Available online at www.sciencedirect.com
ScienceDirect
Computational Statistics & Data Analysis 51 (2006) 570–586
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

Interval estimation in a finite mixture model: Modeling P-values in multiple testing applications

Qinfang Xiang[a], Jode Edwards[b], Gary L. Gadbury[a,*]

[a] *Department of Mathematics and Statistics, University of Missouri – Rolla, Rolla, MO 65409, USA*
[b] *USDA ARS, Department of Agronomy, Iowa State University, Ames, IA, USA*

Received 20 February 2005; received in revised form 6 September 2005; accepted 16 November 2005
Available online 9 December 2005

Abstract

The performance of interval estimates in a uniform-beta mixture model is evaluated using three computational strategies. Such a model has found use when modeling a distribution of P-values from multiple testing applications. The number of P-values and the closeness of a parameter to the boundary of its space both play a role in the precision of parameter estimates as does the "nearness" of the beta-distribution component to the uniform distribution. Three computational strategies are compared for computing interval estimates with each one having advantages and disadvantages for cases considered here.
Published by Elsevier B.V.

Keywords: Bootstrap; Gene expression; Hessian; Interval estimation; MCMC; Microarray; MLE; Uniform beta mixture

1. Introduction

A finite mixture model (McLachlan and Peel, 2000) of the form

$$f(x;\theta) = \lambda f_0(x;\theta_0) + (1-\lambda) f_1(x;\theta_1) \tag{1}$$

has recently been used in multiple testing applications where f_0 models a statistic for which a null hypothesis is true, and f_1 is the model for which an alternative is true (Lee et al., 2000; Efron and Tibshirani, 2002; Allison et al., 2002; Pounds and Morris, 2003; Gannoun et al., 2004). If the statistic is transformed into a P-value using the appropriate reference distribution (Sackrowitz and Samuel-Cahn, 1999), f_0 is the uniform density on (0, 1). The distribution of a P-value when the alternative hypothesis is true, f_1, depends on the effect size, sample size, and the distribution of the test statistic used to define the P-value (Hung et al., 1997). When many P-values are observed from multiple tests, those for which the alternative hypothesis is true should be smaller than expected from a uniform distribution, a fact used by Schweder and Spjøtvoll (1982) and more recently by Delongchamp et al. (2004). Parker and Rothenberg (1988) found that a mixture of a uniform distribution and non-uniform beta distributions was sufficiently flexible to model the distribution of P-values from multiple hypothesis tests.

This paper considers a model of the form,

$$f(p_i) = \lambda + (1-\lambda)\beta(p_i; r, s), \quad p_i \in (0,1),\ i = 1,\ldots,k, \tag{2}$$

* Corresponding author. Tel.: +1 573 341 4648; fax: +1 573 341 4741.
E-mail address: gadburyg@umr.edu (G.L. Gadbury).

0167-9473/$ - see front matter Published by Elsevier B.V.
doi:10.1016/j.csda.2005.11.011

where p_i is a P-value for the ith hypothesis test (out of k tests), $\lambda \in (0, 1)$ is a weight on the uniform component, and $\beta(p; r, s)$ is a beta-probability density function (pdf) with parameters r and s. Such a model has been used by others to model the distribution of P-values from microarray experiments (e.g., Allison et al., 2002; Pounds and Morris, 2003; Gadbury et al., 2004). A typical microarray experiment tests for differential genetic expression across two or more treatment groups. Multiple testing issues arise due to the many thousands of genes that are simultaneously tested for differential expression (see Knudsen, 2002 or Speed, 2003, for more details on microarrays). The parameter λ is thus interpreted as the proportion of genes that have no differential expression due to treatment. An advantage of model (2) is its generality in that any test producing a valid P-value can use this approach. The usual t-test is common but the bootstrap and nonparametric methods have also been proposed (e.g., Tusher et al., 2001; Neuhauser and Lam, 2004). Gadbury et al. (2003), however, noted that the distribution of P-values from randomization tests can be too discrete to model using (2) when sample sizes (number of microarrays) are small.

This paper builds on prior work (Allison et al., 2002) by assessing the precision of maximum likelihood estimates (MLEs) and Bayesian estimates of the parameter $\theta = (\lambda, r, s)$ and functions of θ under varying values of θ and k, particularly when λ is near one and/or when r and s have values reflecting a "near uniform" beta distribution. The effect of the number of tests, k, on the precision of parameter estimates is an important consideration since microarray experiments may involve 1000 tests to over 60,000 tests. Moreover, the present genomic era (Wolfsberg et al., 2002) opens a new realm of high dimensional biology (HDB) where multiple testing situations become the norm rather than the exception. Investigations into genetic polymorphisms, gene expression levels, protein measurements, genetic sequences, or any combination of these and their interactions may lead to experiments with tests numbering in the hundreds or in the hundreds of thousands.

We evaluate the precision of estimates using 95% confidence intervals and a 3^4 complete factorial simulation design. The function of θ considered is a "true positive probability" (TP) evaluated at a threshold τ,

$$TP = \frac{(1-\lambda)B(\tau; r, s)}{\lambda\tau + (1-\lambda)B(\tau; r, s)}, \tag{3}$$

where $B(\tau; r, s)$ is the cumulative distribution function (CDF) of a beta distribution with parameters r and s, evaluated at τ, and τ is a user-selected threshold at which a P-value is declared "significant." TP can be interpreted as the proportion of genes differentially expressed among those so declared at a threshold τ. In microarray experiments, values of TP provide guidance for follow-on research by quantifying how likely certain genes are indeed affected by a treatment. The quantity, $1 - TP$ is similar in concept to the false discovery rate (FDR) reported by others (e.g., Benjamini and Hochberg, 1995, 2000; Storey, 2002). Previous work has produced point estimates for θ (e.g., Parker and Rothenberg, 1988; Allison et al., 2002; Pounds and Morris, 2003; Gadbury et al., 2004) and hence, for TP, but little was done to evaluate the precision of these estimates.

We also evaluate the precision of estimates for a quantity called dTP which is like the expression in (3) except using density functions rather than CDF's. This quantity can be interpreted as a Bayesian posterior probability of expression. Subtracting dTP from one is analogous to the local false discovery rate discussed in other work (Efron, 2004). Since TP can be thought of as an "average" of posterior probabilities over all genes declared differentially expressed, dTP values will be lower than TP when evaluated at the same τ. Hereafter we focus on estimation results for TP, but results for dTP are included in supplementary material (the internet link is given in Section 3.1).

We compare three different methods for interval estimation: one using asymptotic normality of MLE's, another using potential symmetry of sampling distributions but using the bootstrap for standard errors, and the third using a Bayesian model with Markov–Chain Monte–Carlo (MCMC) to estimate a posterior probability distribution. All three methods had their advantages and limitations. Chung et al. (2004) considered a mixture of two exponential distributions and found that MCMC, when appropriately implemented, can offer advantages over the bootstrap when multimodal likelihood surfaces arise due to the label switching problem (Celeux et al., 2000). Another recent study by Dias and Wedel (2004) compared computational methods for estimation of parameters in a Gaussian mixture. Since our initial mixture component was completely specified, we did not need to address the label switching problem, and the implementation of MCMC was more straightforward as a result. Issues did arise, however, when the beta component was near uniform.

In summary, this paper provides insight into answers to two questions: 1. For what values of parameters and number of tests can one obtain precise parameter estimates? and 2. What are the relative advantages and limitations of three computational strategies for computing interval estimates. It is important to note that we only consider the ability to

obtain precise estimates from the model in (2) assuming P-values are generated randomly from the model. So the true values of the model parameters are known and fixed for each simulation case. This was important for computing and comparing the actual coverage of confidence intervals for model parameters. Variance of parameter estimates is with respect to sampling from the data generating model, that is, the uniform beta mixture model. Correlation among P-values was not considered. When P-values are positively correlated, for example, the sample distribution of P-values will vary more from sample to sample, particularly at P-values near zero or one. Schweder and Spjøtvoll (1982) provided some analytical details of this and it was shown using simulations by Allison et al. (2002) and Gadbury et al. (2003). More discussion of this is given in Section 6 as a topic of continuing research.

In the next section, we will introduce the simulation design, methods of estimation, and how those methods are implemented for interval estimation. Some key results will be summarized in Section 3. Somewhat surprisingly, we found that as "few" as 500 tests (P-values) can produce interval estimates for parameters that are unusually wide and difficult to compute. Issues related to MCMC convergence and maximum likelihood estimation using numerical methods will be addressed in this section as well. In Section 4, two data examples are used to further illustrate the results. Then we provide more discussion of the three methods for interval estimation and conclude with some summary recommendations.

2. Methods

2.1. Simulation design

A 3^4 complete factorial design was used to generate 81 simulation cases for varying values of $\theta = (\lambda, r, s)$ and k. The three selected values of each are, $k = (500, 1000, 10, 000)$, $\lambda = (0.5, 0.7, 0.9)$, $r = (0.4, 0.6, 0.8)$, and $s = (2, 4, 8)$. The levels were chosen to vary the magnitude of the log-likelihood given by

$$\ln L(\theta|p) = \sum_{i=1}^{k} \ln f\,(p_i)\,, \tag{4}$$

where $f\,(p_i)$ is given in Eq. (2). When $\theta = (0.9, 0.8, 2)$, the mixture distribution is dominated by the uniform component and the value of (4) is small relative to the case $\theta = (0.5, 0.4, 8)$, where the non-uniform beta distribution is more dominant. We refer to these cases as a "weak" versus a "strong" signal, respectively, and the selected parameter combinations allow for a variety of cases in between. These ranges for parameter values reflect our experience of fitting the uniform-beta mixture model to the many data sets that we have analyzed. Also affecting the strength of the signal is the number of P-values (a larger k yields a larger log-likelihood if holding values of θ fixed). We will see that it is difficult to estimate θ when the signal is weak and k is small (i.e., $k = 500$). The uncertainty in estimates of θ and derived quantities, like TP in (3), was evaluated using the performance of interval estimates. Three different methods of interval estimation were compared: a method using asymptotic normality of MLE's referred to as the Hessian method, a bootstrap technique that relies on the assumption of symmetry of the sampling distribution of MLE's, and a Bayesian approach to estimating a posterior probability distribution of the parameters using an MCMC method, the Metropolis random walk algorithm.

For each of the 81 simulation cases, a set of P-values was randomly generated from the model in (2). Given the parameter set $\theta = (\lambda, r, s)$ and the number of tests, k, the number, k_U, was simulated from a binomial distribution with parameters (λ, k) and k_U P-values then simulated from a uniform distribution and $1 - k_U$ P-values from a beta distribution with parameters (r, s). For each of 81 simulation cases, 1000 sets of P-values were generated. The performance of interval estimates was evaluated using the resulting 1000 confidence intervals for each simulation case. The percent coverage (out of 1000 intervals), average length, median length, and standard deviations of lengths were recorded. Of the intervals that did not cover the true parameter value, the number of those that missed on the lower bound and on the upper bound was also recorded.

2.2. Point estimation

In the Hessian and bootstrap methods, computation of MLE's, $\hat{\theta}$, were required to produce interval estimates. Values of θ that maximize (4) were computed using the R function *optim* with the method *L-BFGS-B*.

2.3. Interval estimation

The general procedure for interval estimation using the Hessian and bootstrap methods is,

(a) Generate a set of k P-values from the mixture model.
(b) Estimate the set of parameters $\theta = (\lambda, r, s)$ and TP (and dTP) at three different thresholds $\tau = 0.01, 0.001, 0.0001$ and compute their estimated variances.
(c) Compute interval estimates for the parameters in (b).
(d) Repeat steps (a)–(c) 1000 times.
(e) Compute the confidence interval performance summaries described above.

The Bayesian approach will be described in a later subsection.

2.3.1. Hessian

The Hessian of (4) is a three-dimensional square matrix, H, with entries,

$$H_{ij} = \frac{\partial^2 \ln L(\theta|p)}{\partial\theta_i \partial\theta_j} \quad \theta = (\theta_1, \theta_2, \theta_3) = (\lambda, r, s). \tag{5}$$

The information matrix is

$$I(\theta) = -E[H]. \tag{6}$$

Under regularity conditions (e.g., the derivatives in (5) exist), MLEs are asymptotically normal (see McLachlan and Peel, 2000, Chapter 2 for more discussion). This gives $\widehat{\theta} \overset{a}{\sim} N[\theta, V(\theta)]$, where $V(\theta) = [I(\theta)]^{-1}$ is the asymptotic variance. The asymptotic variance was estimated using the observed information (Efron and Hinkley, 1978), $i\left(\hat{\theta}\right) = -H|_{\theta=\hat{\theta}}$ so that the estimated asymptotic variance is $\widehat{V(\theta)} = i^{-1}\left(\hat{\theta}\right)$.

The MLE for TP in (3) is a function of the MLEs $\hat{\theta}$. To estimate its variance the vector, $J = \left(\frac{\partial TP}{\partial\lambda}, \frac{\partial TP}{\partial r}, \frac{\partial TP}{\partial s}\right)$ was derived where

$$\frac{\partial TP}{\partial\lambda} = -\frac{\tau B(\tau, r, s)}{[\lambda\tau + (1-\lambda)B(\tau, r, s)]^2},$$

$$\frac{\partial TP}{\partial r} = \frac{\lambda(1-\lambda)\tau(\partial B(\tau, r, s)/\partial r)}{[\lambda\tau + (1-\lambda)B(\tau, r, s)]^2},$$

$$\frac{\partial TP}{\partial s} = \frac{\lambda(1-\lambda)\tau(\partial B(\tau, r, s)/\partial s)}{[\lambda\tau + (1-\lambda)B(\tau, r, s)]^2}$$

and where $B(\tau; r, s)$ is the beta-cumulative distribution function with parameters r and s, evaluated at τ. The estimated variance is then $\hat{J}\widehat{V(\theta)}\hat{J}'$, where $\hat{J}$ is the vector J evaluated at the MLEs. A numerical approach to compute $\partial B(\tau, r, s)/\partial r$, $\partial B(\tau, r, s)/\partial s$ was implemented in R/S-plus by Boik and Robinson-Cox (1998). Use of asymptotic normality of MLE's allows computation of approximate large sample confidence intervals for model parameters, λ, r, s, and TP. Numerical difficulties that were encountered with computation of MLEs and the estimated variance matrix are discussed in Subsection 3.3.

2.3.2. Bootstrap

We used the bootstrap to estimate standard errors of MLEs but assumed approximate normality of their sampling distributions. This bootstrap method was chosen for two reasons, one based on intuition and the other computational

practicality. First, since the number of P-values is 500 or larger, we expected the sampling distribution to be somewhat symmetric (this unfortunately turned out to not quite be the case for weak signals). Second, the number of bootstrap samples can be considerably smaller when only computing standard errors versus what would be required for percentiles and, when the signal was weak, it was sometimes difficult to numerically obtain an MLE within a bootstrap simulation. More discussion of this limitation is given in Section 3.3. In Section 5, we do compute a bias corrected percentile interval for a selected case as illustration.

P-values were resampled with replacement and the model fit to this bootstrapped sample, obtaining $\hat{\theta}^*$ and $\widehat{TP}^*$. This was repeated B times and the standard deviation of the B separate estimates computed yielding the estimated standard error, $\widehat{se}$. A $100(1-\alpha)\%$ confidence interval is $\hat{\theta} \pm t_{k-1,\alpha/2}\widehat{se}$. We used $B = 100$ which is generally believed to be large enough for standard error estimation (Efron and Tibshirani, 1993).

2.3.3. Bayesian approach

A simple Bayesian model was used to compute the joint posterior probability distribution for the three parameters, $\theta = (\lambda, r, s)$. All three parameters were assumed to have uniform priors over the parameter space Ω where

$$\Omega = \{\theta = (\lambda, r, s) : 0 < \lambda < 1, 0 < r \leqslant M, 0 < s \leqslant M\}$$

and the joint prior distribution was the product of the marginals. The upper bound, M, for r and s was chosen to avoid numerical encounters with infinity when computing the value of the density (i.e., the beta-distribution pdf uses a ratio of gamma-functions and is computed in R to be infinity if the numerator is infinite, regardless of the denominator). This upper bound for the parameter space was also used for the bootstrap and Hessian techniques when computing MLEs using the numerical optimizer. We used $M = 170$, a choice that had minimal effect on computed results for any of the methods.

Under this prior specification, the posterior distribution for the parameters given the data was proportional to the joint distribution of the data given the parameters, i.e., the likelihood function, $f_{\Theta|\underline{p}}(\theta) \propto f_{\underline{P}|\theta}\left(\underline{p}\right) I_{\Omega}(\theta)$, where $f_{\underline{P}|\theta}\left(\underline{p}\right) = \prod_i f\,(p_i)$, $f\,(p_i)$ is the model given in (2), and $I_{\Omega}(\theta)$ is the indicator function over the parameter space.

An MCMC random-walk Metropolis algorithm (Chib and Greenberg, 1995; Roberts, 1996) was used to simulate the posterior distribution. First an initial value, $\theta_0 = (\lambda_0, r_0, s_0)$ is chosen. Given that the chain is in the jth state, $\theta_j = \left(\lambda_j, r_j, s_j\right)$, $j = 0, 1, \ldots$, a proposed value, $\theta^* = \left(\lambda^*, r^*, s^*\right)$ was drawn according to the process:

$$\lambda^* = \lambda + t_1 u_1,$$

$$r^* = r + t_2 u_2,$$

$$s^* = s + t_3 u_3,$$

where (u_1, u_2, u_3) are random numbers generated independently from symmetric uniform distributions on $(-1, 1)$, and (t_1, t_2, t_3) are tuning variables. The proposal θ^* was accepted with probability

$$\gamma = \min\left\{1, \frac{l\left(\theta^* | \underline{p}\right)}{l\left(\theta_j | \underline{p}\right)}\right\},$$

where $l(\bullet)$ is the logarithm of the likelihood. This simple acceptance/rejection rule resulted from the assumption of flat priors and a symmetric proposal distribution. If the proposal was accepted, $\theta_{j+1} = \theta^*$, otherwise $\theta_{j+1} = \theta_j$. This was repeated for N iterations and a burn-in period, $j = 0, \ldots, j_b$ was discarded. The resulting $\theta_{b+1}, \ldots, \theta_N$ comprised the estimated posterior distribution. See Gelman et al. (2004, Chapter 11) for a discussion of convergence for the random walk algorithm.

The $(1-\alpha)100\%$ Bayesian support interval for the parameters was computed using the $\alpha/2$ and $1-\alpha/2$ quantiles of this posterior. The starting values we used were $\theta_0 = (0.72, 1.11, 4.79)$ and were the same for all simulation cases. These values were arbitrarily selected from the parameter space to facilitate convergence of the chain, and their values would not affect the estimated posterior providing the MCMC chain was run to convergence and the appropriate burn-in period discarded. The tuning parameters, total number N of MCMC simulations, and the burn-in period varied depending on the case. These details are given in Subsection 3.3.

Table 1
Nine selected cases showing percent Coverage (average length), and number of misses at the lower (low), upper (up) bounds of 95% interval estimates from 1000 simulations for three methods

	True parameters				Percent coverage (average length)			# of miss (low + up)		
	k	λ	r	s	λ	r	s	λ	r	s
Hessian	10,000	0.5	0.4	8	0.953(0.03)	0.951(0.03)	0.956(1.69)	21 + 26	23 + 26	16 + 28
	10,000	0.7	0.6	4	0.954(0.05)	0.950(0.07)	0.948(1.48)	27 + 19	18 + 32	17 + 35
	10,000	0.9	0.8	2	0.901(0.48)	0.979(0.34)	0.880(3.34)	99 + 0	16 + 5	4 + 116
	1000	0.5	0.4	8	0.957(0.09)	0.934(0.09)	0.934(5.43)	23 + 20	29 + 37	24 + 42
	1000	0.7	0.6	4	0.954(0.18)	0.940(0.21)	0.933(4.92)	40 + 6	27 + 33	15 + 52
	1000	0.9	0.8	2	0.839(0.86)	0.974(6.87)	0.916(27.91)	160 + 1	10 + 16	2 + 82
	500	0.5	0.4	8	0.942(0.13)	0.949(0.12)	0.934(7.76)	30 + 28	24 + 27	12 + 54
	500	0.7	0.6	4	0.949(0.30)	0.958(0.31)	0.902(7.16)	43 + 8	16 + 26	7 + 91
	500	0.9	0.8	2	0.845(0.73)	0.970(18.29)	0.971(46.90)	154 + 1	10 + 20	0 + 29
Bootstrap	10,000	0.5	0.4	8	0.944(0.03)	0.950(0.03)	0.944(1.69)	28 + 28	27 + 23	28 + 28
	10,000	0.7	0.6	4	0.958(0.05)	0.951(0.07)	0.959(1.51)	27 + 15	22 + 27	12 + 29
	10,000	0.9	0.8	2	0.949(0.37)	0.975(0.74)	0.906(9.45)	34 + 17	18 + 7	8 + 86
	1000	0.5	0.4	8	0.959(0.10)	0.950(0.09)	0.957(5.61)	20 + 21	25 + 25	11 + 32
	1000	0.7	0.6	4	0.964(0.31)	0.965(0.22)	0.940(5.98)	20 + 16	21 + 14	9 + 51
	1000	0.9	0.8	2	0.948(0.42)	0.992(31.58)	0.989(53.55)	13 + 39	5 + 3	9 + 2
	500	0.5	0.4	8	0.960(0.15)	0.949(0.13)	0.946(8.47)	16 + 24	20 + 31	12 + 42
	500	0.7	0.6	4	0.948(0.43)	0.968(0.67)	0.929(12.08)	28 + 24	15 + 17	2 + 69
	500	0.9	0.8	2	0.954(0.46)	0.991(62.75)	0.970(75.12)	10 + 36	7 + 2	30 + 0
MCMC	10,000	0.5	0.4	8	0.958(0.03)	0.949(0.03)	0.947(1.69)	22 + 20	29 + 22	27 + 26
	10,000	0.7	0.6	4	0.949(0.05)	0.945(0.06)	0.948(1.49)	31 + 20	35 + 20	31 + 21
	10,000	0.9	0.8	2	0.927(0.26)	0.901(0.34)	0.921(3.91)	48 + 25	99 + 0	62 + 17
	1000	0.5	0.4	8	0.944(0.09)	0.932(0.09)	0.928(5.36)	30 + 26	50 + 18	51 + 21
	1000	0.7	0.6	4	0.925(0.26)	0.933(0.21)	0.900(5.27)	45 + 30	50 + 17	71 + 29
	1000	0.9	0.8	2	0.689(0.26)	0.598(2.10)	0.541(10.56)	303 + 8	397 + 5	458 + 1
	500	0.5	0.4	8	0.923(0.14)	0.937(0.13)	0.912(7.94)	44 + 33	46 + 17	69 + 19
	500	0.7	0.6	4	0.917(0.35)	0.922(0.32)	0.889(8.21)	59 + 24	71 + 7	92 + 19
	500	0.9	0.8	2	0.788(0.24)	0.511(1.72)	0.523(15.98)	209 + 3	485 + 4	477 + 0

3. Results

All 81 simulation cases were run on an IBM cluster based on UNIX. The cluster had a 128-processor (64 nodes, dual CPUs, Xeon 2.4 GHz, 2–4 GB memory for each node) with a terabyte storage unit. All the methods were implemented using the R software.

3.1. Presentation of simulation results

For brevity, performance details are provided in Table 1 for nine cases from the 81 simulations. Three cases were selected for $k = 10,000$, 1000, and 500, respectively and these three can be categorized as a strong $\theta = (0.5, 0.4, 8)$, intermediate $\theta = (0.7, 0.6, 4)$, or weak $\theta = (0.9, 0.8, 2)$ class based on their signal. The full result tables can be accessed at www.umr.edu/~geneexp/mcmc-paper/supplement.htm. Table 1 gives performance results for 95% interval estimates, including percentage of intervals that covered the true parameter values, the average length of the intervals, and the number of times that the interval missed covering the parameter value at the lower or at the upper bounds, out of 1000 intervals. The full results at the above web site also provide the median length of intervals and the standard deviations of the lengths of the 1000 intervals. The performance of interval estimates for *TP* are summarized in Table 2, where *TP*0001, *TP*001, and *TP*01 are the values of *TP* evaluated at a threshold $\tau = 0.0001, 0.001, 0.01$, respectively. The simulation parameter values, i.e., the nine cases in Table 2, are the same as those in Table 1. The results for d*TP* computed at the same thresholds are given in tables at the above web site.

Table 2
As for Table 1, the same nine cases showing percent coverage (average length), number of misses at the lower, upper bounds of 95% confidence interval estimates for true positive probability *TP* over 1000 simulations for the three methods. *TP*0001, *TP*001, and *TP*01 represent *TP* evaluated at thresholds $\tau = 0.0001, 0.001, 0.01$, respectively

	True parameters			Percent coverage (average length)			# of miss (low + up)		
	*TP*0001	*TP*001	*TP*01	*TP*0001	*TP*001	*TP*01	*TP*0001	*TP*001	*TP*01
Hessian	0.998	0.994	0.975	0.948(0.0003)	0.935(0.0009)	0.949(0.003)	20 + 32	25 + 40	21 + 30
	0.977	0.944	0.870	0.946(0.01)	0.948(0.02)	0.947(0.02)	19 + 35	23 + 29	27 + 26
	0.558	0.443	0.333	0.993(1.29)	0.993(1.20)	0.996(1.03)	7 + 0	7 + 0	4 + 0
	0.998	0.994	0.975	0.935(0.001)	0.943(0.003)	0.955(0.008)	17 + 48	16 + 41	12 + 33
	0.977	0.944	0.870	0.931(0.03)	0.952(0.06)	0.972(0.08)	1 + 68	2 + 46	4 + 24
	0.558	0.443	0.333	0.995(14.59)	0.989(11.20)	0.991(7.26)	5 + 0	11 + 0	9 + 0
	0.998	0.994	0.975	0.941(0.001)	0.950(0.004)	0.953(0.01)	8 + 51	5 + 45	10 + 37
	0.977	0.944	0.870	0.897(0.05)	0.927(0.09)	0.967(0.15)	0 + 103	2 + 71	2 + 31
	0.558	0.443	0.333	0.999(37.94)	0.993(28.18)	0.989(16.93)	1 + 0	7 + 0	11 + 0
Bootstrap	0.998	0.994	0.975	0.952(0.0003)	0.958(0.0009)	0.952(0.003)	25 + 23	27 + 15	30 + 18
	0.977	0.944	0.870	0.954(0.01)	0.948(0.02)	0.944(0.02)	40 + 6	40 + 12	36 + 20
	0.558	0.443	0.333	0.924(0.56)	0.942(0.45)	0.970(0.36)	38 + 38	34 + 24	21 + 9
	0.998	0.994	0.975	0.953(0.001)	0.950(0.003)	0.951(0.008)	40 + 7	38 + 12	34 + 15
	0.977	0.944	0.870	0.953(0.05)	0.954(0.07)	0.958(0.09)	47 + 0	46 + 0	35 + 7
	0.558	0.443	0.333	0.819(1.02)	0.858(0.84)	0.903(0.62)	21 + 160	41 + 101	56 + 41
	0.998	0.994	0.975	0.958(0.002)	0.955(0.004)	0.954(0.01)	41 + 1	41 + 4	38 + 8
	0.977	0.944	0.870	0.945(0.14)	0.948(0.14)	0.958(0.14)	55 + 0	52 + 0	41 + 1
	0.558	0.443	0.333	0.786(1.09)	0.835(0.93)	0.895(0.71)	28 + 186	47 + 118	63 + 42
MCMC	0.998	0.994	0.975	0.955(0.0003)	0.953(0.0009)	0.956(0.003)	19 + 26	23 + 24	20 + 24
	0.977	0.944	0.870	0.946(0.01)	0.952(0.02)	0.954(0.02)	17 + 34	14 + 34	17 + 29
	0.558	0.443	0.333	0.918(0.49)	0.921(0.39)	0.922(0.30)	22 + 60	27 + 52	28 + 50
	0.998	0.994	0.975	0.933(0.001)	0.933(0.003)	0.939(0.008)	19 + 48	24 + 43	28 + 33
	0.977	0.944	0.870	0.932(0.05)	0.933(0.06)	0.939(0.09)	28 + 40	32 + 35	34 + 270
	0.558	0.443	0.333	0.643(0.57)	0.662(0.50)	0.693(0.41)	9 + 348	11 + 327	10 + 297
	0.998	0.994	0.975	0.941(0.001)	0.945(0.004)	0.936(0.01)	17 + 42	18 + 37	31 + 33
	0.977	0.944	0.870	0.921(0.10)	0.927(0.11)	0.936(0.13)	20 + 59	24 + 49	28 + 36
	0.558	0.443	0.333	0.565(0.53)	0.590(0.47)	0.643(0.40)	4 + 431	3 + 407	7 + 350

The cases are in the same order as those for Table 1. The first three rows for each section (i.e., Hessian, Bootstrap, and MCMC) are $k = 10,000$ for strong, medium, and weak signals. Similarly, the next three rows are for $k = 1000$, and the last three for $k = 500$.

Figs. 1–4 graphically show performance summaries for 54 cases. Figs. 1 and 2 show percentage coverage and mean length of confidence intervals for λ when $k = 10,000$ (27 cases, Fig. 1), and when $k = 1000$ (27 cases, Fig. 2). Figs. 3 and 4 do the same for *TP*001, i.e., the value of *TP* evaluated at $\tau = 0.001$. The true value of *TP* at this threshold ranged from 0.443 to 0.994. All four figures show how the percent coverage and mean length change as a function of the true parameter values of λ and s. The plots are staggered off of the true parameter values for easier visualization of the performance of the three computational techniques. We focus less on the cases when $k = 500$ due to some computational challenges that make the performance summaries uncertain and more difficult to compare among the three techniques. These challenges and limitations are discussed in Subsection 3.3.

3.2. Discussion and comparison of simulation results

The performance of interval estimates is quite good for the three methods when the signal is in the strong or intermediate class and when k is large. For example, when $\theta = (0.5, 0.4, 8)$ and $k = 10,000$, the percent coverage for all parameters is close to 95% and the average length is small. Moreover, all three methods of estimation give similar coverage and average lengths for these particular cases. The difference between the number of interval misses at the lower and at the upper bounds is small meaning that the simulated sampling distributions of MLEs have no extreme

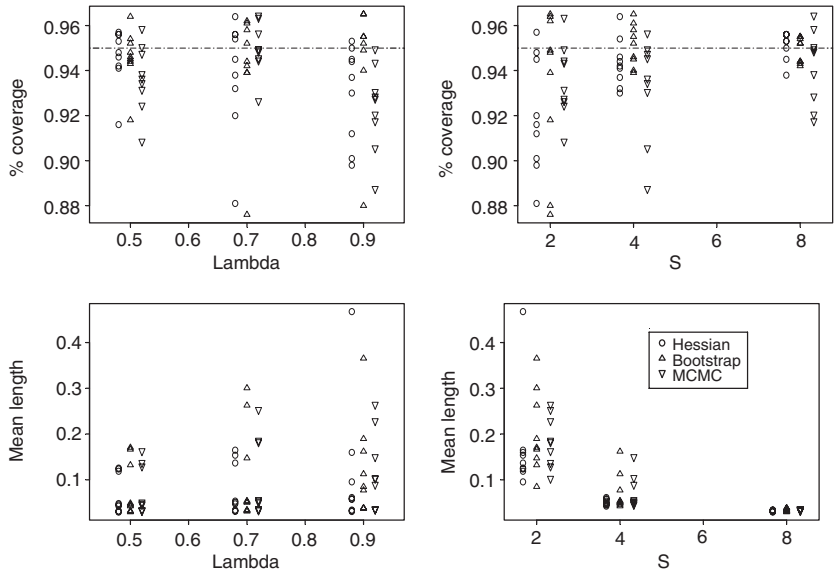

Fig. 1. Interval estimation performance—percentage coverage and average length for 95% interval estimate for λ among 1000 simulations when $k = 10,000$. The coverage and average length of the intervals for λ are plotted against the true value of λ and against the true value of s. The plot is staggered about the true values to improve visibility of the three computational methods. The Hessian method is shown with circles, the bootstrap by triangles, and the MCMC method using inverted triangles. The dashed line on the top row represents nominal coverage of 95%.

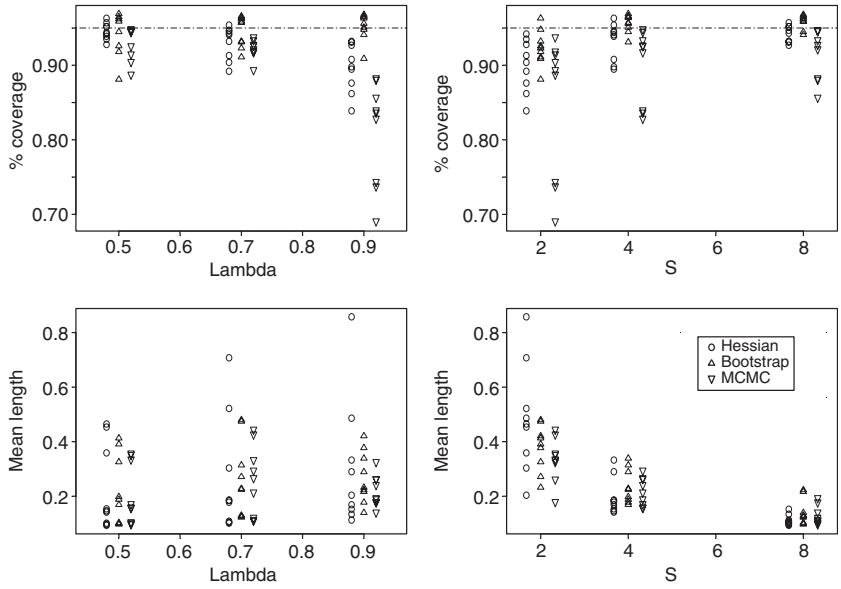

Fig. 2. Same as for Fig. 1 except $k = 1000$.

skewness in either left or right tail. This effect can also be seen in the supplementary tables provided on the above mentioned website where the average lengths of intervals is very similar to the median length.

If we reduce the signal by increasing the uniform component, λ, increasing r, reducing s, while keeping k fixed at 10,000, the coverage becomes more variable and the average length increases. Even when $\lambda = 0.9$, the interval percent coverage for λ is still above 90% for most cases. Estimates for s seem to be least precise relative to λ and r, Fig. 1 shows that coverages for λ are closest to the nominal 95%, and the mean length small when either the true λ is small

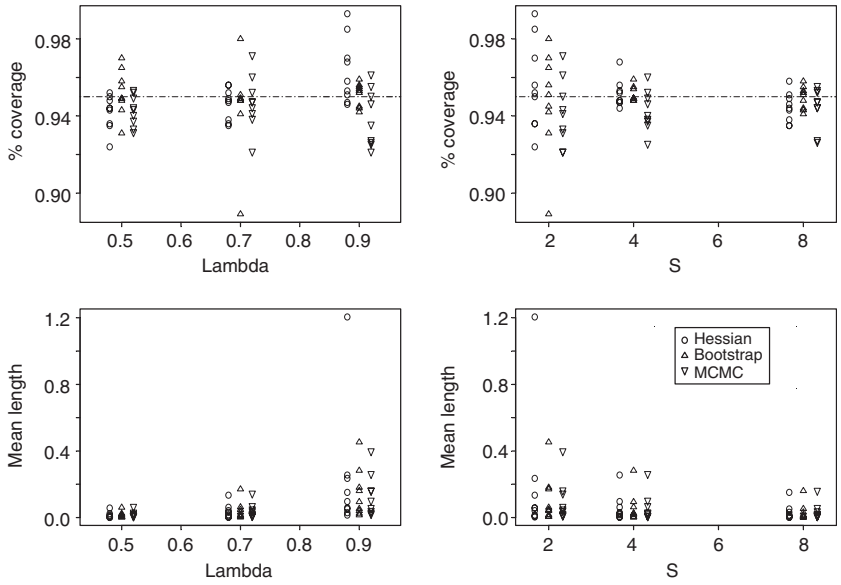

Fig. 3. Same as for Fig. 1 except the performance of intervals for *TP* evaluated at the threshold $\tau = 0.001$ are plotted against the true values of λ and of s, and $k = 10,000$, as for Fig. 1.

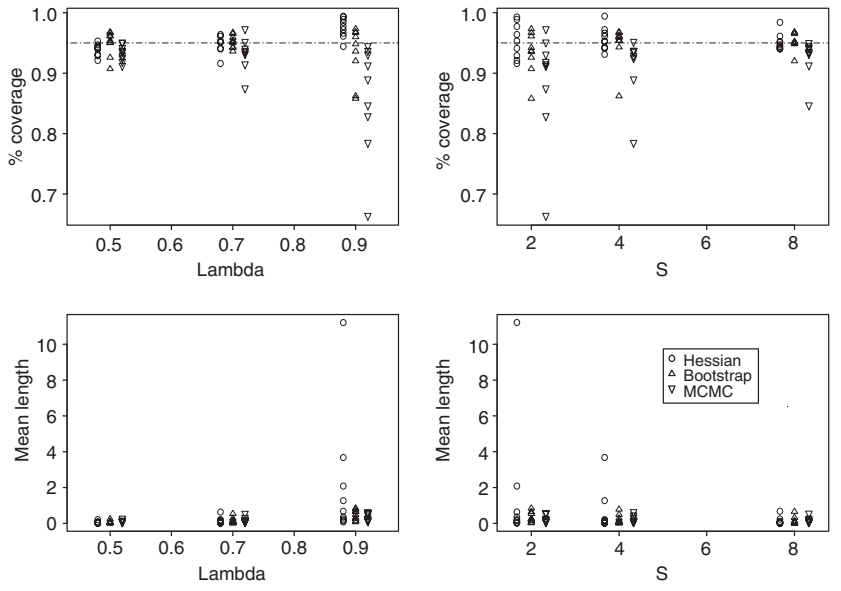

Fig. 4. Same as for Fig. 3 except $k = 1000$.

or s is large. When $\lambda = 0.9$ and $s = 2$, coverage and mean length vary much more over different values of r. A similar but more pronounced effect is seen in Fig. 2 for $k = 1000$ (note the different scales for the vertical axes in Figs. 1 and 2). The two figures also show that the Hessian and bootstrap methods tend to be slightly more conservative in coverage than MCMC for many cases.

The average length of intervals tends to vary more from case to case for the Hessian and bootstrap methods versus MCMC; however, the latter method produces range preserving interval estimates and the former two methods do not necessarily do so. Bootstrap percentile intervals would be range preserving but would have required more

bootstrap simulations. For comparison, we computed this type of interval for a selected case and report the result in Section 5.

When $k = 1000$, or 500 and when the signal is weak, the bootstrap maintains good coverage for λ though the lengths of intervals for r and s can become wide. This was also seen in results for the Hessian method. The MCMC tends to be more liberal in coverage for these cases but, as discussed in the next subsection, these interval estimates may have been computed from some MCMC chains that had not converged. Also noteworthy is that, for these cases, failures of coverage tend to happen on one side (i.e., the lower bound).

Table 2 shows results for the same cases as Table 1 except for TP evaluated at three thresholds. The true value of TP for all cases ranged from a low of 0.333 for $(k, \lambda, r, s, \tau) = (500, 0.9, 0.8, 2, 0.01)$ to a high of 0.998 for $(k, \lambda, r, s, \tau) = (10,000, 0.5, 0.4, 8, 0.0001)$. Results in Table 2 follow a similar pattern as those in Table 1. However, when the signal was weak and when $k = 500$ or 1000, the Hessian method sometimes produced very wide intervals. This can also be seen in Figs. 3 and 4 for TP evaluated at a threshold $\tau = 0.001$. For these cases, the flatness of the log-likelihood surface makes estimation challenging. Also, for these weak signals, the number of interval misses for $TP001$ usually fall to one side, suggesting the MLEs' distributions are not symmetric. Use of asymptotic normality may not be appropriate for these weak signals because k is not sufficiently large, and the Hessian method not suitable for interval estimation. McLachlan and Peel (2000, p. 42) caution that sample sizes need to be very large for asymptotic theory to apply to mixture models.

3.3. More simulation details and some limitations of simulation results

Numerical difficulties were encounted when computing MLEs for some cases, particularly when $k = 500$ and the signal was weak. The likelihood surface becomes flat near the optimum making it difficult to ensure that a unique global optimum was attained. The surface seemed most sensitive to the value of s, and formed a ridge along the s parameter axis when the true value of $s = 2$ and $(\lambda, r) = (0.9, 0.8)$. Fig. 5 compares a 3 dimensional plot of a likelihood surface (obtained from MCMC) for this case versus one for a stronger signal and $k = 10,000$. In cases when the likelihood surface was flat near the optimum, the observed information matrix was near singularity and/or the computed MLE for λ was on the boundary of the parameter space. When the numerical routine reached this boundary, the logarithm of the likelihood became zero and estimates for r and s were meaningless. In these cases up to 25 different combinations of starting values were used in an attempt to obtain a unique MLE inside the parameter space. If this failed, a new set of P-values was generated. The number of times that a new set of P-values was needed ranged from 0 (for the stronger signals or when $k = 10,000$) to 282 (out of 1000 simulated sets of P-values) for the case when $k = 500$ and $\theta = (0.9, 0.8, 2)$. Thus, for cases when this number was large, performance summaries of interval estimates for the Hessian and bootstrap methods may not be directly comparable with MCMC since the MCMC technique did not require that a new set of P-values be generated (i.e., MLEs were not required). This limitation was a numerical limitation rather than one associated with either the Hessian or bootstrap techniques.

Sometimes an MLE could be computed for the original sample of P-values but the same issue described above would arise within a bootstrap resample. The same procedure was used except that if the different combination of starting values did not resolve the problem, a new bootstrap sample was obtained from the original sample of P-values.

The MCMC method also posed challenges for small k and a weak signal. Poor or slow convergence may lead to imprecise inference when using MCMC to summarize the posterior distribution of parameters; however, techniques to monitor convergence are available. Monitoring convergence for all cases within the above-described simulations was not feasible, but we investigated convergence for several cases.

Mengersen et al. (1999) classified the MCMC diagnostics into three categories: exploration, stationary, and estimation. Some available software packages like CODA (Cowles et al., 1997) and BOA (Smith, 2003) implement some of these methods and provide a range of diagnostic tools. We used BOA to diagnose if the simulation output from MCMC had converged. Four popular diagnostic methods, Gelman and Rubin (1992), Geweke (1992), Heidelberger and Welch (1983), and Raftery and Lewis (1992), are coded in BOA.

When the signal is weak and the number of P-values small, i.e., $k = 500$ or 1000, convergence is more dependent upon choice of tuning parameters. It is possible to tune the choice of proposal distributions to optimize the performance of the MCMC simulation. The precision of summary percentiles of the posterior usually increase with the number of simulations but may be reduced by positive correlation such as occurs in a Markov chain (Link et al., 2001). For chains

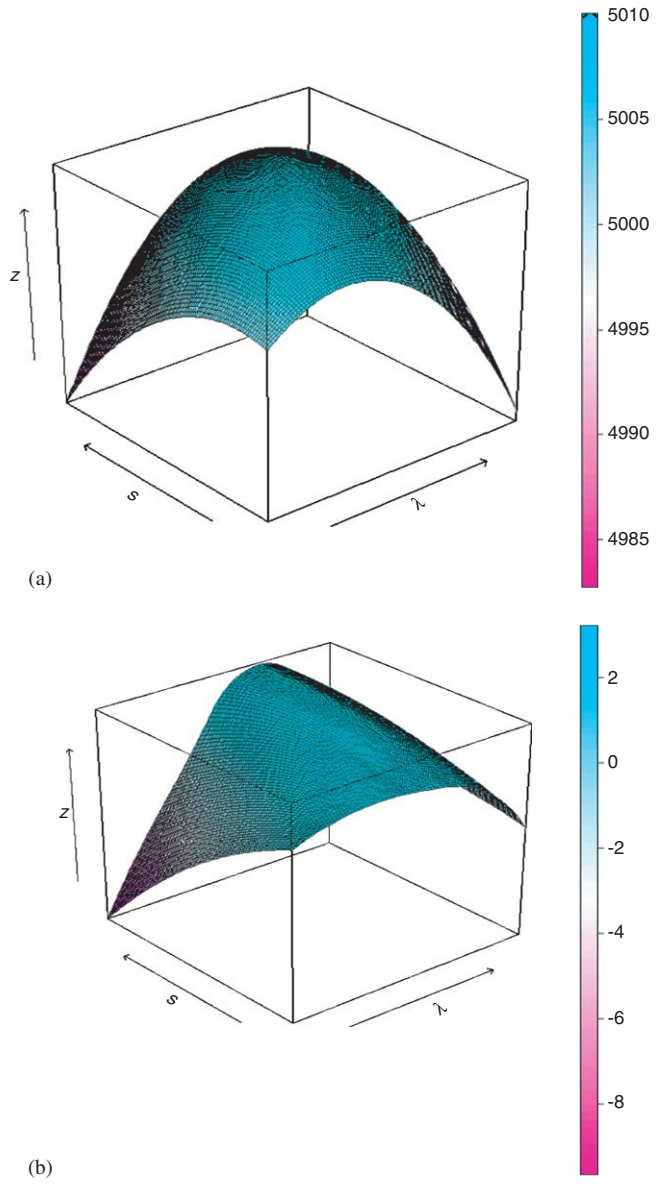

Fig. 5. Three-dimensional plots obtained from MCMC output for two cases: (a) the case $\theta = (0.5, 0.6, 8)$ and $k = 10,000$, (b) the case $\theta = (0.9, 0.8, 2)$ and $k = 500$. The vertical axis, z, is the log-likelihood plotted against λ and s.

generated using the Metropolis–Hasting algorithm, the magnitude of the correlation depends on the relation between current values θ_j and the candidate point θ^*. It also depends on the choice of tuning variables.

We used the diagnostic software to guide us in selecting tuning parameters, the number of MCMC iterations, and the burn-in period. For all cases when $\lambda = 0.5$, the tuning parameters were $(t_1, t_2, t_3) = (0.01, 0.01, 0.2)$ and for all other cases they were set to $(t_1, t_2, t_3) = (0.02, 0.02, 0.2)$. The total number of MCMC iterations(burn-in) was 20,000(5000) for $k = 10,000$, 30,000(15,000) for $k = 1000$, and 50,000(25,000) for $k = 500$. The chain moved more slowly from its starting value for smaller values of k, requiring a longer burn-in period. We are reasonably confident that the MCMC chain had converged for all cases when λ was either 0.5 or 0.7, regardless of k. We are less certain that the chain had converged for some cases when $\lambda = 0.9$, and believe that convergence had likely not occurred for when $\lambda = 0.9$ and $s = 2$. We doubt that convergence occurred for the 9 cases when $\lambda = 0.9$ and $k = 500$. Note that a "case" involves 1000

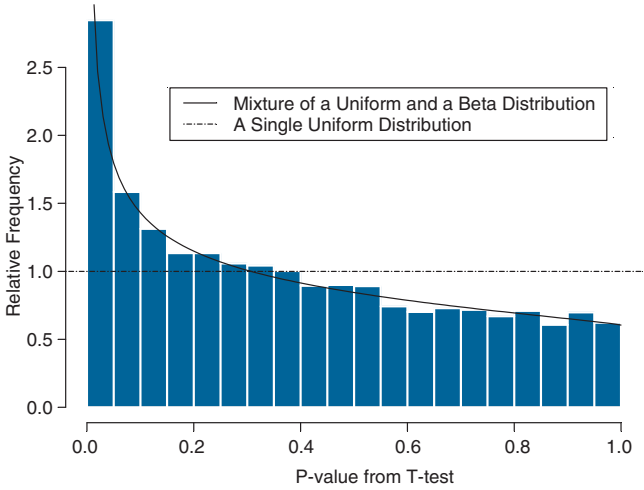

Fig. 6. Fitted mixture model (solid line curve) to 12,625 p-values obtained from the example data set 1. The dashed line is a uniform density function.

different simulations yielding 1000 interval estimates. It is possible that some of these simulations converged for some of these cases, but we doubt that all 1000 converged.

In practice when analyzing a data set, more care can be taken with starting values (for the Hessian or bootstrap method) and with use of convergence diagnostics with MCMC. For example, in other MCMC simulations (not reported here) we restricted the parameter r to be in the unit interval as we have observed it to be in many data sets that we have analyzed. This restriction improved the convergence of MCMC for weak signals when $k = 500$. Use of prior information on model parameters may be practical for some applications and can improve MCMC estimation performance.

4. Application to biological data

As illustration on actual data, confidence intervals for mixture model parameters were computed using data from two different microarray experiments. The first experiment (data set 1) was a study of a human rheumatoid arthritis synovial fibroblast (RASF) cell line. The cell line was randomly divided into two groups of three units with each group receiving different "treatments". Thus the experiment produced six microarrays with three arrays in each of two different treatment groups. The objective of the study was to determine genes that were differentially expressed across the two treatment groups. These differentially expressed genes were hypothesized to be related to the development of RASF cells and thus, potential targets for drugs that would diminish inflammatory activity. These data were analyzed in Gadbury et al. (2003, 2004) with biological details discussed in Zhang et al. (2004). Fig. 6 is a histogram of P-values obtained from two sample t-tests on each of $k = 12,625$ genes. The solid line is the mixture of one uniform plus one beta distribution. The dashed line is a uniform distribution.

The second experiment (data set 2) sought genes that are differentially expressed between CD4 cells taken from five young and five old male rhesus monkeys. Statistical significance for $k = 12,548$ genes was assessed using pooled variance t-tests after quantile–quantile normalization. These data were analyzed in Gadbury et al. (2004).

Table 3 shows the parameter estimates and 95% interval estimates for these two data sets. Interval estimates were obtained from the methods as described for the simulations. The point estimates in Table 3 are MLEs for the bootstrap and Hessian methods, and they are means of posterior distributions for the Bayesian MCMC method. The value of the maximum log-likelihood function for the model fitted to data set 1 is 1484. Interval estimates for all parameters are similar among the three methods. The maximum log-likelihood is 159 for data set 2, indicating that the signal is weaker relative to data set 1. However, $k = 12,548$ is large and interval estimates are still similar among the three methods, though the bootstrap method tends to produce intervals that are slightly more narrow than the other two methods. Fig. 7 shows the simulated marginal posterior distributions from MCMC (column 1) and the

Table 3
Parameter $\theta = (\lambda, r, s)$ estimates and their 95% interval estimates for two example data sets using the Hessian, bootstrap, and MCMC methods

Method	Point estimate (interval estimate)			Total elapsed times (s)
	λ	r	s	
Data set 1				
MCMC	0.598 (0.523, 0.652)	0.540 (0.518, 0.562)	1.830 (1.446, 2.238)	1179
Bootstrap	0.605 (0.555, 0.653)	0.539 (0.520, 0.558)	1.844 (1.546, 2.142)	616
Hessian	0.605 (0.547, 0.662)	0.539 (0.517, 0.561)	1.844 (1.451, 2.236)	8
Data set 2				
MCMC	0.801 (0.757, 0.837)	0.957 (0.874, 1.084)	2.584 (2.011, 3.342)	1172
Bootstrap	0.799 (0.763, 0.834)	0.943 (0.819, 1.067)	2.496 (1.896, 3.095)	906
Hessian	0.799 (0.757, 0.841)	0.943 (0.844, 1.042)	2.496 (1.860, 3.132)	11

The estimates from Hessian and bootstrap methods are MLEs, but the estimates from MCMC are means of posterior distributions after discarding the first 5000 iterations.

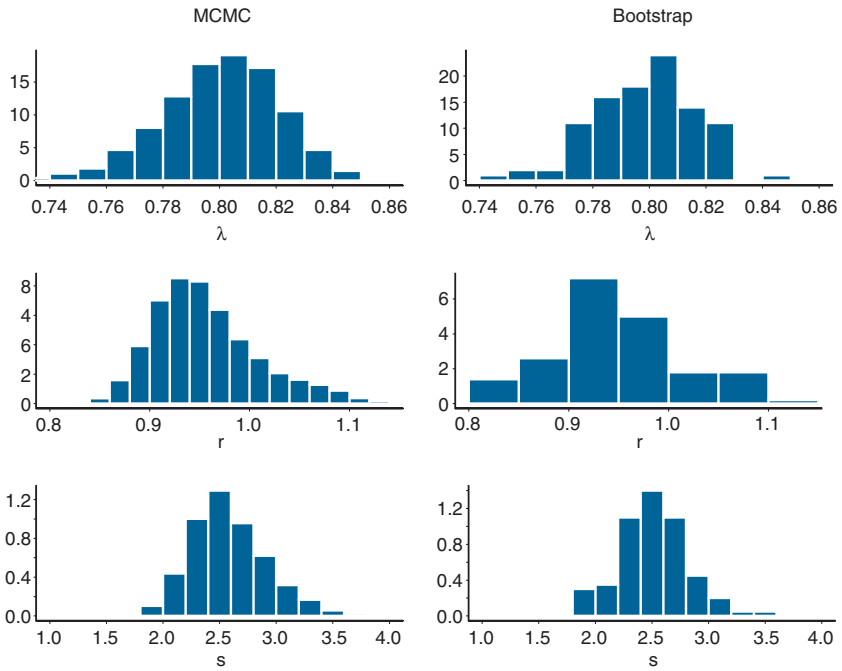

Fig. 7. Column 1 are the simulated MCMC posterior distributions (15,000 values) of parameters and column 2 are the distribution of 100 "bootstrapped" MLEs for data set 2. All plots are relative frequency histograms.

simulated bootstrap sampling distribution of MLEs (column 2) for data set 2. The MCMC posterior was simulated using $N = 20,000$ MCMC simulations and discarding a burn-in period of the first 5000. The MCMC tuning parameters were $(t_1, t_2, t_3) = (0.02, 0.02, 0.2)$. The bootstrap sampling distributions are MLE's computed from 100 bootstrap samples. Comparison of the two columns (i.e., MCMC versus bootstrap) is not ideal because the first are histograms of 15,000 values and the second only 100 values, but it is still interesting to note that a similar shape and range of values is apparent in both. A similar observation was obtained from the same plots for data set 1 (plots not shown).

Also shown in Table 3 are the computing times for the 3 methods for both data sets. The Hessian method executes very quickly, and the bootstrap method executes more quickly than the MCMC method. All three methods were executed for both data sets on a Dell Desktop Dimension 8250 with CPU speed of 2.8 GHz and with 1024 MB RAM.

5. More discussion of computational methods

Three computational methods were chosen to evaluate the precision of the estimates and each one offered advantages and limitations. The biggest advantage of the Hessian method is its computational efficiency. It executes quickly and has good accuracy when the likelihood function has a well-defined maximum. But this method is questionable when the log-likelihood surface becomes flat due to, as mentioned earlier, numerical difficulties in finding a global maximum and near singularities in the observed information matrix. Since the variance is just the reciprocal of the curvature of the log-likelihood surface at the MLE, a flat likelihood surface around the MLE yields a large variance.

We experimented, using the Hessian method, with cases representing very weak signals. When $r = s = 1.1$, coverage for r and s was good but coverage for λ was poor, regardless of the true value for λ (i.e., generally less than 60% coverage). This would represent cases for which there are genes that are differentially expressed, but their magnitude of differential expression small. Also considered was the case when the true value of $\lambda = 0.95$, $r = 0.9$. Coverage for all parameters was good as long as $s \geqslant 4$. This suggests that the mixture model is useful for modeling P-values for situations where 5% of genes are differentially expressed, as long as the magnitude of differential expression is large enough to differentiate the beta-distribution component from the uniform. If the beta distribution "looks" like the uniform, identifiability issues arise when estimating λ.

Finding the MLE's in cases such as these entail careful selection of starting values and could require using different numerical optimizers. In fact, when the observed information matrix suggested flatness of the likelihood at the discovered maximum, the bootstrap method and MCMC were also more difficult to implement and required more care to ensure global maxima are attained or MCMC convergence had occurred. Thus, the observed value of the Hessian can provide advance insight into anticipated difficulties for bootstrap methods or MCMC convergence.

The bootstrap method works well with a strong signal and large k. However, it is computationally intensive since for each bootstrap sample, MLEs must be numerically computed. For small k and weak signals, the bootstrap intervals sometimes appeared to be unstable since MLEs in some bootstrap samples are difficult to obtain and may not represent a global maximum of the likelihood surface. The work around for this issue was to monitor the computed MLEs and to compute the observed information matrix. In cases where the information matrix yielded negative variance estimates, or if the MLE occurred at the boundary, new starting values (up to 25 combinations) for the optimizer were used. If this did not work, a new set of P-values was resampled inside the bootstrap loop.

The t-distribution based bootstrap interval is rather simplistic relative to more modern developments. The computational demand of computing MLEs for each bootstrap sample prevented us from evaluating these here. The computational demand resulted from a combination of the numerical challenges described above, and the fact that the bootstrap loops were nested inside simulation loops. The cluster used for our simulations did not allow execution times for a submitted job that exceeded 96 h. Recoding the simulation software into a language other than the R may improve efficiency.

As a comparison, however, we computed a simple bias corrected percentile interval (Efron, 1982) for a simulation case using $B = 1000$ bootstrap samples. The parameter values were $k = 10,000$ and $(\lambda, r, s) = (0.7, 0.8, 2)$. The percent coverage using the t-distribution based bootstrap interval for λ was 96.2% with an average length of 0.300 and a median length of 0.170. Of the 38 confidence intervals that failed to cover λ, 35 failed on the lower bound and 3 on the upper bound. Using the bias corrected percentile interval, percent coverage was 94.8% with a mean length 0.189, median length 0.151, and of the 52 failures, 25 were at the lower bound and 27 at the upper bound. The bias corrected interval was less sensitive to extreme bootstrap samples than the interval using the t-distribution, and it also helped to correct for asymmetry in the bootstrapped sampling distribution. Coverage for TP was conservative using the t-distribution-based interval and became much closer to the nominal 95% coverage using the bias corrected percentile interval. For example, the coverage for TP at $\tau = 0.001$ went from 0.980 to 0.958. Similar to results for λ, the mean length decreased slightly and number of failures in coverage at the lower and at the upper bounds became much closer.

The Bayesian approach combined with MCMC provides a simulation of the full posterior distribution of the parameters, and hence, is not affected by the numerical difficulties when obtaining MLEs. The Bayesian approach also yields a posterior distribution of the parameters providing for direct probability statements about the true parameter values. The MLE and bootstrap methods provide frequentist intervals that may contain the true value, but no probability statements about the true value. Despite this difference in interpretation between the interval estimates, we have computed and evaluated the stability and performance of Bayesian support intervals using their "frequentist properties," meaning how often they contain a true value.

A disadvantage is that MCMC can have convergence issues that were discussed in Section 3.3. How to choose appropriate tuning variables for the case when k is small and the signal weak is important and can be challenging as well as computationally demanding. For the uniform beta-mixture model, one expects that r will be less than 1 and s greater than 1, since the beta-distribution component is expected to be monotonically decreasing with a mode near zero. Thus one could restrict values for r and s in the MCMC implementation. We did not do this for our simulations but did experiment with some test cases and found that convergence is less problematic with these restrictions.

We chose a simple random walk algorithm to simulate the posterior. This technique is easy to understand and implement by users, and it is one of the earliest MCMC methods. Like the bootstrap methods, there are many more sophisticated MCMC techniques that may perform better than the random walk algorithm for obtaining confidence intervals. See, for example, Robert and Casella (2004) for a presentation of some of these.

6. Conclusions and recommendations

The mixture model fitted to a distribution of P-values yields useful information regarding the proportion of "truly significant results," denoted TP herein. The uncertainty in estimates of TP depends on uncertainty of parameter estimates in the mixture model. Though it is usually not difficult to obtain point estimates of parameters, the uncertainty associated with these estimates is more challenging to assess, depending on true values of parameters and number of P-values. And there are many computational tools available to assess this uncertainty, each one offering different advantages. This paper explored these topics.

An advantage of MLEs is their large sample optimality properties. However, we found that the number of P-values (i.e., samples) must be much larger than originally thought for the particular mixture model considered here. Difficulties with estimation occurred even when $k = 1000$. In general, the Hessian, MCMC, and bootstrap methods could be used together to see if the results from the different methods are similar. In fact, in simulation cases when the interval estimates from the three methods were similar, we felt some comfort in the accuracy of the point estimate and the interval estimate. This similarity of interval estimates was also present in results for the two example data sets.

Thus some guidelines gleaned from this study would be to first, use the Hessian method (perhaps using multiple starting values in the numerical optimizer) to evaluate the curvature of the likelihood near the optimum. If the different starting values result in the same optimum and variance estimates are relatively small, then the Hessian method could be used for interval estimates. Otherwise, in these cases the bootstrap method may produce a narrower interval with coverage closer to nominal, particularly if using one of the more sophisticated methods based on bootstrap percentiles. If numerical difficulties locating a global optimum are encountered and the information matrix is near singularity, the MCMC method may be useful. In such cases, more attention could be given to convergence issues with MCMC and the simulated posterior may yield useful information regarding ridges or multiple optima in the likelihood surface.

This paper considered the sampling variability of estimates due to repeated sampling from the mixture model itself. Correlated P-values would add to the variance of estimates than what was seen in our results. Incorporating uncertainty due to this correlation would likely require simulating gene expression data from some probability distribution and computing the P-values from the simulated data as was done to some extent in Allison et al. (2002) and Gadbury et al. (2003). This probability distribution would need to be highly dimensional multivariate. Computing "correlated" P-values from simulated gene expression data while maintaining known marginals that are the uniform beta-mixture model seemed problematic and is a subject of continuing research involving cupolas (cf., Genest and MacKay, 1986). What type of correlation structure is reasonable for microarray data is also an open question (Mehta et al., 2004).

In practice one may be interested in variability due to sampling of biological specimens for the study. In experiments with larger numbers of arrays, treating the arrays as a "population" and sampling sub-samples of arrays has been used to quantify uncertainty (Pepe et al., 2003). This idea is similar in concept to what has been termed "plasmode" analysis where data from actual microarray studies are used as templates for simulations. Such approaches may also preserve correlation structures in simulated data (Allison et al., in press). How these simulation techniques could be used to assess the precision of estimates for mixture model parameters is a topic of current investigation.

Acknowledgments

Research supported in part by NSF Grants 0090286 and 0217651, and NIH Grant U54CA100949. The authors thank John Mountz for supplying data set 1 and Tsuyoshi Kayo, Tomas Prolla, and Richard Weindruch for supplying data

set 2. The authors also acknowledge helpful discussions with researchers at the Section on Statistical Genetics, School of Public Health, University of Alabama at Birmingham. In particular, Nengjun Yi offered helpful suggestion regarding MCMC. We thank the editor, associate editor, and three referees for helpful reviews that improved the clarity and content of the paper.

References

Allison, D.B., Gadbury, G.L., Heo, M., Fernanfez, J.R., Lee, C., Prolla, T.A., Weindruch, R., 2002. A mixture model approach for the analysis of microarray gene expression data. Comput. Statist. Data Anal. 39, 1–20.

Allison, D.B., Cui, X., Page, G.P., Sabripous, M., Microarray data analysis: from dis-array to consolidation and consensus. Nat. Rev. Genet., in press.

Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. Roy. Statist. Soc. B 57 (1), 289–300.

Benjamini, Y., Hochberg, Y., 2000. On the adaptive control of the false discovery rate in multiple testing with independent statistics. J. Educat. Behavioral Statist. 25, 60–83.

Boik, R.J., Robison-Cox, J.F., 1998. Derivative of the incomplete beta function. Online available at: http://www.jstatsoft.org/v03/i01/beta.der.pdf.

Celeux, G., Hurn, M., Robert, C.P., 2000. Computational and inferential difficulties with mixture posterior distributions. J. Amer. Statist. Assoc. 95, 957–970.

Chib, S., Greenberg, E., 1995. Understanding the Metropolis–Hastings algorithm. Amer. Statist. 49 (4), 327–335.

Chung, H., Loken, E., Schafer, J.L., 2004. Difficulties in drawing inferences with finite-mixture models: a simple example with a simple solution. Amer. Statist. 58 (2), 152–158.

Cowles, K., Best, N., Vines, K., 1997. Convergence Diagnosis and Output Analysis Software (CODA). Online available at: http://www.mrc-bsu.cam.ac.uk/bugs/classic/coda04/readme.shtml#facilities.

Delongchamp, R.R., Bowyer, J.F., Chen, J.J., Kodell, R.L., 2004. Multiple-testing strategy for analyzing cDNA array data on gene expression. Biometrics 60 (3), 774–882.

Dias, J.G., Wedel, M., 2004. An empirical comparison of EM, SEM, and MCMC performance for problematic Gaussian mixture likelihoods. Statist. Comput. 14 (4), 323–332.

Efron, B., 1982. The Jackknife, the Bootstrap, and Other Resampling Plans. Society for Industrial and Applied Mathematics, Philadelphia.

Efron, B., 2004. Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. J. Amer. Statist. Assoc. 99, 96–104.

Efron, B., Hinkley, D.V., 1978. Assessing the accuracy of the maximum likelihood estimator: observed versus expected Fisher information. Biometrika 65 (3), 457–487.

Efron, B., Tibshirani, R.J., 1993. An Introduction to the Bootstrap. Chapman & Hall, New York.

Efron, B., Tibshirani, R.J., 2002. Empirical bayes methods and false discovery rates for microarrays. Genet. Epidemiol. 23, 70–86.

Gadbury, G.L., Page, G.P., Heo, M., Mountz, J.D., Allison, D.B., 2003. Randomization tests for small samples: an application for genetic expression data. Appl. Statist. 52 (3), 365–376.

Gadbury, G.L., Page, G.P., Edwards, J., Kayo, T., Prolla, T.A., Weindruch, R., Permana, P.A., Mountz, J.D., Allison, D.B., 2004. Power and sample size estimation in high dimensional biology. Statist. Methods Med. Res. 13 (4), 325–338.

Gannoun, A., Saracco, J., Urfer, W., Bonney, G.E., 2004. Nonparametric analysis of replicated microarray experiments. Statist. Modeling 4, 195–209.

Gelman, A., Rubin, D.B., 1992. Inference from iterative simulation using multiple sequences. Statist. Sci. 7, 457–511.

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 2004. Bayesian Data Analysis. second ed. Chapman & Hall/CRC, Boca Raton.

Genest, C., MacKay, J., 1986. The joy of cupolas: bivariate distributions with uniform marginals. Amer. Statist. 40, 280–284.

Geweke, J., 1992. Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In: Bernardo, J.M., Berger, J.O., Dawid, A.P., Smith, A.F.M. (Eds.), Bayesian Statistics, vol. 4. Oxford University Press, Oxford.

Heidelberger, P., Welch, P., 1983. Simulation run length control in the presence of an initial transient. Oper. Res. 31, 1109–1144.

Hung, H.M., O'Neill, R.T., Bauser, P., Köhne, K., 1997. The Behavior of the p-value when the alternative hypothesis is true. Biometrics 53, 11–22.

Knudsen, S., 2002. A Biologist's Guide to Analysis of DNA Microarray Data. Wiley–Interscience, New York.

Lee, M.L.T., Kuo, F.T., Whitmore, G.A., Sklar, J., 2000. Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. Proc. Nat. Acad. Sci. 97 (18), 9834–9839.

Link, W.A., Cam, E., Nichols, J.D., Cooch, E.G., 2001. Of bugs and birds: Markov chain monte carlo for hierarchical modeling in wildlife research. J. Wildlife Manage. 66 (2), 277–291.

McLachlan, S., Peel, D., 2000. Finite Mixture Model. Wiley, New York.

Mehta, T., Tanik, M., Allison, D.B., 2004. Towards sound epistemological foundations of statistical methods for high-dimensional biology. Natur. Genet. 36 (9), 943–947.

Mengersen, K., Knight, S., Robert, C.P., 1999. MCMC: how do we know when to stop? Online available at: http://www.stat.fi/isi99/proceedings/arkisto/varasto/meng0251.pdf.

Neuhauser, M., Lam, F.C., 2004. Nonparametric approaches to detecting differentially expressed genes in replicated microarray experiments. In: Proceedings of the Second Asia-Pacific Bioinformatics Conference (APBC2004), Dunedin, New Zealand; Chen, Y.P. (Ed.), Conferences in Research and Practice in Information Technology, vol. 29, ACS, pp. 139–143.

Parker, R.A., Rothenberg, R.B., 1988. Identifying important results from multiple statistical tests. Statist. Med. 17, 1031–1043.

Pepe, M.S., Longton, G., Anderson, G.L., Schummer, M., 2003. Selecting differentially expressed genes from microarray experiments. Biometrics 59, 133–142.

Pounds, S., Morris, S.W., 2003. Estimating the occurrence of false positive and false negative in microarry studies by approximating and partitioning the empirical distribution of p-values. Bioinformatics 19 (10), 1236–1242.

Raftery, A.L., Lewis, S., 1992. How many iterations in the Gibbs sampler? In: Bernardo, J.M., Berger, J.O., Dawid, A.P., Smith, A.F.M. (Eds.), Bayesian Statistics, vol. 4. Oxford University Press, Oxford.

Roberts, G.O., 1996. Markov chain concepts related to sampling algorithms. In: Gilks, W.R., Richardson, S., Spiegelhalter, D.J. (Eds.), Markov Chain Monte Carlo in Practice. Chapman & Hall, New York, pp. 45–57.

Robert, C.R., Casella, G., 2004. Monte Carlo Statistical Methods. second ed. Springer Science and Business Media Inc., New York.

Sackrowitz, H., Samuel-Cahn, E.P., 1999. P values as random variables-expected p values. Amer. Statist. 53 (4), 326–331.

Schweder, T., Spjøtvoll, E., 1982. Plots of p-values to evaluate many tests simultaneously. Biometrika 69 (3), 493–502.

Smith, B.J., 2003. Bayesian outpour analysis program (BOA). Online available at: http://www.public-health.uiowa.edu/boa/.

Speed, T., 2003. Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall, New York.

Storey, J.D., 2002. A direct approach to false discovery rates. J. Roy. Statist. Soc. B 64 (3), 479–498.

Tusher, V.G., Tibshirani, R., Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. Proc. Natl. Acad. Sci. 98 (9), 5116–5121.

Wolfsberg, T.G., Wetterstrand, K.A., Guyer, M.S., Collins, F.S., Baxevanis, A.D., 2002. A user's guide to the human genome. Natur. Genet. (supplement) 32, 1–79.

Zhang, H., Hyde, K., Page, G.P., Brand, J.P.L., Zhou, J., Yu, S., Allison, D.B., Hsu, H., Mountz, J.D., 2004. Novel tumor necrosis factor α-regulated genes in rheumatoid arthritis. Arthritis Rheumat. 50 (2), 420–431.