GAP: A graphical environment for matrix visualization and cluster analysis

Han-Ming Wu [a] Yin-Jing Tien [b] Chun-houh Chen [c,*]

[a] *Department of Mathematics, Tamkang University, Taipei County 25137, Taiwan*

[b] *Institute of Statistics, National Central University, Taoyuan 32001, Taiwan*

[c] *Institute of Statistical Science, Academia Sinica, Taipei 11529, Taiwan*

Abstract

GAP is a Java-designed exploratory data analysis (EDA) software for matrix visualization (MV) and clustering of high-dimensional data sets. It provides direct visual perception for exploring structure of a given data matrix and its corresponding proximity matrices, for variables and subjects. Various matrix permutation algorithms and clustering methods with validation indices are implemented for extracting embedded information. GAP has a friendly graphical user interface for easy handling of data and proximity matrices. It is more powerful and effective than conventional graphical methods when dimension reduction techniques fail or when data is of ordinal, binary, and nominal type.

Key words: Clustering; Data visualization; Exploratory data analysis; Heat map; Java; Matrix visualization; Seriation; Software; Statistical graphics.

1 Introduction

Graphical exploration is a preliminary yet essential step in exploratory data analysis and statistical modeling. The boxplot, histogram and scatterplot have served as major tools in the statistics and machine learning communities for more than 30 years. Quite often, these traditional graphical techniques are equipped with various dimension reduction methods and computer-aided interactive functionalities. Although they are useful for exploring data structure, they often lose effectiveness when it comes to visual exploration of information structure embedded in high dimensional data sets. With striking advances

* Correspondence author: Tel.: +886-2-27835611-407; fax: +886-2-2783-1523.
Email address: cchen@stat.sinica.edu.tw (Chun-houh Chen).

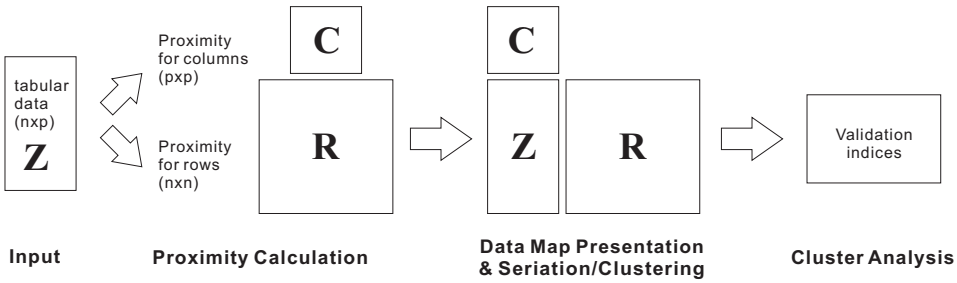

Fig. 1. Architecture of the GAP analysis procedures.

in computing, communication, and high-throughput biomedical instruments nowadays, data sets of relatively large numbers of variables or large sample sizes are generated with more complex structures. As a consequence, more sophisticated visualization techniques and environments that support the efficient, effective and practical exploration of high dimensional data sets are needed.

Bertin (1967) proposed the concept of matrix visualization as a reorderable matrix for systematically presenting data structures and relationships. Over the past few decades, much attention has been devoted to visualizing the raw data matrix (subjects by variables), while little work has been carried out on visualizing the corresponding proximity matrices (subjects by subjects, variables by variables). A detailed review of MV techniques can be found in Wu, Tzeng and Chen (2007). Regarding implementation, a number of MV-related software analogs are available, particularly in the field of bioinformatics for studying microarray gene expression data. They were developed either for exploring the raw data matrix only (color histogram of Wegman (1990); data image of Minnotte and West (1998); Treeview of Eisen *et al.* (1998)) or proximity matrices only (Ling, 1973; Murdoch and Chow, 1996; corrgrams of Friendly (2002)). Chen (1996, 1999, and 2002) integrated visualization for the raw data matrix with two proximity matrices (for variables and samples) into the framework of generalized association plots (GAP). The term, matrix visualization, is therefore referred to as a graphical technique for visualizing and exploring, simultaneously, the associations of subjects, variables and their interactions, without dimension reduction. This color-based representation of re-ordered data matrices tries to display tabular quantities and relationships in a natural and intuitive way for gaining valuable insights into the underlying information.

In this paper, we describe the design and features of a novel exploratory data analysis software package, GAP, for matrix visualization and clustering. GAP is written in Java and implements matrix visualization and various clustering algorithms (e.g., hierarchical clustering, k-means, rank-two elliptical seriation) for interactive exploration of data matrices. Figure 1 depicts the design architecture of the standard GAP analysis procedures. Firstly, two proximity matrices for rows and columns of an input table of data (could be one of

continuous, binary, ordinal or nominal data types) are calculated using user-specified proximity measures. Three matrix maps are then constructed through a suitable color projection. After applying some clustering or seriation algorithms for rows and columns if necessary, the patterns and clusters are found and validated further with different characteristics of criteria. In addition, a wide-range of algorithms and functionalities are provided and operated either directly on the original data matrices or on the two proximity matrices for more comprehensive data exploration. Our design goals of the development were driven by (1) ease of use; (2) flexibility for dealing with different data types; (3) interactive data exploration; (4) platform independence; and (5) state-of-the-art graphing and clustering algorithms. The users have the capability of direct visual mining of the data matrix with the two proximity matrices. GAP, equipped with modern computing power and display, has great potential for visually exploring structure that underlies massive and complex data sets and is intended as a routine EDA tool for general purpose data analysis. To our knowledge, this package is the first publicly available GUI software for implementation of integrated matrix visualization and cluster analysis.

The paper is structured as follows. Section 2 introduces the main procedures of matrix visualization under the framework of generalized association plots, including the system architecture and designed objects. The permutation and clustering algorithms and the cluster validation indices are given in Section 3. Section 4 describes some unique features and extensions. Some modules which extend MV techniques for statistical data analysis including the special application and implementation of MV techniques for visualizing cDNA Microarrays are presented in Section 5. We conclude the article with some perspectives on MV techniques in Section 6.

2 Main procedures of matrix visualization

We summarize the main steps of matrix visualization in terms of calculation, presentation and permutation. The developed visualization method will be presented following the three aspects of Keim (2001): (1) the data to be visualized; (2) the visualization technique; and (3) the interaction technique. For illustration purposes, a subset of the Harvard lung cancer microarray data set **B** described in Bhattacharjee *et al.* (2001), is employed. The subset consists of 30 randomly sampled adenocarcinomas (AD) patients (rows) with 14 marker genes (columns). The samples come with one discrete covariate, gender, and one continuous covariate, age. The 14 marker genes have been clustered in three functional categories by Bhattacharjee *et al.* (2001). In this paper, we only demonstrate the analysis procedures for the continuous type raw data matrix in GAP. The same procedures can be applied to binary, ordinal or nominal data types.

Table 1
Data objects for GAP matrix visualization.

Matrix	Object	Symbol	Dimension	Description
Data	`gap`	$\mathbf{Z} = \{z_{ij}\}$	$n \times p$	a tabular data set
Proximity	`gaprow`	$\mathbf{R} = \{r_{ij}\}$	$n \times n$	proximity for rows (subjects)
	`gapcol`	$\mathbf{C} = \{c_{ij}\}$	$p \times p$	proximity for columns (variables)
Covariate	`gapYd`	$\mathbf{Y_d} = \{y^d_{ij}\}$	$n \times q_d$	discrete covariates for rows
	`gapYc`	$\mathbf{Y_c} = \{y^c_{ij}\}$	$n \times q_c$	continuous covariates for rows
	`gapXd`	$\mathbf{X_d} = \{x^d_{ij}\}$	$m_d \times p$	discrete covariates for columns
	`gapXc`	$\mathbf{X_c} = \{x^c_{ij}\}$	$m_c \times p$	continuous covariates for columns

2.1 Calculation of proximity matrices

2.1.1 The input data table and missing values imputation

The first step of a GAP analysis is the calculation of two proximity matrices, $\mathbf{R}$ and $\mathbf{C}$, for n rows (subjects) and p columns (variables) of a given data matrix $\mathbf{Z}$ with the user-selected similarity (or dissimilarity) measures. Quite often, the collected data may come with additional qualitative or quantitative information, which we refer to as discrete or continuous covariates. We denote $\mathbf{X_d}$, $\mathbf{X_c}$, $\mathbf{Y_d}$ and $\mathbf{Y_c}$ as discrete covariates and continuous covariates for columns and rows, respectively. Since the n and p are treated symmetrically in GAP, we will denote $\mathbf{D}$ as a general proximity matrix for representing either $\mathbf{R}$ or $\mathbf{C}$. Table 1 lists a short description for these seven data matrices to be visualized and manipulated. All the data matrices are stored in a single input file with ASCII format.

Since the scale of the data under study significantly impacts the effectiveness of visualization, a pre-transformation is in general needed to obtain a comparative scale for both numerical and visual considerations. For example, a log transformation reduces outlier effects and enhances overall comprehension of visualization. Some basic mathematical transformations such as log, power, centering are implemented in GAP.

Another processing issue for the input data is missing values. Missing values are allowed and can be coded by the user's preference such as "NA" or "NULL". For further statistical modeling of data containing missing values, GAP includes several imputation methods for continuous type data such as row or column averages, weighted k-nearest neighbors, and singular value decomposition (SVD) (Troyanskaya *et al.*, 2001). Besides being capable of handling

Table 2
Input data type and the corresponding proximity measures provided by GAP.

Data type	Proximity measures
Continuous	Covariance, Euclidean distance (L2), Kendall's τ, Pearson's correlation coefficient r, Spearman's rank, City-block (L1), Absolute(r), Uncentered r, Absolute(uncentered r).
Binary	Hamman, Jaccard, Phi, Rao, Rogers, Simple match, Sneath, Yule.
Ordinal	Kendall's τ_b, Kendall-Stuart's τ_c, Goodman-Kruskal's γ, Somers's d, Wilson's e.
Nominal	Sakoda's contingency coefficient, Goodman-Kruskal τ, Cohen's κ (I=J), Simple match.

a raw data matrix of continuous, binary, ordinal or nominal type, GAP admits direct input of a similarity or distance matrix prepared by the user.

2.1.2 Proximity measures for variables and subjects.

The associations among rows (subjects) and columns (variables) are constructed through the computation of similarity or dissimilarity measures, such as Pearson's correlation coefficient or Euclidean distance. The choice of proximity measure strongly affects association patterns for variables and subjects directly, and the interaction structure in the data matrix indirectly. Table 2 lists some commonly adopted association measures for continuous, binary, nominal and ordinal scales of data. For the lung cancer example, Pearson's correlation is calculated for genes and Euclidean distance for patients. For potential nonlinear relationship, nonlinear geodesic distances (see Section 5.2) such as Isomap (Tenenbaum *et al.* (2000)) can be utilized. On the other hand, for data with missing values, proximities are calculated using pairwise (columns or rows) complete observations.

2.2 Presentation of the data matrices

Seven aforementioned numerical matrix objects are then projected through suitable color spectra as matrix maps, where each numerical entry is represented by a color dot. This is the most fundamental step in matrix visualization. The graphical layout of each data object is arranged and integrated into a single plot as shown in Figure 2. Dendrograms for rows and columns are shown as additional objects when hierarchical clustering methods are adopted (see Section 2.3.2 for a detailed description). This design enables the overall

visual comprehension of all related information in a single window. Java 2D is the major graphical device for designing the viewer, `gapDisplay`, for tabular data matrices.

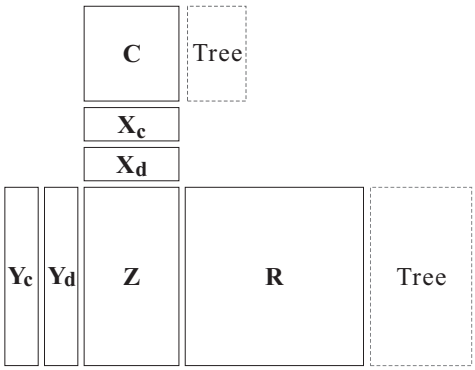

Fig. 2. Graphical layout of data objects in GAP matrix visualization.

2.2.1 Color spectra

The selection of an appropriate color spectrum is usually a subjective process in data and information visualization. The principle in selecting a suitable color spectrum is to ideally project the numerical properties of the data into a corresponding meaningful visual perception. GAP provides several commonly used color spectra and a dialog for designing the user-defined color spectra. Table 3 summarizes characteristics of the available color spectra in GAP.

In an MV plot, a missing value can be simply displayed at the corresponding position (row and column) with a color that can be easily distinguished from the color spectrum of the numerical values. With appropriate permutations for rows and columns, the corresponding variable/sample combinations of missing structure can be visually accessed. The GAP users have a simple yet informative visual perception of the missing mechanism (random or not, ignorable or nonignorable) of the data (variables).

2.2.2 Displaying conditions of the raw data map

Changing the display of colors is similar to select transformations for numerical values. The default display condition in GAP is the *range matrix condition*, where the whole color spectrum is used to represent the complete range of values in the raw data matrix. For a bidirectional color spectrum (green-black-red for gene expression profiling, blue-white-red for correlation coefficients), the *centered matrix condition* enforces the color spectrum to be symmetric around the baseline numeric value (1:1 for $\log_2$ ratio gene expression, zero for correlation coefficient), see Figure 3 (a). On occasion, one might want

Table 3
Color schemes for data objects.

Spectrum type	Property	Name	Spectrum	Suitable data objects
Unidirectional	spectral color	`rainbow-130`		sequential $\mathbf{Z}$, $\mathbf{R}$, $\mathbf{C}$, $\mathbf{Y_c}$,
	single-hue progression	`gray-256`		and $\mathbf{X_c}$.
Bidirectional	diverge from a pivot	`expression-38`		diverging $\mathbf{Z}$ (gene expression),
	point	`correlation-200`		diverging $\mathbf{R}$, $\mathbf{C}$ (correlation, covariance).
Nominal	depict category ra- other than continuity	`category-16`		nominal $\mathbf{Z}$, $\mathbf{Y_d}$, $\mathbf{X_d}$.
Binary	almost all color spectra can be used to color code binary data matrix or proximity matrix.			
2- or 3-hue diverging	the user can create arbitrary linear color spectrum by specifying starting, midpoint, and ending color triplet (r,g,b).			

to down-weight the effects of extreme values in the data set, and the use of ranks as a replacement for numerical values is one possibility. This is termed the *rank matrix condition*. The matrix condition can be easily converted to row or column conditions by a mouse click for contrasting individual variable distributions or subject profiles.

2.3 Permutation of data matrices

The structures embedded in the raw data map and the two proximity maps can be extracted only with some suitable permutations for the matrices. Therefore, it is necessary to permute the matrices such that subjects with similar profiles are placed in neighboring rows and variables with common distribution patterns in neighboring columns. We have implemented several seriation algorithms.

2.3.1 Ellipse ordering

GAP features the elliptical seriation algorithm (Chen, 2002), R2E, which utilizes the property of a converging sequence of iteratively formed correlation matrices. When the sequence reaches an iteration with numerical rank of the correlation matrix equal to two, the p objects fall on the surface of a two-dimensional ellipsoid (ellipse) and have unique relative positions on the ellipse. Elliptical seriation is very effective for identifying global clustering patterns and smooth transitional profiles (Tien *et al.*, 2008) which optimize the Robin-

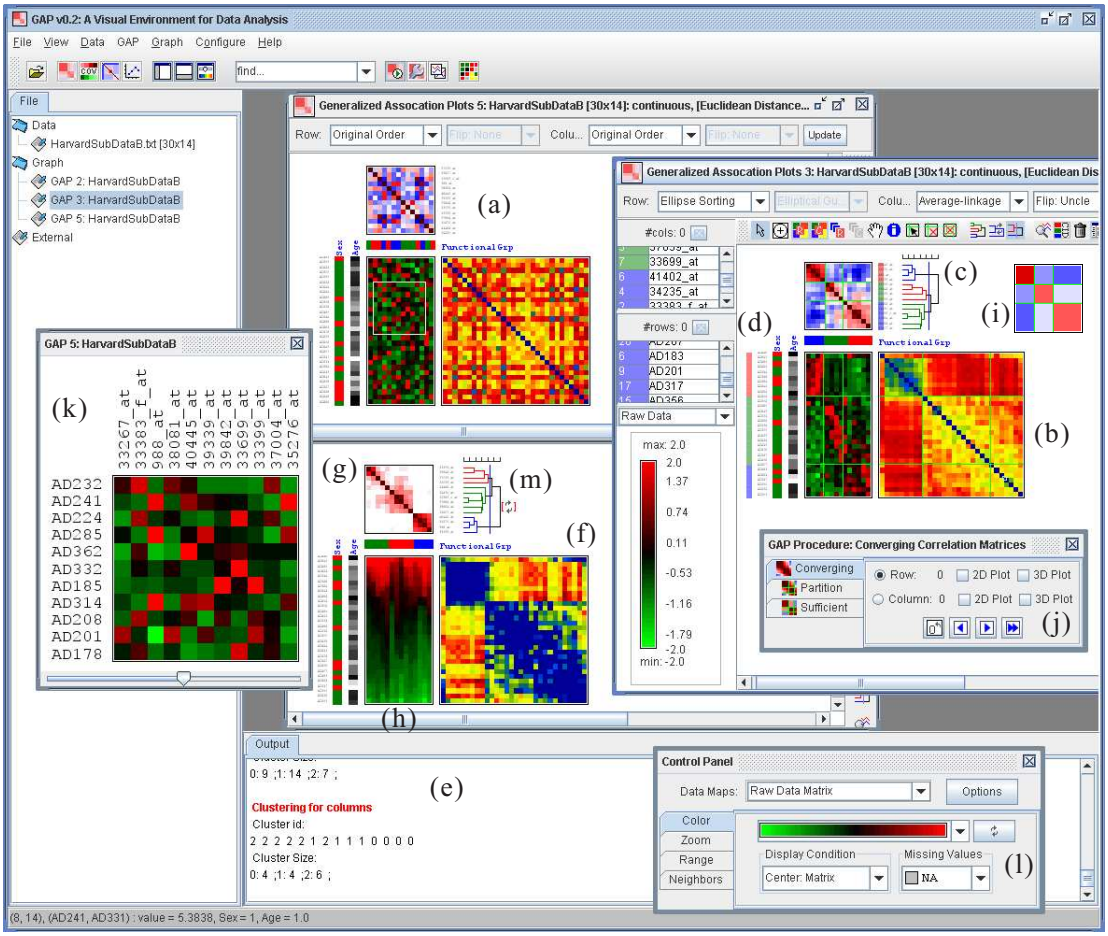

Fig. 3. The standard multi-window operating environment of GAP. This figure demonstrates the GAP GUI using a subset of Dataset **B** from Harvard Lung Cancer Dataset (Bhattacharjee *et al.*, 2001). There are 30 randomly selected patients (rows) of lung adenocarcinoma (AD) with 14 marker genes (columns). (a) The unsorted raw data map and two proximity maps with clinical records on gender and age, on the left. (b) The sorted Euclidean distance matrix for patients. (c) The dendrogram for genes with a user-selected partition. (d) The resulting clusters for patients. (e) The output window. (f) The restricted display for patient Euclidean distance matrix. (g) The sectional display for the genes correlation matrix. (h) The gene sediment display for the expression profile matrix. (i) The mean sufficient display for (c). (j) The GAP converging procedure panel. (k) The zooming window. (l) The control panel. (m) Manual flip of an intermediate node of dendrogram.

son criterion (Robinson, 1951) (see Section 3.1 for a detailed description). This is a unique feature of GAP. R2E can be applied to any given proximity matrix **D**, be it correlation, covariance, Euclidean distance, or other proximity matrix for objects. For example, the Euclidean distance for patients in the lung cancer data is sorted by the relative ellipse shown in Figure 3(b). The rank-two ellipse seriation for obtaining one-dimensional ordering is as follows. Let ϕ be the Pearson's correlation operator and $\{R^{(k)}, k = 1, 2, \cdots\}$ be the sequence of sample Pearson's correlation matrices.

Algorithm of R2E:

(1) Initial: set the initial ordering of p objects as $\{o_1, o_2, \cdots, o_p\}$ and $R^{(0)} = \mathbf{D}$.
(2) Iterative process: $R^{(k)}(\mathbf{D}) = \phi(R^{(k-1)}(\mathbf{D})), k = 1, 2, \cdots$.
(3) Stopping criterion: stop at the κ-th iteration for which the numerical rank of $R^{(\kappa)}(\mathbf{D})$ equals two. (All p column/row vectors of $R^{(\kappa)}(\mathbf{D})$ fall on an ellipse and have unique relative positions on it.)
(4) Splitting: find a cut c such that two successive objects have the largest gap.
   - For each object, o_i in $R^{(\kappa)}(\mathbf{D})$, compute its angle relative to the origin as a_i.
   - Sort the angles to get the relative positions of p objects on the ellipse, $\{a_{(i)}, i = 1, \cdots, p\}$. The corresponding sorted objects are $\{o_{(i)}, i = 1, \cdots, p\}$.
   - A cut c is obtained with the largest gap between two successive objects

$$c = \arg\max_i \{a_{(i+1)} - a_{(i)}\}.$$

(5) Reordering: the ellipse ordering is $\{o_{(c+1)}, o_{(c+2)}, \cdots, o_{(p)}, o_{(1)}, \cdots, o_{(c)}\}$.

Note that the ordering of R2E is successive, that means the maps can be moved back and forth in pursuing possible better visualizations. This algorithm has also been implemented in R (R Development Core Team, 2005) in the seriation package by Hahsler, Hornik and Buchta (2008).

2.3.2 Tree seriation and flipping mechanism

Agglomerative hierarchical clustering is the most popular permutation algorithm for gene expression profiles. One sorts columns and rows of an expression profile matrix using the relative orders of the leaves (terminal nodes) of the corresponding dendrograms constructed for genes and for arrays. One critical but often neglected issue using the leaves of the dendrogram in sorting the rows/columns of a raw data matrix is the flipping of the intermediate nodes. The $n-1$ intermediate nodes for a dendrogram of n objects (subjects or variables) can be flipped independently resulting in 2^{n-1} different permutations for the n objects using identical proximity matrix and tree linkage methods. GAP comes with two internal methods and two external methods to guide the flipping mechanism of the intermediate nodes for the single-, complete-, average- and centroid-linkage tree methods.

The first internal flipping mechanism, the *uncle* approach, is implemented to flip the intermediate node $\mathrm{n}_{(.)}$ such that the distance (in the sense of average-linkage) between its left daughter node to its brother node is larger than the distance between its right daughter node to its brother node. The following

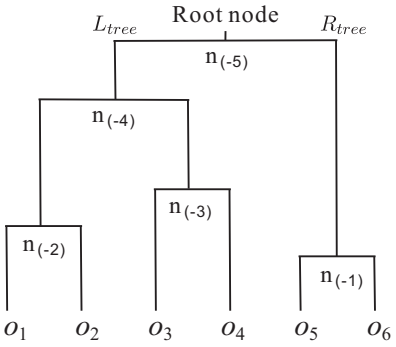

Fig. 4. Intermediate node flipping mechanism of a hierarchical clustering tree.

are some *uncle* flipping examples using a dendrogram of six objects shown in Figure 4.

- Flip the node $\mathrm{n}_{(-2)}$, if $\mathrm{dist}(\{o_1\}, \mathrm{n}_{(-3)}) \leq \mathrm{dist}(\{o_2\}, \mathrm{n}_{(-3)})$.
- Flip the node $\mathrm{n}_{(-3)}$, if $\mathrm{dist}(\{o_3\}, \mathrm{n}_{(-2)}) \leq \mathrm{dist}(\{o_4\}, \mathrm{n}_{(-2)})$.
- Flip the node $\mathrm{n}_{(-1)}$, if $\mathrm{dist}(\{o_5\}, \mathrm{n}_{(-4)}) \leq \mathrm{dist}(\{o_6\}, \mathrm{n}_{(-4)})$.

Figure 3(c) shows the average-linkage tree with the uncle flipping for genes. We see that the permutation result matches well the functional categories of genes.

Another internal flipping mechanism is the *grandpa* approach. Assume the intermediate node $\mathrm{n}_{(\cdot)}$ to be considered for flipping stemmed from the left branches of the root node (denoted by L_{tree}). We shall flip the node such that the distance between its left daughter node to the right branches of the root node (denoted by R_{tree}) is larger than the distance between its right daughter node to R_{tree}. Here are some *grandpa* flipping examples.

- Flip the node $\mathrm{n}_{(-2)}$, if $\mathrm{dist}(\{o_1\}, \{R_{tree}\}) \leq \mathrm{dist}(\{o_2\}, R_{tree})$, where $R_{tree} = \mathrm{n}_{(-1)}$.
- Flip the node $\mathrm{n}_{(-3)}$, if $\mathrm{dist}(\{o_3\}, \{R_{tree}\}) \leq \mathrm{dist}(\{o_4\}, R_{tree})$, where $R_{tree} = \mathrm{n}_{(-1)}$.
- Flip the node $\mathrm{n}_{(-1)}$, if $\mathrm{dist}(\{o_5\}, \{L_{tree}\}) \leq \mathrm{dist}(\{o_6\}, L_{tree})$, where $L_{tree} = \mathrm{n}_{(-4)}$.

Another way for guiding the intermediate node flipping mechanism is through the guidance of some external referencing list which can be imported by the user. The external reference methods make the tree seriation result as close to the external reference as possible. For example, a hierarchical clustering dendrogram guided by the R2E seriation will simultaneously preserve coherent local clusters as well as smooth global grouping structure of the data. Comparison of various seriation methods for matrix visualization can be found in Tien *et al.*, (2008).

2.3.3 Other sorting algorithms

GAP provides options to sort the data maps according to the user's input reference list or row/columns means. For example, a reference list can be obtained from other clustering algorithms (e.g., DIVCLUS-T (Chavent et al., 2007)). This flexibility strengthens the GAP environment through integrating the users' prior knowledge and merits of other clustering algorithms/packages. It is also possible to reverse or randomize the ordering of the dendrograms and data maps. This function is especially useful for interactive teaching of hierarchical clustering tree methods.

2.4 Non-continuous data

For non-continuous data where multiple instances take the same value, conventional displays suffered from overstrikes of data points representing the value (scatterplot type displays) or overstrikes of line segments connecting values of neighboring variables (parallel coordinate plot) (Inselberg, 1985; Wegman, 1990). On the other hand, GAP has a significant advantage over conventional graphical displays on non-continuous data for the following reasons.

GAP directly converts every single numerical value in a data matrix into one color dot in a data map no matter if the data is of continuous, ordinal, binary or nominal scale. The same color spectrum, gray-256 (Table 3) for example, can be used to represent continuous, ordinal or binary data with a suitable partitioning scheme. Colored column-stripes in a GAP raw data map act as stacked histograms (continuous data), or bar/pie-charts (ordinal, binary, and nominal data) while row-bands represent profiles of samples (see Section 4.1 sediment display for more discussion). For a non-continuous variable, only a fixed number (number of categories) of hues are used to code the whole column-stripe with n sample dots. Areas of hues representing proportions of numerical values can be easily assessed after suitable permutation of samples and variables.

The problem becomes more complicated for purely nominal data. Although all the nominal colors in the category-16 (Table 3) spectrum can be used to represent different categories in every individual nominal variable, the combination of colors/variables in the whole data map makes the GAP display incomprehensible. The issue of color-coding for purely nominal variables will be studied in a separate article.

3 Cluster Analysis

Cluster analysis is defined here in a broad sense. That is, the validation indices are implemented both for the ordering produced by a seriation algorithm and the clusters made by a clustering algorithm. These indices measure the goodness of a seriation or a clustering.

3.1 Seriation validation: anti-Robinson criteria

Proposed by Robinson (1951), a Robinson Matrix, $R = [r_{ij}]$, is a symmetric matrix such that $r_{ij} \leq r_{ik}$ if $j < k < i$ and $r_{ij} \leq r_{ik}$ if $i < j < k$. If rows and columns of a symmetric matrix T can be sorted such that it becomes a Robinson matrix, we call T pre-Robinson. Three anti-Robinson loss functions (Streng, 1978) are implemented for each permuted proximity matrix, $D = \{d_{ij}\}$, for assessing the amount of deviation from a Robinson form with distance-type proximity:

$$AR_{\mathrm{n}} = \sum_{i=1}^{n} [\sum_{j<k<i} I(d_{ij} < d_{ik}) + \sum_{i<j<k} I(d_{ij} > d_{ik})],$$
$$AR_{\mathrm{S}} = \sum_{i=1}^{n} [\sum_{j<k<i} I(d_{ij} < d_{ik}) \cdot |d_{ij} - d_{ik}| + \sum_{i<j<k} I(d_{ij} > d_{ik}) \cdot |d_{ij} - d_{ik}|],$$

and

$$AR_{\mathrm{W}} = \sum_{i=1}^{n} [\sum_{j<k<i} I(d_{ij} < d_{ik}) |j-k| |d_{ij} - d_{ik}| + \sum_{i<j<k} I(d_{ij} > d_{ik}) |j-k| |d_{ij} - d_{ik}|].$$

AR_{n} counts only the number of anti-Robinson events in the permuted matrix; AR_{S} sums the absolute value of anti-Robinson deviations; AR_{W} is a weighted version of $AR(s)$ penalized by the difference of column indices of the two entries. In order to compare the performance of different sorting algorithms, the generalized anti-Robinson loss function is defined as the number of deviations from the Robinson form,

$$GAR(w) = \sum_{i=1}^{n} [\sum_{(i-w) \leq j < k < i} I(d_{ij} < d_{ik}) + \sum_{i<j<k \leq (i+w)} I(d_{ij} > d_{ik})],$$

where w is the window-size defining the range of summation. Window-size is the number of columns (rows) away from the diagonal of $\mathbf{D}$ that we consider in calculating the anti-Robinson events. Small w refer to criteria for considering

only local behaviors, and larger window-sizes refer to criteria for more global relationships between subjects. In order to have better comparison among different seriation algorithms for small window-sizes, we define the relative generalized anti-Robinson loss function as

$$RGAR(w) = \frac{\sum_{i=1}^{n}[\ \sum_{(i-w)\le j<k<i} I(d_{ij} < d_{ik}) + \sum_{i<j<k\le(i+w)} I(d_{ij} > d_{ik})\ ]}{\sum_{i=1}^{n}[\ \sum_{(i-w)\le j<k<i} 1 + \sum_{i<j<k\le(i+w)} 1\ ]}.$$

$RGAR(w)$ ranges between 0 (no anti-Robinson events) to 1 (all anti-Robinson events).

3.2 Cluster validation

The sorted matrix maps are generally capable of displaying the raw data structure and the association patterns among subjects and variables. The users are usually more interested in identifying clusters in the permuted matrix maps. One can do this using the dendrogram branching structure. Figure 3 (c) illustrates one such example with a threshold to split genes into three clusters. The user can also manually select the subtrees to make clusters. In addition, the inspection of converging R2E-sorted correlation matrices gives a clue for finding splitting points in the sorted raw data maps (see Section 4.2). Once the clusters have been determined (Figure 3 (d)), GAP reports the cluster indices in the output windows (Figure 3 (e)). The cluster validation indices are helpful for choosing the appropriate number of clusters in the data. Four internal validation measures are implemented for reflecting the compactness, connectedness, and separation of the cluster partitions: connectivity (Handl et al., 2005); the Dunn index (Dunn, 1974); within-cluster variance; and Silhouette width (Rousseeuw, 1987). Furthermore, when the user has an external reference partition, the Rand index, adjusted Rand index (Rand, 1971), Jaccard coefficient, Minikowski (Hubert and Arabie, 1985) are also available for evaluating the degree of agreement.

4 Unique features of GAP

In the complete procedure of matrix visualization, GAP provides many unique and interactive features to assist the users in interacting with all data objects that are not available in other matrix visualization and hierarchical clustering tree software.

4.1 Extending MV displays

4.1.1 Restricted display.

Quite often outliers in the raw data or proximity matrices may exhaust the resolution of the selected color spectrum and mask the overall visual impression. The GAP users can improve this situation by displaying ranks of the data instead of original magnitudes, or by compressing the color spectrum for representing only certain portions of the data range (Figure 3 (f)).

4.1.2 Sectional display.

GAP uses the sectional display to show only those numerical values that satisfy certain criteria in the original data or proximity maps. The users may decide to ignore the values below some threshold by not displaying corresponding hues of color (Figure 3 (g)). One can also choose to emphasize more coherent neighboring structure by displaying only the surrounding neighbors along the main diagonal of a sorted distance map in a dynamic fashion.

4.1.3 Sediment display.

Another useful feature of GAP is the sediment display which can be constructed by individually sorting each subject (variable) vector according to the ascending (descending) order of numerical magnitudes. This display contrasts the distribution structure for all subjects (variables) simultaneously. The interpretation is analogous to a side-by-side bar-chart and box-plot (Figure 3 (h)).

4.1.4 Sufficient display.

With the partitioned matrix maps, GAP provides the sufficient display (Chen, 2002) through a summary statistic (mean, median or standard deviation) for each identified subject-subject, variable-variable and subject-variable blocks. The three sufficient (partitioned) maps summarize the information inherent in the data matrix and the corresponding proximity matrices (Figure 3 (i)).

4.2 Converging correlation matrices

The R2E algorithm (Chen, 2002) improves the singular value decomposition (SVD) method by extracting the elliptical structure of the converging sequence of iteratively formed correlation matrices using eigenvalue decomposition. The

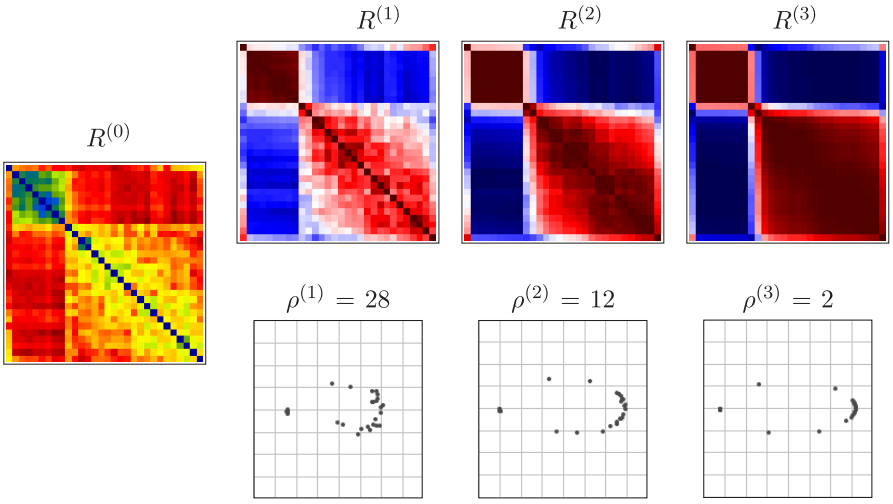

Fig. 5. Sorted correlation maps for the thirty subjects and 2D plots of the first two eigenvectors for the converging sequence of correlation matrices at selected iterations.

converging sequence of the correlation matrices gradually merges relative information structure of the minor (3rd and beyond) eigenvectors into the leading two vectors. At the iteration with rank two, there are only two eigenvectors left with non-zero eigenvalues, and information is reduced to the ellipse spanned by the first two eigenvectors. This property of the converging process serves as two major data exploration mechanisms, one is for identifying the splitting points of sorted proximity maps to make clusters, another is for identifying potential outliers which do not fit well into the elliptical structure with the first two eigenvectors. Figure 5 shows the sorted correlation maps for the thirty subjects and 2D plots of the first two eigenvectors for the converging sequence of correlation matrices at selected iterations. While all the correlation coefficients gradually approach the two extremes of +1 and -1, the leading two eigenvectors gradually form an elliptical structure. The `GAP Procedure` panel shown in Figure 3 (j) controls the converging process, the map partition approaches and the construction of the sufficient maps.

4.3 Manual operations

In addition to the specific tools we have just described, GAP also provides other utilities for data navigation. For examples, the action of mouse brushing/selection on the map can select subsets of the data for zooming (Figure 3 (k)) and further operations. Three parameter-sliders are designed to rescale map resolutions in the vertical, horizontal or both directions. Together with the operations on colors (e.g., color switch, reversion) and the extending displays, tools are collected in a panel, called the `Control Panel`, in (Figure 3 (l)). The dialog of finding similar patterns allows the users to input a pre-

Table 4
Function classes of interaction technique on data objects.

Matrix			Function	Description
Data	Proximity	Covariate		
√	-	-	`Imputation`	missing data imputation algorithms.
√	-	-	`Scale`	scaling for raw data object display condition.
√	√	√	`gapDisplay`	a class for viewing tabular data object.
√	√	√	`Sorting`	sort rows and columns of data objects according to various criteria.
√	√	√	`Color`	functions with reverse, restricted and sectional display.
√	√	√	`Zoom`	a zooming window, zoom in and out in vertical, and horizontal directions.
√	√	√	`Split`	splitting lines.

Table 5
Function classes of interaction technique on tree objects.

Function	Description
`Flip`	flip nodes according to the input or manually operation.
`Color`	mouse drag to color the tree branches.
`Zoom`	rescale dendrogram.
`Select`	mouse click to select subtree.

specified pattern for sorting the rows of the data matrix according their similarities (distance or correlation) to the input pattern.

For interacting with the dendrogram, the users can select subtrees of a dendrogram by a mouse click. Moreover, GAP allows the users to partition the dendrogram by a mouse drag (Figure 3 (c)), and manually flip any intermediate node of the dendrogram by a mouse click (Figure 3 (m)). These are very useful functions for educational purposes.

During the visualization process, the permuted numeric data matrices and images (matrix maps, dendrograms) can be exported in several common file formats (jpg, bmp, png, eps) for further manipulation. Table 4 and 5 lists some Java classes of interaction techniques on data and dendrogram objects for matrix visualization.

5 Modules

Some statistical data analysis tools such as the K-means (Hartigan and Wong, 1979) clustering and principal components analysis (PCA) have been implemented in GAP while the following functionalities are still in their implementation and testing stages.

5.1 Covariates adjustment

In addition to the design matrix for data analysis the users may as well collect covariates such as gender, age or phase of cell cycle that may have some effects on an MV visual pattern analysis. Covariate adjustment has to be taken into consideration in order to reveal the masked visual patterns. We are incorporating the within and between analysis (WABA) (Dansereau *et al.*, 1984) for discrete covariates and partial correlation for continuous covariates, into the GAP environment (Wu and Chen, 2007). In this framework, the Pearson correlation matrix for variables can be decomposed into the model- and residual-component matrices. The contribution of the covariate effects can then be assessed through the relative structure of the model-component to the original correlation matrix while the residual-component becomes the new data matrix for further exploration. A z score map is proposed to identify variable pairs with the most significant differences in correlations before and after a covariate adjustment.

5.2 Geometric nonlinear association

The aforementioned similarity or dissimilarity measures for quantitative data quantify only linear relationships between subjects and variables. In many scientific applications, features may be correlated in a nonlinear manner. The isometric mapping (isomap) (Tenenbaum *et al.,* 2000) algorithm is developed to estimate the geodesic distance between all pairs of points in the data manifold. We have implemented the estimation of this geodesic distance in GAP as *iso-distance* for extracting potential geodesic nonlinear structures.

Figure 6 shows a Swiss-roll data cloud which consists of 500 expression-38-colored points. This is a typical data set for studying the nonlinear manifold problem. The sorted L2-distance and iso-distance matrices sorted by the R2E seriation are shown in Figure 7 (b) and (d) respectively. As can be seen, the sorted L2-distance map could not reflect the non-linear structure (Figure 7 (a) and (b)) while the iso-distance map fully recovered the Swiss-roll non-linear manifold (Figure 7 (c) and (d)).

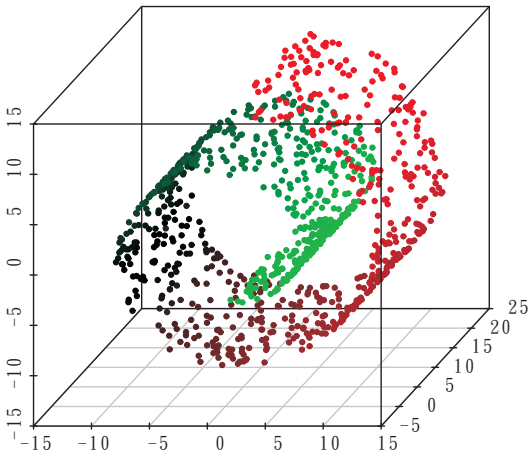

Fig. 6. A nonlinear manifold structure, the Swiss-roll example.

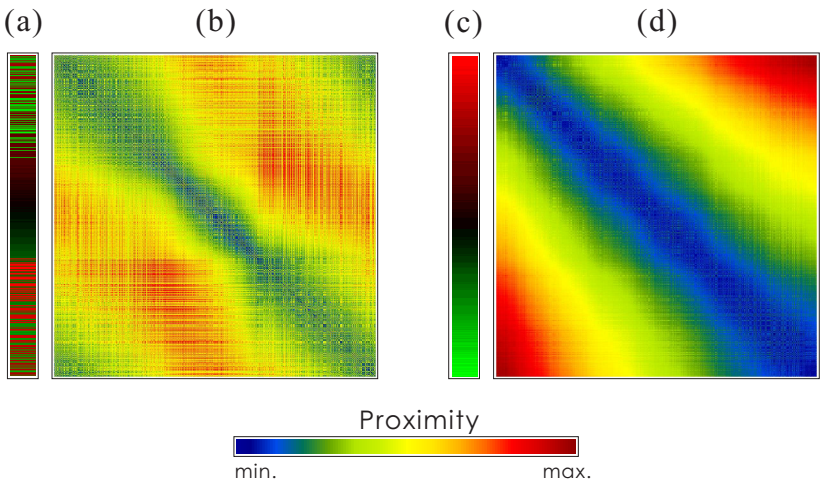

Fig. 7. Matrix visualization of geodesic nonlinear association. (a) The sorted sample indices in Figure 6 with the corresponding (b) L2-distance matrix for Swiss-roll data by R2E algorithm. (c) The sorted sample indices with the corresponding (d) iso-distance matrix by R2E algorithm.

5.3 Array image viewer

Visual inspection of the spotted cDNA microarrays images is the first step in quality control for statistical analysis of gene expression profiling. The array image viewer module in GAP offers a pseudo-colour representation of an array for displaying array design information (e.g., Block, Column, Row, Dia., F635 Mean, B635 Median, F532 Mean, B532 Median, $\cdots$) stored in `gpr` file (GenePix Pro, Molecular Devices Corporation) in the form of the physical array layout and sample plates location map. This is a special application and implementation of MV techniques for visualizing cDNA microarrays. The developed functions enable the users to obtain an instant overview of the design quality of the experiment and for detecting scratches, artefacts and spatial patterns of the arrays. It works on multiple `gpr` files sharing an identical GAL

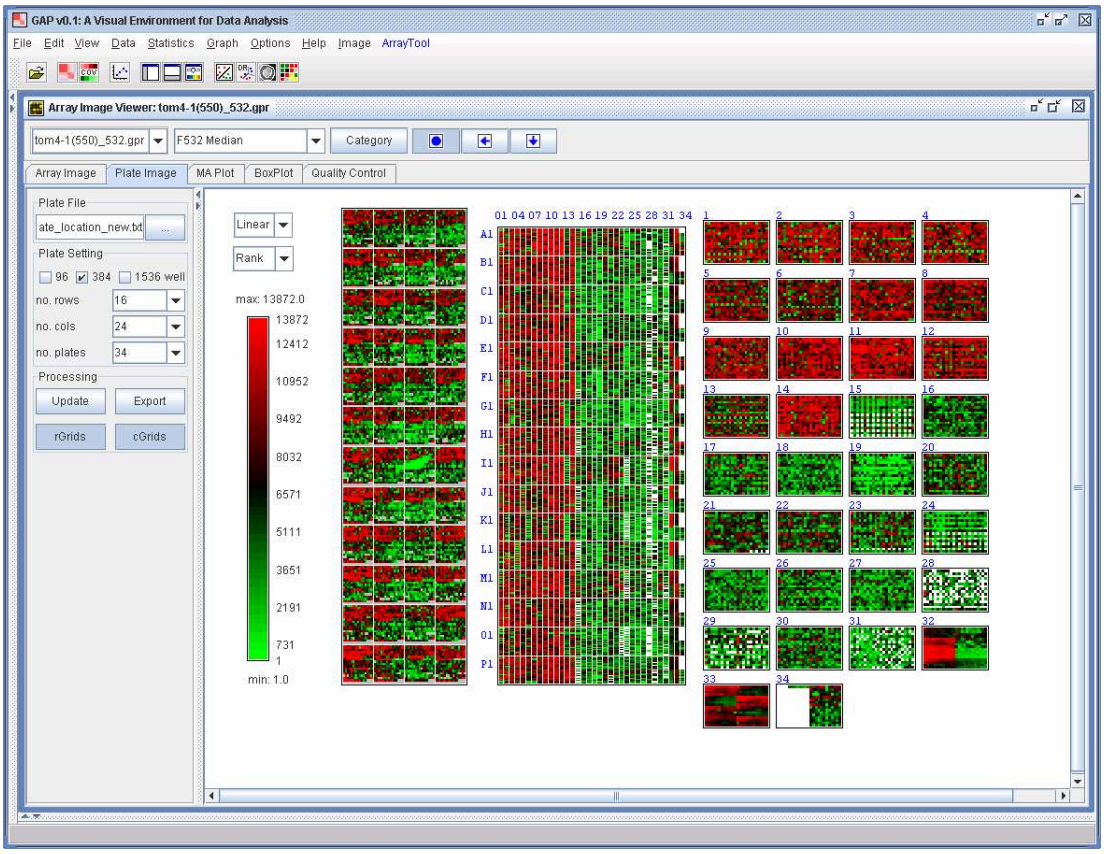

Fig. 8. A snapshot of the array image viewer module in GAP.

file.

Figure 8 has a snapshot of the array image viewer with some real array data. The physical design of the array has 12 by 6 grids with 17 by 17 spots in each grid. The additional numerical information for locations of spots, the sample plates the spots came from, and the sample providers/laboratories are stored and imported. There are 34 plates with the 384-well (16 by 24) layout. Figure 8 shows the "F532 Median" values with the rank matrix condition. This is helpful in contrasting the relative expression intensities with related design information. The color map in the left panel is the pseudo-colored array. The top halves of all grids have systematic higher expression intensities than the lower halves. In the central grid of the array, a possible scratch (in bright green) can be easily spotted. The middle panel displays the map of plates aligned along 384 wells (A1∼A16, $\cdots$, P1∼P16). The physical plate maps in the right panel reveal the message that relative higher expression intensities of the top-halves of the grids are actually associated with the first fourteen plates which contains only purchased commercial samples. The in-house prepared sample plates all contain relatively lower expression intensities. More example array viewer images are available on our web site.

6 Conclusion and outlook

This paper illustrates the design and features of the GAP environment. GAP provides simple yet powerful tools for visually exploring and understanding the structure embedded in high-dimensional data matrices before suitable mathematical operations and appropriate statistical modeling can be introduced for accomplishing a comprehensive data analysis. The users can use the built-in data sets to learn related GAP modules and techniques. In the past few years, several scientific studies using GAP as primary analysis, visualization, and presentation tool have been successfully carried out and published in the fields of psychiatry (Hwu, *et al.,* 2002), microarray data analysis (Lee, *et al.,* 2005) and others (e.g., Fielding, 2007).

Computationally, the R2E algorithm is more time consuming than other seriation methods. It takes a personal computer (Celeron (R) 3.2 GHz CPU with 512 MB RAM) running C++ on Windows XP about 0.09 sec, 9.09 sec, and 2.71 hr to obtain the R2E permutations for proximity matrices with 50, 500, and 5000 rows/columns. The computation complexity for R2E is of order $O(n^3)$. The computing speed is much slower in the current pure Java version GAP package although we are implementing a much faster algorithm now. We have also developed a prototype PC cluster system for performing the proposed methods for very large proximity matrices that will be released after it has been fully tested.

A number of further extensions to the existing version of GAP are underway. Modules for matrix visualization of longitudinal data, canonical data, pure categorical data, data with dependent variable(s), data with dependent (clustered) structure, and data with cartographical information are some GAP projects in preparation.

With the capacity for displaying thousands of variables in a single picture, the flexibility for working with all types of data, and the ability for handling the various manifestations of extraordinary data patterns (missing values, covariate adjustment), we believe the GAP approach matrix visualization has the opportunity to become one of the major graphical tools for the new generation of exploratory data analysis (Tukey, 1977).

Acknowledgements

GAP is available to readers and is free to non-commercial applications. The installation instructions, the user's manual, and the detailed tutorials can be found at http://gap.stat.sinica.edu.tw/Software/GAP. The authors are

grateful to Chen-Hsin Chen, Donald Ylvisaker, Antony Unwin, and ShengLi Tzeng for many valuable suggestions. The authors also thank Pei-Ing Hwang for providing the microarray data demonstrated herein. The comments of the anonymous referees as well as the Associated Editor and Professor Erricos John Kontoghiorghes are gratefully acknowledged. This work was supported partially by the National Science Council of Taiwan, R. O. C. under the grants NSC-94-2118-M-001-025 and NSC-96-3112-B-001-017 in the National Research Program for Genomic Medicine.

References

Bertin, J., 1967. Semiologie Graphique, Paris: Editions gauthier-Villars. English translation by William J. Berg. as Semiology of Graphics: Diagrams, Networks, Maps. TheUniversity of Wisconsin Press, Madison, WI, 1983.

Bhattacharjee, A., Richards, W.G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E.J., Lander, E.S., Wong, W., Johnson, B.E., Golub, T.R., Sugarbaker, D.J., Meyerson, M., 2001. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc. Natl Acad. Sci., USA 98(24), 13790-13795.

Chavent, M., Lechevallier, Y., Briant, O., 2007. DIVCLUS-T: A monothetic divisive hierarchical clustering method. Computational Statistics & Data Analysis 52(2), 687-701.

Chen, C.H., 1996. The properties and applications of the convergence of correlation matrices. 1996 Proceedings of the Section on Statistical Graphics of the American Statistical Association, 49-54.

Chen, C.H., 1999. Extensions of generalized association plots. 1999 Proceedings of the Section on Statistical Graphics of the American Statistical Association, 111-116.

Chen, C.H., 2002. Generalized association plots: information visualization via iteratively generated correlation matrices. Statistica Sinica 12, 7-29.

Chen, C.H., Chen, J.A., 2000. Interactive diagnostic plots for multidimensional scaling with applications in psychosis disorder data analysis. Statistica Sinica 10, 665-691.

Chen, C.H., Hwu, H.G., Jang, W.J., Kao, C.H., Tien, Y.J., Tzeng, S., Wu, H.M., 2004. Matrix visualization and information mining. Proceedings in Computational Statistics 2004 (Compstat 2004), 85-100, Physica-Verlag, Heidelberg.

Dansereau, F., Alutto, J.A., Yammarino, F.J., 1984. Theory Testing in Organizational Behavior: The Varient Approach. Englewood Cliffs, NJ: Prentice-Hall.

Dunn, C., 1974. Well-separated clusters and optimal fuzzy partitions. Journal

on Cybernetics 4, 95-104.
Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl Acad. Sci., USA 95, 14863-14868.
Fielding, A.H., 2007. Cluster and Classification Techniques for the Biosciences, Cambridge University Press; 1 edition.
Friendly M., 2002. Corrgrams: exploratory displays for correlation matrices. The American Statistician 56(4), 316-324.
Hahsler, M., Hornik, K., Buchta, C., 2008. Getting things in order: an introduction to the R package seriation. Journals of Statistical Software 25(3), 1-34.
Handl, J., Knowles, J., Kell, D.B., 2005. Computational cluster validation in postgenomic data analysis. Bioinformatics 21(15), 3201-12.
Hartigan, J.A., Wong, M.A., 1979. A K-means clustering algorithm. Applied Statistics 28, 100-108.
Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification 2(1), 193-218.
Hwu, H.G., Chen, C.H., Hwang, T.J., Liu, C.M., Cheng, J.J., Lin, S.K., Liu, S.K., Chen, C.H., Chi, Y.Y., OuYoung, C.W., Lin, H.N., Chen, W.J., 2002. Symptom patterns and subgrouping of schizophrenic patients: significance of negative symptoms assessed on admission. Schizophrenia Research 56, 105-119.
Inselberg, A., 1985. The plane with parallel coordinates, The Visual Computer, 1: 69-91.
Keim, D., 2001. Visual exploration of large databases. Communications of the ACM, 44(8), 38-44.
Lee, Y.S., Chen, C.H., Chao, A., Chen, E.S., Wei, M.L., Chen, L.K., Yang, K., Lin, M.C., Wang, Y.H., Liu, J.W., Eng, H.L., Chiang, P.C., Wu, T.S., Tsao, K.C., Huang, C.G., Tien, Y.J., Wang, T.H., Wang, H.S., Lee, Y.S., 2005. Molecular signature of clinical severity in recovering patients with severe acute respiratory syndrome coronavirus (SARS-CoV). BMC Genomics 6:132.
Ling, R.L., 1973. A computer generated aid for cluster analysis. Communications of the ACM 16(6), 355-361.
Minnotte, M., West, W., 1998. The data image: a tool for exploring high dimensional data sets. Proceedings of the ASA Section on Statistical Graphics, Dallas, Texas, 25-33.
Murdoch, D.J., Chow, E.D., 996. A graphical display of large correlation matrices. The American Statistician 50, 178-180.
R Development Core Team, 2005. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org`.
Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336), 846-850.
Robinson, W.S., 1951. A method for chronologically ordering archaeological

deposits. American Antiquity 16, 293-301.
Rousseeuw, P., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20(1), 53-65.
Streng, R., 1991. Classification and seriation by iterative reordering of a data matrix. In Classification, Data Analysis, and Knowledge Organization: Models and Methods with Applications Edited by Bock and Ihm), 121-130. Springer-Verlag, NewYork.
Tenenbaum, J.B., de Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319-2323.
Tien, Y.J., Lee, Y.S, Wu, H.M., Chen, C.H., 2008. Methods for simultaneously identifying coherent local clusters with smooth global patterns in gene expression profiles. BMC Bioinformatics 9:155, 1-16.
Troyanskaya, O., Cantor, M., Sherlock, G., Brown P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B., 2001. Missing value estimation methods for DNA microarrays. Bioinformatics 17(6), 520-525.
Tukey, J.W., 1977. Exploratory Data Analysis. Addison-Wesley.
Wegman, E.J., 1990. Hyperdimensional data analysis using parallel coordinates. Journal of the American Statistical Association 85(411), 664-675.
Wu, H.M., Chen, C.H., 2007. Covariate adjusted matrix visualization. Technical Report, Institute of Statistical Science, Academia, Taiwan.
Wu, H.M., Tzeng, S., Chen, C.H., 2008. Matrix visualization. In Chun-houh Chen, Wolfgang Hardle, and Antony Unwin, editors, Handbook of Computational Statistics (Volume III): Data Visualization, Springer-Verlag, Heidelberg.