Information Visualization for DNA Microarray Data Analysis: A Critical Review

Leishi Zhang, Jasna Kuljis, and Xiaohui Liu

***Abstract*—Graphical representation may provide effective means of making sense of the complexity and sheer volume of data produced by DNA microarray experiments that monitor the expression patterns of thousands of genes simultaneously. The ability to use "abstract" graphical representation to draw attention to areas of interest, and more in-depth visualizations to answer focused questions, would enable biologists to move from a large amount of data to particular records they are interested in, and therefore, gain deeper insights in understanding the microarray experiment results. This paper starts by providing some background knowledge of microarray experiments, and then, explains how graphical representation can be applied in general to this problem domain, followed by exploring the role of visualization in gene expression data analysis. Having set the problem scene, the paper then examines various multivariate data visualization techniques that have been applied to microarray data analysis. These techniques are critically reviewed so that the strengths and weaknesses of each technique can be tabulated. Finally, several key problem areas as well as possible solutions to them are discussed as being a source for future work.**

***Index Terms*—Data analysis, gene expression, microarray, visualization.**

I. Introduction

OVER the last few years, it has been common to use high-throughput functional genomics methods to investigate multiple events in a cell or tissue that define a phenotype. DNA microarrays are one such methodology that allows the simultaneous determination of mRNA abundance for many thousands of genes in a single experiment [21], [52]. Given that genes with related functions are likely to be regulated together, microarray techniques provide a mechanism for the initial identification and studying of novel gene sequences with related functions [14], [42]. However, the generation of all these gene expression data will lose most of its potential value unless important conclusions can be extracted from such large datasets [10], [20]. To explore the full potential of microarray data, the data has to be analyzed and presented in a way that biologists can readily understand. One approach to achieving this goal is through the use of data visualization techniques [22], [43].

Multidimensional data analysis is an established technique for exploration, analysis, and presentation of large datasets. A graphical representation is generated from the data content, and viewed by an observer, engaging vision—the human sense with the greatest bandwidth and the ability to recognize patterns subconsciously [4]. It has two fundamental related aspects: 1) structural modeling and 2) graphical representation [12]. *Structural modeling* aims at detecting, extracting, and simplifying underlying relationships by forming a structure that characterizes a collection of documents or other datasets. Structural modeling involves applying appropriate computational, statistical, or data mining techniques to generate virtual structures of the data. *Graphical representation* aims to transforming an initial presentation of structure into a graphical one, so that the structure can be visually examined and interacted with. Graphical representation can be divided into two main components—*intrinsic visualization* (graph drawing) and *extrinsic visualization* (visual form). The former maps object relationships to spatial distances whereas the latter maps object properties to color, shape, texture, and so on [5] (see Fig. 1). However, the boundary of these two aspects of visualization is blurred. Algorithms such as multidimensional scaling (MDS), self-organizing maps (SOMs), and principal component analysis (PCA) can be classified either as structural modeling methods or graph drawing techniques since the outputs of these data mining algorithms can be directly plotted to graphical display without further manipulation.

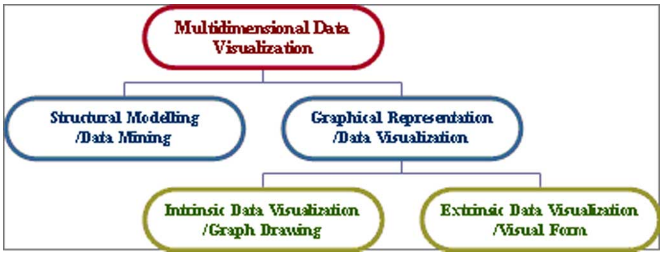

Multidimensional Data Visualization
Structural Modelling /Data Mining
Graphical Representation /Data Visualization
Intrinsic Data Visualization /Graph Drawing
Extrinsic Data Visualization /Visual Form

Fig. 1. Multidimensional data visualization.

Since much work has already been carried out to review the structural modeling aspect of microarray data analysis [2], [46], [53], [55] and existing microarray data visualization tools [40], [50], this paper instead focuses on the graphical representation techniques that have been applied to microarray data analysis, where not much attention has been received. The remainder of this paper is organized as follows. In Section II, some background knowledge of microarray experiment and data visualization is provided for understanding the main features of microarray datasets. Section III discusses the role of visualization in gene expression data analysis. Section IV introduces the concept of graphical representation, which consists of two fundamental components—graph drawing and visual form. Sections V and VI critically review the graph drawing algorithms and visual methods that have been applied to display the

Manuscript received May 30, 2006; revised October 26, 2007. This paper was recommended by Editor V. Marik.

The authors are with Computing Laboratory, University of Kent, Canterbury, U.K., CT2 7NF and also with the School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, UB8 3PH, U.K. (e-mail: l.zhang@kent.ac.uk).

Digital Object Identifier 10.1109/TSMCC.2007.906065

data mining results respectively. Section VII summarizes the whole paper, identifies several key problem areas, and proposes possible solutions.

II. Microarray Experiments

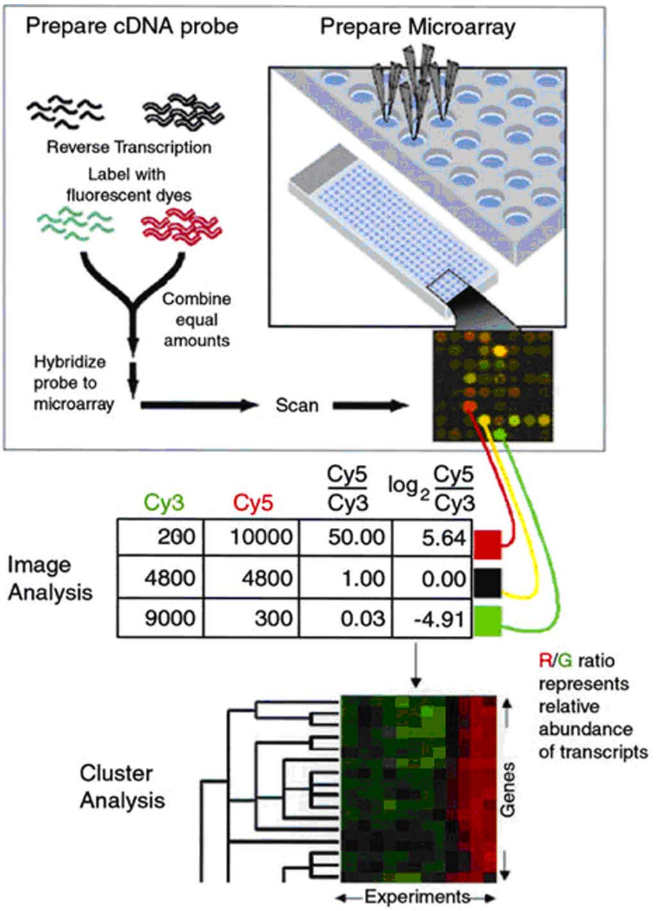
A microarray is normally a glass or silicon slide, onto which single-stranded DNA molecules are attached at fixed locations or spots. Such a microarray may consist of thousands of spots, each related to a single gene. The central principle of the microarray technique is the selective binding of complementary single-stranded nucleic acid sequences (hybridization) and the use of fluorescent probes to visualize the difference in cDNA level that represents mRNA level [15], [24], [41].

One of the most popular experimental methods is to compare the mRNA levels across two cell cultures such as a cancer cell versus a healthy cell as a control (see Fig. 2) [13]. The first step of the experiment is to extract purified mRNA from both cell types. In order to distinguish cDNAs from different cell types, fluorescent labeling molecules of different colors (usually red and green) are used to stain each sample. After labeling the cDNA molecules, both extracts are washed over the microarray where they bind selectively to their complementary DNA strands in the spots according to the principle of the microarray technique described before. The last step of the experiment measures the hybridization level of each spot on the array. If the mRNA from the cancer cell is abundant, the spot will be red; if the mRNA from the healthy cell is abundant, it will be green. If mRNAs from both cells bind equally, the spot will be yellow, while if neither binds, it will not be fluorescent and the spot will be black. The red and green light detection channels will then be normalized so that the two datasets can be compatible for intensity ratio calculation, i.e., each data point produced by a DNA microarray experiment represents the ratio of expression levels of a particular gene under two different experimental conditions.

The data from a series of m such experiments may be represented as a gene expression matrix, in which each of the n rows (each row representing an attribute) consists of an m-element expression vector for a single gene (each element representing a single gene). Since microarray data normally consist of a large number of attributes (genes), the visualizations always face the challenge of displaying a large amount of information on a limited computer display. So, sophisticated searching and analysis methods are required to highlight features hidden in this special dataset. This is the reason why advanced data mining and visualization techniques can potentially be applied to help biologists extract meaningful patterns from microarray data.

III. Gene Expression Data Analysis and Visualization

Microarray experiments often produce such a massive amount of information that is too large to study and interpret manually from a spreadsheet or a plain text file. Visualization uses computer graphics to present data or information in different visual forms so that meaningful patterns can be extracted from the large datasets. Such patterns may, for example, help biologists detect the likely functions of genes, how genes may be regulated, and how they interact with each other in health and disease processes [45].



Cy3	Cy5	$\frac{Cy5}{Cy3}$	$\log_2 \frac{Cy5}{Cy3}$
200	10000	50.00	5.64
4800	4800	1.00	0.00
9000	300	0.03	-4.91

Fig. 2. Microarray experiment (copyrighted Cummings and Relman [13]).

A. Visualization in Gene Expression Data Analysis

Visualization has a special place in data analysis because of the power of the human eye/brain to detect structures. Various visual methods can be used to display data mining processes and outcomes in ways that capitalize the particular strengths of human pattern processing abilities. Appropriate visual methods may provide instant recognition of unknown patterns and unexpected relationships by drawing attention to the areas of interest or answering focused biological questions. In gene expression data analysis, the visualization can facilitate tasks such as determining the characteristics of unknown genes, bringing clarity to previously unknown diagnostic categories [1], extracting focused networks from existing database [49], or confirming prevailing clinic hypotheses [47].

Recently, visualization has helped biologists find many interesting relationships and patterns from gene expression data. For example, Alon *et al.* [3] proposed a two-way clustering method for analyzing a data set consisting of the expression patterns of different cell types where the visualization clearly separates cancerous from noncancerous tissue and cell lines from *in vivo* tissues on the basis of subtle distributed patterns of genes. Bhattacharjee and Richards [8] found distinct adenocarcinoma subclasses by examining the mRNA expression levels corresponding to 12 600 transcript sequences in 186 lung tumor samples. Khan *et al.* [36] used gene expression profiling and artificial neural networks to help classification and diagnostic prediction of cancers. Ramaswamy *et al.* [47] found a gene expression molecular signature of metastasis in primary solid tumors by comparing the gene-expression profiles of adenocarcinoma metastases of multiple tumor types to unmatched primary adenocarcinomas.

Using two examples, discussed next, we will demonstrate how visualization can help find patterns and relationships in the data mining results. The first example shows how two distinct subclasses of a disease can be distinguished from a dendrogram view of gene expression data; the second example illustrates how a meaningful focused interaction subnetwork can be extracted from a massive protein interaction network.

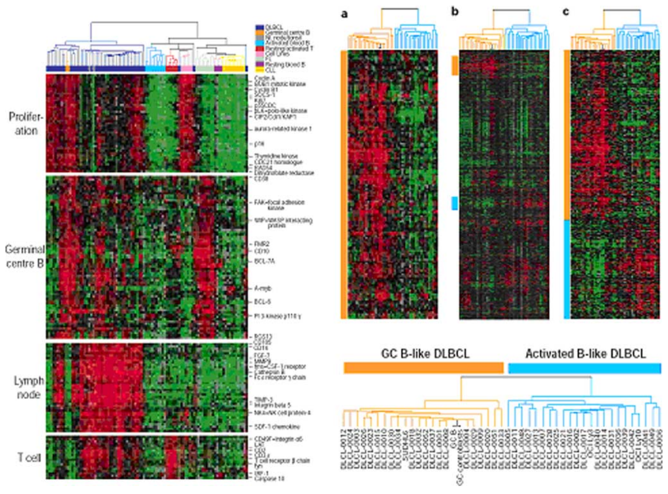


Fig. 3. DLBCL subgroups discovery (copyrighted Alizadeh *et al.* [1]).

Example 1—Refinement of Clinical Classification: The most common method in gene expression data analysis is *clustering* that groups together genes with similar expression profiles. Genes that are similarly expressed often participate in the same cellular processes, so the clustering may suggest possible functional relationships among the clustered genes.

Alizadeh *et al.* [1] have demonstrated that visualizing the clustering result can bring clarity to previously unknown diagnostic categories in cancer research. By visually inspecting the gene expression profiling, they identified two molecularly distinct forms of diffuse large B-cell lymphoma (DLBCL) that had gene expression patterns indicative of different stages of B-cell differentiation. Fig. 3 illustrates how the subtypes of DLBCL are discovered by visually inspecting the data mining results. Firstly, the hierarchical clustering view of all gene expression data is generated; the clustering result indicated that gene expression patterns in DLBCLs might be inhomogeneous and the expression of the germinal center B-cell genes among DLBCLs varied independently from the expression of genes in other gene expression signatures (see the left dendrogram). As a consequence, the expression patterns of the genes that define the germinal center B-cell signature are picked out for reclustering. Two DLBCL subgroups, GC B-like DLBCL (orange) and activated B-like DLBCL (blue), were then defined by this process (see the right dendrograms). The classification was validated by clinical records—patients with GC B-like DLBCL had a significantly better overall survival rate than those with activated B-like DLBCL. The molecular classification of tumors on the basis of gene expression visualization can, thus, help identify previously undetected and clinically significant subtypes of cancer.

Example 2—Extraction of Focused Subnetwork: Another important method in gene expression data analysis is *network modeling* that generates gene networks from microarray data using various data mining techniques [16]. Among all the genes that are selected for a set of microarray experiments, specific groups of genes may be activated by particular signals, which once activated, regulate a common biological process. The group members may regulate each other's transcriptions. Such groups are called genetic regulatory systems. Network modeling is used for observing the interrelationship between genes within a genetic regulatory system. The modeling of gene networks is not only useful for data understanding, but also helpful in generating possibly interesting hypotheses for further investigation. Again, visualization of the network structure provides a clear and concise summary of the regulatory interactions. Moreover, higher level structures can be extracted from the network representation.

An example of such network extraction can be found in the work of Rhodes and Chinnaiyan [49], who tried to gain a deeper biological insight from the cancer gene expression data by visually exploring the links in a protein interaction network and speculating the linchpin of a cancer. Fig. 4 illustrates a known protein interaction network and a focused interaction subnetwork extracted from the network. On the left is a signature of 300 genes significantly overexpressed in multiple myeloma. In the middle is a known network structure according to the Human Protein Reference Database. Within the network, the interactions among 300 genes that are significantly overexpressed in multiple myeloma are highlighted in red. A focused interaction subnetwork in which all members are overexpressed was extracted (on the right). Upon exploring the links in the subnetwork, Rhodes and Chinnaiyan [49] speculated that RAF1 may be the linchpin, as several members of the network (RAS, PAK1, and BAG1) function to activate RAF1. Thus, by targeting RAF1, as opposed to other members of the network, biologists may be able to blunt the effects of the entire subnetwork. Although they stated that the interpretation may be speculative, it does highlight the potential for interaction networks in the analysis of cancer signatures. Insights gained from such analysis can be treated as hypotheses for further research.

The aforementioned examples demonstrate that visualization can play an important role in understanding and further exploring gene expression data analysis. In the next section, the concept of data visualization (graphical representation) and its main contents will be discussed.

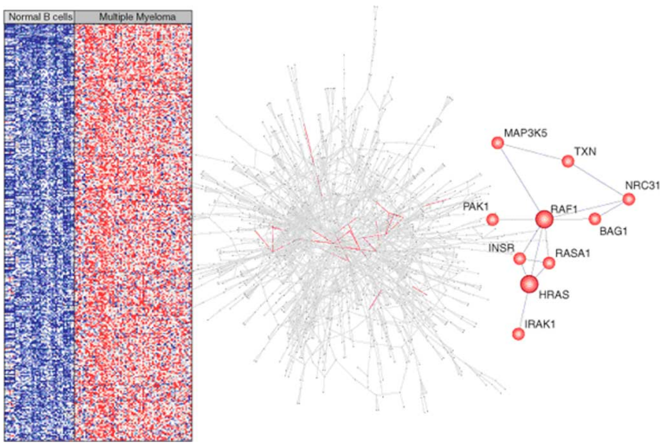


Fig. 4. Interpreting cancer gene-expression signatures (copyrighted Rhodes and Chinnaiyan [49]).

IV. Graphical Representation

A graphical representation refers to the visual interpretation of complex relationships in multidimensional data. It has two main components: intrinsic visualization (graph drawing) and extrinsic visualization (visual form). The former concentrates on drawing the graphical representation of the structures using effective graph layout algorithms while the latter focuses on displaying the virtual structures in appropriate visual forms so that the models can be easily understood and interacted with.

Graph drawing addresses the problem of generating the layouts of graphs automatically. Normally, graphs are depicted with their nodes as points in a plane and their edges as line or curve segments connecting those points. A graph layout algorithm reads as input a combinatorial description of a graph G, and produces as output a drawing of G according to a given graphic standard. Most graphic standards are generally accepted aesthetic criteria, such as distributing the nodes evenly in the frame, minimizing edge crossings, making edge lengths uniform, reflecting inherent symmetry, or conforming to the frame [23]. However, in almost all data presentation applications, the usefulness of a graph drawing depends on its readability, i.e., the capability of conveying the meaning of the diagram quickly and clearly [19].

A choice of visual form also plays an important role in graphical presentation. The same datasets can be visualized using various visual methods. Good visual presentations can effectively convey the key features of a complex structure to a wide range of users and audience, whereas poor ones may obscure the nature of an underlying structure [12]. As any human's ability to measure visual, auditory, tactile, and other stimuli is limited [44], visualization scientists try to harness the perceptual capabilities of the human visual system by choosing appropriate visual forms to display objects. Because of the large quantity and complexity of the microarray data, data analysis is frequently focused on a visualization that reflects a conventional or familiar perspective to the viewer with additional information content provided by highlights, e.g., colors, intensities, etc. There is no "right" way to visualize data as no single visual method can convey all the relevant features of the data, and users will need different visualizations depending on their research targets and previous experience. Consequently, biologists need to be able to access multiple visualization techniques that are geared toward answering different data analysis questions through the presentation of the data from a number of different perspectives.

In the next two sections, graph layout algorithms and visual methods that have been applied in microarray data visualization will be critically reviewed.

V. Graph Drawing

Graph drawing algorithms play a fundamental role in data visualization. They aim to map object relationships to spatial distances by positioning similar objects close to one another, and dissimilar objects far from each other. A great deal of work on the problem of graph layout has been carried out in recent years resulting in a number of sophisticated and powerful algorithms [17], [33], [37], [38], [54], [57]. To draw graphical models effectively, various graph layout algorithms are applied in the visualization of microarray data including *PCA*, *MDS*, *force-directed spring models* (FDSMs), *SOMs*, *minimum spanning tree* (MST), *treemaps*, and *Tang's mapping method*. Given their importance, they are briefly described as follows.

A. Principle Components Analysis

PCA is a standard method in data analysis and visualization [33] that determines linear transformation of a sample of points in an N-dimensional space that exhibits most clearly the properties of the sample along the coordinate axes. Along the new axes, the sample variances are extremes (maxima and minima) and uncorrelated. The new axes are defined as principle axes. According to Rencher's definition [48], the principal axes will include those along which the point sample has little or no spread (minima of variance). Hence, an analysis in terms of principal components can show (linear) interdependence in data. A point sample of N dimensions for whose N coordinates M linear relations hold will show only $(N\text{-}M)$ axes along which the spread is nonzero [9]. Using a cutoff on the spread along each axis, a sample may, thus, be reduced in its dimensionality (see Fig. 5).

The principal axes of a point sample are found by choosing the origin at the "center of gravity" and forming the dispersion matrix t_{ij}

$$t_{ij} = \frac{1}{N} \sum \lfloor (x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle) \rfloor$$

where the sum is over the N points of the sample, the x_i's are the ith components of the point coordinates, and $\langle \cdot \rangle$ stands for the average of the parameter. The principal axes and the variance along each of them are then given by the eigenvectors and associated eigenvalues of the dispersion matrix.

PCA tries to reduce the dimensionality of the data to summarize the most important parts while simultaneously filtering out noise. This is a well-understood and effective algorithm for computing the multidimensional projection that has been widely used to plot microarray data in 2-D or 3-D scatter plots to

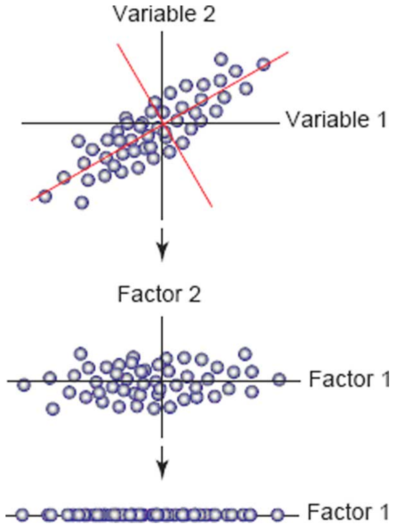


Fig. 5. PCA (copyrighted Gilbert and Schroeder [25]).

display similarly expressed gene groups. However, PCA cannot take into account nonlinear structures consisting of arbitrarily shaped clusters or curved manifolds since it describes the data in terms of a linear subspace. If the data set is highly nonlinear, it may be difficult to visualize it with linear projections on a low-dimensional display even if the projection angle is carefully chosen.

B. Multidimensional Scaling

MDS is a set of data analysis techniques that displays the structure of distance-like data as a geometrical picture [38]. In MDS visualization, the original q-axis and coordinates of points $\mu_i = (\mu_{i1}, \mu_{i2}, \ldots, \mu_{iq})^T$ do not enter the visualization directly. Instead, a configuration of points $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ is found in a space of lower dimension $p < q$, such that all interpoint distances $\|x_i - x_j\|$ match as closely as possible the original distance $\|u_i - u_j\|$. The result is usually a 2-D or 3-D configuration of points, each representing a single element from a data collection. Fig. 6 illustrates an example of the MDS result from 1352 genes. The lines connect genes or experiments that exhibit strong correlations (for example, red links have stronger correlations than do black lines). The coloring of the points expresses their correlation to the selected point.

The most common metric that has been used to evaluate how well a particular configuration reproduces the observed distance matrix is the stress measure. The raw stress value ϕ of a configuration is defined by the summary of stress between objects

$$\phi = \sum \left[d_{ij} - f\left(\delta_{ij}\right)\right]^2$$

where d_{ij} stands for the reproduced distances, given the respective number of dimensions, and δ_{ij} stands for the input data (i.e., observed distances). The expression $f(\delta_{ij})$ indicates a nonmetric, monotone transformation of the observed input data (distances). Thus, it will attempt to reproduce the general rank ordering of distances between the objects in the analysis. The smaller the stress value, the better is the fit of the reproduced distance matrix to the observed distance matrix.

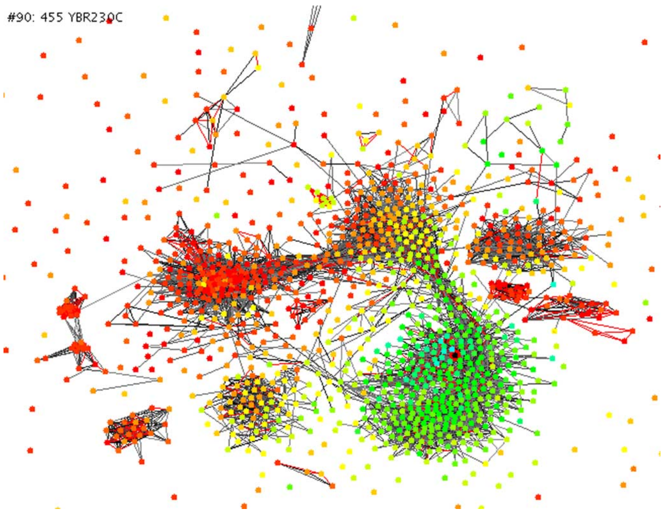


Fig. 6. MDS of 1352 genes (copyrighted Best *et al.* [7]).

The optimization process can be performed by applying a number of established function optimization heuristics such as Newton–Raphson, tabu search, genetic algorithm, and simulated annealing. A good review of these approaches can be found in [4].

The advantage of MDS over other multivariate visualization techniques is that it is independent of the number of variables. As long as it is possible to ascertain the high-dimensional distance between observations, a low-dimensional embedding can be found. However, since in the general case, it is not possible to map all distances accurately onto a lower dimensional space, it is quite possible that the MDS methods preserve most distances approximately and some distances poorly. In particular, in the metric MDS, the long distances will dominate over the shorter, local ones. Another problem with the algorithm is that it is computationally intensive for large datasets. However, the computational complexity can be reduced by restricting attention to a subset of the distances between the data items.

C. Self-Organizing Map

Kohonen's [37] SOM uses a simple analogy with the human brain's way of organizing information in a logical manner. As a visualization technique, the algorithm has been extended by a heatmap-based strategy for visualizing the U-matrix by Hautaniemi and Yli-Harja [29].

Unlike PCA and MDS, SOM is called a topology-preserving map since a topological structure is imposed on the nodes in the network. The map consists of two layers of neurons: an input layer and a competition/output layer (see Fig. 7). The weights of the connections from the input neurons to a single neuron in the competition layer are interpreted as a reference vector in the input space.

Kohonen's method [37] adopts the method of competitive learning with "winner takes all": when an input pattern is presented to the network, that neuron in the competition layer is

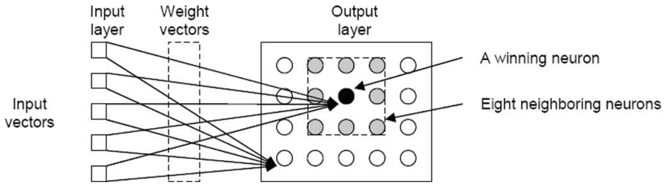


Fig. 7. Architecture of the SOM network.

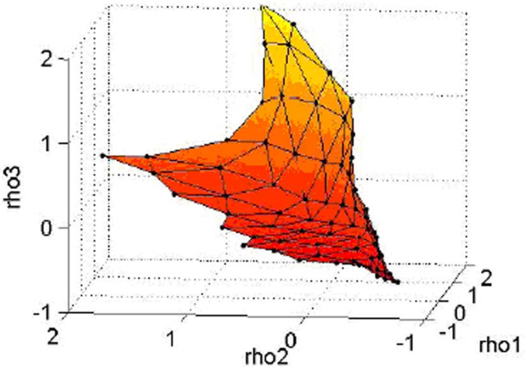


Fig. 8. Training the SOM for 3-D data.

determined, the reference vector of which is closest to the input pattern. This neuron is called the winner neuron, and it is the focal point of all the weight changes. The weights of the connections leading to the winner neuron will then be changed in such a way that the reference vector represented by these weights is moved closer to the input pattern.

In addition, there is a neighborhood relation defined on the competition layer, which indicates which weights of other neurons will also be changed. During the self-organizing procedure, the topologically close relationship of the organized information is maintained. Initially, a large area is treated in a similar fashion. Later in the iteration, this zoom shrinks. By virtue of its learning algorithm, the SOM forms a nonlinear regression of the ordered set of reference vectors into the input space. The reference vectors form an elastic network that follows the distribution of the data (see Fig. 8).

The advantage of SOM against other nonlinear projection methods is that SOM can preserve the topology—the local neighborhood relations. SOM tries to guarantee that items projected to nearby locations are similar, which means that the local order and local clustering structures shown on the map display are as trustworthy as possible. However, in terms of displaying global structure, SOM does not provide as accurate a layout as MDS. In SOM, the size of the map needs to be carefully decided since the computational complexity is determined by the number of map units. Besides the computational complexity, the existence of local minima in the cost functions may also cause problems [34].

D. Force-Directed Spring Model

Force-directed placement is a widely used method for drawing undirected graphs. The earliest force-directed placement model is based on the spring embedder model [17]. The main idea of this force-directed placement is to simulate physical–chemical models. In the heuristics, the nodes are considered as particles. Starting from an arbitrary initial position, the algorithm simulates the movements of the nodes in a physical–chemical model and lowers the energy stepwise such that the nodes come to rest.

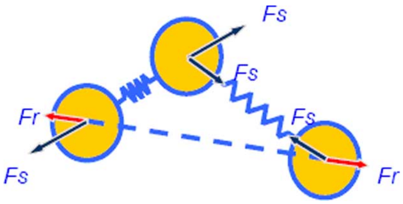


Fig. 9. Spring mode.

The spring model is based on a physical system in which the graph's edges are replaced by springs and the nodes are replaced by rings (see Fig. 9). The forces acting on every node include spring force F_s and repulsion force F_r.

The resultant of force F_s and F_r can be calculated by using the equations

$$F_s(d) = k_s \; \log(d)$$

$$F_r(d) = \frac{k_r}{d^2}$$

where k_s and k_r denote the current distance between a pair of nodes and the distance d is the length of the spring for the connected nodes. Among the parameters that control the forces acting on the nodes and cause their movements are spring length, spring stiffness, spring type, and the initial configuration. Under the influence of spring force between connected nodes and repulsion force between unconnected nodes, the graph will automatically calculate the position of each vertex until the system reaches a stable state.

Force-directed approaches normally produce a nice layout when drawing large graphs with cycles. This can be applied to drawing large gene networks in an aesthetically pleasing way. However, when linear and branched parts of a graph need to be drawn, they do not perform as well as other hierarchical layout algorithms. Therefore, it would be more appropriate to apply hierarchical layout algorithms when the size of graph is not very large. In addition, when the algorithm is applied to display clusters, force-directed approaches have to consider the graph as fully connected, i.e., all nodes are connected to each other making the algorithm extremely computationally demanding.

E. Minimum Spanning Tree

The MST algorithm was developed by Boruvka in 1926 [26]. It can be applied to display the similarity of genes to a hierarchical tree structure. Treating the hierarchical gene clusters as a graph, a spanning tree can be seen as an undirected connected acyclic spanning subgraph. Intuitively, a spanning tree for a graph is a subgraph that has the minimum number of edges for maintaining connectivity. If $w(e)$ is the weight for an edge e in a graph, then the weight of the tree is the sum of all the $w(e)$ in the tree. An MST is a spanning tree where the sum of

$w(e)$ is minimal. In other words, the MST of a graph defines the cheapest subset of edges that keeps the nodes in one connected component.

There are several algorithms available for computing the MTS [6]. One of the easiest to understand is Kruskal's greedy algorithm [38]. The algorithm works as follows: 1) sort the edges of graph G in increasing order by length; 2) keep a subgraph S of G, initially empty; 3) for each edge e in the sorted order, if the endpoints of e are disconnected in S, then add it to S. After all edges have been added to the subgraph, the graph is drawn.

In terms of visualizing tree structures, MST is fast and efficient. However, the resulting graph drawing may vary from one execution to another since the choice of the starting vertex of the algorithm is completely arbitrary in combination with the fact that, at some stage of the execution, some candidate edges may have the same weight, and therefore, the choice of edge may be different each time.

F. Treemaps

Treemaps are proposed by Johnson and Shneiderman as a space filling approach to the visualization of hierarchical trees [54]. Basically, any hierarchical structure can be thought of as a set of nodes, with each child node being some subset of its parent node. Treemaps attempt to take advantage of this comparison by drawing each child node enclosed within its parent. This leads to a collection of embedded rectangular bounding boxes, which readily shows the hierarchical structure of the information space.

A rectangular area is initially allocated to hold the representation of the tree. This area is then subdivided into a set of rectangles that represent the top level of the tree. This process continues recursively on the resulting rectangles to represent each lower level of the tree. The parent–child relationship is indicated by enclosing the child-rectangle by its parent-rectangle, that is, all descendants of a node are displayed as rectangles inside its rectangle. Associated with each node is a numeric value (e.g., size of a directory) and the size of a node's rectangle is proportional to that value (see Fig. 10) [32].

Treemaps efficiently utilize screen space by using all of the available display. The layout allows the user to easily comprehend the overall structure of the hierarchical tree. However, the lack of edges linking among nodes might prevent viewers from understanding the hierarchical structure of datasets. In addition, it is more computationally expensive than the classical tree drawing algorithms.

G. Tang's Mapping Methods

A novel mapping technique that projects n-dimensional vectors into 2-D points has been proposed by Tang *et al.* [57]. The mapping is similar to radial coordinate visualization techniques such as Radviz's [30]. The result preserves the correlation relationship between vectors in the output space. For example, two data points X and Y would be mapped to a radial line, given $X = aY$, where a is a scalar. All vectors with the same pattern will map to the same line, and the similar ones will map to a narrow region around (see Fig. 11).

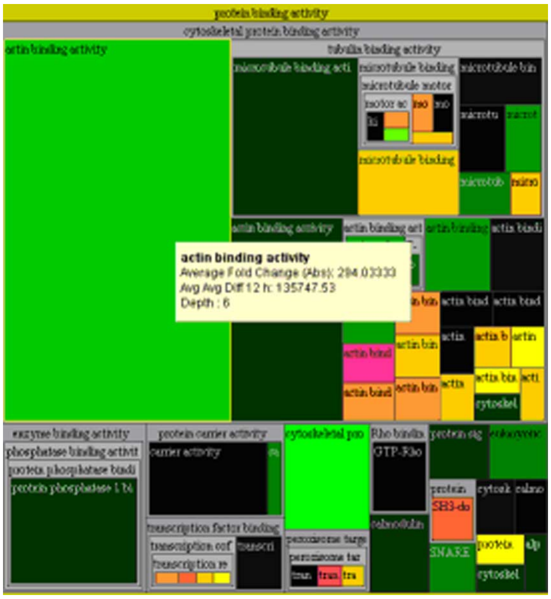


Fig. 10. Nineteen-level hierarchy as a complex directed acyclic graph that catalogues approximately 14 000 genes according to their biological functions.

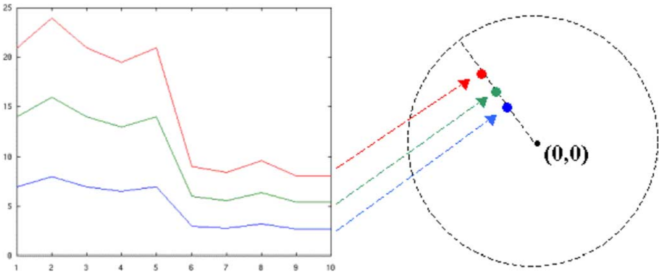


Fig. 11. Projection of the data points with same pattern.

Tang's approach maps a vector $\vec{P}_g = (x_{g1}, x_{g2}, \ldots, x_{gn})$ into a 2-D point $\vec{P}_g^*$ by following the formula

$$\vec{P}_g^* = \sum_{i=1}^{n} \left(w_i * \left(\frac{4}{n}\right) * (x_{gi}) \right) \vec{S}_i$$

where w_i is an adjustable weight for each coordinate, n is the vector length of the original space, $4/n$ is a ratio to centralize the points, and $\vec{S}_i$ $(i = 1, 2, \ldots, n)$ are the unit vectors that divide the center circle of the display screen equally.

Tang's approach was further extended by Zhang and Zhang [59] based on the first harmonic of the discrete Fourier transform (DFT), and implemented in the software VizStruct [59].

The algorithms are computationally efficient and display correlation effectively in a 2-D space. However, although this approach allows automatic adjustment of parameters, in order to get better results manual adjustments are still essential.

H. Discussion

In this section, various graph drawing algorithms that have been applied to microarray data visualization are discussed. Different algorithms can be applied in different circumstances of microarray data visualization depending on what kind of information the biologists are seeking to grasp during their studies. A graph layout method that works well for one research objective may not work well for another due to the differences in the properties of the data and the type of biological insight users are trying to gain.

Among all graph drawing methods, algorithms such as PCA, MDS, SOM, and Tang's mapping have more power in visualizing the group information of a number of genes. MST, FDSM, treemap, radial view, and hyperbolic view (see next section) are normally considered as candidates for drawing hierarchical gene clusters, and in particular, FDSM is one of the most appropriate algorithms for visualizing large gene networks.

Algorithms, such as PCA, MDS, SOM, and Tang's approach, try to project the multidimensional data onto a 2-D or a 3-D display so that a global view of the gene clusters can be found. The aforementioned approaches can successfully reduce the dimensionality of data to two or three dimensions so that each data point can be represented as a point in a 2-D or 3-D display. Each of these four algorithms has its own strengths. PCA displays the linear projection of the datasets and summarizes the most important parts while simultaneously filtering out noise. MDS and SOM can visualize nonlinear structures consisting of arbitrarily shaped clusters or curved manifolds where MDS provides a more accurate layout for global structure, while SOM attempts to display local structures as trustworthily as possible. An alternative method to displaying a global picture of gene groups is Tang's mapping approach. The algorithm is computationally efficient but manual adjustments of parameter values are essential in order to get better results.

MST, FDSM, and treemap can all be applied to draw hierarchical gene clusters. However, MST is one of the most frequently used of all these algorithms. The visualization draws gene clusters as dendrograms—a hierarchical tree view of the similarities between genes. FDSM and treemaps do not perform as well as MST in drawing hierarchical gene clusters. This is because FDSM is not good at drawing linear and branched parts of a graph, while in treemaps, the lack of edges linking among nodes might prevent viewers from understanding the hierarchical structure of the data. Other approaches such as the radial view and the hyperbolic view (see next section) can also be applied to draw hierarchical trees. However, these two techniques were initially designed to draw free trees rather than rooted trees with hierarchical structures; therefore, the hierarchical drawing can be misleading when the focus of observation is moved away from the root node.

Both MST and FDSM can also be applied to display interrelationships between genes as gene networks. MST is more computationally efficient. However, when a large network needs to be drawn, FDSM provides a nicer layout.

A brief summary of the strengths and weakness of graph drawing methods can be found in Table I.

TABLE I
COMPARISON OF GRAPH LAYOUT ALGORITHMS

	STRENGTHS	**WEAKNESSES**
PCA	- reduce the dimensionality of the data to summarize the most important parts whilst simultaneously filtering out noise	-difficult to visualize nonlinear structures consisting of arbitrarily shaped clusters or curved manifolds
MDS	- independent of the number of variables - provide an accurate layout when displaying a global structure	- not possible to map all distances accurately onto a lower dimensional space - may preserve most distances approximately and some distances poorly - computationally intensive
SOM	- can preserve the topology - local order and local clustering structures shown on the map display are as trustworthy as possible	- the size of the map need to be carefully decided - does not provide as accurate layout as MDS when displaying global structure
FDSM	- produce nice layout when drawing large graph	- when linear and branched parts of graph need to be drawn they do not perform as well as other hierarchical layout algorithms - less efficient if used as a mapping method
MST	-fast and efficient	- the visualization result may vary since the start node is arbitrary and some edges may have the same weights
Tree-maps	- produce nice layout when drawing large graph with hierarchical structure	-lack of edges linking between nodes, might prevent viewers from understanding the hierarchical structure - computationally expensive
Tang's Mapping	-computationally efficient	-need manual adjustments of parameter values in order to get better results

VI. VISUAL FORMS

A successful visualization strategy is a purposeful design intended to evoke cognitive relationships in the viewer [58]. Visual design is used to display data in ways that capitalize upon the particular strength of human pattern processing abilities [27]. In this section, we will examine the visual forms that have been widely used in gene expression data analysis, including 2-D or 3-D *scatter plots*, *color-coded heatmaps*, *parallel coordinates*, *dendrograms*, and others.

A. 2-D or 3-D Scatter Plots

The scatter plot is a standard tool for microarray visualization. This technique normally maps similarity between genes to a 2-D or 3-D virtual space to help biologists find clusters, outliers,

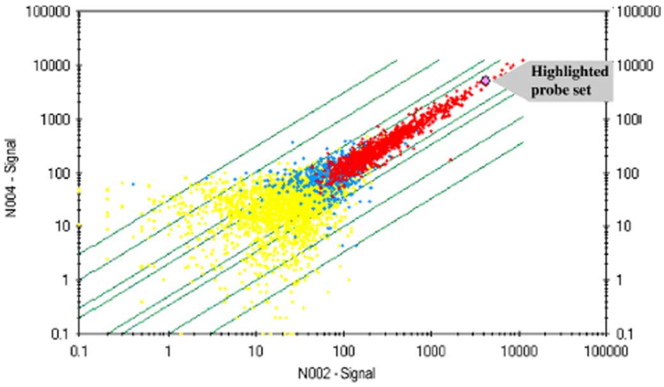


Fig. 12. Scatter correlation graph that highlights the probe selected in the pivot table.

trends, and correlations from data. Brushing and colored class points are widely used to gain additional insight into the data. Fig. 12 shows an example of a scatter plot visualization that highlights the selected probe.

A scatter plot can be extended by animation, shapes, glyphs, icons, colors, and interactions. It can also be extended to a higher dimension by a scatter plot matrix—a grid of scatter plots. The scatter plots matrix is useful for looking for all possible two-way interactions or correlations between dimensions during initial exploration of a data set.

Scatter plots display multivariate data onto a 2-D space. This allows the examination of the correlation between variables. The major difficulty in using this type of method in a data mining setting is that for a large data set, the display becomes overwhelming and incomprehensible. To detect more complicated relationships, more sophisticated methods need to be applied [27].

B. Color-Coded Heatmaps

A color-coded heatmap is a standard visual method for visualizing the multivariate gene expression datasets in a single display. The heatmap can be viewed as a variation of the parallel coordinates plot, in which color is used to convey dimension values.

In color-coded heatmap visualizations, the ratio of gene expression discussed in Section II is displayed using a color scale. For example, in most cases, black is used to indicate no change in expression, while an increase in the experimental relative to the control is shown as red, and a decrease in the experimental relative to the control is shown as green (see Fig. 13). Various color scales are applied in different software tools to display the patterns.

A color-coded heatmap provides a vivid view of gene expression patterns within a microarray experiment by displaying genes with similar induction or repression patterns close to each other. However, the hierarchical relationship between genes cannot be perceived unless the technique is combined with other visual method such as dendrogram trees.

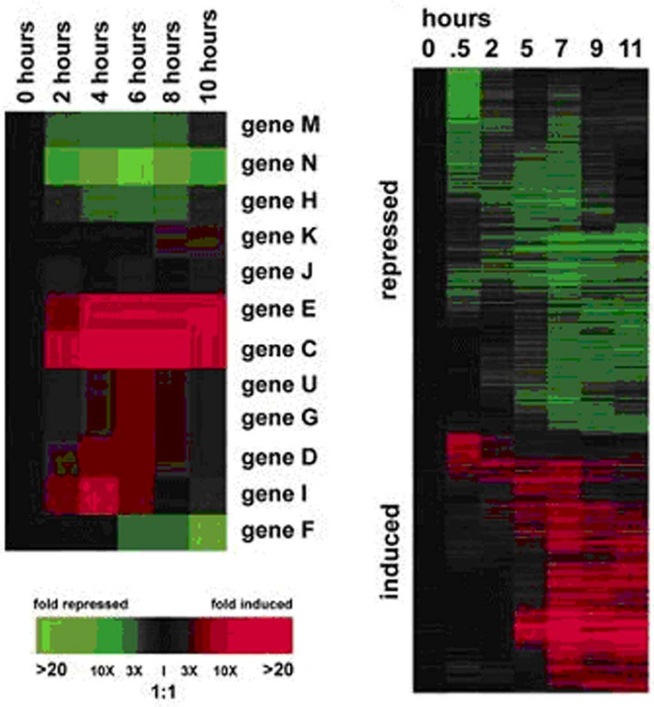


Fig. 13. Gene expression patterns visualized using color-coded heatmaps (image copied from [11]).

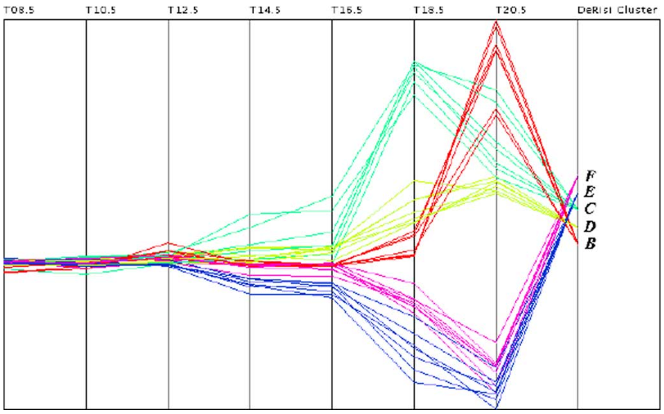


Fig. 14. Parallel coordinates visualization displaying gene expression levels for each cluster.

C. Parallel Coordinates

Parallel coordinates is another commonly used technique for visualizing gene expression patterns. It visualizes multidimensional datasets by arranging axes vertically, and by spacing them uniformly across the plane [31]. Each data point is displayed as a polygonal line connecting the corresponding abscissas on the parallel axes. The technique can be improved by ordering the axes (see Fig. 14).

Parallel coordinates can clearly display multivariables in one graph. In parallel coordinates, the relationship of two dimensions can be easily read if these two dimensions are assigned to neighboring axes. However, different orderings can produce different representations, and for parallel coordinates, the order in which the axes are drawn is arbitrary. In addition, it is

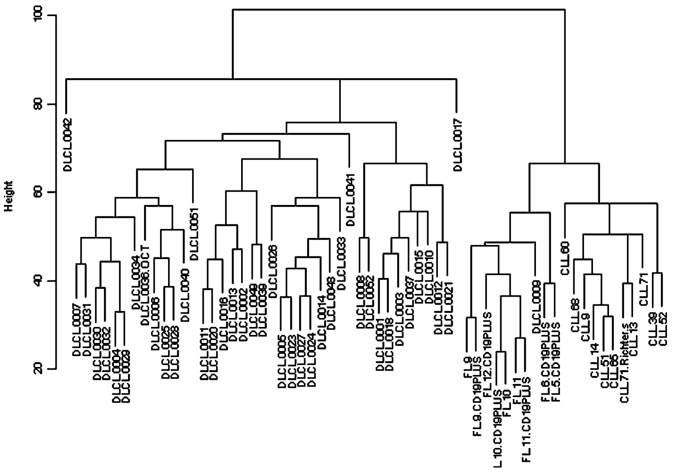
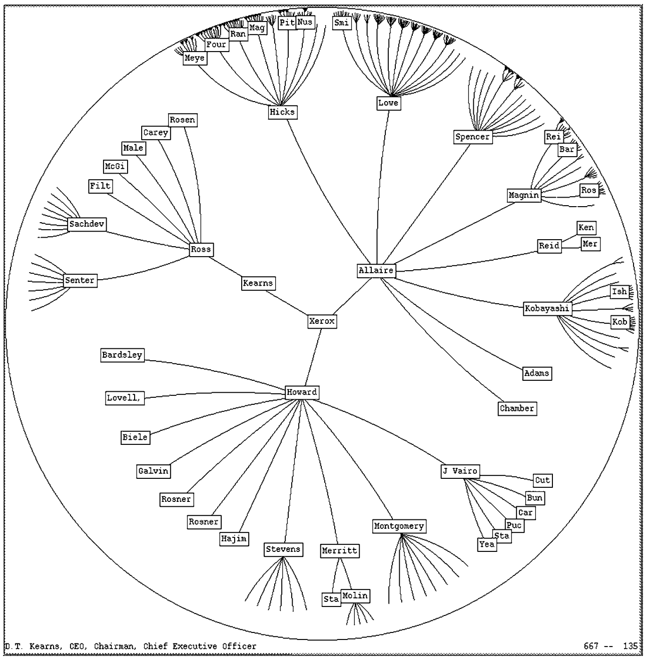


Fig. 15. Hierarchical cluster tree generated using Splus 2000 software.



D.T. Kearns, CEO, Chairman, Chief Executive Officer 667 -- 135

Fig. 16. Hyperbolic view.

hard to observe relationships between more than two dimensions since it is difficult to follow the line across several axes. Furthermore, the number of observations that can be displayed is limited by the size of screen space and the implementation used.

D. Dendrograms

Hierarchical clusters of genes are mostly drawn in two ways: either as a system of nested subsets, maximal in the sense that one cannot identify any additional subsets without violating the nesting, or as a binary tree with a distinguished root that has all the data items at its leaves. The former generates a structure called a *treemap* (see Section IV-E), whereas the latter generates *dendrograms*.

A dendrogram is a rooted tree with the root anchored centrally on the top of the image and the children nodes drawn downward using straight or bended lines (see Fig. 15). Some dynamic programming may be used to find the "most parsimonious" tree for a given set of sequences. For example, an MST can be used to obtain the graphical representation. Another dynamic programming problem arises from the problem of fitting an optimal dendrogram to a distance matrix.

Drawing a dendrogram is simple, fast, and predictable. However, the utilization of geometrical space is not optimized. There are many unused holes in the geometrical plane that waste a large portion of the display space. Furthermore, since all the leaves are shown at the same level in the drawing, the ordering of the leaves is arbitrary, as is their horizontal position. The heights of the internal nodes may also be arbitrary, depending on the metric information used to form the clustering. Such problems make it difficult for dendrograms to scale up well with large amounts of data.

E. Radial View and Hyperbolic View

Apart from the aforementioned techniques, a radial view and a hyperbolic view (see Fig. 16) may also be applied in tree visualization. The radial view [18] recursively positions children of a subtree into circular wedges. The angles of these wedges are proportional to the number of leaves of the subtree. The hyperbolic layout is a new technique developed by Lamping and Rao [39]. This layout technique constructs trees in a hyperbolic plane, and then, maps that structure into an ordinary Euclidean plane.

These two techniques can display a large data set in a relatively small space. However, they were initially designed to draw free trees rather than rooted trees with hierarchical structure in which hierarchical drawing can be misleading when the focus of observation is moved away from the root node.

F. Other Visual Methods

There are many other visual methods that can be applied to microarray data analysis to facilitate the discovery of features and relationships in data. For example, techniques such as zooming, panning, and jittering can improve visualization when too many points overlap or the data resolution causes many data points to lie at the same coordinate [28], [50]. Animation can be applied to display the pattern from different viewpoints. Shapes, glyphs, icons, colors, etc. can be used as extra dimensions of information space to visualize high-dimensional gene expression data [56], [60].

G. Discussion

In this section, various visual forms that have been applied to microarray data visualization are reviewed. As different visual forms may reveal different biological insights from data, there is no "best" way to visualize the data. Normally, several visual methods are combined together to enhance the power of visualization and display data mining results more effectively.

TABLE II
COMPARISON OF VISUAL FORMS

	STRENGTH	WEAKNESS
Scatter Plot	- allows the examination of the correlation between variables	- cannot detect complicated relationships
Color-coded Heatmaps	- display genes with similar induction or repression patterns close to each other - use colour to visualize the expression level	- hierarchical relationships between genes cannot be perceived unless the technique is combined with other visual methods
Parallel Coordinates	- can display multi-variables in one graph clearly - can be used to compare relationships between two dimensions	- the order in which the axes are drawn is arbitrary; different orderings can produce different representations - hard to observe relationships between more than two dimensions
Dendrogram	- simple, fast and predictable - display hierarchical structure effectively	- the utilization of geometrical space is not optimized - difficult to scale up well with large amount of data
Radial view & Hyperbolic view	- space-saving	- the hierarchical drawing can be misleading when the focus of observation is moved away from the root node

Among the aforementioned visual forms, the scatter plot is probably the best method to help biologists find clusters, outliers, trends, and correlations from data. Color-coded heatmaps and dendrograms are normally combined together to display hierarchical clustering results—the visualization provides not only the expression level of each gene among all experimental conditions but also the underlying structure of the data. Radial and hyperbolic views may perform similar tasks; however, their hierarchical drawing can be misleading when the focus of observation is moved away from the root node. In contrast to other visual forms, the strength of parallel coordinates is its ability to clearly draw multivariables as a series of line graphs in one display that makes it convenient to compare relationships between two dimensions, provided that the order of the axes is carefully arranged.

A brief summary of the strengths and weaknesses of each visual method can be found in Table II.

VII. CONCLUSION AND FUTURE WORK

The rapid advances in high-throughput technologies such as DNA microarray have resulted in a great demand for visualizing multidimensional expression data in an effective way so that interesting patterns, features, and relationships can be extracted from a large data set. Visualization of high-dimensional data involves a combination of structural modeling and graphical representations. Structural modeling forms a structure that characterizes a collection of datasets, while graphical representation provides visual interpretation of the complex relationships in a multidimensional data set.

The visualization techniques discussed in this paper are intended to show that there is a large and rich body of existing work that can be adopted or taken as a basis for further research. However, there is no "right" way to visualize microarray data, i.e., no single visualization can convey all the relevant features of the data. As a result, biologists need to be able to access multiple visualization techniques that are geared toward answering different data analysis questions through the presentation of the data from a number of different perspectives. There are several areas that can be improved.

1) Although much effort has been made to reduce the dimensionality of data before plotting each data item onto a computer display, the number of data items is not reduced. For a large gene expression dataset, a global view of all data points is often overcluttered, and consequently, some valuable information may be hidden from the viewer. A possible solution to tackle this problem is to apply some "focus + content" visualization techniques so that the points of interest can be shown in detail, and the overview of the graph is still preserved in the same window. Examples of such visualization techniques include the "fisheye view" [51] or "magnifying glasses" [35].
2) Time series expression experiments are becoming an increasingly popular method for studying a wide range of biological systems. However, visualization techniques that can be applied to display time series are under development but not yet available. Traditional line graph visualization can only provide a local view of gene expression patterns over time, i.e., only one or a few genes can be displayed at a time. Moreover, the dynamic aspect of gene expression patterns over time cannot be effectively visualized through classical presentations. To overcome these problems, various animation techniques may be applied to simulate the dynamic changes of gene expression patterns over time and provide a global view of all these changes.
3) Since most of the data mining and graph layout algorithms are computationally expensive, it is beneficial to speed up the structural modeling and graph drawing process by improving existing algorithms and applying them in the application domain so that a fast and efficient gene expression data analysis and visualization system can be achieved.
4) The integration of multiple visualizations is critical. Different visualization methods display different properties of data. Therefore, no single visual form can convey all the relevant features of a given dataset. How to integrate multiple visualization techniques to answer a particular biology question through the presentation of the data from a number of different perspectives is still a challenging task. For visualization researchers, it is important to have

a thorough understanding of the problem domain before designing an appropriate gene expression data analysis and visualization system.

5) Data integration is yet another important issue in biological research on microarray data. Information from a microarray experiment alone may not be sufficient for biologists to understand the underlying relationships among genes. Access to both internal and external database/knowledge base may provide a more integrated, "global" perspective that takes advantage of all available information.

References

[1] A. Alizadeh, M. Eisen *et al.*, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, pp. 503–511, 2000.

[2] D. Allison, X. Cui, G. Page, and M. Sabripour, "Microarray data analysis: From disarray to consolidation and consensus," *Nat. Rev. Genet.*, vol. 7, pp. 55–65, Jan. 2006.

[3] U. Alon, N. Barkai *et al.*, "Broad patterns of gene expression revealed by clustering analysis of tumour and normal colon tissues probed by oligonucleotide arrays," *Proc. Natl. Acad. Sci. USA*, vol. 96, pp. 6745–6750, Jun. 1999.

[4] W. Basalaj, "Proximity visualization of abstract data," Tech. Rep. 509, Comput. Lab., Univ. Cambridge, Cambridge, U.K., Jan. 2001.

[5] M. Benedikt, "Cyberspace: Some proposals," in *Cyberspace: First Steps*, M. Benedikt, Ed. Cambridge, MA: MIT Press, 1991, pp. 273–302.

[6] D. Bertsekas, *Linear Network Optimisation, Algorithms and Codes*. Cambridge, MA: MIT Press, 1991.

[7] C. Best, R. Zimmer, and J. Apostolakis, "Self-organized soft clustering, feature selection, and network inference using Gaussian processes," presented at the Int. Conf. Intell. Syst. Mol. Biol., Glasgow, U.K., 2004 (poster presentation).

[8] A. Bhattacharjee and W. Richards, "Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses," *Proc. Natl. Acad. Sci. USA*, vol. 98, pp. 13 790–13 795, 2001.

[9] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[10] A. Butte, "The use and analysis of microarray data," *Nat. Rev. Drug Discov.*, vol. 1, pp. 951–960, Dec. 2002.

[11] A. Campbell and L. Heye, *Discovering Genomics, Proteomics and Bioinformatics*. San Francisco, CA: CSHL Press/Benjamin Cummings, 2003.

[12] C. Chen, *Information Visualization and Virtual Environments*. London, U.K.: Springer-Verlag, 1999.

[13] C. Cummings and D. Relman, "Using DNA microarrays to study host–microbe interactions," *Genomics*, vol. 6, no. 5, pp. 513–525, Sep./Oct. 2000.

[14] C. Debouck and P. Goodfellow, "DNA microarrays in drug discovery and development," *Nat. Genet.*, vol. 21, pp. 48–55, Jan. 1999.

[15] J. Derisi, L. Penland, P. Brown, M. Bittner, P. Meltzer, M. Ray, Y. Chen, Y. Su, and J. Trent, "Use of a cDNA microarray to analyse gene expression patterns in human cancer," *Nat. Genet.*, vol. 14, no. 4, pp. 457–460, Dec. 1996.

[16] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: From co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[17] P. Eades, "A heuristic for graph drawing," *Congr. Numerantium*, vol. 42, pp. 149–160, 1984.

[18] P. Eades, "Drawing free trees," *Bull. Inst. Combinatorics Appl.*, vol. 5, pp. 10–36, 1992.

[19] P. Eades and R. Tamassia, "Algorithms for drawing graphs: An annotated bibliography," *Comput. Geom.: Theory Appl.*, vol. 4, no. 5, pp. 235–282, 1994.

[20] M. Eisen, P. Spellman *et al.*, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 14863–14868, 1995.

[21] O. Ermolaeva, M. Rastogi *et al.*, "Data management and analysis for gene expression arrays," *Nat. Genet.*, vol. 20, pp. 19–23, 1998.

[22] R. Ewing and J. Cherry, "Visualization of expression clusters using Sammon's non-linear mapping," *Bioinformatics*, vol. 17, no. 7, pp. 658–659, 2001.

[23] T. Fruchtermann and E. Reingold, "Graph drawing by force-directed placement," *Softw.—Pract. Exp.*, vol. 21, no. 11, pp. 1129–1164, 1991.

[24] D. Gershon, "Microarray technology: An array of opportunities," *Nature*, vol. 416, pp. 885–891, 2002.

[25] D. Gilbert and M. Schroeder, "Interactive visualization and exploration of relationships between biological objects," *Trends Biotechnol.*, vol. 18, no. 12, pp. 487–494, Dec. 2000.

[26] R. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Ann. Hist. Comput.*, vol. 7, pp. 43–57, 1985.

[27] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. Cambridge, MA: MIT Press, 2001.

[28] H. Hauser, F. Ledermann, and H. Doleisch, "Angular brushing of extended parallel coordinates," in *Proc. IEEE Symp. Inf. Vis. (InfoVis 2002)*, Oct. 2002, pp. 127–130.

[29] S. Hautaniemi and O. Yli-Harja, "Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps," *Mach. Learn.*, vol. 52, no. 1/2, pp. 45–66, 2003.

[30] P. Hoffman and G. Grinstein, "DNA visual and analytic data mining," in *Proc. IEEE Vis. 1997,* Phoenix, AZ, Oct., pp. 437–441.

[31] A. Inselberg, "The plane with parallel coordinates," *Vis. Comput.*, vol. 1, pp. 69–91, 1985.

[32] B. Johnson and B. Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information structures," in *Proc. IEEE Vis. (2nd Conf. Vis. 1991)*, pp. 284–291.

[33] I. Jolliffe, *Principle Component Analysis*. New York: Springer-Verlag, 1986.

[34] S. Kaski, "Data exploration using self-organizing maps," Ph.D. thesis, Helsinki Univ. Technol., Helsinki, Finland, 1997.

[35] T. Keahey and E. Robertson, "Techniques for non-linear magnification transformations," in *Proc. IEEE Symp. Inf. Vis.*, 1996, pp. 38–45.

[36] J. Khan, J. Wei *et al.*, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nat. Med.*, vol. 7, pp. 673–679, 2001.

[37] T. Kohonen, *Self-Organization and Associative Memory*. New York: Springer-Verlag, 1984.

[38] J. Kruskal and M. Wish, *Multidimensional Scaling*. Beverly Hills, CA: Sage, 1978.

[39] J. Lamping and R. Rao, "The hyperbolic browser: A focus + context technique for visualizing large hierarchies," *J. Vis. Lang. Comput.*, vol. 7, no. 1, pp. 33–55, 1995.

[40] D. Liu and B. Yao, "Comparative evaluation of microarray analysis software," *Mol. Biotechnol.*, vol. 26, no. 3, pp. 225–232, Mar. 2004.

[41] D. Lockhart, H. Dong *et al.*, "Expression monitoring by hybridization to high-density oligonucleotide arrays," *Nat. Biotechnol.*, vol. 14, pp. 1675–1680, 1996.

[42] B. Lockhart and E. Winzeler, "Genomics, gene expression and DNA arrays," *Nature*, vol. 405, pp. 827–836, 2000.

[43] G. S. Michaels, D. B. Carr *et al.*, "Cluster analysis and data visualization of large-scale gene expression data," in *Proc. Pacific Symp. Biocomput. 1998*, pp. 42–53.

[44] G. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychol. Rev.*, vol. 63, no. 2, pp. 81–96, 1956.

[45] C. A. Orengo, D. T. Jones, and J. M. Thornton, *Bioinformatics*. Oxford, U.K.: BIOS Scientific, 2003.

[46] J. Quackenbush, "Computational analysis of microarray data," *Nat. Rev. Genet.*, vol. 2, pp. 418–427, 2001.

[47] S. Ramaswamy, K. Ross *et al.*, "A molecular signature of metastasis in primary solid tumours," *Nat. Genet.*, vol. 33, pp. 49–54, 2003.

[48] A. C. Rencher, *Methods of Multivariate Analysis*, 2nd ed. Wiley Series in Probability and Statistics. New York: Wiley, 2001.

[49] D. Rhodes and A. Chinnaiyan, "Integrative analysis of the cancer transcriptome," *Nat. Genet.*, vol. 37, pp. S31–S37, 2005.

[50] P. Saraiya, C. North, and K. Duca, "An evaluation of microarray visualization tools for biological insight," in *Proc. IEEE Symp. Inf. Vis. (INFOVIS 2004),* Oct., pp. 1–8.

[51] M. Sarkar and M. Brown, "Graphical fisheye views," *Commun. ACM*, vol. 37, no. 12, pp. 73–84, Dec. 1994.

[52] M. Schena, D. Shalon *et al.*, "Quantitative monitoring of gene-expression patterns with a complementary-DNA microarray," *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[53] W. Shannon, R. Culverhousel, and J. Duncan, "Analyzing microarray data using cluster analysis," *Pharmacogenomics*, vol. 4, no. 1, pp. 41–51, 2003.

[54] B. Shneiderman, "Tree visualization with tree-maps: 2-D space-filling approach," *ACM Trans. Graph.*, vol. 11, no. 1, pp. 92–99, 1992.

[55] D. Slonim, "From patterns to pathways: Gene expression data analysis comes of age," *Nat. Genet. Suppl.*, vol. 32, pp. 502–508, Dec. 2002.

[56] T. Sprenger, R. Brunella, and M. Gross, "H-BLOB: A hierarchical visual clustering method using implicit surfaces," in *Proc. 11th Annu. IEEE Vis. Conf. (Vis 2000)*, pp. 61–68.

[57] C. Tang, L. Zhang, and A. Zhang, "Interactive visualization and analysis for gene expression data," in *Proc. Hawaii Int. Conf. Syst. Sci.*, Big Island, HI, Jan. 2002, vol. 6, pp. 143–151.

[58] E. Tufte, *Visual Explanations*. Cheshire, CT: Graphics Press, 1997.

[59] L. Zhang and A. Zhang, "VizStruct: Exploratory visualization for gene expression profiling," *Bioinformatics*, vol. 20, no. 1, pp. 85–92, Jan. 2004.

[60] L. Zhang, W. Sheng, and X. Liu, "3D visualization of gene clusters and networks," in *Proc. SPIE Vis. Data Anal. (VDA 2005),* Jan. 2005, vol. 5669, pp. 316–326.

Leishi Zhang received the B.A. degree in information management from Anhui University, Hefei, China, in 1995, the M.Sc. degree in applied computing from the University of Dundee, Dundee, U.K., in 2003, the M.Phil and Ph.D. degrees in bioinformatics visualization from Brunel University, Uxbridge, U.K., in 2004 and 2007, respectively.

She is currently working as a research associate in Computing Laboratory, University of Kent, Canterbury, U.K. Her current research interests include computer graphics, graph layout, 3-D visualization, and intelligent data analysis.

Jasna Kuljis received the Dipl.Ing. degree in theoretical mathematics from the University of Zagreb, Zagreb, Croatia, in 1977, the M.S. degree in information science from the University of Pittsburgh, Pittsburgh, PA, in 1986, and the Ph.D. degree in information systems from the London School of Economics, London, U.K., in 1995.

She is currently a Professor of computing in the School of Information Systems, Computing and Mathematics, Department of Information System and Computing, Brunel University, Uxbridge, U.K., where she is also the Director of the People and Interactivity (Pandi) Research Centre. Her current research interests include human–computer interfaces and the development of new paradigms that would further enhance the usability of interactive computer systems. She is the Managing Editor of the journal *Information Visualization*.

Xiaohui Liu received the B-Eng. degree in computer software, Hohai University, Nanjing, China, in 1982, and the Ph.D. degree in computer science from Heriot-Watt University, Edinburg, U.K., in 1988.

He is currently a Professor of computing at Brunel University, Uxbridge, U.K., where he is also the Director of the University Research Centre for Intelligent Data Analysis (IDA), School of Information Systems, Computing and Mathematics, Department of Information System and Computing, and is engaged in performing interdisciplinary research involving artificial intelligence, dynamic systems, image and signal processing, and statistics, particularly for applications in biology, engineering, and medicine. He is on the editorial boards of four computing journals, and founded the biennial international conference series on IDA in 1995. He has given numerous invited and keynote talks in bioinformatics, data mining, and statistics conferences.