genes

MDPI

Article

# Model-Based Clustering with Measurement or Estimation Errors

**Wanli Zhang** [†] **and Yanming Di** *

Department of Statistics, Oregon State University, Corvallis, OR 97330, USA; zhang_wan_li@lilly.com
* Correspondence: diy@stat.oregonstate.edu
† Current address: Eli Lilly & Company, Shanghai 200021, China.

Received: 27 November 2019; Accepted: 5 February 2020; Published: 10 February 2020

**Abstract:** Model-based clustering with finite mixture models has become a widely used clustering method. One of the recent implementations is MCLUST. When objects to be clustered are summary statistics, such as regression coefficient estimates, they are naturally associated with estimation errors, whose covariance matrices can often be calculated exactly or approximated using asymptotic theory. This article proposes an extension to Gaussian finite mixture modeling—called MCLUST-ME—that properly accounts for the estimation errors. More specifically, we assume that the distribution of each observation consists of an underlying true component distribution and an independent measurement error distribution. Under this assumption, each unique value of estimation error covariance corresponds to its own classification boundary, which consequently results in a different grouping from MCLUST. Through simulation and application to an RNA-Seq data set, we discovered that under certain circumstances, explicitly, modeling estimation errors, improves clustering performance or provides new insights into the data, compared with when errors are simply ignored, whereas the degree of improvement depends on factors such as the distribution of error covariance matrices.

**Keywords:** gaussian finite mixture model; clustering analysis; uncertainty; expectation-maximization algorithm; classification boundary; gene expression; RNA-seq

## 1. Introduction

Model-based clustering [1,2] is one of the most commonly used clustering methods. The authors of [3] introduced the methodology of clustering objects through analyzing a mixture of distributions. The main assumption is that objects within a class share a common distribution in their characteristics, whereas objects from a different class will follow a different distribution. The entire population will then follow a mixture of distributions, and the purpose of clustering would be to take such a mixture and analyze it into simple components and estimate the "probabilities of membership", that is, the probabilities that each observation belongs to each cluster.

One of the most recent implementations of model-based clustering is MCLUST [4–6], in which each observation is assumed to follow a finite mixture of multivariate Gaussian distributions. MCLUST describes cluster geometries (shape, volume, and orientation) by reparameterizing component covariance matrices [7], and formulates different models by imposing constraints on each geometric feature. The expectation-maximization (EM) algorithm [8,9] is used for maximum likelihood estimation, and the Bayesian information criterion (BIC) [10,11] is used for selection of optimal model(s).

In most cases, observations to be clustered are assumed to have been precisely measured, whereas there are situations where this assumption is clearly not feasible. This article proposes an extension to Gaussian mixture modeling that properly accounts for measurement or estimation errors in the special case when the error distributions are either known or can be estimated, as well as introduces the clustering algorithm built upon it, which we named MCLUST-ME. The real data example that motivated our study is where we apply clustering algorithm to coefficients from gene-wise regression analysis of an RNA-seq data set (see Section 3.3 for details). For each gene, five of the fitted regression coefficients correspond to log fold changes in mean expression levels between two groups of *Arabidopsis* plants at five time points after treatment. In such a case, we can reasonably approximate the error covariances of the regression coefficients by inverting the observed information matrix. In general, whenever one applies clustering analysis to a set of summary statistics, it is often possible to approximate the distribution of their estimation errors with, for instance, Gaussian distributions. In this paper, we describe how the estimation/measurement errors, with known or estimated error covariances, can be incorporated into the model-based clustering framework. An obvious alternative strategy in practice is to ignore the individual estimation/measurement errors. We will use simulations and the real data example to understand in what circumstances explicitly modeling the estimation errors will improve the clustering results, and to what degree.

In Section 3.3, we will compare the results of applying the MCLUST method and our new MCLUST-ME method to cluster the log fold changes of 1000 randomly selected genes from the RNA-seq data set mentioned above at two of the time points where the gene expressions were most active. Here, we briefly summarize the input data structure for the clustering analysis and the highlights of the results. Columns 2 and 3 of Table 1 list the estimated log fold changes and their standard errors for 15 representative genes at the two time points being analyzed: these are from the regression analysis applied to each row (gene) of the RNA-seq data set. (The standard errors are the square roots of the corresponding diagonal entries of the error covariances.) In particular, we included 10 genes that are classified differently by the MCLUST and the MCLUST-ME methods. We note that there is sizable variation among the standard errors of the log fold changes. When MCLUST was used to cluster the log fold changes, the estimation errors will be ignored: As long as two genes have the same log fold changes at the two time points, they will always belong to the same cluster. However, we understand that, in this context, a moderate log fold change with a high estimation error is less significant than the same log fold change with low estimation error: this would be obvious if we were to perform a hypothesis test for differential expression (DE), but existing clustering methods such as MCLUST cannot readily incorporate such information into a clustering analysis. The MCLUST-ME method we propose in this paper aims to incorporate information about the estimation errors into the clustering analysis. One distinctive feature of the new MCLUST-ME method is that two points with similar log fold changes may not belong to the same cluster: it also depends on the error covariances of the log fold changes. We note that when the 1000 genes were classified into two clusters by MCLUST and MCLUST-ME, the two clusters for this data set roughly correspond to a "DE" cluster and a "non-DE" cluster. Columns 4 and 5 of Table 1 list the probabilities to the "non-DE" cluster estimated by MCLUST and by MCLUST-ME. We see that the genes that were classified into the "non-DE" cluster by MCLUST-ME, but to the "DE" cluster by MCLUST tend to be genes having moderate log fold changes, but relatively large error covariances. We do not have the ground truth for this data set, but the results from the new MCLUST-ME method alert us that not all log fold changes are created equal.

The organization of the rest of this article is as follows. Section 2 briefly reviews the MCLUST method and then introduces our extension, MCLUST-ME. In particular, Section 2.6 investigates decision boundaries of the two methods for two-group clustering. Sections 3.1 and 3.2 give simulation settings and results on comparing MCLUST-ME with MCLUST in terms classification accuracy and uncertainty. Section 3.3 gives an example where we cluster a real-life data set using both methods. Finally, conclusions and perspectives for future work are addressed in Section 4.

**Table 1.** Estimated log fold changes at two time points, associated standard errors, and estimated membership probabilities to the "non-DE" cluster by MCLUST and by MCLUST-ME, for 15 genes selected from the real-data example. Column 2 and 3 are estimated log fold changes at 1 h and 3 h and their standard errors. Column 4 and 5 are estimated membership probabilities to the "non-DE" cluster by MCLUST and by MCLUST-ME. The first 5 rows are randomly selected from 1000 genes that we analyzed. The second 5 rows are selected among the genes that are classified to the "non-DE" cluster by MCLUST, but to the "DE" cluster by MCLUST-ME: the standard errors of the log fold changes tend to be low in this group; the last 5 rows are selected among genes that are classified to the "DE" cluster by MCLUST, but to the "non-DE" cluster by MCLUST-ME: the standard errors of the log fold changes tend to be high in this group.

| Gene ID | Log Fold Change (SE) 1h | Log Fold Change (SE) 3h | $z_1$ MCLUST | $z_1$ MCLUST−ME |
|---|---:|---:|---:|---:|
| AT2G42230 | −0.277 (0.006) | 0.152 (0.006) | 0.920 | 0.921 |
| AT3G56110 | 0.081 (0.121) | 0.228 (0.099) | 0.919 | 0.895 |
| AT1G23330 | 0.351 (0.018) | −0.209 (0.012) | 0.862 | 0.870 |
| AT5G23060 | −0.243 (0.005) | −0.909 (0.005) | 0.684 | 0.751 |
| AT5G06240 | −0.680 (0.022) | 0.103 (0.012) | 0.774 | 0.764 |
| AT3G20350 | −0.952 (0.007) | −0.090 (0.009) | 0.562 | 0.396 |
| AT1G30440 | −1.056 (0.010) | −0.398 (0.009) | 0.511 | 0.375 |
| AT1G30490 | −0.983 (0.008) | −0.322 (0.006) | 0.612 | 0.480 |
| AT1G23400 | −1.017 (0.011) | −0.275 (0.006) | 0.547 | 0.418 |
| AT1G17980 | 0.734 (0.001) | −0.001 (0.006) | 0.524 | 0.363 |
| AT2G30890 | −1.040 (0.150) | −0.142 (0.125) | 0.445 | 0.771 |
| AT5G15160 | −0.044 (0.129) | −1.059 (0.225) | 0.332 | 0.866 |
| AT5G45310 | −0.221 (0.162) | −1.404 (0.313) | 0.042 | 0.837 |
| AT5G46871 | 0.373 (0.065) | 0.886 (0.094) | 0.305 | 0.581 |
| AT2G22240 | 0.076 (0.016) | −0.975 (0.043) | 0.371 | 0.690 |

## 2. Materials and Methods

### 2.1. Review of MCLUST Model

**Finite mixture model** Let $f_1(\boldsymbol{y};\boldsymbol{\Theta}_1), f_2(\boldsymbol{y};\boldsymbol{\Theta}_2), ..., f_G(\boldsymbol{y};\boldsymbol{\Theta}_G)$ be $G$ probability distributions defined on the $d$-dimensional random vector $\boldsymbol{y}$, and a mixture of the $G$ distributions is formed by taking proportions $\{\tau_k\}$ of the population from components $\{f_k\}$, with probability density given by

$$f(\boldsymbol{y};\boldsymbol{\Theta}) = \sum_{k=1}^{G} \tau_k f_k(\boldsymbol{y};\boldsymbol{\Theta}_k), \tag{1}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\Theta}_1, ..., \boldsymbol{\Theta}_G)$ are model parameters.

**Component density** The MCLUST model assumes that the distribution of each $\boldsymbol{y}$ is a mixture of multivariate normal distributions. Under the MCLUST model, the component density of $\boldsymbol{y}$ in group $k$ is

$$f_k(\boldsymbol{y};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) = \frac{\exp\left\{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_k)\right\}}{\sqrt{\det[2\pi\boldsymbol{\Sigma}_k]}}, \tag{2}$$

In other words,

$$\boldsymbol{y}|k \sim N_d(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k). \tag{3}$$

The (marginal) probability density of $\boldsymbol{y}$ is given by

$$f(\boldsymbol{y}) = \sum_{k=1}^{G} \tau_k f_k(\boldsymbol{y};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k). \tag{4}$$

**Likelihood function** Suppose a sample of $n$ independent and identically distributed (iid) random vectors $\boldsymbol{y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_n)$ is drawn from the mixture. The (observed) log likelihood of the sample is then

$$l_O(\boldsymbol{\Theta}; \boldsymbol{y}) = \sum_{i=1}^{n} \log f(\boldsymbol{y}_i) = \sum_{i=1}^{n} \log \sum_{k=1}^{G} \tau_k f_k(\boldsymbol{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{5}$$

where $\boldsymbol{\Theta} = (\tau_1, ..., \tau_G; \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_G; \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_G)$ are the model parameters.

### 2.2. MCLUST-ME Model

We extend the MCLUST model by associating each data point with an error term and assumes that the covariance matrix of each error term is either known or can be estimated.

**Component density** Given that $\boldsymbol{y}$ belongs to component $k$, the MCLUST-ME models assumes that there exists a latent variable $\boldsymbol{w}$, representing its “truth” part, and $\boldsymbol{\epsilon}$, representing its “error” part, such that

$$\begin{cases} \boldsymbol{y} = \boldsymbol{w} + \boldsymbol{\epsilon}, \\ \boldsymbol{w} | k \sim N_d(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \\ \boldsymbol{\epsilon} \sim N_d(\boldsymbol{0}, \boldsymbol{\Lambda}), \end{cases} \tag{6}$$

where $\boldsymbol{w}$ and $\boldsymbol{\epsilon}$ are independent. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are unknown mean and covariance parameters (same as in the MCLUST model), and $\boldsymbol{\Lambda}$ is the known error covariance matrix associated with $\boldsymbol{y}$. The distribution of $\boldsymbol{y}$ being in component $k$ is then

$$\boldsymbol{y} | k \sim N_d(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k + \boldsymbol{\Lambda}), \tag{7}$$

with density function

$$g_k(\boldsymbol{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Lambda}) = \frac{\exp\left\{-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_k)^T (\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda})^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_k)\right\}}{\sqrt{\det[2\pi(\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda})]}}, \tag{8}$$

and the (marginal) probability density of $\boldsymbol{y}$ is given by

$$g(\boldsymbol{y}) = \sum_{k=1}^{G} \tau_k g_k(\boldsymbol{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Lambda}). \tag{9}$$

**Likelihood function** Suppose a sample of $n$ iid random vectors $\boldsymbol{y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_n)$ is drawn from the mixture, where each $\boldsymbol{y}_i$ is associated with known error covariance matrix $\boldsymbol{\Lambda}_i$. The (observed) log likelihood of the sample is then

$$l_O(\boldsymbol{\Theta}; \boldsymbol{y}) = \sum_{i=1}^{n} \log g(\boldsymbol{y}_i) = \sum_{i=1}^{n} \log \sum_{k=1}^{G} \tau_k g_k(\boldsymbol{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Lambda}_i), \tag{10}$$

where $\boldsymbol{\Theta} = (\tau_1, ..., \tau_G; \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_G; \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_G)$ are the model parameters.

In summary, the MCLUST-ME and MCLUST models have the same set of model parameters for the normal components and the mixing proportions. The key difference is that under the MCLUST-ME model, the measurement or observation errors of the observations are explicitly modeled, and observations are each associated with a given error covariance matrix.

### 2.3. Expectation-Maximization (EM) Algorithm

In the original MCLUST method, the EM algorithm is used to estimate the unknown parameters and compute the membership probabilities. In this subsection, we will first review the EM algorithm under the general MCLUST framework, and then highlight the differences in implementation between the MCLUST method and the MCLUST-ME method.

**Complete data log likelihood** Given observations $(\boldsymbol{y}_1, ..., \boldsymbol{y}_n)$, suppose that each $\boldsymbol{y}_i$ is associated with one of $G$ states. Then, there exists unobserved indicator vectors $\{\boldsymbol{z}_i = (z_{i1}, ..., z_{iG})\}$ where $\boldsymbol{z}_i \overset{\text{iid}}{\sim} \text{Mult}_G(1, \boldsymbol{\tau})$ with $\boldsymbol{\tau} = (\tau_1, ..., \tau_G)$. The complete data then consists of $\boldsymbol{x}_i = (\boldsymbol{y}_i, \boldsymbol{z}_i)$. Assuming that the conditional probability density of $\boldsymbol{y}_i$ given $\boldsymbol{z}_i$ is $\prod_{k=1}^{G} f_k(\boldsymbol{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Lambda}_i)^{z_{ik}}$, the complete data log likelihood can be derived as follows,

$$\begin{aligned} l_C &= \log \prod_{i=1}^{n} f(\boldsymbol{y}_i, \boldsymbol{z}_i) \\ &= \log \prod_{i=1}^{n} f(\boldsymbol{y}_i | \boldsymbol{z}_i) f(\boldsymbol{z}_i) \\ &= \log \prod_{i=1}^{n} \left[ \prod_{k=1}^{G} f_k(\boldsymbol{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Lambda}_i)^{z_{ik}} \right] \left[ \prod_{k=1}^{G} \tau_k^{z_{ik}} \right] \\ &= \log \prod_{i=1}^{n} \prod_{k=1}^{G} \left[ \tau_k f_k(\boldsymbol{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Lambda}_i) \right]^{z_{ik}} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{G} z_{ik} \log \left[ \tau_k f_k(\boldsymbol{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Lambda}_i) \right]. \end{aligned} \tag{11}$$

**EM iterations** The EM algorithm consists of iterations of an *M step* and an *E step*, as described below.

- *M step*: Given current estimates of $\{z_{ik}\}$, maximize the complete-data log-likelihood $l_C$ with respect to $(\tau_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- *E step*: Given estimates $(\hat{\tau}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$ from last *M step*, for all $i = 1, ..., n$ and $k = 1, ..., G$, compute the membership probabilities
$$\hat{z}_{ik} = \frac{\hat{\tau}_k f_k(\boldsymbol{y}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k, \boldsymbol{\Lambda}_i)}{\sum_{j=1}^{G} \hat{\tau}_j f_j(\boldsymbol{y}_i; \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j, \boldsymbol{\Lambda}_i)}. \tag{12}$$

The two steps alternate until the increment in $l_O$ is small enough. Upon convergence, a membership probability matrix is produced and each observation is assigned to the most probable cluster, that is,

$$\text{membership of } \boldsymbol{y}_i = \text{argmax}_k \{\hat{z}_{ik}\}, \tag{13}$$

and the classification uncertainty for $\boldsymbol{y}_i$ is defined as

$$1 - \max_k \{\hat{z}_{ik}\}. \tag{14}$$

In two-group clustering, the classification uncertainty cannot exceed 0.5 (otherwise the point is incorrectly assigned).

For MCLUST, the component density $f_k$ is defined in (2), and for MCLUST-ME, $f_k$ is substituted by $g_k$ in (8).

**M-step implementation details** For likelihood maximization in the *M step*, a closed-form solution always exists for $\hat{\tau}_k$, $k = 1, \ldots, G$ (see [12] for more details):

$$\hat{\tau}_k = \frac{1}{n} \sum_{i=1}^{n} z_{ik} \tag{15}$$

We can derive the estimation equations for $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ ($k = 1, \ldots, G$) by taking the partial derivatives of $l_C$ with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ and setting the derivatives to $\mathbf{0}$. For MCLUST-ME (see [13] for a summary of useful matrix calculus formulas, in particular (11.7) and (11.8)):

$$\frac{\partial l_C}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{n} z_{ik}(\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda}_i)^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_k) = \mathbf{0} \tag{16}$$

and

$$\frac{\partial l_C}{\partial \boldsymbol{\Sigma}_k} = \frac{1}{2}\sum_{i=1}^{n} z_{ik}(\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda}_i)^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_k)(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T(\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda}_i)^{-1} - \frac{1}{2}\sum_{i=1}^{n} z_{ik}(\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda}_i)^{-1} = \mathbf{0}. \tag{17}$$

For estimation equations under the MCLUST model, one set all the $\boldsymbol{\Lambda}_i$'s to $\mathbf{0}$ in the above two equations. Note that under MCLUST, if there is no constraint on $\boldsymbol{\Sigma}_k$, there are closed-form solutions for $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$:

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^{n} z_{ik}\boldsymbol{y}_i}{\sum_{i=1}^{n} z_{ik}} \tag{18}$$

and

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\sum_{i=1}^{n} z_{ik}(\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_k)^T}{\sum_{i=1}^{n} z_{ik}}. \tag{19}$$

For MCLUST-ME, each $\boldsymbol{y}_i$ corresponds to a different $\boldsymbol{\Lambda}_i$. One can see that, in general, there is no closed-form solution for $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. In our implementation of the MCLUST-ME M step, we solve $\boldsymbol{\mu}_k$ from (16),

$$\hat{\boldsymbol{\mu}}_k = \left[\sum_{i=1}^{n} z_{ik}(\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda}_i)^{-1}\right]^{-1} \sum_{i=1}^{n} z_{ik}(\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda}_i)^{-1}\boldsymbol{y}_i, \tag{20}$$

and plug it into (17), and then use the limited-memory BFGS, a quasi-Newton method in R (function `optim` [14]), to obtain an optimal solution for $\boldsymbol{\Sigma}_k$ numerically. We obtain $\hat{\boldsymbol{\mu}}_k$ by substituting the resulting $\hat{\boldsymbol{\Sigma}}_k$ into (20).

The complexity of the EM algorithm for the MCLUST-ME increase with the number of clusters, the number of parameters (which is determined by the dimension of the data), and the number of observations. It is much slower than the original MCLUST algorithm due to the fact we have to use a numerical optimization routine to find the maximum likelihood estimate (MLE) of $\boldsymbol{\mu}_k$'s and $\boldsymbol{\Sigma}_k$'s in the M step. (See Conclusion and Discussion for a brief summary of running time of MCLUST-ME on the real data example.)

### *2.4. Initial Values*

Owing to its iterative nature, the EM algorithm can start with either an *E step* or an *M step*. In the context of model-based clustering, initiation with the *M step* takes advantage of the availability of other existing clustering methods, in the sense that, given a data set, we can acquire their initial memberships by first clustering the data with other methods. MCLUST adopts model-based agglomerative hierarchical clustering [7,15] to generate initial memberships. Model-based hierarchical clustering aims at maximizing the *classification likelihood* instead of (5) or (10); at each stage, the maximum-likelihood pair of clusters are merged together. Although the resulting partitions are suboptimal due to its heuristic nature, model-based hierarchical clustering has been shown to often yield reasonable results and is relatively easy to compute [16]. In light of this, we also use model-based hierarchical clustering to obtain initial memberships for MCLUST-ME. For the choice of initial values when starting with *E step* (i.e., initial parameter estimates), see [17] for a nice discussion.

### 2.5. Model Selection

Within MCLUST framework, selection for the number of clusters can be achieved through the use of the Bayesian information criterion (BIC). Given a random sample of $n$ independent $d$-vectors $\boldsymbol{y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_n)$ drawn from (4) and (9) with some value of $G$, the BIC for this $G$-component mixture model is given by:

$$BIC_G = 2l_O(\hat{\boldsymbol{\Theta}}; \boldsymbol{y}) - \nu_G \log(n), \tag{21}$$

where $\hat{\boldsymbol{\Theta}}$ is the MLE for model parameters, $l_O$ is the observed likelihood as in (5) or (10), and $\nu_G$ is the number of independent parameters to be estimated. In the most simplistic case, we allow the mean and covariance of each component to vary freely—this is the case we will focus on in this paper. Therefore, for a $G$-component mixture model, we have $\nu_G = (G-1) + Gd + Gd(d-1)/2$. For comparison purpose, in this paper, we will compare MCLUST-ME results to MCLUST results with the same number of components.

### 2.6. Decision Boundaries for Two-Group Clustering

In this subsection, we examine decision boundaries produced by MCLUST and MCLUST-ME for partitioning a sample into $G = 2$ clusters.

#### 2.6.1. MCLUST Boundary

Suppose we would like to separate a $d$-dimensional i.i.d. random sample $S = \{\boldsymbol{y}_i\}_{i=1}^N$ into two clusters with MCLUST. Let $(\hat{\tau}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$ denote MLEs for $(\tau_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ upon convergence. If we assign each point to the more probable cluster, then the two clusters can be expressed as follows.

$$E_1 = \{\boldsymbol{y}_i \in S : \hat{\tau}_1 f_1(\boldsymbol{y}_i; \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1) - \hat{\tau}_2 f_2(\boldsymbol{y}_i; \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}_2) > 0\}; \quad E_2 = S \setminus E_1, \tag{22}$$

and the decision boundary separating $E_1$ and $E_2$ is

$$B = \{\boldsymbol{t} \in \mathbb{R}^d : \hat{\tau}_1 f_1(\boldsymbol{t}; \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1) - \hat{\tau}_2 f_2(\boldsymbol{t}; \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}_2) = 0\}, \tag{23}$$

where $f_k$, $k = 1, 2$, is defined in (2). Equivalently, the boundary $B$ is the set of all points in $\mathbb{R}^d$ with classification uncertainty equal to 0.5. Notice that since the solution set $B$ does not depend on $i$, a common boundary is shared by *all* observations. When $d = 2$, under the model assumption of MCLUST, the boundary $B$ is a straight line when $\hat{\boldsymbol{\Sigma}}_1 = \hat{\boldsymbol{\Sigma}}_2$, and a conic section when $\hat{\boldsymbol{\Sigma}}_1 \neq \hat{\boldsymbol{\Sigma}}_2$, with its shape and position determined by the values of the MLEs. This can be shown by simplifying the equality in (23) (see [12] for more details).

#### 2.6.2. MCLUST-ME Boundary

Consider the data $S = \{\boldsymbol{y}_i\}_{i=1}^N$ and each $\boldsymbol{y}_i$ is associated with known error covariance $\boldsymbol{\Lambda}_i$ for all $i$. Suppose our goal is to partition $S$ into two clusters. Let $(\tilde{\tau}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k)$ be MLEs from the MCLUST-ME model. If we assign each observation to the more probable cluster, the two clusters can be expressed as follows,

$$E_1^* = \{\boldsymbol{y}_i \in S : \tilde{\tau}_1 g_1(\boldsymbol{y}_i; \tilde{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\Sigma}}_1, \boldsymbol{\Lambda}_i) - \tilde{\tau}_2 g_2(\boldsymbol{y}_i; \tilde{\boldsymbol{\mu}}_2, \tilde{\boldsymbol{\Sigma}}_2, \boldsymbol{\Lambda}_i) > 0\}; \quad E_2^* = S \setminus E_1^*,$$

where $g_k$ is defined in (8). The above decision rule (and therefore boundary) of classifying each point $\boldsymbol{y}_i$ now depends not only on the values of MLEs, but also on the error covariance matrix, $\boldsymbol{\Lambda}_i$, of $\boldsymbol{y}_i$. Instead of producing a common boundary for all points in $S$, the MCLUST-ME model specifies an individualized classification boundary for each $\boldsymbol{y}_i$ as follows,

$$B^*(\boldsymbol{\Lambda}_i) = \{\boldsymbol{t} \in \mathbb{R}^d : \tilde{\tau}_1 g_1(\boldsymbol{t}; \tilde{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\Sigma}}_1, \boldsymbol{\Lambda}_i) - \tilde{\tau}_2 g_2(\boldsymbol{t}; \tilde{\boldsymbol{\mu}}_2, \tilde{\boldsymbol{\Sigma}}_2, \boldsymbol{\Lambda}_i) = 0\}.$$

Similar to our argument in Section 2.6.1, when $d = 2$, $B^*(\mathbf{\Lambda}_i)$ is either a straight line or a conic section.

When $\mathbf{\Lambda}_i = \mathbf{\Lambda}_j$ for some $i \neq j$, that is, when two points are associated with the same error covariance, it can be seen that $B^*(\mathbf{\Lambda}_i) = B^*(\mathbf{\Lambda}_j)$, meaning that the two points share a common classification boundary. In the special case where $\mathbf{\Lambda}_i = \mathbf{\Lambda}_j \; \forall i \neq j$, all boundaries $B^*(\mathbf{\Lambda}_i)$ will coincide with each other.

One consequence of the existence of multiple decision boundaries is that the classification uncertainty of each point will depend on its corresponding value of $\mathbf{\Lambda}_i$. In MCLUST, points with high uncertainty ($\approx 0.5$) are aligned around the single classification boundary, whereas in MCLUST-ME, each highly uncertain point is close to its own boundary. Consequently, as we will see in Section 3.1, our method allows intermixing of points belonging to different clusters, while MCLUST creates clear-cut separation between clusters.

### *2.7. Related Methods*

The authors of [18] discussed a clustering method for data with measurement errors. They also assumed that each observation, $\boldsymbol{y}_i$, is associated with a known covariance matrix, $\tilde{\mathbf{\Lambda}}_i$, but they assume that this covariance matrix is for the distance *between the observation and the center of a cluster*. Their conceptual model, using our notation, assumes that

$$\boldsymbol{y}_i | k \sim N_d(\boldsymbol{\mu}_k, \tilde{\mathbf{\Lambda}}_i) \tag{24}$$

when observation $i$ belongs to cluster $k$ (under their model, group membership is deterministic, not probabilistic). Comparing (24) to our MCLUST-ME model (6) and (7), we see that their model lacks the "model-based" element—the covariance matrix $\mathbf{\Sigma}_k$—for each cluster $k$, $k = 1, \ldots, G$. In other words, their $\tilde{\mathbf{\Lambda}}_i$ plays the role of our $\mathbf{\Sigma}_k + \mathbf{\Lambda}_i$. This is a crucial difference: we understand that in MCLUST and MCLUST-ME models, $\mathbf{\Sigma}_k$'s are used to capture different shapes, orientations, and scales of the different clusters. Also, although it is reasonable to assume that the error covariances of the measurements ($\mathbf{\Lambda}_i$ in MCLUST-ME) are known or can be estimated, it is much more difficult to know $\mathbf{\Sigma}_k + \mathbf{\Lambda}_i$ (i.e., $\tilde{\mathbf{\Lambda}}_i$), as we do not where the centers of the clusters are before running the clustering algorithm.

The authors of that paper discussed two heuristic algorithms for fitting $G$ clusters into observations: hError and kError. Under their model, they need to estimate the $\mu_k$'s for all the clusters and the deterministic (or hard) group memberships for each observation. Both algorithms are distance-based, and not based on an EM algorithm. The hError algorithm is a hierarchical clustering algorithm: it iteratively merges two current clusters with the smallest distances. The error covariances $\tilde{\Lambda}_i$ were incorporated into the distance formula. For each current cluster $k$, let $S_k$ be the collection of observations. The center of cluster $k$ is estimated by a weighted average of the observations:

$$\hat{\boldsymbol{\mu}}_k = (\sum_{i \in S_k} \tilde{\mathbf{\Lambda}}_i^{-1})^{-1} \sum_{i \in S_k} \tilde{\mathbf{\Lambda}}_i^{-1} \boldsymbol{y_i} \tag{25}$$

with covariance matrix

$$\mathbf{\Psi}_k = \mathrm{Var}(\hat{\boldsymbol{\mu}}_k) = (\sum_{i \in S_k} \tilde{\mathbf{\Lambda}}_i^{-1})^{-1}. \tag{26}$$

The distance between any two clusters $k$ and $l$ is defined by

$$d_{kl} = (\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}}_l)^T (\mathbf{\Psi}_k + \mathbf{\Psi}_l)^{-1} (\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}}_l) \tag{27}$$

The kError algorithm is an extension of the $k$-means method. It iterates between two steps: (1) Computing the centers of the clusters using (25). (2) Assigning each point to the closest cluster based on the distance formula

$$d_{ik} = (\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_k)^T \tilde{\mathbf{\Lambda}}_i^{-1} (\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_k). \tag{28}$$

We implemented the simpler kError algorithm as described above and applied it the real-data example. We summarized our findings in Section 3.3.

The authors of [19] proposed another extension to the $k$-means method that incorporates errors on individual observations. Under their model, each cluster is characterized by a "profile" $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$, where $m$ is the dimension of the data. Each observation, $\boldsymbol{g}_i = (g_{i1}, \ldots, g_{im})$, from this cluster is modeled as

$$g_{ij} = \beta_i \alpha_j + \gamma_i + \epsilon_{ij}, \quad j = 1, \ldots, m, \tag{29}$$

where $\epsilon_{ij} \sim N(0, \sigma_{ij})$ with known error variances $\sigma_{ij}$. The distance from an observation $\boldsymbol{g}_i$ to a cluster with profile $\boldsymbol{\alpha}$ is defined as

$$\min_{\beta_i, \gamma_i} \sum_{j=1}^{m} \left[ \frac{g_{ij} - (\beta_i \alpha_j + \gamma_i)}{\sigma_{ij}} \right]^2, \tag{30}$$

essentially the weighted sum of squared errors from a weighted least-squares regression of $\boldsymbol{g}_i$ on the profile $\boldsymbol{\alpha}$. The motivation of this distance measure is that it captures both the euclidean distance and the correlation between an observation and a profile. Their version of $k$-means algorithm, CORE, proceeds by iteratively estimating the profile $\boldsymbol{\alpha}$ for each cluster and then assigning each observation $\boldsymbol{g}_i$ to the closest cluster according to (30). We note that their distance measure is less useful for low-dimensional data, as a regression line needs to be fitted between each observation and the cluster profile. If we force the slope $\beta_i$ to be 0, then we see that their method will be similar to the kError method in [18].

## 3. Results

In our simulations and real-data example, version 5.0.1 of MCLUST was used.

### 3.1. Simulation 1: Clustering Performance

We simulated data from bivariate normal mixture distribution with different parameter settings, and applied both MCLUST-ME and MCLUST to partition the data into two clusters. The purpose of this simulation is twofold: first, to investigate the degree of improvement in clustering performance by incorporating known error distributions, and second, to study how error structure affects clustering result.

#### 3.1.1. Data Generation

The data were generated from a two-component bivariate normal mixture distribution, where each point is either error-free or associated with some known, constant error covariance. The data generation process is as follows.

(1) Generate $\{h_i\}_{i=1}^{n}$ i.i.d. from Bernoulli$(\eta)$. For each $i$, $h_i$ will serve as indicator for error, and on average, a proportion $\eta$ of data points will be associated with error.

(2) Generate $\{z_i\}_{i=1}^{n}$ i.i.d. from Bernoulli$(\tau)$. Parameter $\tau$ will be the mixing proportion.

(3) For $i = 1, ..., n$, generate $\boldsymbol{y}_i$ from

$$z_i N_2(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1 + h_i \boldsymbol{\Lambda}) + (1 - z_i) N_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2 + h_i \boldsymbol{\Lambda}).$$

Values of the above parameters are as follows; $\boldsymbol{\mu}_1 = (0,0)^T$, $\boldsymbol{\mu}_2 = (8,0)^T$, $\boldsymbol{\Sigma}_1 = 64I_2$, $\boldsymbol{\Sigma}_2 = 16I_2$, $n = 300$, $\tau_1 = \tau_2 = 0.5$, and $\boldsymbol{\Lambda} = 36I_2$. As the values of $z_i$ provide us with the true memberships of each observation, we are able to use them to evaluate externally the performance of clustering methods in consideration.

### 3.1.2. Simulation Procedure

The simulation proceeds as follows.

(1) Choose a value for $\eta$ from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$.
(2) Randomly select a random seed.
(3) Generate a random sample following Section 3.1.1.
(4) Run MCLUST and MCLUST-ME, fixing $G = 2$. Initiate with true memberships.
(5) Repeat $(2)$–$(4)$ for 100 different seeds.
(6) Repeat $(1)$–$(5)$ for each value of $\eta$.

The membership for each observation as well as MLEs upon convergence will be recorded.

### 3.1.3. The Adjusted Rand Index

In this simulation study, as the true memberships of the observations are available, we can externally evaluate the performance of both clustering methods by calculating the Rand index [20]. Given $n$ observations and two partitions $R$ and $Q$ of the data, we can use a contingency table (Table 2) to demonstrate their agreement.

**Table 2.** 2 × 2 contingency table for comparing partitions $R$ and $Q$.

| **Partition** | $Q$ | |
|---|---|---|
| $R$ | Pair in same group | Pair in different groups |
| Pair in Same Group | a | b |
| Pair in Different Groups | c | d |

The Rand index (RI) is defined as

$$\text{RI} = \frac{a+d}{a+b+c+d}.$$

There are some pitfalls of the Rand index: for two random partitions, the expected value of RI is not equal to zero, and the value of RI tends to one as the number of partitions increases [21]. To overcome these problems, Hubert and Arabie [22] proposed the adjusted Rand index (ARI), which has an expectation of zero. The ARI is defined as

$$\text{ARI} = \frac{\text{RI} - \text{Expected}(\text{RI})}{1 - \text{Expected}(\text{RI})} = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]}.$$

ARI takes values between $-1$ and 1, with an ARI of 1 indicating perfect agreement between two partitions (i.e., RI $=$ 1), and an ARI of 0 indicating independence between partitions (i.e., RI $=$ Expected(RI)).

Permutation tests can be used to test whether the observed ARI is significantly greater than zero [23]. Although keeping the numbers of partitions and partition sizes the same as the original data, a large number of pairs of partitions are generated at random and ARI is computed for each generated pair. A randomization $p$-value can then be calculated based on the distribution of generated ARI's. Similarly, permutation $p$-values can be obtained for testing whether paired ARI values originating from two clustering methods are equal or not.

### 3.1.4. Simulation 1 Results

**Decision boundary** We first visualize the clustering results from both methods, as well as the theoretical decision boundaries stated in Section 2.6. Figure 1 shows groupings of the same data generated with $\eta = 0.5$ and with random seed 7.

For MCLUST-ME, we identify two distinct decision boundaries: The dotted curve separates points measured *with* errors (solid) into two groups, whereas the dashed curve separates points *without* errors (empty). For MCLUST, one boundary separates all points, regardless of their associated errors. This confirms our findings in Section 2.6.

For this particular simulation, we make two interesting discoveries. First, the two MCLUST-ME boundaries are relatively far apart. Second, none of the three boundaries intersect with each other. As mentioned in Section 2.6.1, the shape and position of these boundaries completely depend upon corresponding values of MLEs, which, in turn, are end results of a procedure of iterative nature (the EM algorithm). We have additional plots similar to Figure 1 for other values of $\eta$ and other random seeds in [12].

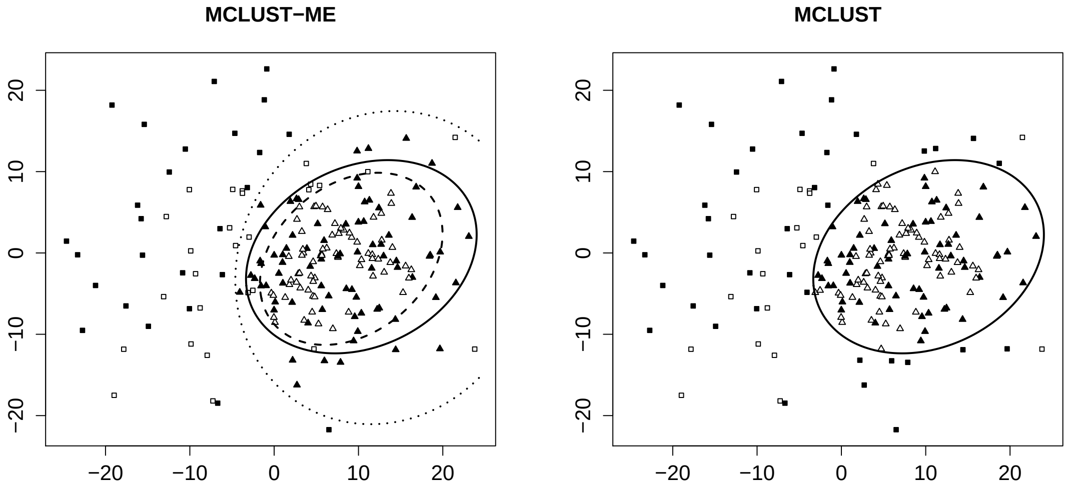


**Figure 1.** Clustering result of the sample generated with random seed = 7 and $\eta = 0.5$. *Both plots*: empty points represent observations with no measurement errors; solid points represent those generated with error covariance $\mathbf{\Lambda}$. Clusters are identified by different shapes. *Left*: clustering result produced by MCLUST-ME. Dashed line represents classification boundary for error-free observations; dotted line represents boundary for those with error covariance matrix $\mathbf{\Lambda}$; solid line represents boundary produced by MCLUST. *Right*: clustering result produced by MCLUST. Solid line is the same as in the left plot.

**Classification uncertainty** In Figure 2, we visualize the classification uncertainty of each point produced by both methods. Observe that for MCLUST, highly uncertain points are found close to the decision boundary, regardless of error. For MCLUST-ME, points with measurement errors (solid) near the outer boundary (dotted) in the overlapping region tend to have high clustering uncertainties. Likewise, error-free points (empty) near the inner boundary (dashed) tend to have high uncertainties. This is consistent with our statement in Section 2.6.2.

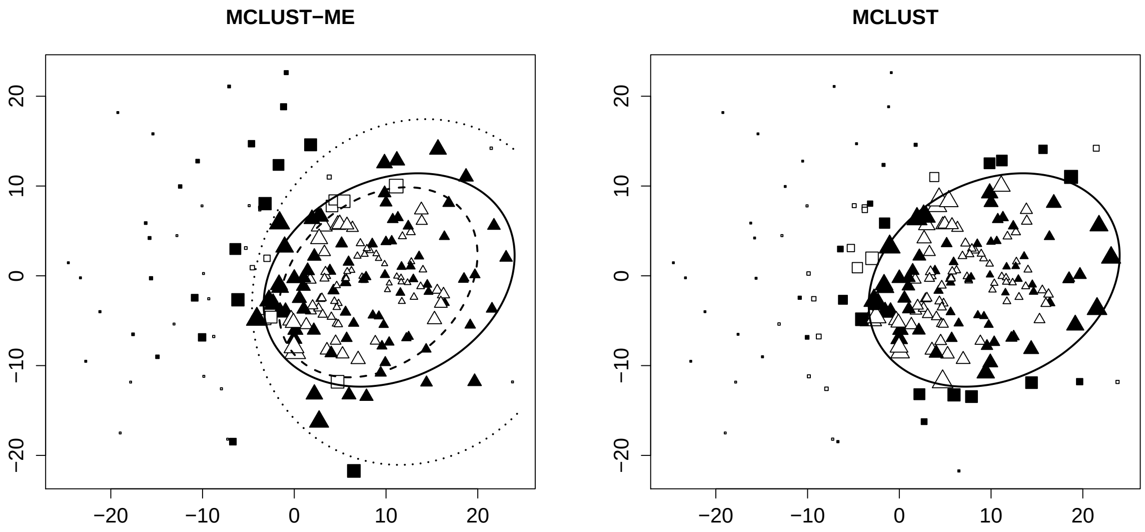


**Figure 2.** Clustering uncertainty of the sample generated with random seed = 7 and $\eta$ = 0.5. Data points of larger size have a higher clustering uncertainty. All other graph attributes are the same as Figure 1.

**Accuracy** We first evaluate the performance of MCLUST and MCLUST-ME individually using ARI (between true group labels and predicted labels) as their performance measure. Figure 3 shows that for both methods, clustering accuracy tends to decrease as error proportion $\eta$ increases. This is intuitively reasonable, because points associated with errors are more easily misclassified due to their high variability, and a larger proportion of such points means a lower overall accuracy.

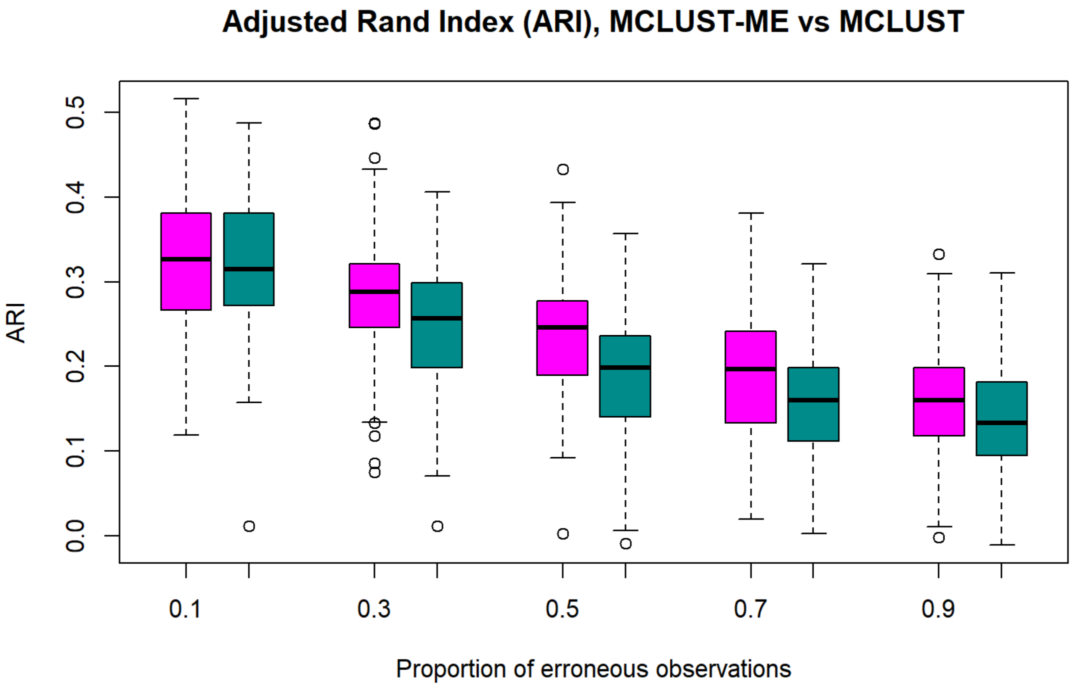


**Figure 3.** Adjusted Rand indices for MCLUST-ME and MCLUST. Five different proportions of erroneous observations ($\eta$) were considered. Magenta: MCLUST-ME; Dark Cyan: MCLUST.

Next, we compare the performances of MCLUST and MCLUST-ME by examining pairwise differences in ARI. Figure 4 shows that on average, MCLUST-ME has a slight advantage in accuracy, and it appears that this advantage is greatest when $\eta = 0.5$, and becomes smaller as $\eta$ gets closer to either zero or one. In the latter situation, error covariances will tend to become constant (all equal to $36\boldsymbol{I}_2$ as $\eta \to 1$, or $\mathbf{0}$ as $\eta \to 0$) across all points, meaning that MCLUST-ME will behave more and more like MCLUST, hence diminishing MCLUST-ME's advantage in accuracy.

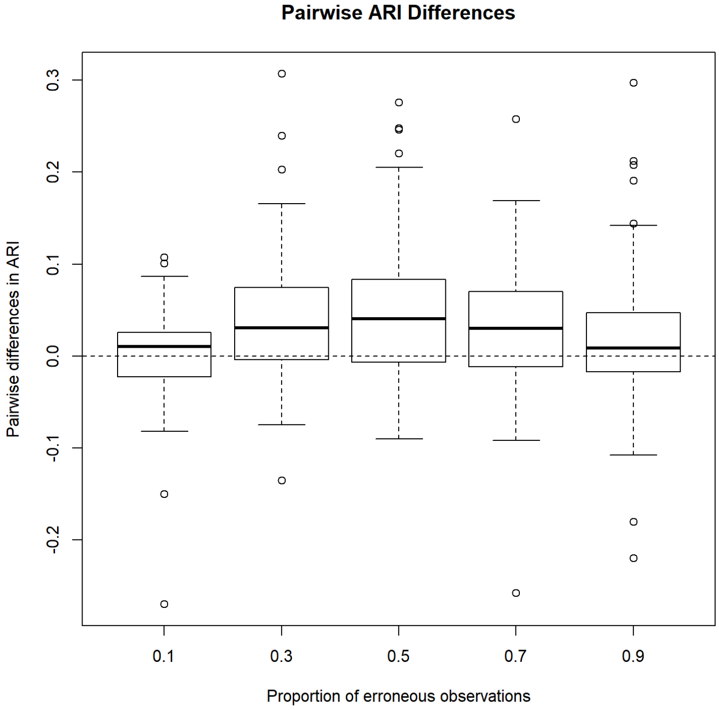


**Figure 4.** Pairwise difference in adjusted Rand indices between MCLUST-ME and MCLUST. Five different proportions of erroneous observations were considered.

Using a permutation test to test the hypotheses $H_0$ : $\text{ARI}_{MCLUST-ME} = \text{ARI}_{MCLUST}$ v.s. $H_1$ : $\text{ARI}_{MCLUST-ME} > \text{ARI}_{MCLUST}$, the $p$-values for the five cases are shown in Table 3. With the exception of $\eta = 0.1$, MCLUST-ME produced a significantly higher ARI than MCLUST.

**Table 3.** Permutation $p$-values for comparing MCLUST and MCLUST-ME ARI's.

| $\eta$ | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
|---|---|---|---|---|---|
| **$p$-value** | 0.256 | 0 | 0 | 0 | 0.002 |

Taking a closer look at the pairwise comparison when $\eta = 0.5$, Figure 5 shows that when MCLUST's accuracy is low, MCLUST-ME outperforms MCLUST most of the time, and when MCLUST's accuracy is relatively high, the two methods are less distinguishable on average.

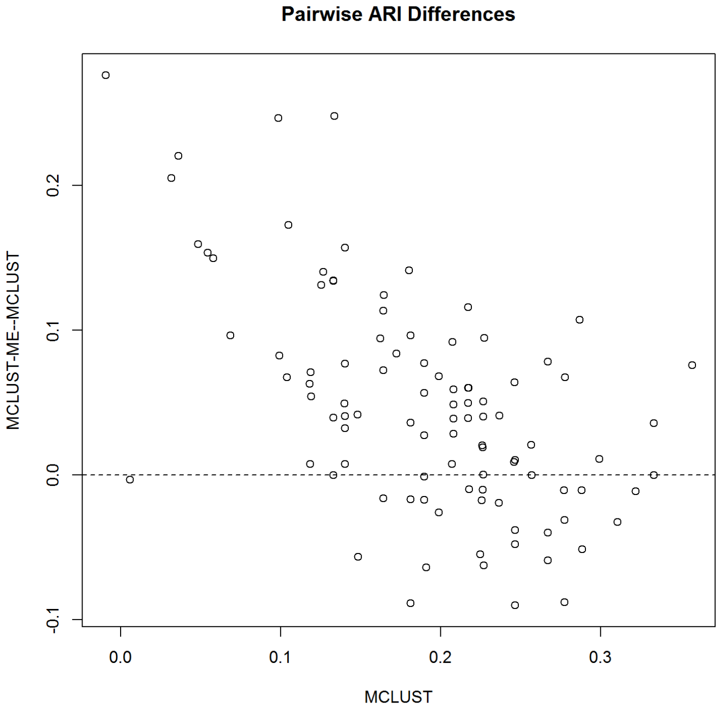


**Figure 5.** Pairwise difference in accuracy relative to MCLUST accuracy. *X-axis*: MCLUST ARI; *Y-axis*: Pairwise difference between MCLUST-ME and MCLUST ARI values.

### *3.2. Simulation 2: Clustering Uncertainties and Magnitudes of Error Covariances*

In this simulation, our focus is on investigating how clustering uncertainties differ between MCLUST-ME and MCLUST: in particular, we want to see how the magnitudes of error covariances affect the uncertainty estimates. For this purpose, we will let the magnitudes of error covariances vary in a wide range.

#### 3.2.1. Data Generation

The data were generated from a two-component bivariate normal mixture distribution with errors whose magnitudes are uniformly distributed. The data generation process is as follows.

(1) Generate $\{S_i\}_{i=1}^n$ i.i.d. from Uniform$(0, S)$, where $S_i$ denotes the magnitude of error covariance for observation $i$.

(2) Generate $\{z_i\}_{i=1}^n$ i.i.d. from Bernoulli$(\tau)$. Parameter $\tau$ will be the mixing proportion.

(3) For $i = 1, ..., n$, generate $\boldsymbol{y}_i$ from

$$z_i N_2(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1 + S_i I_2) + (1 - z_i) N_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2 + S_i I_2),$$

where $I_2$ denotes the 2-dimensional identity matrix.

The parameter values are set as follows; $\boldsymbol{\mu}_1 = (-10, 0)^T$, $\boldsymbol{\mu}_2 = (10, 0)^T$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = 100 I_2$, $n = 200$, $\tau_1 = \tau_2 = 0.5$, and $S = 100$. We chose these parameter values so that there will be quite many points near the classification boundary: these points tend to have high classification uncertainties. We want to see, under MCLUST-ME and under MCLUST, how the error magnitudes, $S_i$, will affect the estimated classification uncertainties (defined in (14)) of the points.

#### 3.2.2. Simulation Procedure

The simulation proceeds as follows.

(1) Generate a random sample following Section 3.2.1.
(2) Run MCLUST and MCLUST-ME, fixing $G = 2$. Initiate with true memberships.
(3) Record cluster membership probabilities and MLEs for model parameters upon convergence.

#### 3.2.3. Simulation 2 Results

In Figure 6, we show the clustering results from MCLUST-ME and MCLUST ($G = 2$). On this data set, the hard partitioning results do not differ much between the two methods: only two points were classified differently by the two methods (highlighted by black circles).

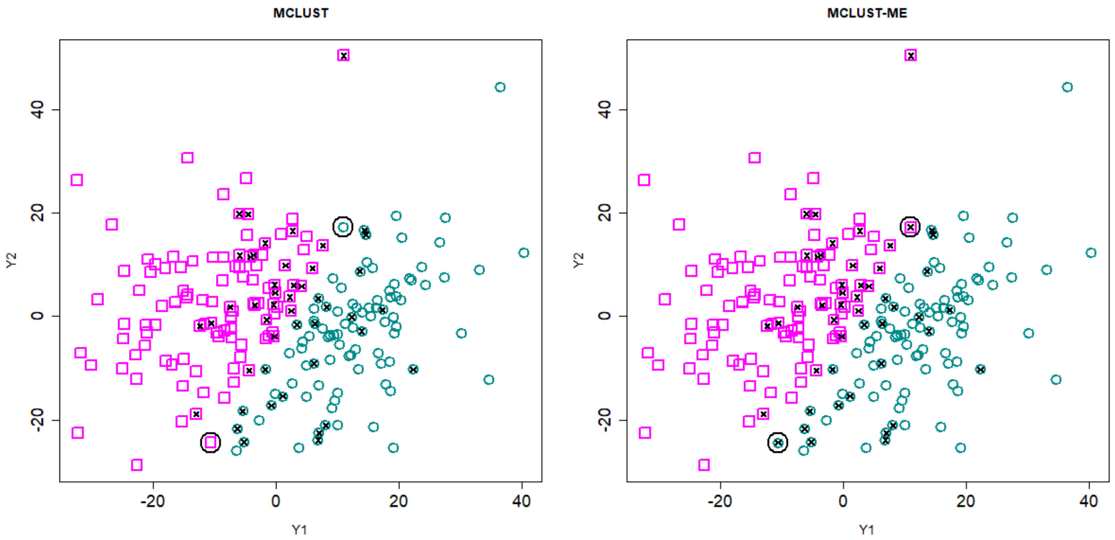


**Figure 6.** Clustering results for Simulation 2. The clustering results are indicated by different colors and symbols. Points with crosses are misclassified points. The two points that are classified differently by MCLUST-ME and MCLUST are circled in black.

Our focus here is on comparing the classification uncertainties estimated under the two methods. For MCLUST, the uncertainty measure for a point depends only on the point location and estimated centers and covariance matrices of the two clusters. Under MCLUST-ME, the uncertainty measure will also depend on the error covariance associated with the point. When two points are at the same location, MCLUST-ME will give higher uncertainty estimate to the point with greater error covariances (see Equation (12)), which is reasonable. In Figure 7, for each observation, we visualize the change in estimated membership probability to cluster 1 between MCLUST and MCLUST-ME with respect to the magnitude of its error covariance($S_i$): the closer the membership probability is to 0.5 the higher the classification uncertainty. The points with most changes in estimated membership probabilities are highlighted in Figure 8. Relative to MCLUST, the MCLUST-ME model tends to adjust the classification uncertainties upwards for points with high error covariances and downwards for points with low error covariances. In other words, relative to the MCLUST-ME results, MCLUST tends to overestimate clustering uncertainties for points with low error covariances and underestimate clustering uncertainties for points with high error covariances. This is expected, as MCLUST treats all points as measured with no errors and absorbs all individual measurement/estimation errors into the variance estimates for the two clusters. As a crude approximation, one can think that MCLUST effectively treats each point as having an error covariance matrix close to the average of all true error covariances. However, the up or down changes in membership probabilities (and thus uncertainty estimates) are not a simple function of $S_i$, and we do not see a clear-cut boundary between the ups and downs in Figure 7, as the estimates of membership probabilities are also affected by differences in estimates of centers and covariance matrices of the two clusters.

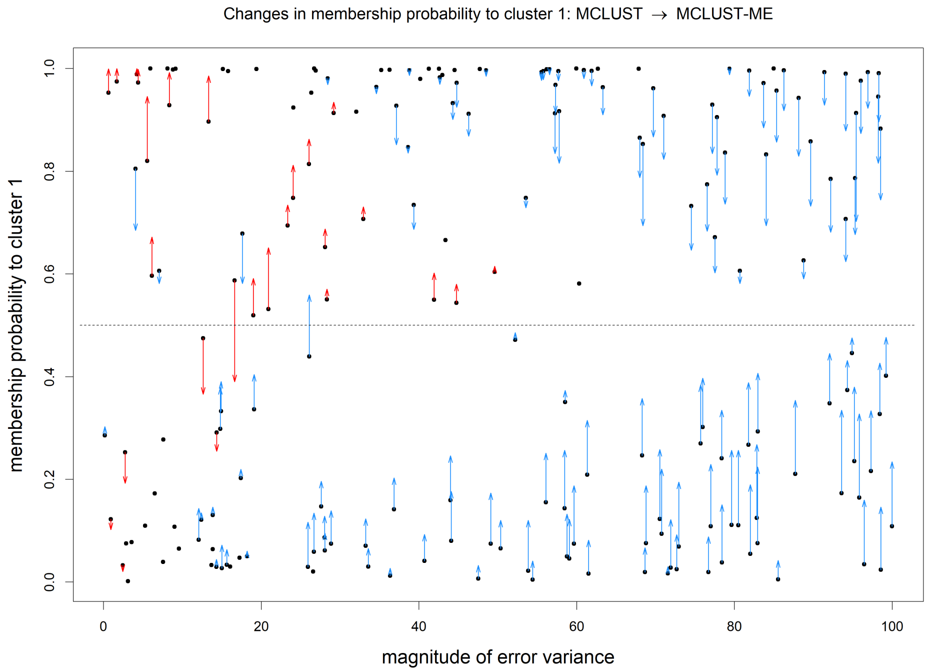


**Figure 7.** Change in estimated membership probability to cluster 1 from MCLUST to MCLUST-ME, plotted against error magnitude. *X-axis*: magnitude of error covariance, $S_i$; *Y-axis*: estimated membership probabilities to cluster 1 by MCLUST (black dots) and by MCLUST-ME (arrowheads). Changes in estimated membership probabilities from MCLUST to MCLUST-ME are highlighted by arrows (no arrow indicates a change less than 0.01). Blue and red arrows indicate an increase and decrease in estimated clustering uncertainty, respectively. With two clusters, the closer the estimated membership probability to 0.5, the higher the classification uncertainty; the classification membership changes when an arrow crosses the horizontal line at 0.5 (the dashed line).

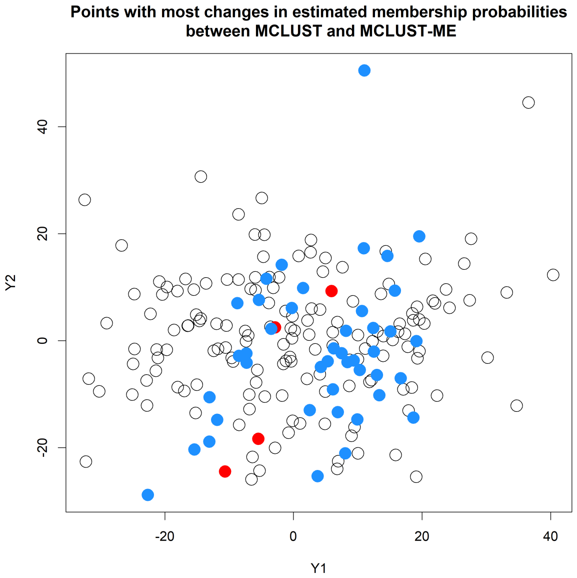


**Figure 8.** Points with most changes in estimated membership probabilities to cluster 1 from MCLUST to MCLUST-ME. The colored dots correspond to points with a change greater than 0.1 in estimated membership probability to cluster 1. Blue and red colors indicate an increase and decrease in estimated clustering uncertainty, respectively.

### *3.3. A Real Data Example*

#### 3.3.1. Data Description

The data come from an unpublished study on the model plant *Arabidopsis thaliana*. Researchers employed RNA-Seq to create a temporal profiling of *Arabidopsis* transcriptome over a *12h* period, with the aim of investigating plant innate immunity after elicitation of leaf tissue with flg22—a 22-amino-acid epitope of bacterial flagellin. A total of 33 *A. thaliana* Col-0 plants were grown in a controlled environment. Fifteen were treated with flg22, 15 with water, and the other 3 were left untreated. At each of five time points (*10 min, 1 h, 3 h, 6 h, 12 h*), three flg22-treated and three water-treated plants were harvested and prepared for RNA-Seq analysis.

A negative binomial regression model was fitted to each row (i.e., each gene) of the RNA-Seq count data. The regression model was parameterized such that the first five regression coefficients correspond to log fold changes in mean relative expression level between flg22- and water-treated groups at the five time points, which make up the temporal profile of each gene. The regression coefficients were estimated by the MLEs using the R package NBPSeq [24]. Furthermore, based on asymptotic normality of MLE, the covariance matrix of the log fold changes can be estimated by inverting the observed information matrix. For the current study, we will use the estimated regression coefficients and associated variance–covariance matrices for a subset of 1000 randomly selected genes at two of the time points (*1 h* and *3 h*) as input for the clustering analysis, as the gene expressions are most active at these two time points.

#### 3.3.2. Cluster Analysis

We applied MCLUST-ME and MCLUST to the data. Both methods have their highest BIC values when $G = 2, 3$, or 4. We focus on the $G = 2$ results as it is simple and yet illuminates the key differences between the two methods. In Figure 9, we show the clustering results from the two methods. In this example, both clustering methods show one cluster near the center and another cluster wrapping around it. This makes sense in the context of a gene expression study: the center cluster roughly represent genes that are not differentially expressed (non-DE) at these two time points; the outer cluster roughly represent genes that are differentially expressed (DE). In Figure 9, we see one signature difference between the two clustering methods: MCLUST gives a smooth boundary, whereas in the MCLUST-ME results, the two clusters are interspersed. This is expected from our theoretical analysis earlier and consistent with Simulation 1 results.

Table 4 summarizes the number of points that are classified differently by the two methods. In Figure 10, we show the standard errors (square roots of the diagonal entries of the error covariance) of the log fold changes estimated at 1 h and 3 h, with points classified differently by the two methods highlighted in colors. When we look at the points that are clustered differently by the two methods, we noticed that they tend to be the points either with very low or very high error covariances (relative to the average error covariance). This is expected as we understand that MCLUST absorbs all the individual error covariances into the estimation of the covariances of the two clusters, and thus is effectively using a middle-of-the-pack error covariance to treat each point. Therefore, we expect the differences in clustering results tend to show up among points with either very high or very low error covariances. This observation is also consistent with what we see in Simulation 2.

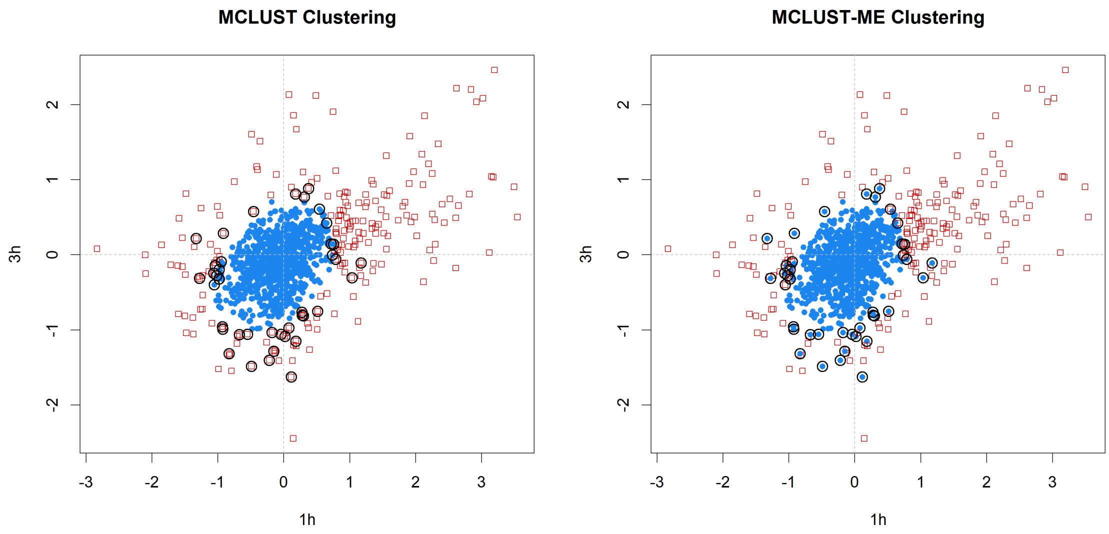


**Figure 9.** Clustering analysis of the log fold changes of 1000 genes randomly selected from the *Arabidopsis* data set. Two-group clustering of the data with MCLUST-ME and MCLUST, showing log fold changes at 1 h and 3 h. Groups are distinguished by point shapes and colors, and identified as non-DE group (blue circles) and DE group (red squares). Observations classified differently by the two methods are circled in black.

**Table 4.** Contingency table for group labels predicted by MCLUST-ME and MCLUST.

| | **MCLUST-ME** | |
|---|---|---|
| **MCLUST** | **Non-DE** | **DE** |
| Non-DE | 775 | 10 |
| DE | 30 | 185 |

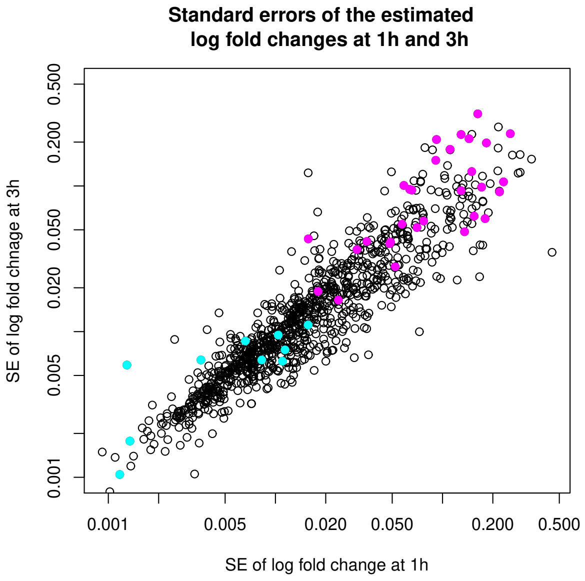


**Figure 10.** Standard errors of estimated log fold changes at 1 h and 3 h. Observations that are classified differently by MCLUST-ME and MCLUST are highlighted in colors. Magenta: classified as "DE" by MCLUST and as "non-DE" by MCLUST-ME; Cyan: classified as "non-DE" by MCLUST and as "DE" by MCLUST-ME. (Note that the axes are on the log scale.)

More interestingly, in this example, we see that the points (genes) that are classified into the “DE” cluster by MCLUST, but into the “non-DE” cluster by MCLUST-ME, tend to have high error covariances. In the MCLUST results, the clustering membership is completely determined by the magnitude of the two regression coefficients, which represent log fold changes between two experimental conditions at the two time points. In MCLUST-ME, membership calculation also considers the estimation uncertainty of the log fold changes. For gene expression data, we know that the uncertainty in log fold change estimation varies greatly (e.g., often depends on the mean expression levels). Although this example is a real data set with no ground truth on each point’s actual group membership, it seems reasonable that points with moderate log fold changes but high error variances should be classified into the non-DE cluster, as MCLUST-ME has done in our example. At the minimum, the MCLUST-ME results warn us that not all points with the same log fold changes are created equal, which is exactly the point we want to highlight in this article. Actually, this example is the data set that motivated us to consider incorporating uncertainty information into the clustering algorithm. In this example, explicitly modeling the error covariances clearly shows a difference.

The error covariance matrices were estimated, and thus associated with their own estimation errors. To get a sense of the uncertainty associated with estimating the error covariance matrices, we simulated additional sets of error covariance estimates by parametric bootstrapping: simulating copies of the RNA-seq data set based on parameters estimated from the real data set and estimating error covariance matrices from the simulated data sets. In Figure 11, we compare the square roots of the diagonal entries of two sets of simulated error covariance estimates (which correspond to the standard errors of the log fold changes at the two time points). We then tried MCLUST-ME method on the original data set with the two sets of simulated error covariance estimates: eight observations were classified differently due to the differences in error covariance estimates (see Table 5 for a summary). For a closer look, in Figure 12, we show the differences in the estimated membership probabilities (to the non-DE cluster) between the two runs of MCLUST-ME with different simulated error covariance estimates, and these differences were much less than the differences between the original MCLUST-ME and MCLUST results. These results show that the uncertainty in covariance estimation does lead to variation in the clustering results, but the variation is much less as compared to the differences between whether or not to model the estimation errors. In this sense, the MCLUST-ME method is robust to the uncertainty in covariance estimation to a certain degree.

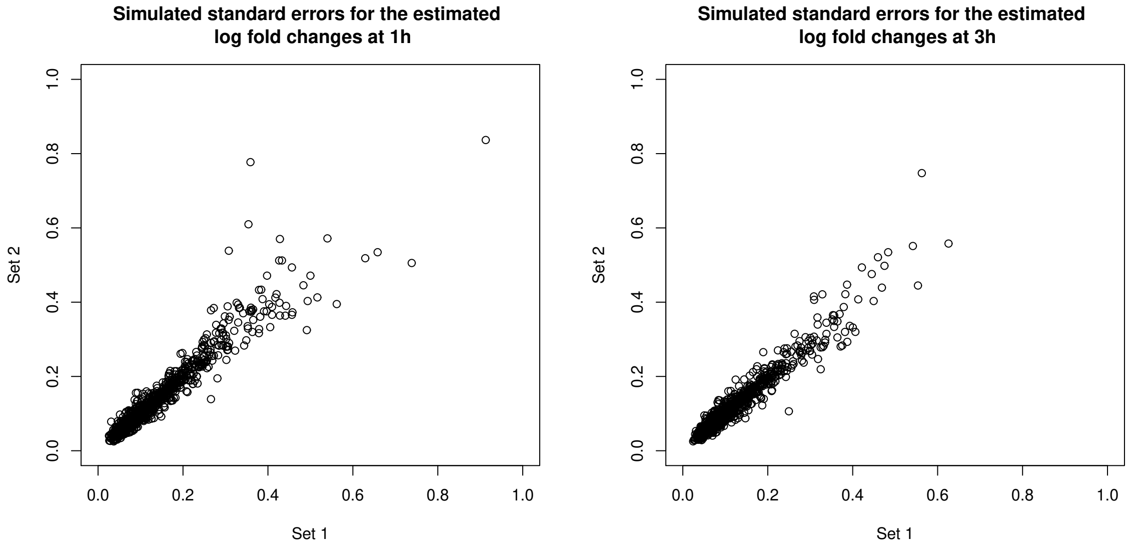
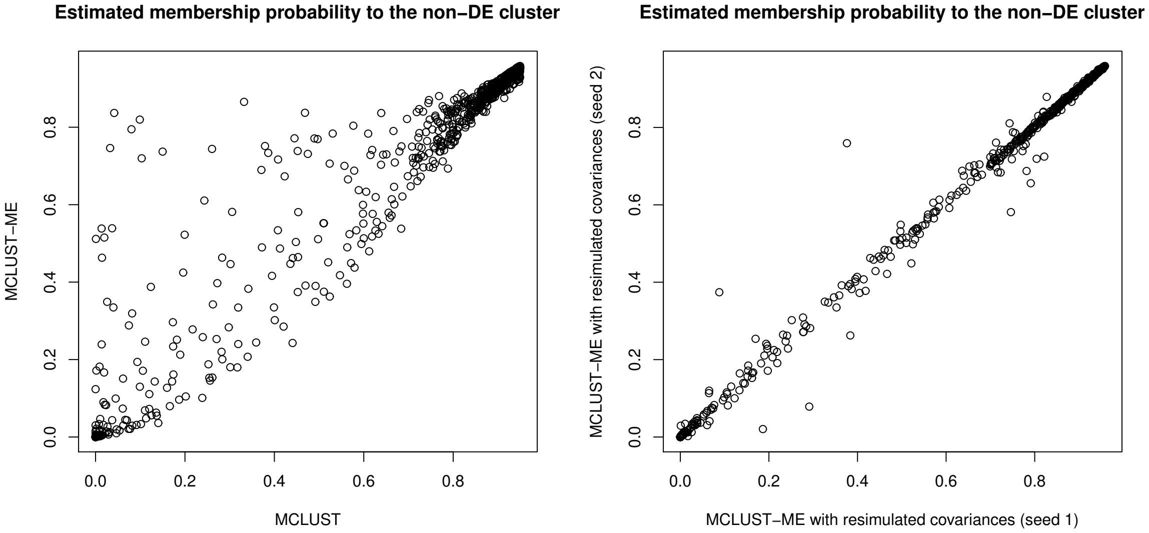


**Figure 11.** Comparing two sets of simulated standard errors for the estimated log fold changes at 1 h (**left**) and at 3 h (**right**). The standard errors correspond to the square roots of the diagonal entries of the simulated error covariance estimates.

**Table 5.** Contingency table for group labels predicted by MCLUST-ME with two sets of simulated error covariance estimates

| | **MCLUST-ME Run 1** | |
|---|---|---|
| **MCLUST-ME Run 2** | **Non-DE** | **DE** |
| Non-DE | 801 | 2 |
| DE | 6 | 191 |



(**a**) (**b**)

**Figure 12.** (**a**) Comparing membership probabilities to the "non-DE" cluster estimated by MCLUST-ME and by MCLUST. (**b**) Comparing membership probabilities to the "non-DE" cluster estimated by MCLUST-ME with two sets of simulated covariance estimates. The decision whether or not to model the error covariances will result in drastic changes in the estimated membership probabilities. In comparison, the uncertainties in covariance estimation cause much less changes in the estimated membership probabilities.

### 3.3.3. Comparison to kError

In Section 2.7, we reviewed the clustering method by the authors of [18], which models the error covariances of individual observations as in MCLUST-ME, but lacks the model-based components ($N_d(\mathbf{0}, \mathbf{\Sigma}_k)$) for modeling individual clusters. We implemented the kError algorithm according to the description in [18] and applied it to the RNA-Seq data set that we analyzed in the previous subsection, using the estimation error covariances as $\tilde{\mathbf{\Lambda}}_i$ and using the memberships predicted by MCLUST-ME as initial values. The clustering results by kError are shown in Figure 13, which can be compared with the MCLUST and MCLUST-ME results in Figure 9.

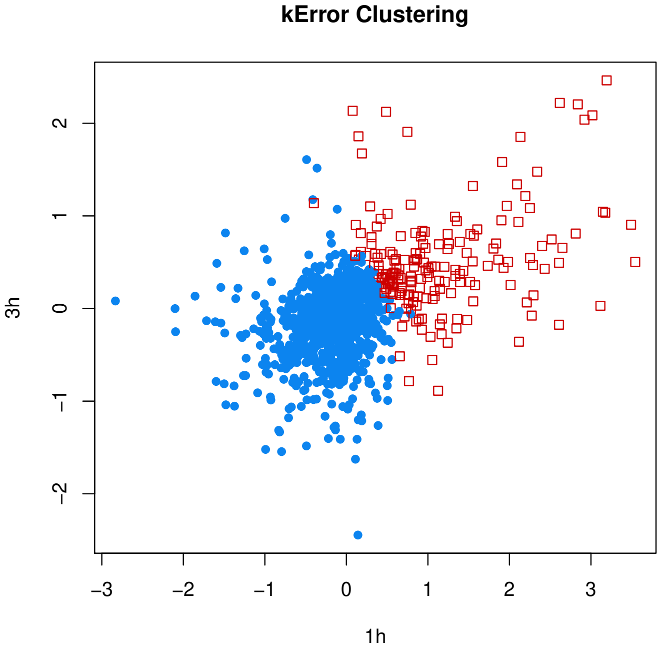


**Figure 13.** Two-grop clustering results by kError. We applied the kError method to the same RNA-Seq data set that was analyzed by MCLUST and MCLUST-ME. Compare with Figure 9.

For this data set, the two clusters estimated by MCLUST or MCLUST-ME have quite different $\boldsymbol{\Sigma}_k$ values: the covariance of the DE cluster is much greater in magnitude than that of the non-DE cluster. The DE cluster is enclosed by the non-DE cluster. Such a structure between the two clusters is difficult for kError method to capture. The way kError split the data sets into two clusters is similar to an ordinary *k*-means method. Interestingly, the two clusters by kError are interspersed without a clean-cut boundary and points with similar values but different covariances can belong to different clusters: This feature is similar to MCLUST-ME.

## 4. Conclusions and Discussion

In this paper, we proposed an extension to model-based clustering approach that accounts for known or estimated error covariances for data observed with uncertainty. The error covariances can often be estimated for data consisting of summary statistics, such as the regression coefficients from a regression analysis. We extended the EM algorithm implemented in MCLUST and implemented our new method MCLUST-ME in R [25].

A distinctive feature of MCLUST-ME is that the classification boundary separating the clusters is not always shared by all observations; instead, each distinct value of error covariance matrix corresponds to a different boundary. Using both simulated and a real data example, we have shown that under certain circumstances, explicitly accounting for estimation error distributions does lead to improved clustering results or new insights, where the degree of improvement depends on the distribution of error covariances.

It is not our intention to claim that MCLUST-ME is universally better than the original MCLUST. We are actually more interested in understanding when it will give different results than MCLUST: in other words, when it is beneficial to explicitly model the measurement error structures when performing clustering analysis. When covariances of estimation errors are roughly constant or small relative to the covariances of the clusters, MCLUST and MCLUST-ME yield highly similar results. We will tend to see meaningful differences when there is significant overlap among clusters (i.e., the difficult cases) and when there is a large variation in the magnitude of error variance.

There are a few natural extensions that can be implemented. For example, in this paper, we focused on the case where the variance–covariance matrices of the clusters are unconstrained (what MCLUST calls "VVV" type). One important feature of the original MCLUST method is that it allows structured constraints on the cluster variance–covariance matrices. Such extension is possible for MCLUST-ME. The main challenge for our current implementation of MCLUST-ME is computational. With MCLUST-ME, each point has its own error covariance matrix, and therefore we no longer have closed-form solutions for estimating the model parameters and have to rely on optimization routines. These factors make MCLUST-ME slower than the MCLUST implementation, but for reasonably-sized low-dimensional data sets, it is still manageable. The running time of the algorithm will depend on the number of clusters (G) and the size and dimension of the observed data. For our real data example, when we classify the 1000 two-dimensional data points into two clusters, it took 19 min. It took 23 h to classify the same data sets into six clusters (on a laptop workstation with an Xeon X3430 processor). To this end, improving the computation routine or exploring approximation methods is a future research topic.

The data and R code for reproducing the results in this paper is available online at https://github.com/diystat/MCLUST-ME-Genes.

**Author Contributions:** Conceptualization, Y.D.; data curation, W.Z.; formal analysis, W.Z. and Y.D.; investigation, W.Z. and Y.D.; methodology, W.Z. and Y.D.; software, W.Z. and Y.D.; supervision, Y.D.; validation, W.Z. and Y.D.; visualization, W.Z.; writing—original draft, W.Z. and Y.D.; writing—review & editing, W.Z. and Yanming Di. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number R01 GM104977.

**Acknowledgments:** The authors gratefully acknowledge Sarah Emerson, Duo Jiang, and Bin Zhuo for their valuable insight and comments, and Gitta Coaker for providing the RNA-seq experiment data. We would like to thank Joe Defilippis for IT support. We would like to thank both reviewers for their constructive comments that improved the paper.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| ARI | Adjusted Rand index |
| BIC | Bayesain information criterion |
| DE | Differentially expressed |
| EM | Expectation-maximization |
| MLE | Maximum likelihood estimate(s) |
| iid | independent and identically distributed |
| RI | Rand index |

## References

1. Fraley, C.; Raftery, A.E. Model-based clustering, discriminant analysis, and density estimation. *J. Am. Stat. Assoc.* **2002**, *97*, 611–631. [CrossRef]
2. Bouveyron, C.; Celeux, G.; Murphy, T.B.; Raftery, A.E. *Model-Based Clustering and Classification for Data Science: With Applications in R*; Cambridge University Press: Cambridge, UK, 2019; Volume 50.
3. Wolfe, J.H. Pattern clustering by multivariate mixture analysis. *Multivar. Behav. Res.* **1970**, *5*, 329–350. [CrossRef] [PubMed]
4. Fraley, C.; Raftery, A.E. Enhanced model-based clustering, density estimation, and discriminant analysis software: MCLUST. *J. Classif.* **2003**, *20*, 263–286. [CrossRef]
5. Fraley, C.; Raftery, A.E.; Murphy, T.B.; Scrucca, L. *Mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*; Tech. Rep. No. 597; Department of Statistics, University of Washington: Washington, DC, USA, 2012.

6. Scrucca, L.; Fop, M.; Murphy, T.B.; Raftery, A.E. mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *R J.* **2016**, *8*, 289. [CrossRef] [PubMed]
7. Banfield, J.D.; Raftery, A.E. Model-based Gaussian and non-Gaussian clustering. *Biometrics* **1993**, *49*, 803–821. [CrossRef]
8. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Society. Ser. B (Methodological)* **1977**, *39*, 1–38.
9. Celeux, G.; Govaert, G. Gaussian parsimonious clustering models. *Pattern Recognit.* **1995**, *28*, 781–793. [CrossRef]
10. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [CrossRef]
11. Dasgupta, A.; Raftery, A.E. Detecting features in spatial point processes with clutter via model-based clustering. *J. Am. Stat. Assoc.* **1998**, *93*, 294–302. [CrossRef]
12. Zhang, W. Model-based Clustering Methods in Exploratory Analysis of RNA-Seq Experiments. Ph.D. Thesis, Oregon State University, Corvallis, OR, USA, 2017.
13. Dwyer, P.S. Some applications of matrix derivatives in multivariate analysis. *J. Am. Stat. Assoc.* **1967**, *62*, 607–625. [CrossRef]
14. Byrd, R.H.; Lu, P.; Nocedal, J.; Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **1995**, *16*, 1190–1208. [CrossRef]
15. Murtagh, F.; Raftery, A.E. Fitting straight lines to point patterns. *Pattern Recognit.* **1984**, *17*, 479–483. [CrossRef]
16. Fraley, C. Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.* **1998**, *20*, 270-281. [CrossRef]
17. Karlis, D.; Xekalaki, E. Choosing initial values for the EM algorithm for finite mixtures. *Comput. Stat. Data Anal.* **2002**, *41*, 577–900. [CrossRef]
18. Kumar, M.; Patel, N.R. Clustering data with measurement errors. *Comput. Stat. Data Anal.* **2007**, *51*, 6084–6101. [CrossRef]
19. Tjaden, B. An approach for clustering gene expression data with error information. *BMC Bioinform.* **2006**, *7*, 17. [CrossRef] [PubMed]
20. Rand, W.M. Objective Criteria for the Evaluation of Clustering Methods. *J. Am. Stat. Assoc.* **1971**, *66*, 846–850. [CrossRef]
21. Santos, J.M.; Embrechts, M. On the use of the adjusted Rand index as a metric for evaluating supervised classification. In Proceedings of the ICANN (International Conference on Artificial Neural Networks), Limassol, Cyprus, 14–17 September 2009; Springer: Berlin, Germany, 2009; pp. 175–184.
22. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [CrossRef]
23. Qannari, E.M.; Courcoux, P.; Faye, P. Significance test of the adjusted Rand index. Application to the free sorting task. *Food Qual. Prefer.* **2014**, *32*, 93–97. [CrossRef]
24. Di, Y. Single-gene negative binomial regression models for RNA-Seq data with higher-order asymptotic inference. *Stat. Its Interface* **2015**, *8*, 405. [CrossRef] [PubMed]
25. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2019.