

VizCluster: An Interactive Visualization Approach to Cluster Analysis and Its Application on Microarray Data

*Li Zhang, Chun Tang, Yong Shi, Yuqing Song and Aidong Zhang**, *Murali Ramanathan*[†]

Abstract

Visualization enables us to find structures, features, patterns and relationship in a dataset by presenting the data in various graphical forms with possible interactions. Recent development of DNA microarray technology can be used to measure the expression levels of thousands of genes simultaneously. It has already had a significant impact on the field of bioinformatics, requiring innovative techniques to efficiently and effectively extract, analysis and visualize these fast growing data.

In this paper, we present VizCluster, an interactive visualization approach to cluster analysis, and its application on microarray data. VizCluster combines the merits of both high dimensional scatter-plot and parallel coordinates. Integrated with useful features, it can give a simple, fast, intuitive and yet powerful view of the data set. VizCluster supports three major analyzing modes: cluster/class discovery, class prediction, and class assessment. Its primary applications are the classification of samples on microarray datasets. The experiments are based on gene expression data from a study of *multiple sclerosis* and *leukemia* patients.

*Department of Computer Science and Engineering, State University of New York at Buffalo

[†]Department of Pharmaceutical Sciences, State University of New York at Buffalo

1 Introduction

Visualization supports finding structures, features, patterns and relationship in data by presenting the data in various forms with different interactions. A visualization can provide a qualitative overview of large and complex datasets, summarize data, and assist in identifying regions of interest and appropriate parameters for more focused quantitative analysis [1]. Clustering and classification are closely related to the structure or pattern of a data set. Visualizing such structure or pattern can be greatly helpful to exploratory data analysis. Because of structure's complexity, most visualization can not be achieved in a static environment. Instead, a highly interactive environment is often required. Last two decades saw a major development of visualization techniques. Comprehensive overview of data visualization techniques can be found in [1, 2, 3].

Knowledge of the spectrum of genes expressed at a given time or under certain conditions proves instrumental to understand the working of a living cell [4]. Recently, the advent of DNA microarray technology allows measuring expression levels for thousands of genes in a single experiment, across different conditions or over the time [5]. The raw microarray data are images which can then be transformed into gene expression matrices where usually the rows represent genes, the columns represent various samples. 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. 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.

A key step in the analysis of gene expression data is to detect groups that manifest similar expression patterns. Most recent approaches are based on the traditional or newly developed clustering techniques. There are two major paradigms: supervised clustering and unsupervised clustering¹. The supervised approach assumes that for some (or all) profiles, there is additional information attached, such as functional classes for the genes or diseased/normal attributes for the samples. Having this information, a typical task is to build a classifier to predict the class labels from the expression profile. On the other hand, unsupervised approaches assume little or no such prior knowledge. The goal of such approaches is to partition the data into statistically meaningful classes or to find groups of co-regulated genes or related samples. A classifier can also be built not based on the prior knowledge but on the partition result.

During the recent years, various clustering methods are applied on gene expression data analysis. Among all clustering methods, hierarchical clustering (HC) is the most frequently used one [6, 7, 8, 9, 10, 11, 12, 13, 14]. Singular Value Decomposition (SVD) [15, 16, 17] and Principle Component Analysis (PCA) [18, 19, 20] are also widely used. For supervised clustering methods, Golub et al. [21] used neighborhood analysis to construct class predictors for samples. They built a weighted vote classifier based on 50 genes of 38 training samples and applied it on a collec-

¹Historically, supervised clustering sometimes has been referred as *classification* and *clustering* stands for unsupervised clustering

tion of 34 new samples. Hastie et al. [22] proposed a tree harvesting method for supervised learning from gene expression data to discover genes that have strong effects on their own as well as genes that interact with others. Major supervised clustering methods include: Support Vector Machine (SVM) [23, 24, 25], Superparamagnetic Clustering [26], Decision Trees [27] and a variety of ranking based methods [28, 29, 30, 31, 32, 33]. The hierarchical and K-means [34] clustering algorithms as well as self-organizing maps (SOM) [35, 36, 37] are major unsupervised clustering methods that have applied to various microarray datasets. Our group has developed two approaches: supervised Maximum Entropy approach [38] and unsupervised Interrelated Two-way Clustering method [39].

Many visualization tools have also been developed to perform analysis on microarray data. Ewing et al. applies Sammon’s nonlinear mapping, a pure visualization method on a 628×28 data set [40]. Currently, most visualizations serve only as assistant tools or graphical presentations of major clustering methods. For instance, the most frequently applied one, TreeView, developed by Michael Eisen [6, 41] provides a computational and graphical environment. However, the theory behind it is hierarchical clustering and the visualization is merely the graphical format of clustering result. GENECLUSTER [21, 36], developed by Golub is based on SOM. J-Express [42] offers visualization clustering results of four major clustering algorithms: hierarchical clustering, SOM, PCA and K-means.

In this paper, we present an integrated visualization environment, VizCluster, a dynamic interactive visualization approach to cluster analysis. It takes the advantage of graphical visualization methods to reveal the underlining data patterns. Combining the merits of both high dimensional scatterplot and parallel coordinate plots, in its core lies a nonlinear projection which maps the n -dimensional vectors into two-dimensional points [43]. This mapping effectively keeps correlation similarity in the original input space. Also proposed is a zip zooming viewing method which preserves the information at different scales and yet reduces the typical problem of parallel coordinates being messy caused by overlapping lines. We also present a measurement to judge the quality of the cluster distribution using the idea of compactness. This paper largely extends our previous version [44].

Integrated with other features, VizCluster can give a simple, fast, intuitive and yet powerful view of the data set. It supports three major modes of data analysis: cluster/class discovery, in both supervised and unsupervised fashion, class prediction, and class assessment. Our primary goal is to perform classification of samples on gene expression data. The experiments are based on gene expression data from a study of *multiple sclerosis* and *Leukemia* patients. Our approach is to build supervised classifiers and use cross validation to evaluate them. To demonstrate its fully potential, we also performed unsupervised cluster discovery and class assessment on low dimension data sets. Our results show that it has the advantage of being simple and yet powerful for large data sets with low dimension (≤ 10). In all these tasks, VizCluster achieves satisfactory results.

The remainder of this paper is organized as follows. Section 2 presents the visualization framework. Section 3 describes VizCluster’s features and its data analyzing approaches. Section 4 demonstrates the experimental results on microarray and other data sets. And finally, the conclusion is provided in Section 5.

2 Visualization Framework

In this section, we present the framework of VizCluster. One of the key obstacles of visualizing microarray data is the high dimensionality. We propose an interactive visualization framework which combines the merits of both high dimensional scatterplot and parallel coordinate plots. Its core is a nonlinear projection which maps the n -dimensional vectors onto two-dimensional image points [43]. This mapping has the property of keeping correlation similarity in the original space. In doing so, part of the information of original input space may be lost. On the other hand, parallel coordinates allow the information of all dimensions to be visualized. However, as the dimensions go higher, the display becomes messy. We propose a *zip zooming* viewing method which extends circular parallel coordinate plots. Instead of showing all dimensional coordinates, it combines dimensions and displays the reduced dimension information. User can select different scales for combination as needed. This results in a simple and intuitive presentation of the data set and yet preserving the information at different levels. Moreover, each dimension allows its weight to be adjusted independently. An interactive projection-pursuit-guided grand tour [45] like viewing method is incorporated in the framework. We call it *dimension tour*.

2.1 Mapping

The most common presentation in visualization is a two-dimensional scatterplot [1, 2, 3]. We propose a nonlinear mapping that maps the n -dimensional dataset onto two-dimensional space. First, a global normalization was performed on the data set to ensure that each attribute has value between 0 and 1. Let vector $\vec{P}_g^* = (x_{g1}, x_{g2}, \dots, x_{gn})$ represent a data item in the n -dimensional space ($n > 2$, also called input space). Total number of items is m , denoted as $P_1^*, P_2^*, \dots, P_m^*$. Formula (1) describes the mapping of \vec{P}_g^* onto a two-dimensional point \vec{Q}_g^* :

$$\vec{Q}_g^* = \sum_{i=1}^n (\lambda_i * (4/n) * x_{gi}) \vec{S}_i \quad (1)$$

where λ_i is an adjustable weight for each dimension (coordinate), the default value is 0.5. n is dimension of the input space, $4/n$ is a ratio to centralize the points, and \vec{S}_i ($i = 1, 2, \dots, n$) are unit vectors which divide the center circle of the display into equally n directions. Essentially, \vec{Q}_g^* is the vector sum of all its coordinates on n directions.

The mapping (1) preserves correlation relationship in the input space onto the two-dimension images. Notice that the point $(0, 0, \dots, 0)$ in the input space will be mapped onto two-dimensional center $(0, 0)$ (assuming all dimension weights are the same). Actually, all points having the format of (a, a, \dots, a) will also be mapped to the center. If \vec{X} and \vec{Y} have the same pattern, i.e. ratios of each pairs of coordinates of \vec{X} and \vec{Y} are all equal, under the mapping, these vectors will be mapped onto a straight line across the center onto the $2D$ displaying space. All vectors with same pattern as \vec{X} and \vec{Y} will be mapped onto that line.

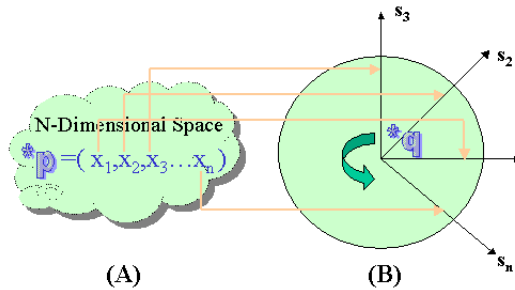


Figure 1. Mapping from n -dimensional input space onto two-dimensional space. (A) A data point P^* in the input n space. (B) Q^* is the corresponding image point using mapping function (Formula (1)) in the $2D$ space. $\vec{S}_i (i = 1, 2, \dots, n)$ are unit vectors which divide the unit circle of the display equally into n directions.

Although the visualization presentation under this mapping is similar to Sammon's mapping [46, 47], they are fundamentally different. Our mapping is significantly different from Sammon's mapping in that Sammon's mapping iteratively minimizes the error defined in (2), while our mapping projects data directly onto $2D$ space.

$$E = \frac{1}{\sum_i \sum_{j < i} d_{ij}^*} \sum_i \sum_{j < i} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*} \quad (2)$$

where d_{ij}^* is the distance $\|x_i - x_j\|$ between points i and j in the n -dimensional input space and d_{ij} is the distance between their $2D$ mapping images.

Dhillon et al. [48, 49] have proposed another optimization oriented class-preserving projections. They want to obtain 2-dimensional projections that best discriminate the q classes with means m_1, m_2, \dots, m_q , each containing n_1, n_2, \dots, n_q data points in d -dimension. The problem is then formulated as the search for orthonormal $w_1, w_2 \in \mathbf{R}^d$ that maximizes

$$C(w_1, w_2) = \text{trace}(W^T S_B W), \quad W = [w_1, w_2] \quad (3)$$

where $w_1^T w_2 = 0$, $w_i^T w_i = 1, i = 1, 2$, and $S_B = \sum_{i=2}^q \sum_{j=1}^{i-1} n_i n_j (m_i - m_j)(m_i - m_j)^T$.

In doing so, class labels are known prior to the projection. However, in our mapping, those labels may be unknown.

2.2 Zip Zooming View

Since mapping (1) could not preserve all the information in the input space, the scatterplot is a lossy visualization representation. By contrast, parallel coordinate plot allows the information of all dimensions to be visualized [1, 2, 50]. In a parallel coordinate system, each dimensional coordinate of a point is plotted along separate parallel axis. A variation of parallel coordinates is a circular version, which the axes

radiate from the center of a circle and extend to the perimeter. The draw back is that as the dimensions goes higher, the displaying has large amount of overlapping lines.

We propose a *zip zooming* (parallel coordinate) viewing method 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 new dimension displayed, we call it a *granularity setting*, can be set by the user. This allows different levels of combination.

Let $\vec{P}_g^* = (x_{g1}, x_{g2}, \dots, x_{gn})$ has the same meaning as in Formula (1), $1 \leq u \leq n$ be a granularity setting and $v = \lfloor n/u \rfloor$ be the number of dimensions to be combined to form one new coordinate. Formula (4) describes the mapping of \vec{P}_g^* onto a set of u two-dimensional points \vec{Q}_g^* :

$$\vec{Q}_g^* = \left\{ \begin{array}{l} Q_{g_1}^* = \sum_{i=1}^v (\lambda_i * (4/n) * x_{gi}) \vec{S}_i \\ Q_{g_2}^* = \sum_{i=v+1}^{2v} (\lambda_i * (4/n) * x_{gi}) \vec{S}_i \\ \vdots \\ Q_{g_u}^* = k * \sum_{i=(u-1)v+1}^n (\lambda_i * (4/n) * x_{gi}) \vec{S}_i \end{array} \right\} \quad (4)$$

Since u may not divide n evenly, the last coordinate $Q_{g_u}^*$ needs a normalization factor $k = v/[n - (u - 1)v]$. A series of such view will allow user to inspect information at different levels from coarse to fine. Figure (2) illustrates this. Note that when granularity becomes finer (u becomes larger), the visualization becomes hard to visualize. On the other hand, under coarse granularity, the presentation becomes cleaner while losing some detailed information.

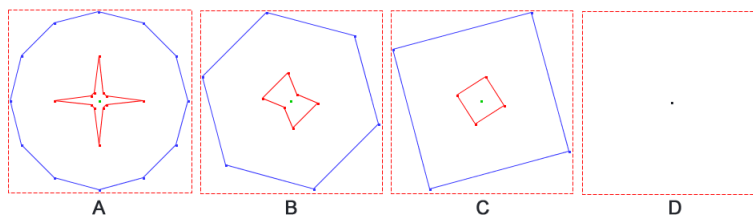


Figure 2. Zip zooming view at different granularity settings. Three data items in 12-dimension input space $(10,10,10,10,10,10,10,10,10,10,10,10)$, $(5,1,1,5,1,1,5,1,1,5,1,1)$, $(0,0,0,0,0,0,0,0,0,0,0,0)$, are represented by blue, red and green. (A) $u = 12$ (B) $u = 6$ (C) $u = 4$ (D) $u = 1$. Notice that all data are mapped to $(0, 0)$ in (D).

Closer look at zip zooming view method reveals that circular parallel coordinate plots and high dimensional scatterplot under (4) are the two extreme cases

while other granularity settings are in between. In fact, according to formula (4), high dimensional scatterplot occurs when $u = 1$ and at $u = n$ it becomes circular parallel coordinate plot. Zip zooming view and high dimensional scatterplot are in complement. The former offers the missing information. They combined allow a simple and intuitive presentation of the data set and yet preserving all the information at different levels.

2.3 Dimension Tour

Two distant points in input space may be mapped to nearby points in $2D$ space or vice versa. One solution is to use zip zooming view to inspect these two points more closely as happened in Figure (2). Another approach is to allow the user to interactively adjust the weights of individual dimension parameters to change data distribution in $2D$ space. It can easily cause the separation of falsely mapped points. 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 . For example, two points $(5, 1, 1, 5, 1, 1, 5, 1, 1, 5, 1, 1)$ and $(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ are far away in the input space. However, by the initial dimension parameter setting, they are both mapped onto the center of the $2D$ space (Figure 2D). Changing any weight λ_i in formula (1) will lead the separation of these two points immediately.

The result of parameter adjustment in scatterplot will cause the redistribution (sometimes dramatic) of the $2D$ representation points. When applied to zip zooming view, each parameter’s adjustment corresponds to a change only on a specific direction. Since the input space dimension can be very high, automatic changing of weights is implemented. So far the discussion of both scatterplot and zip zooming view only limited to static presentations of the data. Dimension parameter adjustment offers not only user interaction but also dynamic visualization. It can perform a projection-pursuit-guided grand tour like view [45, 51, 52]. Based on grand tour idea, we present an animation view method, called *dimension tour*. It is a sequence of either scatterplots or zip zooming views in which each frame has a specific dimension parameter settings. Dimension tour can be defined as a function of time. For scatterplot view, let $\vec{P}_g^* = (x_{g1}, x_{g2}, \dots, x_{gn})$ has the same meaning as in Formula (1) and t be a time index, the sequence of mapping of \vec{P}_g^* are:

$$Q_{g(t)}^* = \sum_{i=1}^n (\lambda_{i(t)} * (4/n) * x_{gi}) \vec{S}_i, \quad \lambda_i(t) \in [-1, 1] \quad (5)$$

where $\lambda_i(t)$ are the parameter settings at time t .

3 Design

In this section, we present measurements of parallel pattern similarity and quality of cluster, VizCluster’s visualization features and three data analyzing modes: cluster/class discovery, class prediction, and class assessment.

3.1 Measurements

Parallel Pattern Similarity

In zip zooming view, when choosing finer granularity, the number of coordinates becomes large. It is hard to judge the similarity purely based on the visualization. Here we use *coefficient of shape difference* [53] created by Penrose [54] as a similarity measurement. This dissimilarity coefficient between two n -dimension points j and k is defined as:

$$z_{jk} = \sqrt{\frac{n}{n-1}(d_{jk}^2 - q_{jk}^2)} \text{ where } q_{jk}^2 = \frac{1}{n^2} \left(\sum_{i=1}^n X_{ij} - \sum_{i=1}^n X_{ik} \right)^2. \quad (6)$$

where d_{jk} is the Euclidean distance between j and k and X_{ij} is the i th coordinate of point j . z_{jk} has the range of $0 \leq z_{jk} \leq \infty$. It equals 0 when two profiles are displaced by the addition of any constant i.e. it is zero when two points are parallel to each other.

Quality of Cluster

In scatterplot view, we need a measurement that evaluates the quality of the cluster. This is especially important to the automatic dimension parameter adjustment. Here we adopt a compactness measurement proposed in [55]. In this report, compactness is defined as the ratio of *external connecting distance* and *internal connecting distance* and cluster has the property of compactness being greater than 1. Detailed discussion and algorithm refers to [55]. In brief, the compactness and cluster are defined in positively weighted graphs. For any graph $G = \langle V, E \rangle$ with positive weight function w and any connected subset $T \subseteq V$, let $L = \{w(p, q) \mid (p, q) \in E, p \in T, q \notin T\}$. The *External connecting distance* (\overline{ECD}) of T with respect to G is defined as: $\overline{ECD}(T; G, w) = \min L$. $\overline{ECD}(T; G, w)$ gives the length of a shortest edge connecting T and $V - T$. Let $L = \{l \in R^+ \mid \langle T, \{(p, q) \in E \mid p \in T, q \in T, w(p, q) \leq l\} \rangle$ is a connected subgraph of $G\}$. The *internal connecting distance* (\overline{ICD}) of T with respect to G is defined as: $\overline{ICD}(T; G, w) = \min L$. $\overline{ICD}(T; G, w)$ gives the shortest length to “maintain” T connected.

T 's compactness thus is defined as $Compactness(T; G, w) = \frac{\overline{ECD}(T; G, w)}{\overline{ICD}(T; G, w)}$. T is compact (or T is a cluster) if its compactness is greater than 1, i.e., $\overline{ECD}(T; G, w) > \overline{ICD}(T; G, w)$.

3.2 Toolkit Features

Many visualization tools have been developed presenting numerous useful features. VizCluster is influenced heavily by XGobi [56], ORCA [57], and XmdvTool [58]. Designed to give a simple, fast, intuitive and yet powerful view of the data set and especially suitable to microarray data analysis, it supports a variety of useful features and functionalities.

VizCluster uses two major viewing modes: scatterplot and zip zooming view. User can choose granularity setting from 1 to n . Up to four settings can be viewed

simultaneously. Highlighting one data item in any view can cause same item be highlighted in other views. For one granularity setting, each data item can be displayed side by side in different frames. User can rearrange these frames, moving data with similar shaper closer together. User can also drag one data item and overlap with another to view the pattern similarity using *coefficient of shape difference*.

Dimension parameters can be adjusted either manually or automatically. All the weight can be obtained when the ideal distribution is reached. Parameter adjustment does not guarantee to find global maximum, often a local maximum is reached. Dimension parameters can be saved and loaded. VizCluster has adopted many useful features appeared in other similar visualization toolkits. Common features such as zooming, selecting and showing/hiding subset of data, brushing (displaying data by different color, legend), are implemented. VizCluster is implemented as Java application.

3.3 Data Analyzing Modes

VizCluster has three data analyzing modes: 1) cluster/class discovery (CD), which can be performed in either supervised or unsupervised manner (SCD or UCD); 2) class prediction (CP), evaluating the performance of a classifier; and 3) class assessment (CA), assessing the class assignment results.

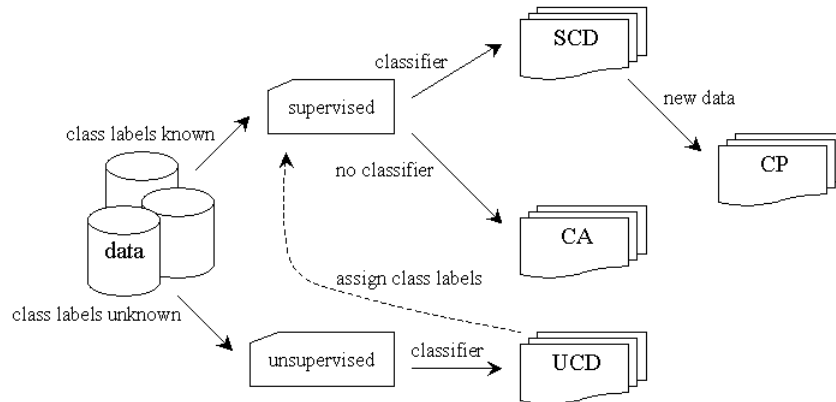


Figure 3. Data Analyzing in VizCluster. There are three data analyzing modes: cluster/class discovery (SCD, UCD), class prediction (CP), and class assessment (CA).

Cluster/Class Discovery

Depend on whether the class labels are known, cluster/class discovery can be divided into two paradigms in VizCluster: *supervised class discovery* (SCD) when class labels are known and *unsupervised cluster discovery* (UCD) otherwise. For either case, the process leads to the construction of a classifier which consists of straight lines to clearly separate the data. In UCD, the goal is to build a classifier

based on the data distribution while in SCD, the goal is to build a classifier for the process of class prediction (CP). Cluster/class discovery is an user interactive process. Initially, data are displayed in scatterplot view using default dimension parameter setting. If the current scatterplot does not indicate a clear separation, user can adjust dimension parameters either manually or automatically. Zip zooming view can help user to find the dimensions which are sensitive to separate data and compare the similarity using detailed information. In automatic adjustment, we use the *compactness* to measure the quality of the distribution of the clusters and decide when to stop the adjustment procedure. To save time, user can choose to inspect the data distribution only visually. The classifier can be saved and later reloaded.

Class Prediction

A classifier should allow the class assignment of newly arrived unlabeled data. This process is called *class prediction* (CP). The data on which the classifier was built is called *training data* while the unlabeled data for class assignment is called *testing data*. Classifier's accuracy is judged by the correctness its class prediction for the testing data. There are two commonly used methods: *holdout* and *cross validation*. In holdout method, data is divided into mutually exclusive training and testing group, then class prediction errors on the testing data are counted. However, when data set is small, the separation of training and testing data may result in insufficient training data. In this case, a method called *leave-one-alone cross validation* [59, 60] is used. All but one data items are used to build a classifier and the last one is withheld as testing data. Then it is repeated in a round robbin way, i.e. each data item is withheld once and the cumulative errors are counted. In VizCluster, different color is used to distinguish the testing data from the training data. Recall in 2.1, a global normalization was performed prior to the mapping of training data. Testing data items are not involved in such global normalization².

Class Assessment

The goal of *class assessment* (CA) is to evaluate the effectiveness of the class assignment. Unlike in class prediction, all data items are involved and no class assignment is performed. Numerous criteria and methods are proposed to evaluate the cluster validity [61]. Here we use an intuitive criteria: the members of each class should be as close to each other as possible, and the classes themselves should be widely spaced. In VizCluster, it is mostly done visually by dimension tour. Since all data items have already been assigned a class label, VizCluster will choose different color (up to 16) or shape for each class. User uses dimension parameter adjustment, zip zooming view and compactness measurement to evaluate the effectiveness of the class assignment.

²There is one subtle issue: one or more dimensions in the testing data may have values outside the range of training data in those dimensions. Testing data will follow the same normalization setting done by training data. For the values outside the range, we simply use 0 or 1 for that dimension. However, the prediction suffers if testing data's value significantly beyond training data's value range.

4 Data Analyzing and Results

In this section, we demonstrate data analyzing results using VizCluster. Tasks involving all three data analyzing modes, cluster/class discover, class prediction, and class assessment are performed. Experiment data sets consist of two low-dimension data sets and two microarray data sets.

4.1 Low-Dimension Data Sets

When facing data sets with low dimension (≤ 10) and large quantity of items, VizCluster is very effective in doing cluster/class discovery and class assessment. Scatterplot is the main viewing method here. Dimension parameter adjustment plays the key role. We will illustrate UCD and CA using two data sets.

Fisher's Iris Data

The first experiment is on Fisher's iris data set. This is a dataset made famous by Fisher, who used it to illustrate principles of discriminant analysis [62]. It contains 4 attributes, *sepal length*, *sepal width*, *petal length*, and *petal width* with 150 observations. The data has 50 plants of each species of iris: *Iris Setosa*, *Versicolor*, and *Virginica* (<ftp://ftp.ics.uci.edu/pub/machine-learning-databases/iris/>).

First we use VizCluster to perform *unsupervised cluster discovery* (UCD). The number of cluster is unknown. User will decide how many clusters is necessary. Figure (4) illustrates this process. Under the scatterplot, two clusters are clearly indicated. Manually or automatically adjusting the dimension parameters will cause a series of redistributions of the points: Figure 4A, 4B and 4C. In this process, hidden cluster may emerge (in Figure 4C). User can stop this interactive process at any time. An unsupervised classifier can be built afterwards.

If class labels have already been given, VizCluster can be used to evaluate the effectiveness of the assignment. Figure (5) illustrates *class assessment* (CA). Different colors are assigned to the classes. User uses parameter adjustment to observe the distribution of these classes. Good assignment keeps classes intact as much as possible under different parameter settings. Here *Setosa* clearly form a class. Although the rest data seem to merge together, under certain settings it can have the trend of being separated.

Gene Hits Data

We present another low-dimension data set for *class assessment* (CA). The data has 448 data items and three attributes. It is about gene hits from 2, 4, and 8 week old animals. Using *neuroKmeans*, a K-means clustering algorithm, 448 data items have been divided into 10 classes. Figure (6) illustrates the class assessment process.

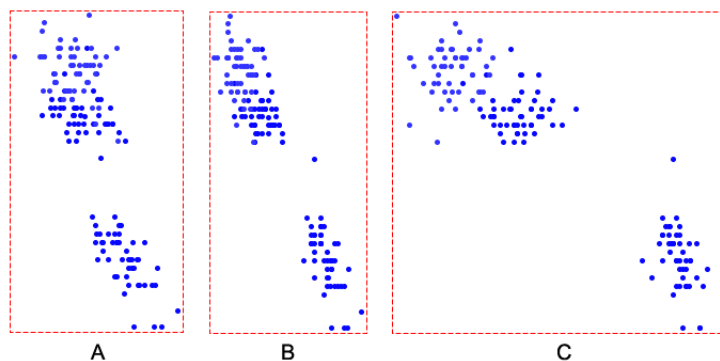


Figure 4. Unsupervised cluster discovery (UCD) using iris data. In unsupervised cluster discovery, all data are using same color. Parameter adjustment plays the key role. (A) (B) (C) show the scatterplots under different parameter settings. It clearly indicates at least two clusters. However, bigger cluster in the upper half may be further separated as multiple smaller clusters. As shown in (C), the bigger cluster in the upper corner starts to break. Three clusters is a reasonable conclusion. An unsupervised classifier can thus be built but not shown here.

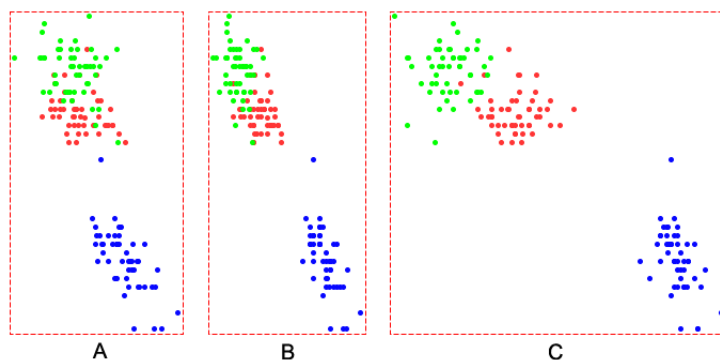


Figure 5. Class assessment (CA) using iris data. Blue is assigned to Setosa, red to Versicolor and green to Virginica. (A) (B) (C) show the scatterplots under different parameter settings. For comparison, in (A) (B) (C) corresponding parameter settings as in Figure 4 are used. Setosa are clearly separated from the other two. After proper adjustment, Versicolor and Virginica can be separated (C). This supports the previous observation in Figure 4(C).

4.2 Classification of Microarray Data

Multiple Sclerosis Data

Our primary experiment is to perform classification of samples on microarray data. The first experiment is based on gene expression data from a study of *multiple sclerosis* patients. Multiple sclerosis (MS) is a chronic, relapsing, inflammatory disease.

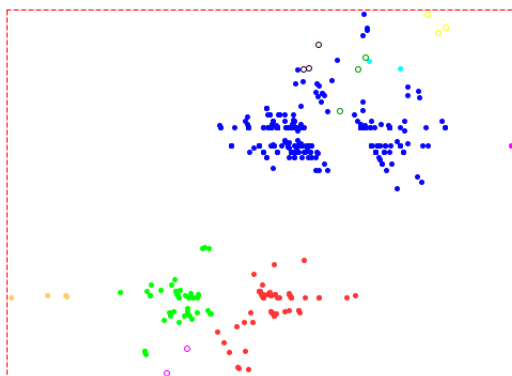


Figure 6. Class assessment using gene hits data. Ten colors are assigned to each of the ten predefined classes. To get a better view, some classes use unfilled circles to display. This snapshot shows the effectiveness of the class assignment. Most classes are indeed have points aggregated closely.

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. It consists of two parts: one contains 28 samples of 14 MS and 14 IFN-treated, we call it MS_IFN group. The other, MS_CONTROL group, contains 30 samples of which 15 are MS and 15 are Health Controls. There are 4132 genes in each group. We will use the VizCluster to distinguish the healthy control, MS and IFN-treated samples by *supervised class discovery* and *class prediction*.

Our approach is to build a supervised classifier from supervised class discovery and then perform class prediction. First we applied neighborhood analysis proposed in [21]. This reduced the gene number from original 4132 to 88. Based on 88 genes, we then built supervised classifiers for each group and apply cross validation for the evaluation. Figure (7) shows one such classifier for the MS_IFN group. Figure (8) shows the classifier for the MS_CONTROL group. The cross validation results are: for the MS_IFN group, samples in both IFN and MS group were all predicted correctly. For the CONTROL_MS group, one sample in the MS group and two samples in the CONTROL group were wrongly classified. The accuracy is 0.90.

Leukemia Data

There is a well-known *leukemia* microarray set which often serves as “benchmark” [63] 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 (http://www.genome.wi.mit.edu/MPR/data_set_ALL_AML.html). It has two groups: training group contains 27 ALL and 11 AML samples; testing group has 20 ALL and 14 AML samples.

We used the similar strategy above. In this case applying neighborhood analysis produced same 50 genes as described in [21]. These 50 genes were then used to

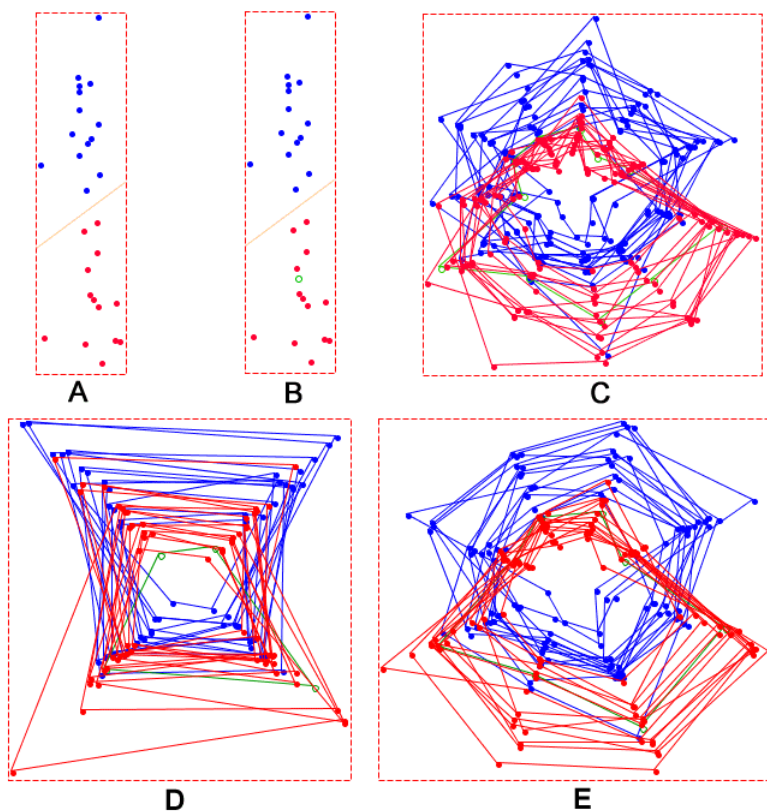


Figure 7. Supervised class discovery and class prediction in *MS_IFN* group. There are 14 samples each for *MS* and *IFN*. (A) A supervised classifier was built using 14 *MS* and 13 *IFN* samples. *MS* samples are colored blue and *IFN* samples red. (B) Class prediction by this classifier. We used the *IFN* sample previously held to test the classifier. The green circle indicates this testing sample. In this case, it was successful. The classifier correctly predicted its class *IFN*. (C) shows a 11 coordinate zip zooming view to display the evaluation in (B). (D) shows a 4 coordinate zip zooming view. (E) shows a 8 coordinate zip zooming view. This gives user more detailed information of each data item. From (D) to (E) to (C), more detail about the data set is revealed. It can help user to find dimensions which are sensitive to the separation of the data. It is also helpful to decide the class labels for closely mapped data items. Notice, although most red items have similar shapes, some have dramatic deviation at certain coordinates.

build a supervised classifier. However, instead of doing cross validation, we directly performed class prediction on the training group (using the same 50 genes) and counted the errors. The result is that five samples were misclassified (out of 34), one *ALL* and four *AML*. The accuracy is 0.85. Most misclassified samples were closed to the line of the classifier. There are two other alternative ways: 1) Apply cross validation on testing group only. 2) Apply cross validation on all 72 samples. Figure (9) shows the process.

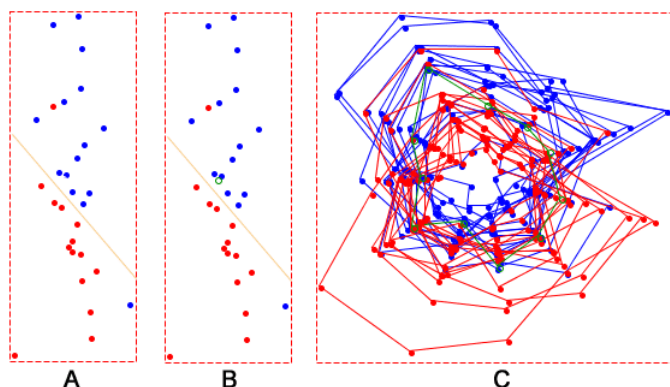


Figure 8. Supervised class discovery and class prediction in *MS_CONTROL* group. There are 15 samples each for *MS* and *CONTROL*. *MS* samples are in blue and *CONTROL* samples in red. (A) A supervised classifier was built using 15 *MS* and 14 *CONTROL* samples. (B) Class prediction of this classifier. We used the *CONTROL* sample previously withheld to test the classifier. In this case, however, it was unsuccessful. The classifier wrongly predicted its class. (C) shows a 11 coordinate zip zooming view the prediction. Samples in this group are harder to separate than that in *MS_IFN* group.

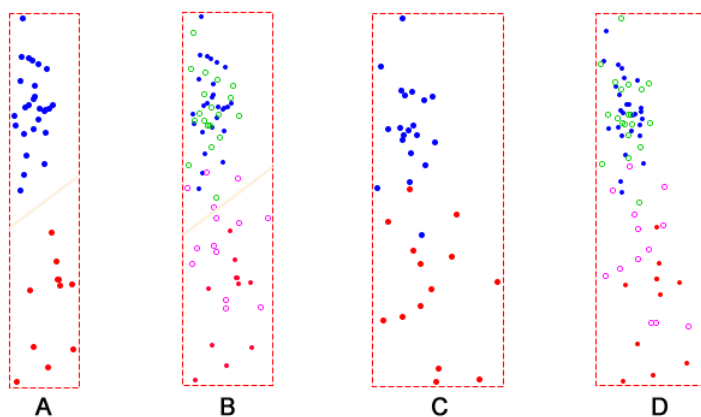


Figure 9. Supervised class discovery and class prediction on leukemia data. (A) A supervised classifier was built using all training samples. It clearly separated *ALL* and *AML* samples. Blue is assigned to *ALL* and *AML* samples are in red. (B) Class prediction by the classifier in (A). Green circles stands for 20 *ALL* samples and magenta circles stand for 14 *AML* samples in the testing group. Overall, the classifier failed to predict one *ALL* and four *AML* samples. (C) The testing group with *ALL* in blue and *AML* in red. (D) All 72 samples from both groups using the same coloring schema as in (B). It indicates fair separation between *ALL* and *AML*. Global normalization involves all the samples in (D) while only training group in (B). This results in different looking between them.

5 Conclusion

Card, Mackinlay, and Schneiderman defined visualization as “*the use of computer-supported, interactive, visual representation of abstract data to amplify cognition*” [64], where cognition is defined as “*the acquisition or use of knowledge*”. We present here visualization environment, VizCluster: a dynamic interactive visualization approach to cluster analysis. It takes the advantage of graphical visualization methods to reveal the underlining data patterns. In practice, VizCluster offers a simple, fast, intuitive and powerful view of the data set.

Information in gene expression matrices is special in that gene expression data has asymmetric dimensionality: sample space has relatively small dimensionality while gene space has very large dimensionality. The framework of VizCluster is designed to suit such analysis. The nonlinear projection mapping and zip zooming viewing method combine the merits of both high dimensional scatter-plot and parallel coordinate plots. The scatter plot is suitable for viewing dense data sets with lower dimension in sacrificing the integrity of information. On the other hand, parallel coordinate plot is efficient in displaying sparse data sets with high dimensions at the cost of the clarity. Zip zooming viewing method serves as the bridge between the two and provide a multiresolution information preservation.

Influenced by XGobi, ORCA, and XmdvTool, there are many appealing features in VizCluster. Besides integrated scatterplot view and zip zooming view, it offers dimension tour, a projecting-pursuit-guided, grand tour like animation view by adjusting dimension parameters. It also implements the measurements of the similarity of parallel shapes and the quality of the cluster distribution.

VizCluster supports three major modes of data analysis: cluster/class discovery, both in supervised (SCD) and unsupervised (UCD) fashion, class prediction (CP), and class assessment (CA). Our results show that it has the advantage of being simple and yet powerful in doing UCD and CA on low dimension data sets such as *iris* data. Our primary goal is to perform classification of samples on gene expression data (SCD and CP). By building supervised classifiers and using hold-out or cross validation evaluation, VizCluster clearly demonstrates its power. In all these tasks, VizCluster achieves satisfactory results.

Visualization is not a substitute for quantitative analysis. Rather, it is a qualitative means of focusing analytic approaches and helping users to 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, we demonstrated that VizCluster’s approach is a promising one to be used for analyzing and visualizing of microarray data sets and further development is worthwhile.

Bibliography

- [1] Usama Fayyad, Georges G Grinstein, Andreas Wierse, editor. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers, San Diego, CA, 2001.
- [2] Pak Chung Wong and R. Daniel Bergeron. 30 Years of Multidimensional Multivariate Visualization. University of New Hampshire.
- [3] Adalbert F.X. Wilhelm, Edward J. Wegman, and Jürgen Symanzik. Visual Clustering and Classification: The Oronsay Particle Size Data Set Revisited.
- [4] Vingron, M. and Hoheisel, J. Computational Aspects of Expression Data. *J. Mol. Med.*, 77:3–7, 1999.
- [5] Alvis Brazma and Jaak Vilo. Minireview: Gene Expression Data Analysis. *Federation of European Biochemical Societies*, 480:17–24, June 2000.
- [6] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster Analysis and Display of Genome-wide Expression Patterns. *Proc. Natl. Acad. Sci. USA*, Vol. 95:14863–14868, December 1998.
- [7] Ash A. Alizadeh, Michael B. Eisen, R. Eric Davis, Chi Ma, Izidore S. Lossos, Adreas Rosenwald, Jennifer C. Boldrick, Hajeer Sabet, Truc Tran, Xin Yu, John I. Powell, Liming Yang, Gerald E. Marti, Tory Moore, James Hudson Jr, Lisheng Lu, David B. Lewis, Robert Tibshirani, Gavin Sherlock, Wing C. Chan, Timothy C. Greiner, Dennis D. Weisenburger, James O. Armitage, Roger Warnke, Ronald Levy, Wyndham Wilson, Michael R. Grever, John C. Byrd, David Botstein, Patrick O. Brown, and Louis M. Staudt. Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling. *Nature*, Vol.403:503–511, February 2000.
- [8] Therese Biedl, Brona Brejova, Erik D. Demaine, Angele M. Hamel, and Tomas Vinar. Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data. Technical report 2001-14, University of Waterloo, Canada, 2001.
- [9] Sige Zou, Sarah Meadows, Linda Sharp, Lily Y. Jan, and Yuh Nung Jan. Genome-Wide Study of Aging and Oxidative Stress Response in *Drosophila*

- Melanogaster. *Proc. Natl. Acad. Sci. USA*, Vol. 97(25):13726–13731, December 2000.
- [10] Min Jiang, Jubin Ryu, Monika Kiraly, Kyle Duke, Valerie Reinke, and Stuart K. Kim. Genome-Wide Analysis of Developmental and Sex-Regulated Gene Expression Profiles in *Caenorhabditis Elegans*. *Proc. Natl. Acad. Sci. USA*, Vol. 98(1):218–223, January 2001.
- [11] Kimmo Virtaneva, Fred A. Wright, Stephan M. Tanner, Bo Yuan, William J. Lemon, Michael A. Caligiuri, Clara D. Bloomfield, Albert de la Chapelle, and Ralf Krahe. 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.
- [12] John B. Welsh, Patrick P. Zarrinkar, Lisa M. Sapinoso, Suzanne G. Kern, Cynthia A. Behling, Bradley J. Monk, David J. Lockhart, Robert A. Burger, and Garret M. Hampton. Analysis of Gene Expression Profiles in Normal and Neoplastic Ovarian Tissue Samples Identifies Candidate Molecular Markers of Epithelial Ovarian Cancer. *Proc. Natl. Acad. Sci. USA*, Vol. 98(3):1176–1181, January 2001.
- [13] Katherine J. Martin, Edgard Graner, Yi Li, Laura M. Price, Brian M. Kritzman, Marcia V. Fournier, Esther Rhei, and Arthur B. Pardee. High-Sensitivity Array Analysis of Gene Expression for the Early Detection of Disseminated Breast Tumor Cells in Peripheral Blood. *Proc. Natl. Acad. Sci. USA*, Vol. 98(5):2646–2651, February 2001.
- [14] Yaron Hakak, John R. Walker, Cheng Li, Wing Hung Wong, Kenneth L. Davis, Joseph D. Buxbaum, Vahram Haroutunian, and Allen A. Fienberg. Genome-Wide Expression Analysis Reveals Dysregulation of Myelination-Related Genes in Chronic Schizophrenia. *Proc. Natl. Acad. Sci. USA*, Vol. 98(8):4746–4751, April 2001.
- [15] Orly Alter, Patrick O. Brown, and David Bostein. Singular Value Decomposition for Genome-wide Expression Data Processing and Modeling. *Proc. Natl. Acad. Sci. USA*, Vol. 97(18):10101–10106, August 2000.
- [16] Neal S. Holter, Amos Maritan, Marek Cieplak, Nina V. Fedoroff, and Jayanth R. Banavar. Dynamic Modeling of Gene Expression Data. *Proc. Natl. Acad. Sci. USA*, Vol. 98(4):1693–1698, February 2001.
- [17] Mike West, Joseph R. Nevins, Jeffrey R. Marks, Rainer Spang, Carrie Blanchette, and Harry Zuzan. DNA Microarray Data Analysis and Regression Modeling for Genetic Expression Profiling. Duke University, 2001.
- [18] Xiling Wen, Stefanie Fuhrman, George S. Michaels, Daniel B. Carr, Susan Smith, Jeffery L. Barker, and Roland Somogyi. Large-Scale Temporal Gene Expression Mapping of Central Nervous System Development. *Proc. Natl. Acad. Sci. USA*, Vol. 95(1):334–339, January 1998.

- [19] K.Y. Yeung, and W.L. Ruzzo. An Empirical Study On Principal Component Analysis for Clustering Gene Expression Data. Technical Report UW-CSE-01-04-02, University of Washington, 2001.
- [20] S. Raychaudhuri, J.M. Stuart, and R.B. Altman. Principal Components Analysis to Summarize Microarray Experiments: Application to Sporulation Time Series. In *Pacific Symposium on Biocomputing*, pages 415–426, 2000.
- [21] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gassenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, D. D. Bloomfield, E. S. Lander. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, Vol. 286(15):531–537, October 1999.
- [22] Trevor Hastie, Robert Tibshirani, David Botstein, and Patrick Brown. Supervised Harvesting of Expression Trees. *Genome Biology*, Vol. 2(1):0003.1–0003.12, January 2001.
- [23] Michael P.S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnetn Terrence S. Furey, Manuel Ares, Jr., and David Hausler. Knowledge-Based Analysis of Microarray Gene Expression Data by Using Support Vector Machines. *Proc. Natl. Acad. Sci. USA*, Vol. 97(1):262–267, January 2000.
- [24] Paul Pavlidis, Jason Weston, Jinsong Cai, and William N. Grundy. Gene Functional Classification from Heterogeneous Data. In *RECOMB 2001: Proceedings of the Fifth Annual International Conference on Computational Biology*, pages 249–255. ACM Press, 2001.
- [25] Terrence S. Furey, Nello Cristianini, Nigel Duffy, David W. Bednarski, Michl Schummer, and David Haussler. Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data. *Bioinformatics*, Vol.16(10):909–914, 2000.
- [26] G. Getz, E. Levine, E. Domany, and M. Q. Zhang. Super-Paramagnetic Clustering of Yeast Gene Expression Profiles. *Physica A*, Vol.279:457–464, 2000.
- [27] Heping Zhang, Chang-Yung Yu, Burton Singer, and Momiao Xiong. Recursive Partitioning for Tumor Classification with Gene Expression Microarray Data. *Proc. Natl. Acad. Sci. USA*, Vol. 98(12):6730–6735, June 2001.
- [28] Wentian Li. Zipf’s Law in Importance of Genes for Cancer Classification Using Microarray Data. Lab of Statistical Genetics, Rockefeller University, April 2001.
- [29] A. Ben-Dor, N. Friedman, and Z. Yakhini. 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.

- [30] Peter J. Park, Marcello Pagano, and Marco Bonetti. A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data. In *Pacific Symposium on Biocomputing*, pages 52–63, 2001.
- [31] Jeffrey G. Thomas, James M. Olson, Stephen J. Tapscott, and Lue Ping Zhao. An Efficient and Robust Statistical Modeling Approach to Discover Differentially Expressed Genes Using Genomic Expression Profiles. *Genome Research*, Vol. 11(7):1227–1236, 2001.
- [32] Virginia Goss Tusher, Robert Tibshirani, and Gilbert Chu. Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *Proc. Natl. Acad. Sci. USA*, Vol. 98(9):5116–5121, April 2001.
- [33] E. J. Moler, M. L. Chow, and I. S. Mian. Analysis of Molecular Profile Data Using Generative and Discriminative Methods. *Physiological Genomics*, Vol. 4(2):109–126, 2000.
- [34] KaYee Yeung, David R. Haynor, and Walter L. Ruzzo. Validating Clustering for Gene Expression Data. *Bioinformatics*, Vol.17(4):309–318, 2001.
- [35] Neal S. Holter, Madhusmita Mitra, Amos Maritan, Marek Cieplak, Jayanth R. Banavar, and Nina V. Fedoroff. Fundamental Patterns Underlying Gene Expression Profiles: Simplicity from Complexity. *Proc. Natl. Acad. Sci. USA*, Vol. 97(15):8409–8414, July 2000.
- [36] Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhu, Sutisak Kitareewan, Ethan Dmitrovsky, Eric S. Lander, and Todd R. Golub. 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.
- [37] Monica Mody, Yanxiang Cao, Zhenzhong Cui, Khoon-Yen Tay, Andy Shyong, Eiji Shimizu, Kelvin Pham, Peter Schultz, Douglas Welsh, and Joe Z. Tsien. Genome-Wide Gene Expression Profiles of the Developing Mouse Hippocampus. *Proc. Natl. Acad. Sci. USA*, Vol. 98(15):8862–8867, July 2001.
- [38] Shumei Jiang, Chun Tang, Li Zhang and Aidong Zhang, Murali Ramanathan. A Maximum Entropy Approach to Classifying Gene Array Data Sets. In *Proc. of Workshop on Data mining for genomics, First SIAM International Conference on Data Mining*, Chicago, IL, 2001.
- [39] Chun Tang, Li Zhang, and Aidong Zhang, Murali Ramanathan. Interrelated Two-way CLustering: An Unsupervised Approach for Gene Expression Data Analysis. In *Proc. of 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, Bethesda, MD, 2001.
- [40] Rob M. Ewing and J. Michael Cherry. Visualization of expression clusters using Sammon’s non-linear mapping. *Bioinformatics*, Vol. 17(7):658–659, 2001.

- [41] Michael Eisen. *Cluster and TreeView Manual*. Stanford University, 1998–1999. rana.lbl.gov/EisenSoftware.html.
- [42] B. Dysvik and I. Jonassen. J-Express: exploring gene expression data using Java. *Bioinformatics*, 17(4):369–370, 2001. Applications Note.
- [43] Debangshu Bhadra. An interactive visual framework for detecting clusters of a multidimensional dataset. Master’s thesis, State University of New York at Buffalo, March 2001.
- [44] Chun Tang, Li Zhang, and Aidong Zhang. Interactive Visualization and Analysis for Gene Expression Data. In *2002 IEEE. Published in the Proceedings of the Hawaii International Conference on System Sciences*, Big Island, HI, 2001.
- [45] D. Asimov. The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal of Scientific and Statistical Computing*, 6(2):128–143, 1985.
- [46] Michael E. Tipping. *Topographical Mappings and Feed-Forward Neural Networks*. PhD thesis, University of Aston in Birmingham, February 1996.
- [47] J.W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.
- [48] Inderjit S. Dhillon, Dharmendra S. Modha, and W. Scott Spangler. Visualizing Class Structure of Multidimensional Data, 1998. IBM Almaden Research Center, San Jose, CA 95120.
- [49] Inderjit S. Dhillon, Dharmendra S. Modha, and W. Scott Spangler. Class Visualization of High-Dimensional Data with Applications, 1999. IBM Almaden Research Center, San Jose, CA 95120.
- [50] Edward J. Wegman and Qiang Luo. High Dimensional Clustering Using Parallel Coordinates and the Grand Tour, 1996. George Mason University.
- [51] Dianne Cook and Andreas Buja. Manual Controls For High-Dimensional Data Projections.
- [52] A. Buja, D. Cook and D. Swayne. Interactive High-Dimensional Data Visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996.
- [53] H.Charles Romesburg. *Cluster Analysis for Researchers*. Lifetime Learning Publications, Belmont, CA, 1984.
- [54] L.S. Penrose. Distance, size and shape. *Annals of Eugenics*, Vol. 18:337–343, 1953.
- [55] Yuqing Song and Aidong Zhang. Cluster and Compactness. Technical report, Department of Computer Science and Engineering, State University of New York at Buffalo, 2001.

- [56] D.F. Swayne, D. Cook and A. Buja. XGobi: Interactive Dynamic Graphics in the X Window System. *Journal of Computational and Graphical Statistics*, 27:299–303, 1996.
- [57] Peter Sutherland, Anthony Rossini, Thomas Lumley, Nicholas Lewin-Koh, Dianne Cook, Zach Cox. ORCA: A Visualization Toolkit for High-Dimensional Data. <http://pyrite.cfas.washington.edu/orca>.
- [58] Matthew O. Ward. XmdvTool: Ingegrating Multiple Methods for Visualizing Multivariate Data, 1999. Worcester Polytechnic Institute.
- [59] Bradley Efron and Robert Tibshirani. Cross-Validation and the Bootstrap: Estimating the Error Rate of a Prediction Rule, 1996. Stanford University.
- [60] Ron Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence*, 1995.
- [61] Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis. On Clustering Validation Techniques, 2000. Athens University of Economics and Business, Athens, Greece.
- [62] R.A. Fisher. The use of multiple measurements on taxonomic problems. *Annals of Eugenics*, pages 179–188, 1936.
- [63] James N Siedow. Meeting Report: Making Sense of Microarrays. *Genome Biology*, Vol.2(2):reports4003.1–4003.2, 2001.
- [64] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editor. *Reading in Information Visualization: Using Vision to Think*. Morgan Koffmann, San Francisco, CA, 1999.