Nucleic Acids Research Advance Access published June 9, 2009

Nucleic Acids Research, 2009, 1–10
doi:10.1093/nar/gkp491

QUBIC: a qualitative biclustering algorithm for analyses of gene expression data

Guojun Li[1,2], Qin Ma[1,2], Haibao Tang[3], Andrew H. Paterson[3] and Ying Xu[1,4,*]

[1]Computational Systems Biology Laboratory, Department of Biochemistry and Molecular Biology, and Institute of Bioinformatics, University of Georgia, Athens, GA 30602, USA, [2]School of Mathematics, Shandong University, Jinan 250100, China, [3]Department of Plant Biology, University of Georgia, USA and [4]College of Computer Science and Technology, Jilin University, Changchun, China

Received March 18, 2009; Revised May 19, 2009; Accepted May 20, 2009

ABSTRACT

Biclustering extends the traditional clustering techniques by attempting to find (all) subgroups of genes with similar expression patterns under to-be-identified subsets of experimental conditions when applied to gene expression data. Still the real power of this clustering strategy is yet to be fully realized due to the lack of effective and efficient algorithms for reliably solving the general biclustering problem. We report a QUalitative BIClustering algorithm (QUBIC) that can solve the biclustering problem in a more general form, compared to existing algorithms, through employing a combination of qualitative (or semi-quantitative) measures of gene expression data and a combinatorial optimization technique. One key unique feature of the QUBIC algorithm is that it can identify all statistically significant biclusters including biclusters with the so-called 'scaling patterns', a problem considered to be rather challenging; another key unique feature is that the algorithm solves such general biclustering problems very efficiently, capable of solving biclustering problems with tens of thousands of genes under up to thousands of conditions in a few minutes of the CPU time on a desktop computer. We have demonstrated a considerably improved biclustering performance by our algorithm compared to the existing algorithms on various benchmark sets and data sets of our own. QUBIC was written in ANSI C and tested using GCC (version 4.1.2) on Linux. Its source code is available at: http://csbl.bmb.uga.edu/~maqin/bicluster. A server version of QUBIC is also available upon request.

INTRODUCTION

DNA microarrays provide a powerful means for probing the functional states of a cell population by allowing simultaneous observation of mRNA expression patterns of all their genes collected over time and/or under different experimental conditions. By comparing the gene expression patterns under different conditions such as cancerous versus healthy tissues, one can possibly derive information about genes associated with a particular cellular condition (e.g. cancerous cells at a specific developmental stage) or even specific biochemical pathways. To analyze the complex microarray data, numerous computational tools have been developed. Among them, clustering of genes based on the similarities of their expression patterns (co-expressed genes) using (traditional) clustering strategies (1–3) represents one of the most popular approaches to microarray data analyses.

The traditional clustering techniques attempt to, in the context of microarray data analyses, partition a set of genes into 'clusters' with similar expression patterns under specified conditions (3), or identify such clusters from an otherwise unstructured microarray data set (4). While useful, such clustering algorithms are known to be inadequate for handling the general gene-expression analyses problems, that often need to identify co-expressed genes under some (to-be-identified) conditions in contrast to finding co-expressed genes under all given conditions. The difficulty in handling the general problem of identifying co-expressed genes is that for any m given conditions, there are 2^m combinations of conditions to consider, making this general clustering problem much more difficult to solve.

A popular way to visualize microarray data for gene expression analyses is to represent the data set as a matrix with rows representing the genes and columns

*To whom correspondence should be addressed. Tel.: 706 542 9779, 706 542 7783; Fax: 706-542-9751/7782; Email: xyn@csbl.bmb.uga.edu

© 2009 The Author(s)
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

representing the conditions (or the other way around) with each element of the matrix representing the relative mRNA abundance of a gene under a specific condition. So identifying groups of genes in a microarray data set that share similar expression patterns under to-be-identified conditions is equivalent to finding submatrices with similar properties. Partitioning a matrix into submatrices with approximately the same values was first studied by Morgan and Sonquist (5) and Hartigan (6). In 2000, Getz *et al*. (7) presented a coupled two-way clustering approach that employs hierarchical clustering to each separate dimension, and then combines the clustering results along each dimension in a somewhat problem-specific manner. It is Cheng and Church (8) who firstly introduced the concept of 'direct clustering', originally proposed by Hartigan (6), to the field of gene expression data analyses, and referred it as 'biclustering', that is to find subsets of conditions under which some (to be identified) subsets of genes have similar expression patterns. Each such submatrix is called a 'bicluster'.

Cheng and Church (8) proposed a quantitative measure, 'mean squared residue', essentially a variability measure, as a guide to search for biclusters in a gene expression data set, which has been adopted by numerous biclustering algorithms (9–11). Recent studies suggest that this measure is useful only for identifying certain classes of co-expressed genes, but not adequate to detect other transcriptionally co-regulated genes (12–14). Another measure was proposed lately by Aguilar-Ruiz (15) to deal with co-regulated genes with 'scaling patterns', which, while more general than the previous measure, was found to be rather challenging to solve algorithmically. Various algorithms have been developed, attempting to solve the biclustering problem as defined either by Cheng and Church (8) or by Aguilar-Ruiz (15) or variations, including the work by Kung *et al*. (16), Li *et al*. (17), Reiss *et al*. (11), Pedro *et al*. (18) and Bryan *et al*. (9,10,19), to name a few, which has led to a number of publicly available computer servers for biclustering analysis of microarray data. Among the published biclustering servers, some have employed combinatorial optimization techniques, such as SAMBA (14), ISA (20), Bimax (13) and NNN (21). A common issue with most of the combinatorial techniques is their high computational complexity, even for the highly simplified cases like using a 0/1 matrix to represent down/up regulations in the observed microarray data.

The state of the art is that the existing biclustering algorithms are generally effective in identifying genes of similar expression values under to-be-identified conditions, but not effective in identifying gene clusters with similar expression patterns in general. Here we report a new biclustering algorithm QUBIC that can effectively and efficiently identify all statistically significant biclusters (allowing overlaps) that cannot be identified by the existing biclustering algorithms and beyond, including both definitions for a biclustering problem given by (8) and (15), as well as finding both positively and negatively correlated expression patterns. We have demonstrated the effectiveness of the QUBIC program and its computational efficiency on a number of benchmark data sets, by comparing it with several salient programs.

METHODS

In our biclustering scheme, we represent the expression values in a qualitative or semi-quantitative manner so that we get a new matrix representation of a gene expression data set under multiple conditions, called a representing matrix, in which the expression level of a gene under each condition is represented as an integer value (see 'Qualitative representation of gene expression data' section for details). We consider that two genes have correlated expression patterns under a subset of conditions if the corresponding integers along the two corresponding rows of the matrix are identical. More generally, we define the similarity level between two genes under a specified set of conditions to be the number of conditions under each of which the two genes have the same (signed) nonzero integer. For applications where identification of negatively correlated genes is desired, we generalize the definition in the same way as the above except that we consider genes with the same corresponding (nonzero) integers but all with opposite signs. We call a submatrix of the above matrix 'feasible' if each pair of rows of the submatrix is either (approximately) the same or the opposite (i.e. the same but with opposite signs across the entire rows). Now our definition of a biclustering problem is to find all the optimal feasible submatrices in a given matrix according to some specified optimization criteria. It is not hard to see that both definitions of a biclustering problem given in (8) and (15) are special cases of our definition. Actually, our definition covers more than just these two cases as we can see from Figures 1A and 2A in the Supplementary Data, where we show two biclustering problems that are more general than both definitions of (8) and (15). Figure 3A in the Supplementary Data shows another biclustering problem in which four biclusters with different expression patterns are implanted in a background matrix. To the best of our knowledge, none of the existing biclustering programs are capable of finding these biclusters.

The key algorithmic idea of our biclustering program is outlined as follows. For a given representing matrix of a microarray data set, we construct a weighted graph G with genes represented as vertices, edges connecting every pair of genes, and the weight of each edge being the similarity level between the two corresponding (entire) rows. Clearly, the higher a weight, the more similar two corresponding rows are. Intuitively, genes in a bicluster should induce a heavier subgraph of G because under a subset of the conditions, these genes have highly similar expression patterns that should make the weight of each involved edge heavier, comparing to the edges in the background. But it should be noted that some heavy subgraph may not necessarily correspond to a bicluster, i.e. genes from a heavy subgraph may not necessarily have similar expression patterns because different edges in a subgraph may have heavier weights under completely different subsets of conditions (see Figure 5 in the Supplementary Data for example). It should also be noted that recognizing all heavy subgraphs in a weighted graph itself is computationally intractable because identification of maximum cliques in a graph is a special case of this, and the maximum

clique problem is a well known intractable problem (NP-hard). So in our solution, we do not directly solve the problem of finding heavy subgraphs in a graph. Instead, we built our biclustering algorithm based on this graph representation of a microarray gene expression data, and tackle the biclustering problem as follows. We find all feasible biclusters (I,J) in the given data set such that min{|I|, |J|} is as large as possible, where I and J are subsets of genes and conditions, respectively.

Our algorithm consists of two key steps: (i) representing a microarray data set using a qualitative matrix as outlined earlier, and (ii) identifying all biclusters in this matrix by finding biclusters one-by-one, where for each bicluster, it starts with the heaviest (unused) edge as a seed to build an initial bicluster and then iteratively recruits additional genes into the current bicluster without violating a prespecified consistency level (see below).

Qualitative representation of gene expression data

The representing matrix is composed of signed integers and 0's, which will be filled based on (i) the decision regarding if each gene has its expression value changed or not, i.e. up- or downregulated, or unchanged under each experimental condition, and (ii) the ranking of all the upregulating conditions for each gene, based on the expression values of the gene under these conditions (a user does not need to preprocess their data, e.g. to determine the fold-change or compute the log values of the raw data); and a similar ranking among all downregulating conditions for each gene. Details follow.

Recognition of unaffected expression values. We use the following method to distinguish those affected expression values from the background data. For each (gene) row i of the original expression data matrix with n rows and m columns, we sort its expression values in the increasing order as follows:

$$v_{i1} \ldots v_{i,s-1} v_{is} \ldots v_{i,c-1} v_{ic} v_{i,c+1} \ldots v_{i,m-s+1} v_{i,m-s+2} \ldots v_{im},$$

where $c = m/2$ and $s-1 = m \times q$, where q is a parameter that can be selected by the user, and its default value in our program is 0.06. A gene i is deemed to be unchanged under condition j if and only if its expression value w_{ij} belongs to the interval $(v_{ic} - d_i,\ v_{ic} + d_i)$, where $d_i = \mathbf{min}\{v_{ic} - v_{is},\ v_{i,\ m-s+1} - v_{ic}\}$. The reason that we define the unaffected expression values in this way is given in the Supplementary Data.

Ranking of regulating conditions. We consider a condition as a downregulating condition for gene i in the above list if its value is $\leq v_{ic} - d_i$, and as an upregulating condition if its value is $\geq v_{ic} + d_i$. We now sort all the upregulating conditions for gene i into the decreasing order of their corresponding expression values, and use this order as the rank of each upregulating condition for gene i; we rank the downregulating conditions in a similar manner except that we sort the relevant gene-expression values into the increasing order, and we use this order as the rank of each downregulating condition for gene i. To distinguish between up- and downregulating conditions, we give each upregulating condition a '+' sign and each downregulating condition a '–' sign. We consider two genes as oppositely regulated under a subset of conditions if they have identical nonzero integers column-wise except with opposite signs.

For practical applications (considering the noisy and stochastic nature of the real gene-expression data), we typically use a predetermined range of ranks, say, rank 1, . . ., 10, which is much smaller than the number of conditions, and then assign multiple conditions with similar expression values for the same gene i into the same rank. The specific range of ranks for a particular application has to be determined using a trial-and-error approach. The QUBIC program provides the flexibility to allow the user to select the levels, r, of ranks for both up- and downregulating conditions with r's default value set to be 1. A basic requirement that needs to be met is that for upregulating conditions, the expression values of rank i should be higher than those of rank $i + 1$ for all $i < r$. A similar requirement needs to hold for the downregulating conditions for each gene. It should be noted that the parameter r allows QUBIC to distinguish up to $r!$ biclusters with different expression patterns in a provided matrix. We omit further discussion about this.

Biclustering through finding a heavy subgraph

Consider a representing matrix M with n rows and m columns as discussed above, representing expression levels of n genes collected under m conditions, and a corresponding weighted graph G with the vertex set V and the edge set E as introduced earlier. Each edge has a weight defined as the number of columns under each of which the two rows (genes) have the same nonzero integer. The basic biclustering problem is to find a submatrix (I, J) of M, with I being a subset of rows (genes) and J a subset of columns (conditions) so that $\mathbf{min}\{|I|, |J|\}$ is maximal and the consistency level of (I, J) is higher than a prespecified value c, $0 < c < = 1.0$, which can be set by the user. In our current program, c is set to be 0.95. The 'consistency level' of a submatrix is defined as the minimum ratio between the number of identical nonzero integers in a column and the total number of rows in the submatrix.

Intuitively, a bicluster should correspond to a maximal and connected subgraph of G consisting of heavier edges, on average, than edges of an arbitrary subgraph not overlapping such bicluster subgraphs, whose total edge-weight is stochastic. Specifically, two genes from the same bicluster should have a heavy edge by nature while two arbitrary genes may have a heavy edge only by chance. Our biclustering algorithm is built on this observation. The algorithm iterates on a set S of seeds (edges). Initially, S is set to be the sorted list of edges in G. An edge $e = g_i g_j$ is considered to be a seed if and only if:

(i) at least one of its genes g_i and g_j is not in any previously identified bicluster, or
(ii) g_i and g_j are in different biclusters $B_1 = (I_1,\ J_1)$ and $B_2 = (I_2,\ J_2)$ with $I_1 \cap I_2 = \emptyset$ and $w(e) \geq \max\{|\ I_1|, |\ I_2|\}$,

where $w(e)$ is the weight of edge e. The algorithm builds an initial bicluster (I, J) based on a selected seed, and then it

expands the bicluster along both the vertical and horizontal directions without violating the preset consistency level, and outputs a bicluster when it cannot be further expanded. Details follow.

Step 1 (Seeding on the representing graph). If S is empty, stop; otherwise, check if the first element of S is a seed. If it is not, remove it from S, and repeat this step; otherwise use it to create a new bicluster as follows: Find all the conditions under which the two genes of the seed have all identical nonzero integer values and set these columns of the two genes as the current bicluster $B = (I, J)$, and go Step 2.

Note that the consistency level of the current bicluster is 1.0. The following step attempts to increase $\min\{|I|, |J|\}$ of the current bicluster by adding additional genes, while maintaining the consistency level at 1.0.

Step 2 (Expansion while mainlining total column-wise consistency). Expand the current bicluster $B = (I, J)$ by adding a new gene (if any) from outside of I which is most consistent with B, giving rise to a new bicluster $B' = (I', J')$, where I' is I after adding the new gene and J' is obtained from J by deleting those columns where the total consistency is lost. If $\min\{|I'|, |J'|\} \geq \min\{|I|, |J|\}$, set B to B', then repeat Step 2; otherwise, if the preset consistency level is 1.0, output B and remove the current seed from S; else go to Step 3.

Step 3 (Expansion allowing less than total consistency). Expand the current bicluster B by adding as many columns as possible without having the consistency level of the bicluster go below c as follows: for each column not in B, if the ratio between the number of identical nonzero integers in the rows of I and $|I|$ is $\geq c$, add it to J. Let $B' = (I' J')$ be the new bicluster and T be the consensus sequence of B' consisting of the dominating elements of the columns of B', where the dominating element is the element with the highest frequency in the column; add as many rows as possible to B' such that each new row has at least $|I'|c$ identical nonzero integers to those of T. Go to Step 4.

We also include negatively co-regulated genes, if any, into our biclusters by executing the following step.

Step 4 (Expansion by adding oppositely regulated genes). Continue to expand the current bicluster B by adding oppositely regulated genes to it: let T be the consensus sequence of B; add as many rows as possible to B such that each added row has at least $|I'|c$ identical nonzero integers but with opposite signs to those of T. Output B and go to Step 1.

The algorithm has a few unique and strong features worthy mentioning: (i) if a significant bicluster is being built but not completed in Step 2 for some reason, leading to a failure of not recognizing the bicluster, this problem could be remedied later with multiple chances by using other edges of the bicluster as seeds; (ii) the algorithm is able to find biclusters not only of positively co-regulated genes but also negatively co-regulated genes; (iii) the program allows a user to provide a set of seeds and build biclusters based on the provided seeds. This capability is included based on the consideration that a biologist may be interested in finding related genes to a specific set of genes; and (iv) although the algorithm is greedy in nature, it does not in general suffer from the issue of getting stuck in local optima since it uses multiple starting points (seeds) to find each bicluster. Our application results strongly indicate this is the case for the program. The pseudo code of the algorithm is provided in the Supplementary Data.

Parameters of QUBIC

QUBIC has a number of parameters, namely, the range r of possible ranks, the percentage q of the regulating conditions for each gene, the required consistency level c for a bicluster, the desired number o of the output biclusters, and the control parameter f for overlaps among to-be-identified biclusters. For each of these parameters, we allow the user to adjust the default value to provide some flexibility.

The parameters r and q affect the granularity of the biclusters. A user can start with a small value of r (the default value is 1 so the corresponding data matrix consists of values '+1', '–1' and '0'), evaluate the results, and then use larger values (should not be larger than half of the number of the columns) to look for fine structures within the identified biclusters. The choice of q's value depends on the specific application goals; that is if the goal is to find genes that are responsive to local regulators, we should use a relatively small q-value; otherwise we may want to consider larger q-values. The default value of q is 0.06 in QUBIC (this value is selected based on the optimal biclustering results on simulated data). The default value of c is 0.95, and o's default value is 100. In addition, we have a parameter f to control the level of overlaps between to-be-identified biclusters (not discussed in the above algorithm); its default value is set to 1 to ensure that no two reported biclusters overlap more than f. QUBIC also provides the option that a user can skip the step of using ranks to represent the actual gene expression values to go directly to the biclustering step on the provided matrix.

RESULTS

We now show the application results of QUBIC first on a number of benchmark data sets developed by Prelic *et al.* (13) and on some simulated data sets constructed by ourselves. The application results on these data sets indicate that our program outperforms the existing and popular biclustering tools, such as SAMBA (14), ISA (20), BIMAX (13), RMSBE (22) and a hierarchical clustering method (HCL) in both the identification accuracy and the computational efficiency. To test the boundaries of our program, we have constructed simulated data sets with tens of thousands of genes under thousands of conditions. The algorithm can find all the embedded biclusters from such large data sets within several minutes on a desktop PC workstation. We then applied the algorithm to actual biological data, and derived a number of new insights

about these microarray data. For all the tests, we have used the following parameters: $r = 1$, $q = 0.06$, $c = 0.95$, $o = 100$, $f = 1$ (unless stated otherwise), and all results are tested on a 64-bit machine.

Applications on Prelic's benchmark data sets

We have tested QUBIC on a benchmark set proposed by Prelic *et al.* (13), which consists of two types of biclusters, constant biclusters and coherent biclusters (23). It is easy to check that both are special cases of our definition of a bicluster and the details about the construction of the benchmark sets can be found in (13).

We have compared our algorithm with four existing algorithms, BIMAX (13), Iterative Signature Algorithm (ISA) (20), SAMBA (14) and HCL but did not include three earlier biclustering algorithms, Cheng–Church method (CC) (8), xMotif (24) and OPSM (12), since they were shown to have rather low performance accuracy (below 50%) in recovering implanted biclusters by previous studies (13,22). In this study, we have used the BIMAX, ISA and HCL algorithms implemented in BICAT (25) and the SAMBA algorithm implemented in EXPANDER (26); both software packages are publicly available. In addition, we included a recently published biclustering algorithm RMSBE (22). The parameters for running these biclustering algorithms were taken either from their default settings or following the parameters suggested by the original authors (see the Supplementary Data on our website at: http://csbl.bmb.uga.edu/~maqin/bicluster/benchmark.html). Preprocessing and postprocessing were performed in a consistent manner with the previous benchmark study (13).

Overall on the Prelic data sets, we found that QUBIC has consistently performed the best in the most general case. It appears that though ISA has the marginal advantage (8%) over QUBIC on the 'noisy' case, its performance drops up to 90% compared to its performance without overlaps when the degree of overlap among coherent biclusters is 10 [see details in Figure 4D in the Supplementary Data]. A more detailed description of the methods' performance on all the Prelic data sets can be found in Figure 4 in the Supplementary Data.

Applications on our simulated data sets

As discussed earlier, biclusters with scaling patterns were considered to be a very challenging problem for any of the existing biclustering algorithm (15). It should be noted that a bicluster with scaling patterns is a special case of our definition of biclusters because a bicluster with scaling patterns in original expression data matrix corresponds to a bicluster with identical rows in its representing matrix. Here we consider two scenarios similar to those of Prelic's benchmark: (i) matrices with varying levels of noise, and (ii) matrices with varying degrees of overlap among the biclusters. We have constructed two sets of gene expression data, for scenario 1 with scaling patterns. For scenario 2, we have constructed one set of gene expression data where the background variation parameter σ was set to 0, and all entries of the first (last) two rows were set to 1 (–1) so that we can simulate the situation where some transcription factors regulate more than one transcriptional modules, i.e. all the implanted biclusters shared the first two and the last two genes. Further construction details can be found in the Supplementary Data.

On all these biclustering problems, our method achieves the optimal identification results almost in every case and always has the best performance among the five programs listed in the 'Applications on Prelic's benchmark data sets' section. In Figure 1A, we can see that all the methods except for RMSBE (with accuracy lower than 20%) achieve almost the optimal identification results. This is not surprising since the problem given in Figure 1A is not much different from the previous test case in Figure 4A in the Supplementary Data. On the more challenging case, as shown in Figure 1B, we start to see some substantial differences in identification accuracies between our and the other programs. For example, when $\sigma = 0.25$, QUBIC

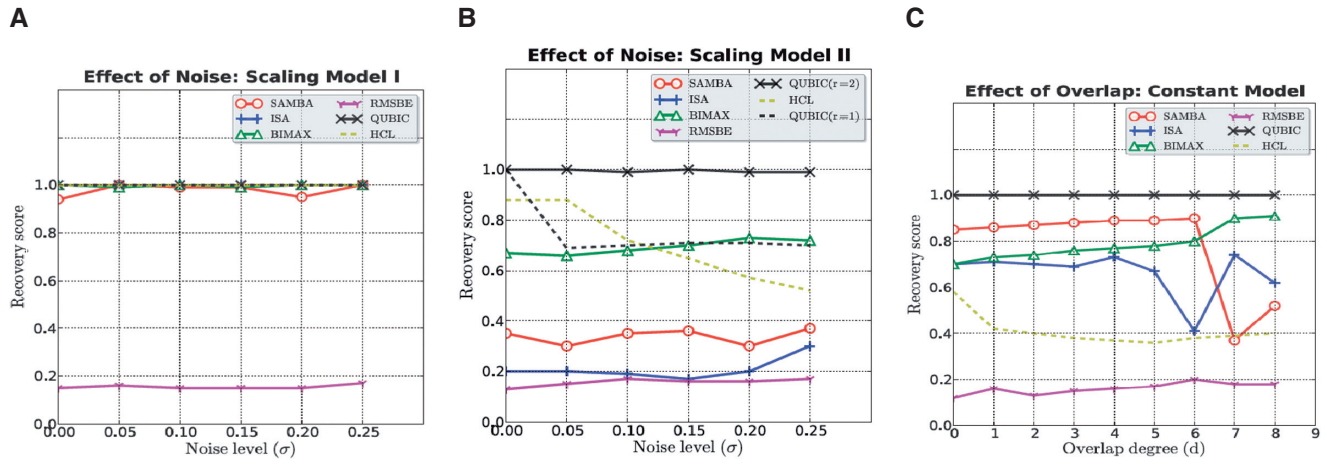

Figure 1. Comparison of recovery accuracy of QUBIC with the other five algorithms. The analysis reveals both the effects of increasing noise levels 'scaling' (**A** and **B**) models and varying degrees of overlapping for 'constant' (**C**) models. Note that the recovery score is calculated similarly to BIMAX using $S^*_G(M_{opt}, M) = \frac{1}{|M_{opt}|} \sum_{G_{opt} \in M_{opt}} \max_{G \in M} \frac{|G_{opt} \cap G|}{|G_{opt} \cup G|}$, where Mopt is the set of implanted biclusters, M is the set of recovered biclusters and G is for genes sets within the bicluster.

with $r = 2$ can achieve almost the optimal identification results (note the accuracy of QUBIC with $r = 1$ is 69%), while the other programs have rather low identification accuracies. Specifically, the accuracies by SAMBA, ISA and RMSBE are all below 50% while BIMAX and HCL are relatively better, at 72% and 52%, respectively. When oppositely regulated genes are considered, we see an even larger difference between our program and the others. Specifically, we have implanted submatrices (see Figure 3 in the Supplementary Data) having some rows or columns with their sums being (approximately) zero, QUBIC finds the implanted submatrices with ~99% accuracy while none of the other programs had better than 40% identification accuracy (by HCL). The detailed information on this is provided as the Supplementary Data on our website (see the performance results in Figure 6 in the Supplementary Data).

We have compared our performance with a recently published program, BUBBLE (19), which is designed to solve biclustering problems with scaling patterns. We have tried the same comparisons as above but found that the BUBBLE program is rather difficult to use and run a large number of samples using it so we compared our program with BUBBLE only on three data sets, representing three different patterns. Overall, QUBIC substantially outperforms BUBBLE on all these data sets, and the detailed performance comparisons are given in Table 9 in the Supplementary Data.

Computational efficiency of QUBIC

To demonstrate the computational efficiency of QUBIC, we have generated a number of large gene-expression data sets ranging from 2000 to 20 000 genes and 1000 conditions (these data sets are available from our website for download; and further details about these data sets can be found in the Supplementary Data). We have run our program and the other five programs on these large test sets on a desktop computer (2.66 GHz Intel Core, 2 Duo CPU, and 4 GB memory). Figure 7 in the Supplementary Data gives the computing time by our program. QUBIC finds the correct biclusters in a few minutes time, essentially independent of any parameters used in the program except for the parameter o while none of the other programs can solve the identification problem when the number of genes goes beyond 12 000. We also tested all the five programs on a real microarray data set with 54 675 transcripts and 18 conditions (an ovarian cancer microarray data set generated by our lab, and it will become available on our website when that paper is published) (Cui *et al.*, manuscript to be submitted). QUBIC finds 100 biclusters in about 5 min.

Applications on global transcriptional data sets

We now evaluate QUBIC on global microarray gene-expression data collected from two different organisms (*Escherichia coli* and yeast). When analyzing the whole transcriptome microarray data, one challenging problem is to find the 'transcriptional modules', which represent modular components in the (global) gene regulatory network, defined as a set of tightly co-regulated genes along with a set of associated conditions that trigger the co-regulation (20), making it a natural application problem for the biclustering methods. It is known that some transcriptional modules show co-regulations only under a narrow range of conditions and have weak global correlations among their gene expression patterns, therefore not easily detectable by the traditional clustering methods. In addition, some transcriptional modules may overlap due to the combinatorial regulation by multiple transcriptional factors (20), which would also complicate the use of the traditional clustering techniques. The goal of this exercise is to test the effectiveness of our biclustering algorithm in identifying such transcriptional modules.

Our first test case includes the microarray gene expression data for 4217 *E. coli* genes collected under 264 conditions from the M3D database (*E. coli* array version 4 build 3) (27). The values in the original microarray data set are log2 values of the fluorescence intensities. The goal of our analysis is to identify biclusters hidden in the microarray data, and study their relationships with known biological pathways, as defined by the GO functional classification scheme (28), as well as by the KEGG pathways (29) and the 'EcoCyc' database (30).

For each identified bicluster, we use the P-value of its most enriched functional class (biological process) as the P-value of the bicluster. Specifically, the probability of having r genes of the same functional class in a bicluster of size n from a genome with a total of N genes can be computed using the following hypergeometric function (31), where P is the percentage of that functional class among all functional classes of genes encoded in the whole genome,

$$\Pr(r|N, p, n) = \frac{\binom{pN}{r}\binom{(1-p)N}{n-r}}{\binom{N}{n}}$$

For each functional class C, we calculate the P-value of our current bicluster enriched with C genes as the probability of selecting at least r genes of the same functional class in the bicluster, where r is the actual number of C genes present in the current bicluster. We then use the smallest P-value among all possible functional classes C as the P-value of the current bicluster. Clearly, the smaller the P-value of a bicluster B is, the more likely that B's genes are from the same biological process. We have run the six biclustering algorithms with their default parameters on this data set, as introduced in 'Applications on Prelic's benchmark data sets' section.

To compare the biclustering results by different algorithms, we have applied a clean-up procedure introduced in Prelic *et al.* (13) to remove the substantially overlapping biclusters so that among the survived biclusters, no two overlap more than 25% of their sizes. For each algorithm, we calculated the proportion of biclusters that have significant P-values (below a pre-selected P-value cutoff) among the survived biclusters after the clean-up step. Then, we score each algorithm using the ratio between

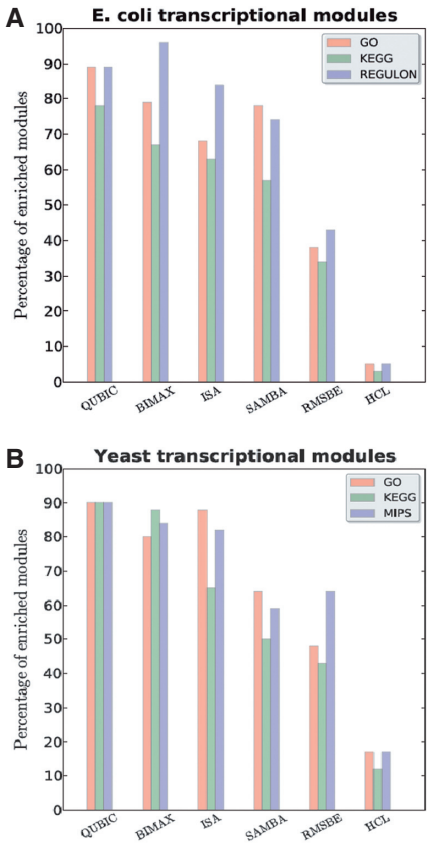

Figure 2. (**A**) Proportions of *E. coli* biclusters that have significant overlap ($P < 0.01$) with GO biological processes, KEGG pathways and experimentally verified regulons. (**B**) Proportions of yeast biclusters that are statistically enriched ($P < 0.01$) in GO biological processes, KEGG pathway and MIPS functional catalog.

the number of significant biclusters and the number of the survived biclusters.

Among the six tested algorithms, QUBIC consistently show the highest enrichment ratios except for the regulon classification from the 'EcoCyc' database. Specifically, when the *P*-value cutoff is 0.01, 89% of the QUBIC biclusters show substantial enrichment with GO biological processes, 89% of the QUBIC results show significant overlap with known regulons and 78% enriched in KEGG pathways (29). The detailed comparisons with other programs are given in Figure 2A. Although the performance of BIMAX (96%) is better than QUBIC for the regulon classification category, we found that 59% of the QUBIC biclusters have *P*-values $<10^{-6}$ while only 48% of the BIMAX biclusters have *P*-values $<10^{-6}$. This suggests that individual QUBIC clusters are more significant than those generated by BIMAX. Indeed, on a case-by-case basis, the biclusters from QUBIC have higher enrichment ratios for more functional classes than those by all the other algorithms (Table 1). As an example, the flagella assembly pathway in *E. coli* is known to consist of 38 genes. Out of these genes, one QUBIC cluster includes 33 out of the total of 52 genes in the cluster, which compares to 20 out of 28 by BIMAX, 35 out of 92 by ISA, 22 out of 220 by MSBE and 36 out of 202 by SAMBA. This comparison highlights the overall better performance by QUBIC among all the programs in terms of their combined identification sensitivity and specificity.

On our second test, we used a yeast (*Saccharomyces cerevisiae*) microarray data set (32). Similar to the *E. coli* data analysis, we evaluated each bicluster (after removing the substantially overlapping biclusters) generated by different algorithms in terms of their functional enrichments based on GO biological processes, MIPS yeast functional catagories (33) and KEGG pathways. From Figure 2B, we can see that QUBIC has the highest functional enrichment among all the tested algorithms based on the three classifications.

Through the above comparative analyses on the performance of six algorithms, we have shown that QUBIC is capable of revealing high quality biclusters in both prokaryotic and eukaryotic microarray expression data, and the genes in each bicluster show strong correlations with known functions and pathways. This study thus suggests the potential in extracting the substructures of metabolic and regulatory networks from gene expression data under multiple conditions using a biclustering method, providing a new and useful tool for biological pathway and network reconstruction.

One potential issue with the above *P*-value based analysis is that the *P*-value is bicluster size-dependent, and hence larger bi-clusters tend to have more significant *P*-values. This is clearly not a unique problem to the biclustering result analysis as other bioinformatics problems, such as the problem of *cis* regulatory motif finding, also face the same issue. Further studies will be carried out aiming to make our *P*-value calculation size independent.

Signature identification for cancer subtyping

We now extend the application of our biclustering algorithm to the problem of cancer subtype classification. The basis of this analysis is that pathways unique to specific cancer subtypes may get activated across the majority of the patients of the subtypes, and hence the genes in these pathways can be possibly used as a signature for specific subtypes. Apparently this problem can be formulated as a biclustering problem on microarray gene expression data. Actually, there have been several studies that used biclustering as part of a larger analysis pipeline to do cancer subtyping (34).

We have used the leukemia data collected by Armstrong *et al.* (35) and searched for biclusters that might be characteristic to different leukemia subtypes (ALL, MLL and AML). This data set consists of 12 533 probes from 72 patients of different subtypes of leukemia (24 ALL, 20 MLL and 28 AML patients, respectively), which were produced on Affymetrix U95A oligo-nucleotide arrays. We did pre-processing based on the experiment background as detailed in the Supplementary Data.

Using QUBIC, we have identified a total of 192 biclusters in the data set (the parameter o is set to 500 and the output results are available on our website). We made the following observations about the predicted biclusters: 17 biclusters contain samples (conditions) from only one cancer subtype, 89 biclusters have samples from two subtypes and 86 biclusters from all three subtypes (see

Table 1. Functional enrichments in the biclusters by different programs for *E. coli* respect to KEGG classes

	QUBIC	*BIMAX*	*ISA*	*MSBE*	*SAMBA*
ABC transporters – Organism-specific	**6e-08 (61%)**	1e-04 (30%)	na	ns	ns
Aminosugars metabolism	**2e-04 (18%)**	na	na	ns	1e-03 (7%)
Arginine and proline metabolism	**4e-03 (12%)**	na	ns	4e-03 (4%)	ns
Ascorbate and aldarate metabolism	**2e-04 (21%)**	2e-03 (22%)	na	2e-03 (3%)	na
Flagellar assembly	**4e-63 (63%)**	4e-37 (71%)	8e-57 (38%)	8e-18 (10%)	3e-45 (17%)
Fructose and mannose metabolism	na	**3e-05 (30%)**	2e-03 (40%)	9e-05 (7%)	na
Galactose metabolism	**2e-07 (50%)**	ns	na	na	3e-06 (5%)
Glycerophospholipid metabolism	**2e-06 (21%)**	9e-03 (22%)	na	ns	na
Nitrogen metabolism	na	**1e-06 (29%)**	3e-05 (6%)	ns	2e-06 (7%)
Pentose and glucuronate interconversions	6e-04 (20%)	**2e-06 (60%)**	na	ns	ns
Phosphotransferase system (PTS)	na	ns	na	2e-03 (7%)	**4e-06 (18%)**
Pyrimidine metabolism	na	**7e-05 (36%)**	na	ns	2e-03 (8%)
Ribosome	**4e-47 (44%)**	na	na	4e-38 (37%)	2e-43 (23%)
Sulfur metabolism	**2e-12 (29%)**	1e-10 (11%)	2e-03 (2%)	na	3e-09 (4%)

Values represent *P*-values followed by the enrichment ratios (the number of genes in both class and bicluster/the number of genes in the bicluster). Each value in bold represents the most significant *P*-value for each functional class.
na:– functional class not present in the results.
ns: functional class present in the results but not significant at level of 0.01.

Figure 8 in the Supplementary Data). Although only 17 biclusters were found to have specificity for a particular subtype, these biclusters are highly significant and distinct. Figure 3 gives an example of three selected biclusters that each shows subtype-specificity (BC000, BC002 and BC074). In this example, QUBIC identifies the classical 'checker-board' substructures inside the original microarray data, where the three selected biclusters each corresponds to a particular leukemia subtype, with BC000 specific to ALL, BC074 specific to MLL and BC002 specific to AML (Figure 3).

We found that these subtype-specific biclusters are informative and in most cases consistent with results reported in previous studies (35,36). For example, the MLL cluster (BC071; Figure 3) contains genes involved in multiple hematopoietic lineages, including PROM1 and FLT3 in progenitor cells and CCNA1 in myeloid cells, which were also observed in (35). While some of the genes in these subtype-specific biclusters may not necessarily make good marker genes for hematopoietic lineages, others do, such as those that encode proteins critical for cell-cycle transitions such as CCNA1, CCND3 and CDK5R1/p35. It is also worth noting that we identified two negatively regulated genes in BC002. Specifically, the last two genes (SEPT9 and CCND3) in BC002 are downregulated while the other genes in BC002 are upregulated. This has been observed for CCND3 (36), but the observation on SEPT9 is new, to the best of our knowledge. We believe that these three subtype-specific biclusters are information rich and further analyses could potentially lead to improved understanding about the molecular mechanisms underlying these three subtypes.The biclusters that contain samples from more than one subtype are probably clinically just as informative as the above subtype-specific biclusters. For example, we have found that among the resulting biclusters, three biclusters (BC011, BC040 and BC148) show an opposite trend for different ALL and AML, and one bicluster (BC025) shows an opposite trend for MLL and AML. In particular,

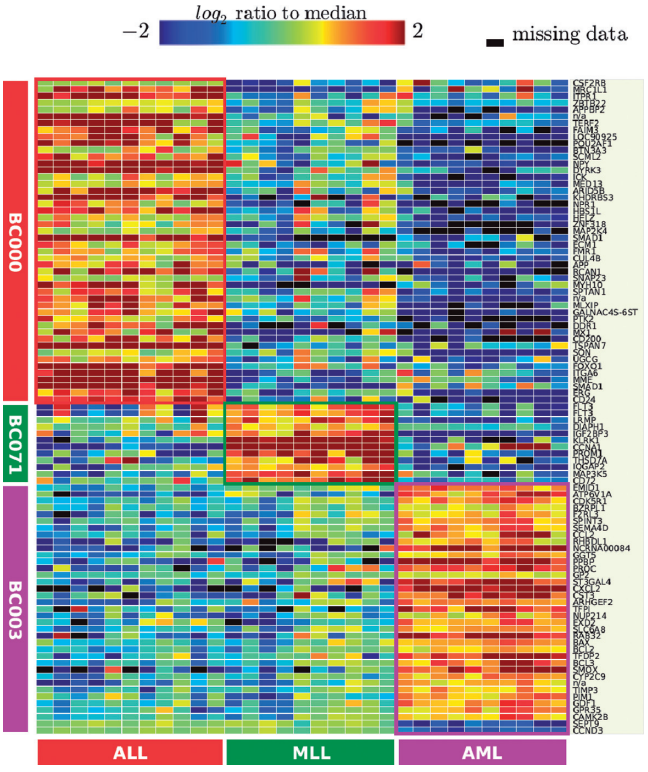

Figure 3. Visualization of three biclusters (BC000, BC002 and BC074), which were selected based on the specificity to certain subtype of leukemia (ALL/AML/MLL). The gene names are given to the right of the heat-map. Some genes are represented twice since there are cases where two different Affymetrix probes are used for the same gene.

within bicluster BC011, samples from ALL patients are all downregulated, while samples from AML patients are all upregulated; BC148 shows exactly the opposite pattern to that of BC011 where ALL samples are upregulated and AML samples are downregulated. These biclusters would contain candidates of selectively expressed genes

for needed molecular targets. Note that this was not possible using some other biclustering algorithms such as BIMAX, since BIMAX only deals with binary data (change versus no-change) (13) as opposed to multiple data in our analysis.

As a result of biclustering on the cancer data, we have shown that QUBIC is capable of uncovering genes that are unique to clinically known subtypes of cancers. Our future work will be focused on mining the subtype-specific biclusters, as well as on integration of the program with additional tools into a classification and characterization pipeline in support of cancer studies.

DISCUSSION

The biclustering strategy has been widely used in analyses of gene expression data since it was first proposed in 2000 because it provides a much increased flexibility and analysis power for identifying co-expressed genes under some but not necessarily all conditions, compared to the traditional clustering methods. As of now, most of the existing biclustering algorithms were designed to solve a rather special class of the biclustering problem, specifically attempting to find biclusters that minimize the so-called mean squared residue value. The QUBIC algorithm has proven to be a useful tool for analyzing gene expression data of tens of thousands of genes for discovering complex relationships among genes and conditions that are difficult to detect using existing biclustering methods. The high computational efficiency and the ability to detect subtly correlated expression patterns among genes under certain conditions will make QUBIC a powerful tool for analyses of microarray gene expression data, particularly large data sets. Furthermore, it can be a useful tool in transcriptional regulation network prediction.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We would like to thank Dr Dongsheng Che and Mr Kun Xu for their help and insightful discussions on the work. Also, we thank the useful suggestions by the two anonymous reviewers.

FUNDING

The National Science Foundation (#NSF/DBI-0354771, #NSF/ITR-IIS-0407204, #NSF/DBI-0542119, and #NSF/CCF-0621700); the U.S. Department of Energy's BioEnergy Science Center (BESC) grant through the Office of Biological and Environmental Research; and grants (60873207, 10631070 and 60373025 to G.J.L.) from NSFC and the Taishan Scholar Fund from Shandong Province, China. Funding for open access charge: NSF DBI-0542119.

Conflict of interest statement. None declared.

REFERENCES

1. Yeung,K.Y., Medvedovic,M. and Bumgarner,R.E. (2003) Clustering gene-expression data with repeated measurements. *Genome Biol.*, **4**, R34.
2. McLachlan,G.J., Bean,R.W. and Peel,D. (2002) A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, **18**, 413–422.
3. Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.
4. Xu,Y., Olman,V. and Xu,D. (2002) *Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees*. **18**, 536–545.
5. Morgan,J.N. and Sonquist,J.A. (1963) Problems in the analysis of survey data, and proposal. *J. Am. Stat. Assoc.*, **58**, 415–434.
6. Hartigan,J.A. (1972) Diret clustering of a data matrix. *J. Am. Stat. Assoc.*, **67**, 123–129.
7. Getz,G., Levine,E. and Domany,E. (2000) Coupled two-way clustering analysis of gene microarray data. *Proc. Natl Acad. Sci. USA*, **97**, 12079–12084.
8. Cheng,Y. and Church,G.M. (2000) Biclustering of expression data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **8**, 93–103.
9. Bryan,K. and Cunningham,P. (2008) Extending bicluster analysis to annotate unclassified ORFs and predict novel functional modules using expression data. *BMC Genomics*, **9(Suppl. 2)**, S20.
10. Bryan,K., Cunningham,P. and Bolshakova,N. (2006) Application of simulated annealing to the biclustering of gene expression data. *IEEE Trans. Inf. Technol. Biomed.*, **10**, 519–525.
11. Reiss,D.J., Baliga,N.S. and Bonneau,R. (2006) Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. *BMC Bioinformatics*, **7**, 280.
12. Ben-dor,A., Chor,B., Karp,R. and Yakhini,Z. (2003) Discovering local structure in gene expression data: the order-preserving submatrix problem. *J. Comput. Biol.*, **10**, 373–384.
13. Prelic,A., Bleuler,S., Zimmermann,P., Wille,A., Buhlmann,P., Gruissem,W., Hennig,L., Thiele,L. and Zitzler,E. (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, **22**, 1122–1129.
14. Tanay,A., Sharan,R. and Shamir,R. (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **18(Suppl. 1)**, S136–S144.
15. Aguilar-Ruiz,J.S. (2005) Shifting and scaling patterns from gene expression data. *Bioinformatics*, **21**, 3840–3845.
16. Kung,S.Y., Mak,M.W. and Tagkopoulos,I. (2006) Symmetric and asymmetric multi-modality biclustering analysis for microarray data matrix. *J. Bioinform. Comput. Biol.*, **4**, 275–298.
17. Li,H., Chen,X., Zhang,K. and Jiang,T. (2006) A general framework for biclustering gene expression data. *J. Bioinform. Comput. Biol.*, **4**, 911–933.
18. Carmona-Saez,P., Pascual-Marqui,R.D., Tirado,F., Carazo,J.M. and Pascual-Montano,A. (2006) Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics*, **7**, 78.
19. Bryan,K., Cunningham,P. and Bolshakova,N. (2006) Application of simulated annealing to the biclustering of gene expression data. *IEEE Trans. Inf. Technol. Biomed.*, **10**, 519–525.
20. Ihmels,J., Bergmann,S. and Barkai,N. (2004) Defining transcription modules using large-scale gene expression data. *Bioinformatics*, **20**, 1993–2003.
21. Huttenhower,C., Flamholz,A.I., Landis,J.N., Sahi,S., Myers,C.L., Olszewski,K.L., Hibbs,M.A., Siemers,N.O., Troyanskaya,O.G. and Coller,H.A. (2007) Nearest Neighbor Networks: clustering expression data based on gene neighborhoods. *BMC Bioinformatics*, **8**, 250.
22. Liu,X. and Wang,L. (2007) Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics*, **23**, 50–56.
23. Madeira,S.C. and Oliveira,A.L. (2004) Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **1**, 24–45.
24. Murali,T.M. and Kasif,S. (2003) Extracting conserved gene expression motifs from gene expression data. *Pac. Symp. Biocomput.*, **8**, 77–88.
25. Barkow,S., Bleuler,S., Prelic,A., Zimmermann,P. and Zitzler,E. (2006) BicAT: a biclustering analysis toolbox. *Bioinformatics*, **22**, 1282–1283.

26. Shamir,R., Maron-Katz,A., Tanay,A., Linhart,C., Steinfeld,I., Sharan,R., Shiloh,Y. and Elkon,R. (2005) EXPANDER—an integrative program suite for microarray data analysis. *BMC Bioinformatics*, **6**, 232.
27. Faith,J.J., Driscoll,M.E., Fusaro,V.A., Cosgrove,E.J., Hayete,B., Juhn,F.S., Schneider,S.J. and Gardner,T.S. (2008) Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Res.*, **36**, D866–D870.
28. Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet.*, **25**, 25–29.
29. Kanehisa,M. (2002) The KEGG database. *Novartis Found. Symp.*, **247**, 91–101; discussion 101–103, 119–128, 244–152.
30. Keseler,I.M., Collado-Vides,J., Gama-Castro,S., Ingraham,J., Paley,S., Paulsen,I.T., Peralta-Gil,M. and Karp,P.D. (2005) EcoCyc: a comprehensive database resource for Escherichia coli. *Nucleic Acids Res.*, **33**, D334–D337.
31. Castillo-Davis,C.I. and Hartl,D.L. (2003) GeneMerge—post-genomic analysis, data mining, and hypothesis testing. *Bioinformatics*, **19**, 891–892.
32. Gasch,A.P., Spellman,P.T., Kao,C.M., Carmel-Harel,O., Eisen,M.B., Storz,G., Botstein,D. and Brown,P.O. (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, **11**, 4241–4257.
33. Ruepp,A., Zollner,A., Maier,D., Albermann,K., Hani,J., Mokrejs,M., Tetko,I., Guldener,U., Mannhaupt,G., Munsterkotter,M. *et al.* (2004) The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res.*, **32**, 5539–5545.
34. Kluger,Y., Basri,R., Chang,J.T. and Gerstein,M. (2003) Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res.*, **13**, 703–716.
35. Armstrong,S.A., Staunton,J.E., Silverman,L.B., Pieters,R., den Boer,M.L., Minden,M.D., Sallan,S.E., Lander,E.S., Golub,T.R. and Korsmeyer,S.J. (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.*, **30**, 41–47.
36. Golub,T.R., Slonim,D.K., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J.P., Coller,H., Loh,M.L., Downing,J.R., Caligiuri,M.A. *et al.* (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.