Performance Analysis of Differential Expression Prediction Algorithms Using Simulated Array Data

G. Grant[1], S. Sokolovsky[2], C. Stoeckert[1]

Microarray gene expression data are now being used routinely to search for genes which are differentially expressed between different experimental conditions. Such conditions might be normal versus diseased tissue, or the same tissue under different developmental stages, or tissues subject to different environmental conditions, etc. There are several statistical methods that have been developed to make predictions of differential expression from replicate data sets. Given that there are no adequate benchmark datasets from real biological systems, the efficacy of these methods is difficult to test. In this study our goal is to generate a generic gene expression data simulation engine that can be used to some basic tests of certain types of methods.

1 Introduction

The general problem is to determine differentially expressed genes between a pair of sample types, which will be referred to as *type 0* and *type 1*. Biological and experimental variability necessitate the use of replicate experiments for each type. The data consist of a set of array experiments, each assaying, for many genes in parallel, the expression levels of one of the two sample types. We assume there are at least two experiments for each of the two sample types. We refer to the set of replicate experiments of the two sample types as *group 0* and *group 1* respectively. Typically the resources are available to generate only a few replicates experiments in each group. The interest is in determining a set of genes which most strongly indicate differential expression. These predictions may then be verified as true or false with single gene assays such as RT PCR.

Besides the general availability of only a few experiments in each group, two other important issues involved in the statistical analysis of this type of data are that of multiple testing and that gene intensity distributions tend to be nonstandard.

Multiple testing refers to the fact that we are performing many experiments in parallel, one for each gene, so that any significance method which could be applied to single gene comparisons cannot be performed in parallel without some kind of correction factor. (see Ewens et al., 2001, Section 3.8). When many genes are involved, this correction factor can have a substantial impact, so differential expression of a single gene that might be evident with a certain number of replicates might require more replicates in order not to get lost in the noise of the many other genes in the assay. It is therefore important to have as many replicates as possible and to use the least conservative statistical approach possible for correcting for multiple testing.

There are many classical methods for testing the null hypothesis of equal means between two distributions, however the fact that the gene intensity distributions are generally of unknown types limits the applicability of such methods. If, for example, normality assumptions could be made about the gene intensity distributions across each group of replicates, then classical two-sample t-tests could be used. But if one or both of the sample types is heterogeneous, for example if one type is Acute Lymphoblastic Leukemia cells which are a combination of B- and T-cells, then the distribution for some of the genes (the B- or T-cell specific genes in the above example), can be bimodal (see Grant et al., (2000) for more details). Using standard t-tests can in this case misidentify genes as being differentially expressed when in fact they just have highly non-normal distributions which have biased the t-test significance measure. Since the normality assumption does not hold in general, such t-tests can be replaced by permutation tests, but this generally requires more replicates.

[1]Penn Center for Bioinformatics, University of Pennsylvania
[2]Department of Computer and Information Sciences, University of Pennsylvania
Contact: ggrant@pcbi.upenn.edu

For some methods, such as the ANOVA methods (Kerr et al., (2000)), simulated data of quite a different nature would have to be generated. Given the lack of knowledge about certain features of expression data, such as the various confounding effects between different factors such as array, dye, gene, etc., it is not clear how that could be done at the present time in a way that would not bias the results unacceptably.

2 Simulating gene expression data – the basic model

It is impossible to model gene expression data precisely since the true nature of such data is not well understood. It is possible however to capture enough of the nature of the data to perform meaningful tests of the algorithms described above. Public data sets elucidate many of the properties of expression data, for example that one-channel data intensities are exponentially distributed, or that gene intensities are not normally distributed as is often assumed (see Grant et al. 2000).

The model is intended to have enough flexibility to elucidate how the results depend on many different parameters.

For our purposes, a *run* will mean the generation of K data sets. Each data set will consists of n replicates in group 0 and m replicates in group 1, each replicate being the intensity levels of N "genes."

For each run, M genes are picked to be differentially expressed. For each of the $N - M$ other genes, a mean intensity level $\mu(g)$ is chosen from a fixed distribution. The distribution type and its parameters are specified in a configuration file. We have implemented the exponential (as is typical of one channel data), or a beta (to model intensities for ratios in two-channel data), however any distribution could be used.

Once a mean intensity is chosen, the distribution of intensities for each gene is modeled with a beta distribution. Beta distributions were used because unlike Gaussian distributions, they have finite range, and their shape can be varied widely by adjusting the two parameters, $\alpha(g)$ and $\beta(g)$. The parameters $\alpha(g)$ and $\beta(g)$ are chosen uniformly in a range $[a, b]$, where a and b are set in the configuration file and are fixed for each run. Any desired percentage of the $\alpha(g)$'s and $\beta(g)$'s can be chosen so that the distribution is symmetric. The left endpoint $L(g)$ of the range of the distribution is chosen uniformly in $[0, \mu(g)]$. The right endpoint is then set to be $2\mu(g) - L(g)$. This allows for a heterogeneous set of gene intensity distributions.

The parameters are fixed throughout the run, so that the intensities for a given gene in a given group are generated by the same distribution for each replicate of each of the K data sets. For the non-differentially expressed genes the same distribution is used for all replicates, regardless of which group it is in. Most of the parameters for the M differentially expressed genes are chosen by hand, and are not randomly generated (however the intensities are randomly generated in each replicate, according to the chosen distributions).

This gives a mechanism for generating as many replicates as desired of a given "experiment," with the differentially expressed genes known a priori. This data can then be used to test the performance of algorithms. Any algorithm that claims to determine differentially expressed genes from array data should be expected to work on this simulated data. If they cannot, then they should not be expected to work well on real data. Of course if they do work on this data, that is no guarantee they will work on real data exactly as well, given that real data will inevitably have more complexities that we will manage to model.

The model described above generates gene intensities in a gene-independent fashion. In reality, gene intensities will contain many dependencies between them, and algorithms for predicting differential expression might take these dependencies into account. The step-down method described above, for example, is designed to exploit gene dependencies. As a simple test of such claims, the model has been extended to generate intensities with dependence in the following extreme way:

If there are G non-differentially expressed genes, and A is chosen to be less than G then A genes are generated independently, and each of the remaining $G - A$ genes intensities are taken to be equal to some one of the A independent genes. For example gene 100 might be linked to gene 350, so that they always have the same expression level for every replicate. This is a very strong kind of dependence, so that if dependencies are affecting the performance of an algorithm, it should become apparent when varying the parameter A from 0 to G.

In practice, values that are below a certain threshold deemed to be background are often set to a default baseline intensity value. The percentage of genes that are thresholded is sometimes the only indication of differential expression (Grant et al., (2001)). In the model, thresholding can be turned on or off, and if on, then for each gene a fixed percentage is chosen from a distribution on $[0, 1]$ and that percentage gives the probability that a value is thresholded in any given replicate. These percentages are chosen by hand for the differentially expressed genes.

The graphs in figures 1 and 2 show some examples of the empirical gene-intensity distributions generated.

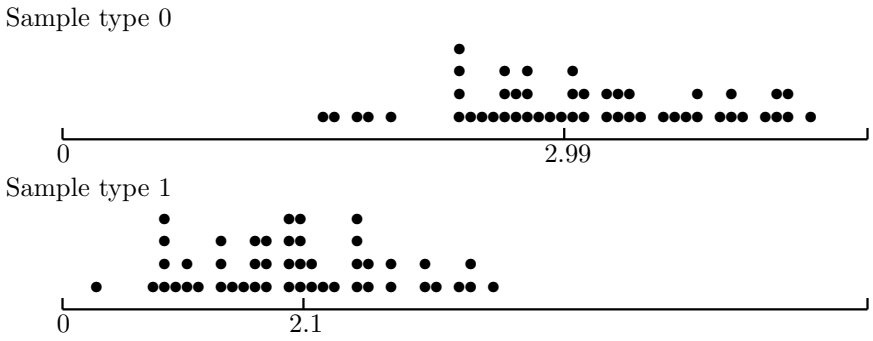

Figure 1: Example of a differentially expressed gene with 50 replicate experiments for each type.

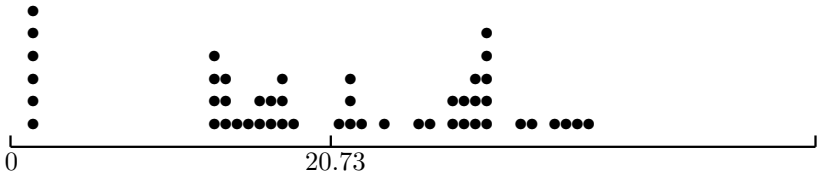

Figure 2: Example of a gene with thresholded values.

No effort was made to simulate pre-normalized or pre-cleansed data. The data sets are supposed to mimic sets of replicates that are already normalized to each other.

The following is an example configuration file

```
;Simulation configuration file
;Comments begin with ';'
;Only one parameter per line is allowed
;Parameter must be separated by one or more spaces from the value(s)
```

```
;DISTPAR is the type of distribution from which the mean values for
;expression levels of non-differentially expressed genes will be selected
;The format is DISTPAR <type> <mean> <spread> <alpha> <beta>
;1 corresponds to exponential distribution
;2 corresponds to beta distribution
;For exponential distribution all parameters after <mean> have to be specified as 0
DISTPAR 1 10.0 0 0 0

;THR - a number between 0 and 1 inclusive which indicates the percent of thresholded genes
for non-differentially expressed genes
;1 - on, 0 - off
THR 0

;NUMA is the number of replicates per gene for cell type A
NUMA 50

;NUMB is the number of replicates per gene for cell type B
NUMB 50

;RAB is the range for alpha and beta parameters in the beta distribution
RAB .5 6

;EQ is the percent chance that alpha and beta parameters are equal
EQ 50

;RT is the percent chance that beta distribution is skewed to the right (given
;unequal alpha and beta)
RT 50

;GENN is the total number of genes
GENN 3000

;INDN is the number of genes that are independently distributed
;The expression levels for these genes will follow directly after
;the differentially expressed genes
INDN 2900

;FILP is the prefix for the files containing simulated data
;file names are in the format <FILP>(A|B)<number>
;List of all files generated is written in file 'simflist'
FILP simf

;DS command starts the simulation generator, and following DS are
;specific instructions for expression data simulation
DS

;DIFF is the command forcing the simulator to produce differentially
;expressed genes.  The format of the command is
;DIFF <number> <ratio thresholded1> <mean1> <ratio thresholded2> <mean2>
;for each DIFF command the simulator will produce <number> of
;differentially expressed genes with the specified means and
;spreads linearly decreasing from <mean> to <mean>/<number>
```

```
DIFF 10 0.0 2.0 0.0 3.0
DIFF 10 0.0 2.0 0.0 4.0
DIFF 10 0.0 2.0 0.0 6.0
DIFF 10 0.0 2.0 0.0 8.0
DIFF 10 0.0 2.0 0.0 10.0

DIFF 10 0.0 10.0 0.0 15.0
DIFF 10 0.0 10.0 0.0 20.0
DIFF 10 0.0 10.0 0.0 30.0
DIFF 10 0.0 10.0 0.0 40.0
DIFF 10 0.0 10.0 0.0 50.0

;END directive tells simulator to continue producing data for
;non-differentially expressed genes until GENN genes is produced
END
```

The plot on the left in Figure 3 shows a scatter plot of two "experiments" generated from this configuration. As can be seen from this plot, the spread of the data is fairly wide. This allows for fairly conservative results. I.e. if the algorithms perform well with such noisy data it can be expected to perform at least as well on cleaner data. The plot on the right shows a scatter plot where each axis is the average of five "experiments" on each axis. Data might be clean enough that the scatter plot between individual experiments was as tight as this scatter plot. In such a case, in the results that follow one only needs divide the number of replicates in each study by five to get the corresponding results for such data.

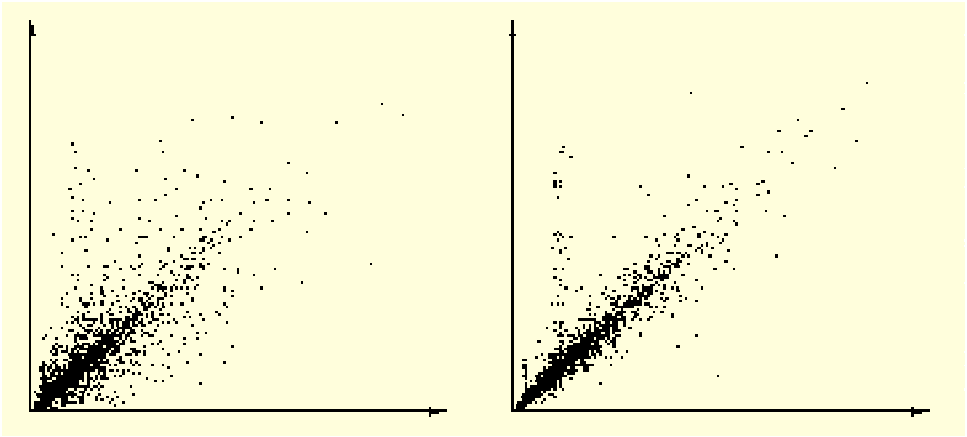

Figure 3: Scatter plots of simulated data. Plot on left shows one experiment versus another, generated as in the configuration file shown in the text. The plot on the right shows the average of five experiments on each axis. As shown in the configuration file above, there are 100 upregulated genes in the type represented by the vertical axis, 50 in the low intensity range and 50 in the medium intensity range.

References

Ewens W.J. and Grant G.R. (2001). 'Statistical Methods in Bioinformatics: An Introduction.' Springer-Verlag, NY.

Grant G.R., Manduchi E., Stoeckert C.Jr. (2001) 'Using Non-Parametric Methods in the Context of Multiple Testing to Identify Differentially Expressed Genes.' Critical Assessment of Microarray

Data Analysis (CAMDA00) (to appear 2001).

Kerr M.K, Martin M., and Churchill G.A. (2000). 'Analysis of variance for gene expression microarray data.' *Journal of Computational Biology*, bf 7 819-837.