Goober: A fully integrated and user-friendly microarray data management and analysis solution for core labs and bench biologists

Wen Luo[1*], Murali Gudipati[1], Kevin Jung[2], Mao Chen[1], and Keith B. Marschke[1]

[1]Ligand Pharmaceuticals, 10275 Science Center Drive, San Diego, CA 92121

[2]Current Address: Department of Interface/Application, Clinical Services, Cardinal Health, 10020 Pacific Mesa Blvd., San Diego, CA 92121

Summary

Despite the large number of software tools developed to address different areas of microarray data analysis, very few offer an all-in-one solution with little learning curve. For microarray core labs, there are even fewer software packages available to help with their routine but critical tasks, such as data quality control (QC) and inventory management. We have developed a simple-to-use web portal to allow bench biologists to analyze and query complicated microarray data and related biological pathways without prior training. Both experiment-based and gene-based analysis can be easily performed, even for the first-time user, through the intuitive multi-layer design and interactive graphic links. While being friendly to inexperienced users, most parameters in Goober can be easily adjusted via drop-down menus to allow advanced users to tailor their needs and perform more complicated analysis. Moreover, we have integrated graphic pathway analysis into the website to help users examine microarray data within the relevant biological content. Goober also contains features that cover most of the common tasks in microarray core labs, such as real time array QC, data loading, array usage and inventory tracking. Overall, Goober is a complete microarray solution to help biologists instantly discover valuable information from a microarray experiment and enhance the quality and productivity of microarray core labs. The whole package is freely available at http://sourceforge.net/projects/goober. A demo web server is available at http://www.goober-array.org.

1 Introduction

It has been more than a decade since microarray technology was first introduced into biological research [1, 2]. Today, microarray has become a popular tool for applications ranging from basic biomedical research to drug discovery and development, and has made unprecedented successes and profound impacts in these fields. However, due to the vast amount of data generated and the complicated biological annotations associated with the tens of thousands of genes on the arrays, data analysis and bioinformatics remain as the most challenging area in microarray experiments according to a recent survey [3]. This barrier is particularly challenging for bench biologists, most of them do not deal with microarray data on a regular basis and often don't possess adequate programming and statistical skills to data mine large amounts of microarray data. Ironically, the bench biologists are often the ones who need the results the most, since they are the ones conducting experiments to test and prove new findings from microarray data and advance biomedical research. Although a number of commercial or academic software packages are available for microarray data

* To whom correspondence should be addressed: wluo@ligand.com

analysis (http://www.bioinformatics.ca/links_directory/?subcategory_id=101), such as Spotfire, GeneSpring, dCHIP, and RMA [4, 5], most of them require significant training before the users can properly analyze the microarray data on their own. In addition, the significant genes identified from a microarray experiment often need to be associated with the relevant biological pathways, and this task is typically handled by separate software such as GenMAPP, KEGG, Ingenuity Pathway Analysis, and GeneGo [6-8]. To select from the ever increasing microarray software list and to learn how to use them is not only a daunting task for the bench biologist who performed the experiment, but often the software can not be shared so that access to the valuable microarray data is prohibited to their colleagues, advisors or managers who could also benefit from the millions of data points generated from the microarray experiment. In contrast to the large number of software tools available for microarray end users, there are only limited packages build for microarray core labs, the centralized facilities where most of the microarray chips are processed.

To overcome these limitations in microarray data analysis, we have developed a microarray package, Goober, with an intuitive web interface to allow scientists to instantly access their microarray data, and related biological annotations and pathways, with little or no prior training. In addition, we have implemented a LIMS-like package which includes a number of features to facilitate the routine tasks in the microarray core lab, such as real time quality control (QC), array usage and inventory tracking. Since its launch over five years ago, this microarray web portal has proved to be extraordinarily useful for many biologists to perform microarray data analysis, and for microarray core lab members to manage their tasks. In the meantime, this microarray package has been continuously refined based on feedback from bench biologists and microarray core lab members over the last five years.

2 Methods

Most of the web interfaces of Goober package were written in PHP, and MySQL was used to host the relational databases that stored sample annotations, probeset annotations, raw data, processed results, inventory, QC parameters etc. Only the pathway web interface was written in Java. The whole system was run on an Apache server on a Linux machine. Embedded hyperlinks were used extensively throughout the package so the users can easily navigate the web interface without inputting any summarization algorithm, mode of normalization and statistical models which often require prior data analysis knowledge to make such decisions. However, drop-down menus are available in almost all Goober web interfaces, so the advanced users can easily adjust many of the parameters to tailor their specific needs. In addition, most of tasks can be accomplished within two or three mouse clicks. The data flow of the Goober microarray package is illustrated in Figure 1, and the data model and other details of the databases are described in user manual [9].

The back-end relational database was designed to be populated by microarray core members using a web-based graphic user interface (GUI). The upload includes sample annotation and array raw data, and then the raw data can be processed to create pre-calculated results such as fold changes and p-values for downstream data analysis.

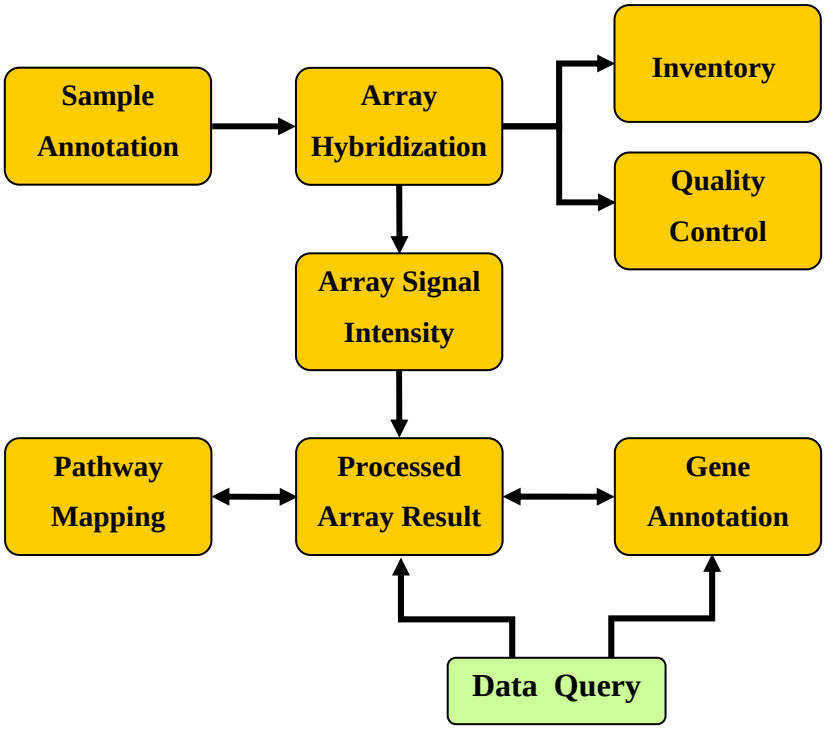

Figure 1: The data flow of Goober microarray package. Each component or module in the package is represented by a square box, and direction of data flow is indicated by the arrow. The data query, which is shown in green box, represents the web interface to access the database.

3 Results

3.1 An Integrative Web Interface for Bench Biologists to Analyze Microarray Data

The primary goal of the Goober web interface is to make microarray data analysis as easy as possible to perform, even by the untrained user, while maintaining the flexibility to conduct more advanced and sophisticated analysis by experienced users. This was accomplished by the extensive use of graphic displays, hyperlinks, drop-down menus and pre-set default parameters. The immense amount of microarray data stored in the database can be accessed by two approaches: the experiment-based query allows the user to perform data analysis on a single microarray experiment; and the gene-based query provides an expression profile for a particular gene in all experiments stored in the database.

3.1.1 Experiment Based Query

For a single microarray experiment, the most common task is to identify the differentially expressed genes and their associated biological pathways. This is probably the most frequently used feature, and this task can be simply completed by a few mouse clicks with Goober. First, the user identifies the microarray experiment to be analyzed. We didn't use a search box requiring keywords to search the database since the first-time user often has no idea which keywords are available in the database. Instead, three drop-down menus were created based on treatment, project/lab name or owner's name, and all experiments matching the selected term(s) will be displayed (Figure 2). A simple mouse click on an experiment will display a list of the differentially expressed genes. In Goober, the end-user doesn't need to perform data normalization, transformation and statistical tests since most of the biologists often are unfamiliar with terms such as RMA, quantile normalization, Z-score, bonferroni correction, and false discovery rate (FDR), and have little idea of their appropriate use. Thus,

we took a simplified approach using MAS 5 data and conducted t-tests based on pre-defined comparison group pairs, and results were pre-loaded onto Goober by the microarray core lab. Goober also uses a default setting for differentially expressed genes based on signal intensity and fold change to eliminate any guessing point for the naïve user. With these approaches, the biologists can immediately access a list of the differentially expressed gene, which is normally the information they care about the most. For advanced users, these pre-set parameters can be easily adjusted using the drill-down boxes provided in the GUI to tailor to their needs. This strategy was used throughout the Goober package as a way to maintain the balance between the ease of use and the flexibility to explore.

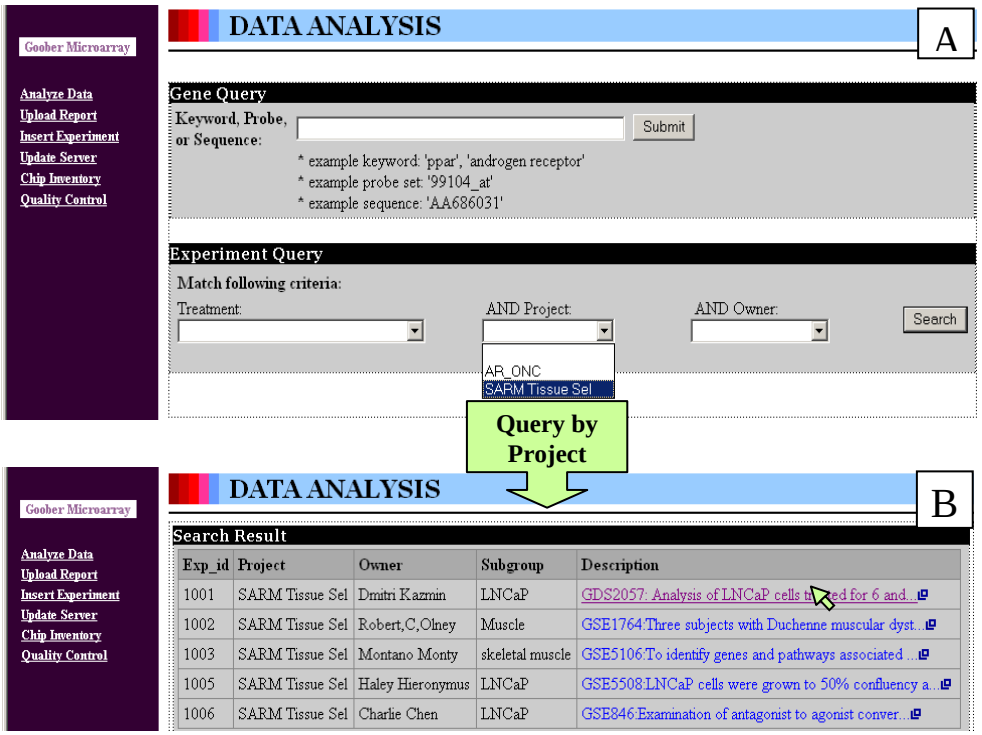

Figure 2: Experiment based query in Goober web interface. (A) The home page of Goober. The bottom area is used for conducting an experiment-based query. (B) The desired experiment one was selected by a mouse click (represented by the green arrow).

The expression intensities for all treatment groups of each probe set are displayed in heatmap-like graphs (Figure 3A), which provide an intuitive way for users to easily identify genes having a particular expression pattern across multiple treatment groups. A bar graph exhibiting raw expression intensity data of any gene/probe set from this list can be displayed by a simple mouse click via the embedded hyperlink (Figure 3D). In most cases, this bar graph actually can be used directly in presentations or reports by copying and pasting the screen display.

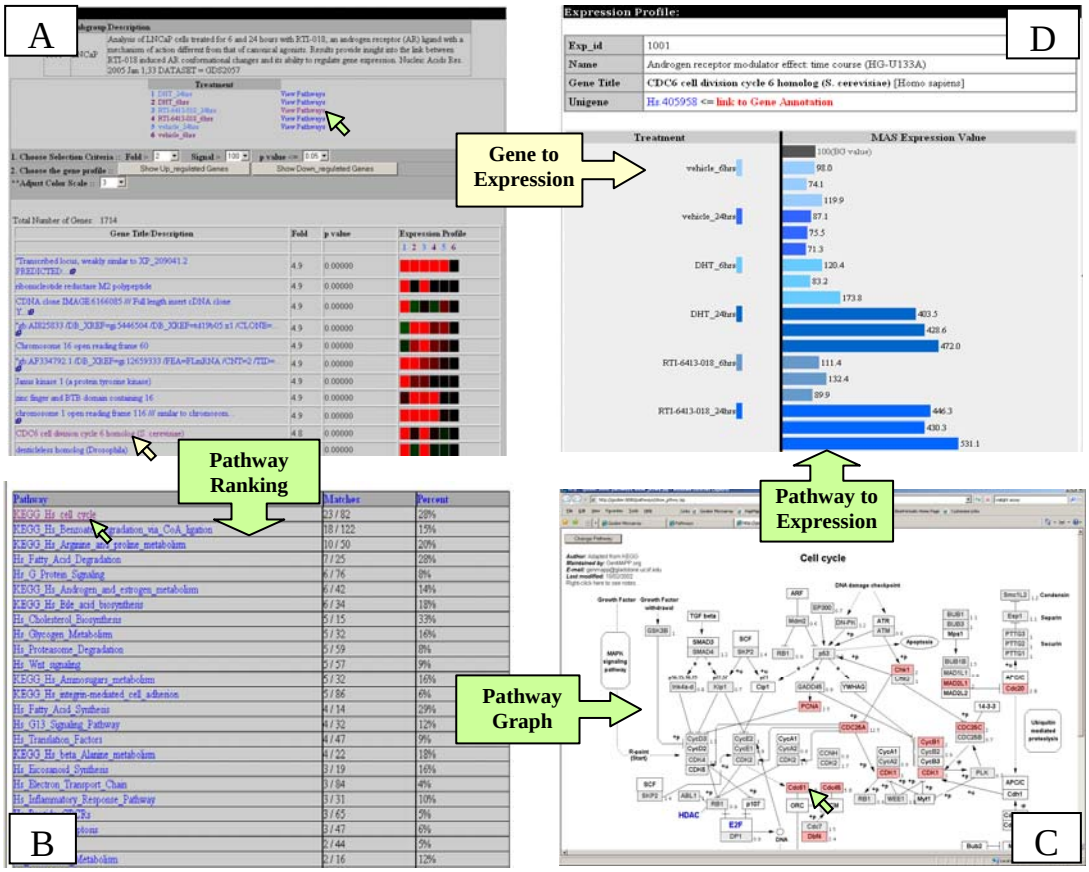

Figure 3: Flowchart of experiment based query in Goober web interface.

In addition to listing hundreds of up and down regulated genes from a microarray experiment, the significant genes often need to be further analyzed under the relevant biological context, such as biological pathways. GenMAPP is one of the first pathway software developed for microarray data analysis [6]. This popular software is very versatile and has great visualization and customization tools. However, like most of the pathway packages, it requires microarray data to be pre-analyzed and re-formatted before it can be uploaded for pathway analysis. To streamline the pathway analysis process, we developed a JAVA-based module so significantly regulated genes identified from a microarray experiment can be automatically mapped to the pathway graphs generated by GenMAPP. As a result, the user can easily start the pathway analysis by clicking the 'view pathway' link next to the treatment group they are interested in, and all of the pathways stored in the database are listed, ranked by either the number or the percentage of significantly regulated genes presented in a particular pathway (Figure 3B). Each pathway in the list is hyperlinked to a GenMAPP-like pathway graph, and the color coding corresponding to the fold change for each gene will be dynamically mapped to the pathway when this graph is displayed (Figure 3C). In addition, each of the mapped genes is linked to the raw expression data display, which can be accessed by a simple mouse-click (Figure 3D).

3.1.2 Gene Based Query

With a growing list of microarray experiments available in our internal database, as well as public databases, biologists often not only wish to examine the expression profile of a gene in their own microarray experiment, but also investigate the expression patterns in other experiments using different tissues and/or under different treatments. This exercise, like searching the gene expression encyclopedia, can help a scientist significantly broaden his/her

understanding or gain additional insight into the function of the genes of interest. Therefore, we have built a gene based query to allow users to access the expression profile of a single gene or probe set across all the microarray experiments stored in Goober. The gene or probe set can be retrieved by entering the gene name, sequence ID, or Affymetrix probe set id (Figure 4A), and Goober will return all matching probe sets or genes. Expression profiles of the selected probe set in all microarray experiments are displayed in heatmap-like graphs (Figure 4B). The significantly regulated expression is shown in red (up-regulated) or green (down-regulated), and the un-regulated expression in black. Compared to other data visualization methods, this type of graphic illustration allows users to easily spot the experiments in which a gene of interest is regulated across a large number of experiments. Similar to the experiment based query, there is a hyperlink embedded in the heatmap of every experiment, so the raw expression data can be accessed by a mouse click (Figure 4C).

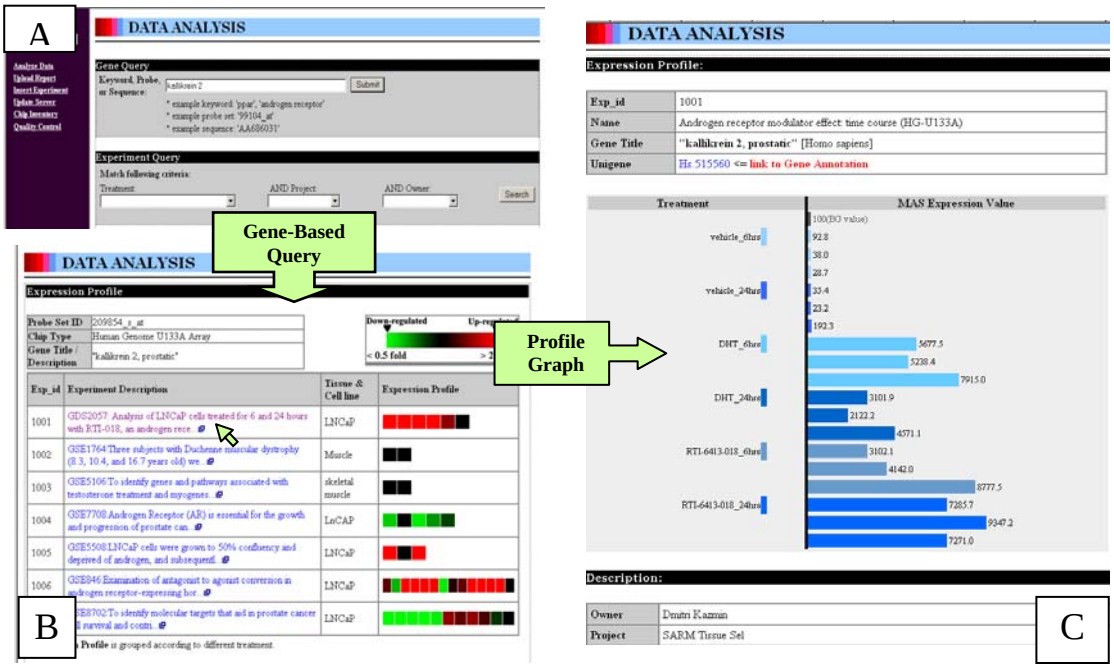

Figure 4: Overview of gene based query in Goober web interface.

3.2 Web Interface for Microarray Core Lab

Nowadays, most of the microarray samples are processed by microarray core labs, which are centralized core facilities in many university campuses, research institutes, and biopharmaceutical companies. However, many routine tasks in the core lab, such as tracking the array usage, updating array inventory and performance of timely QC, are not adequately addressed by most microarray software. Therefore, we have implemented a number of features in Goober to support these common tasks in core labs.

3.2.1 Graphical Array QC - Real Time

Microarray has become a very mature technology in many aspects, including hardware, software, and reagents. In addition, many core labs have years of operating experience, and the success rate has steadily improved. However, failed arrays can still be encountered on any given day for a variety of reasons: poor sample quality, expired reagents, or bad chips. Therefore, QC parameters for all arrays must be thoroughly inspected before any data can be processed. The text report files generated by Affymetrix software are the most common

source for examining the quality of arrays. Scanning through more than a dozen QC parameters in various formats and ranges for each chip is very tedious and error-prone, especially for experiments with a large number of arrays (the new Expression Console™ has made significant improvement on this regard). In addition, the QC is often not performed until all arrays have been processed, which might be too late in some cases. For instance, a whole experiment could be ruined if there is something wrong with the staining or washing buffer. Therefore, we have developed a graphic-based QC monitor in Goober, which can display all essential QC parameters in real time. Any parameter that fails to meet the user-defined criteria is highlighted in red, which can be easily spotted from the green background used to represent parameters meeting the QC criteria (Figure 5A). The raw data of the listed parameters can be easily accessed by a mouse click for further inspection (Figure 5B). All of these procedures require no additional data entry from the operator, since the QC parameters are automatically retrieved from the report file by a parser written in PHP.

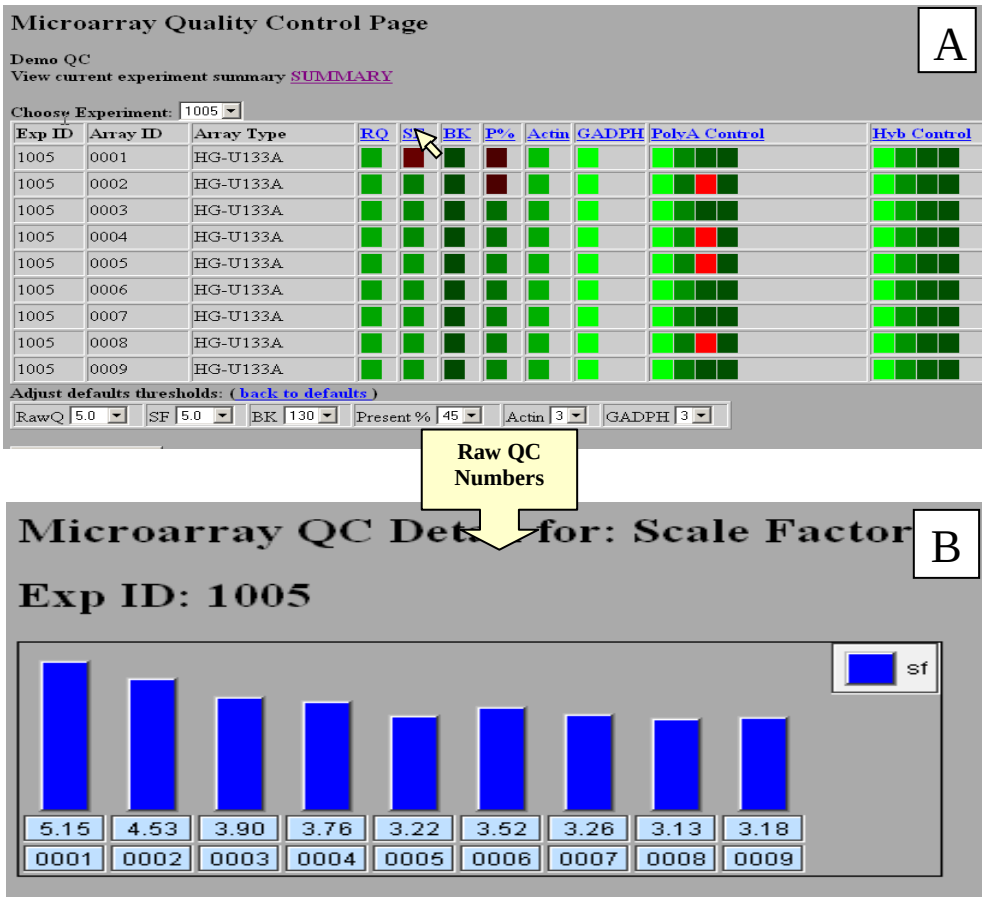

Figure 5: Flowchart of Goober web interface for real-time QC.

3.2.2 Inventory Tracking

Most microarray core labs need to process hundreds or thousands of samples for different labs, projects, or departments. It is essential for core labs to keep track of the usage of these different entities for budgetary or monetary purposes, and an efficient tracking system is also important to insure the core labs have adequate supplies of arrays for the upcoming microarray experiments. All these tasks can be accomplished in Goober with little user intervention, and the database can be updated using a parser in Goober to retrieve the related information stored in the experiment and report.

3.2.3 Experiment Annotation and Data Loading

Goober's experiment annotation is developed based on MIAME (Minimum Information About a Microarray Experiment). The general experiment description can be entered into the system using the web-interface. However, a web-interface might not be the most efficient way to enter the annotation for each sample, especially for an experiment containing a large number of arrays, since there are more than a dozen fields to be filled for every sample, and many of the annotations are redundant. We have developed an Excel template embedded with visual basic macros to facilitate the sample annotation, so simple Excel functions like copy and paste can be used when needed. Then, the entire annotation file can be uploaded to Goober in a split second instead of entering the annotations one-by-one. Expression data loading is also straight forward, and only requires minor reformatting of the pivot Excel data file generated by Affymetrix Software.

4 Discussion

The intent of the Goober microarray package is to simplify the complicated microarray data analysis workflow so bench biologists can easily navigate huge volumes of microarray data to discover new biology without spending too much time on learning new software and statistics. The first version, which was released 6 years ago, allowed end users to query expression of a single gene across all microarray experiments. Based on many valuable feedbacks from the end users, ranking of the differentially expressed genes and pathway analysis were subsequently incorporated. In addition, a wide range of LIMS like tools were implemented for the microarray core lab. Since its launch, Goober has been used by many scientists at Ligand Pharmaceuticals from various disciplines and backgrounds, and proved to be a very robust and useful tool to advance their discovery research. Most of the users could freely navigate the sea of microarray data on Goober with no prior training or with only a few minutes of instruction. As the result, Goober has been accessed more than 7,000 times by about 30 distinct users since its inception.

Among the overwhelmingly long list of microarray software available today, the Goober package is perhaps one of the easiest to use for biologists. For instance, to perform pathway analysis with differentially regulated genes, GenMAPP is one of the earliest and most popular pathway analysis software dedicated for microarray data. However, it often takes hours to learn the software, format the data and load data properly into the software. Goober pathway analysis was built based upon the GenMAPP format. Although Goober can't perform the more complicated tasks GenMAPP does, such as multiple-group comparisons and displays, the pathway ranking and pathway display, with overlaid expression data, have been automated, and they can be easily accessed by a simple mouse. Since we built Goober, a number of web-based pathway tools, such as FatiGo, GEPAT, and KEGGanim [10-12], which also can streamline the microarray pathway analysis, have been published. But most of them are unable to retrieve raw expression data from the pathway display directly as Goober does, a feature that bench biologists have found very useful.

Goober is a hybrid between a data analysis tool and a microarray database. With a database-like approach, microarray data and experimental results have been pre-loaded, so the users don't have to spend time on formatting and uploading the data. It also allows users to query the expression of their interested genes across multiple experiments. Compared to other microarray database packages, such as GEO, Goober offers easier and broader data analysis tools. For instance, to generate a differentially expressed gene list, GEO performs the calculation at run-time taking at least several minutes. Goober meanwhile can display the

similar gene list in seconds because the results which were pre-calculated during data loading have already been stored.

Another feature that differentiates Goober from most of the microarray software packages is the inclusion of a LIMS-like package, which can support a number of common tasks in the core lab such as array QC and inventory tracking. Compared to some "heavy" LIMS packages such as MARS and GeneDirector[13], Goober is a "light" version, which does include some of the routine tasks that are not covered by others, such as array inventory tracking. Array QC is probably the most important task for a microarray core lab, and we implemented a heatmap-like display to track array QC. This feature, which can be used in real-time during array scanning, makes it easier for array operators to catch the problematic arrays compared to reading the text files generated by the Affymetrix software.

Obviously, as a trade off of being easy to use, many alternative normalization methods and advanced data analysis approaches, such as hierarchical clustering, k-mean clustering, ANOVA and FDR, can not be performed on Goober. Thus, Goober is by no means intended to replace other advanced microarray analysis tools such as GESA, Onto-Tools, DAVID, SAM, dCHIP, Spotfire and GeneSpring etc. In fact, Goober can be complimentary to those data analysis tools, and we have found tremendous synergy when they were used together. For instance, the expression data display in Goober can be easily connected to Spotfire via its web link function.

Finally, Goober microarray package was developed based on Affymetrix GeneChip™, which is probably the most poplar microarray platform [3]. However, the core analysis in Goober is based on signal ratio or fold change. Therefore, it can be easily modified to accommodate other popular microarray platforms, such as two-color arrays.

5 Availability and Requirements

Project Name: Goober

Project Home Page: http://sourceforge.net/projects/goober

Documentation: Goober User Manual [9]

Operating Systems: Linux

Web browser: Internet Explorer or Mozilla Firefox

Programming language: PHP, Perl, and Java

Other requirements: MySQL and Apache Tomcat

Demo Web Page: http://www.goober-array.org (Please note that the package is designed for local installation, and some of features might not be available on the demo server)

6 Authors' contributions

WL designed the database and the architecture of the package and wrote the manuscript. MG, KJ and MC wrote the scripts and implemented the database. KM provided various support and advice to the project, and revised the manuscript. All authors read and approved the final manuscript.

7 Acknowledgements

We would like to thank Greg Brown for his help on the pathway programming. We are also in debt to many scientists at Ligand Pharmaceuticals for their valuable support and feedback.

8 References

1. Lockhart DJ, Dong H, Byrne MC, Follettie MT, Gallo MV, Chee MS, Mittmann M, Wang C, Kobayashi M, Horton H *et al*: **Expression monitoring by hybridization to high-density oligonucleotide arrays**. *Nat Biotechnol* 1996, **14**(13):1675-1680.

2. Schena M, Shalon D, Davis RW, Brown PO: **Quantitative monitoring of gene expression patterns with a complementary DNA microarray**. *Science* 1995, **270**(5235):467-470.

3. Harrington C, Hester S, Auer H, Jafari N, Potter S, Tiesman J, Jensen R, Reid L, Massimi A, Denslow N: **The ABRF MARG Microarray Survey 2008: Sensing the State of Microarray Technology**. *2008 Microarray Research Group Study* 2008.

4. Irizarry RA, Hobbs B, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, Speed TP: **Exploration, normalization, and summaries of high density oligonucleotide array probe level data**. *Biostatistics* 2003, **4**(2):249-264.

5. Li C, Wong WH: **Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection**. *Proc Natl Acad Sci U S A* 2001, **98**(1):31-36.

6. Dahlquist KD, Salomonis N, Vranizan K, Lawlor SC, Conklin BR: **GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways**. *Nat Genet* 2002, **31**(1):19-20.

7. Kanehisa M, Goto S, Kawashima S, Okuno Y, Hattori M: **The KEGG resource for deciphering the genome**. *Nucleic Acids Res* 2004, **32**(Database issue):D277-280.

8. Salomonis N, Hanspers K, Zambon AC, Vranizan K, Lawlor SC, Dahlquist KD, Doniger SW, Stuart J, Conklin BR, Pico AR: **GenMAPP 2: new features and resources for pathway analysis**. *BMC Bioinformatics* 2007, **8**:217.

9. Luo W, Gudipati M, Jung K, Chen M, Marschke K: **Goober microarray package user manual**. 2008.

10. Adler P, Reimand J, Janes J, Kolde R, Peterson H, Vilo J: **KEGGanim: pathway animations for high-throughput data**. *Bioinformatics* 2008, **24**(4):588-590.

11. Al-Shahrour F, Minguez P, Tarraga J, Medina I, Alloza E, Montaner D, Dopazo J: **FatiGO +: a functional profiling tool for genomic data. Integration of functional annotation, regulatory motifs and interaction data with microarray experiments**. *Nucleic Acids Res* 2007, **35**(Web Server issue):W91-96.

12. Weniger M, Engelmann JC, Schultz J: **Genome Expression Pathway Analysis Tool--analysis and visualization of microarray gene expression data under genomic, proteomic and metabolic context**. *BMC Bioinformatics* 2007, **8**:179.

13. Maurer M, Molidor R, Sturn A, Hartler J, Hackl H, Stocker G, Prokesch A, Scheideler M, Trajanoski Z: **MARS: microarray analysis, retrieval, and storage system**. *BMC Bioinformatics* 2005, **6**:101.