DNA array analysis in a windows-PC environment

Tyrrell Conway*, Barbara Kraus, Don L. Tucker, Darren J. Smalley, Angela F. Dorman, and Loretta McKibben

Running Title: Array analysis

Department of Botany and Microbiology, The University of Oklahoma, Norman, OK 73069-0245

Keywords: functional genomics, DNA array, computational biology, bioinformatics

* Corresponding author: Tyrrell Conway
  Department of Botany and Microbiology
  The University of Oklahoma
  Norman, OK 73069-0245
  Voice: 405 325 1683
  FAX: 405 325 7619
  E-mail: tconway@ou.edu
ABSTRACT

Windows-based PC computers have evolved to the point that they provide sufficient computational and visualization power for robust analysis of DNA array data. In fact, smaller laboratories might prefer to carry out some or all of their analyses and visualization in a Windows environment, rather than alternative platforms such as UNIX. We have developed a series of manually-executed macros, written in Visual Basic for Microsoft Excel spreadsheets, that allows for rapid and comprehensive gene expression data analysis. The first macro assigns gene names to spots on the DNA array and normalizes individual hybridizations by expressing the signal intensity for each gene as a percentage of the sum of all gene-intensities. The second macro streamlines statistical consideration of the confidence in individual gene measurements for sets of experimental replicates by calculating probability values with the student t-test. The third macro introduces a threshold value, calculates expression ratios between experimental conditions, and calculates the standard deviation of the mean of the log ratio values. Selected columns of data are copied by a fourth macro to create a processed data set suitable for entry into a Microsoft Access database. An Access database structure is described that allows simple queries across multiple experiments and export of data into third-party data visualization software packages. These analysis tools can be used in their present form by others working with commercial *E. coli* membrane arrays or they may be adapted for use with other systems. The Excel spreadsheets with embedded Visual Basic macros and detailed instructions for their use are available on the Internet: [http://www.ou.edu/microarray](http://www.ou.edu/microarray).

INTRODUCTION

With the publication of the first genome-wide expression data 6 years ago (13) the power of the DNA array was immediately evident. Since then, reviewers have predicted a revolution in the way biologists conduct their research (3-5, 12). Today the trend toward wide-spread use of DNA arrays continues. Vast amounts of DNA array data are accumulating and the need for standardized array annotation and data representation is being addressed (http://www.ebi.ac.uk/microarray/MGED/). Relational databases are being developed to handle microarray data storage in a format that facilitates data processing and visualization, allowing researchers to analyze and interpret their experiments, and disseminate the data (16). Generally, the database management software used is Oracle, web interfaces are written in JAVA or XML, and scripts used for data processing, retrieval, etc. are written in Perl, C, or other programming languages. For the common microarray user however, implementation of such a system may be beyond their budget or exceed their actual needs. For these users, a central archive is an attractive alternative for data storage. An example of this, the EcoReg Consortium (http://gobi.lbl.gov/~ecoreg/index.html), is being established as a public database for storage and manipulation of *Escherichia coli* microarray and proteome data. Still, it will be necessary for individual users of consortium databases to process, analyze, and format the data for submission.

DNA arrays have been used to examine the genetics and physiology of the comprehensive biological model, *E. coli*. Some research groups have used commercial membrane-based DNA arrays (2, 17, 18), while others have employed DNA microarrays (6, 11, 19, 20) or oligonucleotide arrays (15). Several practical issues regarding the use of whole-genome arrays have been addressed and the power of this technology as a means for deducing the physiological state of the bacterial cell is now well established.

In our laboratory we routinely use membrane-based DNA arrays for *E. coli* gene
expression profiling. Multiple replicates of each experimental condition are processed and analyzed with manually-executed macros written in Visual Basic and run in Microsoft Excel spreadsheets on Windows-PC computers. These macros, statistical analysis, and data processing protocols are described here and are freely available to the scientific community (http://www.ou.edu/microarray).

MATERIALS AND METHODS
Software and system requirements
The analysis tools described here are written for Microsoft Excel and MS Access: MSOffice 97 or higher is required. Optimal system requirements include a Windows PC computer with a PII processor and 128 Mb of RAM, or higher. MacIntosh computers running MSOffice 98 for MacIntosh can also be used to run the macros in Excel.

Example E. coli data set
In this study we compare a sample data set of E. coli MG1655 grown at pH 7.4 and pH 5.5, under otherwise identical conditions. Cultures were grown aerobically in 50 ml of MOPS (pH 7.4) or MES (pH 5.5) minimal glucose (0.2%) medium (8) in 250 ml flasks (Corning, Acton, MA) at 37°C with 300 rpm agitation, and harvested in mid-logarithmic growth phase. RNA isolation, radioactive labeling during cDNA synthesis, and hybridization to DNA array membranes was described previously (17).

Description of raw data
We routinely use Panorama E. coli Gene Arrays™ (Sigma Genosys Biotechnologies, Inc., The Woodlands, TX) for gene expression profiling. Phosphorimaging of a hybridized membrane array produces a TIFF image file that must be further processed for data analysis. The image analysis software (ArrayVision™ ver 5.1, Imaging Research, Inc., St. Catharines, Ont., Canada) makes use of a customized template to accommodate three grid layers (3x1; 16x24; 4x4) according to the design of the Panorama E. coli gene arrays. The spot labeling protocol was edited such that each spot is named by its unique array coordinate, allowing the spot intensity measurements to be easily associated with the correct gene identifiers in subsequent processing steps. The customized ArrayVision template file for analysis of Panorama E. coli gene arrays, and detailed instructions for its use, are available on our website. The spot intensities are represented in a row-column format and are exported into Excel spreadsheets for further analysis.

Data processing
The macros and sample analyses can be downloaded from our web site: (http://www.ou.edu/microarray). Follow the links to "Macroarray", "Data Analysis", and then "Spot-finding and Image Quantitation" or "E. coli Data Analysis (software downloads)"; alternatively the macroarray section of the website can be accessed directly: (http://www.ou.edu/microarray/macroarray.htm). Raw DNA array data, exported from ArrayVision, are processed in a series of 3 Excel workbooks that are used to manually execute 4 macros written in Visual Basic (Table 1). These macros are designed to filter the data and calculate statistics to allow for further data analysis and interpretation. Detailed, step-by-step instructions for use of these analysis tools are provided on our website.
RESULTS AND DISCUSSION

The workbooks, macros, and subroutines used for DNA array data analysis are outlined in Table 1. The subroutines can be run in order individually, or the macro containing all relevant subroutines can be run once to execute all subroutines. In the following section we provide an overview of important statistical considerations, the specific processing steps, and outcomes.

**Statistical significance**

Various approaches for attaching significance to DNA array data have been published, including a simple “rule of thumb” criterion for the value of the expression ratio (2, 20). Some researchers have used the standard deviation from the mean of the expression ratios as an indicator of confidence (6, 17). Arfin et al (1) applied the student t-test to experimental replicates and considered the P value to be the most important indicator of significance. Richmond et al (11) combined a confidence interval obtained with the student t-test and a rule of thumb criterion for the expression ratio. Others prefer to consider the significance of a single experimental condition based on the coefficient of variation (18). A precedent for statistical analysis of array data has yet to be firmly established and a standard is clearly needed. Whatever statistical approach is adopted, it is essential that DNA array experiments are properly replicated and the uncertainty that lies behind individual gene measurements be considered in order to attach significance to data sets.

We advocate the use of at least 2 replicates of each experimental condition. Membrane arrays typically have duplicate spots for each gene and each spot is considered to be a separate determination, providing a total of 4 determinations for the 2 replicates. Because membrane arrays are hybridized with a single labeled target mRNA and normalized independently (in effect a one-color experiment) the variation in the measurement is at the level of the raw data, not the measurement of the ratio (unlike the statistical approach that is popular with two-color microarrays and involves internal normalization of the measurements). The uncertainty that lies behind individual gene measurements can be variously calculated as the standard deviation of the determinations or the coefficient of variation. Since we are usually interested in the statistical significance of differences between an experimental condition and a control, we prefer the student t-test as a means for calculating this probability, based on the uncertainty of the replicate measurements in both conditions. The student t-test is best applied to natural log-transformed normalized data sets (7). Generally, a P value of <0.05 is chosen to indicate a 95% probability that the difference in gene expression between conditions is significant. However, it has been pointed out that with very large data sets – eg. a bacterium with 5000 genes – choosing a value of p<0.05 means that there could be up to 250 false positives in the data set (1). Thus the researcher is left with two choices: to lower the P value to a level where no false positives are expected (p<0.0002), or to consider a second statistical metric that when combined with a reasonable P value (p<0.05) is an excellent indicator of significance of a ratio value.

We use the standard deviation of the mean of the log ratios – within the context of the P value – to indicate significant up- or down-regulation of gene expression. This approach is meaningful where the expression level of the majority of genes does not change significantly between conditions and where the researcher is interested in genes that show substantially different expression. The standard deviation for the log ratios is calculated and only those genes that differ by more than 3 standard deviations (99.9% confidence in each tail) from the mean of the log ratio (usually zero, or no change) are considered. In practice, emphasis is placed on those genes that have expression ratios greater than 3 standard deviations from the mean and have a reasonable probability of being significantly different between the conditions, based on a P value
<0.05. Where there are 4 or more determinations for each gene, the P value can be lowered to
<0.005 with little change seen in the number of genes that are considered to vary significantly
between conditions.

There may also be situations when the researcher is interested in changes in gene
expression that are not a full 3 standard deviations from the mean, but are still significant -- that
is, where the differences in gene expression between conditions are subtle, yet meaningful. In
this case the student t-test can be used as the sole measure of significance, but the P value must
be adjusted to ensure that false positives are avoided. One approach for this is to apply the
Bonferroni correction that describes a P value for significance in a large data set (7). Various
strategies for implementing this correction factor have been described and its proper use is
somewhat controversial (14). The Bonferroni correction effectively lowers the P value to a point
where false positives and consideration is given only to those genes for which there
is a high degree of confidence in the ratio value. In this light the Bonferroni correction seems to
be a reasonable statistical tool, but may be too stringent for some considerations.

“Image data cruncher” workbook

To begin the data analysis process, the raw data from an experimental replicate is copied
and pasted into a blank spreadsheet named “arvdata” in an Excel workbook named, “Image Data
Cruncher” containing the macro named “AllDataCrunched6” (Fig. 1). The first subroutine in the
macro, “ArvAllSort1”, copies and pastes subsets from the “arvdata” spreadsheet into a second
spreadsheet named “allfields” that contains information provided by the membrane manufacturer
for associating the array location with a spot number that is unique to each target on the array.
The second subroutine, “Nameall2” copies and pastes the data in the “allfields” spreadsheet into
a third spreadsheet that associates the spot number with a unique identifier and associated
genome annotation information for each gene.

Differences in spot intensities between replicate experiments arise from normal
experimental variation, such as differences in growth conditions, radioactive nucleotide
incorporation efficiency, hybridization conditions, or image acquisition. In order to compare
experimental replicates (separate cultures) or technical replicates (same culture and same RNA
sample) the data from each array must be normalized. The third subroutine, “CalcPct3”,
normalizes arrays; if the data are not normalized, the values for replicate experiments, when
plotted, will not pass through zero or be directly proportional (Fig. 2). Array experiments can be
normalized by expressing each gene-specific spot relative to an internal control, if a suitable set
of control spots is present on the arrays. Unfortunately, the intensities of the genomic DNA
control spots on the Panorama E. coli membranes vary significantly (data not shown) and are not
reliable for normalization. In the absence of an internal standard, the preferred approach for
normalization is to express each gene-specific spot as a fraction of the sum of all gene spots (n =
4290), a strategy that at once considers all variables that lie behind the array image (17, 18). The
“CalcPct3” subroutine normalizes the entire data set, expressing each spot as a percentage of the
sum of all spots on the array.

The fourth subroutine, “Cleanup4”, reorganizes the data to facilitate subsequent data
processing steps. The fifth subroutine, “Statistics5” calculates the averages and statistics for the
blank spots (empty spots between genomic DNA control spots) and null spots (empty spots
within array). However, we have not found these values to offer a reliable means for
establishing background on the array or for empirical determination of a threshold value and
therefore do not use them (see below). The “AllDataCrunched6” macro in “Image Data
Cruncher” runs all 5 subroutines at once and results in a file that contains the raw data,
normalized data, and associated genome annotation information for each gene on the array.

**“2-Replicate-Stats” workbook**

The second workbook, “2-Replicate-Stats”, contains a macro named “AllAnalysis8” that is used to calculate the probability that the average of the experimental (test) replicates is significantly different from the average of the control replicates (Fig. 3). The four replicate data sets are sequentially copied from the crunched data files, beginning with the first and then the second replicate of the control, followed by the first and then the second replicate of the experimental, and pasted into the "Enter Data (2 Replicates)" worksheet. The first subroutine, “OrganizebySpotNo1”, sorts the 4 data sets by spot number, which aligns the gene-specific data in rows. The second subroutine, “CalculateAverages2” calculates the mean of the normalized (pct) values for the 4 spot intensities from each experimental condition (2 spots for each gene per membrane). The subroutine, “copyvaluesintoPRaw3” copies the pct values into a separate spreadsheet and the “CalculateLn4” subroutine transforms the raw pct values by the natural log while copying them to an additional spreadsheet.

The subroutines, “CalculatePRaw5” and CalculatePLn6” are used to calculate the P values for the raw and log transformed data, respectively, by application of the student t-test to the 4 determinations for each of the control and experimental conditions. The last subroutine, “CopyAllValues7” reorganizes the data by pasting the data columns to be used in subsequent steps into a separate spreadsheet named “All Values”. Once the “AllAnalysis8” macro is executed the “2-Replicate-Stats” workbook serves as an archive for the raw and normalized data from the replicates being compared and contains the P values that are associated with the ratio calculations in the “Data Analysis” workbook.

**“Data Analysis” workbook**

The third workbook, “Data Analysis” contains two macros. The first macro, “AllAnalysis6” is used to calculate the log ratio of the expression levels in the experimental vs. the control condition (Fig. 4). The average normalized data from the “All Values” spreadsheet in the “2-Replicate-Stats” workbook are pasted into the “crunched data” spreadsheet in the “Data Analysis” workbook. The first subroutine, “SpotSort1”, sorts the data by spot number such that the gene specific data are aligned in rows. The second subroutine, “DataSort2” copies the data and pastes it into the “DataAnalysis” spreadsheet to be used for ratio calculations.

The third subroutine, “ThresholdRatios3” is used to determine a threshold value for ratio calculation, and calculates the absolute value of the ratio of the Test/Control such that genes that are more highly expressed in the test condition are given a positive value and genes that are more highly expressed in the control are given a negative value. The threshold value is chosen to represent the limit of detection of an expressed gene -- that is, the signal intensity at which spots are considered to be significantly higher than the array background. Any spot intensity that falls below the threshold value is raised to that value in order to obtain a reasonable ratio in cases where a gene is expressed below the threshold value in at least one of the two experimental conditions. Ideally, the threshold would be determined independently for each gene, based on the local spot background and the known cross-hybridization to other expressed genes in the sample, but this is not possible because of the dense packing of some membrane arrays and the lack of prior knowledge as to the number and extent of gene expression in a given growth condition. These factors make determination of the threshold value difficult. We have chosen a conservative approximation of the threshold value corresponding to the 500th lowest expressed gene based on the average of the normalized expression levels in the two conditions. This threshold value is similar to that obtained by visual inspection of array images to determine the
faintest of gene-specific spots and is reasonable in light of the predicted number of expressed
genes based on the number of mRNA species in the E. coli cell and the arrangement of genes in
operons (1380 mRNAs x avg. 2.5 genes per operon = 3450 expressed genes) (9). Others have
calculated threshold values as 3 standard deviations above the mean of “blank” spots (10) which
corresponds to ~214 in the example data set used in this study. If desired, researchers can write
alternative approaches for threshold determination into the “ThresholdRatios3” subroutine, or the
threshold level can be edited as described on the web site.

The fourth subroutine, “Cleanup4”, reorganizes the data and calculates the log (base 10)
of the expression ratio. The fifth subroutine, “Stats5”, calculates the standard deviation of the
mean of the log ratios. The “AllAnalysis6” subroutine executes all 5 subroutines and results in a
“DataAnalysis” spreadsheet that contains the averaged raw and normalized data, corresponding
genome annotation information for each gene on the array, and the ratios. The “AllAnalysis6"
macro concludes with creation of two empty columns that are used for manual pasting of the P
value associated with each ratio calculation from the “2-Replicate-Stats” workbook. Finally, the
“MakeDB” macro in the “Data Analysis” workbook can be executed to reorganize and paste key
data columns into a separate spreadsheet that can be used for data entry into a suitable database
(Fig. 5).

Database considerations

It is useful to build a database when multiple time points or multiple conditions are being
compared. A sample MS Access database called “sampleDB” can be downloaded from our web
site. There are 2 tables in the sample database, one populated with annotation information such
as b#, array coordinate, gene, gene product, functional groupings, and accession numbers, and
the other table populated with data generated by the “MakeDB” macro in the “Data Analysis”
workbook. Additional tables can be created and populated by copying the table (structure only),
editing the design as necessary, and pasting the appropriate data from the spreadsheets created by
“MakeDB”. Queries are designed by linking tables by b#, and can be run by manipulation of the
parameters in query design. The query data can be copied or exported from Access into third-
party presentation software packages. There are limitations on the total number of tables and
data elements supported by Access that make it impossible to query more than 20 experiments.
The macros and Excel workbooks described here are also suitable for processing data for entry
into more robust database programs such as MYSQL or Oracle.

Modification and adaptation of VB macros

If desired, the macros described above can be modified in any number of ways to
streamline the process or to tweak the statistical parameters. This is most easily accomplished
using the Microsoft Visual Basic Editor within the Excel workbook environment. For example,
the threshold value can be adjusted by simply changing the “cell” location that is used for the
ratio calculation, or the 4 macros described here could be combined in a single workbook
containing several spreadsheets. (We have chosen not to do so because this would create a very
large, complicated workbook.) More importantly, these macros can be adapted for use with
membrane systems other than the one described here. An example of this is the creation of an
analogous set of macros for processing of data from Clontech Atlas Mouse 1.2 arrays. These
macros and associated Excel workbooks are also publicly available on our website.

Website and public access to analysis tools

A website has been created for users to download these analysis tools and protocols
(http://www.ou.edu/microarray). The protocols on the website describe, in a simplistic way, the
step-by-step implementation of the analysis tools.
Benefits of semi-automated data processing

Our earliest protocol for membrane array data acquisition and analysis required more than 10 man-hours for each experimental replicate (17). Thus it was important to automate the data analysis process to the greatest extent possible. By taking advantage of improvements in commercially available software, and by writing a series of macros in Visual Basic for semi-automated data processing in Microsoft Excel, the time was shortened to 5 minutes per experimental replicate. In addition, it is critical that the possibility of human error be eliminated when carrying out a large number of manipulations of massive data sets such as those involved in gene expression profiling. The macros described here effectively minimize the level of manual data processing by automating much of the process. The integration of 4 macros in 3 workbooks provides for a significant level of human supervision of an otherwise error-free, automated process.

ACKNOWLEDGEMENTS
The authors wish to thank Bill Cuevas and Simon Sims for information prior to publication and helpful discussions. This work was supported by grants from the NIH, RO1 AI48945-01 and RR-01-005.
REFERENCES


Figure 2. Scatter plot view of normalized (top) and raw (bottom) data showing linearity of experimental replicates for normalized data.
Table 1. Useful purpose of workbooks, Visual Basic macros, and subroutines used in this study.

<table>
<thead>
<tr>
<th>Workbook</th>
<th>Macro</th>
<th>Subroutine</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Data Cruncher</td>
<td>AllDataCrunched6</td>
<td>ArvAllSort1</td>
<td>associates array coordinate with spot number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nameall2</td>
<td>associates spot number with unique identifier for gene</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CalcPct3</td>
<td>normalizes data by expressing each spot as percentage of sum of all spot intensities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cleanup4</td>
<td>reorganizes data and calculates avg values for duplicate spots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Statistics5</td>
<td>calculates averages of genomic DNA controls and blank spots</td>
</tr>
<tr>
<td>2-replicate-stats</td>
<td>AllAnalysis8</td>
<td>OrganizebySpotNo1</td>
<td>sorts each of the 4 data sets individually by spot number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CalculateAverages2</td>
<td>calculates averages of volumes and pct values for the control and test replicates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>copyvaluesintoPRaw3</td>
<td>copies and pastes pct values into a separate spreadsheet for calculation of p values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CalculateLn4</td>
<td>copies pct values into a separate spreadsheet and natural log transforms data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CalculatePRaw5</td>
<td>calculates the P value for the raw data by application of the student t-test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CalculatePLn6</td>
<td>calculates the P value for the log transformed data by application of the student t-test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CopyAllValues7</td>
<td>copies and pastes data used for ratio calculations into separate spreadsheet</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>AllAnalysis6</td>
<td>SpotSort1</td>
<td>sorts control and test data sets by spot number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataSort2</td>
<td>copies data set into spreadsheet used for ratio calculations, sorts by total pct value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ThresholdRatios3</td>
<td>calculates ratio of Test/Control using threshold of tot pct value for 500th lowest gene</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cleanup4</td>
<td>reorganizes data and calculates log (10) of ratio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats5</td>
<td>calculates standard deviation of log ratio values and correlation between Test and Control</td>
</tr>
<tr>
<td>(Manual Step)</td>
<td></td>
<td></td>
<td>copy and paste special values Praw and PLn values from 2-replicate-stats to Data Analysis</td>
</tr>
<tr>
<td></td>
<td>MakeDBsheet</td>
<td></td>
<td>reorganizes data and copies into separate spreadsheet for entry into Access database</td>
</tr>
</tbody>
</table>