Parametric Empirical Bayes Methods for Microarray Data

Christina Kendziorski, Deepayan Sarkar, Meng Chen, and Michael Newton

February 25, 2004

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 R package EBarrays. 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 within 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 μ_j and, more specifically, as arising from an observation distribution $f_{obs}(\cdot|\mu_j)$.

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, EE_j, for gene j. Alternatively, there is differential expression, DE_j; our formulation requires that there now be two different means, say μ_{j1} and μ_{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 μ_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 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 μ_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 pattern), altered expression in just one condition (3 patterns), and distinct expression in each condition (1 pattern). We view a pattern of expression for a gene j as a grouping or clustering of conditions so that the mean level μ_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); 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. In particular, two forms for f_0 are currently specified (see section 4), but other assumptions can be considered simply by providing alternative forms for f_0.

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 θ 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 μ_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 α, the quantity $\lambda = \alpha/\mu_j$ has a Gamma distribution with shape parameter α_0 and scale parameter ν. 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 μ_j is a mean for the log-transformed measurements, which are presumed to have a normal distribution with common variance σ^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 μ_j is normal with some underlying mean μ_0 and variance τ_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

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

where $\mathbf{I}_n$ is an $n \times n$ identity matrix and $\mathbf{M}_n$ 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 these assumptions can be checked.

5 EBarrays

The EBarrays package can be loaded by

```
> library(EBarrays)
```

```
Loading required package: Biobase
Welcome to Bioconductor
         Vignettes contain introductory material.  To view,
         simply type: openVignette()
         For details on reading vignettes, see
         the openVignette help page.
```

The main user visible functions available in EBarrays are:

emfit	fits the EB model using an EM algorithm
postprob	generates posterior probabilities for expression patterns
plotMarginal	generates predictive marginal distribution from fitted model and compares with estimated marginal (kernel) density of the data
ebPatterns	generates expression patterns
checkCCV	diagnostic plot to check for constant coefficient of variation
checkModel	generates diagnostic plots to check Gamma or Log-Normal assumption on observation component

along with some other utility functions. The form of the parametric model is specified as an argument to emfit, which can be an object of formal class "ebarraysFamily". These objects are built into EBarrays for the GG and LNN models described above. It is possible to create new instances, using the description given in help("ebarraysFamily-class").

The data can be supplied either as a matrix, or as an "exprSet" object. It is expected that the data be normalized intensity values, with rows representing genes and columns representing chips. Furthermore, the data must be on the raw scale (not on a logarithmic scale). All rows that contain at least one negative value are omitted from the analysis.

The columns of the data matrix are assumed to be grouped into a few experimental conditions. The columns (arrays) within a group are assumed to be replicates obtained under the same experimemntal conditions, and thus to have the same mean expression level across arrays for each gene. This information is usually contained in the "phenodata" slot of an "exprSet" object.

As an example, consider a hypothetical 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, the phenodata can be represented as

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

Thus, there are two, possibly distinct, levels of expression for each gene and two potential patterns or hypotheses concerning its expression levels: $\mu_{j1} = \mu_{j2}$ and $\mu_{j1} \neq \mu_{j2}$. These patterns can be denoted by

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

representing, in this simple case, equivalent and differential expression for a gene respectively. The choice of admissible patterns is critical in defining the model we try to fit. EBarrays has a function ebPatterns that can read pattern definitions from an external file or a character vector that supplies this information in the above notation. For example,

```
> pattern <- ebPatterns(c("1, 1, 1, 1, 1, 1, 1, 1, 1, 1",
+     "1, 1, 1, 1, 1, 2, 2, 2, 2, 2"))
> pattern

 Collection of 2 patterns

    Pattern 1 has 1 group
       Group  1: 1 2 3 4 5 6 7 8 9 10

    Pattern 2 has 2 groups
       Group  1: 1 2 3 4 5
       Group  2: 6 7 8 9 10
```

As discussed below, such patterns can be more complicated in general. For experiments with more than two groups, there can be many more patterns. Zeros can be used in this notation to identify arrays that are not used in model fitting or analysis.

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. EBarrays contains part of a dataset from this study (5000 genes in 4 biological conditions; 10 arrays total) to illustrate the mixture model calculations. For details on the full data set and analysis, see Kendziorski *et al.* (2003).

The data can be read in by

```
> data(gould)
```

The experimental information on this data are as follows: in column order, there is one sample in condition 1, two samples in condition 2, five samples in condition 3, and two samples in condition 4:

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

Before we proceed with the analysis, we need to tell EBarrays what patterns of mean expression will be considered in the analysis. Let us first ignore conditions 3 and 4 and compare conditions 1 and 2. There are two possible expression patterns ($\mu_{Cond1} = \mu_{Cond2}$ and $\mu_{Cond1} \neq \mu_{Cond2}$). This information can be entered as character strings above, or they could also be read from a *patternfile* which contains the following lines:

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

A zero column indicates that the data in that condition are not considered in the analysis. The patterns are entered as

```
> pattern <- ebPatterns(c("1,1,1,0,0,0,0,0,0,0",
+     "1,2,2,0,0,0,0,0,0,0"))
> pattern
```

```
 Collection of 2 patterns

    Pattern 1 has 1 group
       Group  1: 1 2 3

    Pattern 2 has 2 groups
       Group  1: 1
       Group  2: 2 3
```

An alternative approach would be to define a new data matrix containing intensities from conditions 1 and 2 only and define the associated patterns

```
1 1 1
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 2×10 matrix with the last seven 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.

Preliminary data analysis can be done using standard R and Bioconductor functions. There are also diagnostics built into EBarrays. For example, the checkCCV function 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.

Another way to assess the goodness of the parametric model is to look at Gamma or Log-Normal QQ plots for subsets of the data sharing common empirical mean intensities. For this, we can choose a small number of locations for the mean value, and look at the QQ plots for the subset of measured intensities in a small window around each of those locations.

Figure 1 shows that the assumption of a constant coefficient of variation is reasonable for the small data set considered here. If necessary, the data could be transformed based on this cv plot prior to analysis. Figure 2 shows a second diagnostic plot for nine subsets of $nb = 100$ 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 10 iterations. The output from emfit by default contains slots called thetaTrace and probTrace, which contain parameter estimates at each iteration. It is recommended that these be checked for convergence.

```
> gg.em.out <- emfit(gould, family = "GG",
+     hypotheses = pattern, num.iter = 10)
> gg.em.out

 EB model fit
         Family: GG ( Gamma-Gamma )

 Model parameter estimates:
```

```
> checkCCV(gould[, 1:3])
```

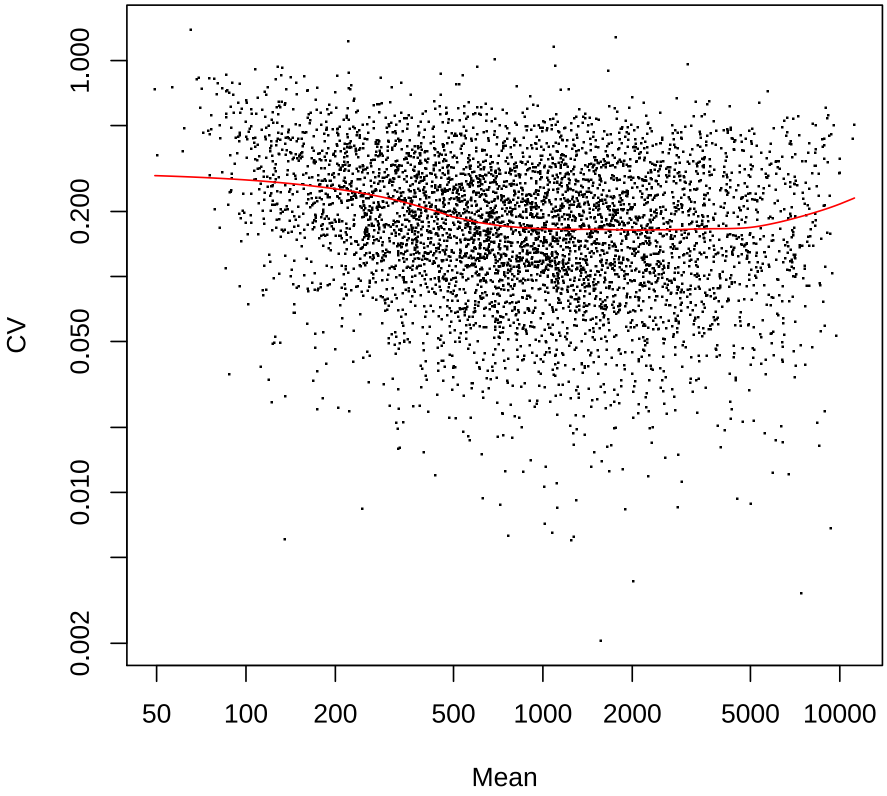

Figure 1: Coefficient of variation (CV) as a function of the mean.

```
> print(checkModel(gould, model = "gamma",
+     nb = 100))
```

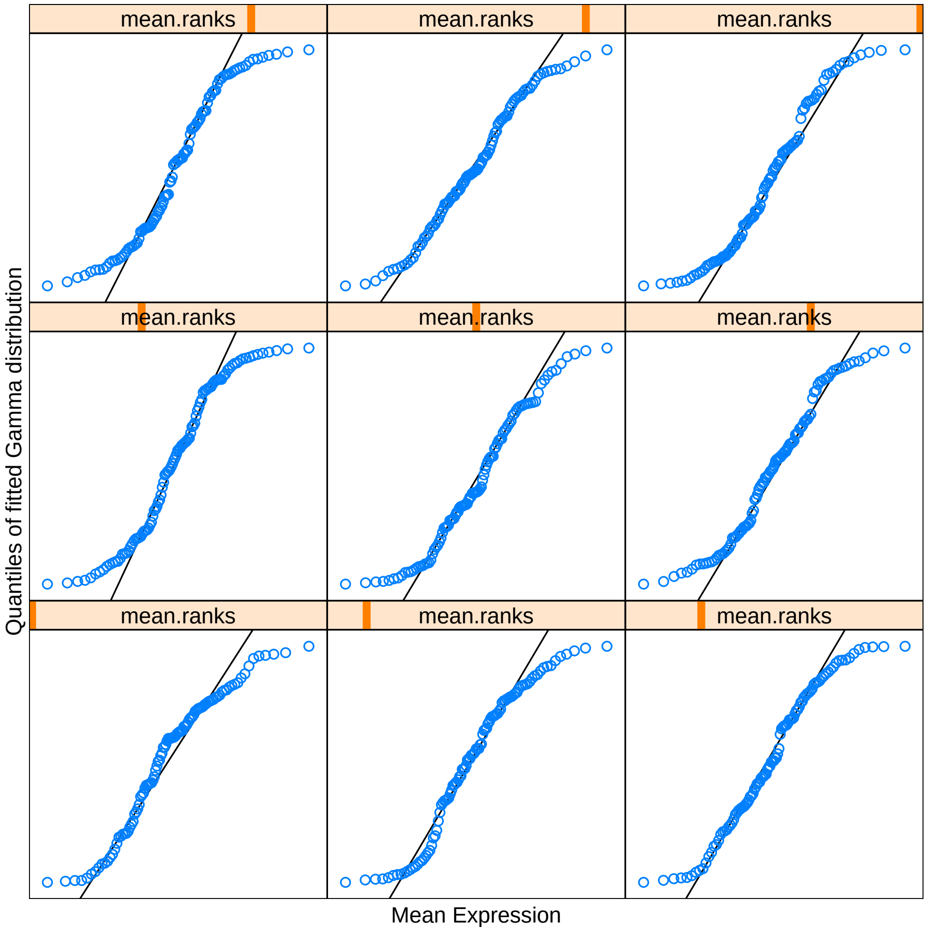

Figure 2: Gamma qq plot.

```
> print(checkModel(gould, model = "lognormal",
+     nb = 100))
```

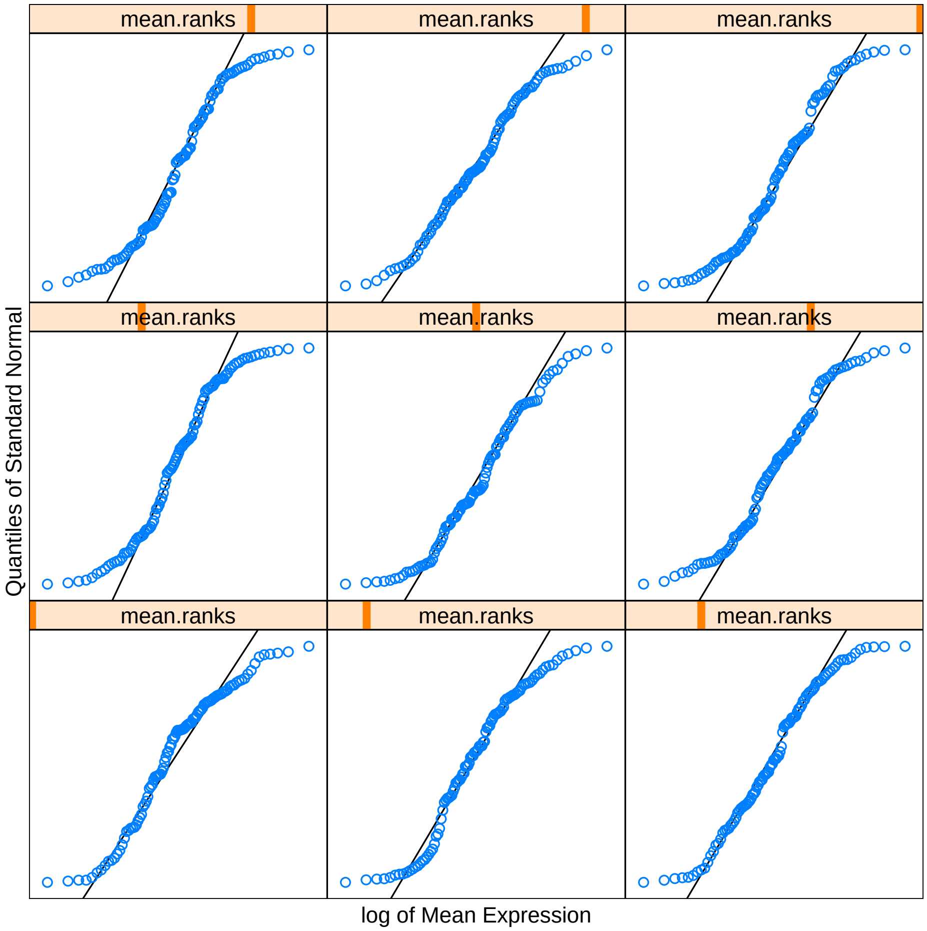

Figure 3: Log-Normal qq plot.

```
    alpha    alpha0        nu
13.262687  1.107481 43.729655

 Estimated mixing proportions:

         p1          p2
0.997019899 0.002980101


 Additional slots: @hypotheses, @thetaTrace, @probTrace

> gg.post.out <- postprob(gg.em.out,
+     gould)
> sum(gg.post.out[, 2] > 0.5)

[1] 8

> lnn.em.out <- emfit(gould, family = "LNN",
+     pattern, num.iter = 10)
> lnn.em.out

 EB model fit
         Family: LNN ( Lognormal-Normal )

 Model parameter estimates:

      mu_0    sigma^2    tao_0^2
6.73320399 0.08110462 1.13608252

 Estimated mixing proportions:

         p1          p2
0.993318883 0.006681117


 Additional slots: @hypotheses, @thetaTrace, @probTrace

> lnn.post.out <- postprob(lnn.em.out,
+     gould)
> sum(lnn.post.out[, 2] > 0.5)

[1] 19
```

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

[1] 6
```

Using 0.5 as the threshold posterior probability, there are 8 genes identified as most likely differentially expressed via the GG model and 19 via the LNN. Note that 6 of the 8 identified by the GG model are also identified by LNN. Further diagnostics are required to investigate model fit and to consider the 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. Figure 4 shows this plot for the Gamma-Gamma model, and Figure 5 for the Lognormal-Normal model. This visual comparison suggests that the LNN model provides a better fit. Additional diagnostics can be useful.

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 pattern matrix for the four group analysis is now given by

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

```
> pattern4 <- ebPatterns(c("1, 1, 1, 1, 1, 1, 1, 1, 1, 1",
+     "1, 2, 2, 2, 2, 2, 2, 2, 2, 2",
+     "1,2,2,1,1,1,1,1,2,2", "1,1,1,1,1,1,1,1,2,2"))
> pattern4

 Collection of 4 patterns

    Pattern 1 has 1 group
       Group  1: 1 2 3 4 5 6 7 8 9 10

    Pattern 2 has 2 groups
       Group  1: 1
       Group  2: 2 3 4 5 6 7 8 9 10
```

```
> print(plotMarginal(gg.em.out, gould[,
+     1:3]))
```

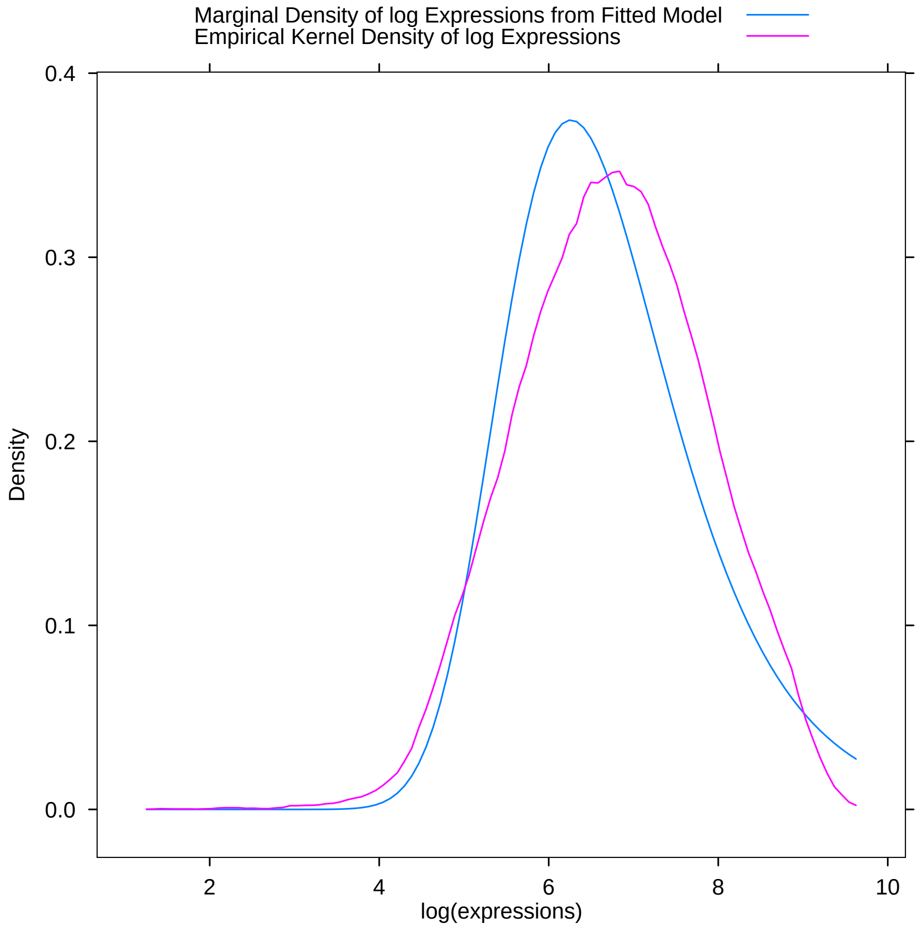

Figure 4: Empirical and theoretical marginal densities of log expressions for the Gamma-Gamma model.

```
> print(plotMarginal(lnn.em.out,
+     gould[, 1:3]))
```

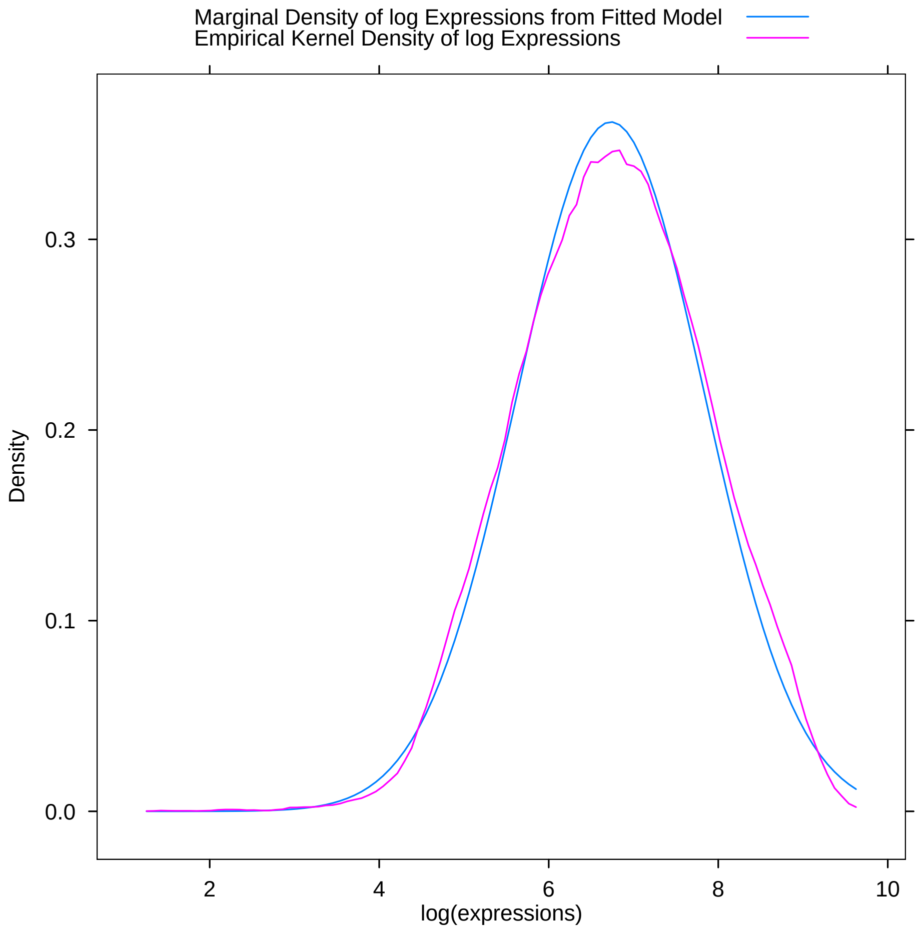

Figure 5: Marginal densities for Lognormal-Normal model

```
    Pattern 3 has 2 groups
       Group  1: 1 4 5 6 7 8
       Group  2: 2 3 9 10

    Pattern 4 has 2 groups
       Group  1: 1 2 3 4 5 6 7 8
       Group  2: 9 10
```

emfit and postprob are called as before.

```
> gg4.em.out <- emfit(gould, family = "GG",
+     pattern4, num.iter = 10)
> gg4.em.out

 EB model fit
         Family: GG ( Gamma-Gamma )

 Model parameter estimates:

    alpha    alpha0        nu
17.870503  1.079342 30.534732

 Estimated mixing proportions:

          p1           p2           p3
0.9784667912 0.0182813122 0.0023965530
          p4
0.0008553436


 Additional slots: @hypotheses, @thetaTrace, @probTrace

> gg4.post.out <- postprob(gg4.em.out,
+     gould)
> lnn4.em.out <- emfit(gould, family = "LNN",
+     pattern4, num.iter = 10)
> lnn4.em.out

 EB model fit
         Family: LNN ( Lognormal-Normal )

 Model parameter estimates:
      mu_0    sigma^2    tao_0^2
6.72002011 0.06051488 1.16232054

 Estimated mixing proportions:

         p1          p2          p3
0.974779403 0.020443944 0.003031844
         p4
0.001744809


 Additional slots: @hypotheses, @thetaTrace, @probTrace
```

```
> lnn4.post.out <- postprob(lnn4.em.out,
+     gould)
```

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 51 genes are identified as being in pattern 2 (using 0.5 as the threshold posterior probability) under the GG model and 61 under the LNN model; 45 of the 51 genes identified by GG are also identified by LNN. Gene id's for 10 are shown. Figures 6 and 7 show marginal plots similar to Figures 4 and 5.

```
> sum(gg4.post.out[, 2] > 0.5)

[1] 51

> sum(lnn4.post.out[, 2] > 0.5)

[1] 61

> sum(gg4.post.out[, 2] > 0.5 & lnn4.post.out[,
+     2] > 0.5)

[1] 45

> gene.ids <- geneNames(gould)[gg4.post.out[,
+     2] > 0.5 & lnn4.post.out[,
+     2] > 0.5]
> gene.ids[1:10]
```

```
 [1] "X59864mRNA.at"
 [2] "D26307cds.at"
 [3] "AF039583.s.at"
 [4] "J05232cds.s.at"
 [5] "J04503.g.at"
 [6] "AB003753cds.2.at"
 [7] "M12822cds.f.at"
 [8] "rc.AA859768.at"
 [9] "S77492.i.at"
[10] "S77528cds.s.at"
```

```
> print(plotMarginal(gg4.em.out,
+     data = gould))
```

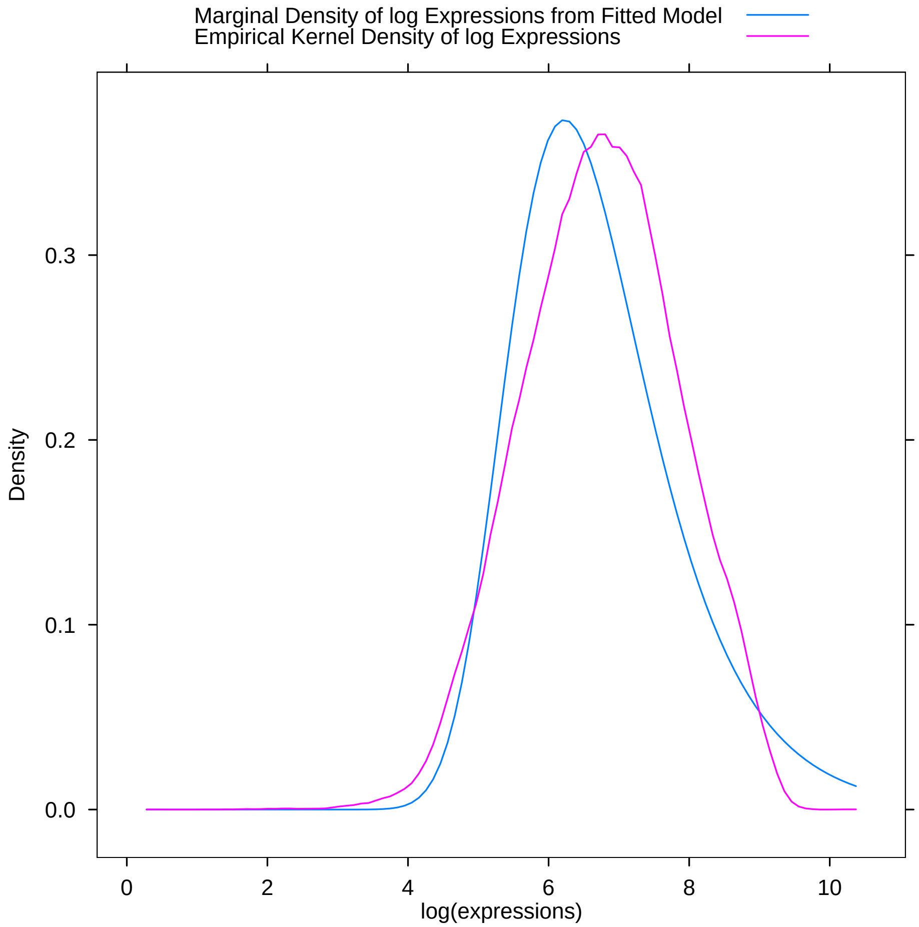

Figure 6: Marginal densities for Gamma-Gamma model

```
> print(plotMarginal(lnn4.em.out,
+     data = gould))
```

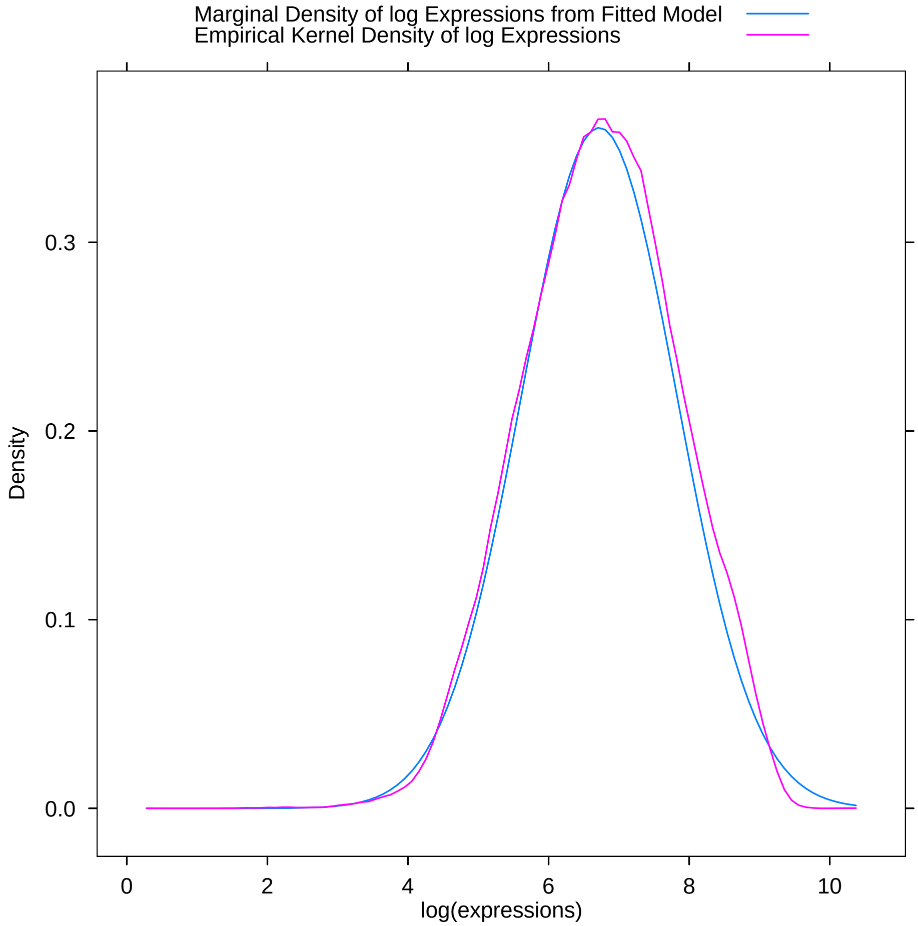

Figure 7: Marginal densities for Lognormal-Normal model

7 Appendix: older versions of EBarrays

The interface for EBarrays has changed considerably since earlier versions, which required, among other things, the data file to be in a particular format. EBarrays contains a utility function createExprSet that reads such a file into an object of class "exprSet". Details are described in the help page for createExprSet.

8 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*, 22:3899-3914.

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.