Nucleic Acids Research, 2004, Vol. 32, Web Server issue ***W465–W470***
DOI: 10.1093/nar/gkh470

Expression Profiler: next generation—an online platform for analysis of microarray data

Misha Kapushesky[1,*], Patrick Kemmeren[1,2], Aedín C. Culhane[3], Steffen Durinck[4], Jan Ihmels[5], Christine Körner[6], Meelis Kull[7,8], Aurora Torrente[1], Ugis Sarkans[1], Jaak Vilo[1,7,8,9] and Alvis Brazma[1]

[1]European Bioinformatics Institute, United Kingdom, [2]Genomics Laboratory, Division of Biomedical Genetics, UMC Utrecht, the Netherlands, [3]Conway Institute, University College Dublin, Ireland, [4]ESAT, KULeuven, Belgium, [5]Weizmann Institute of Science, Israel, [6]University of Leipzig, Germany, [7]Egeen International, Estonia, [8]Department of Computer Science, University of Tartu, Estonia and [9]Estonian Biocenter, Estonia

Received February 14, 2004; Revised and Accepted April 29, 2004

ABSTRACT

Expression Profiler (EP, http://www.ebi.ac.uk/expressionprofiler) is a web-based platform for microarray gene expression and other functional genomics-related data analysis. The new architecture, Expression Profiler: next generation (EP:NG), modularizes the original design and allows individual analysis-task-related components to be developed by different groups and yet still seamlessly to work together and share the same user interface look and feel. Data analysis components for gene expression data preprocessing, missing value imputation, filtering, clustering methods, visualization, significant gene finding, between group analysis and other statistical components are available from the EBI (European Bioinformatics Institute) web site. The web-based design of Expression Profiler supports data sharing and collaborative analysis in a secure environment. Developed tools are integrated with the microarray gene expression database ArrayExpress and form the exploratory analytical front-end to those data. EP:NG is an open-source project, encouraging broad distribution and further extensions from the scientific community.

INTRODUCTION

The complete analysis of microarray gene expression data consists of many steps, starting with image processing, annotation, data normalization and transformation, gene subselection and filtering (1). The analysis that follows can be one of many available techniques, e.g. clustering or class prediction. While there exist applications that integrate some of these analyses within a single tool (some stand-alone and some web-based), the need for an extensible framework with a unified interface to disparate analysis components persists. Expression Profiler: next generation (EP:NG) provides such an online environment, in which diverse components are brought together under a unified look and feel, so that the user is unaware that these algorithms are developed, maintained and run by different groups. In fact, these data analysis components may be distributed across different systems and physical locations. We have also endeavored to make it easy for algorithm developers to contribute their methods to EP:NG, thus making it a suitable platform to become a public compendium of data analysis methods.

In a typical workflow (Figure 1), the user might either upload gene expression data from an external source via the Data Upload component, or retrieve data from the ArrayExpress public repository at the EBI (2). The Data Selection component provides a basic statistical overview of the dataset, which can be used to guide the user in subselecting genes relevant for further analysis. The Data Transformation component can impute missing values in the chosen data subset, and perform other data transformations. Following these optional preprocessing steps, data can be subjected to one or more analyses. The user can explore the overall structure of the data via one of the clustering methods available in the Hierarchical and K-groups Clustering components, and the best number and quality of clusters within the data can be evaluated using a novel algorithm, Clustering Comparison. Alternatively, the user can subject data to a semisupervised method aimed at studying correspondences between groups of samples and genes in the Between Group Analysis component. Users interested in studying a specific gene or a group of genes can use the Similarity Search or the Signature Algorithm components to investigate the structural organization of the data. Thus, EP:NG provides

*To whom correspondence should be addressed. Tel: +44 1223 494 647; Fax: +44 1223 494 468; Email: ostolop@ebi.ac.uk

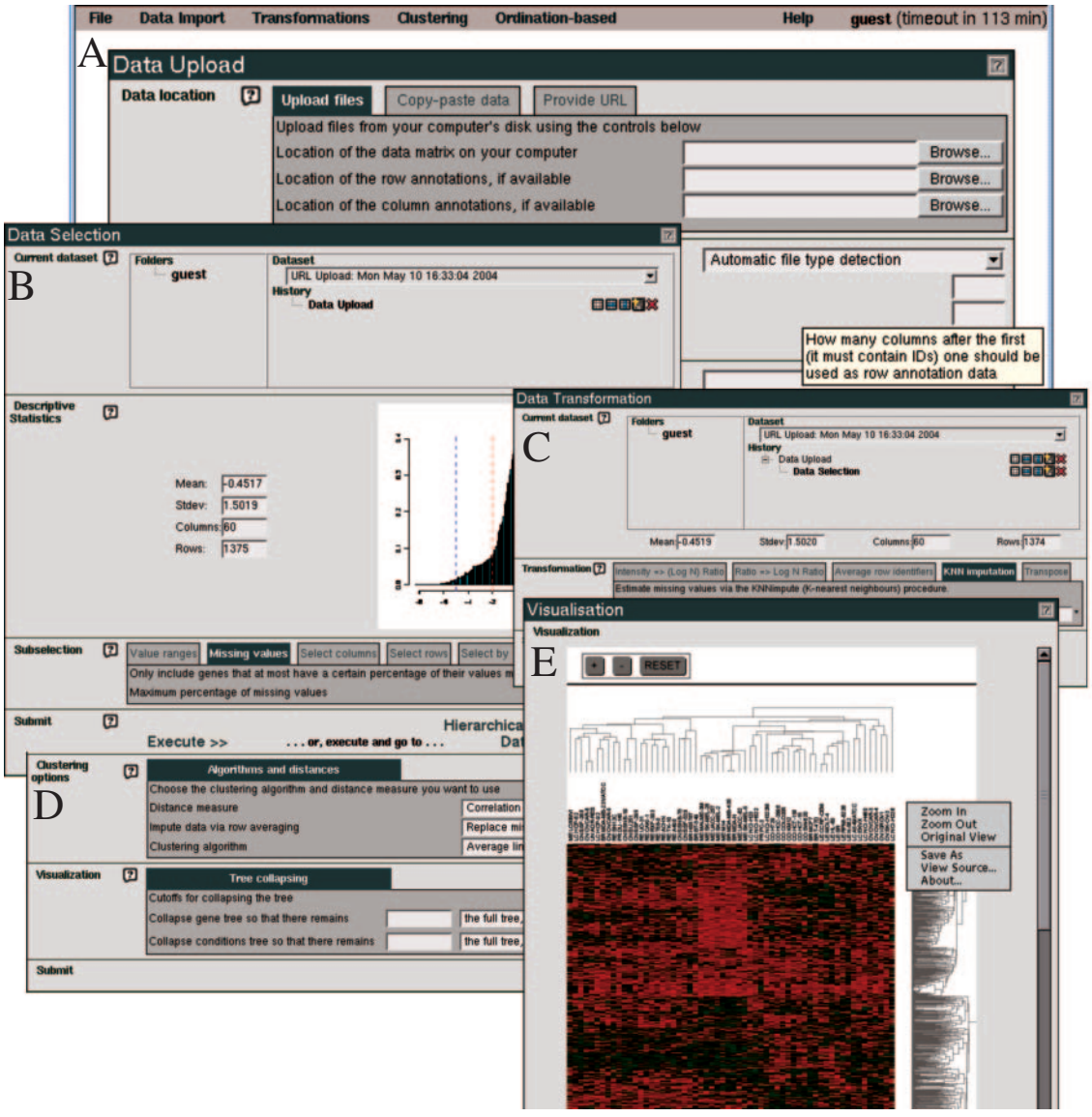

Figure 1. A typical analysis workflow. (**A**) After the user logs in, the data file is uploaded via the Data Upload component. (**B**) A subselection is made via the Data Selection component (the distribution histogram indicates the spread of the data and helps choose the range criteria). (**C**) Data Transformation is used to impute the missing values in the resulting data subset; clicking on the dataset name will return the new matrix with estimated missing values. (**D**) Hierarchical Clustering with tree collapsing is used to get the first overview of the data structure; (**E**) the hierarchical tree is displayed as an SVG image, allowing the user to zoom in on the tree in detail.

a useful tool for exploratory data analysis. EP:NG also integrates the components of the earlier version of EP (3), which remains available at http://ep.ebi.ac.uk and includes several additional online tools. EP:NG offers a new architecture and interface, using the foundations of the original Expression Profiler, keeping the original fast algorithm implementations and building on the initial idea of a simple to use, flexible online data analysis toolbox.

EP:NG PLATFORM ARCHITECTURE

The EP:NG platform gives a unified style to all the constituent components, regardless of the origin of their development, their physical location and their manner of execution. EP:NG supports quick and easy integration of third-party tools and algorithms: for instance, several components are implemented via integration with the statistical package R (4). The software is based on a relational database that stores user account information as well as metadata about the datasets being analyzed. Analysis results and the datasets themselves are stored on the file system in a format that facilitates fast operations on the data. All core algorithms are written in C/C++ or R, while the interfacing between components is done through XML/XSLT transformations. Perl serves as glue between the various parts of the EP:NG platform.

The EP:NG infrastructure is built to support (i) data sharing between groups of collaborators independent of location and format, (ii) execution of algorithms in a pipeline fashion and (iii) the integration of a constantly expanding collection of methods of data analysis. A simple web-service-type interface is exposed for programmatic access to individual components

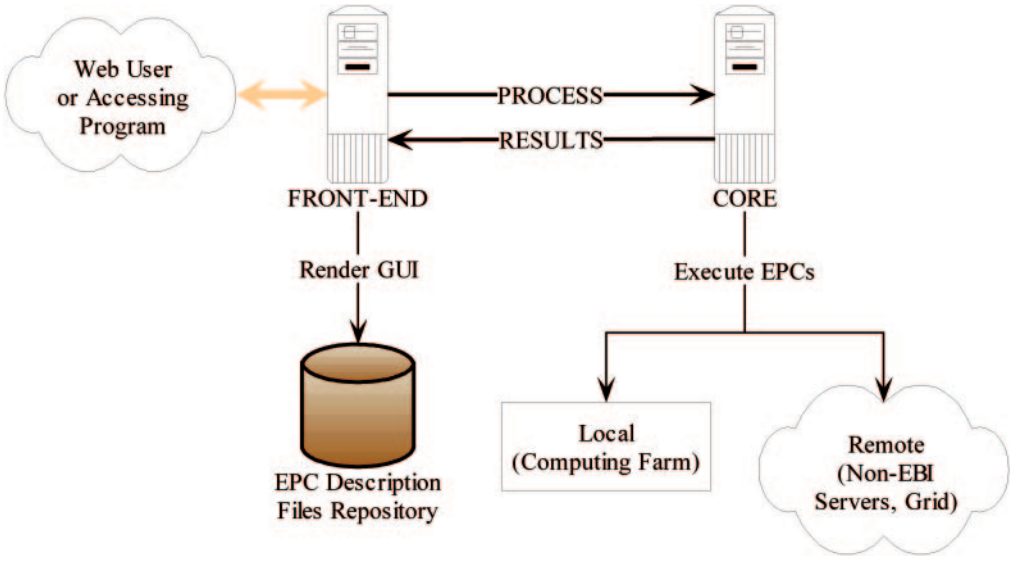

Figure 2. EP:NG system overview. All access is realized through the FRONT-END, which uses the Expression Profiler Component (EPC) description files to present a graphical user interface (GUI), and then sends the user's request to the CORE for either local execution on the Linux server farm at the EBI or remote processing on external (non-EBI) computer servers.

of EP:NG as well as for complex data processing queries. Figure 2 depicts a high-level overview of the entire system. EP:NG is built on the concept of atomic components: each functional software subunit is represented by a separate distinct component, whose inputs and outputs are described separately in a simple format. The system uses these descriptions to present an interface to the user, and runs the component on the back-end computing servers, or sends the data off for remote analysis. This means that in order to integrate a novel method into the EP:NG environment its author needs only to compose an appropriate XML description file and send it to the site maintainers—the method will then appear seamlessly among the other tools in EP:NG.

EXPRESSION PROFILER COMPONENTS (EPCs)

An EP:NG user can log in as guest or can establish a permanent password-protected account. We would recommend the latter as it provides each user with the ability to store and organize data and prior analysis results into nested folders, for further work over multiple sessions. EP:NG user accounts can also belong to one or more user groups, which are managed by an account with group administrator privileges. This facility allows groups of researches to share their data and analysis strategies.

Data Upload

Users can either upload their own data or select a published dataset from the ArrayExpress database. To retrieve data from ArrayExpress, one can explore the database via its online interface and subsequently export an expression matrix by selecting the desired samples and measurements (Assays and Quantitation Types) and then send this to EP:NG.

The Data Upload component can accept data in a number of formats including basic delimited files such as those exported by Microsoft Excel. Uploaded expression datasets must be represented as matrices, with rows and columns corresponding to genes and conditions, respectively. In addition to loading the data matrix, the user can provide various metadata for the dataset. Metadata could include a description of the dataset, studied organism, or row and column annotations, which can be either free-text annotations or identifiers from publicly available ontologies, e.g. GeneOntology (5).

Data Selection

Data Selection presents a brief statistical overview of the data (data distribution histogram, mean, standard deviation) and allows the user to select genes and conditions that have particular expression values. For example, one can select only those genes (data rows) that are outside ± 2 standard deviations of the mean in at least 60% of experiments and which contain no more than 80% missing values. Further subselections can be made at any point in the analysis.

Data Transformation

Either before or after subselecting data, various transformation procedures can be applied. These include *K*-nearest neighbor imputation (6) to fill in missing values (this can be a lengthy procedure especially on large datasets), LOWESS normalization (7) (an integrated third-party component) and conversion of absolute intensity values from a two-channel experiment to log ratios. From here one can proceed to further subselections or apply a suitable analysis component.

Similarity Search

The Similarity Search provides a means of selecting groups of genes related to given ones. The user specifies one or several genes, chooses a similarity measure and receives those genes most closely co-expressed with the selected genes within the dataset. A wide variety of distance measures is available, including the Euclidean metric, Pearson correlation, Manhattan distance, Spearman's ranking and chord distance. From the resulting screen the user can choose a subset of these genes and/or conditions, and then use the Data Selection component

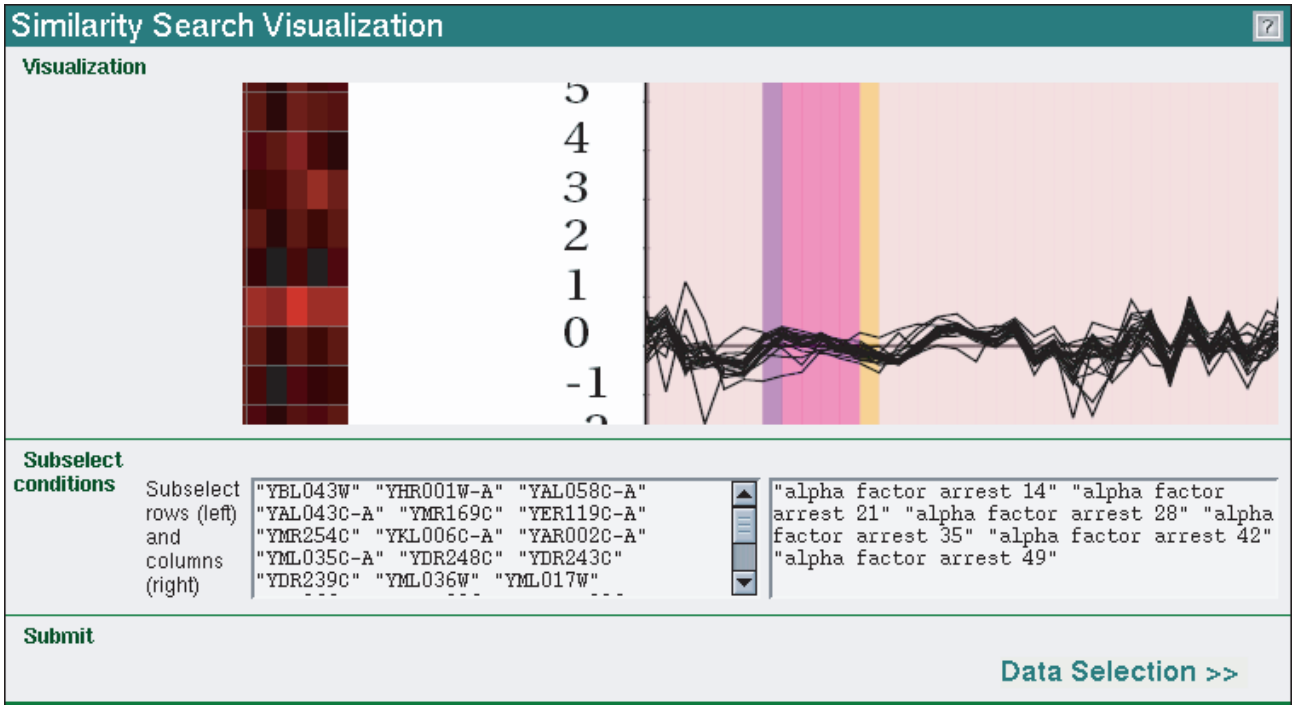

Figure 3. Similarity Search Visualization component. The user can select with the mouse a subset of conditions where the expression line-plot shows correlation by visual inspection. The chosen genes and conditions can be sent to other components for further analysis (e.g. to Data Selection).

to save this subset as a separate dataset for further analysis. Similarity Search Visualization allows a subset of conditions to be subselected under which the displayed genes correlate best (Figure 3).

Hierarchical and *K*-groups Clustering

EP:NG provides both hierarchical and partitioning-based clustering methods. As stated above, there are many distance measures available. The clustering algorithms are implemented in C, and results are visualized as publication-quality vector-based SVGs or in a raster format, PNG. Data can be hierarchically clustered on both experiments and conditions simultaneously and one can interactively zoom in on interesting subtrees. The hierarchical clustering tree produced for large datasets can be difficult to comprehend all at once. An attractive feature of EP:NG is that it provides a facility to display a quick high-level overview of the clustered data. Using the 'Tree Collapsing' option, one can display a specified percentage of major branches, with the other ones collapsed into single nodes.

There are two *K*-groups clustering algorithms in EP:NG: *K*-means and *K*-medoids. The latter is a variant of another well-known approach to partitioning the data into a specified number of clusters. It differs in that it uses existing objects from the dataset as cluster centers in its calculations. This allows the use of a distance matrix derived from any measure; hence, this component can be applied to more diverse data types, such as sequences.

Clustering Comparison

A problem that arises naturally with all partitioning clustering methods is to find the appropriate *K* (number of clusters). The Clustering Comparison component implements an algorithm that takes two *K*-groups clustering results and matches the clusters by membership. By examining the output the user can evaluate the optimal (according to some criteria) number of clusters in the dataset.

Signature Algorithm

Signature Algorithm is the R implementation of an algorithm described in (8). It identifies a co-expressed subset in a user-submitted set of genes, removes unrelated genes from the input and identifies additional genes in the same dataset that follow a similar pattern of expression. Co-expression is identified with respect to a subset of conditions, which is also provided as the output of the algorithm. It is a fast algorithm useful for exploring the modular structure of expression data matrices.

Between Group Analysis and Ordination

Standard multivariate analysis methods, such as principal component analysis (PCA) and correspondence analysis (COA), are provided in the Ordination component. These methods are frequently used to search for underlying structures in datasets. Between Group Analysis (BGA) presents a multiple discriminant approach that can be used with expression data matrices of any dimensionality (9). BGA is carried out by ordinating groups (sets of grouped microarray samples) and then projecting the individual sample locations on the resulting axes. It is used in the framework of conventional ordination techniques such as PCA or COA and, as such, allows for great flexibility with regard to the assumptions that one makes in carrying out the analysis. When combined with COA it is especially powerful as it allows one to examine in detail the correspondences between the grouped samples and

those genes which most facilitate the discrimination of these groupings. This is a semisupervised method, so it is important to test its results using a test dataset, or using resampling accuracy analysis methods. The Ordination and BGA EPCs are implemented using the R multivariate data analysis package ADE-4 (10).

Pipelines and workflows

It is important not only to be able to find and execute analytical procedures, but to do this in a logical, user-defined sequence of steps. This naturally leads to the concept of analysis pipelines. EP:NG was designed to enable the user to combine components into sequences. There are two major ways to do this. First, via the interface: each component provides annotated links to other EPCs that logically follow (e.g. Data Selection links to Data Transformation and to Hierarchical Clustering, etc.). Second, programmatically: a program can send a simple XML query to EP:NG indicating which components to launch and in what sequence. The final result will be shown to the user. Whenever a user or a program runs an EP:NG component, the parameters and the sequence of steps taken to obtain the results are stored in the internal database. This allows one to define an analysis workflow, a process that can later be applied repeatedly with the same parameters to other datasets.

HARDWARE AND SOFTWARE REQUIREMENTS

EP:NG may be run in a distributed fashion, where the GUI and the CORE servers reside on separate machines (and the CORE interfaces to a queuing system, as is the setup at the EBI), or the entire system may run on a single server. This setup involves a Linux system as the front-end server for the GUI, and an alpha station at the back-end, both running Apache web servers. EP:NG uses the LSF queuing system, but can also run in a stand-alone mode or integrate with PBS. EP:NG is an open-source project and the related SourceForge site (http://ep-sf.sourceforge.net) contains further information, including information on how to develop new components and how to obtain and install EP:NG locally.

EP:NG USAGE

EP:NG has two major purposes—to provide tools for exploratory analysis of microarray data (including the data in ArrayExpress) and to support an infrastructure for rapid deployment of specific methods. With this collection of tools a novice user can easily try out several different approaches, compare the results and select those methods that make the most sense. The entire EP:NG system can also be installed locally and be used within one lab or simply as a local data analysis program. Expression Profiler and EP:NG are extensively used by biology labs, research groups and as bioinformatics teaching software at universities.

ArrayExpress data export into EP:NG

In order to analyze data that is stored in ArrayExpress with Expression Profiler, gene expression data matrices need to be exported from the repository into EP:NG. The user can start by querying ArrayExpress (http://www.ebi.ac.uk/arrayexpress) for a specific experiment by the published accession number or through a more general query on combinations of other fields such as experiment type or laboratory. ArrayExpress provides the facility to retrieve the stored data by selecting a combination of (i) BioAssays (hybridizations), (ii) Quantitation Types (data types) and (iii) Array Annotations. These choices determine the composition of the resulting data matrix: for every selected hybridization columns of chosen quantitation types will be exported. Every row of the matrix will be annotated with selected annotations from the corresponding array design. Finally, either the generated gene expression matrix can be downloaded on to the user's computer for later analysis in the user's own account in EP:NG (or other tools) or it can be exported directly into EP:NG for immediate analysis. ArrayExpress help pages (http://www.ebi.ac.uk/arrayexpress/Help/) document these steps in further detail.

FUTURE DEVELOPMENT

The next EP:NG version will include a separate Data Normalization component that will integrate several publicly available normalization algorithms, an improved interactive visualization interface and several advanced data analysis EPCs. We are also implementing support for data input in the MAGE-ML format (11) so that the system can obtain detailed information about a dataset's origin and related array designs, which may make it possible to suggest data analysis strategies to the user. There are ongoing collaborations with other groups to develop a more advanced web-services interface, as well as to provide customized support for Affymetrix data through integration with Bioconductor (4).

SUPPLEMENTARY DATA

The Expression Profiler SourceForge website, http://ep-sf.sourceforge.net, contains various documents regarding the installation, use and development of the EP:NG framework and components. ArrayExpress help pages are useful for learning how to export and analyze data in Expression Profiler: http://www.ebi.ac.uk/arrayexpress/Help.

ACKNOWLEDGEMENTS

We wish to thank Luke Jeffery for helping with the testing. The authors would like to acknowledge the TEMBLOR grant from the EC and the Wellcome Trust for support. J.Vilo and M. Kull acknowledge Estonian Science Foundation grants nos. 5724 and 5722.

REFERENCES

1. Quackenbush,J. (2001) Computational analysis of microarray data. *Nature Rev. Genet.*, **2**, 418–427.
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. (2003) ArrayExpress—a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res.*, **31**, 68–71.

3. Vilo,J., Kapushesky,M., Kemmeren,P., Sarkans,U. and Brazma,A. (2003) Expression Profiler. In Parmigiani,G., Garret,E.S., Irizarry,R. and Zeger,S.L. (eds), *The Analysis of Gene Expression Data: Methods and Software*. Springer Verlag, New York.
4. Ihaka,R. and Gentleman,R. (1996) R: a language for data analysis and graphics. *J. Comput. Graph. Stat.*, **5**, 299–314.
5. Gene Ontology Consortium (2001) Creating the gene ontology resource: design and implementation. *Genome Res.*, **11**, 1425–1433.
6. Troyanskaya,O., Cantor,M., Sherlock,G., Brown,P., Hastie,T., Tibshirani,R., Botstein,D. and Altman,R.B. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.
7. Yang,Y.H., Dudoit,S., Luu,P., Lin,D.M., Peng,V., Ngai,J. and Speed,T.P. (2002) Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.*, **30**, e15.
8. Ihmels,J., Friedlander,G., Bergmann,S., Sarig,O., Ziv,Y. and Barkai,N. (2002) Revealing modular organization in the yeast transcriptional network. *Nature Genet.*, **31**, 370–377.
9. Culhane,A.C., Perrière,G., Considine,E.C., Cotter,T.G. and Higgins,D.G. (2002) Between-group analysis of microarray data. *Bioinformatics*, **18**, 1600–1608.
10. Thioulouse,J., Chessel,D., Doledec,S. and Olivier,J.M. (1997) ADE-4: a multivariate analysis and graphical display software. *Stat. Comput.*, **7**, 75–83.
11. Spellman,P.T., Miller,M., Stewart,J., Troup,C., Sarkans,U., Chervitz,S., Bernhart,D., Sherlock,G., Ball,C., Lepage,M., Swiatek,M., Marks,W.L., Goncalves,J., Markel,S., Iordan,D., Shojatalab,M., Pizarro,A., White,J., Hubley,R., Deutsch,E., Senger,M., Aronow,B.J., Robinson,A., Bassett, D., Stoeckert,C.J.,Jr and Brazma,A. (2002) Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.*, **3**, RESEARCH0046.