An Algorithm for Screening of Genes and Clusters from Microarray Experiments

James E. Blum, University of North Carolina Wilmington, Wilmington, NC

ABSTRACT

This paper presents an implementation, using the SAS® System, of the "cluster scoring" method proposed by Tibshirani, Hastie, Balasubramanian, Eisen, Sherlock, Brown and Botstein (2002) for use in microarray experiments. The program is designed in a modular fashion, using SAS Macro Language, so that implementations for specific experimental cases can be added with ease. Current development accommodates two-class, multi-class, two-way factorial and quantitative response experiments. Also, the program is built with the future addition of a graphical user interface, and possible web deployment, in mind.

INTRODUCTION

In microarray experiments, typically several thousand gene expressions are measured simultaneously with the goal of determining differential expression related to some characteristic of the subjects in question. This characteristic may be different levels of a treatment application, response categories, or even a quantitative response. In order to test for differential expression in cases like these, the Significance Analysis of Microarrays (SAM) procedure was proposed by Tusher, Tibshirani and Chu (2001). A description of this procedure is provided below; however, it can essentially be described as a permutation test based on fairly common test statistics.

Since it is reasonable to expect that groups of genes will operate in conjunction, one would expect a significant association in their expressions. However, SAM (and many other methods) treat expressions as a collection of individuals when it may well be more appropriate to consider both individual genes and clusters of genes. In response to this concern, Tibshirani *et al.* proposed a "cluster scoring" method which allows for a screening of individual genes and clusters of genes for differential expression. It is a generalization of SAM in the sense that it uses a similar style of test statistic to determine significant differential expression; however, it works with a full hierarchical clustering of the genes.

This paper presents an implementation of this "cluster scoring" procedure using the SAS System. Some common experimental situations are discussed along with some generalizations of the method and their implementation.

THE SAM PROCEDURE

The SAM procedure of Tusher *et al.* is a method for determining significant changes in gene expression for different experimental states. The basis for the procedure is a score of the following form:

$$d_i = \frac{r_i}{s_i + s_o}$$

Here r_i measures the relationship between the response variable, y, and the expression of the i^{th} gene, where the response may actually be quantitative or may simply represent different states or classes. For example, if the response is quantitative, r_i can be taken in terms of the Pearson correlation coefficient. For a case where outcomes fall into one of two groups, r_i may be taken as the difference in mean expression levels for the two groups. A discussion of several possible scenarios and their corresponding test statistics is given in the SAM User's Guide.

The quantity s_i is a measure of standard error for r_i. As an example, for the two group case, Tusher *et al.* chose the traditional standard error estimate based on pooling variances from the two groups. The quantity s_o is an adjustment factor designed to prevent genes with low expression (which often have low s_i) from dominating the results. In the SAM procedure, s_o is chosen as a particular quantile from the set of s_i. Details of this choice can be found in the SAM User's Guide. Other choices have also been proposed, see Broberg (2002).

Given that we have a set of p gene expression values and we can compute an appropriate d_i for each, the SAM procedure determines significance in the following way.

OUTLINE OF THE SAM PROCEDURE

1. Compute the order statistics for the d's: $d_{(1)} = \min\{d_i\}$, $d_{(p)} = \max\{d_i\}$, $d_{(1)} \le d_{(2)} \le \ldots d_{(p-1)} \le d_{(p)}$.

2. Take B sets of permutations of the response values/levels. For the b^{th} permutation compute the order statistics $d_{(i)}^{\,b}$ as in step 1. Estimate the expected order statistics by averaging over the B permutations:

$$\overline{d}_{(i)}^{\,b} = \left(\sum_b d_{(i)}^{\,b}\right) / B$$

3. Plot observed values versus expected values. Using the origin as a reference, for increasing values of the observed and expected order statistics, find the first $i = i^*$ such that $d_{(i)} - \overline{d}_{(i)} > \Delta$ for a fixed threshold Δ. All genes past i^* are considered positively significant. Likewise, negatively significant genes are determined by finding the first value where $\overline{d}_{(i)} - d_{(i)} > \Delta$.

4. For a grid of Δ values, compute the expected number of falsely significant genes by taking the median number of genes called significant (based on the cut-offs found in step 3) in the B sets of permutations.

5. The user then may pick a value of Δ that provides them with a reasonable false discovery rate and the significant genes are listed.

THE CLUSTER SCORING PROCEDURE

If one expects associations between gene expression and experimental conditions and/or responses and also expects strong correlations among the gene expressions themselves, then it is reasonable to believe that changes in expression levels may be better described in terms of groups of genes as well as individuals. The cluster scoring procedure of Tibshirani *et al.* uses some of the basic principles of SAM to create a method for determining significant association between gene expression and response for both clusters of genes and individual genes.

The procedure begins with a clustering of the genes. Tibshirani *et al.* choose hierarchical clustering using average linkage as the

method for joining; however, they also note that there are several other methods that may be used. Discussion of possible methods is taken up in the section that describes clustering using the SAS System.

Assuming we begin with a hierarchical clustering of p genes from a sample of n subjects, the cluster scoring method is described by Tibshirani *et al.* as follows:

CLUSTER SCORING

1. For each of the $2p - 1$ clusters (including individual genes), denote a cluster of genes by c, and let p_c denote the number of genes in that cluster. Further, denote the corresponding gene expression average by

$$\bar{x}_c = \left(\bar{x}_{c,1}, \bar{x}_{c,2}, \ldots \bar{x}_{c,n} \right)..$$

2. Define the score for each average gene expression as

$$d_c = \frac{r_c}{s_c + s_o(p_c)}.$$

   Where r_c measures association between gene expression and response, s_c is a measurement of standard deviation and $s_o(p_c)$ is an adjustment factor.

3. For each gene i, let $c(i)$ be the set of clusters containing that gene. For $c \in c(i)$, pick the "winning cluster" as the cluster c for which $|d_c|$ is maximized. Let R denote the set of winning clusters.

4. Apply the SAM procedure to the clusters $c \in R$. When computing false positives, count unique genes.

The adjustment factor $s_o(p_c)$ is computed by dividing the cluster sizes into quantiles $q_1, q_2, \ldots q_k$ and taking $s_o(p_c)$ as the α percentile of $\{s_c \mid p_c \in (q_j, q_{j+1})\}$. (100 quantiles are suggested with $q_1 = 0$ and $\alpha = 0.5$.) Effectively, the adjustment factor for a given cluster score is based on clusters of similar size.

It is important to note that while the quantiles $q_1, q_2, \ldots q_k$ are not specific, they should not be chosen arbitrarily. For example, choosing the quantiles as the set of percentiles 0, 1, 2,…, 99, 100 would not be particularly effective since the median cluster size is one. Of course, the individual genes could (and perhaps should) be taken as their own separate group, setting up quantiles for the remaining clusters, but those still need to be chosen to ensure distinctions among cluster sizes. Methods for setting up these quantiles are discussed in the section describing computation of scores.

IMPLEMENTING THE CLUSTER SCORING PROCEDURE WITH THE SAS SYSTEM.

Implementation of the cluster scoring procedure makes use of several procedures from SAS/STAT and other SAS software. In an effort to make the application flexible enough to accommodate a range of experimental situations, a modular design is employed, making use of SAS Macro Language. There are three basic modules, each constructed using a macro or collection of macros. Two of the modules are computational in nature, one handles the clustering while the other does the scoring for the genes and clusters. The third module selects the winning clusters for each gene and removes redundant observations (winning clusters for multiple genes).

Before continuing with a description of the methods, it seems appropriate to briefly describe one of the data sets that has been used to test the procedure and which will be referred to in later discussions. To study relationships between gene expression and exercise tolerance in mice exposed to hypoxia, McCall, Frierson, Blum, Knowles and Kinsey (2003) collected microarray from a factorial experiment. There were two main factors under consideration, one with two levels: hypoxia exposure and no exposure. The other factor was the strain of mouse, with three strains having been considered, two parental strains and their cross. Five subjects from each strain were exposed to hypoxia, six subjects from two of the strains and seven from another were left in normoxic conditions. RNA samples were taken from each subject and hybridized individually to arrays containing 5163 genes.

SCORING

Assuming clustering has been completed (clustering will be discussed in the next section) computation of the scoring statistic d_c requires a measure of association, r_c, between gene expression and the response or grouping variable and an appropriate error measure, s_c. Of course, the statistics computed will depend on the scenario in question, so the procedures invoked will vary as well.

In the simplest case, all subjects in the experiment are of a single class and RNA from treatment and control conditions are hybridized simultaneously (one with green dye and the other red) to each array. The data typically takes the form $x_i = \log_2(t_i/c_i)$, where t_i and c_i are expression levels for the i^{th} gene under treatment and control conditions, respectively. The test statistic here would be analogous to a single sample t-statistic with the null hypothesis of a zero mean. The UNIVARIATE procedure, among other possible choices, is capable of producing the necessary elements to construct d_c in this situation.

In situations where subjects fall into two or more classes, the ANOVA or GLM procedure, using the OUTSTAT= option, can be used to generate elements for constructing d_c. These procedures could also be applied to construct a reasonable d_c in multi-factor experiments, using appropriate sums of squares to gauge factor effects and random error. For single factor cases or balanced multi-factor designs, the ANOVA procedure is more efficient and would be preferred. However, in unbalanced multi-factor designs—like the hypoxia study described above—basing test statistics on type III sums of squares would generally be preferred, so PROC GLM should be used. A quick note: for the experiment described above there are 10,325 genes and clusters for which four statistics (one for each main effect, one for interaction and one for error) must be computed before the d_c's can be constructed, so computational efficiency will be an important consideration.

For situations with a quantitative response, the REG procedure can provide the necessary information. It is the author's choice to use a test on the slope of the line as proxy for testing the correlation. Specifying the OUTEST= option together with the COVOUT option generates the necessary information.

Other interesting possibilities can be considered as well. In the hypoxia study, the six different factor combinations are under study due to observed differences in exercise tolerance, measured quantitatively in terms of time to fatigue. One parental strain showed moderately high tolerance when exposed to hypoxia while the cross showed extremely high exercise tolerance under hypoxic exposure. However, the precise quantitative response was not available for each mouse from which the RNA samples were drawn. So the major effect of interest was neither a main effect or interaction, but actually a two degree of freedom contrast, which can easily be extracted using the GLM or ANOVA procedure.

Regardless of the situation, in order to compute d_c, the adjustment factor $s_o(p_c)$ must be computed as described previously. While it is possible to construct an algorithm to generate distinct quantiles $q_1, q_2, \ldots q_k$ for the cluster sizes, an alternate approach has been taken. Since the adjustment factor is to be based on error terms for clusters of similar size, one can group the cluster sizes with another clustering procedure. Using the FASTCLUS procedure on the cluster sizes allows them to be placed into a set number of groups determined by the MAXCLUSTERS= option. Once the groups are determined, PROC UNIVARIATE can be used to determine the α percentile for each group, using standard options (MEDIAN, P5, P95, *etc.*) or the PCTLPTS= and PCTLPRE= options.

At this point, all of the necessary elements to compute the d_c's have been constructed. What remains is to search this set to find the maximum value corresponding to each gene. Before that can be accomplished, more information about the clustering of the genes is required.

CLUSTERING

Forming a hierarchical clustering of the genes can be done with the CLUSTER procedure (or with the VARCLUS procedure). The METHOD= option provides several different options for joining clusters, including average linkage. The OUTTREE dataset provides important information for computing scores such as, averages for each cluster and the size of each cluster. It also provides a history of the joining of observations and clusters at every stage of the procedure. Since the cluster scoring method requires considering scores for every cluster for each gene, a record of membership of individual genes for each cluster needs to be created.

It should be noted that other types of clustering could also be used. One could use *k*-means clustering for several values of *k* to get a cluster set that looks much like a hierarchical one. Any method that can produce sets of clusters of a variety of sizes is potentially acceptable. For what follows though, an agglomerative hierarchical scheme is assumed.

To create a record of cluster memberships, consider the concept of an incidence matrix. In this matrix, $M = \{m_{ij}\}$, each row represents a particular cluster and each column represents a gene. If the j^{th} gene is in the I^{th} cluster then $m_{ij} = 1$; if not, $m_{ij} = 0$. To construct this matrix, SAS/IML is used in conjunction with a modification of the OUTTREE data set from the CLUSTER procedure. The process uses two variables from the OUTTREE data set, _NAME_ and _PARENT_, which denote the observation or cluster being joined and the resulting cluster, respectively. These variables each have the form of two characters, either OB or CL, and a set of digits, either observation or cluster number.

Naming the OUTTREE data set as `clresult`, the following code separates the pieces of information contained in each variable and constructs the incidence matrix (`incd`).

```
data clhistory(keep= namelabel namevalue
     parentvalue);
     set clresult;
     namelabel=substr(_name_,1,2);
     namevalue=input(substr(_name_,3),5.);
     parentlabel=substr(_parent_,1,2);
     if parentlabel ne ' ';
     parentvalue=input(substr(_parent_,3),5.);
run;

proc iml;
     use clhistory;
     read all var {namelabel} into label;
     read all var {namevalue parentvalue} into
             number;
     genes=(nrow(label)+2)/2;
     clusters=nrow(label)/2;
     incd=J(clusters,genes,0);

     do i=1 to nrow(label);
       if label[i]='OB' then
        incd[number[i,2], number[i,1]]=1;
       else if label[i]='CL' then
        incd[number[i,2],]=incd[number[i,2],]+
                           incd[number[i,1],];
     end;
```

The feature to note here is that in an agglomerative clustering procedure two entities are joined at each step. The entities joined at a given step are either individual observations not assigned to a cluster or previously formed clusters, with the new cluster replacing the two entities that were joined. For the incidence matrix, if an individual observation is included in forming a new cluster, the appropriate entry should be changed from 0 to 1. If an existing cluster is included in the formation, 0 needs to be replaced with 1 for the set of entries corresponding to those genes in the existing cluster. However, in this context, replacement can be achieved with addition of the row for the existing cluster to the row for the new cluster.
Once the incidence matrix is created, it is output as a data set and can easily be converted to a data set (as shown below) with two variables, one that names the cluster the other that names a gene in that cluster.

```
data cluster_history;
     set incidence;
     %do i=1 %to &numgenes;
       if col&i ne 0 then do;
         _name_='clus'||left(trim(_N_));
          gene='gene'||left(trim(&i));
          output;
       end;
     %end;
     keep _name_ gene;
```

This data set is then merged to the data set containing the d_c values.

At this point the resulting data set can be sorted by gene and by descending value of d_c. Using FIRST. processing in a data step, the records corresponding to the maximum d_c can be extracted and subsequently duplicates can be removed. The resulting data is ready for analysis using the SAM procedure.

EFFICIENCY AND OTHER CONSIDERATIONS

Once the computational questions have been sufficiently answered, the question of efficiency remains. Recall that for the hypoxia study over 10,000 analyses of variance needed to be conducted. It is also of note that the data set created for the clustering history contains nearly 1.2 million records, so sorting it is no minor task. Though extensive benchmarking of the application has not been done at this time, some specific recommendations for improving performance can be made.

STRATEGIES FOR REDUCING COMPUTATION TIME

The GLM, ANOVA and REG procedures all allow for multiple response variables in the model; however, including several thousand variables at once is not necessarily a good idea. A macro could also be created to repeat the procedure for each variable, but this is probably not the best solution either. Somewhere between these two extremes is an optimal solution, likely dependent upon the operating system and hardware in use. Testing on Windows XP based systems in the 0.8 to 2.4 GHz range has shown that computing between 100 and 200 models simultaneously, and concatenating the resulting data sets, provides a reasonable result. The number of variables considered simultaneously is a user controllable parameter in the

current implementation of the program (except in the single class case which uses PROC UNIVARIATE and does all variables simultaneously).

When choosing the maximum score for each gene, sorting an extremely large data set is required. Here, a divide and conquer approach, like those demonstrated in the Optimizing SAS Programs course, pays dividends. Dividing the full set into smaller sets (150-200 thousand records each), sorting those and interleaving the results provided a nearly 80% improvement. In fact, this can be improved further by selecting the maximum score for each gene in each sorted subset (the overall maximum must be the maximum in its subset) and interleaving the now smaller resulting data sets.

Implementation of these strategies provided a 75% improvement in completion time for the hypoxia data set (Windows XP based machine, 2.4GHz, 512Mb RAM) . Searching for more efficient methods and benchmarking current implementations across multiple platforms is an ongoing process in the development of this application.

USER CONTROL

In addition to the three main components of the program, two other components are included to enhance usability. The first component defines a set of global macro variables that control various options. Some of these options include: the clustering method, the experiment/analysis type, selection of a percentile used to compute $s_o(p_c)$, the library name and path, and a variable list. An additional module allows for the reading of delimited raw data files with user control over the delimiter and record length. Column and formatted input may also be made available.

CONCLUSION

The cluster scoring procedure allows for potentially meaningful data reduction in microarray experiments. Hopefully, this implementation within the SAS System will allow for its use for a variety of cases.

Various components of the procedure are to be made available at the author's web site (see contact information below) along with supporting documents that detail available analyses and choices for user specified options. Since the development of the method is, by the admission of Tibshirani *et al*., "exploratory", recommendations for choices of test statistics and adjustment factors will be included as the become available. Hopefully, at some point in the near future, this will all be packaged with a nice user interface (using SAS/AF or SAS/webAF) and help system.

REFERENCES

Broberg, P. (2003) Statistical Methods for Ranking Dfferentially Expressed Genes, *Genome Biology*, 4:R41.

Chu, C., Narasimhan, B., Tibshirani, R., Tusher, V. (2002) *SAM Users Guide and Technical Document*. < http://www-stat.stanford.edu/~tibs/SAM/index.html>

McCall, D., Frierson, D., Blum, J., Knowles, M., Kinsey, S., DNA Microarray Analysis of Hypoxia Induced Fatigue in Skeletal Muscle," *presented at the* International Hypoxia Symposium, Banff, Alberta, Canada, February 2003. (Manuscript in preparation.)

Tibshirani, R., Hastie, T. , Narasimhan, B. , Eisen, M. , Sherlock, G. , Brown, P., and Botstein, D. (2002), Exploratory screening of genes and clusters from microarray experiments, *Statistica Sinica*, 12 (1) , 47-59.

Tusher, V., Tibshirani, R., Chu, C. (2001). Significance Analysis of Microarrays Applied to Transcriptional Responses to Ionizing Radiation. *PNAS*, 5116-5121.

ACKNOWLEDGMENTS

I would like to acknowledge Dale McCall, Stephen Kinsey, Ann Stapleton and Dargan Frierson for introducing me to microarray experiments and for many helpful discussions related to their analysis. I would also like to thank my instructors for the Optimizing SAS Programs course, Roger Staum and Michelle Ensor; and my instructor for the SAS Macro Language course, Cynthia Teague.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

James Blum
University of North Carolina-Wilmington
601 S. College Road
Wilmington, NC 28403-5970
Work: (910) 962-4299
Fax: (910) 962-7107
Email: blumj@uncw.edu
Web: http://people.uncw.edu/blumj