BIOINFORMATICS APPLICATIONS NOTE

Vol. 19 no. 14, pages 1846–1848
DOI: 10.1093/bioinformatics/btg253

# A tool-kit for cDNA microarray and promoter analysis

N. H. Shah[1,*], D. C. King[1], P. N. Shah[2] and N. V. Fedoroff[1]

[1]The Huck Institute of Life Sciences and [2]The Department of Biology, Pennsylvania State University, University Park, PA 16802, USA

Received on February 10, 2003; revised on April 1, 2003; accepted on April 4, 2003

## ABSTRACT

We describe two sets of programs for expediting routine tasks in analysis of cDNA microarray data and promoter sequences. The first set permits bad data points to be flagged with respect to a number of parameters and performs normalization in three different ways. It allows combining of result files into comprehensive data sets, evaluation of the quality of both technical and biological replicates and row and/or column standardization of data matrices. The second set supports mapping ESTs in the genome, identifying the corresponding genes and recovering their promoters, analyzing promoters for transcription factor binding sites, and visual representation of the results. The programs are designed primarily for *Arabidopsis thaliana* researchers, but can be adapted readily for other model systems.

**Availability and Supplementary information:** http://www.personal.psu.edu/nhs109/Programs/

**Contact:** nigam@psu.edu

## INTRODUCTION

Modern genomics experiments produce large amount of microarray and sequence data, efficient analysis of which requires batch processing of experiments and automation of routine tasks. We describe a suite of Perl scripts for this purpose. The first six, called the 'Microarray analysis programs', support rapid processing of Genepix (a common microarray image analysis application) result files to flag bad data points with respect to a number of parameters and normalize data in one of three different ways (Quackenbush, 2002). Separate scripts compile result files into comprehensive data sets that can be analyzed by packages such as cluster (Eisen *et al*., 1998) and SAM (Tusher *et al*., 2001). Additional scripts evaluate the quality of technical and biological replicates and perform row and/or column centering. Finally, we describe an implementation of the QT_Cluster algorithm to find biologically meaningful similarities between gene expression profiles (Heyer *et al*., 1999).

*To whom correspondence should be addressed.

## PROGRAMS

FlagAndNormalize program accepts the gpr file output from Genepix and a list of positive spiking controls and negative controls. The user decides whether to use the mean or median pixel intensities in calculating the expression ratio, using the following formula:

$$X = \frac{[\mathrm{FG} - \mathrm{BG}]^{\mathrm{Treatmentchannel}}}{[\mathrm{FG} - \mathrm{BG}]^{\mathrm{Controlchannel}}}$$

where $X$ is the gene expression ratio, FG the foreground mean or median pixel intensity for a spot and BG the background mean or median intensity for a spot. The program then flags (1) spots called 'bad' by the Genepix program; (2) spots with a low signal to noise ratio (SNR) in either channel [SNR $=$ FG $-$ (BG $+$ 2SD)]; (3) spots whose intensity is low in both channels; and (4) spots for which the FG-mean and FG-median intensities differ by more than 20%. The script calculates a spike-normalization factor based on spiking control spots that were called good and a whole-chip normalization factor based on *all* spots that were not flagged. The user can also specify a normalization factor. The three normalization factors are then used to scale the expression ratio ($X$) for each gene and adjust for differences in red and green channel intensities. The normalized ratio columns are appended to the gpr file. Existing programs such as Gp3 (Fielden *et al*., 2002) do not allow for spiking normalization, which is essential for specialized microarrays (Mahalingam *et al*., 2003). Details on spiking normalization and whole-chip normalization can be found in Supplemental Information.

The ColumnPicker program accepts a list of files made by the previous program and a list of column headings to compile a data set in which each column (or group of columns) represents a microarray slide and the rows represent spots on the slide. This is the most common format used as input for expression analysis programs. The HandleTechReplicates program accepts the data set created by the ColumnPicker program, which has two columns for each microarray slide: a data column and a flag column. It writes an output file of

averaged values from replicate spots, and provides information that is useful in identifying the rows to be kept for further analysis. It counts the flags for each replicate spot and appends a column showing the percentage of slides on which replicate spots were flagged, as well as one that gives the number of slides on which the ratios for the two replicate spots differed by more than 20%. It also removes the data rows for the positive and negative control spots. The program provides the following additional statistics for time course data: (1) the correlation between the profiles of the replicate spots for a gene and (2) the mean-ratios for the rows of replicate spots for each gene. The output of this program can easily be manipulated in Excel to achieve the desired row filtering. Once the selection of 'good spots' or 'good rows' is made, the following additional scripts can be used.

The AverageBioReps program accepts the outputs of the HandleTechReps program and averages the biological replicate expression values for each gene or row. It also reports the correlation between the expression profiles for each gene (or row), an average correlation value for all the rows analyzed and a statistic showing the number of instances (or columns) in which the values from the two replicates differed by more than 20%. The CenterByRowOrColumn program takes in the output of the HandleTechReps program or a tab delimited text file where rows are genes and columns contain expression values from one experiment and centers the data matrix by row, column or both. The user specifies the number of iterations while performing both row and column normalization. The GetSimilarGenes program takes in two expression files, a seed profile file and an expression data file. It extracts profiles that are similar to the seed profile from the expression data file based on either simple or jackknife correlations. The expression data file itself should be used as the seed to implement the first phase of the QT_Cluster approach (Heyer *et al*., 1999).

Once a set of 'interesting genes' is identified, the next step is to map them in the genome and recover their annotations, determine the functional categories of the encoded proteins, and identify transcription factor binding sites in their promoters. The seven 'Motif-search programs' allow the user identify genes corresponding to ESTs, then recover their promoters and analyze them for over-representation of transcription factor binding sites. Search results can be converted into an image showing the location of each binding site in the complete set of promoters. A matrix of the distribution of submitted transcription factor binding sites in the complete set of *Arabidopsis* promoters can also be generated, contributing to the reconstruction of gene networks from microarray and transcription factor binding-site data (Lee *et al*., 2002).

The EstSeqToMips program compiles a list of non-redundant AGI-identifiers for the *Arabidopsis* genes corresponding to the set of EST sequences submitted to BLAST at TAIR (Rhee *et al*., 2003). It will also list the redundant AGI IDs and provide a description of the gene each EST matched, the score, expect value, identities and the associated annotation. The program parses the BLAST result file from TAIR. Adapting the script for model organism databases, which output the BLAST result in a format compatible with Bioperl, would be a straightforward task. The GetPromoterSeq program compiles a set of promoter sequences identified by AGI ids using a list of non-redundant AGI ids as input.

The Motif-finder program searches for a given motif in a set of promoter sequences and computes a *p*-value using a binomial distribution and the frequency with which the motif is represented in the complete set of *Arabidopsis* promoters. The results file also stores the position and strand where the motif was found in a promoter. Binding sites represented as a list of regular expressions can also be submitted for analysis. The Motif-finder-auto program does the same job as Motif-finder, but accepts a list of motifs and a list of upstream sequence files to write a tabular summary for each. The DrawMotifPicture program converts the results file of the Motif-finder into a picture showing the location of each motif in each promoter. Additional programs merge the results files for separate motifs and draw an overview image showing the different motifs in each promoter in different colors. The Motif-Matrix program accepts a list of motifs and makes a matrix comprising a column for each motif and a row for each promoter, reporting number of each motif in each promoter.

## IMPLEMENTATION

The programs are written in Perl 5.6 and run on Windows NT and 2000 platforms. They are available free to the academic user. The programs use the Statistics::Descriptive and the Bioperl modules (Stajich *et al*., 2002). In the supplemental information, we provide a visual overview of the analysis, a detailed step-by-step description of how to use the programs, and sample input and results files. These programs can be 'pipelined' to perform analysis in batches and run as standalone applications. Hence, they offer an advantage in terms of ease and speed of analysis. They also offer the functionality to search for a user-specified motif in a user-specified set of promoters and display the results visually. While these programs were designed for the *Arabidopsis* research community, they can be modified readily for analysis of data from other organisms.

## REFERENCES

Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.

Fielden,M.R., Halgren,R.G., Dere,E. and Zacharewski,T.R. (2002) GP3: GenePix post-processing program for automated analysis of raw microarray data. *Bioinformatics*, **18**, 771–773.

Heyer,L.J., Kruglyak,S. and Yooseph,S. (1999) Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.* **9**, 1106–1115.

Lee,T.I., Rinaldi,N.J., Robert,F., Odom,D.T., Bar-Joseph,Z., Gerber,G.K., Hannett,N.M., Harbison,C.T., Thompson,C.M., Simon,I. *et al*. (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799–804.

Mahalingam,R., Gomez-Buitrago,A., Eckardt,N., Shah,N., Guevara-Garcia,A., Day,P., Raina,R. and Fedoroff,N.V. (2003) Characterizing the stress/defense transcriptome of Arabidopsis. *Genome Biol.* **4**, R20.

Quackenbush,J. (2002) Microarray data normalization and transformation. *Nat. Genet.*, **32** (Suppl.) 496–501.

Rhee,S.Y., Beavis,W., Berardini,T.Z., Chen,G., Dixon,D., Doyle,A., Garcia-Hernandez,M., Huala,E., Lander,G., Montoya,M. *et al*. (2003) The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. *Nucleic Acids Res.*, **31**, 224–228.

Stajich,J.E., Block,D., Boulez,K., Brenner,S.E., Chervitz,S.A., Dagdigian,C., Fuellen,G., Gilbert,J.G., Korf,I., Lapp,H. and Lehvaslaiho,H. (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.

Tusher,V.G., Tibshirani,R. and Chu,G. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl Acad. Sci. USA*, **98**, 5116–5121.