Bioconductor: Assessment of Current Progress
Biocore Technical Report 2

Bioconductor Core

November 30, 2002

This report reviews current developments in Bioconductor, including low-level microarray analysis tools, contributions to analysis of gene expression data, pure computing, annotation, and training and dissemination efforts.

1 Microarray Data Analysis

1.1 Pre-processing Spotted DNA Microarray Data

Image analysis. The raw data from a microarray experiment are the image files produced by the scanner; these are typically pairs of 16-bit tagged image file format (TIFF) files, one for each fluorescent dye. Image analysis is required to extract foreground and background fluorescence intensity measurements for each spotted DNA sequence. We have developed novel methods for processing microarray images, based on seeded region growing segmentation and morphological opening background adjustment (19). These procedures are implemented in an R package, `Spot` (2).

Normalization. The purpose of *normalization* is to identify and remove the effects of systematic variation (*obscuring variation*), for example different labeling efficiencies and scanning properties of the Cy3 and Cy5 dyes; different scanning parameters, such as PMT (photomultiplier tube) settings; print-tip, spatial, or plate effects, other than differential expression (*interesting variation*) in the measured fluorescence intensities. It is necessary to normalize the fluorescence intensities before any analysis that involves comparing expression measures within or between slides (e.g., classification, multiple testing) in order to ensure that differences in intensities are indeed due to differential expression and not experimental artifacts. We have developed location and scale normalization methods which correct for intensity, spatial, and other dye biases using *robust locally weighted regression* (5; 20; 21). The `marray` suite of packages provides functions for exploratory analysis and pre-processing of spotted DNA microarray data, including the robust adaptive normalization procedures mentioned above (11).

`marrayClasses` This package provides the basic class structure and associated methods for handling pre- and post-normalization intensity data and data on probes and targets for batches of arrays.

`marrayInput` This package provides functionality for reading microarray data into R, such as intensity data from image processing output files (e.g., `.spot` and `.gpr` files for the `Spot` and `GenePix` packages, respectively) and textual information on probes and targets (e.g., from `.gal` files and god lists).

`marrayPlots` Examination of diagnostic plots of intensity data is important in order to identify printing, hybridization, and scanning artifacts that can lead to biased inferences concerning gene expression. This package provides functions for diagnostic plots of microarray spot statistics, such as boxplots, scatterplots (e.g., MA-plots of intensity log ratio $M = \log_2 R/G$ vs. the average log intensity $A = \log_2 \sqrt{RG}$), and spatial color images. Plots can be stratified according to various array layout parameters, such as sector and plate origin of the clones.

`marrayNorm` This package implements robust adaptive location and scale normalization procedures, which correct for different types of dye biases (e.g., intensity, spatial, plate biases) and allow the use of control sequences spotted onto the array and possibly spiked into the mRNA samples.

1.2 Pre-processing Affymetrix Oligonucleotide GeneChip Data

Image processing and normalization are two of the main pre-processing steps required in any microarray experiment. In addition, when dealing with Affymetrix high density oligonucleotide arrays, finding appropriate summaries of the probe-level data, that accurately represent the amount of corresponding target mRNA, is an important and challenging problem. Li and Wong (17) pioneered alternatives to standard measures offered by the Affymetrix software (e.g., MAS 4.0 and 5.0). Irizarry et al. (15) assessed the accuracy and precision of the three leading measures, namely those obtained from MAS 4.0, MAS 5.0, and dChip, using spike-in and dilution experiments. Through the careful exploration of probe-level data, they developed a mapping from probe-level intensities to expression measures, the robust multi-array analysis (RMA), that outperform the three other measures.

The Bioconductor `affy` package was developed as an extensible, interactive environment for the exploration of Affymetrix GeneChip probe-level data. As described in the letters of support, various scientist have found it to be a useful tool.

One of the most popular applications of the `affy` package is mapping probe level data to expression measures. To facilitate incorporation of new expression measures, we have modularized the code involved in this mapping into the following steps,

1. background correction;
2. normalization;
3. probe specific background correction, e.g., subtracting *MM*;
4. summarizing the probe set values into one expression measure and, in some cases, a standard error for this summary.

The package was used to develop RMA and as ideas emerge for improved expression measures, the package can be used to test them. The most popular versions can be implemented in C for efficiency and speed, as has been done for RMA in version 1.1 of the package.

In addition, the `affy` package provides a variety of functions for diagnostic plots of probe-level data. These include 2D spatial images of the probe-level intensities (or transformations of the intensities, such as log transformation) as they appear on the array. Such images are usually a first step in quality control and are especially useful for finding spatial artifacts. The histogram method permits the user to explore the distribution of the probe-level data of each array and can be used, for example, to detect saturation problems (as indicated by a mode near the maximum allowable intensity). In a similar way, the boxplot method allows the user to explore the distribution of intensities for multiple arrays and can be used to detect arrays that are behaving differently from the rest in global way.

1.3 Identification of Differentially Expressed Genes

Starting from very different data structures, both the `affy` and `marray` packages produce gene expression measures that can be represented in R objects of class `exprSet` (see `Biobase` package and discussion of experimental metadata in Section 3). This modular approach allows the end-user to rely on the full Bioconductor infrastructure for expression level analyses.

An important and common question in microarray experiments is the identification of differentially expressed genes, i.e., genes whose expression levels are associated with a response or covariate of interest (e.g., survival, tumor class). There are many different approaches that can be taken but, to date, most consider each gene individually and select *interesting genes* based on criteria such as fold-change or t-statistics. These methods can be extended by

considering the problem from a multivariate point of view or by including exploratory and diagnostic methods in the selection process.

While simple tools can provide selection criteria for relatively simple experiments, once the experimental design becomes more complex so does the analysis. We have developed a number of tools aimed at handling these more complex experimental structures. Examples of experiments include time-course experiments, factorial designed experiments, and comparisons of duration (survival or remission). The approach taken in the software package `genefilter` allows researchers to apply any analysis function found in R to select genes in this one-at-a-time approach. Additionally, any diagnostic checks desired are easily accommodated within the same computational framework (e.g., based on coefficient of variation and/or minimum intensity). Additional tests can be devised, implemented, and used without special requirements.

The biological question of differential expression can be restated as a problem in multiple hypothesis testing: the simultaneous test for each gene of the null hypothesis of no association between the expression levels and the responses or covariates. As a typical microarray experiment measures expression levels for thousands of genes simultaneously, large multiplicity problems are generated. The `multtest` package implements permutation based-multiple testing procedures for controlling the family-wise error rate (FWER) and the false discovery rate (FDR). Methodology implemented in this package was used to identify differentially expressed genes in a study of host genomic responses to bacterial infection (1), in a study of gene expression in mice with the apo AI gene knocked-out (3; 12), and in a study of gene expression in liver tumors (4).

A study of gene expression patterns as they relate to endometrial neoplastic transformation (18) was carried out using various statistical techniques, including a permutation approach to detecting significance. The methodology developed for this approach is distributed in the R package `permax`.

Irizarry et al. (16) considered statistical issues in the analysis of a designed experiment to investigate differential gene expression in colon cancer and normal colon tissue. In this experiment, gene expression was measured using radiolabeling-based array filters. Specific statistical issues arise in connection with radiolabeling technology, because of the absence of direct controls, which are replaced by empty spots on the filter, and with designed experiments, because of the opportunity to systematically quantify important sources of random variation. This paper considers three aspects in detail: normalization of expression intensities; shrinkage estimates of intensity ratios between cancer and normal tissue; and ranking of genes by the strength of the evidence that they are differentially expressed. Robust and simple-to-implement procedures for normalization and shrinkage, that addresses in a technology-specific way the problem of estimating ratios in the presence of small and noisy denominators, were proposed. A graphical display to rank genes using a metric based on quantiles of a null distribution obtained by replicating the array experiment in normal tissue was discussed.

1.4 Multivariate Analysis and Computational Inference

Many of the principal investigators have interests in multivariate statistical learning methods and computational inference. Multivariate methods for cluster analysis, class prediction, variable selection, and multiple testing are highly relevant for extracting biological knowledge from large and complex genomic dataset. For instance, in microarray cancer research, cluster analysis can be used to identify groups of genes with similar expression patterns across tumor samples and to identify previously unrecognized subclasses of tumors with distinct molecular and clinical characteristics. Classification methods are important tools in microarray experiments, for the purpose of classifying biological samples and predicting clinical or other outcomes using gene expression data. Extreme forms of multiple hypothesis testing are encountered in DNA microarray studies which involve detecting associations between biological responses and expression measures for thousands of genes simultaneously.

Computational inference methods, such as Markov chain Monte Carlo, the bootstrap, cross-validation, and permutation procedures, can be applied to address a variety of inferential problems in genomics, and are particularly useful in situations where the distributions of the statistics of interest are difficult to obtain. Areas of application include exploratory data analysis, multiple testing, class prediction, variable selection. Some of our contributions to the fields of computational inference and multivariate analysis are summarized next.

Exploratory data analysis. We have developed new methods for expression density diagnostics, including graphical and pattern recognition algorithms for distribution shape classification (see `edd` package). We have begun to develop exploratory analysis procedures based on graph theoretical notions (see `graph` package).

Multiple testing. We have studied the Type I error and power properties of a broad range of multiple testing procedures in the context of microarray data analysis (10). We have developed permutation procedures for estimating adjusted *p*-values for a class of multiple testing procedures controlling the family-wise error rate (14). The Bioconductor package `multtest` implements some of these novel procedures. In addition, we have applied multiple testing procedures to identify differentially expressed genes in microarray experiments (1; 3; 4; 12).

Statistical learning. We have introduced a novel prediction-based resampling method, `Clest`, to estimate the number of clusters, if any, in a dataset(8). We have developed two bagged clustering procedures to improve and assess the accuracy of a partitioning clustering method (6). We have also investigated the performance of different prediction methods for the classification of tumors based on gene expression data (7; 9). The methods include: nearest neighbor classifiers, linear discriminant analysis, support vector machines, and classification trees. Recent machine learning approaches such as bagging and boosting were also considered .

1.5 Visualization

Clustering methods have been used extensively and successfully in microarray experiments to organize and display very large genes-by-arrays data matrices. Eisen et al. (13) presented the first clustered display of gene expression data. Software enabling these visualization methods, Cluster and TreeView, is available from the Eisen Lab at `rana.lbl.gov`. The usefulness of these tools has seen their adoption in a wide variety of software packages.

The `geneplotter` package provides initial implementations of visualization methods that can be used to associate experimentally derived data with the biological metadata. In our case, we are reliant on data in Bioconductor packages since they are updatable, distributable, and are implemented in an efficient manner so that look-up is reasonable fast. These plots include whole genome views as well as per chromosome views of mRNA expression data.

A number of graphical utilities have been provided in the `Affy` package to support low-level analysis of oligonucleotide arrays, and in the `marrayPlots` package to support analysis of spotted DNA arrays. The use of classes and method overloading adds convenience to the resulting toolset, because users familiar with the graphical functions from common data analysis activities have a smooth transition to visualizing genomic data.

For oligo arrays, overall chip appearance can be inspected using an `image` method. Probeset-specific barplots of probe-pair intensities can be generated using the Affy ID, and appropriate use of annotation resources allows generation of such plots on the basis of various nomenclatures. Overall intensity distributions can be obtained using a boxplot method. MA-plots can be generated in a scatterplot matrix for a family of arrays.

For spotted DNA arrays, high-level routines are available for pre- and post-normalization diagnosis. The `maDiagnPlots1` routine in the `marrayPlots` package produces eight plots of pre- and post-normalization cDNA microarray data: color images of Cy3 and Cy5 background intensities, and of pre- and post-normalization log-ratios M; boxplots of pre- and post-normalization log-ratios M by print-tip-group; MA-plots of pre- and post-normalization log-ratios M by print-tip-group.

2 Computing

2.1 R Runtime and Data Interchange

We have enhanced the *R* runtime system in a number of ways that improve performance and also enhance *R*'s ability to take advantage of separate libraries and other software components. We developed a new non-moving generational memory management system that effectively handles the large amount of temporary memory allocation needed by a dynamic system like *R* but does not move objects in memory. This greatly simplifies passing values allocated in *R* to external subroutines. The memory manager also includes support for external references, weak references,

and finalization. These are needed to allow entities allocated outside of *R*'s heap, perhaps in a shared library or on a remote machine, to be accessed reliably from within *R* and allows the *R* memory management system to automatically manage the lifetime of these resources.

A profiling system has been added to *R* that allows accurate identification of computational hot spots. This allows developers to more effectively target their efforts to improve performance when this proves necessary for the effectiveness of an algorithm.

The serialization mechanism in *R* has been revised to allow effective serialization of arbitrary chunks of *R* data and code. This allows the development of flexible persistent storage mechanisms, for example using relational or object databases, and also provides the basis for transmitting code and data for remote execution. The default format used is a portable binary format designed for efficiency of access and storage. The mechanism can easily be adapted to use other formats such as SOAP.

We have begun preliminary work on developing a name space mechanism, a framework for byte code compilation, a high level exception handling system, and support for concurrent threads of execution.

2.2 Netscape-based Interface to R

The delivery of advanced analytic methods to biologists may be substantially enhanced by providing a browser-based interface to the associated software tools. Browser-based use of R can occur through the SNetscape plugin available at `www.omegahat.org`. However, a simpler method for mediating use of R through any browser using HTML and HTTP involves R's generic socket connection facility. A very preliminary version has been supplied as package `HTTPapp` at the Bioconductor CVS repository.

3 Annotation

3.1 Experimental Metadata

Many of the basic microarray data classes have been designed and implemented, and these classes reflect the structure of the underlying experimental materials and of the processes by which they are transformed into numerical and textual data. A basic requirement for orderly auditing and interpretation of an analysis is standardized structured information on the conduct of the experiment. We will refer to such information as *experimental metadata*. Bioconductor provides support for MIAME-compliant documentation of collections of expression experiments. There are two primary contact points at which MIAME information can be supplied or extracted. Within the `affy` package various classes contain CEL file data support the addition of MIAME compliant documentation. This documentation will be propagated to the `exprSet` objects defined in the `Biobase` and hence will be available to the data analyst. This system will be enhanced and extended to the `marray` suite of functions.

3.2 Biological Metadata

Mappings from oligonucleotide probe to LocusLink ID, and from LocusLink ID to Gene Ontology term or chromosomal location are all examples of *biological metadata*. Bioconductor possesses advanced utilities for creating and using such metadata.

The problem of annotating results of expression array experiments was recognized and confronted at the inception of the Bioconductor project. We aimed at the construction of a software framework, in contrast to a fixed annotation processor or database, acknowledging the following issues emerging in the design of software that helps manage associations of biological metadata with experimentally obtained:

1. *evolution condition:* the biological metadata are in flux as research matures;

2. *commonality condition:* there is some commonality of metadata and metadata requirements across species and across measurement instruments;

3. *disclosure sensitivity:* data (either experimental or annotation) may be subject to privacy controls (human subjects or proprietary constraints);

4. *magnitude:* for most species there is far too much metadata available to easily provide access to all relevant data without significant curation.

The evolution condition implies that a flexible design will be needed, with sharp limitation of assumptions about the underlying structure and content of biological metadata such as gene nomenclature or chromosomal mapping. The commonality conditions imply that abstractions will be productive: tool design should not capitalize too strongly on species-specific or instrument-specific features, but should always consider the possibility of embracing a wider class of species or instruments. In particular, interest will be strong in supporting combinations of results obtained from related but non-identical measurement systems. The disclosure-sensitivity condition implies that mechanisms for enhancing information security should be available. The final magnitude condition has two main implications. First, mechanisms for scaling down the resource and management activities to investigator-specified foci will be valuable. Second, it is likely that there will be conflicts among the various metadata resources corresponding to a given biological datum, and methods for conflict resolution (for example, support for weighted combination of sources leading to a decision) will be useful.

The `AnnBuilder` annotation system-building framework of Bioconductor confronts the evolution, commonality, and magnitude conditions of biological metadata. AnnBuilder is a collection of R functions, scripts in Perl and PostgreSQL, XML Document Type Definitions (DTDs), and documentation to define programming and data structure activities that support creation and curation of general genomic annotation resources.

3.3 Packaging Biological Metadata

In most situations the end-user is analyzing data from a particular instrument or technology. It therefore makes sense to package the data correspondingly. One of the roles of `AnnBuilder` is to facilitate the production of such data packages. We routinely produce and distribute data packages for Affymetrix HGU133a, the 5 segments of HGU95 (a-e), Hu6800, murine MGU74a, and rat RGU34a. Each of these environments maps between chip probe IDs and GenBank accession, gene symbol, gene name, UniGene cluster ID, locusLink ID, chromosome location and orientation, PubMed unique IDs, summary of function, gene ontology ID, enzyme commission identifier, pathway name, and certain reverse mappings (such as pathway name to Affymetrix ID).

Hash tables are also provided for 'chip description files' (CDFs) for the aforementioned Affymetrix chips and some others, include huSNP. These provide detailed geometric information on locations of perfect-match and mismatch sequences for probe sets.

The distribution of these packages is handled by functionality contained in the `reposTools` package. These tools allow the user to perform package management (updating and installation) over the Internet via a specific set of interfaces. The same tools are used to manage the data packages as to manage the standard R analysis packages.

3.4 Utilities for Mining Textual Metadata

The PubMed-related utilities of the `annotate` package provide various paths into the PubMed bibliographic resources related to genomics. Given a vector of PubMed IDs linked to gene products we make use of the Web Services offered by the NLM and obtain the associated data via an `http` request. Subsequently the `pubmed` function will extract the associated XML DOM tree (providing access in R to textual data such as the abstract, the chemicals involved in associated publications) or create a browser page with the associated links.

4 Training and Dissemination

We have already made significant progress in the creation of a training infrastructure promoting the effective use and development of computational and statistical methods in bioinformatics. The materials for these courses are freely

available from our Web site and are updated and extended regularly. Courses can be tailored to two different types of audiences, statisticians that would like to know more about computational biology and biologists who would like to know more about the statistical and computational methodology they are using.

These courses provide a number of benefits for the research being proposed here. These include opportunities for the PI's named in this grant to meet and work together for a few days. They introduce the software and design of Bioconductor to a wide audience. Perhaps most importantly they expose us to a wide variety of users from whom we can learn a great deal. These users provide valuable feedback both on what is working and they also often make feature or task requests. Such feedback is essential to good software design.

One of the most difficult aspects of disseminating this software will be to provide good explanations of how to assemble the data and analyze it with the software. We have begun exploring a novel strategy for developing and distributing these explanations. They take the form of a *how-to* document that we will refer to as a vignette. The document is a mix of code and text (modeled along the lines of literate programming). There are special software tools (already developed) that will automatically run all code in all vignettes and report any problems or errors that occur. This will allow us to ensure that documentation keeps pace with code development.

References

[1] J. C. Boldrick, A. A. Alizadeh, M. Diehn, S. Dudoit, C. L. Liu, C. E. Belcher, D. Botstein, L. M. Staudt, P. O. Brown, and D. A. Relman. Stereotyped and specific gene expression programs in human innate immue responses to bacteria. *Proceedings of the National Academy of Sciences USA*, 99(2):972–977, 2002. 3, 4

[2] M. J. Buckley. *The Spot User's Guide*. CSIRO Mathematical and Information Sciences, North Ryde, NSW, Australia, August 2000. `http://www.cmis.csiro.au/IAP/Spot/spotmanual.htm`. 1

[3] M. J. Callow, S. Dudoit, E. L. Gong, T. P. Speed, and E. M. Rubin. Microarray expression profiling identifies genes with altered expression in HDL deficient mice. *Genome Research*, 10(12):2022–2029, 2000. 3, 4

[4] X. Chen, S. T. Cheung, S. So, S. T. Fan, C. Barry, J. Higgins, K.-M. Lai, J. Ji, S. Dudoit, I. O. L. Ng, M. van de Rijn, D. Botstein, and P. O. Brown. Gene expression patterns in human liver cancers. *Molecular Biology of the Cell*, 13(6):1929–1939, 2002. 3, 4

[5] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979. 1

[6] S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 2002. (Accepted) `http://www.stat.Berkeley.EDU/tech-reports/index.html`. 4

[7] S. Dudoit and J. Fridlyand. Classification in microarray experiments. In T. P. Speed, editor, *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall/CRC, 2002. (To appear). 4

[8] S. Dudoit and J. Fridlyand. A prediction-based resampling method to estimate the number of clusters in a dataset. *Genome Biology*, 3(7):0036.1–0036.21, 2002. 4

[9] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002. 4

[10] S. Dudoit, J. P. Shaffer, and J. C. Boldrick. Multiple hypothesis testing in microarray experiments. Technical Report 110, Division of Biostatistics, University of California, Berkeley, 2002. 4

[11] S. Dudoit 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, editors, *The Analysis of Gene Expression Data: Methods and Software*. Springer, New York, 2003. (To appear). 1

[12] S. Dudoit, Y. H. Yang, M. J. Callow, and T. P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, 12(1):111–139, 2002. 3, 4

[13] M. B. Eisen, P. T. Spellman, P. O. Brownand, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, 95(25):14863–14868, 1998. 4

[14] Y. Ge and S. Dudoit. Fast algorithm for resampling-based *p*-value adjustment in multiple testing. (In preparation), 2002. 4

[15] R. A. Irizarry, 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*, 2003. To appear (http://www.biostat.jhsph.edu/ ririzarr/papers). 2

[16] R. A. Irizarry, G. Parmigiani, M. Guo, T. Dracheva, and J. Jen. A statistical analysis of radiolabeled gene expression data. In *Proceedings of Interface 2001*, pages 1–4, Orange County, CA, 2001. 3

[17] C. Li and W.H. Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Science USA*, 98:31–36, 2001. 2

[18] G. L. Mutter, J. P. A. Baak, J. T. Fitzgerald, R. Gray, D. Neuberg, G. A. Kust, R. Gentleman, S. R. Gullans, L.-J. Wei, and M. Wilcox. Global expression changes in constitutive and hormonally regulated genes during endometrial neoplastic transformation. *Gynecologic Oncology*, 83:177–185, 2001. 3

[19] Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed. Comparison of methods for image analysis on cDNA microarray data. *Journal of Computational and Graphical Statistic*, 11(1):108–136, 2002. 1

[20] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4):e15, 2002. 1

[21] Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed. Normalization for cDNA microarray data. In Michael L. Bittner, Yidong Chen, Andreas N. Dorsel, and Edward R. Dougherty, editors, *Microarrays: Optical Technologies and Informatics*, volume 4266 of *Proceedings of SPIE*, pages 141–152, May 2001. 1