Nucleic Acids Research, 2003, Vol. 31, No. 13 ***3461–3467***
DOI: 10.1093/nar/gkg591

GEPAS: a web-based resource for microarray gene expression data analysis

Javier Herrero, Fátima Al-Shahrour, Ramón Díaz-Uriarte, Álvaro Mateos, Juan M. Vaquerizas, Javier Santoyo and Joaquín Dopazo*

Bioinformatics Unit, Centro Nacional de Investigaciones Oncológicas (CNIO), c/Melchor Fernández Almagro 3, 28029, Madrid, Spain

Received February 15, 2003; Revised and Accepted April 3, 2003

ABSTRACT

We present a web-based pipeline for microarray gene expression profile analysis, GEPAS, which stands for Gene Expression Profile Analysis Suite (http://gepas.bioinfo.cnio.es). GEPAS is composed of different interconnected modules which include tools for data pre-processing, two-conditions comparison, unsupervised and supervised clustering (which include some of the most popular methods as well as home made algorithms) and several tests for differential gene expression among different classes, continuous variables or survival analysis. A multiple purpose tool for data mining, based on Gene Ontology, is also linked to the tools, which constitutes a very convenient way of analysing clustering results. On-line tutorials are available from our main web server (http://bioinfo.cnio.es).

INTRODUCTION

Over the last few years technology in the field of genomics has progressed significantly, resulting in vast amounts of biological data available. In particular, microarray technologies (1,2) are generating huge amounts of data, leading to the creation of new tools for data management (3) and analysis (see, for example, a compilation in http://ihome.cuhk.edu.hk/~b400559/arraysoft.html). Data management of microarrays firstly requires the implementation of local databases as well as public repositories, which demand a common data exchange format. Following this, the Microarray Gene Expression Data (MGED; http://www.mged.org) Society, whose aim is to facilitate the sharing of microarray data generated by functional genomics and proteomics, establishing standards of annotations and exchange formats. Less effort, however, has been made in standardising the formats on the side of the analysis of this data. Different tools implementing distinct algorithms are available for diverse platforms. This often makes the application of more than one algorithm to the data cumbersome. Web-based software would avoid this problem, but to date, apart from some exceptions [e.g. Expression Profiler (4)], most of the web tools are spread across different servers internationally and employ different file formats. Here we present an integrated web-based pipeline for the analysis of gene expression pattern where the most popular tools can be used in an integrated interface. This allows a transparent use once the data has been uploaded. The way in which the tools are connected guide the user by suggesting all the available possibilities to continue with analysis.

THE PIPELINE OF MICROARRAY DATA ANALYSIS UNDER A WEB-BASED UNIFIED FRAMEWORK

After hybridisation with the control and/or the query labelled DNAs, the array is scanned. Images corresponding to intensity of the hybridisation process are obtained. Depending on the technology used [cDNA microarrays (1) or Affymetrics oligonucleotide arrays (2)] their processing is different but, essentially, the microarrays must be analysed to identify and quantify the spots corresponding to the probes. Usually, commercial microarray scanner manufacturers provide their own solutions for image processing. In addition, both public-domain and commercial solutions are available. Comparison of the results from different hybridisations requires a normalisation process. The distinct efficiencies in the labelling process and in the detection of the fluorescence in both channels, as well as differences in the initial amount of mRNA in the samples, not to mention problems derived from the manipulation of the samples, cause systematic biases in the measurements. Normalisation procedures are also often implemented in many image analysis programmes. Once the data has been normalised, it is then ready for analysis. Since GEPAS works with gene expression patterns, it is still necessary to merge data coming from different conditions, such as time-courses, cohorts of patients or a series of different drug dosages. The matrix of gene expression values for the different experimental conditions constitute the starting point of the pipeline presented here. Figure 1 shows how the different modules of GEPAS are interconnected and exchange data amongst them. Once the expression pattern data is introduced within the system, all the modules available can be used to analyse them. The pre-processor module acts as a hub

*To whom correspondence should be addressed. Tel: +34 9122 46919; Fax: +34 9122 46972; Email: jdopazo@cnio.es

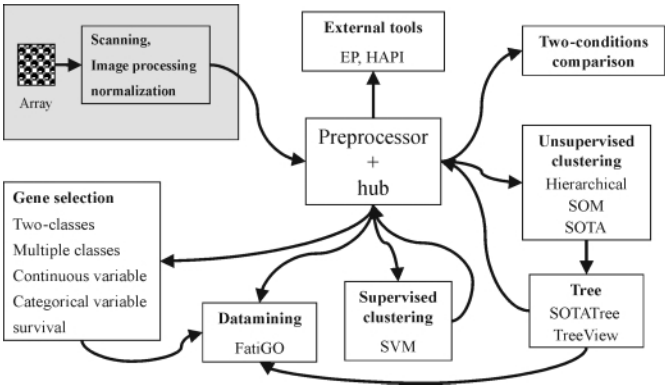


Figure 1. The pipeline of microarray data analysis. After the operations of image processing and data normalisation are performed (grey box on top left), the data enters the pipeline through the preprocessor. Then, depending on the type of analysis the user needs to perform, these can be sent to different modules that implement different tools.

for distributing the information. Obviously, the modules can be used independently without the necessity of accessing them through the pre-processor.

Web-based tools guarantee real cross-platform capabilities. Client-server architecture provided by web tools makes resources available to remote users without the hardware support for heavy calculations that are made on the server side. Beyond these advantages, however, the efficiency of a modular package largely lies on the degree of integration of the different data analysis tools. When users can walk across a complete pipeline of data analysis in a transparent way, without the necessity of performing any reformatting operation or even without executing a simple cut-and-paste, it is because the tools are truly integrated.

File formats and programming details

Gene expression pattern datasets are huge tables with thousands of rows corresponding to the genes or clones present in the DNA array and several columns, one for each experimental condition measured. Consistent with this, the row data file format is a tab-delimited table with a first column containing all the identifiers for the genes (or clones) and as many columns as experiments containing their expression values. Additional information can be added as commentaries in lines beginning with a hash (#).

Since the tools presented here are oriented to the analysis of sets of arrays, instead of the analysis of individual arrays, MAGE-ML (http://xml.coverpages.org/mageML.html) has not been used given that it is more oriented to the description of individual arrays.

SOM and SOM-Tree are based on the original SOM_PACK (ftp://cochlea.hut.fi/pub/som_pak) and use its native format; nevertheless, the system is able to automatically translate the standard expression pattern files into files suitable for SOM_PACK without the necessity of the user's intervention. Programmes that deal with classes require an additional file with its description. The description is quite simple and consists on an enumeration of the class identifiers corresponding to each column. In the case of continuous variables (e.g. the concentration of a metabolite or time) there is a number per column.

Tree descriptions used by GEPAS are compliant with the widely used standard Newick tree format (http://evolution.genetics.washington.edu/phylip/newicktree.html). Gnuplot (http://www.gnuplot.info/) is behind the histograms calculated by the Preprocessor and the plots made by PlotCorr. All the remaining graphics are PNG images made with the GD library (http://www.boutell.com/gd/). C code from R (http://www.r-project.org/) has been used for some of the statistical tests implemented in the modules. The web interfaces are Perl CGIs, whereas the programmes are mainly binary executables written in 'C'. The overall system has been developed for working with an Apache web server.

Data preprocessing

The Preprocessor module is an interactive web tool for preprocessing microarray gene expression data (5). The most interesting and powerful feature of this module is the data preanalyser. It analyses the data, suggests the most appropriate transformations and proceeds with them after the user's agreement. The normal pre-processing steps include scale transformations, replicate handling, management of missing values (by means of different procedures), flat pattern filtering and pattern standardisation; and some of them are required before performing other pattern analysis. The processed data set is available in several file formats so that it can be sent to other pattern analysis tools over the web. The result can also be sent to the FatiGO for comparing genes that have been removed by the filters against remaining ones (see below). The preprocessor can also receive data from other modules and, in that way, plays the role of hub of data facilitating the application of successive analysis steps to the data (Fig. 1). The output of the preprocessor is also connected to the HAPI tool, a data mining tool based on MESH terms (6) and to the Expression Profiler (4).

Two conditions comparison

The PlotCorr module is an interface for graphical comparison between two experimental conditions. It calculates the correlation between both conditions and allows a quick visualisation of the genes that differ between the compared arrays. A threshold for masking genes with a similar expression in both conditions can be dynamically set. Over-, under-expressed genes and genes with similar expression levels (according to the threshold) are plotted in different colours and are listed in the resulting file. PlotCorr is connected to the FatiGO (see below).

Unsupervised clustering

Clustering is perhaps one of the most widely used tools for microarray data analysis. It produces groups of gene expression profiles based on a distance function. Clustering can be used to find groups of co-expressing genes (7), which are often functionally related or to obtain clusters of experimental conditions (8). Depending on the way in which the data is clustered, we can distinguish between hierarchical and non-hierarchical clustering. Hierarchical clustering allows detecting higher order relationships between clusters of profiles whereas the majority of non-hierarchical classification techniques work by allocating expression profiles to a predefined number of clusters, without any assumption on the inter-cluster relationships.

Different distance functions (based on Euclidean or correlation coefficients) which can produce alternative clustering of data are available. Depending on the method, different options are available. SOM and SOTA offer a variety of parameters for producing the clustering under different conditions and constraints.

Aggregative hierarchical method. Aggregative hierarchical clustering (9) is still one of the preferred choices for the analysis of gene expression patterns (mostly because it is available in many packages). It starts by joining the two closest gene expression profiles and substitutes them by an averaged profile. Then, it continues by recursively joining the next two closest profiles (or groups) until only one group remains. At the end, this procedure generates a representation of the data with the shape of a binary tree, in which the most similar patterns are clustered in a hierarchy of nested subsets.

The module Cluster allows for the clustering of genes, experimental conditions or both simultaneously based on different distance functions. The result can be sent to the TreeView module (Fig. 1) for visualisation of the resulting dendrogram.

SOM. It has been noted (10) that standard clustering methods suffer from a lack of robustness when applied to clustering thousands of gene expression profiles. As an alternative, some authors have proposed the use of neural networks (10,11). Unsupervised neural networks, such as Self-Organising Maps (SOM) (12) provide a more robust framework, appropriate for clustering large amounts of noisy data. Due to their properties, neural networks are suitable for the analysis of gene expression patterns. They can deal with real-world data sets containing noisy, ill-defined items with irrelevant variables and outliers, whose statistical distributions do not need to be parametric.

The SOM module implements a web interface over the SOM_PACK (http://www.cis.hut.fi/research/som_lvq_pak.shtml) and the result is graphically represented by the SomPlot module. The user can modify all the available parameters from the original package: the size of the network, the topology, the training parameters and the number of trials for testing several random initial maps. Each cluster can be viewed separately by clicking on it.

SOM-Tree. SOM-Tree is a combination of the two latest methods: in the first part, a SOM is trained with the data and in the second part an average linkage tree is built with the patterns of the SOM nodes. The result is drawn with SomTree (see below). The SOM-Tree is a method for exploratory data analysis (13).

SOTA. The Self-Organising Tree Algorithm (SOTA) is another neural network that has been used for expression pattern clustering (14). SOTA, unlike SOM, has a structure of a binary tree, which grows during the training of the network. This results in several important differences with respect to SOM. Firstly, the number of clusters does not need to be arbitrarily fixed from the beginning and is instead obtained by means of a randomisation test implemented in the programme (see 14 for details). The clustering obtained with SOTA (14) is proportional to the heterogeneity of the data instead of the number of items in each cluster (as in SOM). Thus, regardless of whether a given type of profile is abundant or not, all the similar items will remain grouped together in a single cluster and they will have no direct affect on the rest of the clustering. This is because SOTA mapping of the data in the tree is distribution preserving while SOM mapping on the grid is topology preserving (15). Furthermore, SOTA provides a highly precise classification of samples (16). The SOTArray module allows interactive definition of the parameters and training conditions. As SOM, SOTA offers the possibility of modifying many parameters that affect the convergence, the heterogeneity of the clusters obtained, etc. The default parameters, which produce optimal convergence for most of the conditions, were obtained by using a genetic algorithm. SOTA mirrors are accessible at the EBI (http://ep.ebi.ac.uk/EP/SOTA/) and at ALMA Bioinformatics (http://www.almabioinfo.com/sota)

Supervised methods

In many cases there is information available on the classes and the interest is in constructing a class predictor that, once trained, will be capable of assigning the proper membership to a new sample. Machine learning methods can be used for this purpose. Specifically, Support Vector Machines (SVM) (17) have been successfully applied to the classification of both genes (18) or experimental conditions (19). SVM can be considered as a binary classifier. It proceeds by constructing a hyperplane that separates class members (positive examples) from non-members (negative examples). Unfortunately, most real-world problems involve non-separable data for which there does not exist a hyperplane that successfully separates the positive from the negative examples. SVM provides the solution to the inseparability problem that involves the mapping of data into a higher-dimensional space and defines a separating hyperplane. Different kernels for the SVM can be selected by the user. The implementation of SVM in GEPAS has two parts: the learning part, in which the SVM learns from an example, gives an estimation of the learning rate by means of distinct cross-validation procedures and produces a model that can be saved for further use; and the classificatory, in which a series of samples can be introduced, and using a model previously stored, a prediction of class membership is done for them.

Differential gene expression in class-related or continuous variable related studies

One of the most interesting problems consists of finding genes differentially expressed among two or more conditions (e.g. different cancer types). Conceptually related to this is finding genes related to a given continuous variable (e.g. the level of a metabolite) or the case of survival, a particular case of a continuous variable. Nevertheless, finding the proper group of genes among the thousands present in the arrays is not an easy task.

The Pomelo tool has been designed to control the problem of multiple testing when searching for differentially expressed genes. Using microarray data we are testing for differential expression of a high number genes and we need to account for multiple testing. The problem of using the p-value from each test directly is that we are examining many null hypotheses (one null hypothesis—i.e. non differential expression for each gene). If we were to consider each of the tests with a p-value smaller than, say, 0.05, as significant, we would end up with an excessive number of differentially expressed genes. Despite this, not many authors are aware of this problem and few programmes consider multiple testing in their design. The need to account for multiple testing has been reviewed for the analysis of microarrays by Dudoit *et al*. (20). In Pomelo we have implemented four methods to account for multiple testing; two of them control the Family Wise Error Rate (21) and two others control de False Discovery Rate (22,23).

These methods can be applied to five different statistical tests: the t-test (to compare expression between two conditions), ANOVA (analysis of variance, to compare expression between two or more conditions), linear regression (to examine if the expression of genes is related to variation in a continuous variable, e.g. expression levels of a given metabolite), survival analysis [to examine if gene expression is related to patients' survival (24)] and Fisher's exact test for contingency tables (when both the dependent and independent variables are categorical).

Figure 2 shows the 100 genes that most differentially express amongst the two types of leukaemia studied (25) arranged in increasing order of the adjusted p-value. Worthy of note is that for the gene U50928, the 93rd in the rank, the adjusted p-value obtained is already 0.051459, while the unadjusted p value is still 3.99992e−05. To highlight the importance of using corrected p-values it should be noted that the number of genes that would be accepted as showing a significant differential expression between the classes is 1020 if the uncorrected p-value of 0.05 is used.

Graphical representation

There are several programmes that have been especially developed for producing graphical representations of the results of the modules described above: PaintPom, TreeView, SotaTree, SotaCluster and SomPlot. PaintPom is used for the representation of the Pomelo tool results and generates a colour-coded table of the most differentially expressed genes (Fig. 2). TreeView is used for drawing hierarchical clustering, SOTA trees and single clusters. It provides the classical plot with the tree and the colour-coded gene expression profiles. The representation can be changed in different ways. It allows for a compact representation or an expanded representation in which the names of the genes appear in the plot. Vertical and horizontal size is customisable and labels can be drawn in the plot. SotaTree was developed for SOTArray and now is also used for drawing the result of the SOM-Tree (Fig. 3). SotaCluster is used for displaying a cluster coming from SOM or SOTA (Fig. 3F). Finally, SomPlot draws the resulting SOMs in a 2D grid. Both PaintPom and SomPlot are integrated into their respective tools because of their specificity. The remaining ones are implemented as independent modules and offer numerous options for interactively customising the plots produced.

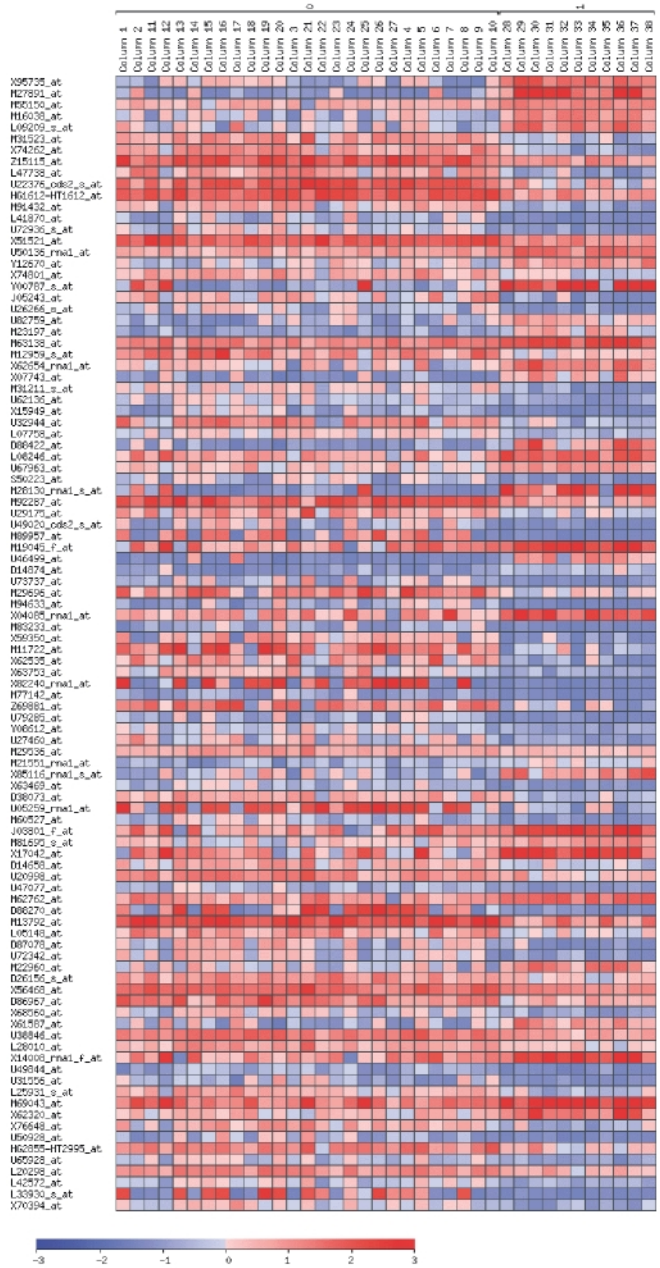
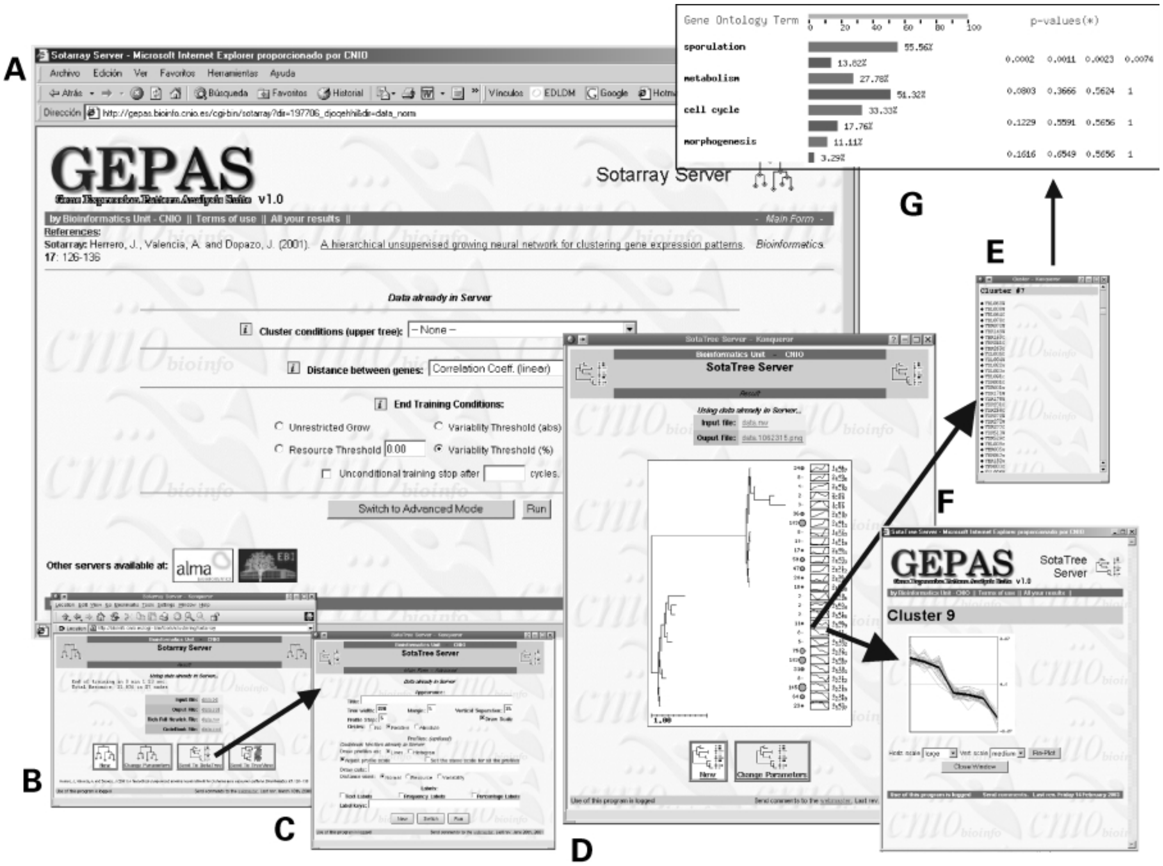


Figure 2. Representation provided by Pomelo of the 100 genes most differentially expressed among two different cancer types, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) labelled as 0 and 1, respectively in the top of the figure. The genes are arranged in increasing order of adjusted p-value. If an adjusted p-value of 0.05 is used, there are 92 genes that present significant differential expression among the classes. Thermal scale in the bottom represents fold of activation or repression in log2 scale. Data from (25).

Extract cluster

This module is probably the simplest part of the pipeline where it plays a key role in the exchange of data among programmes. It allows clustering programmes to send their results to other tools. The module extracts the genes belonging to the selected


A
B
C
D
E
F
G
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Historial Vínculos EDLDM Google
Sotarray Server
by Bioinformatics Unit - CNIO || Terms of use || All your results ||
- Main Form -
Other servers available at: alma EBI
SotaTree Server
Cluster 9
Cluster #7
New
Change Parameters
Re-Plot
Close Window

Sotarray Server - Microsoft Internet Explorer proporcionado por CNIO

Dirección http://gepas.bioinfo.cnio.es/cgi-bin/sotarray?dir=197706_djoqehhi&dir=data_norm

GEPAS v1.0

References:
Sotarray: Herrero, J., Valencia, A. and Dopazo, J. (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics 17: 126-136

Data already in Server

Cluster conditions (upper tree): -- None --

Distance between genes: Correlation Coeff. (linear)

End Training Conditions:

○ Unrestricted Grow ○ Variablity Threshold (abs)
○ Resource Threshold 0.00 ● Variablity Threshold (%)
□ Unconditional training stop after cycles.

Switch to Advanced Mode Run

Gene Ontology Term	0 20 40 60 80 100	p-values(*)			
sporulation	55.56%				
	13.82%	0.0002	0.0011	0.0023	0.0074
metabolism	27.78%				
	51.32%	0.0803	0.3666	0.5624	1
cell cycle	33.33%				
	17.76%	0.1229	0.5591	0.5656	1
morphogenesis	11.11%				
	3.29%	0.1616	0.6549	0.5656	1

Figure 3. Use of SOTAarray and SotaTree modules. Arrows represent clicks for the user. (**A**) The interface to the SOTArray. (**B**) The results page, from which different plots can be obtained. (**C**) SOTA tree program was invoked from the results page. This graphical interface allows for some interactive changes in the final appearance of the tree represented. (**D**) A dendrogram representing the clusters found in the dataset. The diameter of the circle is proportional to the number of co-expressing genes in the cluster. By clicking the circle or the histogram representing the average profile, the list of genes (**E**) and a representation of its profiles (**F**), respectively, can be obtained. This is done by the internal module ExtractCluster. (**G**) Shows part of the graphical representation of FatiGO module, which displays the percentages of GO terms in the selected cluster with respect to the rest of genes and the adjusted p-values obtained for this difference.

cluster from the whole data set (Fig. 3E). The cluster can be sent to the Preprocessor module or to any of the analysis tools, if the user wants to post-analyse the cluster.

Data interpretation using gene ontology

The rationale behind clustering is that, despite the grouping being established by means of a given distance measure, it must reflect some biological property or function. The next step in the analysis consists of extracting the information and biological characteristics common to groups of genes of interest. Unfortunately, a large number of the available resources compiling information on gene (or protein) function or properties are based in the pre-genomic design in which the information is acceded and displayed in format one-gene-at-a-time. Such resources are useless if the aim is to detect some biological property or function shared by a set of genes when thousands of them are involved in the comparison. This gap between the clustering and the final study of the available information for a set of selected genes cannot be performed by hand because the amount of information implied in this step is too great to be processed by traditional methods. The FatiGO module can deal with thousands of genes and extract the GO (26) terms of relevance for a given set of genes with respect to the rest of them. These terms are obtained with the application of a test that takes into account the multiple-testing nature of the statistical contrast. The module produces a graphical representation with a bar chart with the proportion of GO terms in the analysed cluster with respect to the cluster of reference. Adjusted p-values for the differentially represented GO terms are given too. Also, links to the GO terms as well as to the genes are provided.

FatiGO can be applied to the validation of clusters of genes obtained with data of different nature. FatiGO can be applied to yeast, human and mouse genes and, in general, to proteins in TrEMBL/SWISS-PROT.

GEPAS USAGE

The way in which GEPAS is used depends very much on the particular scientific problem for which the microarray experiment has been designed and can be understood by looking at Figure 1. Usually, a first step of data preprocessing is required. It is quite commonly the interest in discovering the groups of genes that co-express under the experimental conditions studied. In this case unsupervised clustering programmes, such as aggregative hierarchical clustering, SOM, SOTA or SOM-Tree can be used. Unsupervised clustering can also be applied to obtain groups of experimental conditions (see for example 8). However, it is quite common to have some previous knowledge on the classes and, consequently, the interest of the experiment is more focused on class prediction (25). In this case programmes such as SVM can be used. If, in addition, the researcher is interested in the particular genes that are differentially expressed among classes, the Pomelo tool can be used. In fact, this tool can not only be used for discrete classes but also to relate genes to continuous measures (e.g. the concentration of a metabolite) or even to survival data. In many cases the application of the FatiGO tool can be useful to understand what are the biological processes and molecular functions of the genes selected as important in any of the previous steps.

EXTERNAL CONNECTIVITY

GEPAS modules can be invoked from other web resources and vice versa. This allows other designers of web tools to use partial or full GEPAS resources. At present, GEPAS can send data files, in the proper format, to Expression Profiler (4), a web tool at the European Bioinformatics Institute (EBI), and to HAPI, a data mining tool based on hierarchies of MESH terms (6).

CONCLUSIONS

Large numbers of algorithms exist for analysing microarray gene expression data (see a compilation in http://ihome.cuhk.edu.hk/~b400559/arraysoft.html). Some of them are parts of packages and others are implemented as stand-alone tools. Different tools or packages have been implemented in distinct platforms. In many cases, tools are implemented for PC computers to take advantage of the graphical capacity of some programming languages. The use of Java allows cross-platform usage of some packages (see, for example MEV, http://www.tigr.org). However, applications, or packages, are usually restricted in terms of their usability . Moreover, its connectivity to other applications or integration in other, more complete, packages is often difficult and cumbersome. Implementation of tools as web applications facilitates the connectivity greatly. Unfortunately, many of the web implementations that exist at present have been conceived as stand-alone tools and do not make use of these advantages. GEPAS has been designed with the intention of taking full advantage of the web properties: connectivity, cross-platform and remote usage. The modular architecture allows the addition of new tools and facilitates the federation of GEPAS with other webbased tools.

GEPAS is part of the pipeline of microarray data analysis in the CNIO (27) and its modules or the complete package is currently being used worldwide. Future work includes the addition of more tools and the improvement of the connectivity with other web-based tools.

ACKNOWLEDGEMENTS

F.A. is supported by grant BIO2001-0068 from MCYT, J.H. is supported by a CNIO fellowship, A.M. is supported by a IBM fellowship, R.D.U. is supported by a Ramón y Cajal research contract from the MCyT. We are indebted to Amanda Wren for revising the English of the manuscript.

REFERENCES

1. Schena,M., Shalon,D., Heller,R., Chai,A., Brown,P.O. and Davis,R.W. (1996) Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proc. Natl Acad. Sci. USA*, **93**, 10614–10619.
2. Lockhart,D.J., Dong,H., Byrne,M.C., Follettie,M.T., Gallo,M.V., Chee,M.S., Mittmann,M., Wang,C., Kobayashi,M., Horton,H. and Brown,E.L. (1996) Expression monitoring by hybridisation to high-density oligonucleotide arrays. *Nat. Biotechnol.*, **14**, 1675–1680.
3. Stoeckert,C.J., Causton,H.C. and Ball,C.A. (2002) Microarray databases: standards and ontologies. *Nature Genet.*, **32**, 469–473.
4. Brazma,A. and Vilo,J. (2000) Gene expression data analysis. *FEBS Lett.*, **480**, 17–24.
5. Herrero,J., Díaz-Uriarte,R. and Dopazo,J. (2003) Gene expression data preprocessing. *Bioinformatics*, **19**, 655–656.
6. Masys,D.R., Welsh,J.B., Fink,J.L., Gribskov,M., Klacansky,I. and Corbeil,J. (2001) Use of keyword hierarchies to interpret gene expression patterns. *Bioinformatics*, **7**, 319–326.
7. Eisen,M., Spellman,P.L., Brown,P.O. and Botsein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.
8. Perou,M., Jeffrey,S.S., van de Rijn,M., Ree,C., Eisen,M.B., Ross,D.T., Pergamenschikov,A., Williams,C.F., Zhu,S.X., Lee,J.C.F. *et al.* (1999) Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl Acad. Sci. USA*, **96**, 9112–9217.
9. Sneath,P.H.A. and Sokal,R.R. (1973) *Numerical Taxonomy.* W.H. Freeman, San Francisco, CA.
10. Tamayo,P., Slonim,D., Mesirov,J., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S. and Golub,T.R. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.
11. Törönen,P., Kolehmainen,M., Wong,G. and Castrén,E. (1999) Analysis of gene expression data using self-organizing maps. *FEBS Lett.*, **451**, 142–146.
12. Kohonen,T. (1997) *Self-organizing Maps.* Springer-Verlag, Berlin.
13. Herrero,J. and Dopazo,J. (2002) Combining hierarchical clustering and self-organizing maps for exploratory analysis of gene expression patterns. *J. Proteome Res.*, **1**, 467–470.
14. Herrero,J., Valencia,A. and Dopazo,J. (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, **17**, 126–136.
15. Dopazo,J. and Carazo,J.M. (1997). Phylogenetic reconstruction using a growing neural network that adopts the topology of a phylogenetic tree. *J. Mol. Evol.*, **44**, 226–233.
16. Mateos,Á., Herrero,J., Tamames,J. and Dopazo,J. (2002) Supervised neural networks for clustering conditions in DNA array data after reducing noise by clustering gene expression profiles. In Lin, M. (ed), *Microarray Data Analysis II.* Kluwer Academic Publishers, pp. 91–103.
17. Vapnik,V. (1998) *Statistical Learning Theory.* Wiley, New York.
18. Brown,M.P.S., Grundy,W.N., Lin,D., Cristianini,N., Sugnet,C.W., Furey,T.S., Ares,M. and Haussler,D. (2000) Knowledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl Acad. Sci. USA*, **97**, 262–267.
19. Furey,T.S., Cristianini,N., Duffy,N., Bednarski,D.W., Schummer,M. and Haussler,D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.
20. Dudoit,S., Yang,Y.H., Callow,M.J. and Speed,T.P. (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, **12**, 111–139.
21. Westfall,P.H. and Young,S.S. (1993) *Resampling-based Multiple Testing: Examples and Methods for p-value Adjustment.* John Wiley & Sons, New York.

22. Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Royal Stat. Soc. B*, **57**, 289–300.
23. Benjamini,Y. and Yekutieli,D. (2001) The control of the false discovery rate in multiple testing under dependency. *Annals Stat.*, **29**, 1165–1188.
24. Klein,J.P. and Moeschberger,M.L. (1997) *Survival Analysis.* Springer-Verlag. New York.
25. 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.
26. 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. *Nature Genet.*, **25**, 25–29.
27. Tracey,L., Villuendas,R., Ortiz,P., Dopazo,A., Spiteri,I., Lombardia,L., Rodríguez-Peralto,J.L., Fernández-Herrera,J., Hernández,A., Fraga,J. *et al.* (2002) Identification of genes involved in resistance to interferon-α in cutaneous T-cell lymphoma. *Am. J. Pathol.*, **161**, 1825–1837.