Parametric Empirical Bayes Methods for Microarrays

Ming Yuan, Deepayan Sarkar, Michael Newton
and Christina Kendziorski

October 24, 2025

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 1.7 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_{ji} 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_c contains the indices for samples in group c. 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) in which case our formulation requires that there now be two different means, 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 equivalently 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 DE, 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 proportion of genes differentially expressed 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 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 with four 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, three 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, to classify genes into distinct expression patterns and to estimate FDR.

4 The Three Models

We consider three 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.

4.1 The Gamma Gamma Model

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 $\lambda_j = \alpha/\mu_j$,

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

for measurements $x > 0$. Note that the coefficient of variation (CV) 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 $\lambda_j = \alpha/\mu_j$ 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

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

4.2 The Lognormal Normal Model

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 CV: $\sqrt{\exp(\sigma^2) - 1}$ 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.

4.3 The Lognormal Normal with Modified Variance Model

In the lognormal normal with modified variance (LNNMV) model, the log-transformed measurements are presumed to have a normal distribution with gene-specific mean μ_j and gene-specific variance σ_j^2. With the same prior for the μ_j as in the LNN model, the density $f_0(\cdot)$ for an n-dimensional input from gene j becomes Gaussian with mean vector $\underline{\mu}_0 = (\mu_0, \mu_0, \ldots, \mu_0)^t$ and exchangeable covariance matrix

$$\mathbf{\Sigma}_n = \left(\sigma_j^2\right) \mathbf{I}_n + \left(\tau_0{}^2\right) \mathbf{M}_n,$$

Thus, only two model parameters are involved once the σ_j^2's are estimated.

In the special case of two conditions, we illustrate how to estimate the σ_j^2's. We assume that the prior distribution of σ_j^2 is the scaled inverse chi-square distribution with ν_0 degrees of freedom and scale parameter σ_0 and that μ_{j1} and μ_{j2}, the gene-specific means in condition 1 and 2, are known. Then the posterior distribution of σ_j^2 is also the scaled inverse chi-square distribution with $n_1 + n_2 + \nu_0$ degrees of freedom and scale parameter

$$\sqrt{\frac{\nu_0\sigma_0^2 + \sum_{i=1}^{n_1}(x_{ji} - \mu_{j1})^2 + \sum_{i=1}^{n_2}(y_{ji} - \mu_{j2})^2}{n_1 + n_2 + \nu_0}},$$

where x_{ji} is the log-transformed measurement in condition 1, j indexes gene, i indexes sample; y_{ji} is the log-transformed measurement in condition 2; n_1, n_2 are the numbers of samples in condition 1, 2 respectively. By viewing the pooled sample variances

$$\tilde{\sigma}_j^2 = \frac{\sum_{i=1}^{n_1}(x_{ji} - \bar{x}_j)^2 + \sum_{i=1}^{n_2}(y_{ji} - \bar{y}_j)^2}{n_1 + n_2 - 2}$$

as a random sample from the prior distribution of σ_j^2, we can get $(\hat{\nu}_0, \hat{\sigma}_0^2)$, the estimate of (ν_0, σ_0^2) using the method of moments [5]. Then our estimate of σ_j^2 is

$$\hat{\sigma}_j^2 = \frac{\hat{\nu}_0\hat{\sigma}_0^2 + \sum_{i=1}^{n_1}(x_{ji} - \bar{x}_j)^2 + \sum_{i=1}^{n_2}(y_{ji} - \bar{y}_j)^2}{n_1 + n_2 + \hat{\nu}_0 - 2},$$

the posterior mean of σ_j^2 with ν_0, σ_0^2, μ_{j1} and μ_{j2} substituted by $\hat{\nu}_0$, $\hat{\sigma}_0^2$, $\bar{x}_j$ and $\bar{y}_j$, where $\bar{x}_j = \frac{1}{n_1}\sum_{i=1}^{n_1} x_{ji}$ and $\bar{y}_j = \frac{1}{n_2}\sum_{i=1}^{n_2} y_{ji}$.

The GG, LNN and LNNMV models characterize fluctuations in array data using a small number of parameters. The GG and LNN models both involve the assumption of a constant CV. The appropriateness of this assumption can be checked. The LNNMV model relaxes this assumption, allowing for a gene specific variance estimate. Our studies show little loss in efficiency with the LNNMV model, and we therefore generally recommend its use over GG or LNN.

5 EBarrays

The EBarrays package can be loaded by

```
> library(EBarrays)
```

The main user visible functions available in EBarrays are:

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

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, LNN and LNNMV 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 "ExpressionSet" 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 experimental conditions, and thus to have the same mean expression level across arrays for each gene. This information is usually contained in the "phenoData" from an "ExpressionSet" 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
```

and 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, as we demonstrate below.

6 Case Study

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 expression will be considered in the analysis. Recall that 15 patterns are possible. 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, 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.

The function emfit can be used to fit the GG, LNN or LNNMV model. Posterior probabilities can then be obtained using postprob. The approach is illustrated below. Output is shown for 10 iterations.

```
> 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:

             alpha   alpha0       nu
```

```
Cluster 1 13.26269 1.107481 43.72966

 Estimated mixing proportions:

          Pattern.1   Pattern.2
Cluster 1 0.9970199 0.002980101

> gg.post.out <- postprob(gg.em.out, gould)$pattern
> gg.threshold <- crit.fun(gg.post.out[,1], 0.05)
> gg.threshold

[1] 0.9432019

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

[1] 4

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

 EB model fit
        Family: LNN ( Lognormal-Normal )

 Model parameter estimates:

              mu_0    sigma.2  tao_0.2
Cluster 1 6.733204 0.08110462 1.136083

 Estimated mixing proportions:

          Pattern.1   Pattern.2
Cluster 1 0.9933189 0.006681117

> lnn.post.out <- postprob(lnn.em.out, gould)$pattern
> lnn.threshold <- crit.fun(lnn.post.out[,1], 0.05)
> lnn.threshold

[1] 0.7707577

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

[1] 12
```

```
> lnnmv.em.out <- emfit(gould, family = "LNNMV", hypotheses = pattern,
+                       groupid = c(1,2,2,0,0,0,0,0,0,0), num.iter = 10)
> lnnmv.em.out

 EB model fit
        Family: LNNMV ( Lognormal-Normal with modified variances )

 Model parameter estimates:

     mu_0  tao_0.2
1 6.735102 1.131062

 Estimated mixing proportions:

       Pattern.1   Pattern.2
p.temp 0.9929931 0.007006875

> lnnmv.post.out <- postprob(lnnmv.em.out, gould,
+                            groupid = c(1,2,2,0,0,0,0,0,0,0))$pattern
> lnnmv.threshold <- crit.fun(lnnmv.post.out[,1], 0.05)
> lnnmv.threshold

[1] 0.7919131

> sum(lnnmv.post.out[,2] > lnnmv.threshold)

[1] 14

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

[1] 3

> sum(gg.post.out[,2] > gg.threshold & lnnmv.post.out[,2] > lnnmv.threshold)

[1] 2

> sum(lnn.post.out[,2] > lnn.threshold & lnnmv.post.out[,2] > lnnmv.threshold)

[1] 7

> sum(gg.post.out[,2] > gg.threshold & lnn.post.out[,2] > lnn.threshold
+     & lnnmv.post.out[,2] > lnnmv.threshold)

[1] 2
```

>

The posterior probabilities can be used as described in Newton *et al.* (2004) to create a list of genes with a target FDR. Using 0.05 as the target FDR, 3, 12 and 14 genes are identified as most likely DE via the GG model, the LNN model and the LNNMV model, respectively. Note that all 3 DE genes identified by the GG model are also identified by LNN; 2 genes are identified as DE by both GG and LNNMV; 7 genes are identified as DE by both LNN and LNNMV; and all three methods agrees on 2 genes being DE. Further diagnostics are required to investigate model fit and to consider the genes identified as DE by only one or 2 models.

When using the GG or LNN model, the function checkCCV can be used to see if there is any relationship between the mean expression level and the CV. Another way to assess the goodness of the parametric model is to look at Gamma or 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. Since the LNNMV model assumes a log-normal observation component, this diagnostic is suggested when using LNNMV model as well. A third diagnostic consists of plotting the marginal distributions of each model and comparing with the empirical distribution to further assess model fit. This diagnostic is designed especially for the GG and LNN models since their marginal distributions are not gene specific. Finally, checkVarsQQ and checkVarsMar are two diagnostics used to check the assumption of a scaled inverse chi-square prior on σ_j^2, made in the LNNMV model. checkVarsQQ generates a QQ plot of the quantiles of gene specific sample variances against the quantiles of the scaled inverse chi-square distribution with parameters estimated from data. checkVarsMar plots the density histogram of gene specific sample variances and the density of the scaled inverse chi-square distribution with parameters estimated from the data.

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 $nb = 50$ 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. Figures 3 and 4 show the same diagnostics for the LNN model and the LNNMV model, respectively. A visual comparison between figure 5 and figure 6 suggests that the LNN model provides a better fit. Finally, figure 7 and figure 8 provide checks for the assumption of a scaled inverse chi-square prior on gene specific variances. The LNN model seems most appropriate here. Figure 8 demonstrates that the assumptions of LNNMV are not satisfied, perhaps due to the small sample size ($n = 3$) used to estimate each gene specific variance.

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

    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
Cluster 1 17.89911 1.077009 30.36874

 Estimated mixing proportions:

          Pattern.1  Pattern.2   Pattern.3
Cluster 1 0.9780804 0.01864805 0.002403049
            Pattern.4
Cluster 1 0.0008684683

> gg4.post.out <- postprob(gg4.em.out, gould)$pattern
> gg4.threshold2 <- crit.fun(1-gg4.post.out[,2], 0.05)
> sum(gg4.post.out[,2] > gg4.threshold2)

[1] 40

> 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
Cluster 1 6.72002 0.06051488 1.162321

 Estimated mixing proportions:

          Pattern.1  Pattern.2   Pattern.3
Cluster 1 0.9747794 0.02044394 0.003031844
            Pattern.4
Cluster 1 0.001744809

> lnn4.post.out <- postprob(lnn4.em.out, gould)$pattern
> lnn4.threshold2 <- crit.fun(1-lnn4.post.out[,2], 0.05)
> sum(lnn4.post.out[,2] > lnn4.threshold2)
```

```
[1] 45

> lnnmv4.em.out <- emfit(gould, family="LNNMV", pattern4,
+                        groupid = c(1,2,2,3,3,3,3,3,4,4), num.iter = 10)
> lnnmv4.em.out

 EB model fit
         Family: LNNMV ( Lognormal-Normal with modified variances )

 Model parameter estimates:

      mu_0  tao_0.2
1 6.725249 1.146925

 Estimated mixing proportions:

       Pattern.1  Pattern.2    Pattern.3
p.temp 0.9619806 0.03731805 0.0001265073
          Pattern.4
p.temp 0.0005748571

> lnnmv4.post.out <- postprob(lnnmv4.em.out, gould,
+                             groupid = c(1,2,2,3,3,3,3,3,4,4))$pattern
> lnnmv4.threshold2 <- crit.fun(1-lnnmv4.post.out[,2], 0.05)
> sum(lnnmv4.post.out[,2] > lnnmv4.threshold2)

[1] 76

> sum(gg4.post.out[,2] > gg4.threshold2 & lnn4.post.out[,2] > lnn4.threshold2)

[1] 34

> sum(gg4.post.out[,2] > gg4.threshold2 & lnnmv4.post.out[,2] > lnnmv4.threshold2)

[1] 25

> sum(lnn4.post.out[,2] >  lnn4.threshold2 & lnnmv4.post.out[,2] > lnnmv4.threshold2)

[1] 29

> sum(gg4.post.out[,2] > gg4.threshold2 & lnn4.post.out[,2] > lnn4.threshold2
+     & lnnmv4.post.out[,2] > lnnmv4.threshold2)

[1] 22
```

The component pattern 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 40 genes are identified as being in pattern 2 using 0.05 as the target FDR under the GG model, 45 under the LNN model and 76 under the LNNMV model; 34 of the 40 genes identified by GG are also identified by LNN. 25 genes are commonly identified by GG and LNNMV. 29 genes are commonly identified by LNN and LNNMV. 22 genes are identified by all the three models. Figures 9 and 10 show marginal plots similar to Figures 5 and 6. Figure 11 shows a plot similar to figure 8. Here the violation of assumptions made in the LNNMV model is not as severe.

7 Appendix: Comparison with the older versions of EBarrays

Major changes in this version include:

1. A new model LNNMV has been added, as described here.

2. Ordered patterns can be considered. Instead of only considering $\mu_{j1} \neq \mu_{j2}$, $\mu_{j1} > \mu_{j2}$ and $\mu_{j1} < \mu_{j2}$ can also be considered in the current version. For details, please refer to the help page for ebPatterns and Yuan and Kendziorski (2006).

3. Both clustering and DE identification can be done simultaneously by specifying the number of clusters in emfit. For details, please refer to the help page for emfit and Yuan and Kendziorski (2006).

8 References

1. Kendziorski, C.M., Newton, M.A., Lan, H., Gould, M.N. (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, 22:3899-3914.

2. Newton, M.A., Kendziorski, C.M., Richmond, C.S., Blattner, F.R. (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 Kendziorski, C.M. 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.

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

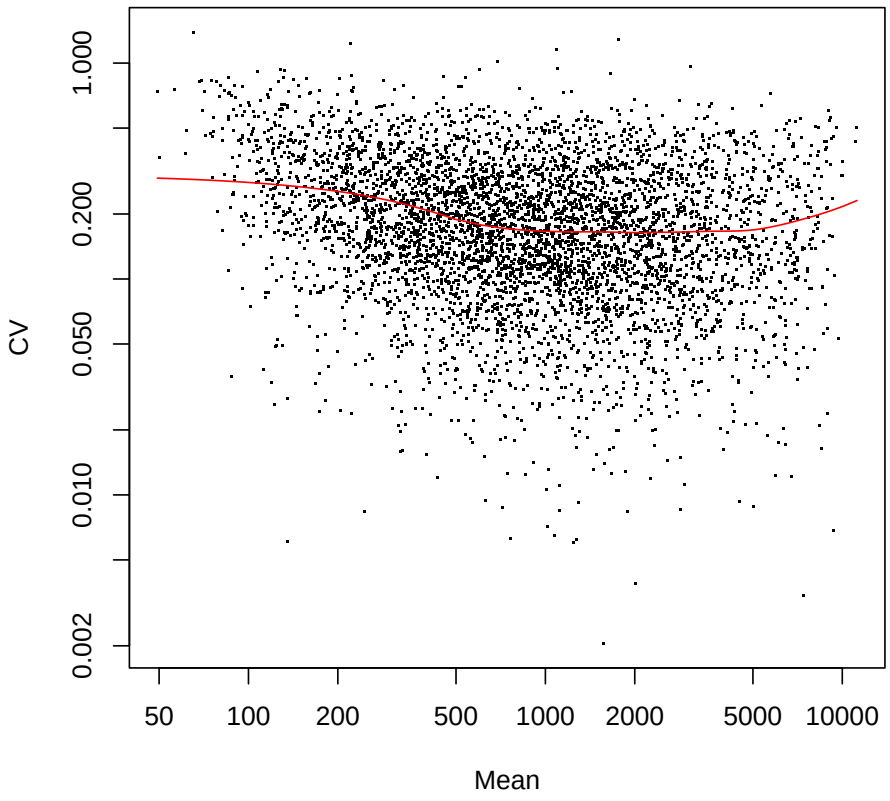

Figure 1: Coefficient of variation (CV) as a function of the mean.

```
> print(checkModel(gould, gg.em.out, model = "gamma", nb = 50))
```

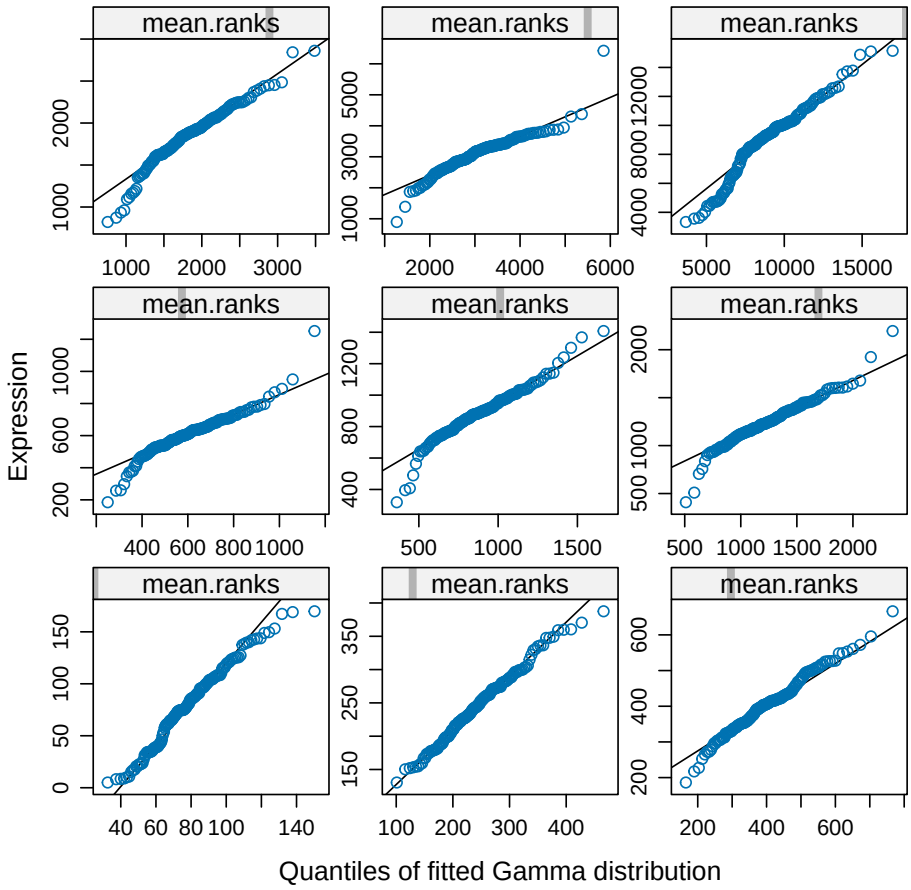

Figure 2: Gamma qq plot.

```
> print(checkModel(gould, lnn.em.out, model = "lognormal", nb = 50))
```

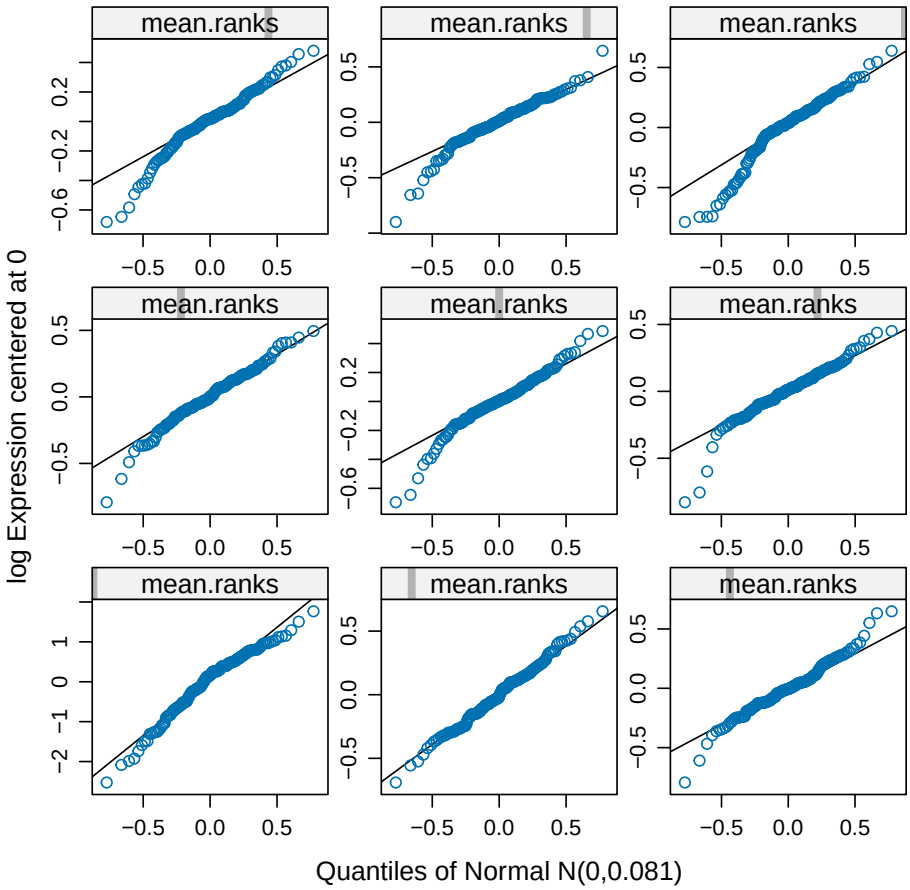

Figure 3: Normal qq plot.

```
> print(checkModel(gould, lnnmv.em.out, model = "lnnmv", nb = 50,
+                  groupid = c(1,2,2,0,0,0,0,0,0,0)))
```

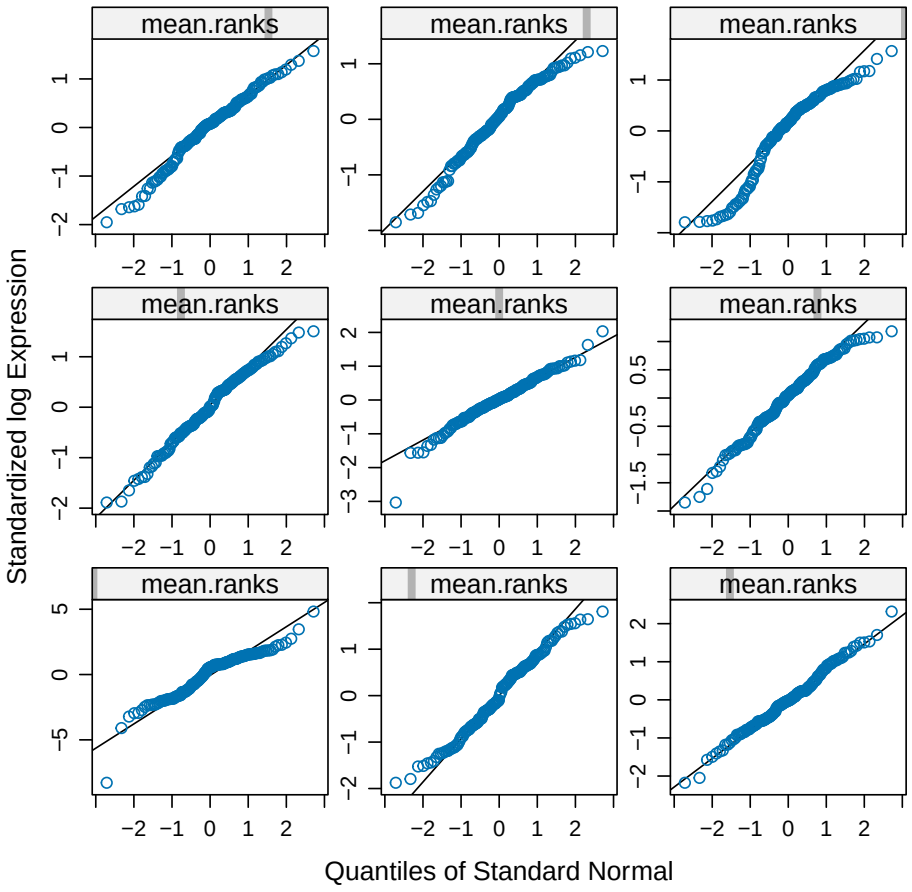

Figure 4: Normal qq plot.

```
> print(plotMarginal(gg.em.out, gould))
```

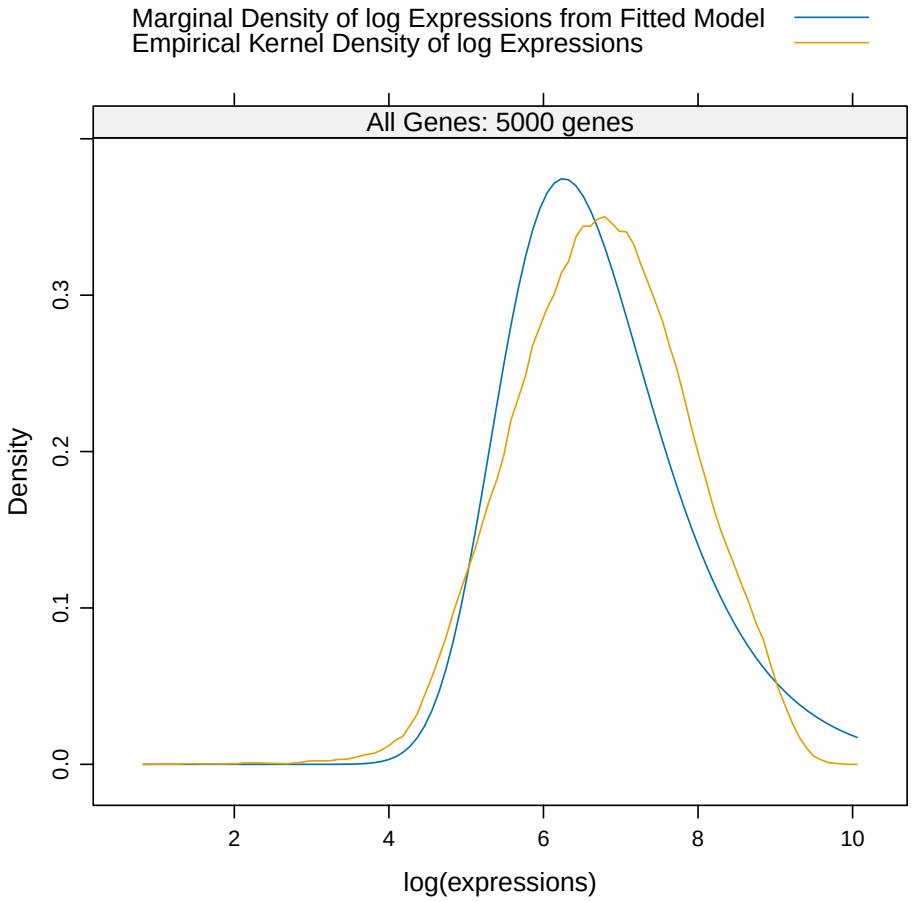

Figure 5: Empirical and theoretical marginal densities of log expressions for the Gamma-Gamma model.

```
> print(plotMarginal(lnn.em.out, gould))
```

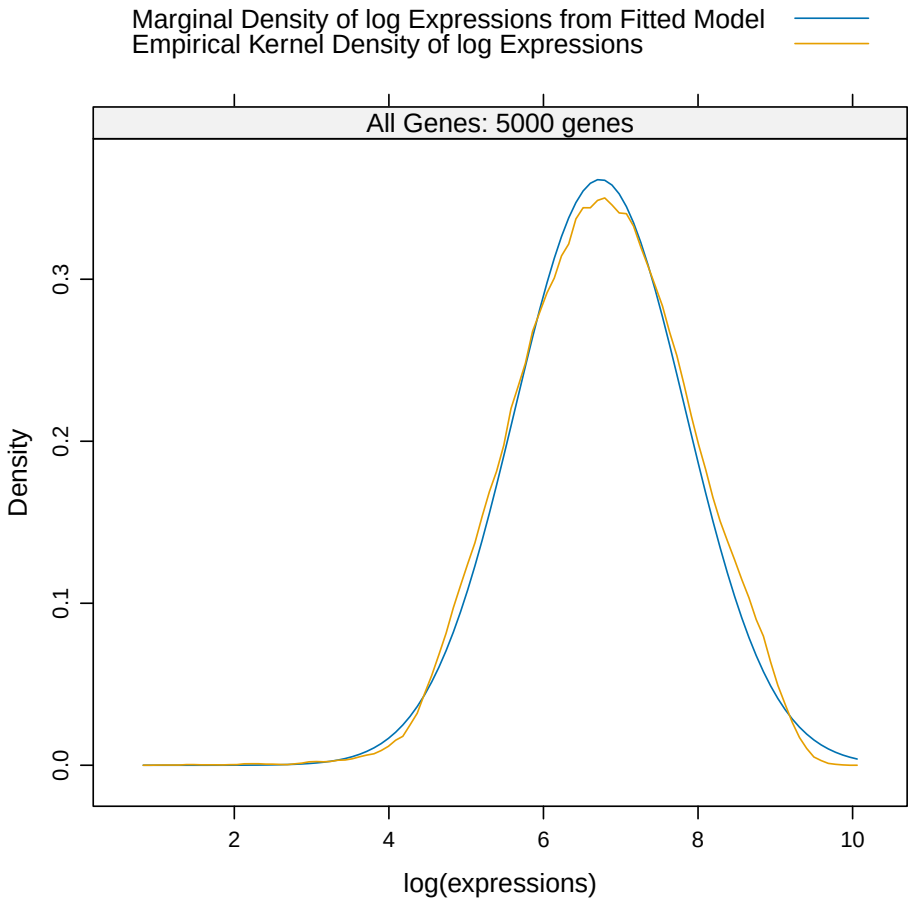

Figure 6: Empirical and theoretical marginal densities of log expressions for the Lognormal-Normal model

```
> print(checkVarsQQ(gould, groupid = c(1,2,2,0,0,0,0,0,0,0)))
```

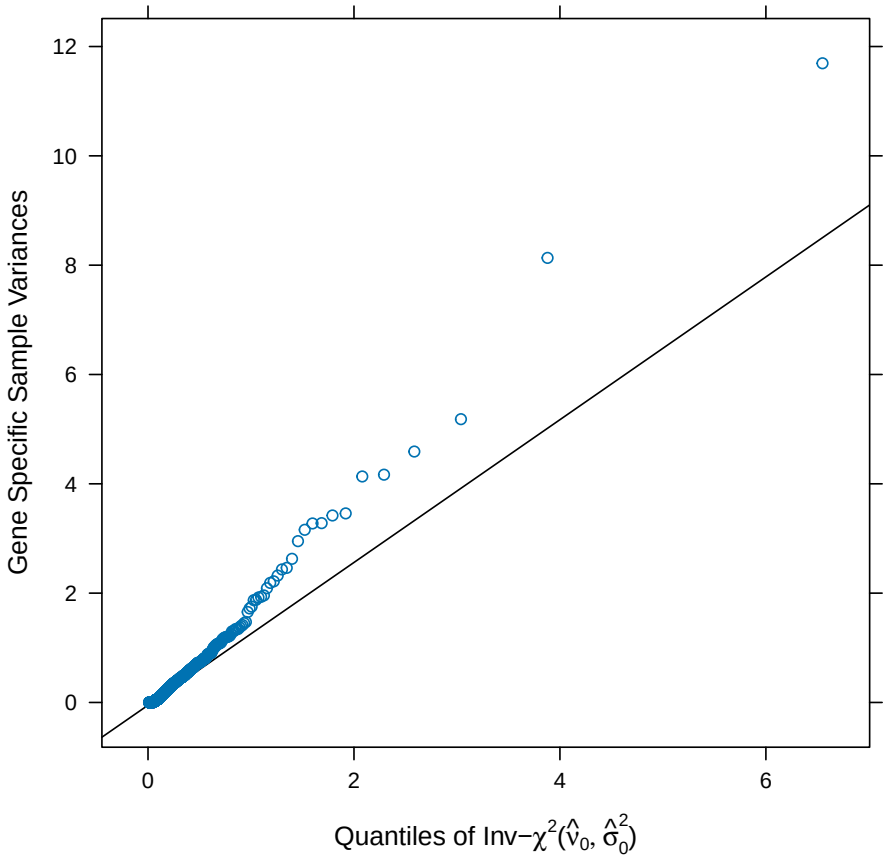

Figure 7: Scaled inverse chi-square qq plot.

```
> print(checkVarsMar(gould, groupid = c(1,2,2,0,0,0,0,0,0,0),
+                    nint=2000, xlim=c(-0.1, 0.5)))
```

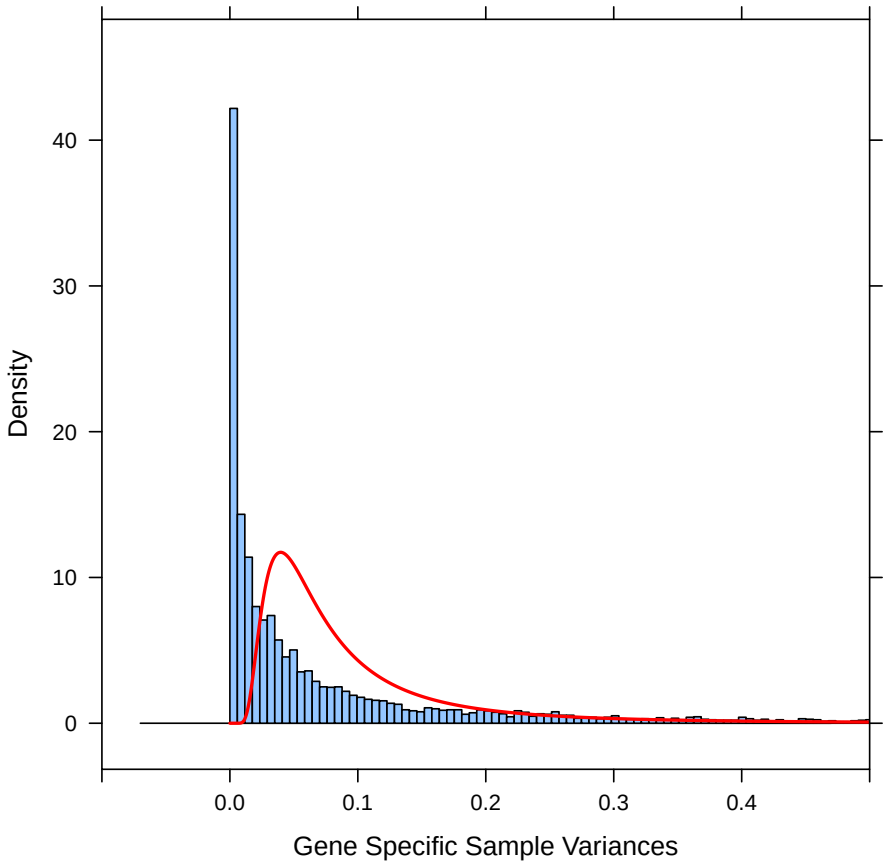

Figure 8: Density histogram of gene specific sample variances and the scaled inverse chi-square density plot (red line).

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

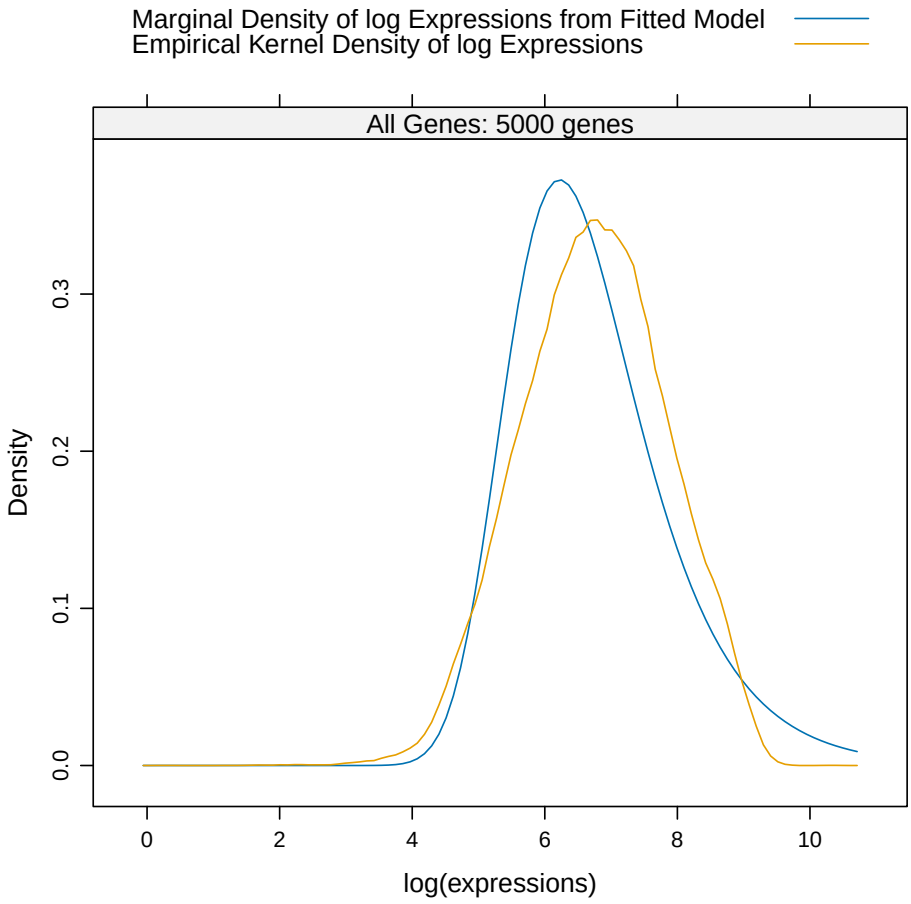

Figure 9: Marginal densities for Gamma-Gamma model

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

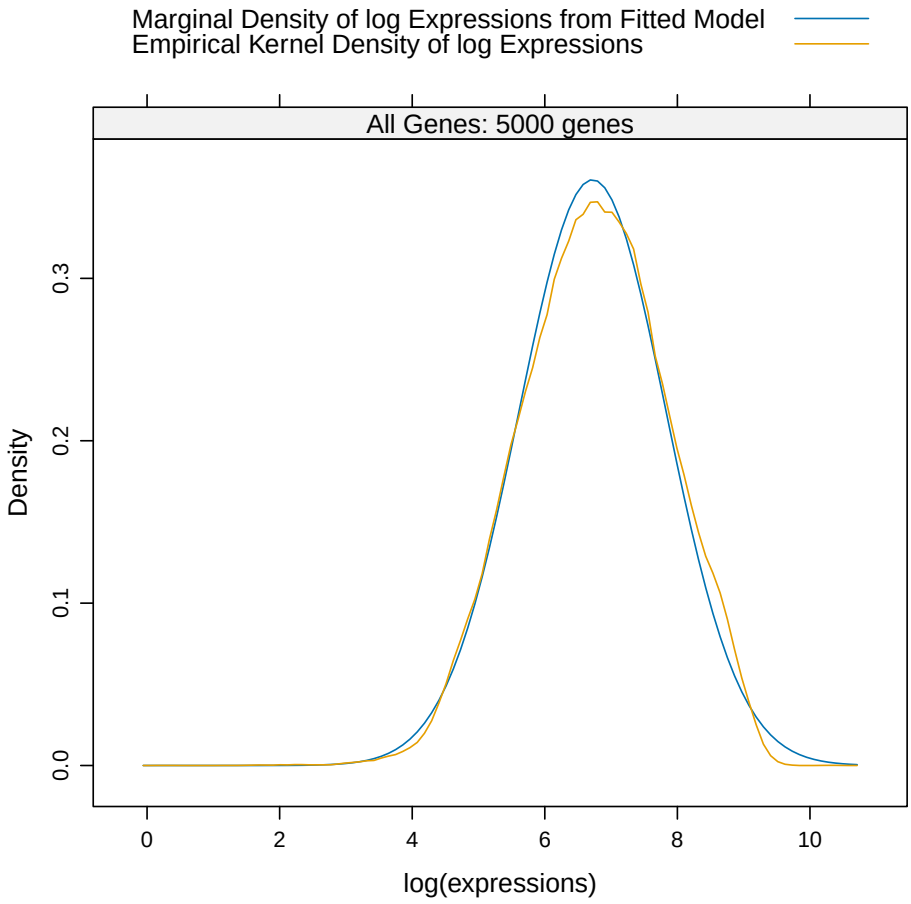

Figure 10: Marginal densities for Lognormal-Normal model

```
> print(checkVarsMar(gould, groupid = c(1,2,2,3,3,3,3,3,4,4),
+                            nint=2000, xlim=c(-0.1, 0.5)))
```

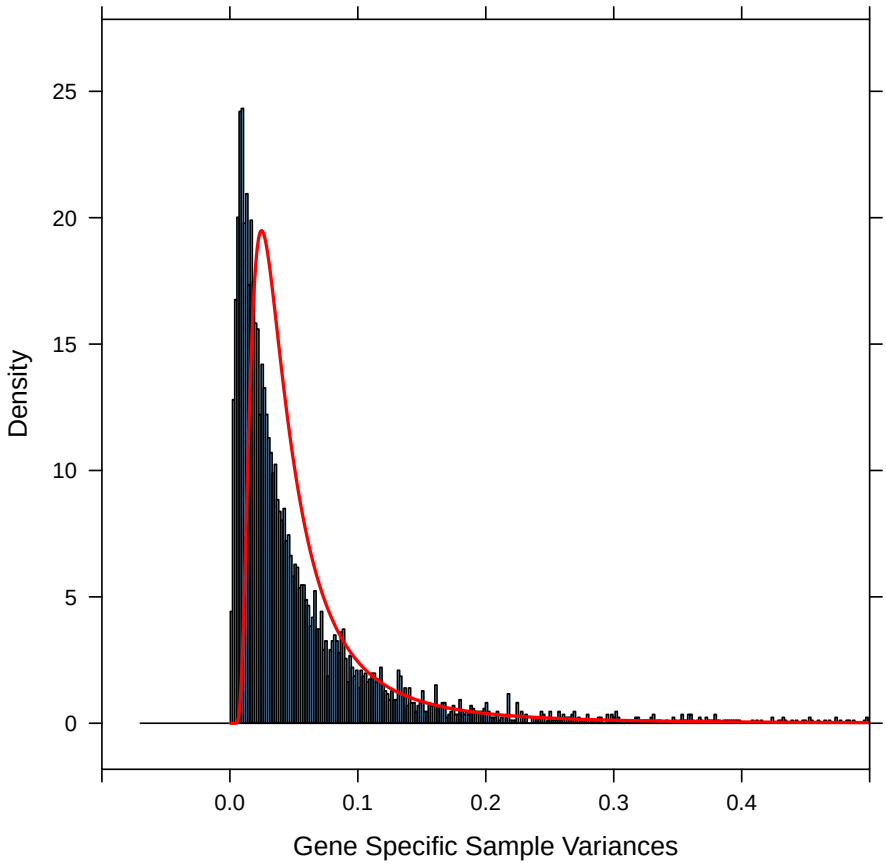

Figure 11: Density histogram of gene specific sample variances and the scaled inverse chi-square density plot (red line).

4. Newton, M.A., Noueiry, A., Sarkar, D., and Ahlquist, P. (2004). Detecting differential gene expression with a semiparametric hierarchical mixture model. *Biostatistics* **5**, 155-176.

5. Shao, J. (1999) *Mathematical Statistics*. Springer-Verlag, New York.

6. Yuan, M. and Kendziorski, C. (2006), A unified approach for simultaneous gene clustering and differential expression identification, *Biometrics*, 62(4), 1089-1098.