However, the gene modules retrieved in this step have rigid patterns with no relaxation (i.e., noise), which contrasts to real data containing noise and biological flexibility. Thus, in the next step, these "core modules" are compared with each other and merged into bigger modules containing noise. As the final output, we obtain each module consisting of a subset of genes and a subset of experiments. By scrutinizing both gene functions and experimental relationships in each module, we can formulate new and interesting hypotheses on gene functions and experimental groups.

2.1. Rat Chemical Exposure Dataset and Normalization

1. Download gene expression data of chemical effects on rat liver by J & J from the Web site: http://cebs.niehs.nih.gov/. To retrieve the data, select the subject "J&J Hepatotoxicant Library" in the "Display All Studies" page or limit data by the organization name "Johnson and Johnson" in the "Study Characteristics" page. There are 133 toxicochemical groups containing 964 microarray experiments. Go to "View Selected Microarray Data by Studies/Experiments" from the bottom of page, select the dataset, and go to "View Details about Selected Experiment(s)." Then, download microarray data files after entering "Click to Download" page. As the file size is 117.3 MB, downloading will take 5 min to hours depending on the user's network conditions.
2. Unzip downloaded files and select files to analyze. In this study, we select only experiments that have both chemical exposure done and control data collected on the same day. The number of such experiments is 298. Among 9,215 probes in the dataset, we delete low-abundance genes that show no expression in all of the 298 experiments and select only 7,614 probes. Every pair of gene expression values is transformed into log-fold-change abundance by subtracting control values from the chemical exposure after taking $\log_2$ and subtracting the median value in each array (*see* **Note 1**). Finally, the log-fold-change values are normalized to Z-score by $(x - \text{mean})/\text{SD}$.
3. To perform gene functional analysis in later process, check if each probe has a link to UniGene database. Only 5,832 probes have links to UniGene database. Then, to remove probe redundancies, take the average if multiple probes correspond to the same UniGene ID. As a result, we obtain 2,497 averaged probes that have links to UniGene database for 298 chemical exposure experiments.

2.2. Formatting Database by Discretization

1. Convert normalized dataset into rank-ordered discrete data within each experiment. To do this, select one experiment and sort genes by expression value. Then, put all the genes into 10,000 bins of the same size and assign every gene a rank value equal to the number of bins (1st–10,000th from low to high) that it belongs to.

2. Select one gene and make a distribution of rank values for all the 298 experiments. Then, set a discretization parameter and thresholds. In this study, we set the degree of discretization at 3 (±1, 0) and the thresholds at 3% or 5% from each side of top and bottom (i.e., 6% or 10% in total) in rank value distribution. Using these parameters, we assign discrete values to rank values.

2.3. Query Data and Database Compression

1. Given query gene expression data, discretize gene expression values using the same procedure as that employed in the above database formatting process. Use the same degree and thresholds to discretize query data as that used in the database discretization.
2. Compare discretized query to discretized database at the gene level. Then, delete all discrete values that differ from the query in the database. In addition, delete all zero values in the database. This procedure compresses the database to an extremely small size (*see* **Note 2**).

2.4. LCM for Data Mining of Core Gene Modules

1. LCM *(12, 13)* is an ultrafast algorithm for data mining that has been used to retrieve maximum common itemsets from a large list of itemsets called a transaction database (*see* **Note 3**). To apply this algorithm, assign item names to all the existing combinations of gene names and discrete values in the gene expression database. For example, we give item names "a – 1" and "a – 2" to the case that gene "a" has values of both "–1" and "–2" in the database. This procedure converts each gene expression experiment datum into an itemset list that symbolizes gene expression status consisting of gene names and their discrete values.
2. Write itemset list to a file in the format of one experiment in one line. Then, download the LCM program from the Web site: http://research.nii.ac.jp/~uno/codes.htm. Run the LCM program with the itemset list file with parameters of minimum size of gene modules to extract: m genes × n experiments. For example, the input command to run LCM on a Linux machine is: % lcm CqI –l m [input_file] n [output_file] (*see* **Note 3**).

2.5. Merging Redundant Modules

1. Raw gene modules (core modules) extracted by LCM are expected to be highly redundant. To reduce almost the same or quite similar gene modules in output data, merge them if they meet conditions specified by the following procedure. First, sort gene modules by size. Then, select the largest module and merge it with other modules one by one from large to small ones. Suppose we are merging modules A and B. If A and B share genes g1 and g2 and experiments e1 and e2 but A has another gene g3 and B has another experiment e3, the merged module will have genes g1, g2, and g3 and experiments e1, e2,

and e3 by adding missing gene (g3) and experiment (e3) to B and A, respectively. However, if any gene or experiment of the merged module contains inconsistent values, such as missing or different discrete values, the percentages of inconsistencies in each line and row of the merged module should be checked. If the inconsistency in every line and row is less than the threshold (in this study, 0.4 and 0.5 are adopted), execute the merge and replace the larger module with it. (Delete the smaller one.) Repeat this "check and merge" process for this larger module until it reaches the smallest one.

2. Repeat the above "check and merge" process from the next largest module to the smallest module. Once the smallest module is reached, sort the modules by size again and perform "check and merge" from the largest module to the smallest one for a new list of modules.
3. Repeat the above "sort, check, and merge" until no merge happens. Then, output the final (merged) modules to a file (*see* **Note 4**). The whole process from formatting data to merging modules is illustrated in **Fig. 1**. Here an example of two degrees (high and low expressions, or +1 and −1) of discretization is shown.

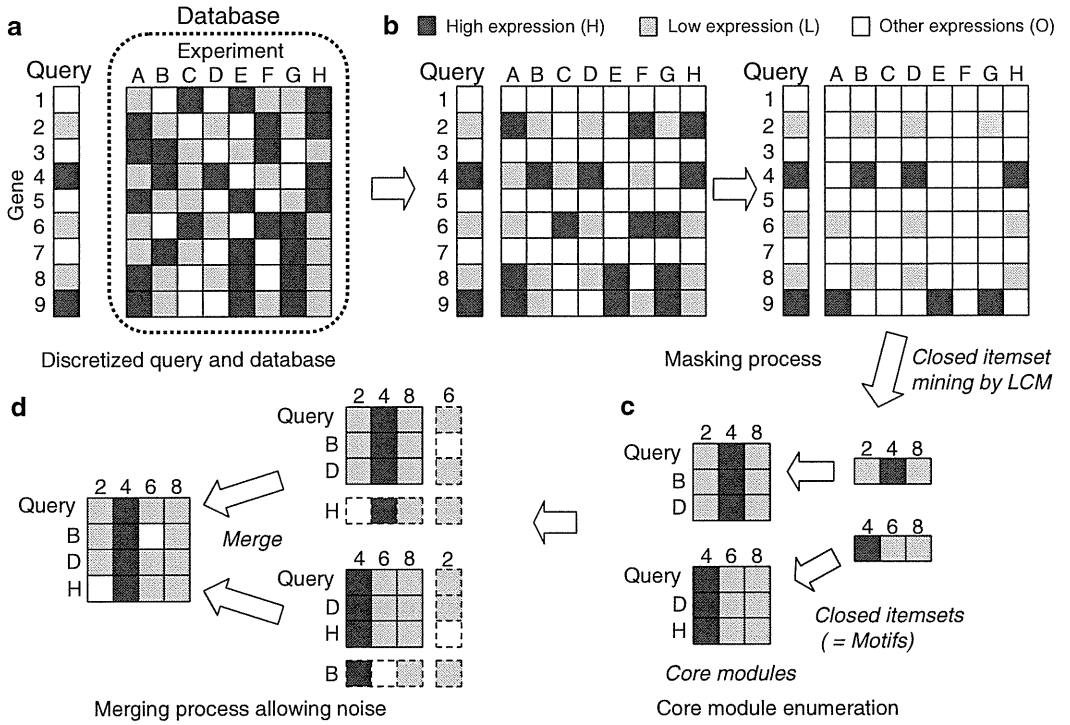


Fig. 1. Whole scheme of gene module search from gene expression database. The system consists of four steps (**a**) discretization of query and database, (**b**) database masking, (**c**) enumeration of core modules, and (**d**) module merging. In this example, the query and the database are discretized into only three degrees "High (+1)," "Low (−1)," and "Other expressions (0)".

2.6. Evaluation by GO and KEGG

Once the final set of modules is obtained, it is necessary to check the biological validities of those modules to verify if the selected parameters (discretization degree, noise threshold, module size, etc.) work properly. To approach this, compare each gene module with known biological functions or pathways to investigate if genes in a single module are statistically "enriched" for a particular category of functions. Here we describe the method of performing categorical enrichment analysis based on Gene Ontology (GO) functions and KEGG biological pathways.

1. Download necessary files from two FTP sites (a) gene2unigene and gene2go.gz files from ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/ for gene name conversion and GO term information, respectively; and (b) gene_map.tab for KEGG pathway map index information and rno_gene_map.tab files for rat specific pathway gene list from ftp://ftp.genome.ad.jp/pub/kegg/pathway/.
2. Take one gene expression module. Convert UniGene IDs into EntrezGene IDs via the cross-reference list in gene2unigene file (*see* **Note 5**).
3. Assign GO function terms to the converted (Entrez-) genes. Obtain four parameters (a) U, the number of EntrezGenes found in the input gene expression data (2,497 UniGenes); (b) M, the number of EntrezGenes assigned to each GO term; (c) k, the number of EntrezGenes in each gene module; and (d) m, the number of EntrezGenes in each GO term found in each module.
4. Assign KEGG pathway map names to the converted (Entrez-) genes. Obtain the four parameters for each pathway map in the same way as the GO terms.
5. Evaluate each GO term and KEGG pathway map names for each gene module with standard hypergeometric distribution statistics by giving m as positives out of k samples and M known positives among the total population of U. To critically assess p value threshold, shuffle gene names (UniGene IDs) in the input dataset and do the same statistical analysis for each module, and use them as a null-distribution model.
6. The numbers of extracted modules for 298 query chemicals under two different discretization thresholds (3% and 5%) are summarized in **Table 1**. Two different noise ratios in the module merging process are also tested in both data.

2.7. Analysis of Common Reactions Among Chemicals by Gene Modules

The main objective in gene module analysis of reverse chemical genomics is to find new functional relationships between different chemicals. Once a set of gene modules annotated by GO and KEGG is obtained, check the common function names for every combination of query chemicals with a significant p value threshold.

Table 1
Numbers of obtained modules under various parameter conditions

	Noise ratio	
Discretization	**0.4**	**0.5**
3%	2,859	2,088
5%	92,100	61,852

1. First, assign *p* values to each gene module by GO and KEGG categorical enrichment analysis as described in 2.6. Choose the most significant (the smallest) *p* value among various functional candidates for a single gene module. Then, plot each module by its (–log) *p* value, as shown in **Fig. 2a**.
2. Plot the same module in which gene IDs are shuffled by its *p* value, as shown in **Fig. 2b**. Compare the two plots. Set the *p* value threshold at the critical point where raw modules are still observed but gene-shuffled modules disappear. Here, we arbitrarily choose 1e – 5.5 and 1e – 4 as the thresholds for GO and KEGG, respectively, for the modules extracted by the parameters of 5% discretization and 0.4 of noise ratio.
3. Take every pair of query chemicals and check if they share common function or pathway names at the above threshold. Store the results.
4. Find biological hypotheses that can explain the obtained relationships among two or more chemicals. For example, in our data, some modules obtained from a query of Benzbromarone are found to affect the pathway of "Biosynthesis of unsaturated fatty acids." This pathway was also found with queries of Fenbufen, Clofibrate, and Dichloroacetate. Three genes are involved in the obtained modules: acyl-CoA thioesterase 3 (Rn.11326), acyl-CoA thioesterase 7 (Rn.6024), and acyl-Coenzyme A oxidase 1, palmitoyl (Rn.31796), all of which are important in unsaturated fatty acid metabolism. This result and information from literature suggest that these chemicals could affect the rat liver in a similar manner that is related to carcinogenesis via PPARα-derived oxidative reaction. **Figure 3b** is an example of the graphical output of these modules produced by the web-based GUI version of the SAMURAI system.

2.8. Usage of SAMURAI System

To enhance the above analysis in a coordinated way, we have developed a gene module extraction and evaluation system called SAMURAI. The free trial version of the program coded in java/C++/Perl is available from http://samurai.cbrc.jp/download/SAMURAI-Progressive/ (*see* **Note 6**). Here we describe briefly the usage of the system.

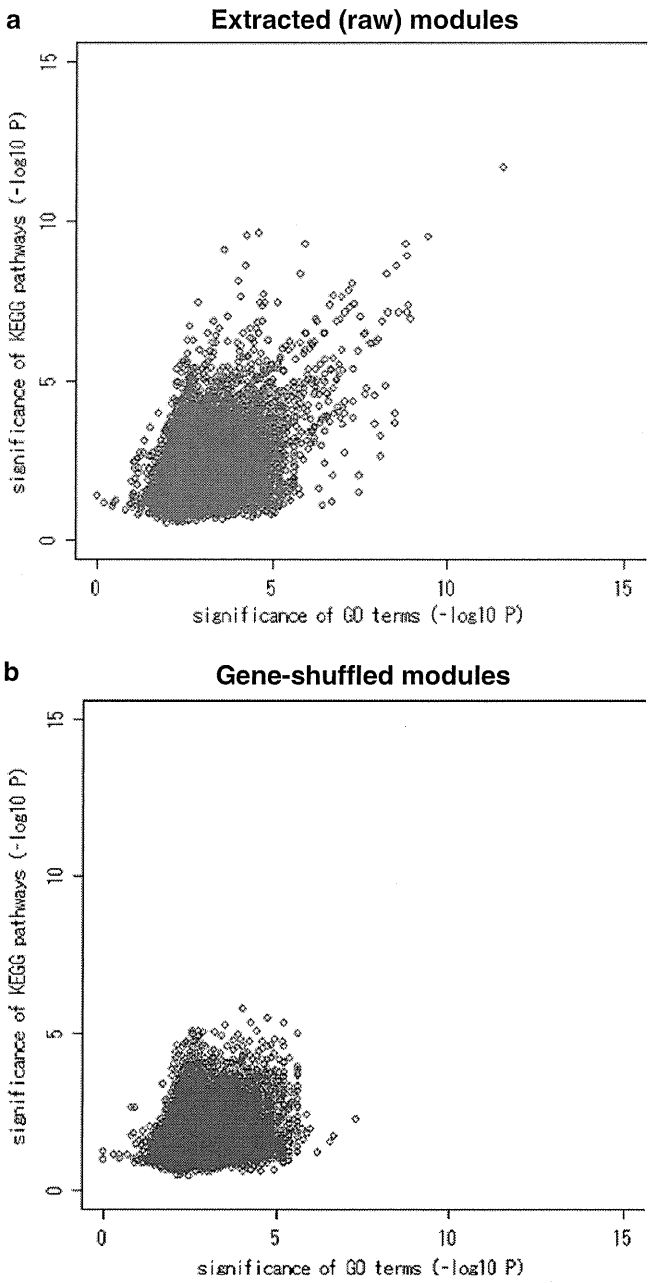


Fig. 2. Analysis of significant gene modules by GO and KEGG function groups. Each gene module (with 5% discretization threshold and 0.4 noise ratio) is evaluated with GO terms and KEGG pathway maps based on hypergeometric distribution statistics and plotted by the most significant *p* values. A total of 92,100 obtained gene modules are plotted in (**a**), and the same gene modules in which their gene ids are shuffled are plotted in (**b**). *p* values are dramatically increased due to randomness in (**b**). There is a significant correlation between *p* values of GO and KEGG in only (**a**). We arbitrarily choose 1e – 5.5 and 1e – 4 as the threshold for GO and KEGG, respectively.

1. Download SAMURAI program from the above site. Select SAMURAI-P program. Uncompress and untar the frozen file on your Linux machine. Go to the expanded directory and type "make compile" to compile all the programs in the system.

2. Transform your gene expression dataset into the "CellMontage" format *(14)* where a gene expression profile consists of a single description line, starting with ">," followed by one or more data lines. The data consist of elements separated by a space or a new line. Each element consists of a UniGene identifier and an expression value separated by a colon. See **Fig. 3a** for example data.

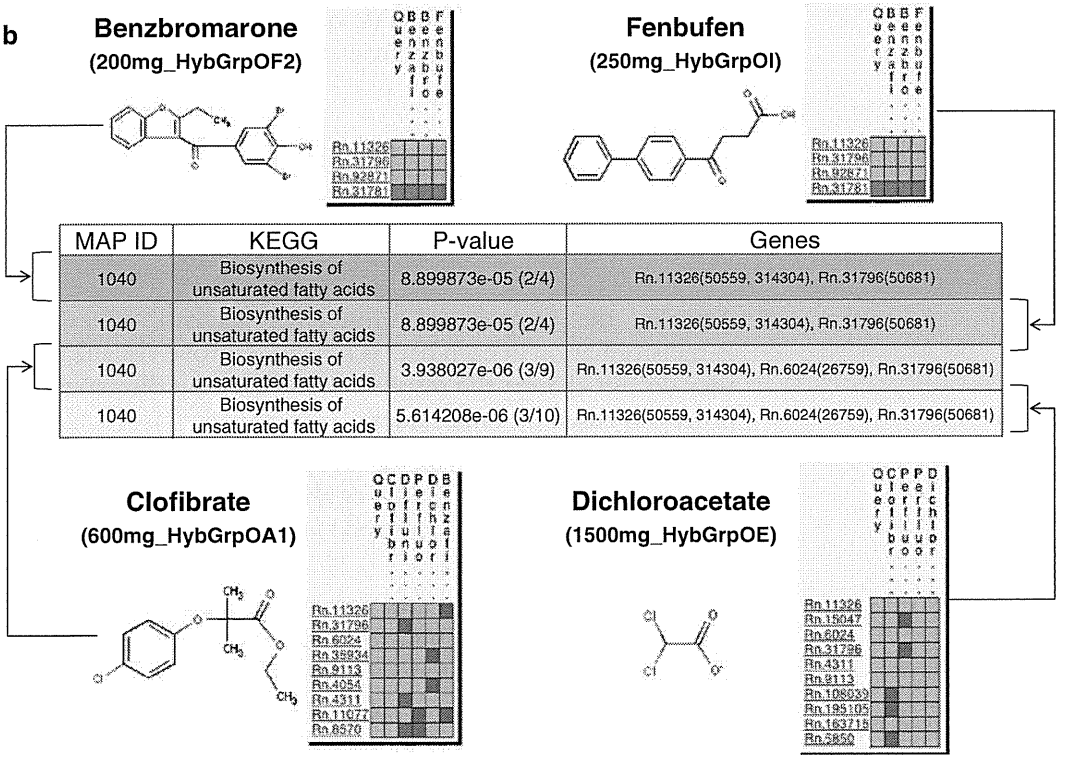
a

An example of partial profile in CellMontage format, where only three genes and their expression values are shown.

```
>Clofibrate_600mg_HybGrpOA2
Rn.98209:0.26226233    Rn.53257:0.550381825
Rn.94195:-0.285033381
```

A gene expression profile consists of a single description line, starting with ">", followed by one or more data lines. The data consist of elements, separated by a space or a new line. Each element consists of a UniGene identifier and an expression value separated by a colon.

b



MAP ID	KEGG	P-value	Genes
1040	Biosynthesis of unsaturated fatty acids	8.899873e-05 (2/4)	Rn.11326(50559, 314304), Rn.31796(50681)
1040	Biosynthesis of unsaturated fatty acids	8.899873e-05 (2/4)	Rn.11326(50559, 314304), Rn.31796(50681)
1040	Biosynthesis of unsaturated fatty acids	3.938027e-06 (3/9)	Rn.11326(50559, 314304), Rn.6024(26759), Rn.31796(50681)
1040	Biosynthesis of unsaturated fatty acids	5.614208e-06 (3/10)	Rn.11326(50559, 314304), Rn.6024(26759), Rn.31796(50681)

Fig. 3. Examples of input and output of SAMURAI-GUI system. An example of analysis with queries tnts (**a**) the CellMontage format as input and (**b**) graphical output of modules that share common biological function. Four example modules are searched by Benzbromarone, Fenbufen, Clofibrate, and Dichloroacetate. KEGG pathway functions for each module with the first and the second most significant *p* values are only shown.

3. Format data as described in 2.2 "Formatting Database by Discretization." Use the "formatdb" command. To discretize your data with the thresholds, as exemplified in 2.2, type "%formatdb dataset_file 0.03 (or 0.05)." This command takes a while and creates both the discretized data "dataset_file.db" and its gene index file "dataset_file.idx."
4. Run the LCM algorithm to extract core modules and merge them by typing: "% java –Xmx2000m xSamurai –M –i dataset_file.idx –d dataset_file.db –q query_file –n noise_threshold –s minimum_module –r result_dir."

   The query_file must also be written in the CellMontage format. The "noise_threshold" must be in the range of [0,1]. The "minimum_module" parameter sets the minimum size of gene modules to output. The "result_dir" parameter indicates the directory to write the results of final gene modules.
5. To perform module evaluations with GO and KEGG, execute the command: "%EA = GO_KEGG_test GO_KEGG_test/assignKEGG.pl dataset_file.idx P result_dir/*."

   The "GO_KEGG_test" is the directory for GO/KEGG evaluation package (*see* **Note** 7).
6. The GUI-based web version for multi-CPU calculation and module visualization in color is also available from a commercial site. To view free test results, visit: http://samurai.cbrc.jp/ and try the "Module search from a large-scale database" Web page (*again, see* Note 6).

3. Notes

1. Actually, the purpose of subtracting median values and Z-transformation in each data is only to improve visualization; they do not change the results as the following discretization process is based on rank values.
2. With an average query that represents only 10% of its discrete values are active (up- or down-regulated) genes, it is estimated that the database will be reduced to ($0.05 \times 0.05 \times 2 =$) 0.5% of its original size.
3. LCM program usually takes several seconds to finish, but sometimes takes several minutes. A faster program is currently available from the developers' Web site.
4. Many versions of merging processes are available. For example, implementing merge after finishing all the "checks" of module pairs is an option. Take the best approach depending on the situation (computational resource, purpose, etc.).

5. The conversion of genes from UniGene into EntrezGenes generates often more than single (one-to-one) correspondences.
6. The full license and web-enhanced server versions are available from HPC Solutions Inc. (http://www.hpc-sol.co.jp/).
7. Before running GO/KEGG evaluation script, do not forget to download necessary data files, including gene2unigene, gene2go.gz, and KEGG pathway maps. To do this, add your species code to "getKEGGmaps.pl" script in the "GO_KEGG_test" directory and then type "make compile" at the top directory.

Acknowledgments

We would like to thank Dr. Takeaki Uno at National Institute of Informatics for advice and kindly providing the LCM program for free use.

References

1. Barrett, T., Suzek, T.O., Troup, D.B., Wilhite, S.E., Ngau, W.C., Ledoux, P., Rudnev, D., Lash, A.E., Fujibuchi, W., and Edgar, R. (2005) NCBI GEO: mining millions of expression profiles – database and tools. *Nucleic Acids Res.* **33** (Database issue), D562–D566.
2. Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Lara, G.G., Oezcimen, A., Rocca-Serra, P., and Sansone, S.A. (2003) ArrayExpress – a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res.* **31**, 68–71.
3. Cheng, Y., and Church, G. (2000) Biclustering of expression data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 93–103.
4. Tanay, A., Sharan, R., and Shamir, R. (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics* **18**, S136–S144.
5. Ben-Dor, A., Chor, B., Karp, R., and Yakhini, Z. (2002) Discovering local structure in gene expression data: the order-preserving submatrix problem. *Proceedings of the 6th Annual International Conference on Computational Biology, ACM Press, New York, NY, USA*, 49–57.
6. Murali, T.M., and Kasif, S. (2003) Extracting conserved gene expression motifs from gene expression data. *Pac. Symp. Biocomput.* **8**, 77–88.
7. Ihmels, J., Bergmann, S., and Brkai, N. (2004) Defining transcription modules using large-scale gene expression data. *Bioinformatics* **20**, 1993–2003.
8. Wu, C.J., and Kasif, S. (2005) GEMS: a web server for biclustering analysis of expression data. *Nucleic Acids Res.* **33**, W596–W599.
9. Prelic, A. et al. (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* **22**, 1122–1129.
10. Okada, Y., and Fujibuchi, W. (2007) Mining a Large-scale Microarray Database for Similar Gene Expression Modules to Find Distant Relationships between Down Syndrome and Huntington's Disease. *Proceedings of Critical Assessment of Microarray Data Analysis 07, Valencia, Spain.*
11. http://camda.bioinfo.cipf.es/camda07/
12. Uno, T., Asai, T., Uchida, Y., and Arimura, H, (2004) An efficient algorithm for enumerating closed patterns in transaction databases, *Lecture Notes in Artificial Intelligence* **3245**, 16–31.
13. Uno, T., Kiyomi, M., and Arimura, H, (2002) LCM ver.2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets, *IEEE ICDM'04 Workshop FIMI'04* **126**.
14. Fujibuchi, W., Kiseleva, L., Taniguchi, T., Harada, H. and Horton, P. (2007) CellMontage: Similar Expression Profile Search Server, *Bioinformatics* **23**, 3103–3104.