machine applies the user feedback to retrieve more accurate results in the next round. In this iterative way, Relevance Feedback algorithms learn to achieve a dramatically improved performance.

3.2 Variants of Relevance Feedback

The early work in Relevance Feedback focused on heuristic techniques, e.g., feature axis weighting in feature space [23] and tree-structured self-organizing map (TS-SOM) [24]. The intuition is to emphasize those features that best cluster the positive examples and separate the positive from the negative examples. The assumption of feature independence is rather artificial. Learning in Relevance Feedback has been used in a more systematic way in the framework of optimization [25], [26], probabilistic models [27], learning with small samples [28], pattern classification [8], active learning [29], concept learning [30], and genetic algorithms [31]. There are many variants of Relevance Feedback, but, typically, they cover several or all aspects of the following issues: *What is the user looking for? What should the feedback be? How should images be represented? What should we learn and how should we learn it?* For a survey of state-of-the-art Relevance Feedback techniques, see [3].

3.3 Relevance Feedback in Microarray Analysis

Though successful in informational retrieval, Relevance Feedback has rarely been used in the field of bioinformatics. In this work, we introduce Relevance Feedback for microarray analysis and propose an interactive semisupervised learning framework for gene classification. The aim is to bridge the semantic gap between the temporal expressions and the associated functions.

As shown in Fig. 3, a scenario for Relevance Feedback applied in microarray analysis is described as follows:

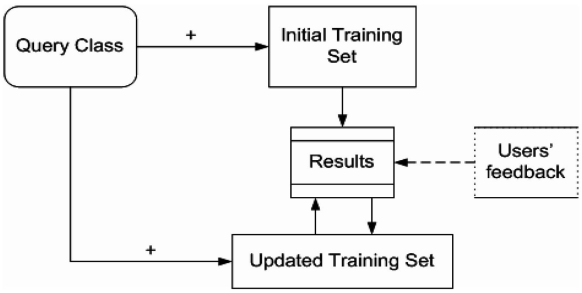

Fig. 3. The scenario of Relevance Feedback applied in microarray analysis.

Step 1. The machine provides the first classification results with the initial training set through query class.

Step 2. Users provide feedback on the classification result as to whether, and to what degree, they belong to that class.

Step 3. The machine updates the training set based on feedback and produces new classification results with the updated training set. Go to Step 2.

Through this procedure, users give feedback based on their knowledge, such as Gene Ontology classification and functional annotation, and retrain our training set to achieve more accurate classification [32].

4 Experiments and Results

4.1 KDEM on Yeast Cell Cycle Regulation Data Set

4.1.1 Data Set

In order to evaluate KDEM on gene expression data, we first chose the yeast (*Saccharomyces cerevisiae*) cell cycle expression data [33] as our benchmark test data set, which contains expression vectors from a total of 80 DNA different microarray hybridization experiments on 6,221 yeast ORFs (open reading frames).

Since the sequencing and functional annotation of the whole *S. cerevisiae* genome have been completed, it serves as an ideal model system and testbed to estimate the accuracy of proposed methods. According to the Comprehensive Yeast Genome Database (CYGD), a repertoire of molecular structures and functional networks in the yeast genome, 4,449 out of a total of 6,221 genes have annotated functions.

The 80 microarray experiments cover a wide spectrum of conditions for cell cycle synchronization and regulations, including α factor-based synchronization, Cdc15-based synchronization, elutriation synchronization, Cln3 and Clb2 experiments, and the conditions under nitrogen deficiency and glucose depletion. The microarray data also include spotted array samples in mitotic cell division cycle, spore morphogenesis, and diauxic shift. It has been shown that combining multiple microarray studies can improve functional classification [34]. This data set has been used in numerous microarray studies and is publicly available at http://rana.lbl.gov/EisenData.htm.

To compare the performance of classification techniques, we focused on five representative functional classes that have been previously analyzed and demonstrated to be learnable by Brown et al. [35] and Mateos et al. [36]. Biologically, they represent categories of genes expected to

TABLE 1
Functional Classes and Distribution of Member Genes Used in Our Evaluation

Class ID	Functional Class	Number of genes
1	TCA Cycle	18
2	Respiration	68
3	Cytoplasmic ribosome	171
4	Proteasome	77
5	Histone/Chromosome	51
6	Other classes	1939
Total		2324

exhibit similar expression profiles [34]. These five classes are shown in Table 1.

Out of the 4,449 annotated yeast genes, those with incomplete expression data were filtered out to assure accurate evaluation. The remaining data set provided 2,324 annotated genes for our comprehensive evaluations. Among these, 385 genes belong to the aforementioned five functional classes and the remaining 1,939 genes have other functions. The distributions of the 2,324 genes in each functional class based on the CYGD annotation are given in Table 1.

4.1.2 Experiments

In a well-cited microarray classification study [35], the use of SVM, two decision tree learners (C4.5 and MOC1), and Parzen windows, etc., has been investigated for gene classification within the same data set. That study showed congruent results: SVM, especially SVM with kernel functions, significantly outperformed the other algorithms for the functional classification. Therefore, we focused on the comparison of KDEM with SVM using the same polynomial and radial basis kernel (RBF) functions. In our experiments, the polynomial kernel functions were $K(\boldsymbol{X},\boldsymbol{Y}) = (\boldsymbol{X}^*\boldsymbol{Y}+1)^d$, with $d = 1,2,3,4$ and the RBF functions used were $K(\boldsymbol{X},\boldsymbol{Y}) = \exp(-||\boldsymbol{X}-\boldsymbol{Y}||^2/2\alpha^2)$. In this work, α was set to be a commonly used value, the median of the Euclidean distances from each positive example to the nearest negative example [35].

By examining how well the classifier identified the positive and negative examples in the test sets, we measured the performance of each classifier. In order to compare to the SVM, we performed a two-class classification with positive genes from one functional class and negative genes from the remaining classes. It should be noted that our method is not limited to binary classification, as is SVM, since it can classify multiple classes as well. Hence, each gene could be classified in one of the following four ways: *true positive* (TP), *true negative* (TN), *false positive* (FP), and *false negative* (FN), according to the CYGD annotation and classifier results. The yeast gene data set is an imbalanced data set in which the number of negative genes is much larger than the number of positive genes. For example, there are only 18 positive instances of the TCA cycle and 2,306 negatives. In this situation, accuracy and single *precision* are not good evaluation metrics because FN is more important than FP [35]. Thus, we chose to use $f\_measure = 2^*(Recall^*Precision)/(Recall + Precision)$ to measure the overall performance of each classifier, which takes both $Precision$ and $Recall$ factors into account [37]. $Recall$ is a measure of the completeness of the retrieved set and $Precision$ measures the purity of the retrieved set. Usually, a trade-off must be made between these two measures since improving one will sacrifice the other. By definition, $Precision =$ (number of TP instances)/(number of TP + FP predictions) and $Recall =$ (number of TP instances)/(number of TP + FN instances). In the case of imbalanced data where negative instances are dominant, $Recall$ is a more important measure as it focuses more on FN predictions.

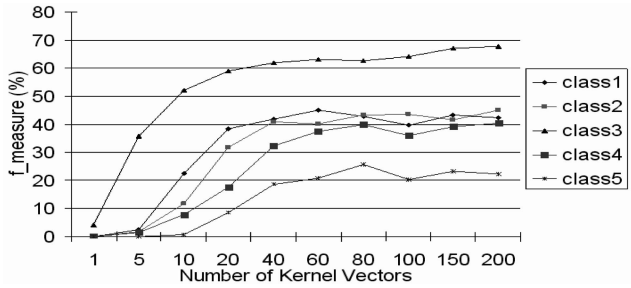

Fig. 4. The average $f\_measure$ for KDEM with the RBF kernel under a varying number of kernel vectors on five classes of yeast data.

The entire ground truth data set includes the expression of 2,324 annotated genes during the cell cycle (Table 1). In our experiments, each method classified the genes in the test set to the above five learnable functional classes and their performance was compared. For each class, we randomly selected 2/3 positive genes and 2/3 negative genes as a training set and used the remaining data for testing classification. This procedure was repeated 100 times. Finally, we obtained the average values of $Recall$, $Precision$, and $f\_measure$ of the 100 rounds.

In our experiments, we use the popular SVM package, SVM Light (version 6.01, 2004), which can be downloaded from http://svmlight.joachims.org/. As mentioned before, parameter α in the RBF kernel was set to be the median of the Euclidean distances from each positive example to the nearest negative example [35]. There is still one parameter to be determined for KDEM using the RBF kernel—the number of kernel vectors. Previously, there has been no good way to choose the number of kernel vectors. In our experiments, we tested KDEM using the RBF kernel under different numbers of kernel vectors and we determined a good number that showed the relative good $f\_measure$ for most classes. Fig. 4 shows the average $f\_measure$ percentage for KDEM with RBF under a varying number of kernel vectors on yeast data. Empirically, 40-200 kernel vectors give good and stable performance for the five classes. Considering overfitting problem, we chose 40 as the kernel setting used in the rest of our experiments.

4.1.3 Results

Table 2 shows the $Precision$, $Recall$, and $f\_measure$ for 11 different classifiers on the five yeast functional classes. The first five methods are SVM using four different polynomial kernels D-p 1 to D-p 4 and the RBF kernel. The sixth is DEM and the seventh to the eleventh are KDEM with the four polynomial kernels and the RBF kernel.

TABLE 2
Comparison of $Precision$, $Recall$, and $f\_measure$ for Various Classifications on the Yeast Cell Cycle Regulation Set

Class	method	SVM(%)					DEM	KDEM(%)				
		D-p 1	D-p 2	D-p 3	D-p 4	RBF	(%)	D-p 1	D-p 2	D-p 3	D-p 4	RBF
TCA Cycle	*Precision*	0.0	**60.56**	65	28.89	3.33	35.52	28.66	16.57	11.45	4.30	33.6
	Recall	0.0	13.33	16.67	5.56	0.56	47.78	45	30.56	27.22	10.56	**59.44**
	f_measure	0.0	21.15	25.64	9.10	0.95	40.22	34.12	20.38	15.72	6.01	**42.44**
Respiration	*Precision*	0.0	71.21	61.64	47.31	**90.17**	44.94	43.84	31.49	22.6	16.95	45.29
	Recall	0.0	20.28	22.22	11.39	11.94	32.64	35.56	27.22	18.89	13.61	**45.83**
	f_measure	0.0	31.13	32.26	17.84	20.68	37.44	38.67	28.81	20.24	14.94	**44.97**
Cytoplasmic ribosome	*Precision*	88.27	89.06	86.12	85.97	**96.28**	69.82	70.13	60.21	57.06	54.29	66.04
	Recall	47.84	46.55	45.85	43.27	45.67	56.55	58.25	55.67	52.11	46.02	**70.12**
	f_measure	61.8	60.89	59.6	57.31	61.72	62.38	63.43	57.74	54.15	49.48	**67.85**
Proteasome	*Precision*	0.0	1.667	0.0	0.83	**72.5**	42.96	44.39	25.09	11.46	6.775	49.44
	Recall	0.0	0.123	0.0	0.12	5.56	15.19	12.35	13.58	7.778	6.173	**34.44**
	f_measure	0.0	0.23	0.0	0.22	10.17	21.94	18.91	17.09	9.206	6.366	**40.25**
Histone/ Chromosone	*Precision*	10	**90.28**	65.29	52.93	86.67	26.07	23.17	18.86	11.51	8.982	27.93
	Recall	0.59	11.57	11.18	8.824	9.02	14.90	14.51	15.88	13.14	9.608	**19.8**
	f_measure	1.11	20.2	18.52	14.66	16.16	18.64	17.57	17	12.09	9.103	**22.18**

From this table, we clearly see that KDEM with the RBF kernel outperformed other methods using $Recall$ or $f\_measure$ as criteria. As discussed earlier, these are more important evaluation factors than $Precision$ for the imbalanced data set.

The SVM failed for most classes with small sample size and yielded very low $f\_measure$. The reason is that, given a small sample size, SVM could not find sufficient labeled data to train classifiers well. By contrast, DEM and KDEM overcame the small sample size problem by incorporating a large number of unlabeled data. Fig. 5 confirms our expectation by showing the declining performance of KDEM, DEM, and SVM on class Histone/Chromosome as the size of training samples drops from 2/3 to 1/5 of the total samples. It is clear that the performance of KDEM and DEM were relatively stable while the performance of SVM declined much faster with smaller training samples.

Not surprisingly, the SVM method showed fairly good performance on its relatively high $Precision$ value, but some exceptions were still observed in the results. For example, among five classes, the D-p 1 SVM achieved zero $precision$ and zero $recall$ for three classes, which means that all of the positive instances recognized were wrong. Even though the higher-dimensional dot product kernel seemed to have better classification, it was hard to tell which dimension d performed the best result.

Compared to DEM, KDEM achieved superior performance in all five classes. This shows that KDEM, when used with good kernel functions, has a better capacity than DEM to separate linearly nonseparable data. For example, for the Proteasome class, the $f\_measure$ of KDEM was 40.25 percent, whereas DEM was only 21.94 percent. Fig. 6 validates this observation by showing typical transformed data sets by linear and discriminant analysis, in a projected 2D subspace of the Cytoplasmic ribosome and Proteasome classes. We find KDEM often projects classes to approximately Gaussian clusters in the transformed spaces which facilitate their modeling with Gaussian or Gaussian mixtures.

The best $f\_measure$ values obtained by SVM, DEM, and KDEM on yeast five functional classes are compared in Fig. 7. This figure clearly shows the superior classification results of KDEM over other methods, thereby demonstrating its promise for classifying microarray gene expression data.

4.2 Validation on *P. falciparum* Microarray Data Set

Previous experiments showed that KDEM performed well on the yeast data set. We then applied KDEM to microarray time series data from another model organism of infectious agents, malaria parasite *P. falciparum*, to predict novel genes with potential functions.

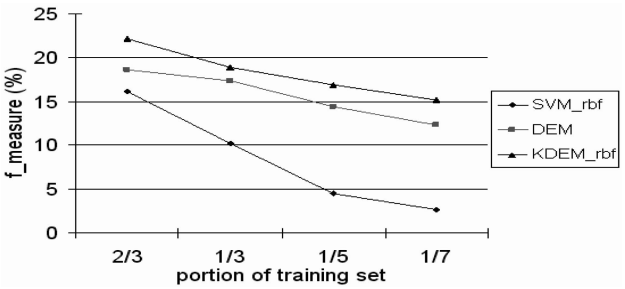

Fig. 5. Comparison of $f\_measure$ for KDEM, DEM, and SVM on the Histone/Chromosome class with different sizes of training set.

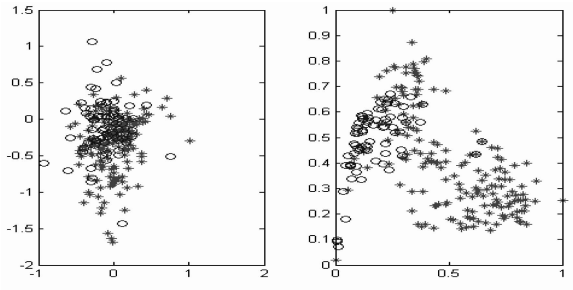

Fig. 6. Data distribution in the projected subspace: (a) DEM and (b) KDEM. Different samples are more separated and clustered in the nonlinear subspace by KDEM (*: class Cytoplasmic ribsome, o: class Proteasome).

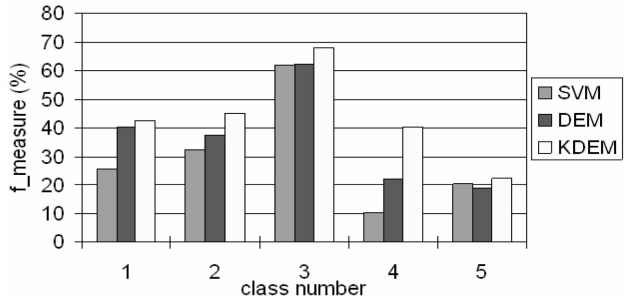

Fig. 7. Comparison of the best $f\_measure$ value obtained by SVM, DEM, and KDEM on yeast five functional classes.

4.2.1 Data Set

Malaria is one of the most devastating infectious diseases, imposing significant health and economic costs in endemic regions. Approximately 500 million cases are reported and about 2 million people die yearly. The causative agent of the most burdensome form of human malaria is a protozoan parasite *Plasmodium falciparum*. The rapid spread of multidrug resistance among these parasites has led to the urgent need for new antimalarial drugs and prevention strategies. The whole genome sequencing of *P. falciparum* predicted over 5,400 genes [38], of which about 60 percent are annotated as "hypothetical" proteins, having insufficient homology to any other functional proteins to allow valid functional assignments. This represents a significant limitation of traditional comparative genomics approach to achieve a systems level understanding of fundamental biology and pathogenesis of the parasite.

The release of genome data made it possible to carry out expression studies and map the results back to the genes. Microarray technology has become a powerful tool in malaria research since it provides a transcriptional profile of parasites at various developmental stages (temporal profiles) and subcellular locations (spatial profiles). A *P. falciparum*-specific DNA microarray using long oligonucleotides (70mers) as representative elements for predicted ORFs in the sequenced genome was developed by the DeRisi lab [39], which established and investigated the expression profiles of *P. falciparum* every hour for the entire durtion of the blood stage (48 hours), the stage when clinical symptoms of malaria occur. The original data is downloadable from http://malaria.ucsf.edu/SupplementalData.php, which includes the profiles of 46 consecutive time points, excluding the 23 hour and 29 hour time points. After standard quality control filtering and normalization, a complete data set consists of signals for 7,091 oligonucleotides corresponding to more than 4,000 Open Reading Frames (ORFs) [38]. Note that the spots with array features that had a sum of median intensities smaller than the local background plus two times the standard deviation of the background were recorded as empty; hence, their $\log_2(Cy5/Cy3)$ value cannot be calculated. In our experiments, we set these empty values as zero because they are very small nonnegative values.

In the original paper [39], 14 functional classes of proteins were shown to exhibit distinct developmental profiles by Fourier Transform. A total of 523 genes belong to these 14 classes, including components involved in genetic information flow (DNA replication, transcription, and translation), metabolic pathways (glycolysis, TCA cycle, ribonucleotide, and deoxynucleotide synthesis), cellular regulatory networks (proteasome), organellar activities (plastid, mitochondria, and organellar translation machinery), and parasite-specific activities (meroziote invasion, actin-myosin motility, and early ring activity) (Table 3).

TABLE 3
Functional Classes and Number of Member Genes Reported in [31]

Group ID	Functional Class	Number of genes
1	Transcription machinery	23
2	Cytoplasmic Translation machinery	159
3	Glycolytic pathway	14
4	Ribonucleotide synthesis	18
5	Deoxynucleotide synthesis	7
6	DNA replication	40
7	TCA cycle	11
8	Proteasome	35
9	Plastid genome	20
10	Merozoite Invasion	87
11	Actin myosin motors	17
12	Early ring transcripts	34
13	Mitochondrial	19
14	Organellar Translation machinery	39
Total		523

4.2.2 Experiments

In our second experiment, we used the classified genes in Table 3 as our ground truth (for classification) and their corresponding 46-hour expressions as their feature representation. Because the number of genes in groups 3 and 7 is too small to train, we combined group 3 with group 7 to form a large group, given that glycolysis and TCA are naturally consequential in metabolic pathways, and combined group 4 with group 5, given that they both represent nucleotide synthetic pathways. Some data with low expressions was also filtered from the data set, leaving a total of 12 groups consisting of 472 genes.

At first, we performed the same two-class classification by SVM, DEM, and KDEM on the *P. falciparum* data to see if KDEM still performed well in this data set. The polynomial and RBF kernels were the same as our first experiment in Section 4.1. Since the malaria data set is also imbalanced, we used $f\_measure$ as the overall performance measure of each classifier. For each class, we randomly selected 2/3 positive genes and 2/3 negative genes as the training set and the remaining data for classification testing. This procedure was repeated 100 times to produce average values of $Precision$, $Recall$, and $f\_measure$ for each class.

Because of the limited space, we only list $f\_measure$ values for these 11 different classifiers on the 12 functional classes in Table 4. From this table, we clearly see that:1) KDEM outperformed SVM for 11 classes out of 12. SVM yielded zero $f\_measure$ on most classes with small size. 2) When the sample size was large, for example, for class cytoplasmic translation, KDEM also performed at least comparably to SVM. 3) KDEM provided good kernel functions and also achieved better performance on most classes other than DEM except for class DNA replication,

TABLE 4
Comparison of $f\_measure$ for Various Classification Method on *P. Falciparum* Data Set

Functional Class	SVM(%)					DEM	KDEM(%)				
	D-p 1	D-p 2	D-p 3	D-p 4	RBF	(%)	D-p 1	D-p 2	D-p 3	D-p 4	RBF
Transcription	0.0	0.0	0.0	0.0	0.0	16	19.5	19	20.4	19.1	**30.0**
Cytoplasmic Translation	82.9	86.1	86.3	86	**87.5**	79.7	16.4	16.7	16.5	20.3	87.2
Glycolysis pathway and TCA cycle	0.0	0.0	0.0	0.0	1.33	17.6	21.3	17.1	19.9	18.6	**35.6**
Nucleotide synthesis	0.0	0.0	0.0	0.0	0.0	22.0	23.0	18.2	20.3	18.4	**23.6**
DNA replication	0.0	0.0	0.0	0.0	17.6	**59.9**	17.2	18.5	21.1	20.8	58.4
Proteasome	0.0	0.0	0.0	0.0	71	28.8	18.9	16.7	19.3	18.4	**87.4**
Plastid genome	0.0	0.0	0.0	0.0	57.9	67.3	21.3	18.5	19.7	20.1	**81.3**
Merozoite invasion	79	77.9	75.1	74.1	84.1	80.7	17.8	15.7	16.9	19.3	**86.5**
Actin myosin motors	0.0	0.0	2.06	7.98	0.0	32.7	20.3	20.2	21.7	19.9	**35.3**
Early ring transcripts	84.9	86	85.8	85.2	91.3	90.6	16.7	17.6	20.2	18.5	**91.4**
Mitochondrial	0.0	0.0	0.0	0.0	0.0	27.3	18.9	16.8	21.4	21.7	**35.5**
Organellar Translation	0.0	0.0	0.0	0.0	0.0	26.2	21.6	16.4	20.9	21.8	**42.4**

which is probably due to the fact that that data is more likely linearly separable. This shows that KDEM, provided with good kernel functions, has a better capacity than DEM to separate linear, nonseparable data.

4.3 Interactive Learning by Relevance Feedback

After validation of our algorithms on a small set of genes (472) with ground truth from *P. falciparum* microarry data set, we applied our semisupervised learning schemes to classify a large amount of unknown genes in the complete data set. Because of the gap between the temporal expressions and the associated functions, we incorporated specialists' feedback to retrain our classifier. We implemented an interactive Relevance Feedback system in which our semisupervised learning using Kernel DEM was a first step for gene classification. Then, we asked specialists to give their opinions on the genes that our classifier was most unsure about. New classification results were obtained after each Relevance Feedback.

In the third experiment, we selected eight classes from Table 3 and used their 48-hour expression as our training data set. The expression profiles of the rest of the genes (3,776) in the complete data set were considered as our testing data set. For the same reason as we mentioned before, we combined the genes in group 4 with group 5 and group 3 with group 7. In total, we have six groups:

1. Transcription machinery,
2. Cytoplasmic Translation machinery,
3. Glycolytic pathway and TCA cycle,
4. Ribonucleotide synthesis and Deoxynucleotide synthesis,
5. DNA replication, and
6. Proteasome.

Fig. 8 shows a screen shot of our interactive semisupervised learning system for *Plasmodium falciparum* gene classification and retrieval. The display area is divided into three panels from top to bottom: setting panel, result panel, and information panel. The top setting panel consists of three parameters: filter number n ($n = 0$, 5, or 10), gene group number (from 1 to 6), and domain field (time domain or frequency domain analysis). Filter number n ($n = 0$, 5, or 10) means to filter such oligonucleotide which contains more than n empty data in the complete data set. Hence, filter 0 has the most strict and clean data compared to filter 5 and filter 10.

Once a gene group is specified for classification, all of the genes in this group are considered positive examples and the genes in the remaining five groups are considered negative examples. The middle result panel displays the classification results. This panel also has two displays, left and right. The left panel displays the *most positive* genes (to the specified gene group) according to their decreasing rankings in terms of membership probability in the complete gene list (of 3,776 genes). The higher the rank, the higher the probability that this gene belongs to the specified gene group is. The right panel displays the *most unsure* (e.g., the ones with probability around 0.5) genes of our classifier, which are ordered by the value of the difference between their probability and 0.5. The smaller the value, the closer they are to the classification boundary. Each gene contains oligo id, gene id, and three radio buttons for specialists' feedback.

The Gene ID button is linked to a Web page: www.plasmodb.org, the *Plasmodium* genome resource, which assists specialists in looking for the relevant information. The specialists can give their feedback by clicking one of the three radio buttons: *positive*, *negative*, and *unsure*. If they think the oligonucleotide belongs to the specified group, they select *positive*. If they think the oligonucleotide belongs to other groups, they select *negative*. If they are not sure, they select *unsure*.

In our leaning framework, both most positive examples and most unsure examples are returned and displayed for user feedback. From the machine learning point of view, the most unsure examples are those examples which lie on or are close to the boundary of the classifier and, thus, are more *informative* than the examples far away from the boundary of the classifier, i.e., most positive examples. The specialists' feedback on these unsure examples will provide the most useful information to retrain our classifier. This is the idea of *active learning* [40].

The bottom information panel displays the information about our classifier, such as the number of genes in the specified group, the size of the training data set, and the

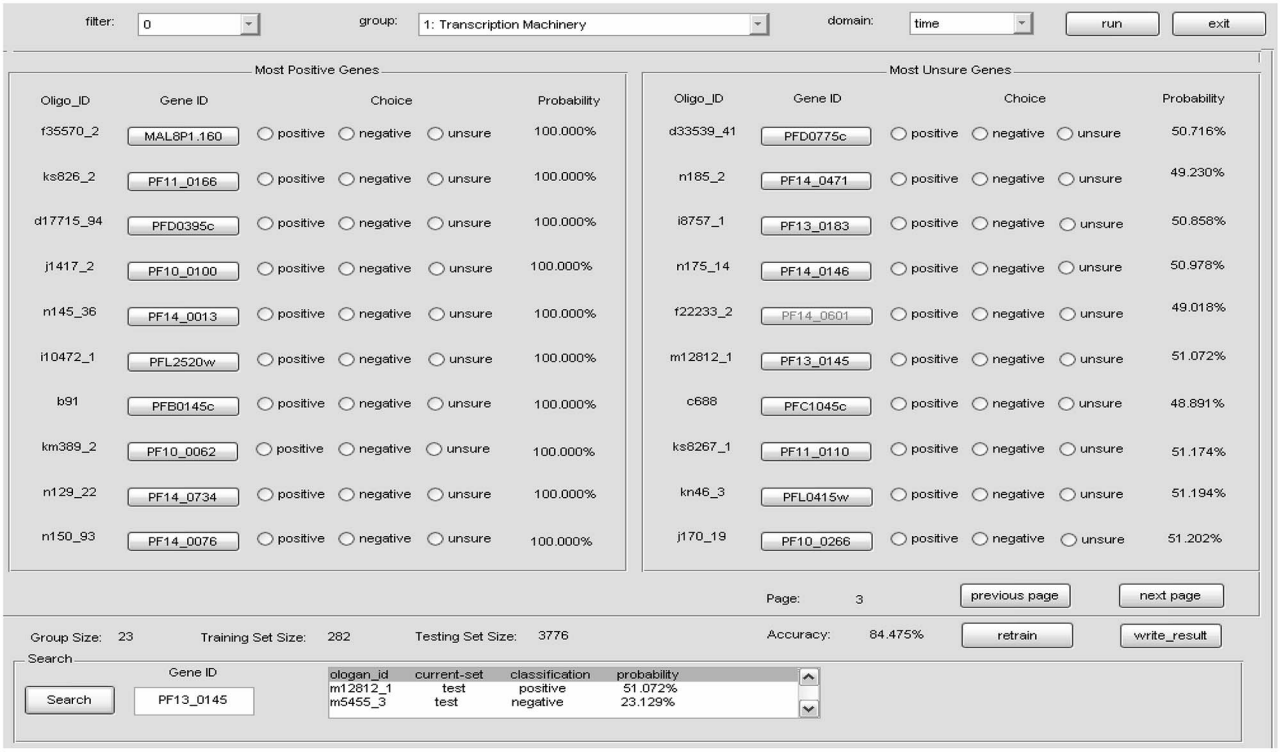

Fig. 8. The interactive semisupervised learning system for gene classification and retrieval.

total number of the testing data set. In this area, users can also use the gene id to search for a particular gene in the complete data set and find its classification result.

Finally, after relevance feedback, we can retrain our classifier with the new information from the specialists. In our experiments, the more feedback from the specialist, the better our classification result is.

4.4 Putative Genes of Specific Functional Classes Identified by KDEM

Using this interactive semisupervised learning system, we applied Kernel DEM as the first step to classify putative malaria genes into specific functional categories based on their distinct developmental profiles across the 48 hour erythrocytic cycle. Table 5 shows several representative genes that were predicted to belong to six functional classes. Their potential functionality is confirmed by independent predictions based on Gene Ontology [41], demonstrating that semisupervised learning is a powerful expression classification method. Such classification could shed light on novel network components and interactions.

In this initial proof of concept study on gene networks, the six selected classes represent different types of biological interactions:

1. Transcription, translation, and DNA replication machineries are complex networks that involve fine regulations of DNA (RNA)-protein and protein-protein interactions. For instance, besides essential enzymes (DNA-directed RNA polymerase complex), transcriptional factors such as Gas41 and Sir2 homolog and transcriptional activators may participate in the regulation of transcription (Table 5). The promoter regions of these regulators are yet to be discovered.
2. Glycolysis/TCA cycle and Nucleotide (DNA or RNA) synthesis exemplify metabolic networks which involve protein-metabolite interactions. For example, the presence of a cascade of coexpressed enzymes, including glucose-6-phosphate isomerase, glycerol-3-phosphate dehydrogenase, pyruvate kinase, lactate dehydrogenase (Table 5), not only suggests that malaria parasite possesses conserved key components in carbohydrate metabolism, but also portrays the various cofactors and metabolites that are involved in the activity of each enzyme.
3. Proteasome is a tightly wrapped complex of threonine proteases and regulatory proteins that mediate protein-protein interactions in cell cycle control and stress response. In previous work [42], we predicted a number threonine proteases and ubiquitin hydrolases, sketching the core elements of malarial proteasome. A concerted regulation pattern revealed by this study is consistent with the postulation of an essential ATP-dependent ubiquitin-proteasome pathway, which was inferred from the results of inhibition assays [43].

4.5 Improved Learning by Relevance Feedback

In addition to the ability to classify novel genes, this interactive semisupervised learning system also offers a powerful means for an annotation feedback. The sequencing of the *P. falciparum* genome was extremely difficult because it is highly AT-rich [38]. Consequently, the gene prediction and annotation based on homology transfer were not error free. Our analysis clearly pinpointed some errors.

TABLE 5
Representative Coexpressed Genes of Specific Functional Classes

Class	Oligo_ID	Gene_ID	Annotation	Prob(%)
Transcription	f22700_1	PFC0805w	DNA-directed RNA pol II	96.4
	opfc0750			77.9
	m44300_14	PF13_0152	sir2 homologue	99.6
	f21506_2	MAL8P1.131	Gas41 homologue	51.3
	M33088_1	MAL13P1.213	transcription activator	98.2
Translation	c430	PFC0635c	TIF E4	89.8
	f26262_1	PF07_0117	eukaryotic TIF2 α	55.4
	j346_4	PF10_0103	eukaryotic TIF2,β	96.3
	j353_17	PF10_0136	Initiation factor 2 subunit	76.9
	ks142_1			92.1
	popfj52810			88.9
	opfI0097	PFL2430c	eukaryotic TIF2b	63.5
	a3310_7	PFA0495c	elongation factor	69.5
	c578	PFC0870w	elongation factor 1	97.9
	f64345_2	PFL1590c	elongation factor g	85.0
	f41218_2	MAL7P1.20	peptide chain release factor	50.9
	opff72453	MAL6P1.210	nascent polypeptide associated complex alpha c	92.8
DNA replication	D17715_47	PFD0475c	replication factor a protein	99.9
	D12635_36	PFD0950w	ran binding protein 1	94.5
	F64125_2	PFE0520c	topoisomerase I	51.2
	F16271_1	PF07_0105	exonuclease I	99.9
	F16210_1	MAL7P1.145	DNA mismatch repair	79.6
	oPFG0045		protein pms1 homologue	95.1
	F57777_1	MAL6P1.125	DNA polymerase epsilon	99.9
Nucleotide synthesis	m38941_10	PF13_0349	diphosphate kinase b	67.2
Glycolysis TCA	j21_14	PF10_0363	pyruvate kinase	89.7
	ks152_12	PF11_0157	glycerol-3-phosphate dehydrogenase (GPDH)	89.7
	L2_270	PFL0780w	GPDH	81.7
	Z_5_70	PF13_0141	L-lactate dehydrogenase	99.9
	Z_5_80			99.2
	Z_5_90			99.9
	m16243_2	PF13_0269	glycerol kinase	99.9
	N132_136	PF14_0341	glucose-6-phosphate isomerase	54.1
	E714_14	PFE0225w	3-methyl-2-oxobutanoate dehydrogenase	97.5
	J158_3	PF10_0218	itrate synthase	97.3
	PFBLOB0009	PF10_0334	succinate dehydrogenase	91.8
Proteasome	D6287_29	PFD0165w	ubiquitin-specific protease	74.6
	D23156_23	PFD0680c	Ubiquitin terminal hydrolase a	99.9
	Z_7_90	MAL8P1.142	proteasome β-subunit	93.3
	oPFL0014	PFL2345c	tat-binding protein homolog	99.9

The classification is based on their expression profiles during the erythrocytic developmental cycle in the malaria parasite.

For instance, two oligonucleotide probes, f23846_3 and opfh0036, both were predicted to correspond to gene PF08_0034; however, these two probes display apparently different developmental profiles: The former is positively classified into Group 1 with probability 60.3 percent, whereas the latter is negative with probability 20.9 percent. This discrepancy is probably due to the error in the gene model. In other words, these two probes may represent two different genes rather than one.

Representative coexpressed genes of specific functional classes. The classification is based on their expression profiles during erythrocytic developmental cycle in malaria parasite.

It is worth emphasizing that this system achieves an improved performance by Relevance Feedback. In our experiments, after a simple trial of correcting four ambiguous training examples (PF14_0601, PF14_0104, PF13_0178, and PFI1020c) based on Gene Ontology predictions, the classification accuracy increases from 84.5 percent to 87.2 percent.

5 Discussions and Conclusions

This paper proposed an interactive semisupervised subspace learning framework for microarray analysis. This framework not only addresses the small sample size and the high dimensionality problem by applying semisupervised learning in an optimal nonlinear discriminant subspace, but also bridges the gap between gene expressions and the associated functions that are fundamental challenges in microarray analysis. The proposed approach is applied for gene classification of yeast cell cycle regulation data and *Plasmodium falciparum* data set. The superior performance proves it is a very promising and efficient approach.

The main contributions of this work are:

1. This paper extends the linear DEM to a nonlinear kernel algorithm, Kernel DEM (KDEM). It is a three-step iteration by inserting kernel discriminant

analysis between E-Step and M-Step in the standard expectation-maximization (EM) algorithm. The proposed algorithm is applied for gene classification on the yeast and *Plasmodium falciparum* data set and compared to the state-of-the-art algorithm SVM with polynomial and RBF kernel functions. KDEM outperforms SVM in the extensive tests.

2. In order to bridge the gap between gene expressions and the associated functions, which is a fundamental challenge in microarray analysis, an interactive learning framework *Relevance Feedback* is also introduced for microarray analysis and a real-time demo system is constructed for gene classification and retrieval. Some unknown genes in the *P. falciparum* data set are identified with the agreement from both gene ontology and the proposed algorithm. The effect of having the same annotation from two independent approaches reduces the uncertainty (or dimensionality) of functional assignment. More important, this system appears to improve learning significantly after a few iterations in Relevance Feedback, which exhibits the advantage of human in the loop very well.
3. The insights provided by semisupervised learning on transcriptomic data into the dynamics of gene networks could shed light on as yet unrecognized network interactions [44].

A significant roadblock on the use of genomic data to better understand infectious diseases is our inability to assign gene functionality. Malaria parasite *Plasmodium falciparum* appears among the most problematic: 60 percent of the open reading frames are annotated as "hypothetical" [44]. Our study may provide an effective means to circumvent this problem. By identifying coexpressed genes in a developmental cycle, it also helps us to identify what could conceivably be network modules. Any network module could contain a range of proteins and regulatory elements [45]. The key components of these modules may have stringent functional constraint and, hence, are conserved across species [46]. Subtracting these known from the modules, the remaining "hypothetical" in transcriptomic maps represent lineage-specific gaps in gene networks. The ability to assign a "hypothetical" gene to a specific network module opens an opportunity toward a tempo-specific functional characterization because, for a parasite with multiple hosts (human and mosquito) and a dynamic life cycle, the "when and where" to initial wet-lab experiments is of critical importance. Some unknown genes in the *P. falciparum* data set are identified with the agreement from both Gene Ontology and the proposed algorithm. The effect of having the same annotation from two independent approaches reduces the uncertainty (or dimensionality) of functional assignment. This network view should allow us to locate choke points in the parasite —potential vulnerabilities that could result in new malaria control strategies.

Our future work includes using both biased and unbiased discriminant analysis in KDEM to better handle the imbalanced data and a hybrid discriminant analysis to incorporate Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for classification. We will also apply this interactive learning framework to cancer classification with gene expression profiles including Leukemia, Colon, Prostate, Lymphoma, Brain, etc.

ACKNOWLEDGMENTS

This work is supported in part by San Antonio Life Science Institute (SALSI) and US Army Research Office grant W911NF-05-1-0404 to Q. Tian and US National Institutes of Health (NIH) 1R21AI067543-01A1, San Antonio Area Foundation, and a University of Texas San Antonio Faculty Research Award to Y. Wang. Y. Wang is also supported by NIH RCMI grant 2G12RR013646-06A1. M. Sanchez is partially funded by NIH/NIGMS MBRS-RISE GM60655. The authors thank Arthur W. Wetzel, Jie Yu, and the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill, 1992.
[2] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool in Interactive Content-Based Image Retrieval," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 8, no. 5, pp. 644-655, Sept. 1998.
[3] X. Zhou and T.S. Huang, "Relevance Feedback in Image Retrieval: A Comprehensive Review," *ACM Multimedia Systems J.,* vol. 8, no. 6, pp. 536-544, 2003.
[4] S. Dudoit, J. Fridlyand, and T.P. Speed, "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data," Technical Report 576, 2000.
[5] L. Li, C.R. Weinberg, T.A. Darden, and L.G. Pedersen, "Gene Selection for Sample Classification Based on Gene Expression Data: Study of Sensitivity to Choice of Parameters of the GA/KNN Method," *Bioinformatics,* vol. 17, no. 12, pp. 1131-1142, 2001.
[6] J. Khan et al., "Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks," *Nature Medicine,* vol. 7, no. 6, pp. 673-679, 2001.
[7] M.P.S. Brown et al., "Knowledge-Based Analysis of Microarray Gene Expression Data by Using Support Vector Machines," *Proc. Nat'l Academy of Sciences USA,* pp. 262-267, 2000.
[8] Y. Wu, Q. Tian, and T.S. Huang, "Discriminant EM Algorithm with Application to Image Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2000.
[9] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. Workshop Computational Learning Theory (COLT),* pp. 92-100, 1998.
[10] O. Chapelle, B. Schölkopf, and A. Zien, *SemiSupervised Learning.* MIT Press, 2006.
[11] T.M. Huang, V. Kecman, and I. Kopriva, *Kernel Based Algorithms for Mining Huge Data Sets, Supervised, Semisupervised and Unsupervised Learning,* p. 96. Springer-Verlag, 2006.
[12] X. Zhu, "Semisupervised Learning Literature Survey," http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf, 2006.
[13] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. John Wiley and Sons, 2001.
[14] K.R. Muller et al., "An Introduction to Kernel-Based Learning Algorithms," *IEEE Trans. Neural Networks,* vol. 12, no. 2, Mar. 2001.
[15] V. Vapnik, *The Nature of Statistical Learning Theory,* second ed., 2000.
[16] B. Schölkopf, A. Smola, and K.R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, pp. 1299-1319, 1998.
[17] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K. Müller, "Fisher Discriminant Analysis with Kernels," *Proc. IEEE Workshop Neural Networks for Signal Processing,* 1999.
[18] B. Schölkopf and A.J. Smola, *Learning with Kernels.* MIT Press, 2002.

[19] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K. Müller, "Constructing Descriptive and Discriminative Nonlinear Features: Rayleigh Coefficients in Kernel Feature Spaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, May 2003.
[20] Q. Tian, J. Yu, Q. Xue, and N. Sebe, "A New Analysis of the Value of Unlabeled Data in Semisupervised Learning for Image Retrieval," *Proc. Int'l Conf. Multimedia and Expo (ICME '04),* June 2004.
[21] T. Kurita and T. Kato, "Learning of Personal Visual Impression for Image Database Systems," *Proc. Int'l Conf. Document Analysis and Recoginition,* 1993.
[22] R.W. Picard, T.P. Minka, and M. Szummer, "Modeling User Subjectivity in Image Libraries," *Proc. Int'l Conf. Image Processing,* 1996.
[23] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool in Interactive Content-Based Image Retrieval," *IEEE Trans Circuits and Systems for Video Technology,* vol. 8, no. 5, pp. 644-655, 1998.
[24] J. Laakaonen, M. Koskela, and E. Oja, "PicSOM: Self-Organizing Maps for Content-Based Image Retrieval," *Proc. IEEE Int'l Conf. Neural Networks,* 1999.
[25] Y. Rui and T.S. Huang, "Optimizing Learning in Image Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 236-243, 2000.
[26] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "MindReader: Query Database through Multiple Examples," *Proc. 24th VLDB Conf.,* 1998.
[27] I.J. Cox, M.L. Miller, T.P. Minka, and T.V. Papsthomas, "The Bayesian Image Retrieval System, Pichunter: Theory, Implementation, and Pychophysical Experiments," *IEEE Trans. Image Processing,* vol. 9, no. 1, pp. 20-37, 2000.
[28] X. Zhou and T.S. Huang, "Small Sample Learning During Multimedia Retrieval Using biasMap," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2001.
[29] S. Tong and E. Chang, "Support Vector Machine Active Learning for Image Retrieval," *Proc. ACM Int'l Conf. Multimedia,* pp. 107-118, 2001.
[30] L. Raton, O. Maron, W.E.L. Grimson, and T. Lozano-Pérez, "A Framework for Learning Query Concepts in Image Classification," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 1999.
[31] Z. Stejic, Y. Takama, and K. Hirota, "Genetic Algorithm-Based Relevance Feedback for Image Retrieval Using Local Similarity Patterns," *Information Processing and Management,* vol. 39, no. 1, pp. 1-23, 2003.
[32] M. Ashburner et al., "Gene Ontology: Tool for the Unification of Biology," *The Gene Ontology Consortium. Nature Genetics,* vol. 25, pp. 25-29, 2000.
[33] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein, "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc. Nat'l Academy of Sciences USA,* vol. 95, no. 25, pp. 14863-14868, 1998.
[34] S. Ng, S. Tan, and V.S. Sundararajan, "On Combining Multiple Microarray Studies for Improved Functional Classification by Whole-Dataset Feature Selection," *Genome Informatics,* vol. 14, pp. 44-53, 2003.
[35] M.P. Brown et al., "Knowledge-Based Analysis of Microarray Gene Expression Data by Using Support Vector Machines," *Proc. Nat'l Academy of Sciences USA,* vol. 97, no. 1, pp. 262-267, 2000.
[36] A. Mateos et al., "Systematic Learning of Gene Functional Classes from DNA Array Expression Data by Using Multiplayer Perceptrons," *Genomes Research,* vol. 12, no. 11, pp. 1703-1715, 2002.
[37] C. van Rijsbergen, *Information Retrieval,* second ed. Butterworths, 1979.
[38] M.J. Gardner et al., "Genome Sequence of the Human Malaria Parasite Plasmodium Falciparum," *Nature,* vol. 419, pp. 498-511, 2002.
[39] Z. Bozdech, M. Llinas, B.L. Pulliam, E.D. Wong, J. Zhu, and J.L. DeRisi, "The Transcriptome of the Intraerythrocytic Development Cycle of Plasmodium Falciparum," *Plos Biology,* vol. 1, no. 1, pp. 1-16, 2003.
[40] S. Tong and E. Chang, "Support Vector Machine Active Learning for Image Retrieval," *Proc. ACM Int'l Conf. Multimedia,* pp. 107-118, Oct. 2001.
[41] The Gene Ontology Consortium, "Gene Ontology: Tool for the Unification of Biology," *Nature Genetics,* vol. 25, pp. 25-29, 2000.
[42] Y. Wu, X. Wang, X. Liu, and Y. Wang, "Data-Mining Approaches Reveal Hidden Families of Proteases in the Genome of Malaria Parasite," *Genome Research,* vol. 13, pp. 601-616, 2003.
[43] S.M. Gantt, J.M. Myung, M.R. Briones, W.D. Li, E.J. Corey, S. Omura, V. Nussenzweig, and P. Sinnis, "Proteasome Inhibitors Block Development of Plasmodium spp.," *Antimicrobial Agents in Chemotherapy,* vol. 42, pp. 2731-2738, 1998.
[44] H. Kitano, "Systems Biology: A Brief Overview," *Science,* vol. 295, pp. 1662-1664, 2002.
[45] P.M. Bowers, S.J. Cokus, D. Eisenberg, and T.O. Yeates, "Use of Logic Relationships to Decipher Protein Network Organization," *Science,* vol. 306, pp. 2246-2249, 2004.
[46] R. Sharan, S. Suthram, R.M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R.M. Karp, and T. Ideker, "Conserved Patterns of Protein Interaction in Multiple Species," *Proc Nat'l Academy of Sciences USA,* vol. 102, pp. 1974-1979, 2005.

Yijuan Lu is a PhD candidate in computer science at the University of Texas at San Antonio (UTSA). She received the bachelor's degree from Anhui University, China, in 2001. From 2001 to 2003, she was a research assistant in the Key Lab of Intelligence Computing and Signal Processing, Chinese Ministry of Education. Since 2003, she has been a research assistant and teaching assistant in the Department of Computer Science at UTSA. In the summer of 2006, she was a summer intern researcher at the Pittsburgh Super Computer Center, Pittsburgh, Pennsylvania. She became a student member of the IEEE in 2005 and she is the recipient of a Student Travel Award from the ACM Multimedia Conference (MM '06). Her current research is concerned with pattern recognition and bioinformatics.

Qi Tian received the PhD degree in electrical and computer engineering in 2002 from the University of Illinois at Urbana-Champaign. He received the MS degree in electrical and computer engineering from Drexel University in 1996 and the BE degree in electronic engineering from Tsinghua University China in 1992, respectively. He is an assistant professor in the Department of Computer Science, the University of Texas at San Antonio (UTSA), and an adjunct assistant professor in the Department of Radiation Oncology, the University of Texas Health Science Center at San Antonio. He was a summer intern (2000, 2001) and visiting researcher (2001) at the Mitsubishi Electric Research Laboratories (MERL), Cambridge, Massachusetts. In the summer of 2003, he was a visiting professor at NEC Laboratories America, Inc., Cupertino, California, in the Video Media Understanding Group. His research interests include multimedia information retrieval, computational systems biology, and pattern recognition. He has published more than 60 refereed book chapters, journal, and conference papers in these fields. His research projects are funded by the US Army Research Office, San Antonio Life Science Institute, the Center of Infrastructure Assurance and Security, and UTSA. He received a Best Student Paper Award with Jie Yu from the IEEE ICASSP 2006. He has served on the international steering committee for the ACM Workshop Multimedia Information Retrieval (MIR) (2006-2009), as conference cochair for the ACM Workshop MIR (2005), SPIE Internet Multimedia Management Systems (2005), and Multimedia Systems and Applications VIII, SPIE's International Symposium on Optics East (2006), publicity chair of the ACM Multimedia (2006) and International Conference of Image and Video Retrieval (2007), and track chair of multimedia content analysis for the IEEE International Conference on Multimedia and Expo (2006). He also served as a session chair and a technical program committee member for a number of conferences, including ICME, ICPR, ICASSP, CIVR, HCI, VCIP, and MIR. He is a guest editor of the *Journal of Computer Vision and Image Understanding* for a special issue on similarity matching in multimedia and computer vision and is on the editorial board of the *Journal of Multimedia*. He is a senior member of the IEEE and a member of the ACM.

Feng Liu received the PhD degree in biochemistry from Iowa State University in 1990 and the BS degree in biochemistry from Wuhan University, China, in 1982. He is a professor in the Department of Pharmacology and Biochemistry, the University of Texas Health Science Center at San Antonio (UTHSCSA). He did his postdoctoral training at Stanford University from 1991 to 1995. One of Dr. Liu's research interests focuses on the insulin signal transduction pathway, which is activated when the hormone insulin binds to its cell surface receptors, resulting in a cascade of biochemical reactions that culminates in regulation of metabolic processes such as glucose uptake and glycogen synthesis. He is also interested in the molecular mechanisms regulating aging. He has published more than 50 refereed journal and conferences papers in these fields. His honors and awards include the CAREER DEVELOPMENT AWARD (1997) from the American Diabetes Association, the Howard Hughes Medical Institute New Faculty Award (1997) from UTHSCSA, and the Lyndon Baines Johnson Research Award (1996) from the American Heart Association, Texas Affiliate. His current projects are supported by two R01 grants from the US National Institutes of Health and one research award from the American Diabetes Association.

Maribel Sanchez received dual BS degrees in biology and computer science from the University of Texas at San Antonio (UTSA) in 2004. From 2000 to 2004, she was a research scientist associate at UTSA. She was a US National Institutes of Health Minority Biomedical Research Support-Research Initiative in Science Enhancement (MBRS-RISE) and Minority Access to Research Careers-Undergraduate Student Training for Academic Research (MARC-U*STAR) fellowship recipient. Currently, she is a systems analyst II in UTSA's Department of Biology. Her current research focuses on bioinformatics comparative genomics with an emphasis in infectious diseases and cell cycle regulation.

Yufeng Wang received the BS degree in genetics from Fudan University, Shanghai, China, the MS degrees in statistics and genetics in 1998, and the PhD degree in bioinformatics and computational biology in 2001 from Iowa State University, Ames. From 2001 to 2003, she was a research scientist at the American Type Culture Collection (ATCC) and an affiliate research assistant professor at George Mason University, Manassas, Virginia. Since 2003, she has been with the University of Texas at San Antonio, where she is an assistant professor with the Department of Biology. She is also an affiliate professor at the South Texas Center for Emerging Infectious Diseases at San Antonio, Texas. Her current research interests include comparative genomics, molecular evolution, and population genetics, with a special emphasis on the evolutionary mechanisms and systems biology of infectious diseases. She is a member of the IEEE.