Visualized Classification of Multiple Sample Types

Li Zhang and Aidong Zhang
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260
lizhang, azhang@cse.buffalo.edu

Murali Ramanathan
Department of Pharmaceutical Sciences
State University of New York at Buffalo
Buffalo, NY 14260
murali@acsu.buffalo.edu

ABSTRACT

The goal of the knowledge discovery and data mining is to extract the useful knowledge from the given data. Visualization enables us to find structures, features, patterns, and relationships in a dataset by presenting the data in various graphical forms with possible interactions. Recently, DNA microarray technology provides a board snapshot of the state of the cell by measuring the expression levels of thousands of genes simultaneously. Such information can thus be used to analyze different samples by the gene expression profiles. Last few years saw many cluster analysis and classsification methods extensively be applied to capture the similarity pattern of gene expressions. A novel interactive visualization approach, VizCluster, was presented and applied to classify samples of two types. It combines the merits of both high dimensional projection scatter plot and parallel coordinate plot, taking advantage of graphical visualization methods to reveal the underlining data patterns. In this paper, we expand VizCluster to classify multiple types of samples. First, we identify genes which are differentially expressed across the sample groups. Then we apply VizCluster to build classifiers based on those genes. Finally, classifiers were evaluated by either hold out or cross validation. Five gene expression data sets were used to illustrate the approach. Experimental performance demonstrated the feasibility and usefulness of this approach.

1. INTRODUCTION

Background

Knowledge of the spectrum of genes expressed at a certain time or under given conditions proves instrumental to understand the working of a living cell. Recently introduced DNA microarray technology allows measuring expression levels for thousands of genes in a single experiment, across different conditions, or over the time. The raw microarray data (images) can then be transformed into gene expression matrices where usually rows represent genes and columns represent samples. The numeric value in each cell characterizes the expression level of the particular gene in a particular sample. Microarray technology has a significant impact on the field of bioinformatics, requiring innovative techniques to efficiently and effectively extract, analysis, and visualize these fast growing data.

Information in gene expression matrices is special in that the sample space and gene space are of very different dimensionality. Typically, there are between $1,000$ to $10,000$ genes comparing with only 10 to 100 samples in a gene expression data set. Furthermore, it can be studied in both sample dimension and gene dimension. Samples are classified by the gene expression patterns while genes can be grouped by the similarity across the samples. By systematically investigating thousands of genes in parallel, microarray technology offers great promise for the study of the classification of different samples based on global gene expression profiles. Last few years saw large amount of literatures addressing this issue [10; 1; 15; 23; 11]. They intended to identify malignant and normal samples, distinguish samples before and after the treatment, or discover subtypes of some disease samples.

Related Work

It is natural to apply clustering techniques to group samples or genes together by their similarities. During recent years, traditional or newly developed clustering (or classification) methods were applied on gene expression data analysis. *Jiang et al.* [14] presented a detailed survey for those methods. Visualization supports finding structures, features, patterns, and relationships in data by presenting the data in various forms with different interactions which enable human involvement and incorporate the perceptivity of humans. Multivariate visualization techniques have been developed rapidly and some visualization tools have also been adapted to perform analysis on microarray data [8; 20; 9].

Most visualizations have been served mainly as graphical presentations of major clustering methods. For instance, *TreeView* [8], provides a computational and graphical environment but the visualization (the dendrogram) is the graphical format of hierarchical clustering output. A novel interactive visualization approach to classifying samples was presented based on the framework of VizCluster [25]. VizCluster uses a nonlinear projection which maps the n-dimensional vectors onto two-dimensional points. This mapping effectively keeps correlation similarity in the original input space. It combines the merits of both scatter plot and parallel coordinate plot, introduces *zip zooming viewing* and *dimension tour* methods to compensate the information lost by the mapping, and offers user interactions. The framework of VizCluster is suitable for microarray data analysis. The scatter plot is suitable for viewing dense data sets with low dimensions in sacrificing the loss of information while the parallel coordinate plot is efficient in displaying low quantity of data with high dimensions at the cost of the presentation clarity. Zip zooming viewing method serves as the bridge between the two and provides a multiresolution information preservation.

Visualized Classification

Our visualized classification model works as follows: first, we identify informative genes (genes which significantly differentially expressed across different sample classes). Then we use VizCluster to build classifiers based on the visual data distribution of different classes. Finally, the classifiers are evaluated by either hold out or cross validation. We started analysis with a 4-dimensional real data set *iris* to illustrate the framework of VizCluster. Our primary objective focused on the classification of samples on gene expression data. We then performed binary classification using two gene expression data sets: *leukemia-A* and *multiple sclerosis*. VizCluster clearly separated two group of samples and in the evaluation process, assigned most unlabelled samples into the correct groups. Next came the multiple classification. Three data sets were analyzed: 3-class *leukemia-B*, 3-class *BRCA*, and 4-class *SRBCT*. In all tasks, the performance was satisfactory.

Contribution of This Paper

In [25], only binary classification, i.e., classifying two sample types was performed. In this paper, we expand VizCluster to classify multiple types of samples. The projection mapping is slightly modified. The way of identifying informative genes is changed from *neighborhood analysis* to *SAM* approach. Dimension arrangement issue is addressed and an algorithm for obtaining a canonical dimension ordering is briefly discussed. Some of the effects of different orderings are given. The strategy of constructing classifiers – straight lines to separate the data class, is discussed. Compared with [25], three new gene expression datasets are analyzed.

The rest of this paper is organized as follows. Section 2 presents the model of visualized classification. In section 3, we show the analyzing results on five gene expression data sets. The last section discusses some issues in this paper.

2. METHODS

Our approach treats both binary classification and multiple classification uniformly. In both cases, we started with informative gene identification, then built classifiers based on those genes and finally performed the evaluation of those classifiers. Figure 1 illustrates the process.

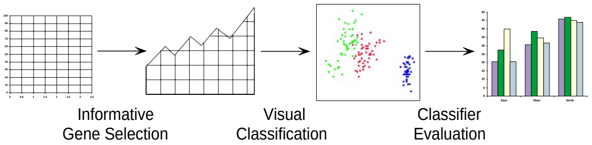

Figure 1: Schematic illustration of visualized classification process. Binary and multiple classification are treated uniformly.

2.1 Identify Informative Genes

Our statistical method of identifying informative genes is a slight variation of *SAM*, Significance Analysis of Microarrays [22; 5]. SAM assigns a score to each gene on the basis of change in gene expression relative to the standard deviation of repeated measurements. For genes with scores greater than an adjustable threshold, SAM uses permutations of the repeated measurements to estimate the percentage of genes identified by chance. When there are only two sample types, standard t test can be used to assess each gene's change over the two conditions. However, with so many genes in each microarray, the control of the false positive rate becomes an important issue. Even for a traditionally acceptable p-value, say 0.01, in a microarray with 5,000 genes would identify about 50 genes by chance. One strategy is to perform permuted t test and calculate the adjusted p-values [22; 7]. in the permutation t test, the standard t statistic was computed. Next, sample labels were randomly permuted and the t statistic for each gene in the permuted data set was computed. Repeat this process 100–10,000 times. Finally, a critical value of the t statistic was determined for each gene based on the empirical distribution of t from permuted data sets for that gene. If the t statistic for a gene in the original labelling of samples was larger than its critical value, the gene is considered as differentially expressed. The permutation F test is similar and is used when there are more than two groups. SAM algorithm is listed in the Appendix.

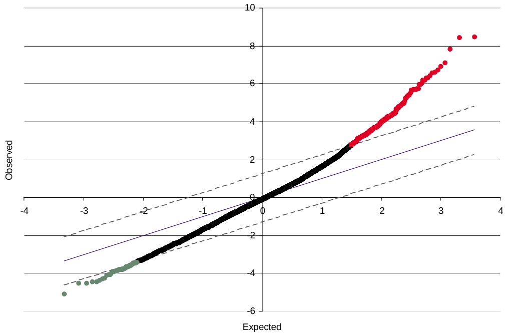

Figure 2: SAM scatter plot of the observed relative difference $d_{(i)}$ versus the expected relative difference $\overline{d}_{(i)}$. The solid line indicates the line for $d_{(i)} = \overline{d}_{(i)}$, while the dotted lines are drawn at a distance $\Delta = 1.27$. In this case, 400 genes are called significant among 7129 genes with *false discovery rate*, $FDR = 1.33\%$. More significant positive genes (red, in the upper left) than negative genes (green, in the lower right) are selected.

There are two main reasons we choose *SAM* over previous used *neighborhood analysis* for identifying informative genes. First, unlike *neighborhood analysis* only working for two-class case, SAM can be applied on both two and multiple classes. Second, SAM is a robust and straightforward method that can be adapted to a broad range of experimental situations and proved to be superior to conventional methods for analyzing microarrays [22]. In practice, SAM is in favor of selecting significant positive genes. See Figure 2. We allowed more significant negative genes to be included when there were overwhelmingly positive genes in the list. In practice, we balanced three factors: (1) the number of significant called genes was between 1% and 5% of total number of genes. (2) $FDR \leq 10\%$. (3) The ratio of positive and negative significant was between 0.2 and 5.0.

2.2 Visualization

The Mapping and Dimension Arrangement

One of the key obstacles in visualizing microarry data is the high dimensionality. VizCluster proposed an interactive visualization framework combining the merits of both high dimensional scatter plot and parallel coordinate plot. A nonlinear projection is used to map the n-dimensional vectors onto two-dimensional image points. This mapping has the property of keeping correlation similarity in the original space [25]. First, a global normalization was performed on the data set to ensure that each dimension has value between 0 and 1. Let P be a n-dimensional data set of m entities and vector $\vec{P_g} = (x_{g1}, x_{g2}, \ldots, x_{gn})$ represent a data entity in the n-dimensional space (also called input space). Formula (1) describes the mapping $\Psi : \mathbb{R}^n \rightarrow \mathbb{C}$, which maps $\vec{P_g}$ onto a point $\vec{Q_g^*}$ in a two-dimensional complex plane $\mathbb{C}$:

$$\vec{Q_g^*} = \Psi(\vec{P_g}) = \sum_{k=0}^{n-1} (\lambda_k * x_{g\,k+1}) e^{i\frac{2\pi}{n}k} \qquad \lambda_k \in [-1, 1] \quad (1)$$

where λ_k (default value is 0.5) is an adjustable weight for each dimension, n is dimensions of the input space, and i is the imaginary unit. Essentially, $\vec{Q_g^*}$ is the vector sum of all its dimensions on n directions.

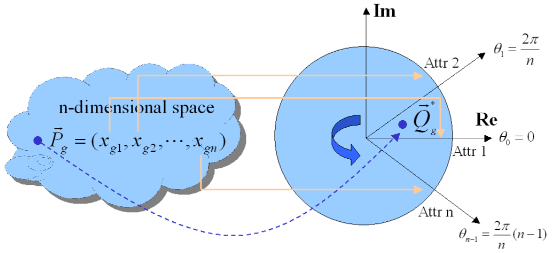
Im
$\theta_1 = \frac{2\pi}{n}$
Attr 2
n-dimensional space
$\vec{Q}^*_g$
Re
$\vec{P}_g = (x_{g1}, x_{g2}, \cdots, x_{gn})$
$\theta_0 = 0$
Attr 1
Attr n
$\theta_{n-1} = \frac{2\pi}{n}(n-1)$

Figure 3: Mapping from n-dimensional input space onto 2-dimensional complex plane. Each dimension of $\vec{P_g}$ is mapped onto an evenly divided direction. The sum of n complex numbers is $\vec{Q_g^*}$, the final image of $\vec{P_g}$.

This non-linear mapping (1) preserves correlation relationship in the input space onto the two-dimensional images. Notice, all data entities having the format of $(a, a, \ldots, a)$ will be mapped to the center (assuming all dimension weights are the same). If $\vec{X}$ and $\vec{Y}$ have the same pattern, i.e., ratios of each pair of dimensions of $\vec{X}$ and $\vec{Y}$ are all equal ($\vec{Y} = \alpha\vec{X}$, α is a scaler), under the mapping, they will be mapped onto a straight line across the center. All vectors with same pattern as $\vec{X}$ and $\vec{Y}$ will be mapped onto that line. Points with similar pattern of $\vec{X}$ or $\vec{Y}$ will be mapped onto a narrow strip region around that line.

In the original VizCluster paper [25], the issue of dimension ordering and arrangement was not addressed. The mapping (1) is affected by the order and arrangement of dimensions. Here, we proposed a canonical dimension ordering. Details will be published in a separate report. The basic idea is to order genes according to their similarity to a predefined sample class pattern which allows the better class separation. The sketchy algorithm is described in Appendix.

Zip Zooming View and Dimension Tour

Since mapping (1) could not preserve all the information in the input space, the scatterplot is a lossy visualization representation. By contract, parallel coordinate plot allows the information of all dimensions to be visualized. In VizCluster a *zip zooming* (parallel coordinate) viewing method was proposed extending circular parallel coordinate plots. Instead of showing all dimensional information, it combines several adjacent dimensions and displays the reduced dimension information. The number of dimensions displayed, called *granularity setting*, can be set by the user. A series of such views would allow user to inspect information at different levels from coarse to fine. Closer look at zip zooming view method reveals that circular parallel coordinate plot and high dimensional scatterplot are the two extreme cases while other granularity settings are in between. Their combination allows a simple and intuitive presentation of the data set and yet preserving all the information at different levels.
Another viewing method in VizCluster is *dimension tour*, an interactive projection-pursuit-guided grand tour [2; 4; 6] like viewing method. By adjusting the coordinate weights of the dataset, data's original static state is changed into dynamic state which may compensate the information loss from the mapping. Each dimension parameter can be adjusted from -1 to 1. The result of parameter adjustment in scatterplot will cause the redistribution (sometimes dramatically) of the 2-dimensional image points. *Dimension tour* is a sequence of either scatterplots or zip zooming views in which each frame has a specific dimension parameter settings.

A Non Gene Expression Example

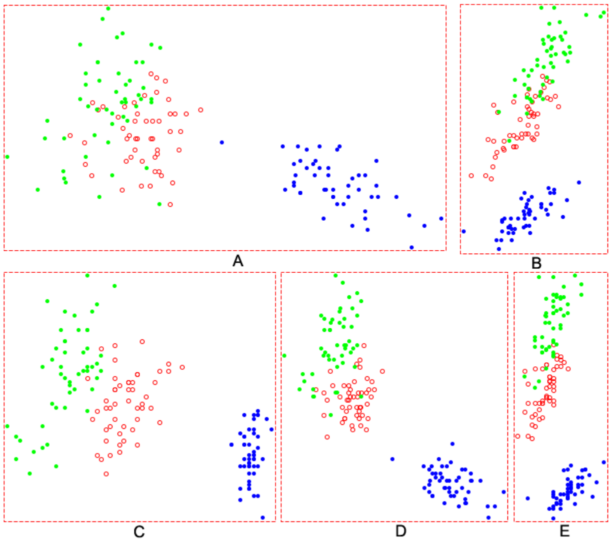

Figure 4: *Iris* data visualized in VizCluster. Color blue was assigned to Setosa species, red to Versicolor, and green to Virginica. (A) through (E) show the scatter plots under different dimension parameter settings.

To illustrate the visualization under VizCluster, we used a 4-dimensional real data set, the famous Fisher's *iris*. The data has 50 plants of each species of iris: *Setosa*, *Versicolor*, and *Viginica*. It contains four attributes, *sepal length*, *sepal* *width*, *petal length*, and *petal width* which are ordered by the above algorithm. The visualization clearly indicates the separation of the three classes.

2.3 Classifier Construction and Evaluation

A classifier should allow the class assignment of newly arrived unlabelled data. The data on which the classifier was built is called *training data* while the unlabelled data for class assignment is called *testing data*. After visualizing training data in VizCluster, we constructs a classifier which consists of straight lines to separate the data based on the visual data distribution. Initially, data are displayed in scatterplot view using default dimension parameter setting. If the current scatterplot does not indicate a clear separation, we adjust dimension parameters either manually or automatically through dimension tour.

Classification points even on 2-dimensional space is a nontrivial issue. We adapted *oblique decision trees* [16] approach. Oblique allows the hyperplanes at each node of the tree to have any orientation in parameter space thus constructs straight lines with arbitrary slope to separate known data classes. Algorithm OC1 [17] was applied to construct oblique decision trees. OC1 combines deterministic hill-climbing with two forms of randomization to find a good oblique split at each node of a decision tree. The overview of the OC1 algorithm for a single node of a decision tree is given in Appendix. In VizCluster, user is allowed to adjust those lines.

Classifier's accuracy is judged by the correctness of its class prediction for the testing data. There are two commonly used methods: *hold out* and *leave one out cross validation*. In hold out method, the data is divided into mutually exclusive training and testing sets, the class prediction errors on the testing data is counted using the classifier built on the training data. However, when data set is small, the separation of training and testing data may result in insufficient training data for constructing the classifier. In this case, cross validation is applied. All but one data entity are used to build the classifier and the last one is withheld as testing data. This process is repeated in a round robbin way, i.e., each data entity is withheld once, and the cumulative errors are counted.

3. RESULTS

3.1 Two Sample Classes

Leukemia-A

We started with binary classification, i.e. samples coming from two classes. Usually this task involves distinguish malignant samples from healthy control, samples before and after some treatment, or two subtypes of tumors. Two gene expression data sets were analyzed: *leukemia-A* and *multiple sclerosis*. The well-known Golub's *leukemia-A* microarray set [10] often serves as *benchmark* [19] for microarray analyzing methods. It contains measurements corresponding to ALL and AML samples from bone marrow and peripheral blood. The data involves 72 leukemia samples of 7129 genes and it has been divided into two groups: training group with 27 ALL and 11 AML samples; testing group of 20 ALL and 14 AML samples. We first selected 50 informative genes, genes which most differentially expressed between ALL and AML samples in the training group. These 50 genes were then used to build a classifier. Next, we performed hold out evaluation on the classifier using the testing group (based on the same 50 genes) and counted the errors. The result was that five samples were misclassified (out of 34), one ALL and four AML. The accuracy was 85%. Most misclassified samples were close to the line of the classifier. Figure 5 shows the classification.

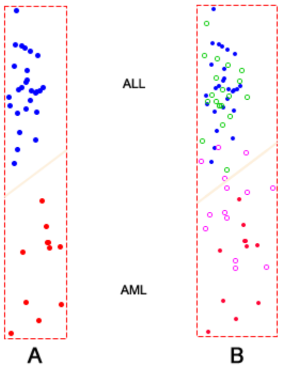

Figure 5: Binary classification of *leukemia-A* data set. (A) A classifier was built using all 27 ALL and 11 AML training samples. Blue was assigned to ALL samples and AML samples were in red. (B) The evaluation of the classifier in (A). Green circles stood for 20 ALL samples and magenta circles standed for 14 AML testing samples. Overall, the classifier failed to predict one ALL and four AML samples.

Multiple Sclerosis

The second experiment was based on gene expression data from a study of multiple sclerosis patients. Multiple sclerosis (MS) is a chronic, relapsing, inflammatory of the brain disease. Interferon-β (IFN-β) has been the most important treatment for the MS disease for the last decade. The data was collected from DNA microarray experiments in the Neurology and Pharmaceutical Sciences departments at State University of New York at Buffalo. It consists of two parts: one contains 28 samples where 14 MS patients are before and 14 are after IFN-treatment, we call it MS_IFN group. The other, MS_CONTROL group, contains 30 samples of which 15 are MS patients and 15 are healthy controls people. There are 4132 genes in each group. The task is to perform two binary classifications and not one 3-class classification. The reason is that MS and IFN are paired groups but MS and Control are not. Figure 6 illustrates the classification.

The two classifiers were build on 88 informative genes and were evaluated by cross validation. (A) A classifier was built using 14 MS and 13 IFN samples. MS samples were colored blue and IFN samples red. (B) Class prediction by this classifier. We used the IFN sample previously held to test the classifier. The green circle (indicated by an arrow) stood for this testing sample. In this case, it was successful. (C) A classifier was built using 15 MS and 14 CONTROL samples. (D) Class prediction of this classifier. We used the CONTROL sample previously withheld (indicated by a green arrow) to test the classifier. In this case, however, it was unsuccessful. The classifier wrongly predicted its class. Overall, for the MS_IFN group, samples in both IFN and MS group were all predicted correctly. For the MS_CONTROL group, one sample in the MS group and two samples in the

CONTROL group were wrongly classified. The accuracy was 90%.

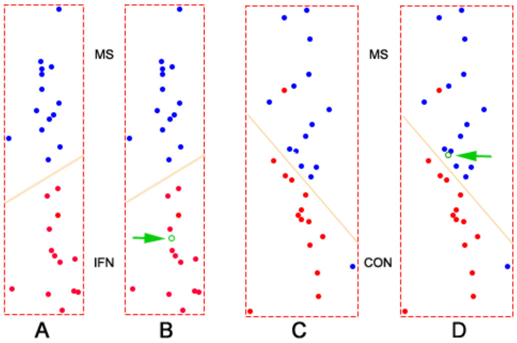

Figure 6: Binary classifications on MS_IFN and MS_CONTROL group of *multiple sclerosis* data set. Notice that in (C) there were two misclassified samples. Those training data errors were not counted in the cross validation.

3.2 More Than Two Sample Classes

Leukemia-B

In [23], *Virtaneva et al.* studied global gene expression in AML+8 patients, AML-CN patients, and normal CD-34+ cells. Their study showed that AML patients clearly distinct from CD34+ normal individuals. The gene expression data has 7129 genes and 27 samples. Among the samples, 10 are AML+8, 10 are AML-CN, and 7 are CD-34+. We performed multiple sample classification on this data. 50 informative genes were used to build a classifier and cross validation was used to evaluate it. Figure 7 shows the tertiary classification process. One classifier was built with leave one AML-CN sample out (indicate by the arraw). AML+8 samples were colored blue, AML-CN samples were colored red (also use unfilled circle), and CD-34+ samples were green. Here, CD-34+ samples were clearly separated from AML samples but AML+8 and AML-CN samples were tend to be mixed. In this case, the AML-CN sample was misclassified. Overall, 8 samples were misclassified.

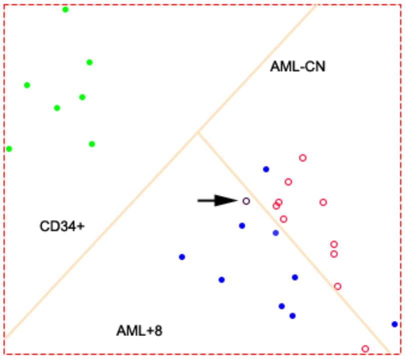

Figure 7: Tertiary classification of *leukemia-B* data set. One classifier was built with leave one AML-CN sample out (indicated by the arrow) and it was misclassified. There were three uncounted training errors, one AML-CN and two AML+8 in this classifier.

Breast Cancer

Another multiple classification used the *BRCA* data set from the work of *Hedenfalk et al.* [11]. They reported on a microarray experiment concerning the genetic basis of breast cancer. Tumors from 22 women were analyzed, with 7 of the women known to have the BRCA1 mutation, 8 known to have BRCA2, and 7 labelled *Sporadics*. Each sample had 3226 genes. We performed another tertiary classification trying to distinguish these 3 subtype of breast cancer samples. Cross validation was used to evaluate the classifier. One classifier is shown in Figure 8. It was built on 50 informative genes and 21 samples with one BRCA1 sample out. BRCA1, BRCA2, and Sporadic samples were colored with blue, green, and red. In this case, they were marginally separated. The classifier successfully assigned the class label to the BRCA1 testing sample indicated by the blue arrow. Overall, 100% accuracy was achieved.

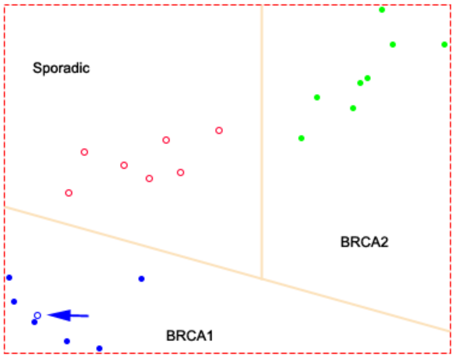

Figure 8: Tertiary classification of *BRCA* data set. The classifier successfully assigned the class label to the BRCA1 testing sample indicated by the blue arrow.

Small Round Blue-Cell Tumors

We concluded our analysis with a multiple classification of a 4-class data set *SRBCT*. *Khan et al.* [15] studied the diagnose of the small, round blue-cell tumors (SRBCTs). SRBCTs include rhabdomosarcoma (RMS), Burkitt lymphomas (BL, a subset of Hodgkin lymphoma), neuroblastoma (NB), and the Ewing family of tumors (EWS). They published a data sets with 2308 genes and 88 (63 training and 25 testing) samples. The 63 training samples include 23 EWS, 8 BL, 12 NB, and 20 RMS. The testing samples include 6 EWS, 3 BL, 6 NB, 4 RMS, and 6 other types. Here we used 63 training samples with 100 informative genes to build classifiers and applied 19 (excluded 6 samples of other types) testing samples to evaluate. Figure 9 illustrates the process. Color blue, red, green, and magenta were assigned to the sample class EWS, BL, NB, and RMS. Filled dots were the training samples and unfilled circle were the testing samples. All four classes of samples are grouped together. The overall accuracy was 95%. One NB testing sample was misclassified as RMS.

3.3 Classification Summary

The summary of all classifications in this section is listed in Table 1.

Data Set	Size	Classes	Size of Classifier	Evaluation	Testing Size	Errors	Accuracy
Leukemia-A	7129×72	2	50	holdout	34	5	85%
MS_IFN	4132×28	2	88	cv	28	0	100%
MS_CON	4132×30	2	88	cv	30	3	90%
Leukemia-B	7129×27	3	50	cv	27	8	70%
BRCA	3226×22	3	50	cv	22	0	100%
SRBCT	2308×82	4	100	holdout	19	1	95%

Table 1: Summary of sample classifications: binary and multiple. cv stands for cross validation.

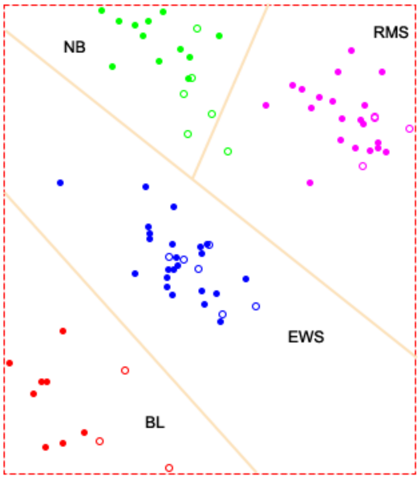

Figure 9: Quaternary classification of *SRBCT* data set. Solid circle were training samples and unfilled circles were testing samples. Four colors were assigned for each sample class.

4. DISCUSSION

Recent work demonstrated that samples can be classified based on gene expression using DNA microarray technology [10; 1; 15; 23; 11]. Our approach is to utilize prior known knowledge (class labels of the train data) and to take the advantage of graphical visualization. VizCluster uses a natural scatter plot to view high dimension data sets and reveals the underlining data patterns. In practice, the run time for the mapping is $O(mn)$, where m is the number of data entities (samples) and n is the number of dimensions (genes). VizCluster does not allow missing value in the data set. It implements *weighted k-nearest neighbor* (KNNimpute) algorithm [21] to fill the missing entries. The mapping (1) shares the common spirit of radial coordinate visualization (Radviz) [12; 13]. However, it does not lie in the same line as multidimension scaling (MDS) methods. In MDS, a dissimilarity stress function is proposed and later iterations are used to optimize the stress function.

The mapping (1) is affected by the order arrangement of dimensions. Since there are $n!$ ways to arrange dimensions for a n-dimensional data set, *canonical dimension ordering* is proposed. Figure 10 shows the effects. To ensure that canonical ordering does not create pseudo class, a random data set is analyzed. See Figure 11.

Our approach to binary and multiple classification is uniform. Unlike methods in [3; 18] which only work on two sample groups, SAM's approach to informative gene identification is similar for both two and multiple sample classes. There are two popular ways to construct classifiers for multiple classes. One is to combine multiple binary classifiers [24]. The other is to directly build classifier for multiple classes. Our approach adopted the second approach. In OC1, by default, oblique decision trees are built by a combination of oblique and axis-parallel methods. OC1 also supports other modes: (1) axis parallel splits at each node which results in axis parallel trees (2) using CART's deterministic perturbation algorithm (3) only oblique splits at each node. Figure 12 shows an axis-parallel decision tree on BRCA data (compare with Figure 8).

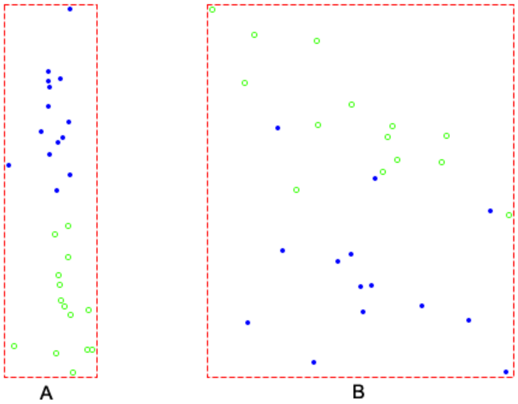

Figure 10: MS_IFN data under different dimension orders. (A) Under canonical order. This is the same as in Figure 6. (B) Under a permutated order.

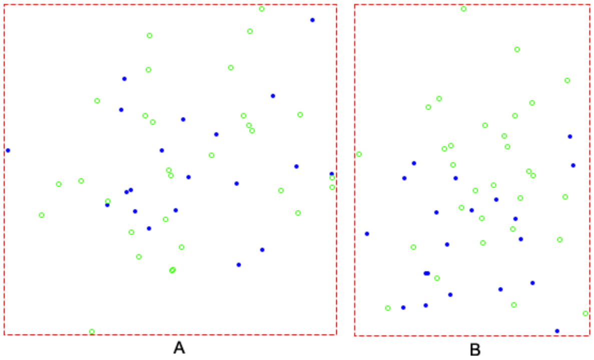

Figure 11: Effect of canonical ordering on a 50 × 100 random data set. Two classes were arbitrarily created and 20 samples were assigned to class 1 and rest to class 2. (A) Under a permuted order. (B) Under canonical order. The result was slightly better but no pseudo class was created by aggregating points in one class together.

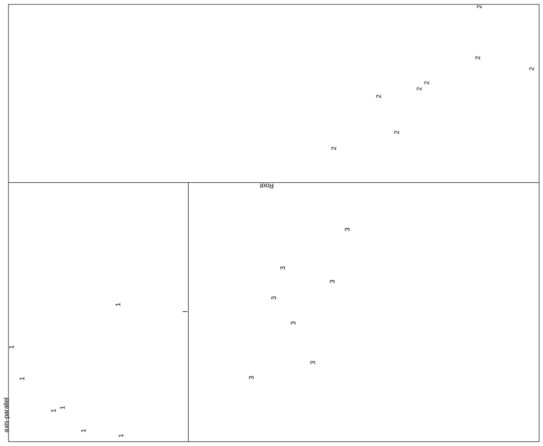

Figure 12: Axis-parallel decision tree on BRCA data. This figure was generated by the original OC1 program. The sample layout is different from Figure 8 because x and y axis scales used here are different.

One should always be aware of making any claim in high dimensional data analysis due to the *curse of dimensionality*. This is particularly true on gene expression data sets. By various constraints (available patients, money etc.), it is hard to dramatically increase the number of samples. Here we apply the classification on different data sets in order to validate our approach. Visualization is not a substitute for quantitative analysis. Rather, it is a qualitative means of focusing analytic approaches and helping users select the most appropriate parameters for quantitative techniques. In this paper, we have not attempted to claim this approach being superior to traditional data analysis methods. Instead, from our experiments, it is demonstrated that visual classification approach has the advantage of taking the global view of the data. It is promising for analyzing and visualizing microarray data sets.

APPENDIX

SAM Algorithm

Formally in [5], let data is x_{ij}, $i = 1, 2, \ldots, p$ genes, $j = 1, 2, \ldots, n$ samples, and response data y_j, $j = 1, 2, \ldots, n$.
(1) compute a statistic

$$d_i = \frac{r_i}{s_i + s_0} \quad i = 1, 2, \ldots, p. \qquad (2)$$

(2) compute order statistics $d_{(1)} \leq d_{(2)} \ldots \leq d_{(p)}$.
(3) Take B permutations of the response values y_j. For each permutation b compute statistics d_i^{*b} and corresponding order statistics $d_{(1)}^{*b} \leq d_{(2)}^{*b} \ldots \leq d_{(p)}^{*b}$.
(4) From the set of B permutations, estimate the expected order statistics by $\overline{d}_{(i)} = \frac{1}{B}\sum_b d_{(i)}^{*b}$ for $i = 1, 2, \ldots, p$.
(5) Plot the $d_{(i)}$ values versus the $\overline{d}_{(i)}$. See Figure 2.
(6) For a fixed threshold Δ, starting at the origin, and moving up to the right find the first $i = i_1$ such that $d_{(i)} - \overline{d}_{(i)} > \Delta$. All genes past i_1 are called *significant positive*. Similarly, start at the origin, move down to the left and find the first $i = i_2$ such that $\overline{d}_{(i)} - d_{(i)} > \Delta$. All genes past i_2 are called *significant negative*. For each Δ define the upper cut-point $cut_{up}(\Delta)$ as the smallest d_i among the significant positive genes, and similarly define the lower cut-point $cut_{down}(\Delta)$.
(7) For a grid of Δ values, compute the total number of significant genes $|G_{sig}|$, and the median number of falsely called genes $|G_{mfc}|$, by computing the median number of values among each of the B sets of $d_{(i)}^{*b}$,$i = 1, 2, \ldots, p$, that fall above $cut_{up}(\Delta)$ or below $cut_{down}(\Delta)$.
(8) Let $\hat{\pi}_0 = \min(\#d_i \in (q_{25}, q_{75}/(0.5p), 1)$, where q_{25} and q_{75} are 25% and 75% points of the permuted d values.
(9)False Discovery Rate (FDR) is defined as $\hat{\pi}_0|G_{mfc}|/|G_{sig}|$.

Canonical Dimension Ordering Algorithm

Let data be x_{ij}, genes $x_i, i = 1, 2, \ldots, m$ and samples y_j, $j = 1, 2, \ldots, n$. The total number of sample classes is K. $C_k = \{j : y_j = k\}$ for $k = 1, 2, \ldots, K$. Let $|C_k|$ be the size of C_k, $\overline{x}_{ik} = \sum_{j \in C_k} x_{ij}/|C_k|$. Let B be the set of $n!$ sequences of all permutations of $1, \ldots, K$.
(1) For each $b \in B$, find set of genes $x^b = \{x_i | \overline{x}_{ik(1)} \leq \overline{x}_{ik(2)}, \ldots, \leq \overline{x}_{ik(K)}$ and $b = k(1), k(2), \ldots, k(K)\}$.
(2) Let $b^* = argmax_{b \in B}|x^b|$. It is some permutation of $1, 2, \ldots, K$, denoted as $b_{k(1)}, \ldots, b_{k(K)}$.
(3) Create a sample class pattern q based on b^*
$q = \{\underbrace{b_{k(1)}, \ldots, b_{k(1)}}_{|C_1|}, \underbrace{b_{k(2)}, \ldots, b_{k(2)}}_{|C_2|}, \ldots, \underbrace{b_{k(K)}, \ldots, b_{k(K)}}_{|C_K|}\}$.
(4) For each gene x_i, compute class coefficient
$r_i = \sigma_{x_i q}/\sqrt{\sigma_{x_i}\sigma_q}$, i.e. Pearson's correlation coefficient with class pattern q. Then sort these r_is.
(5) The canonical order is defined as: $i_{(1)}, i_{(2)}, \ldots, i_{(p)}$ where $r_{i(1)} \leq r_{i(2)}, \ldots, \leq r_{i(p)}$.

OC1 Algorithm

The following is the overview of the OC1 algorithm for a single node of a decision tree [17]. OC1 stands for **O**blique **C**lassifier **1**.

```
To find a split of a set of examples $T$:
  Find the best axis-parallel split of $T$. Let $I$ be the impurity of this split.
  Repeat $R$ times:
    Choose a random hyperplane $H$.
    (For the first iteration, initialize $H$ to be the best axis-parallel split.)
    Step 1: Until the impurity measure does not improve, do:
    Perturb each of the coefficients of $H$ in sequence.
    Step 2: Repeat at most $J$ times:
    Choose a random direction and attempt to perturb $H$ in that direction.
    If this reduces the impurity of $H$, to to Step 1.
    Let $I_1$ = the impurity of $H$. If $I_1 < I$, then set $I = I_1$.
  Output the split corresponding to $I$.
```

A. REFERENCES

[1] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D, and Levine, A. J.g. Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Array. *Proc. Natl. Acad. Sci. USA*, Vol. 96(12):6745–6750, June 1999.

[2] Asimov, D. The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal of Scientific and Statistical Computing*, 6(2):128–143, 1985.

[3] Ben-Dor, A., Friedman, N., and Yakhini, Z. Class Discovery in Gene Expression Data. In *RECOMB 2001: Proceedings of the Fifth Annual International Conference on Computational Biology*, pages 31–38. ACM Press, 2001.

[4] Buja, A., Cook, A., Asimov, D., and Hurley, C. Theory and Computational Methods for Dynamic Projections in High-Dimensional Data Visualization, 1996.

[5] Chu, G., Narasimhan, B., Tibshirani, R., and Tusher, V. SAM: Significance Analysis of Microarrays Users Guide and Technical Document, 2001. Standford University.

[6] Cook, C., Buja, A., Cabrera, J., and Hurley, C. Grand Tour and Projection Pursuit. *Journal of Computational and Graphical Statistics*, 2(3):225–250, 1995.

[7] Dudoit, S., Yang, Y. H., Callow, M. J., and Speed, T. P. Statistical Methods for Identifying Differentially Expressed Genes in Replicated cDNA Microarray Experiments. Technical report 578, Stanford University, Department of Biochemistry Stanford University School of Medicine, August 2000.

[8] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. Cluster Analysis and Display of Genome-wide Expression Patterns. *Proc. Natl. Acad. Sci. USA*, Vol. 95:14863–14868, December 1998.

[9] Ewing, R. M., and Cherry, J. M. Visualization of Expression Clusters Using Sammon's Non-Linear Mapping. *Bioinformatics*, Vol. 17(7):658–659, 2001.

[10] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gassenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, D.D., and Lander, E. S. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, Vol. 286(15):531–537, October 1999.

[11] Hedenfalk, I., Duggan, D., Chen, Y. D., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Kallioniemi, O. P., Wilfond, B., Borg, A., and Trent, J. Gene-Expression Profiles in Hereditary Breast Cancer. *The New England Journal of Medicine*, Vol. 344(8):539–548, February 2001.

[12] Hoffman, P. E., Grinstein, G. G., Marx, K., Grosse, I., and Stanley, E. DNA Visual and Analytic Data Mining. In *IEEE Visualization '97*, pages 437–441, Phoenix, AZ, 1997.

[13] Hoffman, P., Grinstein, G. G., and Pinkney, D. Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations. In *Workshop on New Paradigms in Information Visualization and Manipulation (NPIVM '99), in conjunction with CIKM '99*, pages 9–16, Kansas City, Missouri, November 1999. ACM.

[14] Jiang, D. X., and Zhang, A. Cluster Analysis for Gene Expression Data: A Survey. Technical Report 2002-06, State University of New Your at Buffalo, 2002.

[15] Khan, J., Wei, J. S., Ringnr, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., and Meltzer, P. S. Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks. *Nature Medicine*, Vol.7(6):673–679, 2001.

[16] Murthy, S. K., Kasif, S., and Salzberg, S. A System for Induction of Oblique Decision Trees. *Journal of Artificial Intelligence Research*, (2):1–33, 1994.

[17] Murthy, S. K., Kasif, S., Salzberg, S., and Beigel, R. OC1: A Randomized Induction of Oblique Decision Trees. In *National Conference on Artificial Intelligence*, pages 322–327, 1993.

[18] Park, P. J., Pagano, M., and Bonetti, M. A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data. In *Pacific Symposium on Biocomputing*, pages 52–63, 2001.

[19] Siedow, J. N. Meeting Report: Making Sense of Microarrays. *Genome Biology*, Vol.2(2):reports 4003.1–4003.2, 2001.

[20] Tamayo, P. , Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation. *Proc. Natl. Acad. Sci. USA*, Vol. 96(6):2907–2912, March 1999.

[21] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. Missing Value Estimation Methods for DNA Microarrays. *Bioinformatics*, Vol.17(6):520–525, 2001.

[22] Tusher, V. G., Tibshirani, R., and Chu, G. Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *Proc. Natl. Acad. Sci. USA*, Vol. 98(9):5116–5121, April 2001.

[23] Virtaneva, K., Wright, F., Tanner, S., Yuan, B., Lemon, W., Caligiuri, M., Bloomfield, C., Chapelle, A., and Krahe, R. Expression Profiling Reveals Fundamental Biological Differences in Acute Myeloid Leukemia with Isolated Trisomy 8 and Normal Cytogenetic. *Proc. Natl. Acad. Sci. USA*, Vol. 98(3):1124–1129, January 2001.

[24] Yeang, C. H., Ramaswamy, S., Tamayo, P., Mukherjee, S., Rifkin, R. M., Angelo, M., Reich, M., Lander, E., Mesirov, J., and Golub, T. R. Molecular Classification of Multiple Tumor Types. *Bioinformatics*, Vol. 17 Supplement 1:S316–S322, 2001.

[25] Zhang, L., Tang, C., Shi, Y., Song, Y. Q., Zhang, A., and Ramanathan, M. VizCluster: An Interactive Visualization Approach to Cluster Analysis and Its Application on Microarray Data. In *Second SIAM International Conference on Data Mining*, pages 29–51, April 2002.