Soft Comput (2011) 15:111–129
DOI 10.1007/s00500-010-0542-x

ORIGINAL PAPER

Gene expression studies with DGL global optimization for the molecular classification of cancer

Dongguang Li

Published online: 28 January 2010

Abstract This paper combines a powerful algorithm, called Dongguang Li (DGL) global optimization, with the methods of cancer diagnosis through gene selection and microarray analysis. A generic approach to cancer classification based on gene expression monitoring by DNA microarrays is proposed and applied to two test cancer cases, colon and leukemia. The study attempts to analyze multiple sets of genes simultaneously, for an overall global solution to the gene's joint discriminative ability in assigning tumors to known classes. With the workable concepts and methodologies described here an accurate classification of the type and seriousness of cancer can be made. Using the orthogonal arrays for sampling and a search space reduction process, a computer program has been written that can operate on a personal laptop computer. Both the colon cancer and the leukemia microarray data can be classified 100% correctly without previous knowledge of their classes. The classification processes are automated after the gene expression data being inputted. Instead of examining a single gene at a time, the DGL method can find the global optimum solutions and construct a multi-subsets pyramidal hierarchy class predictor containing up to 23 gene subsets based on a given microarray gene expression data collection within a period of several hours. An automatically derived class predictor makes the reliable cancer classification and accurate tumor diagnosis in clinical practice possible.

D. Li (✉)
School of Computer and Security Science,
Edith Cowan University, 2 Bradford Street,
Mount Lawley, WA 6050, Australia
e-mail: d.li@ecu.edu.au

Keywords Microarray gene expression · Classification · Cancer · Bioinformatics · Global optimization · Orthogonal arrays · Data mining

1 Introduction

DNA chip technology enables the study of gene expression in a large scale (Hacia 1999; Enright et al. 1999; Churchill 2002; Barrett 2005). Large-scale gene expressions are used to determine drug targets, identify co-regulated genes and study the response to environmental conditions and the effect of a single gene or a group of genes on the entire genome (Debouck and Goodfellow 1999; Marcotte et al. 1999; Baggerly et al. 2001; Bergmann et al. 2003; Abruzzo et al. 2005). Recent advances in biotechnology allow researchers to measure expression levels for thousands of genes simultaneously, across different conditions and over a time period. Analysis of data produced by such experiments offers potential insight into gene function and regulatory mechanisms (Bowtell 1999; Cheung et al. 1999; Brown and Botstein 1999; Duggan et al. 1999; Lipshutz et al. 1999; Perou et al. 1999; Alizadeh et al. 2000; Abul et al. 2005; Allison et al. 2006).

Computation is required to extract meaningful information from the large amount of data generated by expression profiling (Bassett et al. 1999; Aittokallio et al. 2003; Zhang and Gant 2004). Most of the algorithms commonly applied to microarray data analysis have been correlation-based approaches named cluster analysis (Alon et al. 1999; Cho et al. 2004). An efficient two-way clustering algorithm was applied to a colon cancer dataset consisting of the expression patterns of different cell types. Gene expression in 40 tumor and 22 normal colon tissue samples was analyzed across 2,000 genes (Alon et al. 1999). Cluster analysis groups

genes, involved in microarray data, which are similar in patterns of expression. Those clustered genes are likely to be functionally linked and need to be look into closely. Although cluster analysis has been accepted widely in analyzing the patterns of gene expression, the methods developed may not be able to fully extract the information from the microarray data corrupted by high-dimensional noise. If the noise from the genes that are irrelevant is not sufficiently reduced, incorrect classification for samples or misleading information on selecting informative genes may result. For selecting informative genes for sample classification a neighborhood analysis method was developed to obtain a subset of genes that discriminate between the acute lymphoblastic leukemia (ALL) and the acute myeloid leukemia (AML) successfully (Golub et al. 1999). In the microarray dataset containing 7,129 genes, those genes whose expression levels differ significantly in ALL and AML were identified and then they were subsequently used to predict the class membership (either ALL or AML) of new leukemia cases. Both approaches described above (Alon et al. 1999; Golub et al. 1999) were focused on comparing samples in each single gene dimension and assumed that the relevant genes were similarly and uniformly expressed among samples of each type. A multivariate approach that compares samples in a multi-gene dimension using the genetic algorithms (GA) was proposed (Li et al. 2001a, b). Samples were classified based on the class membership of their *k*-nearest neighbors (KNN) in the gene space. The dimensionality (length) of the gene subset was arbitrarily set to 50. GA was used to select hundreds and thousands of subsets of 50 genes that could potentially discriminate between two classes of samples (tumor and normal tissues). The frequency with which genes were selected was statistically analyzed in the large number of 50D gene subsets. The most frequently selected 50 genes were used to predict 34 new samples. Although the performance of GA predictor with 50 genes was remarkable only 29 of 34 test samples were correctly predicted with high confidence (Li et al. 2001a, b). To improve the successful rate of classification more reliable and accurate algorithms are needed.

Many machine learning and the data mining technologies have recently been introduced in the field of analyzing the microarray data to process many subsets of genes simultaneously (Brown et al. 2000; Aizenberg et al. 2002; Ooi and Tan 2003; Anderle et al. 2003; Wren et al. 2004). It is obvious that there is not a feasible approach to evaluate all the possible subsets of genes in a given dataset consisting of several thousands of genes. Even with a moderate number of gene elements in a gene subset and a small number of choices for each gene element, the number of possible gene combinations for the gene subset increases rapidly. The true magnitude of the problem can be seen by considering a scanning approach, which measures the objective function value for every possible combination of genes. For example, let us consider scanning a ten-gene subset using the colon data with 2,000 genes (2,000 gene expression measurements per sample). The total number of possible combinations is approximately more than 10^{30} (2,000! divided by 1,990!), which would take years for even a super computer. Efficient algorithms are needed to sample from fewer subsets to find the best performing subsets (optimal or near optimal solutions). Obviously, it is one of the optimization or global optimization problems. In order to solve the hard problems, such as gene selection, classification, and clustering, suitable optimization algorithms must be used.

During the past five decades the field of the global optimization has been growing at a rapid pace and many new theoretical, algorithmic, and computational contributions have resulted (Horst and Pardalos 1995). Global optimization is concerned with the computation and characterization of global minima (or maxima) of non-linear functions. Global optimization problems are widespread in the mathematical modeling of real world systems for a very broad range of applications. The majority of problems can be described as some form of global optimization procedures. In the gene selection problem, one would need to find how to form gene subsets to obtain the optimum classification response—changing one gene element in a given subset may improve the classification performance of the subset at one testing sample, but worsen it at another.

An objective function is necessary to evaluate how close each gene subset gets to the target requirement. The gene selecting process involves finding the gene subset that corresponds to the minimum (or maximum) of the objective function. Plotting the objective function against the gene search space of each element gene in the gene subset, one axis per element gene would be needed, plus the orthogonal axis for the objective function. The objective function plot would appear as a multi-peak, multi-variable plot. Because there are an enormous number of inter-related possible gene combinations, the best gene subset cannot be found by any simple process. It is not obvious how to select the genes analytically to find the best solution. The methods currently used in gene selection, such as the clustering, neighborhoods analysis, and GA, almost all depend on a starting condition either selected by the user or generated internally by the program that is sometimes not obvious. Changing the initial conditions will give a different result, and one has no way of knowing how much improvement could be effected.

Currently available multi-variable optimization algorithms for selecting the gene subset may not give optimum solutions. Usually those algorithms obtain their final solutions either from optimizing a starting guess or by techniques, which may or may not involve a pseudo-random process that gives different answers every time, depending upon the initial conditions. A true global optimization

algorithm should always find the very best solution possible within the boundary conditions stipulated. The possibility of creating a true global optimization algorithm for a large number of inter-dependent variables has been proposed in this study. Although many optimization algorithms may be appropriate for the gene classification problem, Dongguang Li (DGL) global optimization was proposed and applied to the cancer classification in this study for its superb performance in theory and applications.

It is a challenge to discover the optimum gene subset solutions from a microarray gene expression system with a large number of interacting gene variables. It is also well known that orthogonal arrays (OAs) have a number of advantages when they are used in designs of experiments (Hedayat et al. 1999; Dey and Mukerjee 1999). With the help of an established objective function based on KNN, DGL global optimization combines an OA's sampling procedure with some search space reduction strategies for constructing a multi-subset class predictor with a pyramidal hierarchy to predict the types of tumor tissues correctly.

With the DGL global optimization algorithms proposed in this research, one knows that the solution gene subsets found are optimized within the criteria set—there is no need to try other starting conditions for the same gene subset structure at a given length, because there are no starting guesses. The algorithms inexorably must find the optimum solution that exists within the boundary conditions. This efficiency has powerful economic consequences. For example, previous solutions which need excessive numbers of genes can now be replaced with fewer genes to get the same classification performance and better confidence. One can improve classification performance as well as offer previously unavailable and undetectable gene subsets as class predictors.

Some of strategies of DGL global optimization were firstly successfully applied to the optical thin film design problem (Li and Nathan 1996). It was also a candidate for the real function test bed of the First International Contest on Evolutionary Optimization to solve ten hard mathematical multivariable optimization problems (Li and Smith 1996). It is of great interest to develop techniques for extracting useful information from the microarray datasets. In this paper I report, the application of the DGL global optimization approach for classifying and validating two well-known datasets (Alon et al. 1999; Golub et al. 1999) consisting of the expression patterns of different cell types.

In previous years, many clinicians have been unable to provide a clear cut classification of cancerous patients, based upon the biopsy. However, with the system proposed here, the surveying of the expression of thousands of genes, is made practical.

This paper outlines a very workable concept, which with more development will bring groundbreaking new potential for accurate diagnosis. Its biggest advantage lies in the fact that the global optimum is always found with little prior knowledge.

The paper is arranged as follows. After an introductory section, the principles and the methodologies of this study are then proposed first. Second, a more rigorous and detailed explanation of some of the main concepts on the DGL global optimization are described in the section headed methods. The experimental results on the structures of the established multi-subsets class predictors for both the colon and the leukemia data are presented in a tabular format. The performance of those predictors in classifying cancer tissue samples in the two publicly available datasets are given in details. Finally, the discussion and conclusion have some remarks on the current applications and the direction of future research.

2 Methods

2.1 Datasets

Two popular microarray gene expression datasets, colon and leukemia, were used in this study.

2.1.1 Colon data

The original gene expression data were downloaded from the website (http://dir.niehs.nih.gov/microarray/datamining/public_html/colon.html). The matrix I2000 contains the expression of the 2,000 genes with highest minimal intensity across the 62 tissues (Alon et al. 1999). The genes are placed in order of descending minimal intensity. Each entry in I2000 is a gene intensity derived from the $\sim$20 feature pairs that correspond to the gene on the chip. The data are otherwise unprocessed (for example it has not been normalized by the mean intensity of each experiment). The name file contains the EST number and description of each of the 2,000 genes, in an order that corresponds to the order in I2000. The identity of the 62 tissues is given in file tissues data. The numbers correspond to patients, a positive sign to a normal tissue, and a negative sign to a tumor tissue. The data contain the expression levels of 2,000 genes across the 62 samples, of which 40 are tumor tissues and 22 normal tissues. Other researchers indicated that there were five tissue samples (Normal34, Normal36, Tumor30, Tumor33 and Tumor36) identified as likely to have been contaminated (Li et al. 2001a, b). To avoid having uncertainties those five samples were removed from the colon dataset. Like the previous study (Li et al. 2001a, b) the remaining 57 samples were then divided into a training set (the first 40 samples) and a test set (17 samples). The numbers of tumor and normal tissue samples are 27 and 13 in the training set and 10 and 7 in the test set, respectively.

2.1.2 Leukemia data

The original data were downloaded from the web (http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi). The data contain the expression levels of 7,129 genes across the 72 samples, of which 47 are the ALL samples and 25 the AML samples. These datasets contain measurements corresponding to ALL and AML samples from bone marrow and peripheral blood, that is divided into a training set (38 samples) and a test set (34 samples).

2.2 Overall methodology

The proposed DGL global optimization method in this study includes the following major steps:

- Sampling within search spaces by using a suitable orthogonal array instead of conducting a random search.
- Construction of objective function for optimization algorithms.
- Search spaces reduction strategies.
- Searching for global optimal solutions.
- Building up a multi-subsets pyramidal hierarchy class predictor for classification.
- Predicting through a voting mechanism.

Figure 1 describes the general procedures of the computing system.

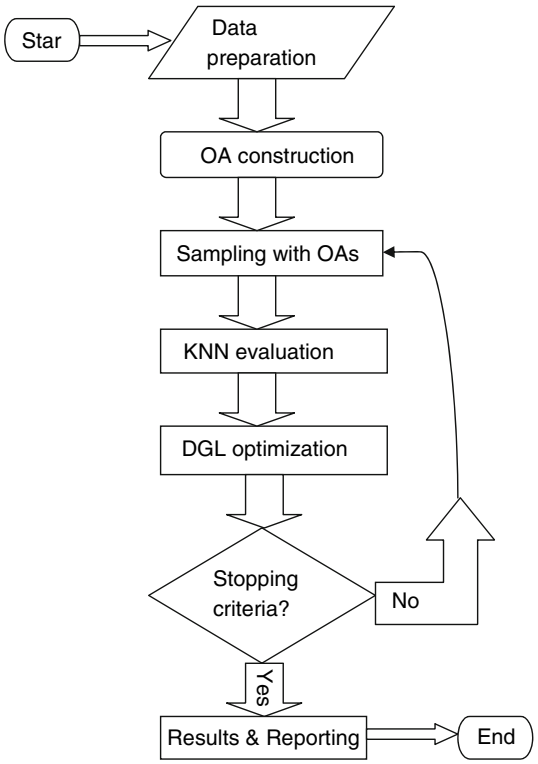

Fig. 1 The flowchart for the molecular classification of cancer based on microarray gene expression data

2.3 OAs and sampling procedure

Orthogonal arrays were discovered and introduced in the middle of last century (Rao 1946, 1947, 1949). Many statistical texts on experimental designs include OAs (Cochran and Cox 1957; Chi and Bloebaum 1996; Montgomery 1997). OAs are often employed in industrial experiments to study the effect of several control factors. An orthogonal array is a type of experiment where the columns for the independent variables are "orthogonal" to one another.

An orthogonal array A is a matrix of n rows and k columns, with every element being one of the q levels, which is normally represented in the form of $L_n(k^q)$. The rows of the array represent the experiments or tests to be performed. The columns of the orthogonal array correspond to the different variables whose effects are being analyzed. The entries in the array specify the levels at which the variables are to be applied. A typical OA $L_{12}(11^2)$ is shown below.

1	1	1	1	1	1	1	1	1	1	1
2	2	2	1	2	2	1	2	1	1	1
1	2	2	2	1	2	2	1	2	1	1
1	1	2	2	2	1	2	2	1	2	1
1	1	1	2	2	2	1	2	2	1	2
2	1	1	1	2	2	2	1	2	2	1
1	2	1	1	1	2	2	2	1	2	2
2	1	2	1	1	1	2	2	2	1	2
2	2	1	2	1	1	1	2	2	2	1
1	2	2	1	2	1	1	1	2	2	2
2	1	2	2	1	2	1	1	1	2	2
2	2	1	2	2	1	2	1	1	1	2

Pick any two columns; say the first and the last from the above table:

1	1
2	1
1	1
1	1
1	2
2	1
1	2
2	2
2	1
1	2
2	2

Each of the four possible rows one might see there, {(1, 1), (1, 2), (2, 1), (2, 2)}, does appear, and they all appear the same number of times (three times here), which is the property that makes it an orthogonal array. Since only 1's and 2's appear, this is called a two-level array.

There are 11 columns, which means one can vary the levels of up to 11 different variables, and 12 rows, which means 12 different combinations of variables can be tested in experiments. The aim is to investigate not only the effects of the individual variables on the outcome, but also how the variables interact.

Owen (1992, 1994) and Loh (1996) describe some uses for randomized OAs, in numerical integration, computer experiments and visualization of functions. Those references contain further references to the literature that provide further explanations.

The orthogonal array used in this research is $L_{242}(11^{23})$ that is too large to be shown here. The OA $L_{242}(11^{23})$ has 242 rows (observations or tests), 23 columns (factors or variables) and 11 levels for each factor. The complete $L_{242}(11^{23})$ is available on the website http://www.scis.ecu.edu.au/dli/.

The $L_{242}(11^{23})$ was initially used in selecting a gene subset with 23 gene elements. The search space of 2,000 genes in the colon data was divided into 11 levels equally. If all the genes are assigned a unique ID number from 1 to 2000 and the initial search space ranges from 1 to 2,000, then the selected gene IDs are 1, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, and 2000, respectively. As the first row of $L_{242}(11^{23})$ reads (1, 10, 2, 3, 8, 8, 2, 4, 8, 9, 5, 4, 10, 5, 7, 1, 5, 5, 8, 1, 10, 11, 2) the constructed gene subset will read (1, 1800, 200, 400, 1400, 1400, 200, 600, 1400, 1600, 800, 600, 1800, 800, 1200, 1, 800, 800, 1400, 1, 1800, 2000, 200). Since the duplicated gene IDs are not allowed in a gene subset, those repeated gene IDs are shift forward or backward a little bit. The modified 23-gene subset now reads (1, 1800, 200, 400, 1400, 1399, 199, 600, 1401, 1600, 800, 599, 1799, 799, 1199, 2, 801, 798, 1401, 3, 1798, 2000, 201). According to $L_{242}(11^{23})$, 242 different 23-gene subsets were created and evaluated with the defined objective function. All the 242 subsets were ranked based on their values of objective function. 10% top performers in classifying the training set were kept and those gene IDs included in the top 10% gene subsets were ranked to work out the minimum ID and the maximum ID. The new and reduced search space ranged from the minimum ID to the maximum ID. The above process was repeated until the search space was small enough (e.g. <11 genes left) or the objective function could not be improved any further. The rank No. 1 gene subset in the last round of optimization was chosen as the optimal solution for the 23-gene subsets. The optimization was run 23 times with different lengths (23, 22, …, 2, 1) of gene subsets at each run. A total of 23 optimal solutions were obtained. All the 23 optimal solutions constructed a multi-subset cancer class predictor and then were used to classify the samples in the test dataset. All the 23 gene subsets were arranged to form a pyramidal layer-by-layer hierarchy with the shortest subset (1 gene) on the top and the longest subset (23 genes) in the bottom (see Tables 2, 4 for details).

2.4 Objective function

An objective function is also called a fitness or merit function, which is a measure of the ability for a selected gene subset to classify the training set samples according to the DGL optimization procedure. There are several ways, such as neighborhood analysis (Golub et al. 1999), support vector machines (Peng et al. 2003; Liu et al. 2005), and KNN (Li et al. 2001a, b), to construct an objective function for the optimization and gene selection algorithms. Among them KNN is used for the proposed DGL global optimization because it is easy to compute. The Euclidean distance between a single sample (represented by its pattern vector $\boldsymbol{V}_m$) and each of the pattern vectors of the training set containing M samples is calculated.

$\boldsymbol{V}_m = (g_1, g_2, \ldots, g_n)$, where n is the number of genes in the vector that can be set to from 1 to 23 to form the gene vectors (or subsets) with different lengths; g_n is the expression level of the nth gene in the mth sample; $m = 1, 2, \ldots, M$. For the colon dataset $M = 40$ and $M = 38$ for the leukemia dataset.

The "fitness" of each gene vector is subsequently evaluated by its ability to correctly classify samples using KNN. For each set of n selected genes, the pair-wise Euclidean distances between the samples in the n-dimensional space is computed. The class membership of a sample is then declared by its KNN. If the actual class membership of the sample matches its KNN-declared class, a score of one is assigned to that sample; otherwise, a score of zero is assigned. Summing these scores across all samples provides a fitness measure for the gene vector. A perfect score would correspond to the number of samples in the training set.

Each sample is classified according to the class membership of its KNN as determined by the Euclidean distance in n-dimensional space. If all or majority of the KNN of a sample belongs to the same class, the sample is classified as that class. Otherwise, the sample is considered unclassifiable. The k was arbitrarily set to 5 in this study. The detailed rules are shown in Table 1.

If the class membership of a training set sample and its five nearest neighbors in the particular n-dimensional space defined by a gene subset agree or four out of five nearest neighbors agree, the sample is classified and a score of 1 is assigned to that sample. These agreement scores are summed across the training set. For the convenience, this sum is divided by the number of training samples (40 in colon and 38 in leukemia) as the value of the objective function for the selected gene subset. The bigger the value is, the better the selected gene subset

Table 1 KNN rules ($k = 5$)

	Among the ranked five nearest neighbors	Classified as	Class code
Colon	All five are normal samples	Normal	1
	All five are tumor samples	Tumor	−1
	Four are normal and one is tumor	Normal	1
	Four are tumor and one is normal	Tumor	−1
	Three are normal and two are tumor	Unknown	0
	Three are tumor and two are normal	Unknown	0
Leukemia	All five are ALL samples	ALL	1
	All five are AML samples	AML	−1
	Four are ALL and one is AML	ALL	1
	Four are AML and one is ALL	AML	−1
	Four are ALL and two are AML	Unknown	0
	Four are AML and two are ALL	Unknown	0

performs in classification. A maximal objective function value is 1, which means all the samples in the training set are classified correctly by the gene subset under testing. The goal of the optimization procedure is to discover the optimal gene subset (optimal solution) with the maximal value of the objective function.

As in other methods, an objective function is calculated for each subset of genes by the sum over all classifying scores of the samples in the training dataset. The optimization process then conducts the searching for the gene subset that has the best objective function value (minima or maxima). Therefore, by finding the lowest or highest value of the objective function one will have the best performing gene subset discovered. This procedure can be made more sophisticated by introducing weighting factors to increase the importance of user specified samples in training sets, as well as using other forms of the distance formula between one subset and another.

2.5 Search spaces reduction for global search

With local optimization (a fast method for a large number of genes), the program finds the nearest minimum and stops. For some so-called global optimization procedures, the algorithm not only finds a local minimum but can also find some neighboring minima. The process, however, is a hit and miss situation, because starting at a different place can result in different solutions.

The global algorithm in DGL repeatedly narrows the region where the global minimum is known to lie by using a special OA's sampling that operates simultaneously in all orthogonal dimensions, one for each gene in the gene subset to find the optimum solution. As the process runs, one can observe the range of genes for each gene variable in an n-dimensional subset being reduced.

The DGL global optimization algorithm operates to discover the optimum solution. An analogy illustrates the principles involved. Assume plotting the objective function against 2,000 genes in the colon data with a goal of finding a gene or a gene subset corresponding to the maximum objective function value F (or $1/F$ for minimum value for the convenience in the illustration). See Fig. 2 for one-dimensional analogy showing local and global optimization process.

As discussed earlier, a single objective function number can be used to describe the classification performance of a current gene subset. By plotting a multi-dimensional graph with objective function as one of the axes, one can visualize the process. One requires as many orthogonal axes as the number of variables (genes) plus one for the objective function. Thus, for a two-gene problem, a 3D plot is required.

To see the process used in a simplified form image, a 2D array along the x and y axes, which corresponds to a two-gene problem. Let the values along the x axis represent gene IDs of the first gene variable, and along the y axis the

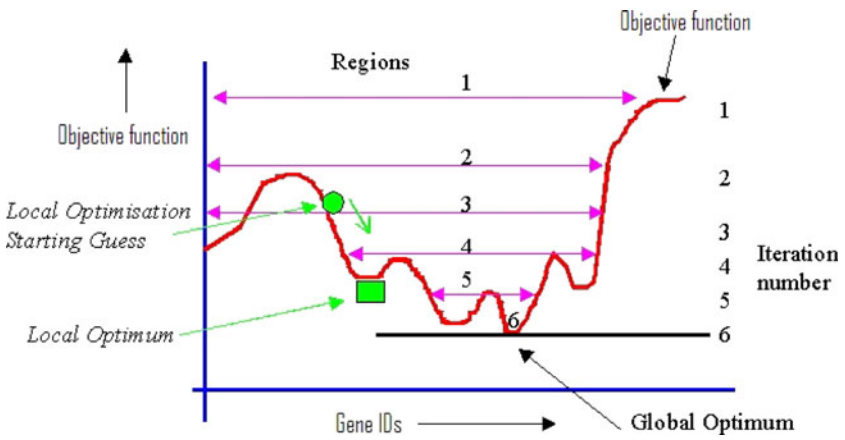

Fig. 2 Local optimization methods locate the minimum closest to the starting point. Global optimization techniques may find other local minima but cannot ensure the absolute lowest has been found. The region investigated by DGL sampling is progressively narrowed, as designated *1* to *6*, to ascertain a true global minimum

gene IDs of the second gene variable. The objective function value is plotted in the z direction.

The task of the process is then to find the x and y values that generate the highest value of the objective function. In this description, one shall invert the objective function back since a peak is easier to see than a valley. One form of objective function can be transformed into the other by taking a reciprocal of it. Therefore, one chooses the form of objective function whose value increases as the design performance improves, i.e., we want to maximize the objective function. The game is to change each gene variable to maximize the objective function. To help understand the optimization process, consider the analogy of a man wandering in a cratered terrain, with a global positioning satellite (GPS) receiver, which displays his absolute x, y, and z coordinates. Height (z) is the objective function, and x, y represent the gene IDs of each gene variable in a two-gene problem. His task is to find the highest point (largest objective function value) bounded by the user defined maximum and minimum values of x and y. If he just walks upwards until he can go no further uphill, he will have found the local maximum. The 'best guess approach' is based on a starting design (position) that may be based on many years of experience. There are several mathematical methods available such as the gradient method, which alter several gene variables simultaneously, see what happens to the objective function, and move the gene subset in the direction of a maximum of the objective function. These find the 'local optimum', and the end subset solution is completely dependent on the starting guess. A derivation of this is the GA method which, by the use of evolution in populations and random numbers, is able to find better maxima, exploring other subsets by 'jumping away' from the nearest peak.

The GA optimization methods use a completely different technique to optimize the gene subsets. It does have the advantage of being capable of producing complex gene subsets with minimum user interaction; however, the solution found is unlikely to be the global optimum. The final solution is still dependent on the initial starting gene selection, and many more genes more than necessary are often required for a given performance.

The DGL optimization operates by a process of searching for all regions in the gene space where a height greater than a specific level is located. This is a kin to creating a contour map by slicing parameter space at a constant value of objective function. The levels (choices) of the genes were equally spaced within every reduced search space.

In Fig. 3a–c, one sees a representation of a two-gene subset problem. Using a search analogy, to find the peak, the region in which the highest peak must lie is narrowed each time the plane is raised. This occurs until there is only one peak left. Its coordinates correspond to the gene IDs of the optimum solution. This is the equivalent of a plane parallel to the x–y plane at height z (Fig. 3b). This plane intersects the topography and identifies the entire region within which the peak is known to lie. By raising the slicing plane repeatedly, the region within which the peak must lie is made smaller and smaller until only the highest peak remains (see Fig. 3c). Its coordinates correspond to the gene IDs of the optimum gene subset. In practice, the surface is a mathematical construct of as many orthogonal dimensions as there are genes in the gene subset with a given length.

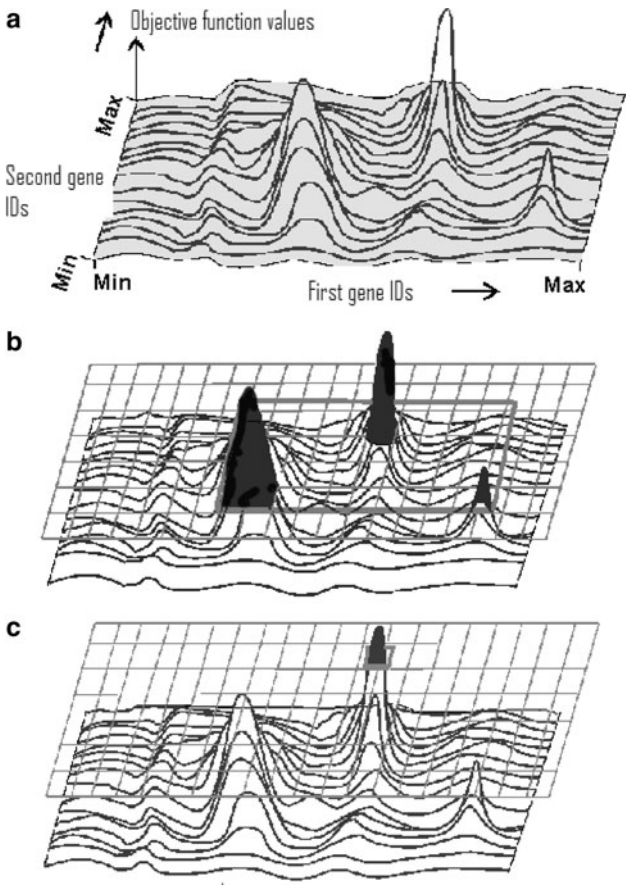

Fig. 3 **a** In this description of a 2-gene-subset problem, the boundary conditions are defined by the gene variable's maximum and minimum IDs. One seeks to find the gene IDs that give rise to the maximum value of the objective function. **b** A plane is constructed of 'constant objective function' and the boundary of regions having a higher objective function than that of the plane is identified. **c** As the plane is raised, the region within which the peak of the objective function exists is narrowed. The process is repeated until the two gene IDs (coordinates of the peak) which give rise to the highest peak are uniquely identified

No starting guess is necessary (or even possible), and the operator only has to define the basic parameters, such as the number of genes in a gene subset and the minimal and maximal gene IDs for each gene variable (i.e., the boundary conditions). After the process is started, the operator can

observe the values of maximal and minimal (within which the global optimum resides) for the various gene variables approaching each other. At the end of the run, there will be no gene variable whose search space (maximal–minimal) is greater than the specified value (as little as 1 gene). Using DGL global optimization strategies, an operator can be assured that he/she has found the best solution physically possible, independent of his/her so-called best guess.

2.6 Mathematical form of DGL optimization

Consider a multi-dimensional continuous function $f(\mathbf{x})$ with multiple global minima and local minima on subset G of R^n

1. A local minima is defined as follows.

For a given point $\mathbf{x}^* \in G$, if there exists a δ-neighborhood of $\mathbf{x}^*$, $O(\mathbf{x}^*, \delta)$, such that for $x \in O(\mathbf{x}^*, \delta)$, and

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \tag{1}$$

then $\mathbf{x}^*$ is called a local minimal point of $f(\mathbf{x})$.

2. Definition of global minima.

If for every $\mathbf{x} \in G$ the inequality (1) is correct, then $\mathbf{x}^*$ is called a global minimum of $f(\mathbf{x})$ on G, and the global minima of $f(\mathbf{x})$ on G form a global minimum set.

3. How to find the global minima.

Now for a given constant C_0 such that the level set $H_0 = \{\mathbf{x} | f(\mathbf{x}) < C_0,\ \mathbf{x} \in G\}$ is non-empty, if $\mu(H_0) = 0$, where μ is the Lebesque measure of H_0, then C_0 is the minimum of $f(\mathbf{x})$ and H_0 is the global minimum set.

Otherwise, assume that $\mu(H_0) > 0$ and C_1 is the mean value of $f(\mathbf{x})$ on H_0.

Then

$$C_1 = 1/\mu(H_0) \int_{H_0} f(\mathbf{x}) \mathrm{d}\mu \tag{2}$$

and

$$C_0 \geq C_1 \geq f(\mathbf{x}^*) \tag{3}$$

One then gradually constructs the level set H_k and mean value C_{k+1} of $f(\mathbf{x})$ on H_k as follows:

$$H_k = \{\mathbf{x} | f(\mathbf{x}) < C_k,\ \mathbf{x} \in \mathrm{G}\} \tag{4}$$

and

$$C_{k+1} = 1/\mu(H_k) \int_{H_k} f(\mathbf{x}) \mathrm{d}\mu \tag{5}$$

With the assistance of OA's sampling, a decreasing sequence of mean values $\{C_k\}$ and a sequence of level sets $\{H_k\}$ are obtained.

Let

$$\lim_{k \to \infty} C_k = C^* \tag{6}$$

and

$$\lim_{k \to \infty} H_k = H^* \tag{7}$$

It can be proven that C^* is the minimum of $f(\mathbf{x})$ on G, and H^* is the global minimum set.

There are several strategies to avoid missing the global optimum when seeking the minimum solution. Among these, the most important step is to select or design a suitable orthogonal array with which the function within domains can be repeatedly sampled. The algorithm is automatically constrained to stay within the function domain and will not request function evaluations outside this domain.

There are two stopping criteria possible; either when the target objective function value is reached, or when the maximum domain length is smaller than the user selected value. In this research, one uses the latter stop criteria, corresponding to the variation possible for each gene element in the subset, which can be as little as one gene. This means that the global minimum has been found for a particular gene selection range of each gene element, with a variation of less than one gene for each gene element. Strictly speaking then, the global optimum is not defined at a point but as lying within a region.

2.7 Multi-subsets class predictor

Although DGL optimization will result in an optimal gene subset with a given length, the classification performance varies. It seems that for both the colon and leukemia datasets there is no guarantee to name a single gene subset that is capable of classifying all the samples in the testing set correctly. It is observed that the gene subsets with different lengths tend to misclassify or unclassify the different samples in the testing datasets. In another words, the gene subsets with the same length will always misclassify a few same samples in the testing datasets, although those are all the optimal subsets identified by optimization procedures. This fact indicates that the key factor to improve the signal to noise ratio in classifying the very noisy data, such as the microarray gene expressions, is the length of the gene subset. Based on the above observation a multi-subsets class predictor was constructed for classification by using all the 23 optimal gene subsets with the lengths from 1 to 23 genes. The maximal number of genes involved in the predictor is 276 in total. As some of genes may appear more than one time, the actual number of the unique gene IDs is a bit less and varies from case to case.

Table 2 Optimal gene subsets and selected genes for the colon data class predictor

Gene subsets	Gene IDs																						
1-gene-subset	1227																						
2-gene-subset	619	1286																					
3-gene-subset	249	550	1102																				
4-gene-subset	802	164	693	765																			
5-gene-subset	1842	1558	1423	853	581																		
6-gene-subset	1774	567	249	1095	164	1539																	
7-gene-subset	1042	1286	996	1423	1550	1676	581																
8-gene-subset	164	1327	1487	1110	1362	1461	567	249															
9-gene-subset	251	1360	206	249	1020	1415	1437	2000	1812														
10-gene-subset	1670	773	1611	698	1915	855	1562	1087	245	924													
11-gene-subset	581	1735	698	276	118	1247	249	629	1551	504	802												
12-gene-subset	1043	2000	662	1351	782	567	164	642	66	245	760	977											
13-gene-subset	247	1680	1380	1706	1042	765	1282	1058	1880	633	1208	1196	1674										
14-gene-subset	572	1686	1905	1699	1294	1372	897	323	1009	493	1809	295	1216	513									
15-gene-subset	1033	1402	1085	1201	679	245	1272	1312	897	66	1020	1436	1066	1623	164								
16-gene-subset	853	542	1058	2000	1600	1263	355	1393	1714	898	249	164	911	316	1589	902							
17-gene-subset	1424	1644	757	249	1626	1067	804	778	1231	1670	1656	1412	2000	1819	1445	1173	888						
18-gene-subset	1802	1171	1138	1663	1657	249	1086	327	594	138	1596	1800	1582	309	1264	1538	228	871					
19-gene-subset	741	765	1662	1863	623	394	1217	417	605	2000	1818	1071	567	885	164	665	1943	1366	1908				
20-gene-subset	254	980	723	1043	547	1678	1819	1041	1519	813	245	1600	1840	479	288	1873	510	883	1805	1166			
21-gene-subset	249	948	904	401	1846	583	1165	264	1539	457	2000	1025	1600	228	1945	1213	1610	344	1360	1430	1850		
22-gene-subset	1497	1786	1800	831	1849	239	994	513	1370	857	240	362	1608	1890	228	1761	508	341	586	249	539	340	
23-gene-subset	712	1209	422	66	471	1792	1970	164	599	990	1362	1128	1472	1469	1557	1672	245	1665	1103	1070	541	767	1490

2.8 Validation (predicting through a voting mechanism)

The established multi-subsets class predictor is validated with the testing datasets for both the colon and the leukemia data. Each gene subset in the predictor predicts the class of every sample in the testing datasets independently according to the same KNN rules ($k = 5$) used in the training stage. The predicted class code (in colon data: 1 for normal; -1 for tumor; 0 for unknown and in leukemia data: 1 for ALL; -1 for AML and 0 for unknown) is assigned to the particular sample accordingly. Each single class code is treated as a single vote. For each sample in the testing datasets up to 23 votes contributed by 23 gene subsets in the predictor can be obtained. The final class predicted by the predictor depends on the sign of the sum of the 23 votes of the sample under test. A positive sign indicates there are more gene subsets in the predictor vote for class 1 (normal for colon and ALL for leukemia) and the sample is finally classified as 1 by the multi-subsets predictor. A negative sign indicates there are more gene subsets in the predictor vote for class -1 (tumor for colon and AML for leukemia) and the sample is finally classified as -1 by the multi-subsets predictor. When the sum is zero, there are equal numbers of gene subsets among the 23 gene subsets for the class 1 and the class -1. In this case, the corresponding sample should be classified as 0 (unknown or unclassified). It is not difficult to interpret the actual values of the classification results. The absolute value of the sum of the 23 votes should indicate the predicting strength. The larger the value is, the more confident the prediction is.

3 Experimental results

A Microsoft Windows-based computer program with a user-friendly graphic interface has been written. The entire experimental computation was carried out on a personal laptop computer (1.7 GHz Intel Pentium Pro/II/III). The software can be downloaded from the supporting website of this paper (Li 2006) and is available free to researchers. Both the colon cancer and the leukemia samples were classified 100% correctly. The classification processes are automated after the gene expression data being inputted. It can find the global optimum solutions and construct a multi-subsets class predictor containing up to 23 gene subsets based on a given microarray gene expression data collection, such as the colon or leukemia data, within a period of several hours.

For the convenience of computation every gene was assigned a unique integer ID number (from 1 to 2000 for colon and from 1 to 7129 for leukemia) according to the order in their original datasets. The aim was to study how changes in the choices of various gene element variables for a gene subset with a given length affect a response variable (success rate in classifying training samples). For each of the gene elements that are used to form a gene subset eleven choices (levels) were selected for inclusion in the OA's sampling based on $L_{242}(11^{23})$. Those eleven choices of gene IDs were generated by the formula (the length of the current search space divided by 10) at an equal distance. Some shifting on the selected gene IDs was necessary to avoid having any repeating genes in a single gene subset. 242 subsets were evaluated with the objective function in the current iteration and 10% top performing subsets were used to reduce the search space. Only two top performing gene subsets were passed to the next iteration.

Within the search space of 2,000 genes for the colon data and 7,129 genes for the leukemia data, DGL global optimization found 23 optimal gene subsets with different lengths from 1 gene to 23 genes, which formed two pyramidal hierarchy class predictors, respectively (Tables 2, 4). Those gene subsets were assumed to be the best performing gene combinations for classifying the gene datasets used in this study. The selected gene subsets were then used in classifying the test samples in both the colon and leukemia datasets. Tables 3 and 5 are the classification results. Once the validation of all the 23 optimal gene subsets was completed the proposed multi-subsets voting mechanism was adopted. One of the classification results (1 for class 1; -1 for class 2; 0 for unclassified) was obtained by balancing the votes from the 23 gene subsets for the particular testing sample of interest. It is a process of counting votes to make a final decision on the class of the sample under test. For example, the sample N28 in the colon dataset (shown in Table 3) receives 23 votes in total from the 23 gene subsets in the class predictor. Among the 23 votes there are 11 votes of normal, 6 of tumor, and 6 of unknown. The class code (1 for normal; -1 for tumor; 0 for unknown) is assigned to the votes and then the sum of the votes $\{11 \times (+1) + 6 \times (-1) + 6 \times 0 = +5\}$ is calculated. Since the sum is $+5$, which mean there are more votes favoring the normal class, the sample N28 is classified as "normal" and is given the class code of $+1$. Otherwise the sample should be classified as "tumor" with the code of -1 if the sum of the votes is a negative value. The absolute value of the sum (ranges from 1 to 23) indicates the predicting strength. When the sum is just equal to 0, the sample under test is unclassifiable.

4 Discussion

The optimization algorithms are playing a significant role in the field of gene selection and sample classification for

Table 3 Validation of the colon data with the multi-subsets class predictor

Gene subsets	Prediction of 17 test samples																	Results of classification			
	T28	N28	N29	T29	T31	T32	N32	N33	T34	T35	N35	T37	T38	T39	N39	T40	N40	Correct	Incorrect	Unknown	Success rate %
1-gene-subset	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	1	−1	−1	11	6	0	64.7
2-gene-subset	−1	0	−1	−1	−1	−1	0	−1	−1	−1	0	−1	−1	−1	−1	−1	0	10	3	4	58.8
3-gene-subset	−1	1	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	17	0	0	100
4-gene-subset	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	10	7	0	58.8
5-gene-subset	−1	−1	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	0	−1	1	15	1	1	88.2
6-gene-subset	−1	0	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	16	0	1	94.1
7-gene-subset	−1	0	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	0	−1	1	15	0	2	88.2
8-gene-subset	−1	0	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	16	0	1	94.1
9-gene-subset	−1	−1	1	−1	−1	−1	−1	1	−1	−1	1	−1	−1	−1	1	−1	1	15	2	0	88.2
10-gene-subset	−1	1	0	−1	−1	−1	−1	1	−1	−1	1	−1	0	−1	−1	−1	1	13	2	2	76.5
11-gene-subset	−1	1	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	17	0	0	100
12-gene-subset	−1	1	1	−1	−1	−1	1	0	−1	−1	1	−1	−1	−1	−1	−1	1	15	1	1	88.2
13-gene-subset	−1	−1	1	−1	−1	−1	1	1	−1	−1	0	−1	−1	−1	−1	−1	−1	13	3	1	76.5
14-gene-subset	−1	0	1	−1	−1	−1	0	1	−1	−1	1	1	−1	−1	1	−1	1	14	1	2	82.3
15-gene-subset	−1	1	1	−1	−1	−1	1	0	−1	−1	1	−1	−1	−1	0	−1	1	15	0	2	88.2
16-gene-subset	−1	0	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	16	0	1	94.1
17-gene-subset	−1	1	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	17	0	0	100
18-gene-subset	0	1	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	16	0	1	94.1
19-gene-subset	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	10	6	1	58.8
20-gene-subset	−1	1	0	−1	−1	−1	−1	1	−1	−1	1	−1	−1	−1	−1	−1	0	13	2	2	76.5
21-gene-subset	0	1	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	0	−1	1	15	0	2	88.2
22-gene-subset	−1	1	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	16	1	0	94.1
23-gene-subset	−1	1	0	−1	−1	−1	1	0	−1	−1	1	−1	−1	−1	−1	−1	1	14	1	2	82.3
Sum of votes	−21	5	13	−23	−23	−23	9	12	−23	−23	15	−21	−22	−23	3	−23	13				
Classified as	−1	1	1	−1	−1	−1	1	1	−1	−1	1	−1	−1	−1	1	−1	1	17	0	0	100

Each every gene subset is used to classify the 17 test samples. The predicted class for every sample is represented by a vote value (1 for normal, −1 for tumor, and 0 for unknown)

For the sum of votes one adds up the vote values across all 23 gene subsets for every sample. If the sum of the 23 prediction votes is positive value (which indicates there are more gene subsets favoring the normal class than the tumor class) the corresponding sample is classified as a normal sample with the code of 1. If the sum of the 23 prediction votes is negative value (which indicates there are more gene subsets favoring the tumor class than the normal class) the corresponding sample is classified as a tumor sample with the code of −1. In case of the sum of the 23 prediction votes is equal to zero the corresponding sample is unclassified with the code of 0

Table 4 Optimal gene subsets and selected genes for the leukemia data class predictor

Gene subsets	Gene IDs																						
1-gene-subset	5501																						
2-gene-subset	3320	1068																					
3-gene-subset	2020	4782	2348																				
4-gene-subset	4270	2039	4050	2642																			
5-gene-subset	2642	1837	4050	1488	5605																		
6-gene-subset	3137	2642	3336	2368	2852	4050																	
7-gene-subset	2020	1725	2531	2096	4991	2348	2120																
8-gene-subset	2642	4492	307	6368	3753	4708	5655	4050															
9-gene-subset	1481	4991	2224	2642	109	4050	5094	3565	6441														
10-gene-subset	2619	3119	3056	2971	4339	5297	2861	2020	5247	2001													
11-gene-subset	1584	4023	2020	1506	2852	4459	1060	6467	2295	2348	2483												
12-gene-subset	6910	4669	2642	6939	1891	4050	2020	4916	6487	1442	4950	2128											
13-gene-subset	1934	3906	3010	3392	5906	7129	4453	4724	4961	2280	2642	1	4050										
14-gene-subset	1362	2642	6771	4050	2090	6681	2811	988	4574	4727	5673	3191	1427	3565									
15-gene-subset	2020	1853	501	1387	4414	3565	3056	1630	6243	1143	2342	5251	4139	4720	1834								
16-gene-subset	4751	6438	3414	4224	5949	4889	1056	6559	2642	2648	5210	5166	1	4050	3888	5134							
17-gene-subset	7129	1834	4640	3189	6872	3118	2433	4050	1740	5326	4768	2469	6042	1	4444	2642	4252						
18-gene-subset	5828	442	3299	6548	2400	2378	3525	5452	4127	2642	5770	5342	6319	1945	4050	2780	6136	4464					
19-gene-subset	714	1714	5912	4711	3839	3215	2506	2642	3804	1900	5299	5609	4050	1	6655	6372	2791	1211	3068				
20-gene-subset	3515	7129	3854	6762	5826	4050	1250	2416	1021	3322	5451	5508	4410	2642	2327	4037	6639	4278	4334	5745			
21-gene-subset	92	6833	2642	1385	1801	3102	4251	4050	6832	5651	2449	4189	1925	5826	301	1126	3034	6940	1594	3342	5384		
22-gene-subset	5064	5692	6034	4050	6435	2642	628	501	4960	4908	5882	2227	3565	3998	2004	4723	7021	1	2829	1513	3423	3642	
23-gene-subset	7129	5477	714	4534	4572	643	3066	4991	2327	1229	4050	1425	1	4634	3565	6416	4452	3149	1250	4063	3026	2642	3780

Table 5 Validation of the leukemia data with the multi-subsets class predictor

Gene subsets	Prediction of 34 test samples (No.39–No.72)																																				Results of classification			
	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	Correct	Incorrect	Unknown	Success rate %
1-gene-subset	+1	0	+1	0	+1	+1	+1	+1	+1	+1	+1	0	0	−1	0	0	0	+1	0	−1	+1	0	0	0	0	+1	−1	0	+1	+1	+1	0	0	+1	18	2	14	52.9
2-gene-subset	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	+1	−1	0	+1	+1	0	−1	+1	+1	+1	+1	−1	−1	−1	+1	+1	+1	+1	+1	+1	+1	27	5	2	79.4
3-gene-subset	+1	+1	+1	−1	+1	+1	+1	+1	+1	+1	+1	−1	−1	+1	−1	−1	+1	+1	0	−1	+1	+1	0	0	0	−1	+1	+1	+1	+1	+1	+1	+1	+1	25	5	4	73.5
4-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2
5-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2
6-gene-subset	+1	−1	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	0	+1	+1	+1	+1	+1	30	1	3	88.2
7-gene-subset	+1	+1	+1	0	+1	+1	+1	+1	+1	+1	+1	−1	−1	+1	−1	−1	+1	+1	−1	−1	+1	+1	0	0	−1	−1	+1	+1	+1	+1	+1	+1	+1	+1	27	4	3	79.4
8-gene-subset	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	+1	−1	−1	+1	+1	0	−1	+1	+1	−1	−1	−1	−1	−1	+1	+1	0	+1	+1	+1	+1	29	3	2	85.3
9-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2
10-gene-subset	+1	+1	+1	0	+1	+1	+1	+1	+1	+1	+1	−1	−1	+1	−1	+1	+1	+1	+1	−1	+1	+1	+1	0	+1	−1	+1	+1	+1	+1	+1	+1	+1	+1	24	8	2	70.6
11-gene-subset	+1	+1	+1	−1	+1	+1	+1	+1	+1	+1	+1	−1	−1	+1	−1	−1	+1	+1	0	−1	+1	+1	0	+1	−1	−1	+1	+1	+1	+1	+1	+1	+1	+1	26	6	2	76.4
12-gene-subset	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	32	0	2	94.1
13-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2
14-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2
15-gene-subset	+1	+1	+1	0	+1	+1	+1	+1	+1	+1	+1	−1	−1	0	−1	+1	+1	+1	0	−1	+1	+1	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	+1	+1	27	4	3	79.4
16-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2
17-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	−1	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	32	0	2	94.1
18-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	−1	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	32	0	2	94.1
19-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2
20-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2
21-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	31	0	3	91.2

Table 5 continued

Gene subsets	Prediction of 34 test samples (No.39–No.72)																										Results of classification			
	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	Correct	In-correct	Un-known	Success rate %
22-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	−1	−1	−1	−1	−1	0	+1	+1	+1	+1	+1	31	0	3	91.2
23-gene-subset	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	0	+1	−1	−1	+1	−1	−1	0	−1	−1	−1	−1	0	+1	+1	+1	+1	+1	30	0	4	88.2
Sum of votes:	+23	+7	+23	+15	+23	+23	+23	+23	+23	+23	+23	−22	−22	−10	−22	−17	+7	+23	−15	−23	+22	-8	−15	−4	−19	−21	−13	−8	+20	+22	+23	+22	+22	+23				
Classified as	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1	+1	+1	−1	−1	+1	−1	−1	−1	−1	−1	−1	−1	+1	+1	+1	+1	+1	+1	34	0	0	100

Each every gene subset is used to classify the 34 test samples. The predicted class for every sample is represented by a vote value (+1 for ALL, −1 for AML, and 0 for unknown)

For the sum of votes one adds up the vote values across all 23 gene subsets for every sample. If the sum of the 23 prediction votes is positive value (which indicates there are more gene subsets favoring the normal class than the tumor class) the corresponding sample is classified as a normal sample with the code of 1. If the sum of the 23 prediction votes is negative value (which indicates there are more gene subsets favoring the tumor class than the normal class) the corresponding sample is classified as a tumor sample with the code of −1. In case of the sum of the 23 prediction votes is equal to zero the corresponding sample is unclassified with the code of 0

microarray data. Many advanced local and global optimization techniques, such as clustering and GAs, have been successfully applied to gene subset selection for classifying cancer tissue samples. Any optimization algorithm applied to a particular selection problem should first address the issue of choosing a reasonable starting solution, which is always a big obstacle to an inexperienced operator. To find the true global optimized solution for a gene selection problem, one need to solve an array of interlinked multi-dimensional simultaneous equations. For a gene subset with more than just a few gene elements, until recently this has been a very difficult task, requiring the use of a supercomputer and highly skilled programming. With the help of the DGL global optimization there is no need to solve any equations. The global optimized solutions can be found with affordable computing cost through the orthogonal sampling.

It is worth observing that the established multi-subsets class predictor could be reduced in size through removing the first five or more unstable short gene subsets. The remaining subsets would still perform well, that is shown on the supporting website (Li 2006). In general, the predicting strength may be improved. However, having those genes selected in the short subsets included may be significant to biologists as they could well be informative.

Another interesting observation is that there are not many genes that play a more important role than any other gene. The most frequently appeared genes involved in the colon predictor were 249 and 164, which appeared 10 times and 8 times, respectively. For most genes in the predictor, they were selected only once. For the leukemia predictor, the situation is quite similar. The genes 2642 and 4050 were the most frequently used genes being included 16 times. Both gene IDs assigned by this study and real gene accession numbers from the original datasets are listed in Tables 6 and 7, respectively. The gene appearance frequency for the colon class predictor is also given in the Table 6.

Some previous research works proposed to find out many near optimal gene subsets through a well tuned GA procedure and pick up top 50–200 most frequently appeared genes to construct a long gene subset as a predictor (Li et al. 2001a, b). Although the performance of such a predictor was reasonable good the large amount of computation might not be affordable or cost effective and might not be necessary. One more experiment was quickly carried out by forming a subset with seven most frequently appearing genes identified from the colon predictor. They are gene 249, 164, 2000, 245, 567, 66, and 581. Using such a 7-gene-subset to classify the samples in the test dataset the results were very encouraging. All of the 17 samples were correctly classified (data not shown here), which indicate that selecting the most frequently appeared genes

Table 6 The 221 genes selected by the colon class predictor and their appearance frequency

IDs	Genes	Frequency	IDs	Genes	Frequency	IDs	Genes	Frequency	IDs	Genes	Frequency	IDs	Genes	Frequency	IDs	Genes	Frequency	IDs	Genes	Frequency
66	T71025	3	479	U20428	1	757	T79595	1	1025	T74896	1	1231	H49870	1	1497	R00254	1	1706	H71627	1
118	T72889	1	493	R87126	1	760	M36341	1	1033	M69066	1	1247	X74295	1	1519	X69115	1	1714	X79888	1
138	M26697	1	504	X55362	1	765	M76378	3	1041	R54467	1	1263	T40454	1	1538	R67987	1	1735	H62885	1
164	X57351	8	508	H01677	1	767	U07695	1	1042	R36977	2	1264	H47107	1	1539	H78346	2	1761	T94350	1
206	L16510	1	510	D90188	1	773	R40184	1	1043	M86737	2	1272	M92843	1	1550	X53799	1	1786	M22050	1
228	J03040	2	513	M22382	2	778	X69550	1	1058	M80815	2	1282	L25081	1	1551	U20141	1	1792	M85169	1
239	H48027	1	539	H65182	1	782	X02750	1	1066	R83349	1	1286	D16294	2	1557	M21186	1	1800	U18062	2
240	T40507	1	541	X78706	1	802	X70326	2	1067	T70062	1	1294	U21049	1	1558	R49416	1	1802	T40674	1
245	M76378	5	542	M13686	1	804	R76254	1	1070	R70535	1	1312	M86934	1	1562	R49459	1	1805	H82741	1
247	T79813	1	547	M37984	1	813	M83738	1	1071	H40108	1	1327	T84082	1	1582	X63629	1	1809	H24310	1
249	M63391	10	550	M94630	1	831	T53868	1	1085	D42053	1	1351	X68688	1	1589	R71651	1	1812	X93510	1
251	U37012	1	567	X65488	4	853	H15813	2	1086	R39531	1	1360	H09719	2	1596	X51435	1	1818	H40699	1
254	H80975	1	572	T58756	1	855	L37033	1	1087	T55117	1	1362	M95549	2	1600	M90516	3	1819	T71049	2
264	M26252	1	581	T51571	3	857	R43976	1	1095	H04282	1	1366	Z11502	1	1608	L12723	1	1840	D38537	1
276	T53694	1	583	H59599	1	871	J03069	1	1102	T51558	1	1370	H22579	1	1610	U15782	1	1842	M87434	1
288	T48102	2	586	H17897	1	883	H04235	1	1103	T56460	1	1372	X75208	1	1611	H92195	1	1846	T52806	1
295	X15183	1	594	T89422	1	885	H27202	1	1110	L08069	1	1380	M92287	1	1623	T94993	1	1849	V00520	1
309	T70331	1	599	T53412	1	888	H89688	1	1128	D11466	1	1393	H77536	1	1626	T95612	1	1850	M22760	1
316	U26401	1	605	J03075	1	897	H43887	2	1138	T56475	1	1402	M38690	1	1644	R80427	1	1863	X89985	1
323	U28963	1	619	H89087	1	898	J02906	1	1165	H63354	1	1412	X61118	1	1656	X16504	1	1873	L07648	1
327	U37408	1	623	H11324	1	902	R00544	1	1166	R44740	1	1415	X12548	1	1657	R99935	1	1880	R09502	1
340	T87527	1	629	T60318	1	904	M64929	1	1171	U23852	1	1423	J02854	2	1662	H08751	1	1890	M19156	1
341	D13315	1	633	H87344	1	911	K02566	1	1173	M55543	1	1424	D31887	1	1663	X16663	1	1905	R23907	1
344	H15542	1	642	X16316	1	924	U09848	1	1174	R60906	1	1430	R09468	1	1665	X59842	1	1908	L25851	1
355	M30474	1	662	X68277	1	948	H45299	1	1196	D13315	1	1436	D10523	1	1670	H23975	2	1915	H87193	1
362	M36205	1	665	R38513	1	977	L40904	1	1201	X75304	1	1437	T67433	1	1672	D42047	1	1943	D29808	1
394	M99564	1	679	X70944	1	980	U06698	1	1208	H72965	1	1445	M13560	1	1674	T67077	1	1945	M69181	1
401	L19956	1	693	L11370	1	990	D38549	1	1209	R96357	1	1461	L34840	1	1676	U03851	1	1970	R06749	1
417	R61332	1	698	T51261	2	994	T51858	1	1213	M93651	1	1469	R38736	1	1678	R96070	1	2000	T49647	6
422	R77824	1	712	T90036	1	996	X79683	1	1216	R61324	1	1472	L41559	1	1680	M31516	1			
457	H22939	1	723	H65019	1	1009	X86018	1	1217	T62067	1	1487	L25941	1	1686	L47162	1			
471	J04173	1	741	T77446	1	1020	X56253	2	1227	T96873	1	1490	L00354	1	1699	R39540	1			

Table 7 The 219 genes selected by the leukemia class predictor

IDs	Gene accession number	IDs	Gene accession number	IDs	Gene accession number	IDs	Gene accession number	IDs	Gene accession number
5501	Z15115	1506	L36051	2342	M90696	6136	U28749_s	1925	M31165
3320	U50136_rna1	4459	X67683	5251	D28791	4464	X68149	301	D25303
1068	J03040	1060	J02883	4139	X13956	714	D87443	1126	J04809_rna1
2020	M55150	6467	U29463_s	4720	X85134_rna1	1714	M14123_xpt2	3034	U31449
4782	X90908	2295	M85169	1834	M23197	5912	HG880-HT880	6940	Z30644
2348	M91432	2483	S73813	4751	X87342	4711	X84195	1594	L41147
4270	X54936	6910	U84388	6438	S77154_s	3839	U82320	3342	U51166
2039	M57471	4669	X81889	3414	U56814	3215	U43522	5384	U13022
4050	X03934	6939	Z30643	4224	X52001	2506	S77576	5064	Z15108
2642	U05259_rna1	1891	M28713	5949	M29610	3804	U80017_rna2	5692	D89377_s
1837	M23379	4916	X99657	4889	X98263	1900	M29273	6034	U50360_s
1488	L34357	6487	X75346_s	1056	J02843	5299	L07919	6435	U05012_s
5605	D29675	1442	L27479	6559	U41315_rna1_s	5609	X14085_s	628	D83784
3137	U38846	4950	Y07596	2648	U05875	6655	Z11518_s	4960	Y07846
3336	U50939	2128	M63379	5210	Z79581	6372	M81182_s	4908	X99268
2368	M93284	1934	M31642	5166	Z48804	2791	U14550	5882	HG417-HT417_s
2852	U18004	3906	U89278	3888	U86782	1211	L05512	2227	M76558
1725	M14636	3010	U30245	5134	Z35491	3068	U33818	3998	U96629_rna2
2531	S81221	3392	U53476	1834	M23197	3515	U62437	2004	M37763
2096	M61156	5906	X07618_s	4640	X80062	3854	U83303_cds2	4723	X85372
4991	Y09615	7129	Z78285_f	3189	U41813	6762	M21388	7021	M33318_r
2120	M62994	4453	X67155	6872	M92642	5826	HG3125-HT3301_s	2829	U16296
4492	X69908_rna1	4724	X85373	3118	U37283	1250	L08424	1513	L36645
307	D26067	4961	Y07847	2433	S34389	2416	M97639	3423	U57099
6368	M80397_s	2280	M83651	1740	M15841	1021	HG511-HT511	3642	U70732_rna1
3753	U79249	1	AFFX-BioB-5	5326	M13577	3322	U50315	5477	X71661
4708	X84002	1362	L19067	4768	X89750	5451	X14766	4534	X74104
5655	U58046_s	6771	X87344_cds10_r	2469	S70348	5508	HG2157-HT2227	4572	X76105
1481	L33881	2090	M60749	6042	L10333_s	4410	X64643	643	D85376
2224	M76424	6681	X74874_rna1_s	4444	X66534	2327	M88282	3066	U33447
109	AC002115_cds4	2811	U15177	4252	X53742	4037	X02751	1229	L07077
5094	Z24727	988	HG4245-HT4515	5828	HG3187-HT3366_s	6639	U83598	1425	L25270
3565	U66048	4574	X76180	442	D45370	4278	X55666	4634	X79865
6441	S78873_s	4727	X85750	3299	U49187	4334	X59711	6416	S57153_s
2619	U03644	5673	D85425_s	6548	Z69030_s	5745	HG2261-HT2351_s	4452	X67098
3119	U37352	3191	U41816	2400	M95925	92	AB003698	3149	U39412
3056	U32944	1427	L25444	2378	M94167	6833	J00220_cds5	4063	X04434
2971	U27185	3565	U66048	3525	U63289	1385	L20348	3026	U31120_rna1
4339	X59812	1853	M25077	5452	X15422	1801	M21154	3780	U79287
5297	L07615	501	D50931	4127	X12901	3102	U36501		
2861	U18288	1387	L20773	5770	X52009_s	4251	X53587		
5247	D17532	4414	X64838	5342	M37712	6832	J00210_rna1		
2001	M37435	1630	L47738	6319	M60450_s	5651	D50477_s		
1584	L40410	6243	M24486_s	1945	M32315	2449	S76992		
4023	X01059	1143	J05213	2780	U13737	4189	X16667		

to form a subset for classification may gain significant advantages although those genes may show more biological significance. The more important point is the length of the gene subset can be shorted greatly from around 100 genes to around 10 by DGL.

In order to compare with the outputs from the GA algorithms closely (Li et al. 2001a, b) the validation experiments were carried out further. There are two genes appearing most frequently in the colon class predictor. One is gene 249 (10 times) and another is 164 (8 times). A subset with only these two genes is able to classify 16 out of 17 samples in the colon test dataset while 1 sample remains as unclassifiable. Although the gene 164 (X57351) was included in the 50 genes identified by a GA class predictor, the most frequently selected gene 249 (M63391) in this research was not captured before. The most frequently selected gene by GA was the human monocyte-derived neutrophil-activating protein (MONAP). Previous studies have demonstrated that the expression level of MONAP gene, whose gene ID is 1671 in this research, directly correlates with the progression of several human cancers (Shi et al. 1999). Unfortunately the gene 1671 was missed completely by the DGL method, which might reflect the fact that there were fundamental differences between GA and DGL in the way of sampling the search spaces to solve the problems.

For the leukemia data, 219 (shown in Table 7) out of 7,129 genes in the dataset were selected by DGL for constructing the class predictor.

Table 8 lists the genes appearing more than once in the leukemia class predictor based on frequency rank. It is worthwhile to note that the gene 2642 (U05259_ma1) and the gene 4050 (X03934) both appear 16 times (their frequency is much higher than other's). A subset with only these 2 genes is able to classify 31 out of 34 samples in the leukemia test dataset and 3 samples remain as unclassified. When a subset of top 4 genes (2642, 4050, 2020, and 1) is used, 32 out of 34 samples can be predicted correctly with 2 remain as unclassified. There are four genes (2642, 2020, 2348, and 3056) in Table 8, which were identified by the previous researchers in their 50 genes most highly correlated with the ALL–AML class distinction (Golub et al. 1999). With the method of Golub et al., 29 out of 34 test samples could be classified correctly, while the DGL classified all of them correctly. Moreover, the DGL method, by selecting sets of genes based on their joint ability to discriminate, can identify genes that are important jointly, but do not discriminate individually. This indicates that the DGL method has potential in identifying genes that not only discriminate between the ALL and AML but also distinguish existing subtypes without applying any prior knowledge.

5 Conclusion

DNA microarrays make it practical, for the first time, to survey the expression of thousands of genes under thousands of conditions. This technology makes it possible to study the expression of all of the genes at once. Large-scale expression profiling has emerged as a leading technology in the systematic analysis of cellular physiology. However, method development for analyzing gene expression data is still in its infancy.

Table 8 The genes appear more than once in the leukemia class predictor

Rank	Gene IDs	Frequency	Gene accession number	Gene description
1	2642	16	U05259_rna1	MB-1 gene
2	4050	16	X03934	GB DEF = T-cell antigen receptor gene T3-delta
3	2020	6	M55150	Fumarylacetoacetate (FAH)
4	1	6	AFFX-BioB-5	AFFX-BioB-5_at (endogenous control)
5	3565	4	U66048	Clone 161455 breast expressed mRNA from chromosome X
6	7129	4	Z78285_f	GB DEF = mRNA (clone 1A7)
7	2348	3	M91432	Acyl-coenzyme A dehydrogenase (ACADM), C-4 to C-12 straight chain
8	4991	3	Y09615	GB DEF = mitochondrial transcription termination factor
9	2852	2	U18004	HSU18004 Homo sapiens cDNA
10	3056	2	U32944	Cytoplasmic dynein light chain 1 (hdlc1) mRNA
11	5826	2	HG3125-HT3301_s	Estrogen receptor (Gb:S67777)
12	501	2	D50931	KIAA0141 gene
13	714	2	D87443	KIAA0254 gene
14	2327	2	M88282	T-cell surface protein tactile precursor
15	1250	2	L08424	Achaete scute homologous protein (ASH1) mRNA

The DGL optimization uses a mathematical method based on orthogonal sets of numbers. By slicing the multi-dimensional parameter space with a horizontal plane of the objective function, with each parameter independent of the others, a peak is always surrounded by a slope. By finding all regions in which the objective function has values above that of the plane, one can narrow the search region. After finding the boundary of all the isolated regions where this occurs, the plane is raised again, and the process repeated.

Orthogonal arrays are immensely important in all areas of human investigation. In statistics they are primarily used in designing experiments. An orthogonal array (OA) is an array of numbers constructed by utilizing orthogonal Latin squares, one can form an array of several dimensions which are orthogonal to each other, and therefore allow the calculation of a resultant using many interdependent variables. Combining OA's sampling with function domain contraction techniques, results in an optimization with two desirable properties. First, the number of function evaluations can be greatly reduced, and second, there is a guarantee of finding the global optimum solution. In this study, a carefully selected OA was successfully used for conducting an orthogonal search space sampling.

Using an orthogonal array and other mathematical techniques, it is practical to develop a global optimization program for cancer classification and validation on a desktop computer. The primary advantages of this technique are that the global optimum is always found, excellent solutions can be found with little prior knowledge, and the new objective functions can be created according to whatever combination of parameters are required.

The mathematical procedures used in this form of global optimization are possible to apply to a variety of other previously unsolved problems relating to the resultant of dependent variables, including experimental design and manufacturing variations. There are many other approaches people have adopted, but until now (with the exception of scanning), they all depend either on a starting design, some form of local optimization or some random variation. Each method will usually give rise to different solutions. For gene subsets using a large number of genes, these are still the only methods possible. In contrast, the DGL optimization described here is a methodical global method.

The proposed pyramidal hierarchy of the predictor for classification can effectively improve the signal to noise ratio in mining the high-dimensional microarray datasets. While the research in cancer classification with microarray expression data is the first to benefit from this method, the mathematical procedures, DGL global optimization, used in this study are also applicable to a variety of other unsolved problems related to linked multi-variable problems. The application of this technique will undoubtedly have implications well beyond cancer classification application.

It is still too early to predict what the ultimate impact of microarray will be on our understanding of cancer although the possibility of the accurate diagnosis of cancers based on microarray expressions has emerged. This innovative research truly brings to light one of the hardest problems yet; the ability to accurately classify medical neoplasm. The DGL method provides a precise diagnostic tool which can find the true global optima with questions relating to gene malignancy.

Furthermore, genetic screening for diseases are playing an increasingly important role in preventative medicine; if we can detect the presence of disease or predict the malignancy through microarray expression data with a desktop computer, before clinical diagnosis, a more efficient and clear cut treatment plan can be formulated, eliminating the possibility of clinician bias. More importantly, an unbiased and digital data-based approach can be easily applied to distinctions relating to future clinical outcome, such as drug response or survival. In cancer research, fundamental mechanisms that cut across distinct types of cancers could also be discovered through mining microarray data by the DGL global strategies.

Acknowledgments I wish to thank Jesse Li of the third-year medicine student at the University of Western Australia for his useful suggestions and assistance in preparing the final version of this paper.

References

Abruzzo LV, Wang J, Kapoor M, Medeiros LJ, Keating MJ, Highsmith WE, Barron LL, Cromwell CC, Coombes KR (2005) Biological validation of differentially expressed genes in chronic lymphocytic leukemia identified by applying multiple statistical methods to oligonucleotide microarrays. J Mol Diagn 7(3):337–345

Abul O, Alhajj R, Aruk Polat F, Barker K (2005) Finding differentially expressed genes for pattern generation. Bioinformatics 21:445–450. doi:10.1093/bioinformatics/bti189

Aittokallio T, Kurki M, Nevalainen O, Nikula T, West A, Lahesmaa R (2003) Computational strategies for analyzing data in gene expression microarray experiments. J Bioinform Comput Biol 1(3):541–586. doi:10.1142/S0219720003000319

Aizenberg I, Myasnikova E, Samsonova M, Reinitz J (2002) Temporal classification of Drosophila segmentation gene expression patterns by the multi-valued neural recognition method. Math Biosci 176(1):145–159. doi:10.1016/S0025-5564(01)00104-3

Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu J, Lu L, Lewis DB, Tibshirani R, Sherlock G, Chan WC, Greiner TC, Weisenburger DD, Armitage JO, Warnke R, Levy R, Wilson E, Grever MR, Byrd JC, Botstein D, Brown PO, Staudt LM (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403:503–511. doi:10.1038/35000501

Allison DB, Cui X, Page GP, Sabripour M (2006) Microarray data analysis: from disarray to consolidation and consensus. Nat Rev Genet 7:55–65. doi:10.1038/nrg1749

Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by

clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci USA 96:6745–6750. doi:10.1073/pnas.96.12.6745

Anderle P, Duval M, Draghici S, Kuklin A, Littlejohn TG, Medrano JF, Vilanova D, Roberts MA (2003) Gene expression databases and data mining. Biotechniques 34(Suppl):S36–S44

Baggerly KA, Coombes KR, Hess KR, Stivers DN, Abruzzo LV, Zhang W (2001) Identifying differentially expressed genes in cDNA microarray experiments. J Comput Biol 8(6):639–659. doi:10.1089/106652701753307539

Barrett MT (2005) Stacking the chips for biological discovery. Nat Genet 37:S1. doi:10.1038/ng1574

Bassett DEB Jr, Eisen MB, Boguski MS (1999) Gene expression informatics—it's all in your mine. Nat Genet 21 (Suppl):51–55. doi:10.1038/4478

Bergmann S, Ihmels J, Barkai N (2003) Iterative signature algorithm for the analysis of large-scale gene expression data. Phys Rev E Stat Nonlin Soft Matter Phys 67(3):031902. doi:10.1103/PhysRevE.67.031902

Bowtell DDL (1999) Options available—from start to finish—for obtaining expression data by microarray. Nat Genet 21 (Suppl): 25–32. doi:10.1038/4455

Brown PO, Botstein D (1999) Exploring the new world of the genome with DNA microarrays. Nat Genet 21 (Suppl):33–37. doi:10.1038/4462

Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares M Jr, Haussler D (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc Natl Acad Sci USA 97:262–267. doi:10.1073/pnas.97.1.262

Cheung VG, Morley M, Aguilar F, Massimi A, Kucherlapati R, Childs G (1999) Making and reading microarrays. Nature 402:15–19. doi:10.1038/46898

Chi H-W, Bloebaum CL (1996) Mixed variable optimization using Taguchi's orthogonal arrays. Struct Multidiscip Optim 12(2–3): 147–152

Cho JH, Lee D, Park JH, Lee IB (2004) Gene selection and classification from microarray data using kernel machine. FEBS Lett 571(1–3):93–98. doi:10.1016/j.febslet.2004.05.087

Churchill GA (2002) Fundamentals of experimental design for cDNA microarrays. Nat Genet 32(Suppl):490–495. doi:10.1038/ng1031

Cochran WG, Cox GM (1957) Experimental designs, 2nd edn. Wiley, New York

Debouck C, Goodfellow PN (1999) DNA microarrays in drug discovery and development. Nat Genet 21 (Suppl):48–50. doi:10.1038/4475

Dey A, Mukerjee R (1999) Fractional factorial plans. Wiley, New York

Duggan DJ, Bittner M, Chen Y, Meltzer P, Trent JM (1999) Expression profiling using cDNA microarrays. Nat Genet 21(Suppl):10–12. doi:10.1038/4434

Enright AJ, Iliopoulos I, Kyrpides NC, Ouzounis CA (1999) Protein interaction maps for complete genomes based on gene fusion events. Nature 402:86–90. doi:10.1038/47056

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286:531–537. doi:10.1126/science.286.5439.531

Hacia JG (1999) Resequencing and mutational analysis using oligonucleotide microarrays. Nat Genet 21 (Suppl):42–47. doi:10.1038/4469

Hedayat AS, Sloan NJA, Stufken J (1999) Orthogonal arrays—theory and applications. Springer, New York

Horst R, Pardalos PM (1995) Handbook of global optimization. Kluwer, The Netherlands

Li D (2006). http://www.scis.ecu.edu.au/dli

Li D, Nathan B (1996) Global optimization advances multivariable thin-film design. Laser Focus World 5:135–136

Li D, Smith C (1996) A new global optimization algorithm based on Latin Square theory. In: Proceedings of 1996 IEEE international conference on evolutionary computation, pp 628–630. ISBN: 0-7803-2902-3

Li L, Darden TA, Weinberg CR, Levine AJ, Pedersen LG (2001a) Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbour method. Comb Chem High Throughput Screen 4(8):727–739

Li L, Weinberg CR, Darden TA, Pedersen LG (2001b) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17(12):1131–1142. doi:10.1093/bioinformatics/17.12.1131

Lipshutz RJ, Fodor SPA, Gingeras TR, Lockhart DJ (1999) High density synthetic oligonucleotide arrays. Nat Genet 21 (Suppl): 20–24. doi:10.1038/4447

Liu JJ, Cutler G, Li W, Pan Z, Peng S, Hoey T, Chen L, Ling XB (2005) Multiclass cancer classification and biomarker discovery using GA-based algorithms. Bioinformatics 21(11):2691–2697. doi:10.1093/bioinformatics/bti419

Loh W (1996) A combinatorial central limit theorem for randomized orthogonal array sampling designs. Ann Stat 24(3):1209–1224. doi:10.1214/aos/1032526964

Marcotte EM, Pellegrini M, Thompson MJ, Yeates TO, Eisenberg DA (1999) A combined algorithm for genome-wide prediction of protein function. Nature 402:83–86. doi:10.1038/47048

Montgomery DC (1997) Design and analysis of experiments, 4th edn. Wiley, New York

Ooi CH, Tan P (2003) Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics 19(1):37–44. doi:10.1093/bioinformatics/19.1.37

Owen AB (1992) Orthogonal arrays for computer experiments: integration and visualization. Stat Sin 2(2):439–452

Owen AB (1994) Lattice sampling revisited: Monte Carlo variance of means over randomized orthogonal arrays. Ann Stat 22(2):930–945. doi:10.1214/aos/1176325504

Peng S, Xu Q, Ling XB, Peng X, Du W, Chen L (2003) Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. FEBS Lett 555:358–362. doi:10.1016/S0014-5793(03)01275-4

Perou CM, Jeffrey SS, van de Rijn M, Rees CA, Eisen MB, Ross DT, Pergamenschikov A, Williams CF, Zhu SX, Lee JC, Lashkari D, Shalon D, Brown PO, Botstein D (1999) Proc Natl Acad Sci USA 96:9212. doi:10.1073/pnas.96.16.9212

Rao CR (1946) Hypercubes of strength d leading to confounded design in factorial experiments. Bull Calcutta Math Soc 38:67–78

Rao CR (1947) Factorial experiments derivable from combinatorial arrangements of arrays. Suppl J R Stat Soc 9:128–139

Rao CR (1949) On a class of arrangements. Proc Edinb Math Soc 8:119–125

Shi Q, Abbruzzese JL, Huang S, Fidler IJ, Xiong Q, Xie K (1999) Constitutive and inducible interleukin 8 expression by hypoxia and acidosis renders human pancreatic cancer cells more tumorigenic and metastatic. Cancer Res 5:3711–3721

Wren JD, Yao M, Langer M, Conway T (2004) Simulated annealing of microarray data reduces noise and enables cross-experimental comparisons. DNA Cell Biol 23(10):695–700. doi:10.1089/dna.2004.23.695

Zhang S, Gant TW (2004) A statistical framework for the design of microarray experiments and effective detection of differential gene expression. Bioinformatics 20(16):2821–2828. doi:10.1093/bioinformatics/bth336