# Parametric Empirical Bayes Methods for two Microarray Data

Christina Kendziorski, Michael Newton, Deepayan Sarkar

September 30, 2003

## Contents



# 1 Introduction

We have developed an empirical Bayes methodology for gene expression data to account for replicate arrays, multiple conditions, and a range of modeling assumptions. The methodology is implemented in the `EBarrays` R package. Functions calculate posterior probabilities of patterns of differential expression across multiple conditions. Model assumptions can be checked. This vignette provides a brief overview of the methodology and its implementation. For details on the methodology, see Newton *et al.* 2001, Kendziorski *et al.*, 2003, and Newton and Kendziorski, 2003. We note that some of the function calls in version 1.1 of EBarrays have changed.

## 2 General Model Structure: Two Conditions

Our models attempt to characterize the probability distribution of expression measurements $\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jI})$ taken on a gene $j$. As we clarify below, the parametric specifications that we adopt allow either that these $x_{j,i}$ are recorded on the original measurement scale or that they have been log-transformed. Additional assumptions can be considered with this framework. A baseline hypothesis might be that the $I$ samples are exchangeable (i.e., that potentially distinguishing factors, such as cell-growth conditions, have no bearing on the distribution of measured expression levels). We would thus view measurements $x_{ji}$ as independent random deviations from a gene-specific mean value $\mu_j$ and, more specifically, as arising from an observation distribution $f_{obs}(\cdot|\mu_j)$. For example, a gene with a large $\mu_j$ typically exhibits high expression measurements and high variability.

When comparing expression samples between two groups (e.g., cell types), the sample set $\{1, 2, \ldots, I\}$ is partitioned into two subsets, say $s_1$ and $s_2$; $s_k$ contains the indices for samples in group $k$. The distribution of measured expression may not be affected by this grouping, in which case our baseline hypothesis above holds and we say that there is equivalent expression, $\mathsf{EE}_j$, for gene $j$. Alternatively, there is differential expression, $\mathsf{DE}_j$; our formulation requires that there now be two different means, say $\mu_{j1}$ and $\mu_{j2}$, corresponding to measurements in $s_1$ and $s_2$, respectively. We assume that the gene effects arise independently and identically from a system-specific distribution $\pi(\mu)$. This allows for information sharing amongst genes. Were we instead to treat the $\mu_j$'s as fixed effects, there would be no information sharing and potentially a loss in efficiency.

Let $p$ denote the fraction of genes that are differentially expressed (DE); then $1 - p$ denotes the fraction of genes equivalentlly expressed (EE). An EE gene $j$ presents data $\mathbf{x}_j = (x_{j1}, \ldots, x_{jI})$ according to a distribution

$$f_0(\mathbf{x}_j) = \int \left( \prod_{i=1}^{I} f_{obs}(x_{ji}|\mu) \right) \pi(\mu)\, d\mu. \tag{1}$$

Alternatively, if gene $j$ is differentially expressed, the data $\mathbf{x}_j = (\mathbf{x}_{j1}, \mathbf{x}_{j2})$ are governed by the distribution

$$f_1(\mathbf{x}_j) = f_0(\mathbf{x}_{j1})\, f_0(\mathbf{x}_{j2}) \tag{2}$$

owing to the fact that different mean values govern the different subsets $\mathbf{x}_{j1}$ and $\mathbf{x}_{j2}$ of samples. The marginal distribution of the data becomes

$$p f_1(\mathbf{x}_j) + (1 - p) f_0(\mathbf{x}_j). \tag{3}$$

With estimates of $p$, $f_0$, and thus $f_1$, the posterior probability of differential expression is calculated by Bayes' rule as

$$\frac{p\ f_1(\mathbf{x}_j)}{p\ f_1(\mathbf{x}_j) + (1 - p)\ f_0(\mathbf{x}_j)}. \tag{4}$$

To review, the distribution of data involves an observation component, a component describing variation of mean expression $\mu_j$, and a discrete mixing parameter $p$ governing the pattern of expression between conditions. The first two pieces combine to form a key predictive distribution $f_0(\cdot)$ (see (1)), which enters both the marginal distribution of data (3) and the posterior probability of differential expression (4).

# 3 Multiple Conditions

Many studies take measurements from more than two cellular conditions, and this leads us to consider more patterns of mean expression than simply DE and EE. For example, with three conditions, there are five possible patterns among the means, including equivalent expression across the three conditions (1), altered expression in just one condition (3), and distinct expression in each condition (1). We view a pattern of expression for a gene $j$ as a grouping or clustering of conditions so that the mean level $\mu_j$ is the same for all conditions grouped together. With microarrays from four cell conditions, there are 15 different patterns, in principle, but with extra information we might reduce the number of patterns to be considered. We discuss an application in Section 6 in which ten array sets are measured across four cell conditions, but the context tells us to look only at a particular subset of four patterns.

We always entertain the null pattern of equivalent expression among all conditions. Consider $m$ additional patterns so that $m+1$ distinct patterns of expression are possible for a data vector $\mathbf{x}_j = (x_{j1}, \ldots, x_{jI})$ on some gene $j$. Generalizing (3), $\mathbf{x}_j$ is governed by a mixture of the form

$$\sum_{k=0}^{m} p_k f_k(\mathbf{x}_j), \tag{5}$$

where $\{p_k\}$ are mixing proportions and component densities $\{f_k\}$ give the predictive distribution of measurements for each pattern of expression. Consequently, the posterior probability of expression pattern $k$ is

$$P(k|\mathbf{x}_j) \propto p_k f_k(\mathbf{x}_j). \tag{6}$$

The pattern-specific predictive density $f_k(\mathbf{x}_j)$ may be reduced to a product of $f_0(\cdot)$ contributions from the different groups of conditions, just as in (2), and this suggests that the multiple-condition problem is really no more difficult computationally than the two-condition problem except that there are more unknown mixing proportions $p_k$. Furthermore, it is this reduction that easily allows additional parametric assumptions to be considered within the EBarrays framework.

The posterior probabilities (6) summarize our inference about expression patterns at each gene. They can be used to identify genes with altered expression in at least one condition, to order genes within conditions, or to classify genes into distinct expression patterns.

# 4 The Gamma Gamma and Lognormal Normal models

We consider two particular specifications of the general mixture model described above. Each is determined by the choice of observation component and mean component, and each depends on a few additional parameters $\theta$ to be estimated from the data. As we will demonstrate, the model assumptions can be checked using diagnostic tools implemented in EBarrays, and additional models can be easily implemented.

In the Gamma-Gamma (GG) model, the observation component is a Gamma distribution having shape parameter $\alpha > 0$ and a mean value $\mu_j$; thus, with scale parameter $\lambda = \alpha/\mu_j$,

$$f_{obs}(x|\mu_j) = \frac{\lambda^\alpha x^{\alpha-1} \exp\{-\lambda x\}}{\Gamma(\alpha)}$$

for measurements $x > 0$. Note that the coefficient of variation in this distribution is $1/\sqrt{\alpha}$, taken to be constant across genes $j$. Matched to this observation component is a marginal distribution $\pi(\mu_j)$, which we take to be an inverse Gamma. More specifically, fixing $\alpha$, the quantity $\lambda = \alpha/\mu_j$ has a Gamma distribution with shape parameter $\alpha_0$ and scale parameter $\nu$. Thus, three parameters are involved, $\theta = (\alpha, \alpha_0, \nu)$, and, upon integration, the key density $f_0(\cdot)$ has the form

$$f_0(x_1, x_2, \ldots, x_I) = K \frac{\left(\prod_{i=1}^I x_i\right)^{\alpha-1}}{\left(\nu + \sum_{i=1}^I x_i\right)^{I\alpha+\alpha_0}}, \tag{7}$$

where

$$K = \frac{\nu^{\alpha_0} \Gamma(I\alpha + \alpha_0)}{\Gamma^I(\alpha)\, \Gamma(\alpha_0)}.$$

In the lognormal normal (LNN) model, the gene-specific mean $\mu_j$ is a mean for the log-transformed measurements, which are presumed to have a normal distribution with common variance $\sigma^2$. Like the GG model, LNN also demonstrates a constant coefficient of variation: $\sqrt{\exp(\sigma^2) - 1}$ on the raw scale. A conjugate prior for the $\mu_j$ is normal with some underlying mean $\mu_0$ and variance $\tau_0^2$. Integrating as in (1), the density $f_0(\cdot)$ for an $n$-dimensional input becomes Gaussian with mean vector $\underline{\mu}_0 = (\mu_0, \mu_0, \ldots, \mu_0)^t$ and exchangeable covariance matrix

$$\Sigma_n = \left(\sigma^2\right) \mathbf{I_n} + \left(\tau_0{}^2\right) \mathbf{M_n},$$

where $\mathbf{I}_n$ is an $n \times n$ identity matrix and $M_n$ is an $n \times n$ matrix of ones.

The GG and LNN models characterize fluctuations in array data using a small number of parameters, and both involve the assumption of a constant coefficient of variation (CV). The appropriateness of the Gamma observation component can be checked

# 5 EBarrays

The main functions available in version 1.1 of EBarrays are:

| | |
|---|---|
| `createExprSet` | reads in data and records data characteristics |
| `ebPatterns` | generates expression patterns |
| `modeldiag` | generates diagnostic plots to check for a coefficient of variation and to check Gamma or Log-Normal assumption on observation component . |
| `emfit` | EM algorithm to fit the EB model |
| `postprob` | posterior probabilities for expression patterns |
| `margplot.gg` | generates GG predictive distribution and compares with data. |
| `margplot.lnn` | generates LNN predictive distribution and compares with data. |

These functions are supplemented by other functions that are not called directly by the user, including `complete.loglik`, which calculates the complete data log likelihood (??), and functions to evaluate the key predictive distribution $f_0$. The current package contains two possible forms for $f_0$, `f0gg` for the GG model and `f0lnn` for the LNN model. Note that other families can be added by the user.

Each analysis requires input files that contain the normalized intensities ($datafile$), identify the replicate samples ($repfile$), and specify the patterns to be considered ($patternfile$). EBarrays assumes that input files are tab or single-space delimited ASCII text files. $datafile$ contains intensities in $J$ rows and $I$ columns; row and column names should be provided. $repfile$ specifies which of the $I$ samples are considered replicates. It contains one row with $I$ columns $(r_1, r_2, \ldots, r_I)$, where $r_1 = 1$ and for $k = 2, 3, \ldots, I$, $r_k = r_{k-1}$ if samples $k$ and $k-1$ are considered replicates and $r_k = r_{k-1} + 1$ otherwise. $patternfile$ specifies the patterns to be considered in the analysis. The $k$th row identifies which samples are assumed to have the same mean level of expression in pattern $k$. Typically, the $(k, 1)$ element is 1 for every $k$ and for $l = 2, 3, \ldots, I$, $(k, l) = (k, l-1)$ if samples $l$ and $l-1$ are assumed to have the same mean level of expression in pattern $k$ and $(k, l) = (k, l-1) + 1$ otherwise.

As an illustration, consider a dataset with $I = 10$ arrays taken from two conditions (five arrays in each condition ordered so that the first five columns contain data from the first condition). In this case, there are two, possibly distinct, levels of expression for each gene and two potential patterns ($\mu_{j1} = \mu_{j2}$ and $\mu_{j1} \neq \mu_{j2}$). The replicate and pattern files are, respectively,

```
1 1 1 1 1 2 2 2 2 2
```

and

```
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 2 2 2 2 2
```

The exception to this convention is that zero columns can be used to identify arrays that are not used in model fitting or analysis. An example of this is given in the next section.

# 6 Application

In collaboration with Dr. M.N. Gould's laboratory in Madison, we have been investigating gene expression patterns of mammary epithelial cells in a rat model of breast cancer. We use a subset of the data from this study (1000 genes in 4 biological conditions; 10 arrays total) to illustrate the mixture model calculations and EBarrays. For details on the full data set and analysis, see Kendziorski *et al.* (2003).

We note that, in column order, there is one sample in condition 1, five samples in condition 2, two samples in condition 3, and two samples in condition 4. To reflect this arrangement, the *repfile* (called `reps.txt`) is

```
1 2 2 2 2 2 3 3 4 4
```

The last bit of input is the *patternfile*, which tells EBarrays what patterns of mean expression will be considered in the analysis. Let us first ignore conditions 2 and 3 and compare conditions 1 and 4. There are two possible expression patterns ($\mu_{Cond1} = \mu_{Cond4}$ and $\mu_{Cond1} \neq \mu_{Cond4}$). These are represented in the *patternfile* (called `patternfile.txt`)

```
1 0 0 0 0 0 0 0 0 1 1
1 0 0 0 0 0 0 0 0 2 2
```

A zero column indicates that the data in that condition are not considered in the analysis. An alternative approach would be to define a new data matrix containing intensities from conditions 1 and 4 only and define the associated *patternfile* as

```
1 1 1
1 2 2
```

The corresponding *repfile* would be

```
1 2 2
```

This may be useful in some cases, but in general we recommend importing the full data matrix and defining the pattern matrix as a $2 \times 10$ matrix with intermediate columns set to zero. Doing so facilitates comparisons of results among different analyses since certain attributes of the data, such as the number of genes that are positive across each condition, do not change.

The data file, (vigdata.txt), repfile (repfile.txt), and patternfile (patternfile.txt) are read into EBarrays using `createExprSet` and `ebPatterns`

```
exprdata <- createExprSet("vigdata.txt", "repfile.txt")
patterns <- ebPatterns("patternfile.txt")
```

Preliminary data analysis can be done using standard R functions. There are also diagnostics built into EBarrays. For example, `modeldiag` can be used to see if there is any relationship between the mean expression level and the coefficient of variation. Recall that both GG and LNN models assume a constant CV. In addition, `modeldiag` generates Gamma QQ plots and Log-Normal QQ plots for subsets of data sharing common empirical mean intensities.

Figures 1-3 were generated by

```
modeldiag(exprdata, nb=20)
```

`nb` specifies the number of genes to consider for each QQ plot.

Figure 1. Coefficient of variation (CV) as a function of the mean (ranked). Shaded vertical bars indicate nine subsets of (nb=20) genes, each examined more closely in Figures 2 and 3.

Figure 2: Gamma *qq* plots for the nine subsets shown in Figure 1.

Figure 3: Log-Normal *qq* plots for the nine subsets shown in Figure 1.

Figure 1 shows that the assumption of a constant coefficient of variation is reasonable for the small data set considered here, with a slight violation at very low and very high mean expression levels. (Similar results were obtained using the full data set.) Figure 2 shows a second diagnostic plot for nine subsets of $nb = 20$ genes spanning the range of mean expression. Shown are *qq* plots against the best-fitting Gamma distribution. The fit is reasonable here. Note that we only expect these *qq* plots to hold for equivalently expressed genes, so some violation is expected in general. Figure 3 shows the same diagnostic for the LNN model.

Using `emfit`, we can fit either the GG or the LNN model. We recommend fitting both for the sake of comparison. Posterior probabilities can then be obtained using `postprob`. The approach is illustrated below. Output is shown for 50 iterations. The default output from `emfit` gives thetaTrace and probTrace, which contain parameter estimates at each iteration. It is recommended that these be checked for convergence.

■emfit, plot=TRUE■

demo(ebarrays)

plotMarginal(testdata, em.out)

gg.em.out <- emfit(exprdata, family = "GG", patterns, verbose = TRUE, num.iter = 50, theta.init = c(7, 1, 1), p.init = c(0.95, 0.05))

gg.post.out<-postprob(gg.em.out,exprdata)

sum(gg.post.out[,2]>0.5)

lnn.em.out <- emfit(exprdata, family = "LNN", patterns, verbose = TRU num.iter=50,theta.init = c(7, 1, 1), p.init = c(0.95, 0.05))

lnn.post.out<-postprob(lnn.em.out,exprdata)

sum(lnn.post.out[,2]>0.5) sum(gg.post.out[,2]>0.5lnn.post.out[,2]>0.5)

Using 0.5 as the threshold posterior probability, there are 12 genes identified as most likely differentially expressed via the GG model and 52 via the LNN. Note that the 12 identified by the GG model are also identified by LNN. Further diagnostics are required to investigate model fit and to consider the 40 genes identified by the LNN but not by the GG. A plot of the marginal distributions under each model can be compared with the empirical distribution to further assess model fit. These plots are shown in Figures 4 and 5. Figure 4 was generated by `margplot.gg(exprdata,gg.em.out)` and Figure 5 by `margplot.lnn(exprdata,lnn.em.out)`. This visual comparison suggests that both models provide reasonable fits. Further comparisons are suggested.

Figure 4: A histogram of the intensity values (log scale) is shown along with the marginal distribution under the GG model (solid line).

Figure 5: A histogram of the intensity values (log scale) is shown along with the marginal distribution under the LNN model (solid line).

A nice feature of EBarrays is that comparisons among more than two groups can be carried out simply by changing the pattern matrix. For the four conditions, there are 15 possible expression patterns; however, for this case study, four were of most interest. The null pattern (pattern 1) consists of equivalent expression across the four conditions. The three other patterns allow for differential expression. Differential expression in condition 1 only is specified in pattern 2; DE in condition 4 only is specified in pattern 4.

The *repfile* is the same as before, but now the pattern matrix (`pattern4group.txt`) for the four group analysis is given by

```
1 1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 2
1 1 1 1 1 1 1 1 2 2
```

The data are imported using the same commands as in the comparison between two groups:

```
exprdata <- createExprSet("vigdata.txt", "repfile.txt")
patterns4group <- ebPatterns("pattern4group.txt")
```

`emfit` and `postprob` are also called as before.

```
gg4group.em.out<-emfit(exprdata,family="GG",patterns4group,verbose=T,num.iter=50,theta.

 Checking for negative entries...
         21 rows out of 1000 had at least one negative entry
         These rows will not be used in the EM fit

 Generating summary statistics for patterns.
 This may take a few seconds...

 Starting EM iterations (total 50 ).
 This may take a while

         Starting iteration 1 ...
         Starting iteration 2 ...
         Starting iteration 3 ...

         Starting iteration 48 ...
         Starting iteration 49 ...
         Starting iteration 50 ...
```

```
 Fit used 33.48 seconds user time


gg4group.post.out<-postprob(gg4group.em.out,exprdata)

lnn4group.em.out<-emfit(exprdata,family="LNN",patterns4group,verbose=T,num.iter=50,thet

 Checking for negative entries...
        21 rows out of 1000 had at least one negative entry
        These rows will not be used in the EM fit

 Generating summary statistics for patterns.
 This may take a few seconds...

 Starting EM iterations (total 50 ).
 This may take a while

        Starting iteration 1 ...
        Starting iteration 2 ...
        Starting iteration 3 ...

        Starting iteration 48 ...
        Starting iteration 49 ...
        Starting iteration 50 ...


Fit used 143.09 seconds user time
```

The output from postprob is now a matrix with number of rows equal to the number of genes and number of columns equal to 4 (one for each pattern considered). A brief look at the output matrices shows that 8 genes are identified as DE (using 0.5 as the threshold posterior probability) under the GG model and 24 under the LNN model. The 8 identified by GG are also identified by LNN. Their gene id's are shown.

```
> sum(gg4group.post.out[,2]>0.5)
[1] 8
> sum(lnn4group.post.out[,2]>0.5)
[1] 24
> sum(gg4group.post.out[,2]>0.5&lnn4group.post.out[,2]>0.5)
[1] 8
```

```
> gene.ids<-(1:1000)[gg4group.post.out[,2]>0.5&lnn4group.post.out[,2]>0.5]

> gene.ids
[1] 125 139 149 388 510 752 851 854
```

# 7 References


1. Kendziorski CM, Newton MA, Lan H, Gould MN (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, in press, 2003.

2. Newton MA, Kendziorski CM, Richmond CS, Blattner FR (2001). On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, 8:37-52.

3. Newton, M.A. and C.M. Kendziorski. Parametric Empirical Bayes Methods for Microarrays in *The analysis of gene expression data: methods and software*. Eds. G. Parmigiani, E.S. Garrett, R. Irizarry and S.L. Zeger, New York: Springer Verlag, 2003.