The Annals of Applied Statistics
2013, Vol. 7, No. 3, 1334–1361
DOI: 10.1214/13-AOAS650

BAYESIAN CLUSTERING OF REPLICATED TIME-COURSE GENE EXPRESSION DATA WITH WEAK SIGNALS

BY AUDREY QIUYAN FU[1], STEVEN RUSSELL[1], SARAH J. BRAY[1] AND SIMON TAVARÉ[1,2]

University of Cambridge, University of Cambridge, University of Cambridge, and University of Cambridge and University of Southern California

To identify novel dynamic patterns of gene expression, we develop a statistical method to cluster noisy measurements of gene expression collected from multiple replicates at multiple time points, with an unknown number of clusters. We propose a random-effects mixture model coupled with a Dirichlet-process prior for clustering. The mixture model formulation allows for probabilistic cluster assignments. The random-effects formulation allows for attributing the total variability in the data to the sources that are consistent with the experimental design, particularly when the noise level is high and the temporal dependence is not strong. The Dirichlet-process prior induces a prior distribution on partitions and helps to estimate the number of clusters (or mixture components) from the data. We further tackle two challenges associated with Dirichlet-process prior-based methods. One is efficient sampling. We develop a novel Metropolis–Hastings Markov Chain Monte Carlo (MCMC) procedure to sample the partitions. The other is efficient use of the MCMC samples in forming clusters. We propose a two-step procedure for posterior inference, which involves resampling and relabeling, to estimate the posterior allocation probability matrix. This matrix can be directly used in cluster assignments, while describing the uncertainty in clustering. We demonstrate the effectiveness of our model and sampling procedure through simulated data. Applying our method to a real data set collected from *Drosophila* adult muscle cells after five-minute Notch activation, we identify 14 clusters of different transcriptional responses among 163 differentially expressed genes, which provides novel insights into underlying transcriptional mechanisms in the Notch signaling pathway. The algorithm developed here is implemented in the R package DIRECT, available on CRAN.

1. Introduction. We are interested in the dynamics of the transcriptional response to activation of the Notch signaling pathway [Housden et al. (2013)]. During transcription, RNA molecules are produced using the DNA sequence of the genes as templates, leading to the notion of these genes being "expressed." Some

Received September 2012; revised March 2013.
[1]Supported in part by BBSRC Grant BBF00897X.
[2]Supported in part by NIH Grant P50 HG002790.
Key words and phrases. Bayesian clustering, mixture model, random effects, Dirichlet process, Chinese restaurant process, Markov-chain Monte Carlo (MCMC), label switching, multivariate analysis, time series, microarray gene expression.

of the RNA molecules, mRNA specifically, are subsequently translated into proteins, which directly regulate all kinds of biological processes. The highly conserved Notch signaling pathway mediates communication between neighbouring cells. Located on the cell surface, the Notch protein receives signals from adjacent cells and releases an intracellular protein fragment that, along with other proteins, elicits changes in gene expression in the receiving cell. Critical to the normal development of many organisms, the Notch signaling pathway is under active and extensive investigation [see Bray (2006) for a review].

Using *Drosophila* as a model system, we aim to characterise patterns of the transcriptional responses of the whole genome following a pulse of Notch activation [Housden et al. (2013)]. Previous studies have examined the changes in transcription at a single time-point following Notch pathway activation [Jennings et al. (1994), Krejci et al. (2009)]. However, it is unclear whether the regulated genes can have different temporal profiles, and whether there are particular patterns of up-regulation (increased expression) or down-regulation (decreased expression) amongst the genes whose expression changes. To generate the data we analyse here, Notch signaling was initiated in *Drosophila* adult muscle cells and stimulated for a short pulse of 5 minutes, and mRNA levels were measured in these treated cells relative to untreated cells, using microarrays for 4 biological replicates at 18 unevenly-spaced time points during the 150 minutes after activation [Housden (2011), Housden et al. (2013); also see Section 5 for details on the experiment and preprocessing of the data]. We aim to address the following questions for the 163 differentially expressed genes: (i) how many different expression patterns are there and what are these patterns? and (ii) which genes exhibit what expression pattern? These questions naturally call for a clustering approach to analyse these data.

However, there are several challenges associated with this data set. First, these data are different from the conventional time series data. Time series often refer to the measurements of a single subject over time. In the microarray experiment, a biological replicate refers to a *population* of cells, and the expression levels at any time point are measured for a *distinct sample* of cells from the starting population. Although the cells from the same biological replicate are typically assumed to be homogeneous, the heterogeneity among cells is nonnegligible and contributes to the noise in the data [Spudich and Koshland (1976), McAdams and Arkin (1997), Elowitz et al. (2002)]. Second, since only a short pulse of Notch activation was applied, the level of (relative) expression, measured as $\log_2$-transformed fold change, in our data is often not much different from 0 (Figure 1). Specifically, the mean expression level across time points and across replicates is only 0.1 with a standard deviation of 0.5, leading to a signal-to-noise ratio of only $\sim$0.2. Meanwhile, the median of the lag 1 autocorrelation across replicates is only 0.4 (interquartile range: 0.2–0.6), indicating that the temporal dependence is weak. Third, existing clustering software programs such as MCLUST [Fraley and Raftery (2002), Fraley and Raftery (2006)] and SplineCluster [Heard, Holmes and Stephens (2006)] give vastly different results (see Section 5 for detail).

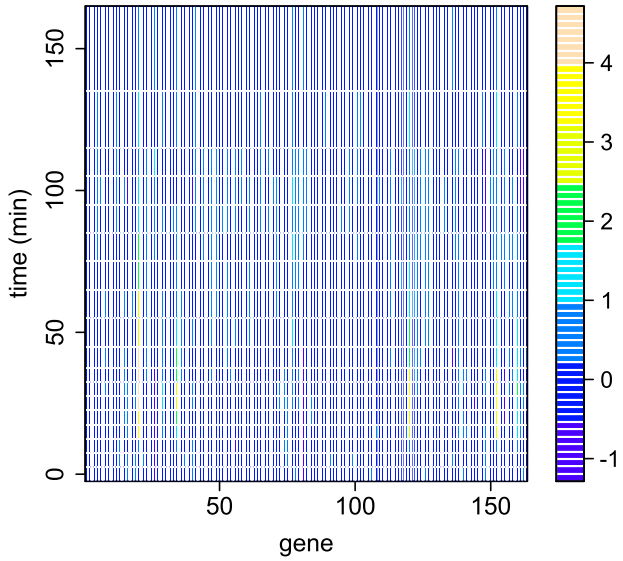

FIG. 1. *Mean profiles of* 163 *significantly expressed genes* [*false discovery rate* 10% *by EDGE*; *Storey et al.* (2005)] *over the time course of* 18 *time points. Each value is the mean* (*taken over the four replicates*) *of* $\log_2$ *fold change in treated cells relative to untreated cells.*

These scientific questions and the challenges in the data thus motivated the clustering method we develop here. Our clustering method consists mainly of a random-effects mixture model coupled with a Dirichlet-process prior. We propose the random-effects model to tackle the high level of noise in the data that arises from several sources. Under the random-effects model, we make use of the full data, rather than reducing the data to the means across replicates, which may not be accurate with this level of noise. Under this model, we also do not make many assumptions about the underlying biological process, which is still largely unknown. Novel patterns detected this way are unlikely to be the result of potentially inappropriate assumptions. The use of a Dirichlet-process prior enables us to estimate the number of clusters directly from the data. Below we review existing relevant work, which laid the foundation for our method.

Most clustering methods that are shown to be effective on time-course data are model-based, with the distribution following a mixture of multivariate Gaussian components [Fraley and Raftery (2002), Medvedovic and Sivaganesan (2002), Medvedovic, Yeung and Burngarner (2004), Celeux, Martin and Lavergne (2005), Beal and Krishnamurthy (2006), Fraley and Raftery (2006), Heard, Holmes and Stephens (2006), Ma et al. (2006), Qin (2006), Zhou and Wakefield (2006), Lau and Green (2007), Booth, Casella and Hobert (2008), Rasmussen et al. (2009), McNicholas and Murphy (2010), Green (2010), Cooke et al. (2011)]. Different methods take different approaches to modeling the mean vectors and covariance structures. Several methods attempt to account specifically for the tempo-

ral dependence by modeling the (prior) mean vector in terms of spline functions [Heard, Holmes and Stephens (2006), Ma et al. (2006)] or as a random walk [Zhou and Wakefield (2006)]. As for the covariance structure, some methods [Medvedovic and Sivaganesan (2002), Medvedovic, Yeung and Burngarner (2004), Heard, Holmes and Stephens (2006), Qin (2006), Lau and Green (2007), Green (2010)] assume independence across items and across time points a priori. Both Fraley and Raftery (2006) and McNicholas and Murphy (2010) take a matrix decomposition approach and consider various models for the covariance matrix by constraining no or some decomposed terms to be identical across clusters. However, whereas Fraley and Raftery (2006) apply eigenvalue decomposition, which is applicable also to data types other than time-course data, McNicholas and Murphy (2010) use a modified Cholesky decomposition, which has connections with autoregressive models and is thus specifically designed for time-course data. Another common approach to modeling the covariance structure is random-effects models, which account for variability arising from different sources [Celeux, Martin and Lavergne (2005), Ma et al. (2006), Booth, Casella and Hobert (2008)]. We take this approach in our clustering method. Indeed, with a random-effects mixture model, we demonstrate that specific modeling of the temporal structure may not be essential for clustering replicated time-course data.

Estimating the number of clusters, or mixture components, under a model-based framework, has been a difficult problem. Several approaches exist, largely falling into two categories: optimization for a single "best" partition and a fully Bayesian approach that weights the partitions by their probabilities given the data. In the optimization category, the penalised likelihood approach, using criteria such as the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) and so on, has been used by Fraley and Raftery (2002), Celeux, Martin and Lavergne (2005), Schliep et al. (2005), Ma et al. (2006) and McNicholas and Murphy (2010). Heard, Holmes and Stephens (2006) in their program SplineCluster and Booth, Casella and Hobert (2008) maximise the posterior probability of partitions given the data. Arguing that the maximal posterior probability of partitions may be difficult to compute reliably and may not be representative, Lau and Green (2007) suggest maximizing posterior loss, an approach followed in Green (2010). However, the main drawback with the optimization approach is that competing partitions with similar (penalised) likelihoods are simply ignored. Methods based on optimization may also suffer from numeric instability, as our experience with MCLUST [Fraley and Raftery (2002)] suggests (explained in Section 5). When clustering is used as an exploratory data analysis tool to understand the heterogeneity in the data, it is often desirable and realistic to explore more than one partition and to understand how and why the data support multiple competing partitions. We therefore find the fully Bayesian approach more appealing with this rationale. In this category, Zhou and Wakefield (2006) implemented the Birth-Death Markov Chain Monte Carlo (BDMCMC) scheme initially developed by Stephens (2000a),

which designs a birth-death process to generate new components and eliminate existing ones. Medvedovic and Sivaganesan (2002), Medvedovic, Yeung and Burngarner (2004), Beal and Krishnamurthy (2006), Qin (2006), Booth, Casella and Hobert (2008) and Rasmussen et al. (2009) developed Markov Chain Monte Carlo (MCMC) schemes under a Dirichlet-process prior. The Dirichlet-process prior, a popular tool in nonparametric Bayesian statistics, can induce sparse partitions among items [Ferguson (1973), Antoniak (1974)] and has been widely used in analyses such as nonparametric density estimation [Escobar and West (1995), Fox (2009)]. Here, we take the fully Bayesian approach and use a Dirichlet-process prior to induce a prior distribution on partitions, which helps us to explore different numbers of clusters and to sample from partitions under each number. The clustering obtained from the Bayesian approach is essentially an average of all possible solutions weighted by their posterior probabilities.

However, two major challenges remain in the clustering methods under the Dirichlet-process priors. One is efficient sampling. Many MCMC methods have been developed under Dirichlet-process priors for conjugate priors of the parameters [such as those reviewed in Neal (2000)], restricting the choices of priors. Alternative sampling methods have been developed, such as Gibbs samplers designed for nonconjugate priors [MacEachern and Müller (1998)], several Metropolis–Hastings (MH) samplers under the Chinese-restaurant representation [Neal (2000)], split-merge sampling [Jain and Neal (2004), Jain and Neal (2007)], another two-stage MH procedure under an implicit Dirichlet-process prior [Booth, Casella and Hobert (2008)], retrospective sampling [Papaspiliopoulos and Roberts (2008)] and slice sampling [Walker (2007), Kalli, Griffin and Walker (2011)], both of which are developed under the stick-breaking process representation. Several of these and related methods are reviewed recently in Griffin and Holmes (2010). Here, we develop a novel MH sampler under the Chinese-restaurant representation. Our MH sampler does not introduce additional (auxiliary or latent) variables or tuning parameters. It also does not require separate split and merge steps, but rather allows for dimension changes in a single step. In addition, it is based on standard MH calculations and is therefore straightforward to understand and easy to implement.

The other major challenge is posterior inference. Existing approaches [Medvedovic and Sivaganesan (2002), Medvedovic, Yeung and Burngarner (2004), Beal and Krishnamurthy (2006), Rasmussen et al. (2009), Dhavala et al. (2010)] attempt to make use of the posterior "similarity" matrix, whose entries are the posterior probability of allocating two items to the same cluster, by applying linkage-based clustering algorithms to this matrix. Focusing on this matrix in effect converts the original clustering problem into another one, while discarding other valuable information in the MCMC samples. We propose a two-step posterior inference procedure that involves resampling and relabeling to estimate the posterior allocation probability matrix, which may be used more directly in forming clusters and other inference.

In this paper, we present our method DIRECT, the Dirichlet process-based random-effects model as a clustering tool. We describe the random-effects mixture model in Section 2 and the Bayesian inference in Section 3, which includes a novel MH MCMC algorithm for sampling partitions under the Dirichlet-process prior, as well as the two-step posterior inference procedure. We examine the performance of our method through simulation studies in Section 4. We apply our method to the time-course microarray gene expression from the Notch experiment in Section 5. Compared with SplineCluster [Heard, Holmes and Stephens (2006)] and MCLUST [Fraley and Raftery (2002), Fraley and Raftery (2006)], our method appears to be more accurate and sensitive to subtle differences in different clusters, in both simulation studies and the real application. In addition, the analysis of the real data reveals several novel insights into the transcriptional dynamics after the pulse of Notch activation. We summarise and discuss the features of our method in Section 6.

2. Random-effects mixture model. Consider N genes measured at J time points in each of the R replicates. Let M_{ijr}, $i = 1, \ldots, N$, $j = 1, \ldots, J$, $r = 1, \ldots, R$, be the measurement for the ith gene at the jth time point from the rth replicate. The J time points may be unevenly distributed. We assume that there are no missing data. We use a random-effects mixture model to describe the heterogeneity in replicated time-course data, and explain the details of the model below.

Following the standard mixture model formulation with a known number of mixture components, K, we assume that data vectors

$$\mathbf{M}_i = (M_{i11}, \ldots, M_{i1R}, \ldots, M_{iJ1}, \ldots, M_{iJR})^{\mathrm{T}}$$

are independent and identically distributed realizations drawn from a mixture distribution with K components and a set of mixing proportions w_k, $k = 1, \ldots, K$. The superscript T represents "transpose" and ensures that $\mathbf{M}_i$ is a column vector. The probability density function of $\mathbf{M}_i$, denoted by f, can be written as a weighted average:

$$f(\mathbf{M}_i|\boldsymbol{\Theta}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} w_k g_k(\mathbf{M}_i|\boldsymbol{\Theta}^k, \boldsymbol{\Sigma}^k),$$

where g_k is the probability density function of the kth mixture component, and $\boldsymbol{\Theta} = (\boldsymbol{\Theta}^1, \ldots, \boldsymbol{\Theta}^K)$ and $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}^1, \ldots, \boldsymbol{\Sigma}^K)$ are parameters of the mixture distribution, with component-wise mean vector $\boldsymbol{\Theta}^k$ and covariance matrix $\boldsymbol{\Sigma}^k$, $k = 1, \ldots, K$. Whereas it is possible to define a cluster by more than one mixture component, for presentation purposes we consider here the case where one mixture component defines a cluster and use "mixture component" and "cluster" interchangeably. Let Z_i denote the cluster membership for the ith gene. Then,

$$\Pr(Z_i = k|\mathbf{w}, \boldsymbol{\Theta}, \boldsymbol{\Sigma}) = w_k,$$

where $\mathbf{w}$ is the set of mixing proportions. Following the notation in Stephens (2000a) and denoting the data by $\mathbf{M} = (\mathbf{M}_1, \ldots, \mathbf{M}_N)$, we define the posterior allocation probabilities as $\Pr(Z_i = k|\mathbf{M})$, $i = 1, \ldots, N$ and $k = 1, \ldots, K$, which form the posterior allocation probability matrix $\mathbf{P}$ of dimension $N \times K$. We aim to estimate $\mathbf{P}$ as part of the inference and to form clusters based on the estimated $\mathbf{P}$, using, for instance, the most likely allocation.

Inspired by variance components approaches [Searle, Casella and McCulloch (2006)] and random-effects models frequently used in longitudinal studies [Dunson (2010)], we constrain the covariance matrix of each mixture component, $\mathbf{\Sigma}^k$, by attributing the total variability to three sources: clustering, sampling across multiple time points (or more broadly speaking, multiple experimental conditions), and sampling a limited number of replicates. Whereas the first source of variability is due to "grouping" of the genes, the latter two are defined by the design of the time-course experiment. If the ith gene is sampled from the kth mixture component (i.e., $Z_i = k$), the random-effects model can be written as follows:

$$M_{ijr}|\{Z_i = k\} = \Theta_j^k + \phi_i^k + \tau_{ij}^k + \varepsilon_{ijr}^k, \tag{2.1}$$

where

$$\begin{aligned}
\mathrm{E}\big(M_{ijr}|\{Z_i = k\}\big) &= \Theta_j^k, \\
\phi_i^k|\big\{Z_i = k, \lambda_\phi^k\big\} &\sim_{\text{i.i.d.}} \mathrm{N}\big(0, \lambda_\phi^k\big), \\
\tau_{ij}^k|\big\{Z_i = k, \lambda_\tau^k\big\} &\sim_{\text{i.i.d.}} \mathrm{N}\big(0, \lambda_\tau^k\big), \\
\varepsilon_{ij}^k|\big\{Z_i = k, \lambda_\varepsilon^k\big\} &\sim_{\text{i.i.d.}} \mathrm{N}\big(0, \lambda_\varepsilon^k\big).
\end{aligned}$$

In this formulation, Θ_j^k represents the "true" value (fixed effect) at the jth time point, ϕ_i^k the within-cluster random effect, τ_{ij}^k the cross-experimental-condition random effect and ε_{ijr}^k the replicate random effect. Here, the experimental conditions are time points. We assume that random effects ϕ_i^k, τ_{ij}^k and ε_{ijr}^k are independent across clusters and of each other. Each of the three random effects has a corresponding variability term: λ_ϕ^k is the within-cluster variability, λ_τ^k the cross-experimental-condition variability, and λ_ε^k the residual variability. The three types of variability are all component specific.

Given cluster membership $Z_i = k$, replicated measurements of the ith gene, $\mathbf{M}_i$, follow a multivariate normal distribution:

$$\mathbf{M}_i|\big\{Z_i = k, \mathbf{\Theta}^k, \lambda_\phi^k, \lambda_\tau^k, \lambda_\varepsilon^k\big\} \sim_{\text{ind}} \mathrm{N}_{JR}\big(\mathbf{\Theta}_{\text{agg}}^k, \mathbf{\Sigma}_{\text{agg}}^k\big),$$

which has aggregated mean vector $\mathbf{\Theta}_{\text{agg}}^k = ({\mathbf{\Theta}^k}^{\mathrm{T}}, \ldots, {\mathbf{\Theta}^k}^{\mathrm{T}})^{\mathrm{T}}$, where $\mathbf{\Theta}^k$ repeats R times, and aggregated covariance matrix $\mathbf{\Sigma}_{\text{agg}}^k$ whose entry is

$$\mathrm{Cov}(M_{ijr}, M_{ij'r'}) = \lambda_\phi^k + \lambda_\tau^k \mathbf{1}\big(j = j'\big) + \lambda_\varepsilon^k \mathbf{1}\big(j = j', r = r'\big), \tag{2.2}$$

where $\mathbf{1}(A)$ is the indicator function that takes value 1 if condition A is satisfied and 0 otherwise. In addition, $\mathbf{M}_i$ and $\mathbf{M}_j$, where $i \neq j$, are independent of each other.

Parameters of interest under this random-effects mixture model include the number of mixture components, K, component-specific parameters $\boldsymbol{\Theta}^k$, λ_ϕ^k, λ_τ^k and λ_ε^k, where $k = 1, \ldots, K$, and posterior allocation probability matrix $\mathbf{P}$ of dimension $N \times K$.

3. Bayesian inference.

3.1. *The Dirichlet-process prior.* As mentioned in Section 1, Dirichlet processes help to partition the parameter space without prior knowledge of the number of partitions, K, and thus provide a coherent framework for directly estimating K from data and for sampling in a parameter space of variable dimensions. Denote the parameter of interest for *each* gene by $\boldsymbol{\gamma}_i$, which, in our case, may include a mean vector $\boldsymbol{\Theta}_i$ and three terms of variability, namely, λ_{ϕ_i}, $\lambda_{\tau i}$ and $\lambda_{\varepsilon i}$, such that

$$\begin{aligned} \boldsymbol{\gamma}_i &= \{\boldsymbol{\Theta}_i, \lambda_{\phi_i}, \lambda_{\tau i}, \lambda_{\varepsilon i}\}, \\ \mathbf{M}_i &\sim F(\boldsymbol{\gamma}_i), \qquad i = 1, \ldots, N, \end{aligned}$$

where F represents a distribution, which is a multivariate normal distribution in our case. We assume that $\boldsymbol{\gamma}_i$s follow a random distribution G, which is in turn a random draw from a (compound) Dirichlet process, denoted as follows:

$$\boldsymbol{\gamma}_i \sim G, \tag{3.1}$$

$$G \sim \mathrm{DP}(\alpha, G_0), \qquad \alpha \geq 0, \tag{3.2}$$

with base distribution G_0 (continuous in our case), which describes how values in the space are generated, and concentration parameter α, which is nonnegative. Note that $\boldsymbol{\gamma}_i$s are identically distributed, but not necessarily independent. The dependence among them under the Dirichlet process specifically refers to their values being clustered, that is, some $\boldsymbol{\gamma}_i$s may take on identical value.

Indeed, the Dirichlet process describes a mechanism by which clustered parameters $\boldsymbol{\gamma}_i$ may be simulated. We can generate a realization for one of them, say, $\boldsymbol{\gamma}_1$, from G_0. The value of $\boldsymbol{\gamma}_2$ may be identical to $\boldsymbol{\gamma}_1$, with probability $1/(1+\alpha)$, or an independent realization also from G_0 and different from $\boldsymbol{\gamma}_1$, with probability $\alpha/(1+\alpha)$. Generally, having generated n realizations, the value of the $n+1$st realization follows the following distribution [Antoniak (1974)]:

$$\begin{aligned} &\Pr(\boldsymbol{\gamma}_{n+1} = \boldsymbol{\gamma} | \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_n, \alpha) \\ &\quad = \begin{cases} \dfrac{\sum_{i=1}^n \mathbf{1}(\boldsymbol{\gamma}_i = \boldsymbol{\gamma})}{n+\alpha}, & \boldsymbol{\gamma} \in \{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_n\}, \\ \dfrac{\alpha}{n+\alpha}, & \boldsymbol{\gamma} \notin \{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_n\}. \end{cases} \end{aligned} \tag{3.3}$$

In other words, the probability of $\boldsymbol{\gamma}_{n+1}$ being identical to one of the existing values is proportional to the number of times this value has already shown up. This sampling process is also known as the Chinese restaurant process [reviewed in Pitman (2006)], a useful representation for Neal (2000) to derive the Metropolis–Hastings sampling procedures, which are explained in the next section.

The sampling distribution above induces a distribution on the partition of the N values, $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_N$, with a random number of partitions, K. Specifically, the partition distribution is with respect to the cluster memberships Z_i, $i = 1, \ldots, N$, as well as K [Antoniak (1974)]:

$$\Pr(Z_1, \ldots, Z_N, K | \alpha > 0) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \alpha^K \prod_{l=1}^{K} (N_l - 1)!, \tag{3.4}$$

where N_l is the size of the lth cluster, and

$$\Pr(Z_1 = \cdots = Z_N, K = 1 | \alpha = 0) = 1. \tag{3.5}$$

We use this distribution as the prior in our Bayesian inference.

As a measure of "concentration," very small α leads to a small probability of taking on a new value in the Dirichlet process, as equation (3.3) suggests, and hence to the probability mass being concentrated on a few distinct values, as equations (3.4) and (3.5) suggest. As $\alpha \to 0$, $\boldsymbol{\gamma}_i$s are identical, which corresponds to a single draw from the base distribution G_0. On the other hand, large α leads to a large probability of taking on new values in the Dirichlet process of equation (3.3) and an appreciable probability for having a range of distinct values in equation (3.4). As $\alpha \to \infty$, $\boldsymbol{\gamma}_i$s are all different and form an independent and identically distributed sample from G_0. Therefore, α effectively controls the sparsity of partitioning (or clustering).

We note that equations (3.3)–(3.5) characterise the canonical Dirichlet process with parameter α, denoted DP(α), for an arbitrary space, as Antoniak (1974) defined it. The representation in expression (3.2), which we consider a compound Dirichlet process, includes the additional information on how the elements of the space arise: they are realizations of the base distribution G_0.

3.2. *A Metropolis–Hastings sampler for cluster memberships.* The key step in the MCMC algorithm is sampling partitions, specifically, cluster memberships Z_i, under the Dirichlet-process prior. We develop a Metropolis–Hastings sampler that allows nonconjugate priors for parameters and efficient mixing.

Similar to Neal (2000), we design the MH procedure to sample each Z_i during an MCMC update. Let the current value of Z_i be z', which, together with all the other Z_j, gives the current number of clusters as $K = k'$. We propose a new value z^* for Z_i, which gives rise to the proposed value k^* for K. Let $\boldsymbol{\xi}$ be the parameter vector of interest for the random-effects mixture model under the Dirichlet-process prior, such that

$$\boldsymbol{\xi} = \{K, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K, \lambda_\phi^1, \ldots, \lambda_\phi^K, \lambda_\tau^1, \ldots, \lambda_\tau^K, \lambda_\varepsilon^1, \ldots, \lambda_\varepsilon^K, Z_1, \ldots, Z_N, \alpha\}.$$

We accept the proposal with probability $\min(1, H)$, where H is the Hastings ratio computed as follows:

$$
\begin{aligned}
H &= \frac{\pi(Z_i = z^*)}{\pi(Z_i = z')} \frac{g(Z_i = z' | Z_i = z^*)}{g(Z_i = z^* | Z_i = z')} \\
&= \frac{\Pr(\mathbf{M}_i | Z_i = z^*, \cdot) \Pr(z_1, \ldots, z^*, \ldots, z_N, k^* | \alpha)}{\Pr(\mathbf{M}_i | Z_i = z', \cdot) \Pr(z_1, \ldots, z', \ldots, z_N, k' | \alpha)} \frac{g(z' | z^*)}{g(z^* | z')} \\
&= \frac{\Pr(\mathbf{M}_i | z^*, \cdot)}{\Pr(\mathbf{M}_i | z', \cdot)} \frac{\Pr(z^*, k^* | \mathbf{z}_{-i}, \alpha)}{\Pr(z', k' | \mathbf{z}_{-i}, \alpha)} \frac{g(z' | z^*)}{g(z^* | z')},
\end{aligned} \tag{3.6}
$$

where $\cdot$ refers to current estimates of parameters in $\boldsymbol{\xi}$ other than Z_i and $\mathbf{z}_{-i}$ denotes the cluster memberships of all genes except for the ith one, which do not change when we update Z_i.

Under the Dirichlet-process prior, we can compute the conditional probability $\Pr(z', k' | \mathbf{z}_{-i}, \alpha)$ as in Proposition 1:

PROPOSITION 1. *Consider N values drawn from a Dirichlet process with concentration parameter $\alpha \geq 0$. These values can be partitioned into K clusters, where K is a random variable, with Z_i, $i = 1, \ldots, N$, indicating the cluster membership. Then the following conditional probability holds*:

$$
\begin{aligned}
&\Pr(Z_i = z, K = k | \mathbf{Z}_{-i} = \mathbf{z}_{-i}, \alpha) \\
&\quad = \begin{cases} \dfrac{N_z - 1}{N - 1 + \alpha}, & Z_i \text{ is not in a singleton cluster}, \\ \dfrac{\alpha}{N - 1 + \alpha}, & Z_i \text{ is in a singleton cluster}, \end{cases}
\end{aligned} \tag{3.7}
$$

where $\mathbf{Z}_{-i}$ with value $\mathbf{z}_{-i}$ denotes the cluster memberships, excluding the ith gene, and N_z is the size of the zth cluster.

PROOF. See the Appendix. □

Neal (2000) then proposed an MH procedure, using the conditional probability in equation (3.7) as the proposal distribution g, which led to a simplified Hastings ratio:

$$
H = \frac{\Pr(\mathbf{M}_i | z^*, \cdot)}{\Pr(\mathbf{M}_i | z', \cdot)}. \tag{3.8}
$$

The main problem with this MH sampler is slow mixing: because the probability of a move is proportional to the size of the cluster, the Markov chain can be easily stuck, especially when there exist one or a few large clusters. For example, consider $N = 200$ and current clusters 1–3 of size 185, 10 and 5, respectively. A gene currently allocated to cluster 1 may be much more similar to cluster 3,

TABLE 1

Hastings ratio for four cases under the proposed Metropolis–Hastings sampler for cluster membership Z_i with current value z' and proposed value z^*. k^* and k' are the number of clusters after and before the proposed move, respectively

	Current cluster a singleton	**Proposal an existing label**	$k^* - k'$	**Hastings ratio**
1	Yes	Yes	-1	$\frac{\Pr(\mathbf{M}_i \mid z^*,\cdot)}{\Pr(\mathbf{M}_i \mid z',\cdot)} \frac{N_{z^*}}{\alpha} \frac{k'}{k'-1}$
2	Yes	No	0	$\frac{\Pr(\mathbf{M}_i \mid z^*,\cdot)}{\Pr(\mathbf{M}_i \mid z',\cdot)}$
3	No	Yes	0	$\frac{\Pr(\mathbf{M}_i \mid z^*,\cdot)}{\Pr(\mathbf{M}_i \mid z',\cdot)} \frac{N_{z^*}}{N_{z'}-1}$
4	No	No	1	$\frac{\Pr(\mathbf{M}_i \mid z^*,\cdot)}{\Pr(\mathbf{M}_i \mid z',\cdot)} \frac{\alpha}{N_{z'}-1} \frac{k'}{k'+1}$

implying a high likelihood ratio as in the simplified Hastings ratio (3.8). However, the probability of proposing such a favorable move from cluster 1 to cluster 3 is only $5/(199+\alpha)$, where α is usually small to induce a parsimonious partition. The probability of moving a gene to a previously nonexistent cluster is $\alpha/(199+\alpha)$, which can be even smaller.

We develop a novel MH MCMC strategy to deal with poor mixing of Neal's MH sampler. Our proposal distribution for a cluster membership is discrete uniform on the integer set from 1 to $k'+1$, excluding the current cluster the gene belongs to, where k' is the number of existing clusters. This proposal distribution forces the proposed cluster membership always to be different from the current one, and makes the Markov chain move to a new or small cluster more easily. Whether to accept the proposal or not depends on the Hastings ratio, which needs to be recalculated as in Proposition 2.

PROPOSITION 2. *For cluster membership Z_i with current value z', if proposal z^* is generated from a discrete uniform distribution over the integer set $\{1,\ldots,z'-1,z'+1,\ldots,k'+1\}$, where k' is the current number of clusters, then the Hastings ratio takes on values as listed in Table 1, where four cases, including a generation of a new cluster and elimination of an existing cluster, are considered.*

PROOF. The proof of this proposition can be found in Section 1 of the supplemental material [Fu et al. (2013)]. □

3.3. *Other prior distributions.* The base distribution G_0 specifies the prior on the cluster mean vector $\boldsymbol{\Theta}_k$, each of the three types of variability λ_ϕ^k, λ_τ^k and λ_ε^k, for all k. We use a uniform distribution on $[0,u]$ as the prior for the λs, and experiment with different values of the upper bound u. Values of u are guided by the data.

We experiment with three options for $\boldsymbol{\Theta}_k$: (i) a zero vector of length J, where J is the number of time points. This is a natural choice for our data considering that the relative gene expression level on the $\log_2$ scale is not much different from 0; (ii) a realization generated from an Ornstein–Uhlenbeck (OU) process [Merton (1971)]. An OU process has four parameters: the starting value, the mean and variation of the process, and the mean-reverting rate. We therefore specify the normal distribution of the starting value, and the normal distribution of the process mean, the uniform distribution of the process variation, and the gamma distribution for the mean-reverting rate; and (iii) a realization generated from a Brownian motion with drift. This process has three parameters: the starting value, the mean and the variation [Taylor and Karlin (1998)]. Similarly, we specify the normal distribution of the starting value, and the normal distribution of the process mean, and the uniform distribution of the process variation. Values of the parameters in these distributions are again guided by the summary statistics of the data.

For the concentration parameter α, we experiment with two options: (i) a Gamma prior with the shape and rate parameters, which can be updated by a Gibbs sampler, as described in Escobar and West (1995); and (ii) a uniform prior on $[0, u']$, where u' can be different values, which is updated by an MH sampler [see Section 2 of the supplemental material; Fu et al. (2013)].

3.4. *The MCMC algorithm for* $\boldsymbol{\xi}$. The complete MCMC algorithm for sampling $\boldsymbol{\xi}$ consists of two major steps in each iteration:

Step 1. For each i from 1 to N, update Z_i using the MH sampler described above;

Step 2. Given the partition from *Step 1*, update other parameters in $\boldsymbol{\xi}$ using Gibbs or MH samplers. Details of this step are in Section 2 of the supplemental material [Fu et al. (2013)].

If the total number of MCMC iterations is S, then the time complexity of this MCMC algorithm is roughly $\mathcal{O}(SJR(4N + K))$, where 4 comes from the steps required in the MH sampler described above, such as generating a proposal, computing the likelihoods and the Hastings ratio.

3.5. *Two-step posterior inference under the Dirichlet-process prior.* For probabilistic clustering, we would like to estimate the posterior allocation probability matrix $\mathbf{P}$ of dimension $N \times K$ with entries $p_{ik} = \Pr(Z_i = k|\mathbf{M})$, each of which is the probability of the ith gene belonging to the kth cluster given the data. This matrix is not part of the parameter vector $\boldsymbol{\xi}$ and is therefore not sampled during MCMC. Below, we propose resampling followed by relabeling to estimate $\mathbf{P}$ from H MCMC samples of $\boldsymbol{\xi}$, while dealing with label-switching [Stephens (2000b)]:

1. *Resampling*: Let $\mathbf{Q}^{(h)}$ of dimension $N \times K^{(h)}$, whose entries are $q_{ik}^{(h)}$, $h = 1, \ldots, H$, be the posterior allocation probability matrix from the hth MCMC sample with arbitrary labeling. The resampling step builds upon an alternative representation of the Dirichlet process as an infinite mixture model [Neal (2000), Green

(2010)]. Specifically, for a Dirichlet process defined in equations (3.1) and (3.2) with concentration parameter α and base distribution G_0, an infinite mixture model representation corresponds to taking the limit in the finite mixture model below, letting $K \to \infty$ and $\alpha^* \to 0$, such that $\alpha^* K \to \alpha$ [Green (2010)]:

$$\begin{aligned} \boldsymbol{\gamma}_k^* &\sim G_0, \qquad k = 1, \ldots, K, \\ (w_1, \ldots, w_K) &\sim \text{Dirichlet}_K(\alpha^*, \ldots, \alpha^*), \\ \Pr(\boldsymbol{\gamma}_i = \boldsymbol{\gamma}_k^*) &= \Pr(Z_i = k) = w_k. \end{aligned}$$

Conditional on the hth MCMC sample, the mixture model for the data becomes finite:

$$\begin{aligned} \bigl(w_1^{(h)}, \ldots, w_{K^{(h)}}^{(h)}\bigr) &\sim \text{Dirichlet}_{K^{(h)}}\bigl(\alpha^{(h)}, \ldots, \alpha^{(h)}\bigr), \\ \Pr\bigl(Z_i^{(h)} = k | \mathbf{w}^{(h)}\bigr) &= w_k^{(h)}, \\ \mathbf{M}_i | \bigl\{Z_i^{(h)} = k, \boldsymbol{\Theta}_k^{(h)}, \boldsymbol{\Sigma}_k^{(h)}\bigr\} &\sim N_{JR}\bigl(\boldsymbol{\Theta}_k^{(h)}, \boldsymbol{\Sigma}_k^{(h)}\bigr). \end{aligned}$$

Then, the posterior probability $q_{ik}^{(h)}$ can be sampled from the following distribution using the hth MCMC sample $\boldsymbol{\xi}^{(h)}$:

$$\begin{aligned} q_{ik}^{(h)} &= \Pr\bigl(Z_i^{(h)} = k | \mathbf{M}, \boldsymbol{\xi}^{(h)}\bigr) \\ &\propto N_{JR}\bigl(\mathbf{M}_i | \boldsymbol{\Theta}_k^{(h)}, \boldsymbol{\Sigma}_k^{(h)}\bigr) w_k^{(h)} \\ &\propto N_{JR}\bigl(\mathbf{M}_i | \boldsymbol{\Theta}_k^{(h)}, \boldsymbol{\Sigma}_k^{(h)}\bigr) \text{Dirichlet}_{K^{(h)}}\bigl(w_k^{(h)} | \alpha^{(h)}, \ldots, \alpha^{(h)}\bigr), \end{aligned}$$

where mixing proportion $w_k^{(h)}$ is generated from a (conditionally) finite Dirichlet distribution. The time complexity of this step is roughly $\mathcal{O}(H(NJR + K))$.

2. *Relabeling*: Labels in $\mathbf{Q}^{(h)}$, $h = 1, \ldots, H$, of dimension $N \times K^{(h)}$, are arbitrary: for example, cluster #2 in $\mathbf{Q}^{(s)}$ does not necessarily correspond to cluster #2 in $\mathbf{Q}^{(t)}$, where $s \neq t$. To deal with arbitrary labeling (also known as "label-switching"), we apply the relabeling algorithm from Stephens (2000b) (Algorithm 2 in that paper) to matrices $\mathbf{Q}$ to "match" the labels across MCMC samples. The dimension of $\mathbf{Q}$s are set to be $N \times K_{\max}$, where $K_{\max}$ is the maximum number of clusters from all recorded MCMC samples. We fill in matrices of lower dimensions with 0s such that all $\mathbf{Q}$s have the same dimension. Stephens' relabeling algorithm then finds a set of permutations, one for the columns of each $\mathbf{Q}$, and the resulting matrix $\mathbf{P}$, such that the Kullback–Leibler distance between $\mathbf{P}$ and column-permuted $\mathbf{Q}$s is minimised. Details of our application, which also implements the Hungarian algorithm [aka Munkres assignment algorithm; Kuhn (1955), Munkres (1957)] for minimisation can be found in Section 3 of the supplemental material [Fu et al. (2013)]. If L is the number of iterations for the relabeling step to achieve convergence, then the time complexity of this step is roughly $\mathcal{O}(LH(NJR + K^3))$, as the time complexity of the Hungarian algorithm is $\mathcal{O}(K^3)$ [Munkres (1957)].

4. Simulations. We investigate the performance of our MH MCMC algorithm and compare the clustering performance of our method with MCLUST [Fraley and Raftery (2006)] and SplineCluster [Heard, Holmes and Stephens (2006)] on data sets simulated from multiple settings, each with different values of variabilities. The size of each data set is comparable to the number of differentially expressed genes we identify from the real time-course data, which we introduced in Section 1 and will describe in detail in Section 5: the number of items N is between 100 and 200, the number of experimental conditions (time points) J is 18, and the number of replicates R is 4. The last two values are identical to those of the real data. However, to keep track of the parameters for individual clusters, we consider only 6 clusters instead of the 14 or 19 clusters our method infers for the real data (Section 5).

For each cluster, we simulated data from a multivariate normal distribution. Specifically, we generated the mean vector from an Ornstein–Uhlenbeck (OU) process with three parameters, which are the initial value, the overall mean and the mean-reverting rate. We constructed the covariance matrix as specified in equation (2.2) with true values of the three types of variability (Table 2). In simulations #1 and #2, all three types of variability are nonzero, with simulation #2 having more extreme within-cluster variability in some clusters. In particular, the level of different types of variability in simulation # 1 is largely comparable to that of 6 of the 14 clusters our method infers for the real time-course data (Section 5). In simulations #3 and #4, only the residual variability is nonzero, with simulation #4 having high variability in some clusters. The simplified covariance structure in the latter two simulations has been adopted in SplineCluster and other methods [Medvedovic and Sivaganesan (2002), Medvedovic, Yeung and Burngarner (2004), Qin (2006)]. Since SplineCluster and MCLUST allow only one replicate per item, we average over the replicates in simulated data and use these sample means as input for SplineCluster and MCLUST, and use default settings in both programs. Also note that neither DIRECT nor MCLUST assumes temporal dependence, whereas SplineCluster does.

Table 3 summarises the performance of DIRECT and compares it to that of SplineCluster and MCLUST. Correctly inferring the number of clusters is key to the overall performance: when the inferred number of clusters is close to the truth, all three methods manage to allocate most of the items to the right clusters and thus achieve a high corrected Rand Index, and vice versa (Tables 3 and 4). Below we discuss the performance of each method in turn.

DIRECT recovers the true clustering consistently well in all the simulations, obtaining high accuracy of cluster assignments of individual items, which is reflected in the high corrected Rand Index (Table 3). Accuracy and consistency come from recovering the true number of (nonsingleton) clusters, as indicated in Table 4. This good performance persists even when the data were simulated under the "wrong" model (simulations #3 and #4). However, DIRECT tends to produce singleton clusters, when those singletons are simulated from clusters of high variation (Table 4).

TABLE 2

Key parameter values used in four sets of simulations. Ten data sets were simulated under each setting. The true number of clusters is 6. *True standard deviations of the three types of variability* (*within-cluster variability, cross-experimental-condition variability and residual variability*) *are given. Size refers to the number of items simulated for each cluster. Standard deviations used in simulation #*1 *are close to some of the clusters inferred for the real time-course data*

		Standard deviation			
Simulations (reps)	K	**Within-cluster** $\sqrt{\lambda_\phi}$	**Expt. cond.** $\sqrt{\lambda_\tau}$	**Resid.** $\sqrt{\lambda_\varepsilon}$	**Size**
#1 (10)	6	0.05	0.01	0.2	80
		0.1	0.05	0.2	20
		0.1	0.05	0.2	10
		0.1	0.05	0.1	10
		0.2	0.1	0.2	70
		0.5	0.1	0.6	10
#2 (10)	6	0.01	0.5	0.5	20
		0.1	0.5	0.5	20
		0.1	0.5	0.5	20
		0.5	0.5	0.5	20
		0.5	0.5	0.5	20
		1	0.5	0.5	20
#3 (10)	6	0	0	0.26	80
		0	0	0.35	20
		0	0	0.35	10
		0	0	0.25	10
		0	0	0.50	70
		0	0	1.20	10
#4 (10)	6	0	0	1.01	20
		0	0	1.1	20
		0	0	1.1	20
		0	0	1.5	20
		0	0	1.5	20
		0	0	2.0	20

MCLUST achieves high accuracy in three out of the four simulations. However, its performance is much worse than DIRECT and SplineCluster in simulation #1: MCLUST tends to infer a higher number of clusters with large variability (Table 4).

In contrast, SplineCluster tends to infer fewer clusters for more heterogeneous data. The dependence structure in simulations #3 and #4 is in fact employed in SplineCluster. However, while SplineCluster infers the number of clusters correctly and allocates the items correctly in simulation #3, it infers a much lower number of clusters in simulation #4, which leads to a much lower corrected Rand Index (Tables 3 and 4). The heterogeneity in simulation #4 (as well as in simula-

TABLE 3

Comparison of methods on simulated data in terms of the corrected Rand Index [Hubert and Arabie (1985)] *to assess clustering accuracy*: *the higher the corrected Rand Index*, *the closer the inferred clustering is to the truth. Each cell displays the mean* (*standard deviation in parentheses*) *of the Rand Index over the* 10 *data sets simulated under each setting. Highest values* (*accounting for variability*) *in each scenario are highlighted*

Simulations	True K	DIRECT	SplineCluster	MCLUST
#1	6	**0.99 (0.01)**	0.84 (0.02)	0.60 (0.13)
#2	6	**0.69 (0.08)**	0.47 (0.10)	**0.71 (0.06)**
#3	6	**0.99 (0.01)**	**1.00 (0.00)**	**1.00 (0.00)**
#4	6	**0.95 (0.04)**	0.47 (0.00)	**0.97 (0.03)**

tion #2) is too high for SplineCluster to distinguish different clusters, it therefore settles on a more parsimonious clustering than the truth.

5. Application to time-course gene expression data.

5.1. *Experimental design and data preprocessing.* As explained in the Introduction, gene expression data were collected using two-colour microarrays from four independent biological replicates of *Drosophila* adult muscle cells at 18 unevenly-spaced time points (in minutes): 0, 5, 10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90, 100, 110, 120, 150, where 0 is the start of a 5-minute treatment of Notch activation [Housden (2011)]. Similar to other gene expression data, the expression measured here is in fact the relative expression of treated cells to control cells, evaluated as the $\log_2$ fold change. The two colours of the microarray were used to distinguish treated and control cells. We applied quantile normalization to the distributions of spot intensities of the two colours across all $18 \times 4 = 72$ arrays. Mapping of the oligonucleotide probes on the microarray to the *Drosophila*

TABLE 4

Comparison of methods on simulated data in terms of the number of nonsingleton (*NS*) *clusters and the number of singleton* (*S*) *clusters inferred under each method. Each cell displays the mean* (*standard deviation in parentheses*) *number of clusters over the* 10 *data sets simulated under each setting. The NS number closest to the truth* (*i.e.*, 6) *in each scenario is highlighted*

		DIRECT		SplineCluster		MCLUST	
Simulations	True K	NS	S	NS	S	NS	S
#1	6	**6.2 (0.4)**	1.7 (1.1)	7.3 (0.5)	0.0 (0.0)	12.0 (2.2)	0.0 (0.0)
#2	6	**7.5 (1.4)**	19.6 (7.2)	3.8 (0.6)	0.2 (0.4)	7.7 (1.1)	0.1 (0.3)
#3	6	6.2 (0.6)	0.6 (0.5)	**6.0 (0.0)**	0.0 (0.0)	**6.0 (0.0)**	0.0 (0.0)
#4	6	6.1 (0.3)	2.8 (2.2)	3.0 (0.0)	0.0 (0.0)	**6.0 (0.0)**	0.0 (0.0)

genome followed FlyBase release 4 and earlier for *Drosophila melanogaster*. After the initial quality screen we retained 7467 expressed genes, that is, the absolute expression levels of genes in the treated and control cells are detectable by the microarray. These retained genes are about half of the *Drosophila* genome. We further imputed missing values in the temporal profiles of these genes [see Section 4 of the supplemental material; Fu et al. (2013)]. These data were challenging to analyse, as the (relative) expression levels of most of these genes were close to 0. To identify differentially expressed (DE) genes over the time course, we applied EDGE [Storey et al. (2005)] to identify 163 such genes at a false discovery rate (FDR) of 10% and 270 genes at an FDR of 25%. However, even among the 163 DE genes, the (relative) expression levels are generally very low (Figure 1).

5.2. *Results from DIRECT.* We ran DIRECT multiple times on both data sets with different initial values. Each run consisted of 10,800 iterations, including 20% burn-in. MCMC samples were recorded every 54th iteration. These runs each took about 8 hours for 163 genes and 12 hours for 270 genes on 2.3 GHz CPUs, including approximately 1 hour for resampling and a few minutes for relabeling. Since the results were consistent across runs, we report below the results from only one run for each data set, averaging the inferred posterior allocation probability matrix across MCMC iterations and defining clusters in terms of the most likely allocations a posteriori.

Our DIRECT method identified 14 clusters for the 163 genes. Clusters differ in both the mean vectors (Figure 2) and the three types of variability (Figure 3). The cluster means differ in the magnitude and timing of the maximal or minimal expression. Because more genes than those allocated to a cluster may have been used for inference of the mean vector, the inferred mean vectors (represented by the coloured curves) are not necessarily located amid the profiles of the genes in that cluster (e.g., cluster # 10, which shows a rather extreme example). In terms of variability, the inferred clusters are homogeneous visually and numerically: the within-cluster variability is small for most inferred clusters, whereas in all clusters the majority of the variability left unexplained by the mixture model is the residual variability, which is the variability between replicates (Figure 3). In several clusters, such as #9, #12 and #14, the estimated within-cluster variability in Figure 3 may seem higher than the clustered mean profiles would indicate (Figure 2). This is because, as mentioned earlier, our probabilistic clustering method estimated these variability terms using more genes than those assigned to the corresponding cluster based on the highest posterior allocation probability. Including these additional genes may increase the within-cluster variability.

Whereas the mean profile plot (Figure 2) and the variability plot (Figure 3) visualise different features of inferred clusters, they do not display the uncertainty and complexity in inferred clustering. For example, gene *CG6018*, inferred to belong to cluster #3 (with peak expression appearing around 100 min; very late response) with probability 0.51, also has a substantial probability of 0.46 to be associated

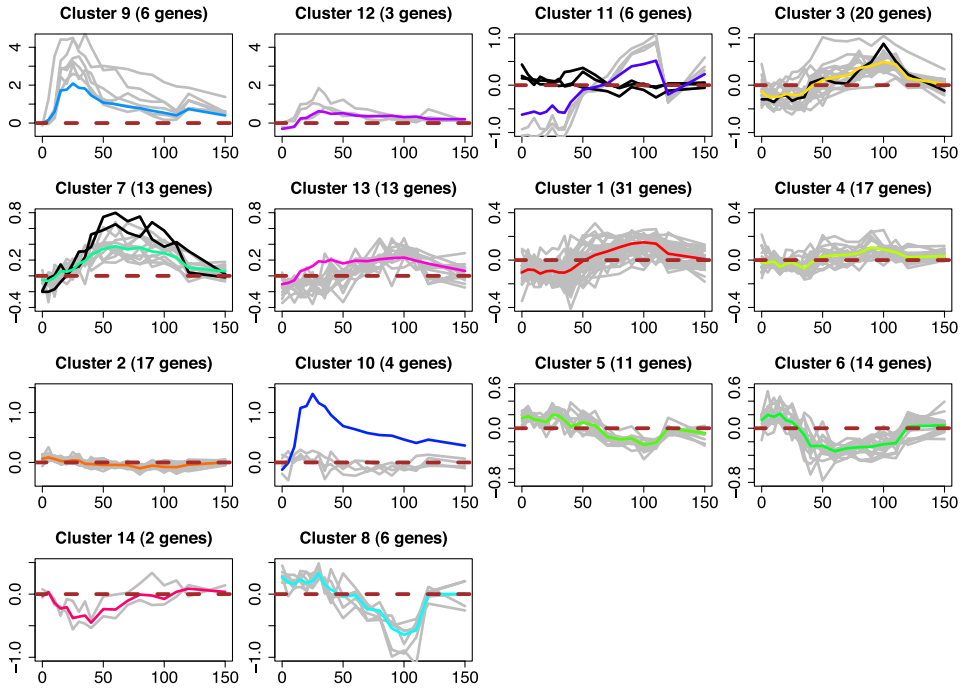

FIG. 2. *Mean profiles* (*gray and black lines*) *of individual genes in clusters inferred under our DIRECT method for the* 163 *significantly expressed genes. Each pair of plots, starting from the top left panel, display the same range on the vertical axis. Each coloured line is the posterior mean estimate of the cluster-specific mean vector. Because more genes than those allocated to a cluster may have been used for inference of the mean vector, the coloured curves* (*inferred mean vectors*) *are not necessarily located amid the profiles of the genes in that cluster* (*e.g., cluster #*10, *which shows a rather extreme example*). *Genes with black lines are analysed in more detail and presented in Figure* 4. *In particular, the three genes with black lines in cluster #*11 *are also allocated to cluster #*10 *or cluster #*5 *with a similar posterior probability* (*see Figure* 4).

with cluster #7 (with peak expression appearing between 50 and 100 min; late response); see Figure 4. Indeed, the replicated profiles of this gene show similarity to the cluster mean profiles of both clusters. Our inference indicates that the temporal profile of *CG6018* is better described by a two-component mixture distribution, sharing features with both clusters. In contrast, the profiles of genes *Cecropin C* (or *CecC*) and *pebbled* (or *peb*) can be adequately represented by one multivariate normal component (Figure 4). Three genes, *CG10080*, *CG12014* and *CG17508*, are better described by a three-component mixture distribution, that is, their expression profiles share features with three clusters (Figure 4).

We apply principal components analysis (PCA) to the posterior allocation probability matrix to visualise the uncertainty and complexity in clustering. Figure 5 shows the scores of the probability matrix based on the first two principal components. Since each row of the probability matrix represents the distribution of

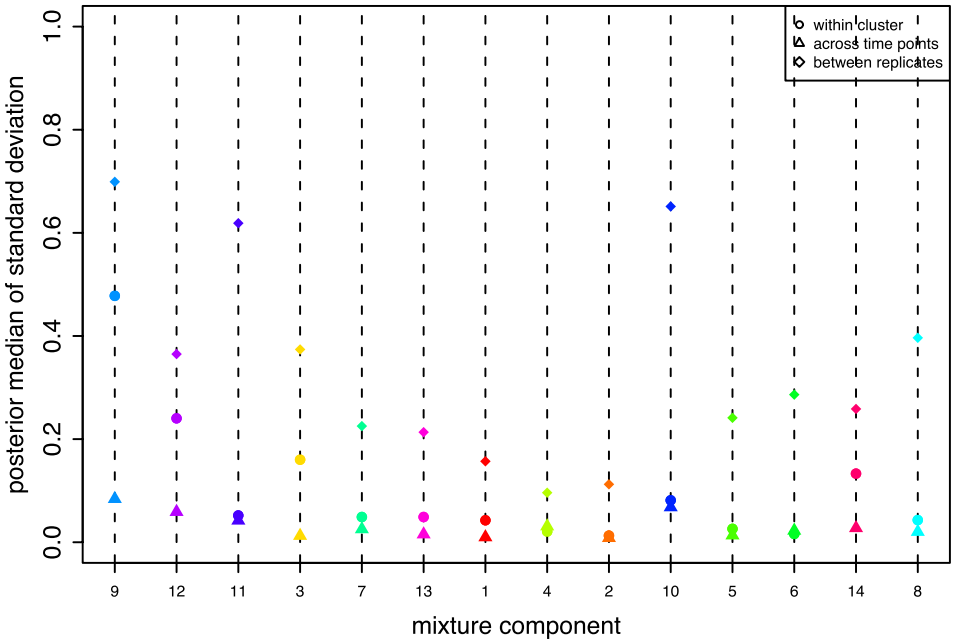

FIG. 3. *Posterior median estimates of standard deviations from our DIRECT program for the three types of variability in each inferred mixture component for the* 163 *significantly expressed genes. Colours and numbering match those in Figure* 2.

cluster allocation for an individual gene, the PCA plot displays the positions of individual genes relative to their allocated clusters and to other clusters. Genes with similar posterior allocation probabilities are located next to each other. Specifically, most of the genes are allocated to a single cluster with probability above 0.8 and stay close to each other in the same cluster on the PCA plot. On the other hand, genes associated with multiple clusters each with a substantial probability are located in between those clusters. For example, the aforementioned gene *CG6018* is positioned between clusters #3 and #7 on this plot.

To examine the sensitivity of our method to specification of the priors, we experimented with different options regarding the priors described in Section 3.3. Specifically, we considered values of 1 and 2 for the upper bound u in the uniform prior for the variability parameters λs, considering that the overall standard deviation in the data is 0.5. We tried all the three options for generating the mean vectors. We computed summary statistics from the data to use as the parameters in the OU process and Brownian motion. For example, we used the sample mean and standard deviation of the data at 0 min as the mean and standard deviation, respectively, of the normal distribution we assume for the starting values of the OU process or the Brownian motion. We also compared the Gibbs and the MH samplers for the concentration parameter α. These different choices turned out not to have much impact on the results.

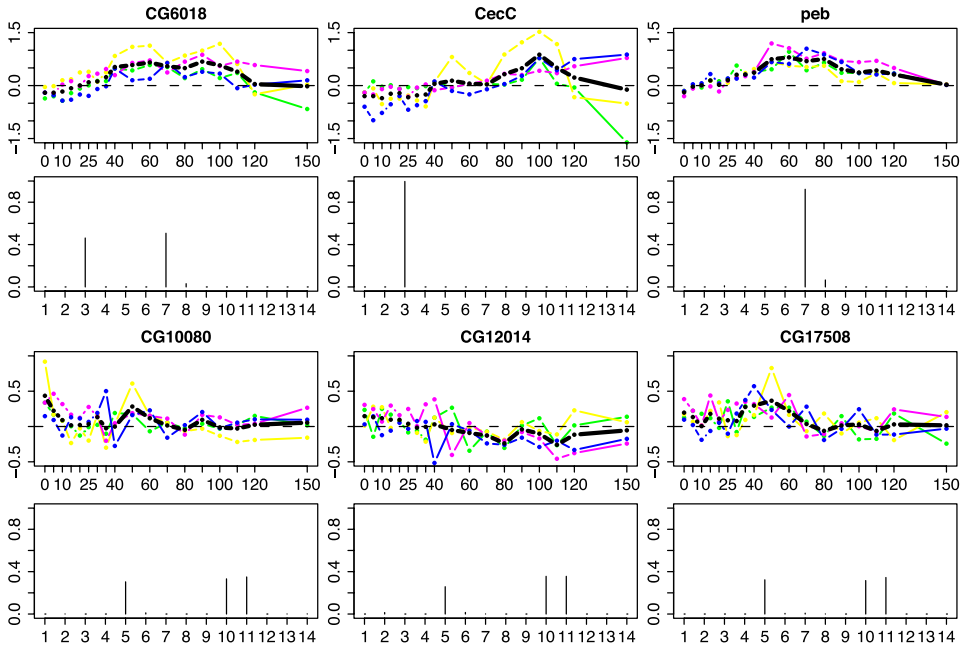

FIG. 4. *Replicated and mean temporal profiles, as well as posterior allocation probabilities, of six genes from the* 163 *gene set. These genes correspond to the black lines in Figure* 2. *For each gene, the top plot shows the replicated (coloured) and mean (black) temporal profiles. Colouring here indicates replicates rather than clustering. The bottom plot shows the inferred posterior probabilities (vertical lines) of allocating the corresponding gene to a cluster (or mixture component). The lengths of the vertical lines sum up to* 1 *in each of these three plots.*

To examine the sensitivity of our method to changes in the data, we applied DIRECT also to the larger data set of 270 genes, identified at an FDR of 25% by EDGE. DIRECT identified 19 clusters for this larger data set [Figures 1–3 in the supplemental material; Fu et al. (2013)]. The cluster allocation is similar to that for the 163 genes, with the additional 107 genes allocated to most of the clusters identified for the 163 genes [Figures 1–3 in the supplemental material; Fu et al. (2013)].

5.3. *Biological implications.* The inferred clustering suggests roughly three stages of gene expression in response to a pulse of Notch activation (Figure 2): before 50 min (early response), between 50 and 100 min (late response), and around and after 100 min (very late response). Clusters 9 and 12 showing early transcriptional responses contain most of the known target genes, that is, Notch has a direct impact on the transcriptional changes of these genes. Cluster 7 showing late responses also contains 3–5 known targets [Krejci et al. (2009)], but approximately 10 other genes in this cluster may also be Notch targets. Genes in other late or very late response clusters may be Notch targets as well. Together with our collabora-

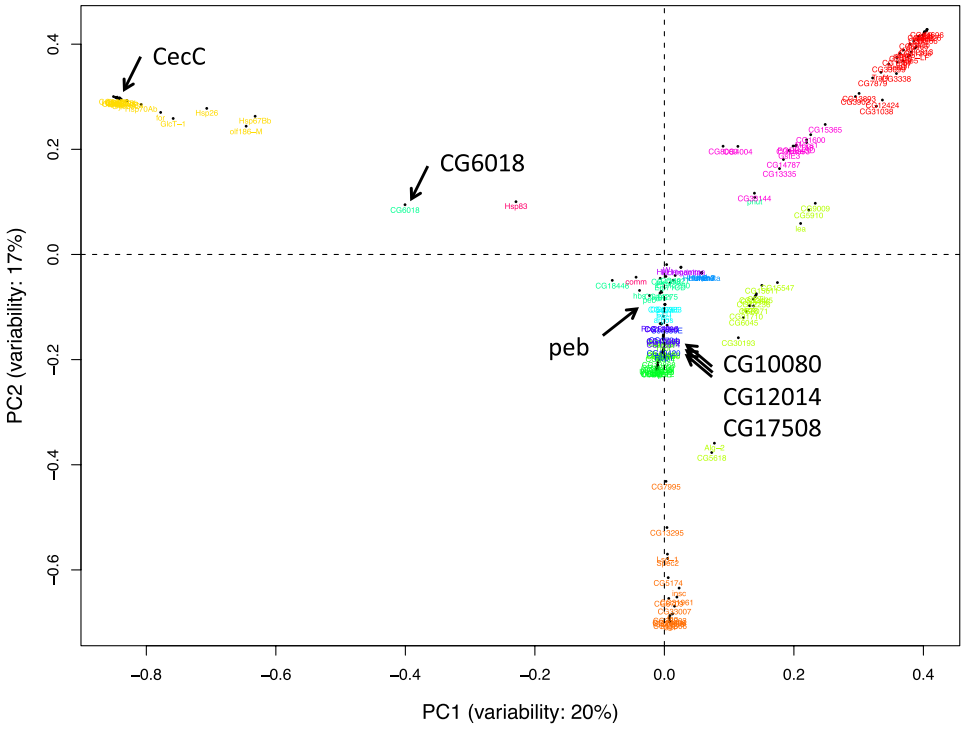

FIG. 5. *PCA plot of the posterior allocation probability matrix for* 163 *genes. These colours match those in Figures* 2 *and* 3. *Six arrows point to the six genes also highlighted in Figure* 2 *and examined in Figure* 4.

tors, we analysed data from additional experiments to examine whether this is the case [Housden et al. (2013)]. Furthermore, it is known that Notch generally promotes transcription rather than represses it, and that the early-upregulated genes in cluster 9 are strong repressors. Our clustering therefore suggests unknown, complex regulation mechanisms involving interactions between different clusters of genes. With additional experiments, Housden et al. (2013) investigated possible transcriptional regulation mechanisms and identified a feed-forward regulation relationship among clusters 9, 6 and 7.

5.4. *Results from SplineCluster and MCLUST.* For comparison, we ran SplineCluster and MCLUST on the two real data sets, using the average profiles and the default settings (Table 5). SplineCluster inferred only 7 clusters for both data sets, with the inferred clusters exhibiting a much higher level of heterogeneity than under our DIRECT method [Figures 4–5 in the supplemental material; Fu et al. (2013)]. This result is consistent with its performance on simulated data: SplineCluster also tends to infer a lower number of clusters in case of high heterogeneity (Section 4 and Table 4). MCLUST inferred 15 clusters for 163 genes,

TABLE 5
Numbers of clusters estimated by three clustering methods: *DIRECT*, *SplineCluster and MCLUST for genes identified by EDGE* [*Storey et al.* (2005)] *to be differentially expressed over the time course*

	No. of inferred clusters	
	163 Genes (FDR 10%)	**270 Genes (FDR 25%)**
DIRECT	14	19
SplineCluster	7	7
MCLUST	15	2

which is comparable to our DIRECT method [Figures 6 and 8 in the supplemental material; Fu et al. (2013)]. However, it inferred only 2 clusters for 270 genes and a different covariance model [Figures 7 and 9 in the supplemental material; Fu et al. (2013)]. This sensitivity of clustering to the relatively minor change in the data may have arisen from MCLUST trying to simultaneously select the number of clusters and the covariance model. Selection of the covariance model adds another layer of complexity to the problem of clustering, particularly when none of the different covariance models considered by MCLUST is compatible with the experimental design. The uncertainty in the covariance model selection may also explain the particularly high variability in the inferred number of clusters for simulated data in simulation #1 (Table 4).

6. Discussion. In this paper we developed DIRECT, a model-based Bayesian clustering method for noisy, short and replicated time-course data. We implemented this method in the R package DIRECT, which may be downloaded from CRAN (http://cran.r-project.org/web/packages/). We also applied this method to analyse the time-course microarray gene expression levels following Notch activation in *Drosophlia* adult muscle cells. Our analysis identified 14 clusters in 163 differentially expressed genes and assigned probabilities of cluster membership for each gene. The clustering results indicate three time periods during which genes attain peak up- or down-regulation, which was previously unknown, and suggest possibilities for the underlying mechanisms of transcription regulation that may involve interactions between genes in different clusters. Hypotheses on the biological mechanisms are further investigated in Housden et al. (2013). Here we discuss several additional aspects of the clustering method.

Our method has four main features. First, the random-effects mixture model decomposes the total variability in the data into three types of variability that arise from clustering (λ_ϕ), from sampling across multiple experimental conditions (λ_τ), and from sampling a limited number of replicates (λ_ε). This variance decomposition regularises the covariance matrix with constraints that are consistent with

the experimental design. It is simultaneously parsimonious and identifiable for the replicated data: the replicated profiles at multiple time points of a single gene are already informative for λ_τ and λ_ε, and having at least 2 genes in a cluster makes λ_ϕ estimable. Second, our method uses the Dirichlet-process prior to induce a prior distribution on clustering as well as the number of clusters, making it possible to estimate directly both unknowns from the data. Third, we have developed a novel Metropolis–Hastings MCMC algorithm for sampling under the Dirichlet-process prior. Our MH algorithm allows the use of nonconjugate priors. It is also efficient and accurate, as simulation studies demonstrate. Fourth, our method infers the posterior allocation probability matrix through resampling and relabeling of the MCMC samples. This probability matrix can then be used directly in forming clusters and making probabilistic cluster allocations. Simulation studies and application to real data show that DIRECT is sensitive enough to variability in the data to identify homogeneous clusters, but not too sensitive to minor changes in the data.

Several other model-based clustering methods construct their models along similar lines [Celeux, Martin and Lavergne (2005), Ma et al. (2006), Zhou and Wakefield (2006), Booth, Casella and Hobert (2008)]. In fact, our model in equation (2.1) coincides with the random-effects model E3 in Celeux, Martin and Lavergne (2005). However, those authors decided to focus on a slightly simpler model, which is similar to equation (2.1) but without the within-component random effects ϕ_i^k. They based their decision on the nearly identical likelihoods of the two models for simulated data. Ma et al. (2006) and Zhou and Wakefield (2006) did not deal with replicated data and included in their model only two types of variability: the within-cluster variability and the variability due to multiple time points. Similar to us, Booth, Casella and Hobert (2008) worked with replicated time-course data and used random effects to account for different types of noise, but their partition of the total variability is not based on the experimental design and is therefore much less straightforward. Specifically, they allowed for dependence among different items in the same cluster but did not explicitly account for the random effect due to time (or experimental condition).

Note that our DIRECT method does not account for the temporal structure, but rather focuses on modeling the covariance matrix. This approach is similar to MCLUST, which applies eigenvalue decomposition to the covariance matrix and considers various constraints on the decomposed covariance matrix (i.e., whether the shape, orientation or volume of the covariance matrix is identical across mixture components), although the constraints considered in MCLUST are not based on any experimental design. The good performance of our method on both simulated and real data, and of MCLUST in several cases, suggests that accounting for the temporal structure in the mean vectors, such as via splines functions as in SplineCluster or via Gaussian processes as in Zhou and Wakefield (2006) and others, may not be necessary. We also followed the approach in Zhou and Wakefield (2006) and modeled the mean vector of each mixture component as a Brownian

motion (with drift) and, extending this idea, as an Ornstein–Uhlenbeck process. The clustering results such as the inferred number of clusters and allocation of individual genes did not change much, because these approaches impose the temporal structure on the mean vector: conditioning on the correct clustering, the data are directly informative of the cluster mean, a main parameter of interest. Incidentally, DIRECT is applicable also in more general cases of multiple experimental conditions, where dependence among conditions is nonexistent, unclear or unknown.

Similar to other MCMC methods, our DIRECT method does not aim to optimise the runtime. Whereas MCLUST and SplineCluster, both non-MCMC methods, took only seconds or at most minutes to run on the simulated and real data here, we ran DIRECT for hours to ensure the consistency in results across different runs, which indicated that the Markov chain had mixed well.

We have used only the one-parameter Dirichlet-process prior in our method. The concentration parameter in the Dirichlet-process prior simultaneously controls the number of clusters as well as the size of each individual cluster. The prior has the tendency of creating clusters of very different sizes. The posterior inference to generate the posterior allocation probability matrix is therefore critical to balance out the unevenness: although certain clusters may be very small or very big in a single iteration, items allocated to these tiny clusters are likely allocated to other, possibly larger, clusters over a sufficient number of MCMC iterations. Nonetheless, as pointed out by the Associate Editor and an anonymous reviewer, other exchangeable priors, such as the two-parameter Dirichlet process [aka the Pitman-Yor process; Pitman and Yor (1997)] and many other extensions of the Dirichlet process reviewed in Hjort et al. (2010), may also be adopted under our framework. Indeed, these other exchangeable priors may offer more flexibility and suggest an important direction to extend our current work.

Under our and Neal (2000)'s MH MCMC algorithms, the Markov chain is constructed for the cluster memberships of individual items. Generation of a new cluster and elimination of an existing cluster are implied rather than enforced. In contrast, reversible-jump MCMC [Richardson and Green (1997)] and birth-death MCMC [Stephens (2000a)] enforce changes in dimensions by designing the MCMC moves around the number of clusters. Their strategy may not be efficient for clustering multivariate data, because even a fixed number of clusters may correspond to a large number of possible partitions and a large space of the cluster-specific parameter values. For clustering it seems more sensible for the Markov chain to move to the neighbourhood of the "correct" number of clusters and to fully explore the parameter space in this neighbourhood, as under Neal's approaches and under our method.

APPENDIX: PROOF OF PROPOSITION 1

We use the joint distribution of clustering and the number of clusters given in equation (3.4) for derivation. Let K_{-i} be the number of clusters when the ith gene

is excluded. Then,

$$\begin{aligned}
&\Pr(Z_i = z, K = k | \mathbf{Z}_{-i} = \mathbf{z}_{-i}, \alpha) \\
&\quad = \frac{\Pr(\mathbf{Z} = \mathbf{z}, K = k | \alpha)}{\Pr(\mathbf{Z}_{-i} = \mathbf{z}_{-i}, K_{-i} = k_{-i} | \alpha)} \\
&\quad = \frac{\Gamma(\alpha)/\Gamma(\alpha + N)\alpha^k \prod_{l=1}^{k}(N_l - 1)!}{\Gamma(\alpha)/\Gamma(\alpha + N - 1)\alpha^{k_{-i}} \prod_{s=1}^{k_{-i}}(N_s - 1)!} \\
&\quad = \begin{cases} \dfrac{N_z - 1}{N - 1 + \alpha}, & Z_i \text{ is not in a singleton cluster}, \\ \dfrac{\alpha}{N - 1 + \alpha}, & Z_i \text{ is in a singleton cluster}. \end{cases}
\end{aligned}$$

Alternatively, Neal (2000) derived the above result first under the finite mixture model, treating K as a constant, and then letting $K \to \infty$.

Acknowledgements. The authors thank Ben Housden, Alena Krejci and Bettina Fischer for collecting and sharing the time-course microarray gene expression data analysed here. AQF also thanks Jurgen Van Gael, Richard Samworth, Richard Nickl, Pat Altham and Matthew Stephens for helpful discussions on statistics and to Robert Stojnic and Laurent Gatto for advice on R programming. Thanks also go to Editor Dr. Karen Kafadar, the Associated Editor and two anonymous reviewers, whose thorough and insightful comments greatly improved the manuscript.

SUPPLEMENTARY MATERIAL

Appendices (DOI: 10.1214/13-AOAS650SUPP; .pdf). The pdf file contains the proof of Proposition 2, details on the MCMC algorithm and additional figures.

REFERENCES

ANTONIAK, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.* **2** 1152–1174. MR0365969

BEAL, M. J. and KRISHNAMURTHY, P. (2006). Gene expression time course clustering with countably infinite hidden Markov models. In *Proc. Conference on Uncertainty in Artificial Intelligence*.

BOOTH, J. G., CASELLA, G. and HOBERT, J. P. (2008). Clustering using objective functions and stochastic search. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **70** 119–139. MR2412634

BRAY, S. J. (2006). Notch signalling: A simple pathway becomes complex. *Nat. Rev. Mol. Cell Bio.* **7** 678–689.

CELEUX, G., MARTIN, O. and LAVERGNE, C. (2005). Mixture of linear mixed models for clustering gene expression profiles from repeated microarray experiments. *Stat. Model.* **5** 1–25.

COOKE, E. J., SAVAGE, R. S., KIRK, P. D. W., DARKINS, R. and WILD, D. L. (2011). Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements. *BMC Bioinformatics* **12** 399.

DHAVALA, S. S., DATTA, S., MALLICK, B. K., CARROLL, R. J., KHARE, S., LAWHON, S. D. and ADAMS, L. G. (2010). Bayesian modeling of MPSS data: Gene expression analysis of bovine *Salmonella* infection. *J. Amer. Statist. Assoc.* **105** 956–967. MR2752592

DUNSON, D. B. (2010). Nonparametric Bayes applications to biostatistics. In *Bayesian Nonparametrics* (N. L. Hjort, C. Holmes, P. Müller and S. G. Walker, eds.) *Cambridge Series on Statistical and Probabilistic Mathematics* **28** 223–273. Cambridge Univ. Press, Cambridge. MR2730665

ELOWITZ, M. B., LEVINE, A. J., SIGGIA, E. D. and SWAIN, P. S. (2002). Stochastic gene expression in a single cell. *Science* **297** 1183–1186.

ESCOBAR, M. D. and WEST, M. (1995). Bayesian density estimation and inference using mixtures. *J. Amer. Statist. Assoc.* **90** 577–588. MR1340510

FERGUSON, T. S. (1973). A Bayesian analysis of some nonparametric problems. *Ann. Statist.* **1** 209–230. MR0350949

FOX, E. B. (2009). Bayesian nonparametric learning of complex dynamical phenomena. Ph.D. thesis, MIT, Cambridge, MA.

FRALEY, C. and RAFTERY, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *J. Amer. Statist. Assoc.* **97** 611–631. MR1951635

FRALEY, C. and RAFTERY, A. E. (2006). MCLUST version 3 for R: Normal mixture modeling and model-based clustering. Technical Report 504, Dept. Statistics, Univ. Washington, Seattle, WA.

FU, A. Q., RUSSELL, S., BRAY, S. J. and TAVARÉ, S. (2013). Supplement to "Bayesian clustering of replicated time-course gene expression data with weak signals." DOI:10.1214/13-AOAS650SUPP.

GREEN, P. J. (2010). Colouring and breaking sticks: Random distributions and heterogeneous clustering. In *Probability and Mathematical Genetics* (N. H. Bingham and C. M. Goldie, eds.). *London Mathematical Society Lecture Note Series* **378** 319–344. Cambridge Univ. Press, Cambridge. MR2744245

GRIFFIN, J. and HOLMES, C. (2010). Computational issues arising in Bayesian nonparametric hierarchical models. In *Bayesian Nonparametrics* (N. L. Hjort, C. Holmes, P. Müller and S. G. Walker, eds.). *Cambridge Series on Statistical and Probabilistic Mathematics* **28** 208–222. Cambridge Univ. Press, Cambridge. MR2730664

HEARD, N. A., HOLMES, C. C. and STEPHENS, D. A. (2006). A quantitative study of gene regulation involved in the immune response of *Anopheline* mosquitoes: An application of Bayesian hierarchical clustering of curves. *J. Amer. Statist. Assoc.* **101** 18–29. MR2252430

HJORT, N. L., HOLMES, C., MÜLLER, P. and WALKER, S. G., eds. (2010). *Bayesian Nonparametrics. Cambridge Series in Statistical and Probabilistic Mathematics* **28**. Cambridge Univ. Press, Cambridge. MR2722987

HOUSDEN, B. (2011). Notch targets and EGFR pathway regulation. Ph.D. thesis, Univ. Cambridge.

HOUSDEN, B. E., FU, A. Q., KREJCI, A., BERNARD, F., FISCHER, B., TAVARÉ, S., RUSSELL, S. and BRAY, S. J. (2013). Transcriptional dynamics elicited by a short pulse of Notch activation involves feed-forward regulation by *E(spl)/Hes* genes. *PLoS Genet.* **9** e1003162.

HUBERT, L. and ARABIE, P. (1985). Comparing partitions. *J. Classification* **2** 193–218.

JAIN, S. and NEAL, R. M. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *J. Comput. Graph. Statist.* **13** 158–182. MR2044876

JAIN, S. and NEAL, R. M. (2007). Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Anal.* **2** 445–472. MR2342168

JENNINGS, B., PREISS, A., DELIDAKIS, C. and BRAY, S. (1994). The Notch signalling pathway is required for *Enhancer of split* bHLH protein expression during neurogenesis in the *Drosophila* embryo. *Development* **120** 3537–3548.

KALLI, M., GRIFFIN, J. E. and WALKER, S. G. (2011). Slice sampling mixture models. *Stat. Comput.* **21** 93–105. MR2746606

KREJCI, A., BERNARD, F., HOUSDEN, B. E., COLLINS, S. and BRAY, S. J. (2009). Direct response to Notch activation: Signaling crosstalk and incoherent logic. *Sci. STKE* **2** ra1.

KUHN, H. W. (1955). The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.* **2** 83–97. MR0075510

LAU, J. W. and GREEN, P. J. (2007). Bayesian model-based clustering procedures. *J. Comput. Graph. Statist.* **16** 526–558. MR2351079

MA, P., CASTILLO-DAVIS, C. I., ZHONG, W. and LIU, J. S. (2006). A data-driven clustering method for time course gene expression data. *Nucleic Acids Res.* **34** 1261–1269.

MACEACHERN, S. N. and MÜLLER, P. (1998). Estimating mixture of Dirichlet process models. *J. Comput. Graph. Statist.* **7** 223–238.

MCADAMS, H. H. and ARKIN, A. (1997). Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA* **94** 814–819.

MCNICHOLAS, P. D. and MURPHY, T. B. (2010). Model-based clustering of longitudinal data. *Canad. J. Statist.* **38** 153–168. MR2676935

MEDVEDOVIC, M. and SIVAGANESAN, S. (2002). Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics* **18** 1194–1206.

MEDVEDOVIC, M., YEUNG, K. Y. and BURNGARNER, R. E. (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics* **20** 1222–1232.

MERTON, R. C. (1971). Optimum consumption and portfolio rules in a continuous-time model. *J. Econom. Theory* **3** 373–413. MR0456373

MUNKRES, J. (1957). Algorithms for the assignment and transportation problems. *J. Soc. Indust. Appl. Math.* **5** 32–38. MR0093429

NEAL, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *J. Comput. Graph. Statist.* **9** 249–265. MR1823804

PAPASPILIOPOULOS, O. and ROBERTS, G. O. (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika* **95** 169–186. MR2409721

PITMAN, J. (2006). *Combinatorial Stochastic Processes. Lecture Notes in Math.* **1875**. Springer, Berlin. MR2245368

PITMAN, J. and YOR, M. (1997). The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *Ann. Probab.* **25** 855–900. MR1434129

QIN, Z. S. (2006). Clustering microarray gene expression data using weighted Chinese restaurant process. *Bioinformatics* **22** 1988–1997.

RASMUSSEN, C. E., DE LA CRUZ, B. J., GHAHRAMANI, Z. and WILD, D. L. (2009). Modeling and visualizing uncertainty in gene expression clusters using Dirichlet process mixtures. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **6** 615–628.

RICHARDSON, S. and GREEN, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **59** 731–792. MR1483213

SCHLIEP, A., COSTA, I. G., STEINHOFF, C. and SCHÖNHUTH, A. (2005). Analyzing gene expression time-courses. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2** 179–193.

SEARLE, S. R., CASELLA, G. and MCCULLOCH, C. E. (2006). *Variance Components*. Wiley-Interscience, Hoboken, NJ. MR2298115

SPUDICH, J. L. and KOSHLAND, J. D. E. (1976). Non-genetic individuality: Chance in the single cell. *Nature* **262** 467–471.

STEPHENS, M. (2000a). Bayesian analysis of mixture models with an unknown number of components—an alternative to reversible jump methods. *Ann. Statist.* **28** 40–74. MR1762903

STEPHENS, M. (2000b). Dealing with label switching in mixture models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **62** 795–809. MR1796293

STOREY, J. D., XIAO, W., LEEK, J. T., TOMPKINS, R. G. and DAVIS, R. W. (2005). Significance analysis of time course microarray experiments. *Proc. Natl. Acad. Sci. USA* **102** 12837–12842.

TAYLOR, H. M. and KARLIN, S. (1998). *An Introduction to Stochastic Modeling*, 3rd ed. Academic Press, San Diego, CA. MR1627763

WALKER, S. G. (2007). Sampling the Dirichlet mixture model with slices. *Comm. Statist. Simulation Comput.* **36** 45–54. MR2370888

ZHOU, C. and WAKEFIELD, J. (2006). A Bayesian mixture model for partitioning gene expression data. *Biometrics* **62** 515–525. MR2236834

A. Q. FU
DEPARTMENT OF PHYSIOLOGY, DEVELOPMENT AND NEUROSCIENCE
UNIVERSITY OF CAMBRIDGE
DOWNING STREET
CAMBRIDGE CB2 3DY
UNITED KINGDOM
AND
CAMBRIDGE SYSTEMS BIOLOGY CENTRE
TENNIS COURT ROAD
CAMBRIDGE CB2 1QR
UNITED KINGDOM
CURRENT ADDRESS:
DEPARTMENT OF HUMAN GENETICS
UNIVERSITY OF CHICAGO
920 E 58TH STREET
CHICAGO, ILLINOIS 60637
USA
E-MAIL: audreyqyfu@uchicago.edu

S. J. BRAY
DEPARTMENT OF PHYSIOLOGY, DEVELOPMENT AND NEUROSCIENCE
UNIVERSITY OF CAMBRIDGE
DOWNING STREET
CAMBRIDGE CB2 3DY
UNITED KINGDOM
E-MAIL: sjb32@cam.ac.uk

S. RUSSELL
DEPARTMENT OF GENETICS
UNIVERSITY OF CAMBRIDGE
DOWNING STREET
CAMBRIDGE CB2 3EH
UNITED KINGDOM
AND
CAMBRIDGE SYSTEMS BIOLOGY CENTRE
TENNIS COURT ROAD
CAMBRIDGE CB2 1QR
UNITED KINGDOM
E-MAIL: s.russell@gen.cam.ac.uk

S. TAVARÉ
DEPARTMENT OF APPLIED MATHEMATIC AND THEORETICAL PHYSICS
UNIVERSITY OF CAMBRIDGE
CENTRE FOR MATHEMATICAL SCIENCES
WILBERFORCE ROAD
CAMBRIDGE CB3 0WA
UNITED KINGDOM
AND
PROGRAM IN MOLECULAR AND COMPUTATIONAL BIOLOGY
UNIVERSITY OF SOUTHERN CALIFORNIA
1050 CHILDS WAY, RRI 201B
LOS ANGELES, CALIFORNIA 90089-2910
USA
E-MAIL: st321@cam.ac.uk