Web-based Microarray Analysis using Bioconductor

by Colin A. Smith

Introduction

The Bioconductor project is an open source effort which leverages R to develop infrastructure and algorithms for the analysis of biological data, in particluar microarray data. Many features of R, including a package-based distribution model, rapid prototyping, and selective migration to high performance implementations lend themselves to the distributed development model which the Bioconductor project uses. Users also benefit from the flexible command line environment which allows integration of the available packages in unique ways suited to individual problems.

However, features to one individual may be roadblocks to another. The use of microarrays for gene expression profiling and other applications is growing rapidly. Many biologists who perform these experiments lack the programming experience of the typical R user and would strongly object to using a command line interface for their analysis.

Here we present an overview of a web-based interface that attempts to address some of the difficulties facing individuals wishing to use Bioconductor for their microarray data analysis. It is distributed as the **webbioc** package available on the Bioconductor web site at `http://www.bioconductor.org/`.

Target audience and interface goals

While targeted at many user types, the web interface is designed for the lowest common denominator of microarray users, e.g. a biologist with little computer savvy and basic, but not extensive, statistical knowledge in areas pertinant to microarray analysis. Note that although this is the lowest level user targeted by the web interface, this interface also caters to power users by exposing finer details and allows flexibility within the preconstructed workflows.

This article presents only the first version of a Bioconductor web interface. With luck, more R and Perl hackers will see fit to add interfaces for more aspects of microarray analysis (e.g. two-color cDNA data preprocessing). To help maintain quality and provide future direction, a number of user interface goals have been established.

- *Ease of use.* Using the web interface, the user should not need to know how to use either a command line interface or the R language. Depending on the technical experience of the user, R tends to have a somewhat steep learning curve. The web interface has a short learning curve and should be usable by any biologist.

- *Ease of installation.* After initial installation by a system administrator on a single system, there is no need to install additional software on user computers. Installing and maintaining an R installation with all the Bioconductor packages can be a daunting task, often suited to a system administrator. Using the web interface, only one such installation needs to be maintained.

- *Discoverability.* Graphical user interfaces are significantly more discoverable than command line interfaces. That is, a user browsing around a software package is much more likely to discover and use new features if they are graphically presented. Additionally, a unified user interface for the different Bioconductor packages can help add a degree to cohesiveness to what is otherwise a disparate collection of functions, each with a different interface. Ideally, a user should be able to start using the web interface without reading any external documentation.

- *Documentation.* Embedding context-sensitive online help into the interface helps first-time users make good decisions about which statistical approaches to take. Because of its power, Bioconductor includes a myriad of options for analysis. Helping the novice statistician wade through that pool of choices is an important aspect of the web interface.

Another aspect of the target audience is the deployment platform. The web interface is written in Perl, R, and shell scripts. It requires a Unix-based operating system. Windows is not supported. It also uses Netpbm and optionally PBS. For further information, see the **webbioc** vignette at `http://www.bioconductor.org/viglistingindex.html`.

User-visible implementation decisions

There are a number of existing efforts to create web interfaces for R, most notably Rweb, which presents the command line environment directly to the user. See `http://www.math.montana.edu/Rweb/`. The Bioconductor web interface, on the other hand, entirely abstracts the command line away from the user. This results in an entirely different set of design decisions.

The first choice made was the means by which data is input and handled within the system. In an R session, data is instantiated as variables which the user can use and manipulate. However, in the web interface, the user does not see variables associated with an R session but rather individual files which hold datasets, such as raw data, preprocessed data, and analysis result tables.

Different stages of microarray analysis are divided into individual modules. Each module leads the user through a series of steps to gather processing parameters. When ready, the system creates an R script which it either runs locally or submits to a computer cluster using the Portable Batch System. Any objects to be passed to another module are saved in individual files.

Another decision which directly impacts the user experience is that the system does not maintain accounts for individual users. Instead, it uses the concept of a uniquely identified session. When a user first starts using the web interface, a session is created which holds all the uploaded and processed data. The system provides the user with a session token comprised of a random string of letters and numbers. The token allows the user to return to their session at a future date.

This offers a number of advantages: 1) At the discretion of the local system administrator, the web analysis resource can be offered as either a public or a private resource. Such a restriction can be made at the web-server level rather than the code level. 2) It allows rapid development of the web interface without being bogged down in the implementation or integration of a user infrastructure. 3) As opposed to having no session whatsoever, this allows a user to input data only once. Raw data files are often quite large. Uploading multiple copies of such datasets for each change in parameters is not desirable.

Lastly, the web interface brings the idea of design-by-contract used in the Bioconductor project down to the package level. That is, individual interface modules are responsible for a specific stage or type of analysis. Modules may take the user through any number of steps as long as they use standard input and output formats. This allows the system to grow larger and more powerful over time without making individual components more complex than is necessary to fulfill their function.

Analysis workflow

The web interface is currently limited to processing data from microarray experiments based on the Affymetrix GeneChip platform. It does however handle an entire workflow going from raw intensity values through to annotated lists of differentially expressed genes.

Affymetrix microarray data comes in the form of CEL files containing intensity values for individual probes. Because all processing is done server-side, that data must first be transferred with the Upload Manager. While raw data files can each be over ten megabytes, today's fast ethernet networks provide very acceptable performance, with file transfers taking only a few seconds.

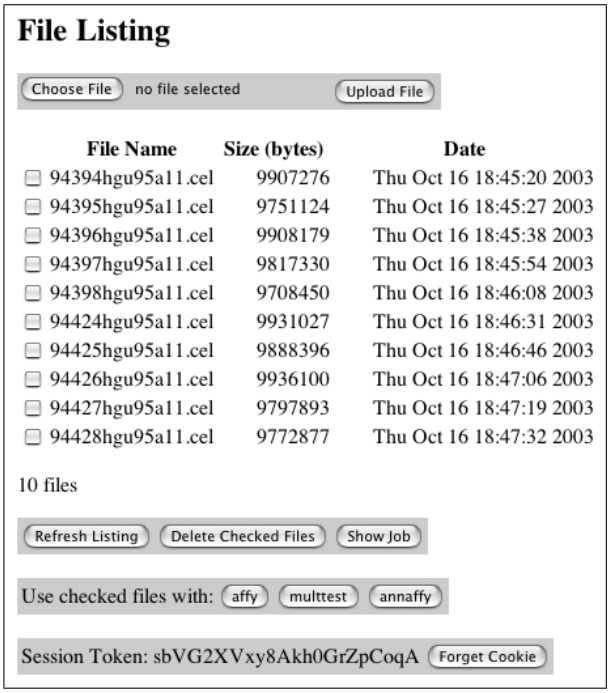

Figure 1: Upload Manager

Affymetrix preprocessing is handled by the **affy** Bioconductor package. The core functionality of that package is captured by only a handful of functions and thus lends itself to a simple web interface. The user may choose either the built-in high performance function for RMA or a custom expression measure. The custom expression measure also uses additional plug-in methods from the **vsn** and **gcrma** packages, which leverage the modular design of **affy**.

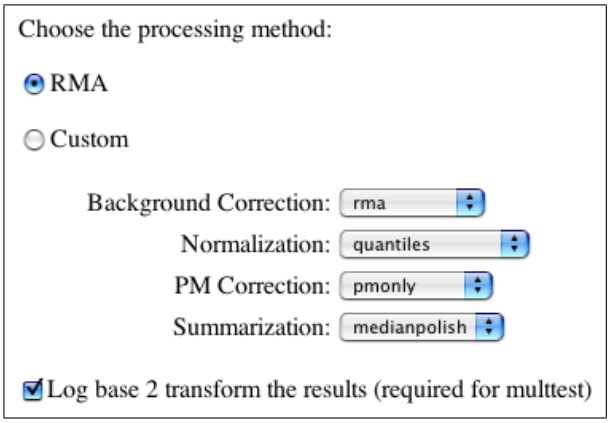

Figure 2: Affymetrix Data Preprocessing

There are a number of methods for identifying differentailly expressed genes. The web interface currently uses basic statistical tests (t-tests, F-tests,

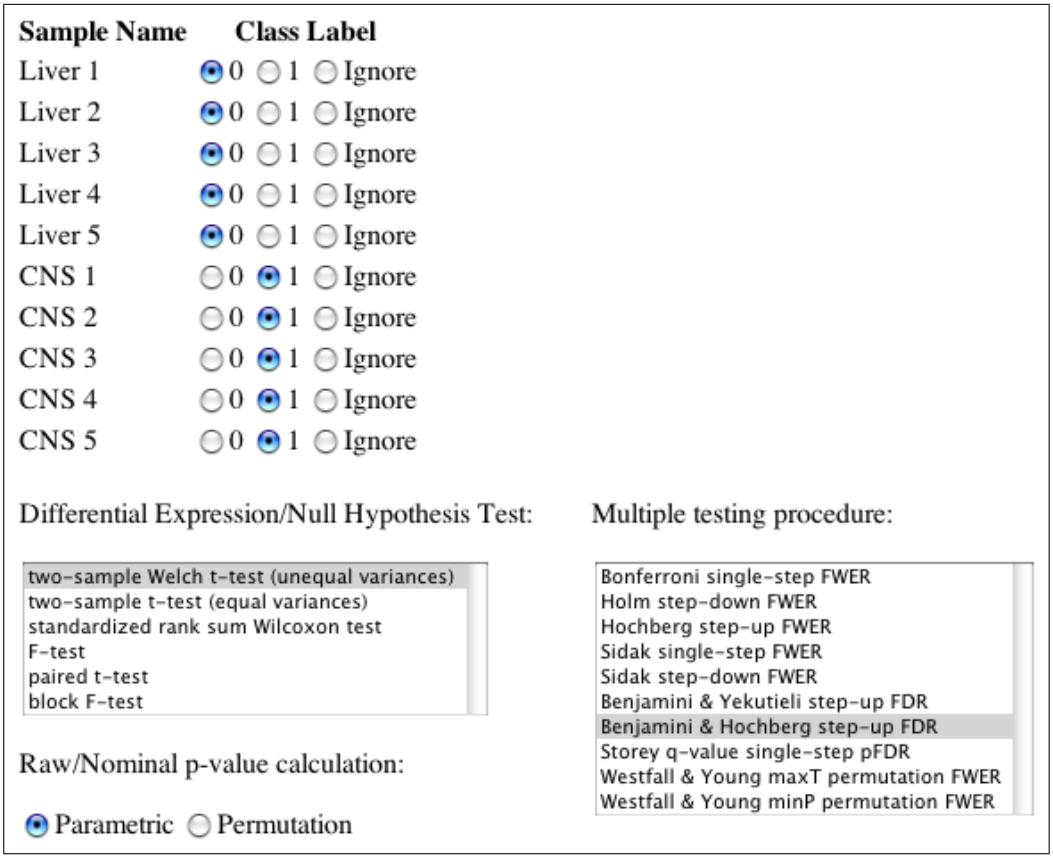

Figure 3: Differential Expression and Multiple Testing

etc.) combined with multiple testing procedures for error control of many hypotheses. These are implemented in the **multtest** package. Additionally, the web interface automatically produces a number of diagnostic plots common to microarray analysis. Those include M vs. A (log fold change vs. overall expression) and normal quantile-quantile plot.

The web interface completes the workflow by producing tables with integrated results and metadata annotation. Where appropriate, the annotation links to other online databases including a chromosome viewer, PubMed abstracts, Gene Ontology trees, and biochemical pathway schematics. The metadata presentation is handled by **annaffy**, another Bioconductor package.

In addition to presenting metadata, the web interface provides facilities for searching that metadata. For instance, it is trivial to map a set of GenBank accession numbers onto a set of Affymetrix probeset ids or find all genes in a given Gene Ontology branch. This assists biologists in making specific hypotheses about differential gene expression while maintining strong control over the error rate.

Lastly, because the web interface stores intermediate data as R objects, users of Bioconductor through either the command line or web interface can easily exchange data back and forth. Data exchange is currently limited to `exprSet` objects, which is the standard class for microarray data in Bioconductor. Future development of the interface should yield more data exchange options enabling novel collaboration between R users and non-users alike.

Final thoughts

An important consideration worthy of discussion is the inherent lack of flexibility in graphical user interfaces. The R command line interface does not box one into pre-scripted actions in the way that the web interface does. It allows one to exercise much more creativity in analysis and take more non-linear approaches. In the GUI trivial questions may by impossible to answer simply because of unforeseen limitations.

There are, however, a number of strengths in the web interface beyond what is available in R. The aforementioned interface goals are good examples of this. Additionally, the web interface can help reduce errors by distilling long series of typed commands into simple point-and-click options. All actions and parameters are tracked in a log for verification and quality control.

Secondly, the web interface easily integrates into existing institutional bioinformatics resources. The web has been widely leveraged to bring univer-

GeneLogic 20 μg Liver vs. CNS

Description	LocusLink	PubMed	Gene Ontology	Fold Change	t statistic	Bonferroni-value
NAD(P)H dehydrogenase, quinone 1	1728	22	NAD(P)H dehydrogenase (quinone) activity cytochrome b5 reductase activity nitric oxide biosynthesis response to toxin synaptic transmission, cholinergic xenobiotic metabolism electron transport cytoplasm oxidoreductase activity	15.379	123.152	2.83157e-10
Rho GDP dissociation inhibitor (GDI) beta	397	7	Rho GDP-dissociation inhibitor activity GTPase activator activity negative regulation of cell adhesion Rho protein signal transduction development immune response cytoplasmic vesicle actin cytoskeleton organization and biogenesis	0.134763	-123.716	6.47617e-10
tripartite motif-containing 16	10626	4	transcription factor activity cytoplasm	4.64131	104.939	1.09899e-09

Figure 4: Annotated Results and Online Database Links

sally accessible interfaces to common command-line bioinformatics tools. The system presented here can sit right next to those tools on a web site. Because it already uses PBS for dispatching computational jobs, the web interface can take advantage of existing computer clusters built for genomic search tools, such as BLAST, and can scale to many simultaneous users.

The web interface has been deployed and is currently in use by two research groups. One group is split between institutions located in different states. They use common session tokens and collaborate by sharing data and analysis results over the web.

Lastly, Bioconductor has implementations of a number of algorithms not otherwise freely available. Some newer algorithms have been exclusively implemented in Bioconductor packages. The web interface helps bring such innovations to the mainstream. It may even wet the appetite of some users, convincing them to take the plunge and learn R.

Colin A. Smith
NASA Center for Computational Astrobiology and Fundamental Biology
webbioc@colinsmith.org