Open Source Software for the Analysis of Microarray Data

Sandrine Dudoit[1], Robert C. Gentleman[2], and John Quackenbush[3,4]

[1]University of California, Berkeley, CA; [2]Harvard School of Public Health and Dana Farber Cancer Institute, Boston, MA; [3]The Institute for Genomic Research, Rockville, MD; and [4]George Washington University, Washington, DC, USA

BioTechniques 34:S45-S51 (March 2003)

ABSTRACT

DNA microarray assays represent the first widely used application that attempts to build upon the information provided by genome projects in the study of biological questions. One of the greatest challenges with working with microarrays is collecting, managing, and analyzing data. Although several commercial and noncommercial solutions exist, there is a growing body of freely available, open source software that allows users to analyze data using a host of existing techniques and to develop their own and integrate them within the system. Here we review three of the most widely used and comprehensive systems, the statistical analysis tools written in R through the Bioconductor project (http://www.bioconductor.org), the Java®-based TM4 software system available from The Institute for Genomic Research (http://www.tigr.org/software), and BASE, the Web-based system developed at Lund University (http://base.thep.lu.se).

INTRODUCTION

DNA microarray analysis (1,2) has become the most widely used technique for generating genome-wide expression profiles and represents the first practical application that allows the results of whole-genome sequencing projects to be used effectively to address biologically relevant questions. Applications have ranged from the study of particular processes such as the cell cycle (3) to the clinically relevant problem of patient classification based on expression profiles (4–8). As the microarray technology matures, our ability to generate large numbers of high-quality assays quickly and efficiently is accelerating. One can no longer expect findings from microarray studies to be accepted without adequate replication and sampling to assure that the results reflect the underlying biological processes. Studies that previously would have been done with a small number of hybridizations now involve tens or hundreds of assays, and the challenge is moving from generating data to collecting, managing, and analyzing the data to identify statistically and biologically significant patterns of gene expression.

Although many have drawn analogies between data management in functional genomics and genome sequencing, the data structure in microarray studies and the ancillary data needed to accurately assess the results and to provide meaningful biological interpretation are much richer and more complex than those in genome sequence analysis. In the context of DNA sequencing, the age of the sample source, the tissue source, and any external treatments almost never have any bearing on the final sequence that is obtained. In contrast, when working with expression-based techniques, all of these factors, as well as other factors, can and do contribute to expression and must be considered. Furthermore, the identification and estimation of systematic errors and sources of variation in the data become fundamental to developing a sensible interpretation of the results.

Achieving these goals requires each laboratory's development of a comprehensive system of tools for data management and analysis. The rapid growth of the expression profiling field has prompted the development of an increasing number of commercial solutions to various aspects of this problem, each of which has particular strengths and weaknesses. However, many of the most innovative approaches to data analysis have been developed in the large number of academic laboratories practicing the technology, and a long time often elapses between when these approaches are first described to when they appear in commercial products.

In response to both the rapid growth of the field and the need for affordable, state-of-the-art tools, several efforts have recently sought to develop open source software for expression analysis, with the availability of both the program source code and well-defined standards for adding functionality to the software. Projects that aim to create comprehensive yet flexible systems for microarray data analysis are gaining an increasing presence in the field. Here we review three of the most widely used and comprehensive systems, the statistical analysis tools written in R (9) through the Bioconductor project (http://www.bioconductor.org), the TM4 software system available from The Institute for Genomic Research (TIGR; Rockville, MD, USA) (http://www.tigr.org/software), and the BioArray Software Environment (BASE) developed at Lund University (http://base.thep.lu.se).

THE VALUE OF OPEN SOURCE

There are a variety of definitions of open source software that differ primarily in the licensing agreements that govern how the source code and modifications to it can be used and distributed, particularly with respect to commercialization. The core definition that we will use is the free availability of the program source code and the availability of a clear, well-defined application

program interface (API) that allows developers to integrate the software with other systems and to add new functionality. Details on licensing agreements can be found at http://www.opensource.org and http://www.gnu.org.

Two obvious questions that arise are why anyone would want to release their software code and why others would want to add new utilities and functionality to someone else's software. Aside from the obvious benefits of creating a community resource that can advance the field, there are several advantages to an open source approach to software development in a scientific environment, including:

- full access to the algorithms and their implementation, which allows users to understand what they are doing when they run a particular analysis
- the ability to fix bugs and extend and improve the supplied software
- encouraging good scientific computing and statistical practice by providing appropriate tools, instruction, and documentation
- providing a workbench of tools that allow researchers to explore and expand the methods used to analyze biological data
- ensuring that the international scientific community is the owner of the software tools needed to carry out research
- promoting reproducible research by providing open and accessible tools with which to carry out that research (reproducible research as distinct from independent verification).

The creation of open source software is not unique to the scientific community. The best-known example is probably the development of the Linux operating system. For Linux, a worldwide community of developers has allowed the creation of an operating system that now commands a significant portion of the market, particularly for high-end systems. It is our hope that by helping to create an environment that encourages scientists to develop new applications and to make them accessible to laboratory biologists, we can create the same sort of community-based effort to drive the development of software and, in doing so, advance the general state of the art in functional genomics.

BIOCONDUCTOR: STATISTICAL ANALYSIS TOOLS DEVELOPED IN R

Use of R in Computational Biology

Computational biology (and the particular subset of it associated with the analysis of microarray data) is a challenging area of scientific exploration. Good methodology relies on many techniques and algorithms, including statistical and visualization methods. The use of a proper, well-designed, open source language in this role is essential. We believe that an existing open source language and environment for data analysis and visualization, called R, has a great deal to offer. While some may find command-line driven software challenging, there are significant advantages to having access to a high-level programming environment, together with a sophisticated software packaging and testing paradigm.

The R base package and packages from the Contributed R Archive Network (CRAN; http://cran.r-project.org) provide implementations for a broad range of state-of-the-art statistical and graphical techniques, including linear and nonlinear modeling, cluster analysis, prediction, resampling, survival analysis, and time-series analysis. Additionally, R has several mechanisms that allow it to interact directly with software that has been written in many different languages (e.g., intersystem interfaces provided at http://www.omegahat.org) and allow users to incorporate additional analysis modules. Viewed in that context, adopting R as a vehicle does not exclude other development environments and paradigms. Rather, R provides connectivity, thereby linking what might otherwise be different products or projects.

An Overview of the Bioconductor Project

Bioconductor (http://www.bioconductor.org) is an open source project for computational biology. All Bioconductor software packages are released under licenses such as BSD, GPL2, and LGPL. The project began in the fall of 2001 and currently involves 21 core developers who are based at research institutions in North America and Europe. The main focus is to deliver high-quality infrastructure and end-user tools for expression analysis. The primary delivery vehicle is R and the R package system. Bioconductor is an open development initiative. Users are encouraged to become developers, either by supplying Bioconductor-compliant packages, by adding to or improving existing packages, or by producing Bioconductor-compliant documentation. In addition to providing genomic data analysis tools, Bioconductor has a commitment to reproducible research and integrated, dynamic documentation. Each Bioconductor package contains at least one vignette, which is a document that provides a textual, task-oriented description of the package's functionality and can be used interactively. These executable documents are generated using the function Sweave from the R tools package (10). Additional supporting software for vignettes is being developed to aid users with obtaining data and sample code, step through specific analyses, and apply these analyses to their own data using Bioconductor's DynDoc package.

The first Bioconductor software release occurred on May 2, 2002 and included 15 packages. The second release occurred on November 18, 2002 and included five new packages in addition to enhancements to existing packages. Bioconductor packages are listed in Table 1 and some are described in greater detail below. Although initial efforts focused primarily on DNA microarray data analysis, many of the software tools are general and can be used broadly for the analysis of genomic and expression data, such as SAGE, sequence, or SNP data. Object-oriented programming with well-defined classes is one of the best tools available for overcoming data complexity. Another advantage of adopting an object-oriented approach is that basic classes can be extended to deal with special situations. This is usually a much smaller task than an entirely new implementation. For these reasons, Bioconductor has adopted object-oriented programming as its primary programming paradigm (11).

General Programming Tools

Biobase. This package provides base functions needed by other Bioconductor packages (12). An object-oriented class/

method programming approach is used for the efficient representation and manipulation of large biological datasets of multiple types. In particular, the exprSet class provides a systematic representation of microarray expression data, which includes not only the expression measures but also responses and covariates associated with each target sample (e.g., patient survival, tumor class, response to treatment, age, and sex) and annotation information on probe sequences. Microarray object classes are designed to follow the MIAME (minimum information about a microarray experiment) standards (13).

Table 1. Packages from Bioconductor

Task	Packages
General programming tools	Biobase, graph, tkWidgets, reposTools, rhdf5
Annotation	AnnBuilder, annotate
Graphics	geneplotter, hexbin
Preprocessing microarray data	affy
	marrayClasses, marrayInput, marrayNorm, marrayPlots, marrayTools
	vsn
Differential gene expression	edd, genefilter, multtest, ROC

tkWidgets. The tkWidgets package represents one of the more ambitious components of the project. Perhaps the largest problem with using a language such as R is that first-time users can be discouraged by the complexity of the language. To overcome this barrier to entry, we are designing a widget mechanism to provide interactive access to much of the Bioconductor functionality. A widget can be thought of as a small-scale graphical user interface (GUI). tkWidgets builds on the tcltk package that provides an interface and language bindings to Tcl/Tk GUI elements in R. The tkWidgets package was used to generate widgets for file browsing and data input in the affy and marrayInput packages. Extensions are planned for the near future.

geneplotter. This package provides graphical tools for genomic data; for example, for plotting expression data along a chromosome or producing color images of expression data matrices (e.g., alongChrom function). Once again, a modular approach is emphasized. The functions in this package will interact with any data package produced by AnnBuilder so that the functionality is independent of the source of the data and the organism (certain biological information, such as number of chromosomes, will of course be needed).

Preprocessing Affymetrix Data: affy Package

This package provides functions for the exploratory analysis and preprocessing of Affymetrix oligonucleotide chip data (14). It contains class definitions and associated methods for handling probe-level Affymetrix data. Functions and widgets are provided for data input from CDF and CEL files and the automatic generation of microarray data objects. Methods are defined to produce basic diagnostic plots of probe-level data (e.g., 2-D spatial images, boxplots, and histograms). The package implements several procedures for background estimation, probe-level normalization (quantile and curve-fitting normalization), and computation of expression measures [MAS 4.0 AvDiff, MAS 5.0 Signal, model-based expression index (MBEI) (15), robust multichip analysis (RMA) of Irizarry et al. (16)].

Preprocessing cDNA Microarray Data: marrayClasses, marrayInput, marrayNorm, and marrayPlots Packages

This suite of four R packages provides functions for exploratory analysis and preprocessing of two-color cDNA microarray data (17). The marrayClasses package contains class definitions and associated methods for handling pre- and post-normalization intensity data for batches of arrays. The marrayInput package supplies functions and Tcl/Tk widgets to automate data input and the creation of microarray-specific R objects for storing these data. Functions for diagnostic plots of microarray spot statistics, such as boxplots, scatter plots, and spatial color images are provided in marrayPlots. Finally, the marrayNorm package implements robust adaptive location and scale normalization procedures, which correct for different types of dye biases using robust local regression (e.g., intensity, 2-D spatial, and plate normalization using loess) and allow the use of control sequences spotted onto the array and possibly spiked into the mRNA samples.

Differential Gene Expression

In the context of microarray data analysis, the following four packages can be used to identify differentially expressed genes (i.e., genes whose expression levels are associated with responses or covariates of interest). However, the functions are general and can be used in large multivariate problems to study relationships among groups of variables.

edd. The edd package implements graphical methods and pattern recognition algorithms for distribution shape classification. Grouping genes into several expression distribution classes should lead to more efficient procedures for assessing differential expression.

genefilter. The genefilter package provides tools for sequentially filtering genes using a wide variety of filtering functions. Examples of filters include: the number of missing values; the coefficient of variation of the expression measures across arrays; an analysis of variance (ANOVA) p-value; and a Cox model p-value. The package separates the gene filtering process into two main tasks: assembling the filters using the function filterfun and applying the filters using the function genefilter. Several filter functions are already supplied in the package; it is also straightforward for the user to define new filters.

multtest. Multiple testing procedures that control the family-wise error rate (FWER) and the false discovery rate (FDR) are provided in the multtest package (18). Tests can be based on t- or F-statistics for one- and two- factor designs. Permutation procedures are available to estimate adjusted p-values.

Annotation for Genomic Data

annotate. Microarray data gain relevance from their association with biological metadata from Web resources such as GenBank®, LocusLink, and PubMed. The annotate package provides functions for associating microarray and other genomic data in real time to biological metadata from Web databases. Environments (similar to hash tables in other languages) are used to provide maps between different sets of probe identifiers, such as those from Affymetrix, GenBank, PubMed, UniGene, and The Gene Ontology (GO) Consortium (19). Databases such as PubMed can be queried for multiple genes at a time, and the results of the search can be retrieved, stored, and processed within R. The package also provides tools for incorporating the results of statistical analyses in HTML reports with links to annotation Web resources. Note that this package is not tied to any one set of annotation. The functionality embodied here is independent of the data source, although we presume that it is related in some fashion to genes. The AnnBuilder package, described next, packages data in a format suitable for use with the annotate package.

AnnBuilder. This package provides tools for assembling and processing genomic annotation data from databases such as GenBank, the GO Consortium, LocusLink, UniGene, the UCSC Human Genome Project (20). Parsers are supplied to handle multiple data sources of varying formats. The resulting metadata assemblies are exported as XML files that can be used by different systems. The XML annotation files are further processed to be used with the annotate package.

AnnBuilder is used by the Bioconductor project to construct all the data packages that are distributed. Current offerings include mappings for all Affymetrix human arrays as well as mouse and rat arrays. New data packages, customized for cDNA or other short oligonucleotide arrays, are easily created.

TM4: A JAVA-BASED SYSTEM FOR MICROARRAY EXPRESSION ANALYSIS

Although Bioconductor has the advantage of building on the existing toolkit of statistical applications, the command-line driven applications present a challenge for many users in the laboratory. The tkWidgets package provides some tools for creating GUIs, but this requires additional programming. An alternative approach is to create a series of Java-based tools that provide users with an intuitive, easy-to-use interface that guides them through the process of data collection and analysis. The TM4 microarray analysis suite of tools (21) was developed to provide the microarray community with a comprehensive set of tools to handle all aspects of the microarray process. TM4 consists of four major applications and a MySQL database (http://www.mysql.com). TM4 is freely available to the research community and may be obtained with source code at http://www.tigr.org/software. Although developed for spotted two-color arrays, many of the components can be easily adapted to work with single-color formats such as filter arrays and Affymetrix GeneChips™. The creation of objects for reading Affymetrix data files is currently underway.

AGED: A Database for Expression Analysis

Microarray expression analysis typically generates a prodigious quantity of data. Effective and efficient analysis relies on capturing and recording data on the genes arrayed, the samples used for hybridization, the laboratory conditions and protocols used in the assays, and the parameters associated with obtaining and analyzing the hybridization data. Our group has extensive experience with developing databases for biological data, and we created a gene expression database (AGED) to support microarray projects. Although AGED was engineered to comply with the MIAME standards for microarray data reporting (13), it was primarily designed to serve as part of a laboratory information management system (LIMS) and therefore includes tables for tracking materials and protocols through a microarray experiment. AGED was originally implemented in Sybase®, but is now available in in MySQL (http://www.mysql.com), a freely available, open source ANSI SQL database system. To facilitate user interaction with AGED, we developed a series of Java-based data entry tools, known collectively as Microarray Data Manager (MADAM), and created a transaction management system to ensure the consistency of the data. Several groups have now adapted AGED for other database systems including Oracle®.

MADAM

MADAM is a data entry and management tool for microarray experiments that uses a series of data entry pages for each stage of the experiment. On the left side of each page is a map that tracks the user's progress through the process. Data entry pages have a variety of mandatory fields that change from red to white as data are entered as well as optional fields and comments. Tabs allow the users to change from data entry pages to a reports page that allow data retrieval for preloaded database queries and display the results in a simple format. A third tab provides the user with access to several data entry programs that were initially developed as stand-alone applications but are now integrated into MADAM, including a general SQL query tool, PCR Score, a bar-coding system, and other analysis tools. MADAM can read in data from a variety of image analysis tools, including Spotfinder (TIGR) and GenePix® (Axon Instruments, Union City, CA, USA), and adding utilities to parse other standard formats is relatively simple. MADAM is currently being adapted to read and write MAGE-ML (22), the recently adopted standard microarray data exchange format. MADAM can easily be adapted to work with the standardized arrays across multiple sites to facilitate entry into a common database while allowing consortium members access to all data in a standardized format.

TIGR Spotfinder: Semi-Automated Image Analysis Software

Image analysis is crucial for the analysis and interpretation of microarray expression data. Although there are now a large number of commercially available software systems that perform image processing and analysis, several years ago, we instituted the development of the TIGR Spotfinder. This program employs an innovative dynamic thresholding algorithm for spot intensity determination, provides quality measures for each spot, and inter-

faces directly with the AGED database (although MADAM allows the expression data measures from other image analysis software systems).

MIDAS: Data Normalization and Filtering

Many of the data mining algorithms used are memory and computationally intensive, and often users spend a good deal of time normalizing and filtering the data to select a subset. Further, many of these steps have become standardized in our analysis for specific projects, while data mining generally requires the manual inspection and exploration of the data. The Microarray Data Analysis System (MIDAS) uses a simple graphical scripting interface that allows users to define and apply a data reduction process that includes background- and quality-control trimming, normalization using locally weighted linear regression (lowess) and other approaches (23,24), replicate analysis and filtering, and the identification of differentially expressed genes using intensity-dependent Z-scores and user-defined fixed fold-change cut-offs. Running on a desktop PC, MIDAS can process a pair of replicate 32 000 element arrays through pen-tip lowess normalization, replicate filtering, and identification of significantly differentially expressed genes in less than 3 min. Recent additions to MIDAS include variance regularization, both within and between arrays to facilitate comparisons, and graphical displays of the data that allow the process to be evaluated. We are currently testing a Java implementation of Kerr and Churchill's MAANOVA tools for the analysis of array data and the identification of differentially expressed genes (25) that should be released in early 2003.

MeV: a Tool for Data Mining

To facilitate the assessment of microarray expression data, we developed the TIGR Multiexperiment Viewer (MeV), a data visualization and analysis tool. MeV was designed with the intent of making it widely accessible and available to researchers using microarrays and other techniques for assessing expression. It displays microarray expression measurements from single or multiple experiments using several intuitive representations of the data. The display elements are active, providing users with a direct link to annotation and other information associated with the underlying genes. Expression data from each experiment can be normalized using a number of approaches, although most users now normalize and filter data using MIDAS.

A large number of distance measures and analytic techniques have been implemented, including:

- Hierarchical clustering (HCL) (26)
- Bootstrapped/jackknifed HCL
- *k*-means/*k*-medians clustering (KMC) (27)
- *k*-means/*k*-medians support (iterative KMC)
- Self-organizing maps (SOMs) (28)
- Self-organizing trees (SOTA) (29)
- Cluster affinity search technique (CAST) (30)
- Figure of merit for CAST and KMC (soon also for SOMs) (31)
- *QT*-clust (32)
- Principal component analysis (PCA) (33)
- Gene shaving (34)
- Relevance networks (35)
- Support vector machines (SVMs) (36)
- Classification approaches, including template matching (37)
- *t* tests

The development of MeV is ongoing, and work is underway on a set of tools for ANOVA, as well as tools for annotation, clustering, and visualization based on GO terms (19), metabolic pathway analysis, and genome localization. Other development efforts include tools for the analysis of time-series experiments and additional tools for expression-based classification and class discovery. MeV was developed to be used in a client-server architecture, with a thin client on the user's desktop computer passing computationally intensive jobs to a more powerful central server. However, the default installation sets the client and server to be the same machine; in practice, most datasets are easily handled by modern desktop workstations.

TM4 Documentation and Training

To assist users of TM4, we developed an extensive training manual based on workshops that we have presented over the past few years. This manual takes users step by step through the various software components, with screen snapshots to make the process easier. The manual starts with PCR amplification and moves through the analysis of single and multiple hybridization assays. As is the case with the software, the manual has been through multiple revisions and has benefited significantly from comments and suggestions from users.

BASE: THE BIOARRAY SOFTWARE ENVIRONMENT

While TM4 overcomes some of the limitations of a command-line driven system, it has the disadvantage of requiring users to maintain current copies of the software locally and to update them as the system evolves. A possible way to circumvent these problems is to implement a Web-accessible system that uses standard browsers to interact with a central microarray database and appropriate data analysis tools. This approach has the advantage that users of the system do not have to worry about whether their software is current and that the calculations underlying any analysis can be effortlessly passed to more powerful central servers rather than being limited to users' desktop computers. BASE (http://base.thep.lu.se) was developed by Peterson and colleagues (38) at Lund University using such a Web-based approach.

An Overview of the BASE System

BASE was designed with the goal of supporting a variety of microarray platforms. Underlying the complete system is a MIAME-supportive, customizable database implemented in MySQL that tracks the elements used to construct the arrays and their annotations, the layout and design of the array itself, the biological samples used in each hybridization assay, and both the raw and transformed data; users have the option of including other LIMS components for tracking samples and reagents in the laboratory. The software that interacts with this database was de-

veloped under the Linux operating system in PHP and uses a freely available Apache Web server (http://httpdf.apache.org) to provide Web access to its functionality. The interface uses Java®, JavaScript®, and HTML to provide added utility, and some of the more computationally intensive analysis methods that are carried out on the server have been implemented in C++. Because of its flexible design, BASE can be used for the analysis of one- and two-color systems on a variety of substrates, cDNA and oligonucleotide arrays, Affymetrix GeneChips, and both expression analysis and comparative genomic hybridization analysis.

Data Management and Annotation in BASE

The analysis of expression data relies on the careful annotation of the biological materials, including both the biological reporter molecules on the arrays and the samples that are hybridized to the arrays, is critical. BASE has a well-designed system that allows complex ancillary annotation such as genotype, mutation profile, patient data, or the immunohistochemical status of samples for particular proteins to be recorded and tracked through the process to be used as an aid in the interpretation of the experimental results. This is accomplished through a user-customizable Web-based interface that is closely integrated with the data analysis system.

Because there are now widely available, good-quality image-processing programs, the BASE system does not have its own image-processing system but instead relies on a flexible, interactive data import wizard that can read hybridization data and ancillary annotation for the genes and samples from tab-delimited files, such as those produced by any of the widely used microarray image-processing programs. The data matrix from either single or paired biological samples can be stored in the database as a unique dataset object, and users can designate groups of hybridizations into experimental groups, annotated, and analyzed.

Data Analysis

Because methods for expression analysis tools are evolving rapidly, BASE has a plug-in architecture that allows new modules to be easily added for data transformation, analysis, or visualization. Any executable program that runs on Linux and can read and write a standard data format (currently their "BASE-file" format) can be adapted as a plug-in.

BASE currently contains three analysis modules that were developed as a demonstration of the plug-in structure. The Normalizer application performs within-slide global mean or median ratio-based normalization. Lowess performs the intensity-dependent locally weighted linear regression (23,24,39) described earlier. The MDS module does multidimensional scaling, using the gene expression vectors to calculate distances between samples and to reduce that to 2- or 3-D representations that are easily visualized. The 3D Data Viewer allows users to visualize and explore the 3-D projection. Because data analysis typically contains multiple steps, BASE incorporates a data analysis interface that allows users to define an analysis method that passes data through multiple routines and to create transformed datasets and subsets. This allows the original unmodified data to be analyzed in a variety of ways to create multiple analyses. All parameters and settings are stored at each step, and the analysis history is presented visually.

BASE allows data to be visualized in a variety of ways. Unmodified and transformed datasets can be plotted interactively as scatter plots, displayed in histograms, or viewed as tables. Multiple hybridization experiments can be viewed in a variety of overview plots, based on their annotation and summary figures, and tables can be generated. The Experiment Explorer allows users to browse the data array element by element in the context of sample and element annotation. The data can also be exported for other analyses programs that can be run locally, such as Cluster (26) and J-Express (40).

SUMMARY: LOOKING TO THE FUTURE

Bioconductor, TM4, and BASE represent different approaches to the same problem, and each has its advantages and disadvantages. Bioconductor builds on the existing power of the R statistical analysis tool development community and allows for the rapid development and dissemination of new methods. However, the R command-line environment and language complexity can be discouraging to first-time users. Several efforts are underway to simplify and enhance the user interface. TM4 gives users a graphical interface that is easy to navigate and the architecture provides great flexibility for development. However, implementation of new statistical tools requires the creation of new analysis libraries and users have to install new software releases. BASE minimizes the software update problem by using a Web-based approach and, as such, could easily integrate the Bioconductor utilities, but it loses a good deal of the graphical functionality that local applications can provide.

These projects have in common is the availability of their software source code, which allows users to modify the program to both meet the local needs in each laboratory and to continue to expand its functionality. The recent establishment of the MAGE-ML standard (22) for representing microarray data promises to provide a means by which these and other systems can communicate and exchange data and results. One might hope that the software development efforts described here and other projects will converge and that their integration will result in set of tools that has the advantages of all of these without their limitations. The possibility that this will happen depends on open access to the source code, which will allow the community to leverage our collective expertise to the benefit of everyone working in gene expression analysis and related areas.

ACKNOWLEDGMENTS

We would like to thank Vincent Carey, Jeff Gentry, Rafael Irizarry, Yee Hwa Yang, Jianhua Zhang, Terence P. Speed, Alexander Saeed, Vasily Sharov, Joseph White, Jerry Li, Wei Liang, John Braisted, Tracey Currier, Mathangi Thiagarajan, and Eleanor Howe for valuable contributions. S.D. is supported by grant no. R01 LM007609-01 from the National Institutes of Health. J.Q. is supported by grant nos. U01 HL66580-01, U01-CA8552-01A1, and 1 R33 HL3712-01 from the National Institutes of Heath, and grant no. DBI-0177281 from the National Science Foundation.

REFERENCES

1. **Schena, M., D. Shalon, R.W. Davis, and P.O. Brown.** 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science *270*:467-470.
2. **Shalon, D., S.J. Smith, and P.O. Brown.** 1996. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. Genome Res. *6*:639-645.
3. **Iyer, V.R., M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee, J.M. Trent, L.M. Staudt, et al.** 1999. The transcriptional program in the response of human fibroblasts to serum. Science *283*:83-87.
4. **Bhattacharjee, A., W.G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, et al.** 2001. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc. Natl. Acad. Sci. USA *98*:13790-13795.
5. **Golub, T.R., D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, et al.** 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science *286*:531-537.
6. **Perou, C.M., T. Sorlie, M.B. Eisen, M. van de Rijn, S.S. Jeffrey, C.A. Rees, J.R. Pollack, D.T. Ross, et al.** 2000. Molecular portraits of human breast tumours. Nature *406*:747-752.
7. **Su, A.I., J.B. Welsh, L.M. Sapinoso, S.G. Kern, P. Dimitrov, H. Lapp, P.G. Schultz, S.M. Powell, et al.** 2001. Molecular classification of human carcinomas by use of gene expression signatures. Cancer Res. *61*:7388-7393.
8. **van't Veer, L.J., H. Dai, M.J. van de Vijver, Y.D. He, A.A. Hart, M. Mao, H.L. Peterse, K. van der Kooy, et al.** 2002. Gene expression profiling predicts clinical outcome of breast cancer. Nature *415*:530-536.
9. **Ihaka, R. and R. Gentleman.** 1996. R: Language for data analysis and graphics. R:{A}. Language Data Anal. Graph. *5*:299-314.
10. **Leisch, F.** 2002. Sweave: dynamic generation of statistical reports using data analysis. *In* W. Härdle and B. Rönz (Eds.), Compstat 2002—Proceedings in Computational Statistics. Physika Verlag, Heidelberg, Germany. (http://www.ci.tuwien.ac.at/~leisch/Sweave).
11. **Chambers, J.M.** 1998. Programming with Data: A Guide to the S Language. Springer-Verlag, New York.
12. **Gentleman, R. and V. Carey.** Visualization and annotation of genomic experiments. *In* G. Parmigiani, E.S. Garrett, R.A. Irizarry, and S.L. Zeger (Eds.), The Analysis of Gene Expression Data: Methods and Software. Springer-Verlag, New York. (In Press.)
13. **Brazma, A., P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, et al.** 2001. Minimum information about a microarray experiment (MIAME)—toward standards for microarray data. Nat. Genet. *29*:365-371.
14. **Irizarry, R.A., L. Gautier, and L. Cope.** An R package for analyses of Affymetrix oligonucleotide arrays. *In* G. Parmigiani, E.S. Garrett, R.A. Irizarry, and S.L. Zeger (Eds.), The Analysis of Gene Expression Data: Methods and Software. Springer-Verlag, New York. (In press.)
15. **Li, C. and W.H. Wong.** 2001. Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. Proc. Natl. Acad. Sci. USA *98*:31-36.
16. **Irizarry, R.A., B. Hobbs, F. Collin, Y.D. Beazer-Barclay, K.J. Antonellis, U. Scherf, and T.P. Speed.** Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics (In press.)
17. **Dudoit, S. and Y.H. Yang.** Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. *In* G. Parmigiani, E. S. Garrett, R.A. Irizarry, and S.L. Zeger (Eds.), The Analysis of Gene Expression Data: Methods and Software. Springer-Verlag, NY (In press.)
18. **Dudoit, S., J.P. Shaffer, and J.C. Boldrick.** 2002. Multiple hypothesis testing in microarray experiments. Technical Report 110, Division of Biostatistics, University of California, Berkeley. URL http://www.bepress.com/ucbbiostat/paper110/.
19. **Ashburner, M., C.A. Ball, et al.** 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. *25*:25-29.
20. **Zhang, J., V. Carey, and R. Gentleman.** An extensible application for assembling annotation for genomic data. Bioinformatics *19*:155-156.
21. **Saeed, A.I., V. Sharov, J. White, J. Li, W. Liang, N. Bhagabati, J. Braisted, M. Klapa, et al.** TM4: a free, open source system for microarray data management and analysis. BioTechniques (In press.)
22. **Spellman, P.T., M. Miller, J. Stewart, C. Troup, U. Sarkans, S. Chervitz, D. Bernhart, G. Sherlock, et al.** 2002. Design and implementation of microarray gene expression markup language (MAGE-ML). Genome Biol. *3*:RESEARCH0046.
23. **Yang, I.V., E. Chen, J.P. Hasseman, W. Liang, B.C. Frank, S. Wang, V. Sharov, A.I. Saeed, et al.** 2002. Within the fold: assessing differential expression measures and reproducibility in microarray assays. Genome Biol. *3*:RESEARCH0062.
24. **Yang, Y.H., S. Dudoit, P. Luu, D.M. Lin, V. Peng, J. Ngai, and T.P. Speed.** 2002. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. Nucleic Acids Res. *30*:e15.
25. **Kerr, M.K., M. Martin, and G.A. Churchill.** 2000. Analysis of variance for gene expression microarray data. J. Comput. Biol. *7*:819-837.
26. **Eisen, M.B., P.T. Spellman, P.O. Brown, and D. Botstein.** 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA *95*:14863-14868.
27. **Soukas, A., P. Cohen, N.D. Socci, and J.M. Friedman.** 2000. Leptin-specific patterns of gene expression in white adipose tissue. Genes Dev. *14*:963-980.
28. **Kohonen, T.** 1998. Self-organized formation of topologically correct feature maps. Biol. Cybern. *43*:59-69.
29. **Herrero, J., A. Valencia, and J. Dopazo.** 2001. A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics *17*:126-136.
30. **Ben-Dor, A., R. Shamir, and Z. Yakhini.** 1999. Clustering gene expression patterns. J. Comput. Biol. *6*:281-297.
31. **Yeung, K.Y., D.R. Haynor, and W.L. Ruzzo.** 2001. Validating clustering for gene expression data. Bioinformatics *17*:309-318.
32. **Heyer, L.J., S. Kruglyak, and S. Yooseph.** 1999. Exploring expression data: identification and analysis of coexpressed genes. Genome Res. *9*:1106-1115.
33. **Raychaudhuri, S., J.M. Stuart, and R.B. Altman.** 2000. Principal components analysis to summarize microarray experiments: application to sporulation time series. Pac. Symp. Biocomput. 455-466.
34. **Hastie, T., R. Tibshirani, M.B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W.C. Chan, D. Botstein, and P. Brown.** 2000. Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. Genome Biol. *1*:RESEARCH0003.
35. **Butte, A.J. and I.S. Kohane.** 1999. Unsupervised knowledge discovery in medical databases using relevance networks. Proc. AMIA Symp. 711-715.
36. **Brown, M.P., W.N. Grundy, et al.** 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc. Natl. Acad. Sci. USA *97*:262-267.
37. **Pavlidis, P. and W.S. Noble.** 2001. Analysis of strain and regional variation in gene expression in mouse brain. Genome Biol. *2*:RESEARCH0042.
38. **Saal, H., C. Troein, J. Vallon-Christersson, S. Gruvberger, A. Borg, and C. Peterson.** 2002. BioArray Software Environment (BASE): a platform for comprehensive management and analysis of microarray data. Genome Biol. *3*:SOFTWARE 6003.
39. **Cleveland, W.S. and S.J. Devlin.** 1988. Locally-weighted regression: an approach to regression analysis by local fitting. J. Am. Stat. Assoc. *83*:596-610.
40. **Dysvik, B., I. Jonassen.** 2001. J-Express: exploring gene expression data using Java. Bioinformatics *17*:369-370.

Address correspondence to:

Dr. John Quackenbush
The Institute for Genomic Research
9712 Medical Center Drive
Rockville, MD 20858, USA
e-mail: johnq@tigr.org