ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS

Journal of the Royal Statistical Society
Applied Statistics
Series C

Appl. Statist. (2018)
67, *Part* 1, *pp.* 3–23

A Bayesian model selection approach for identifying differentially expressed transcripts from RNA sequencing data

Panagiotis Papastamoulis and Magnus Rattray

University of Manchester, UK

[Received October 2015. Final revision December 2016]

Summary. Recent advances in molecular biology allow the quantification of the transcriptome and scoring transcripts as differentially or equally expressed between two biological conditions. Although these two tasks are closely linked, the available inference methods treat them separately: a primary model is used to estimate expression and its output is post processed by using a differential expression model. In the paper, both issues are simultaneously addressed by proposing the joint estimation of expression levels and differential expression: the unknown relative abundance of each transcript can either be equal or not between two conditions. A hierarchical Bayesian model builds on the BitSeq framework and the posterior distribution of transcript expression and differential expression is inferred by using Markov chain Monte Carlo sampling. It is shown that the model proposed enjoys conjugacy for fixed dimension variables; thus the full conditional distributions are analytically derived. Two samplers are constructed, a reversible jump Markov chain Monte Carlo sampler and a collapsed Gibbs sampler, and the latter is found to perform better. A cluster representation of the aligned reads to the transcriptome is introduced, allowing parallel estimation of the marginal posterior distribution of subsets of transcripts under reasonable computing time. Under a fixed prior probability of differential expression the clusterwise sampler has the same marginal posterior distributions as the raw sampler, but a more general prior structure is also employed. The algorithm proposed is benchmarked against alternative methods by using synthetic data sets and applied to real RNA sequencing data. Source code is available on line from https://github.com/mqbssppe/cjBitSeq.

Keywords: Collapsed Gibbs sampler; Mixture models; Reversible jump Markov chain Monte Carlo sampling; Ribonucleic acid sequencing

1. Introduction

Quantifying the transcriptome of a given organism or cell is a fundamental task in molecular biology. Ribonucleic acid sequencing (which is known as 'RNA-seq') technology produces transcriptomic data in the form of short reads (Mortazavi *et al.*, 2008). These reads can be used either to reconstruct the transcriptome by using *de novo* or guided assembly, or to estimate the abundance of known transcripts given a reference annotation. Here, we consider the latter scenario in which transcripts are defined by annotation. In such a case, millions of short reads are aligned to the reference transcriptome (or genome) by using mapping tools such as 'Bowtie' (Langmead *et al.*, 2009) (or 'TopHat' (Trapnell *et al.*, 2009)). Of particular interest is the identification of differentially expressed transcripts (or isoforms) across different samples. Throughout this paper the term transcript refers to isoforms, so differential transcript detection has the same meaning as differential isoform detection. Most genes in higher eukaryotes can be spliced into

Address for correspondence: Panagiotis Papastamoulis, Faculty of Life Science, University of Manchester, B.1082 Michael Smith Building, Oxford Road, Manchester, M13 9PL, UK.
E-mail: panagiotis.papastamoulis@manchester.ac.uk

alternative transcripts that share specific parts of their nucleotide sequence. Thus, a short read is not uniquely aligned to the transcriptome and its origin remains uncertain, making transcript expression estimation non-trivial. Probabilistic models provide a powerful means to estimate transcript abundances as they can take this ambiguous read assignment into consideration in a principled manner.

There are numerous methods that estimate transcript expression from RNA-seq data, including 'RNA-seq by expectation–maximization' (Li and Dewey, 2011), 'isoform estimation by expectation–maximization' (Nicolae *et al*., 2011), 'Cufflinks' (Trapnell *et al*., 2010, 2013), 'BitSeq' (stage 1) (Glaus *et al*., 2012), 'transcript isoform estimation with gapped alignment of RNA-seq data' (Nariai *et al*., 2013) and 'Casper' (Rossell *et al*., 2014). Some of these methods also include a second stage for performing differential expression (DE) analysis at the transcript level (e.g. 'Cuffdiff' and BitSeq stage 2) and stand-alone methods for transcript level DE calling have also been developed such as 'EBSeq' (Leng *et al*., 2013) and 'MetaDiff' (Jia *et al*., 2015). Cuffdiff uses an asymptotically normal test statistic by applying the delta method to the log-ratio of transcript abundances between two samples, given the estimated expression levels using Cufflinks. EBSeq estimates the Bayes factor of a model under DE or non-DE for each transcript, building a negative binomial model on the estimated read counts from any method. BitSeq stage 2 ranks transcripts as differentially expressed by the probability of positive log-ratio based on the Markov chain Monte Carlo MCMC output from BitSeq stage 1, which estimates the expression levels by assuming a mixture model. Gene level DE analysis is also available by using count-based methods such as 'edgeR' (Robinson *et al*., 2010) and 'DESeq' (Anders and Huber, 2010) but here we limit our attention to methods that are designed for transcript level DE calling.

All existing methods for transcript level DE calling apply a two-step procedure. The mapped RNA-seq data are used as input of a first-stage analysis to estimate transcript expression. The output of this stage is then post processed at a second stage to classify transcripts as differentially expressed or non-differentially expressed. The bridge between the two stages is based on certain parametric assumptions for the distribution of the estimates of the first-stage and/or the use of asymptotic results (as previously described above). Also, transcript level expression estimates are correlated through sharing of reads and this correlation is typically ignored in the second stage. Such two-stage approaches are quite useful in practice since the DE question is not always the main aim of the analysis; therefore estimating expression is useful in itself. However, when the main purpose of an experiment is DE calling then the two-stage procedure increases the modelling complexity and may result in overfitting, since there is no guarantee that the underlying assumptions are valid. Note that a recent method (Gu *et al*., 2014) addresses the joint estimation of expression and DE modelling of exon counts under a Bayesian approach but at the gene level rather than the transcript level that is considered here.

The contribution of this paper is to develop a method for the joint estimation of expression and DE at the transcript level. The method builds on the Bayesian framework of the BitSeq (stage 1) model where transcript expression estimation reduces to estimating the posterior distribution of the weights of a mixture model by using MCMC sampling (Glaus *et al*., 2012). The novelty in the present study is that DE is addressed by inferring which weights differ between two mixture models. This is achieved by using two samplers. A reversible jump MCMC (RJMCMC) algorithm (Green, 1995) updates both transcript expression and DE parameters, and a collapsed Gibbs algorithm is developed which avoids transdimensional transitions. The high dimensional setting of RNA-seq data studies makes the convergence to the joint posterior distribution computationally challenging. To alleviate this computational burden and to allow easier parallelization, a new cluster representation of the transcriptome is introduced which collapses the problem to subsets of transcripts sharing aligned reads.

The rest of the paper is organized as follows. The mixture model that is used in the original BitSeq set-up is reviewed in Section 2.1. The prior assumptions of the new (clusterwise joint BitSeq) model called cjBitSeq is introduced in Section 2.2. The full conditional distributions are given in Section 2.3 and two MCMC samplers are described in Section 2.4. A cluster representation of aligned reads and transcripts is discussed in Section 2.5 and details over false discovery rate (FDR) estimation are given in Section 2.6. Large-scale simulation studies are presented in Section 3.2 and the method proposed is illustrated on a real human data set in Section 3.3. The paper concludes in Section 4 with a synopsis and discussion.

2. Methods

In the BitSeq model, the mixture components correspond to annotated transcript sequences and the mixture weights correspond to their relative expression levels. The data likelihood is then computed by considering the alignment of reads (or read pairs) against each mixture component. Essentially, this model is modified here to construct a well-defined probability of DE or non-DE when two samples are available.

We induce a set of free parameters of varying dimension, depending on the number of different weights between two mixture models. Assuming two independent Dirichlet prior distributions, the Gibbs sampler (Geman and Geman, 1984; Gelfand and Smith, 1990) draws samples from the full conditionals, which are independent Dirichlet and generalized Dirichlet (Connor and Mosimann, 1969; Wong, 1998, 2010) distributions. This representation allows the integration of the corresponding parameters as stated in theorem 2. Therefore, we provide two MCMC samplers depending on whether transcript expression levels are integrated out or not. These samplers converge to the same target distribution but using different steps to update the state of each transcript: the first uses a birth–death move type (Richardson and Green, 1997; Papastamoulis and Iliopoulos, 2009) and the second is a block update from the full conditional distribution. After detecting clusters of transcripts and reads, it is shown that the parallel application of the algorithm to each cluster converges to proper marginals of the full posterior distribution.

2.1. BitSeq

Let $\mathbf{x} = (x_1, \ldots, x_r)$, $x_i \in \mathcal{X}$, $i = 1, \ldots, r$, denote a sample of r short reads aligned to a given set of K transcripts. The sample space $\mathcal{X}$ consists of all sequences of letters A, C, G and T. Assuming that reads are independent, the joint probability density function of the data is written as

$$\mathbf{x}|\boldsymbol{\theta} \sim \prod_{i=1}^{r} \sum_{k=1}^{K} \theta_k f_k(x_i). \tag{1}$$

The number of components, K, is equal to the number of transcripts and it is considered as known since the transcriptome is given. The parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K) \in \mathcal{P}_{K-1}$ denotes relative abundances, where

$$\mathcal{P}_{K-1} := \left\{ p_k \geqslant 0, k = 1, \ldots, K-1 : \sum_{k=1}^{K-1} p_k \leqslant 1; p_K := 1 - \sum_{k=1}^{K-1} p_k \right\}.$$

The component-specific density $f_k(\cdot)$ corresponds to the probability of a read aligning at some position of transcript k, $k = 1, \ldots, K$. Since we assume a known transcriptome, $\{f_k\}_{k=1}^{K}$ are known as well and they are computed according to the methodology that is described in Glaus *et al*. (2012) (see also appendix A in the on-line supplementary material), taking into account position and sequence-specific bias correction methods.

A priori it is assumed that $\boldsymbol{\theta} \sim \mathcal{D}_{K-1}(\alpha_1, \ldots, \alpha_K)$, with $\mathcal{D}_j$ denoting the Dirichlet distribution defined over $\mathcal{P}_j$. Furthermore, it is assumed that $\alpha_1 = \ldots = \alpha_K = 1$, which is equivalent to the uniform distribution in $\mathcal{P}_{K-1}$. In the original implementation of BitSeq (Glaus *et al.*, 2012), MCMC samples are drawn from the posterior distribution of $\boldsymbol{\theta}|\mathbf{x}$ by using the Gibbs sampler whereas more recently variational Bayes approximations have also been included for faster inference (Papastamoulis *et al.*, 2014; Hensman *et al.*, 2015).

Given the output of BitSeq stage 1 for two different samples, BitSeq stage 2 implements a one-sided test, probability of positive log-ratio, PPLR, for DE analysis. However, this approach does not define transcripts as differentially expressed or non-differentially expressed and is therefore not directly comparable with standard two-sided tests that are available in most other packages (Trapnell *et al.*, 2013; Leng *et al.*, 2013). Also, correlations between transcripts in the posterior distribution for each sample are discarded during the DE stage, leading to potential loss of accuracy when making inferences. To deal with these limitations, a new method for performing DE analysis is presented next.

2.2. *cjBitSeq*

Assume that we have at hand two samples $\mathbf{x} := (x_1, \ldots, x_r)$ and $\mathbf{y} := (y_1, \ldots, y_s)$ denoting the number of (mapped) reads for sample $\mathbf{x}$ and $\mathbf{y}$ respectively. Now, let θ_k and w_k denote the unknown relative abundance of transcript $k = 1, \ldots, K$ in sample $\mathbf{x}$ and $\mathbf{y}$ respectively. Define the parameter vector of relative abundances as $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{K-1}; \theta_K) \in \mathcal{P}_{K-1}$ and $\mathbf{w} = (w_1, \ldots, w_{K-1}; w_K) \in \mathcal{P}_{K-1}$. Under the standard BitSeq model the prior on the parameters $\boldsymbol{\theta}$ and $\mathbf{w}$ would be a product of independent Dirichlet distributions. In this case the probability $\theta_k = w_k$ under the prior is 0 and it is not straightforward to define non-differentially expressed transcripts. To model DE we would instead like to identify instances where transcript expression has not changed between samples. Therefore, we introduce a non-zero probability for the event $\theta_k = w_k$. This leads us to define a new model with a non-independent prior for the parameters $\boldsymbol{\theta}$ and $\mathbf{w}$.

Definition 1 (state vector). Let $c := (c_1, \ldots, c_K) \in \mathcal{C}$, where $\mathcal{C}$ is the set defined by

(a) $c_k \in \{0, 1\}$, $k = 1, \ldots, K$,
(b) $c_+ := \Sigma_{k=1}^{K} c_k \neq 1$.

Then, for $k = 1, \ldots, K$ let

$$\begin{cases} \theta_k = w_k, & \text{if } c_k = 0, \\ \theta_k \neq w_k, & \text{if } c_k = 1. \end{cases}$$

We shall refer to vector c as the state vector of the model.

For example, assume that $K = 6$ and $c = (1, 0, 0, 1, 0, 1)$. According to definition 1, $\theta_k = w_k$ for $k = 2, 3, 5$ and $\theta_k \neq w_k$ for $k = 1, 4, 6$. From definition 1 it is obvious that the sum of the elements in c cannot be equal to 1 because either all θs must be equal to ws, or at least two of them must be different. The introduction of such dependences between the elements of $\boldsymbol{\theta}$ and $\mathbf{w}$ has non-trivial effects on the prior assumptions of course. It is clear that with this approach we should define a valid conditional prior distribution for $\boldsymbol{\theta}, \mathbf{w}|c$.

First we impose a prior assumption on c. We shall consider the Jeffreys (Jeffreys, 1946) prior distribution for a Bernoulli trial, i.e. $P(c_k = 1|\pi) = \pi$ with π following a beta distribution. Since $c_+ \neq 1$, the prior distribution of the state vector c is expressed as

$$\pi \sim \text{beta}(\tfrac{1}{2}, \tfrac{1}{2}), \tag{2}$$

$$P(c|\pi) = P(c|c_+ \neq 1, \pi) = \frac{\pi^{c_+}(1-\pi)^{K-c_+}}{1-K\pi(1-\pi)^{K-1}}, \qquad c \in \mathcal{C}. \tag{3}$$

Next we proceed to the definition of a proper prior structure for the weights of the mixture. At this step extra care should be taken for everything to make sense as a probabilistic space. It is obvious that $(\boldsymbol{\theta}, \mathbf{w})$ should be defined conditionally on the state vector c. What it is less obvious is that $(\boldsymbol{\theta}, \mathbf{w})$ should be defined conditionally on a parameter of varying dimension. At this point, we introduce some extra notation.

Definition 2 (dead and alive subsets and permutation of the labels). For a given state vector c, define the order-specific subsets

$$C_0(c) := \{\tau_1 < \ldots < \tau_{K-c_+} \in \{1, \ldots, K\} : c_{\tau_k} = 0 \quad \forall\, k = 1, \ldots, K - c_+\}$$

and

$$C_1(c) := \{\tau_{K-c_++1} < \ldots < \tau_K \in \{1, \ldots, K\} : c_{\tau_k} = 1 \quad \forall\, k = K - c_+ + 1, \ldots, K\}.$$

These sets will be called dead and alive subsets of the transcriptome index respectively. Moreover, $\tau = (\tau_1, \ldots, \tau_K)$ denotes the unique permutation of $\{1, \ldots, K\}$ obeying the ordering within the dead and alive subsets.

As will be made clear later, it is convenient to define a unique labelling within the dead and alive subsets so we also explicitly define the corresponding permutation τ of the labels. To clarify definition 2, assume that $c = (1, 0, 0, 1, 0, 1)$. Then definition 2 implies that $C_0(c) = \{2, 3, 5\}$, $C_1(c) = \{1, 4, 6\}$ and $\tau = (2, 3, 5, 1, 4, 6)$. The order-specific definition of these subsets excludes $\{3, 2, 5\}$ (for example) from the definition of a dead subset.

It is clear that, if $C_0(c) = \emptyset$, then both $\boldsymbol{\theta}$ and $\mathbf{w}$ have $K - 1$ free parameters each. However, if $C_0(c) \neq \emptyset$, the free parameters lie in a lower dimensional space. This means that $(\boldsymbol{\theta}, \mathbf{w})$ should be defined given c by taking into account the set of free parameters that are actually allowed by the state vector. In particular, $(\boldsymbol{\theta}, \mathbf{w})$ are pseudoparameters. The actual parameters of our problem are defined in lemma 1.

In what follows, the notation $\tau\boldsymbol{\sigma}$ should be interpreted as the reordering of vector $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K)$ under permutation τ. For example, assume that $\tau = (3, 1, 2)$ and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$; then $\tau\boldsymbol{\sigma} = (\sigma_3, \sigma_1, \sigma_2)$. Let also τ^{-1} denote the inverse permutation of τ.

Lemma 1 (existence and uniqueness of free parameters). For every $(c, \tau, \boldsymbol{\theta}, \mathbf{w})$ respecting definitions 1 and 2 there is a unique set of free parameters:

$$(\mathbf{u}, \mathbf{v}) \in \mathcal{P}_{K-1} \times \mathcal{P}_{c_+-1}, \tag{4}$$

such that

$$\boldsymbol{\theta} = \tau^{-1}\mathbf{u}, \tag{5}$$

$$\mathbf{w} = \tau^{-1}\boldsymbol{\varpi}, \tag{6}$$

where $\boldsymbol{\varpi} = (\{u_{\tau_k^{-1}} : k \in C_0(c)\}, \mathbf{v}\Sigma_{k \in C_1(c)} u_{\tau_k^{-1}})$ under the conventions $\mathcal{P}_{-1} := \emptyset$ and $\emptyset\, \Sigma_{k \in \emptyset}\, u_k := \emptyset$.

Proof. It is trivial to show that $(c, \tau, \mathbf{u}, \mathbf{v}) \to (\boldsymbol{\theta}, \mathbf{w})$ is a 'one-to-one' and 'onto' mapping (bijective function).

For example, assume that $c = (1, 0, 0, 1, 0, 1)$, where $C_0(c) = \{2, 3, 5\}$ and $C_1(c) = \{1, 4, 6\}$. Then, $\tau = (2, 3, 5, 1, 4, 6)$ and $\tau^{-1} = (4, 1, 2, 5, 3, 6)$. According to state c we should have that

$\theta_2 = w_2$, $\theta_3 = w_3$ and $\theta_5 = w_5$, whereas $\theta_k \neq w_k$ for $k \in C_1(c)$. Lemma 1 states that $\boldsymbol{\theta}$ and $\mathbf{w}$ can be expressed as a transformation of two independent parameters: $\mathbf{u} = (u_1, u_2, u_3, u_4, u_5, u_6) \in \mathcal{P}_5$ and $\mathbf{v} = (v_1, v_2, v_3) \in \mathcal{P}_2$. According to equation (5), $\boldsymbol{\theta}$ is a permutation of the vector $\mathbf{u}$:

$$\boldsymbol{\theta}|(c, \mathbf{u}) = (u_4, u_1, u_2, u_5, u_3, u_6).$$

Next, $\mathbf{w}$ is obtained by a permutation of $\boldsymbol{\varpi}$, which is a linear transformation of $\mathbf{u}$ and $\mathbf{v}$, i.e. $\boldsymbol{\varpi} = (u_1, u_2, u_3, v_1(u_4 + u_5 + u_6), v_2(u_4 + u_5 + u_6), v_3(u_4 + u_5 + u_6))$. According to equation (6),

$$\mathbf{w}|(c, \mathbf{u}, \mathbf{v}) = (v_1(u_4 + u_5 + u_6), u_1, u_2, v_2(u_4 + u_5 + u_6), u_3, v_3(u_4 + u_5 + u_6)).$$

Comparing the last two expressions for $\boldsymbol{\theta}$ and $\mathbf{w}$, it is obvious that $\theta_2 = w_2$, $\theta_3 = w_3$ and $\theta_5 = w_5$, whereas $\theta_k \neq w_k$ for all remaining entries, which is the configuration that is implied by the state vector c. Note also that $\{u_{\tau_k^{-1}}; k \in C_0(c)\} = (u_1, \ldots, u_{K-c_+})$ and $\{u_{\tau_k^{-1}}; k \in C_1(c)\} = (u_{K-c_+ +1}, \ldots, u_K)$ and $\Sigma_{k \in C_1(c)} w_k = \Sigma_{k \in C_1(c)} \theta_k = \Sigma_{k \in C_1(c)} u_{\tau_k^{-1}}$.

Now, it should be clear that given a state vector c, as well as the independent free parameters $\mathbf{u}$ and $\mathbf{v}$, the pseudoparameters $\boldsymbol{\theta}$ and $\mathbf{w}$ are deterministically defined. In other words, the conditional distributions of $\boldsymbol{\theta}$ and $\mathbf{w}$ are Dirac distributions, gathering all their probability mass into the single points defined by equations (5) and (6). Hence, the conditional prior distribution for transcript expression is written as

$$f(\boldsymbol{\theta}, \mathbf{w}|c, \tau, \mathbf{u}, \mathbf{v}) = \mathbf{1}_{\boldsymbol{\theta}, \mathbf{w}}[\{\boldsymbol{\theta}(c, \tau, \mathbf{u}), \mathbf{w}(c, \tau, \mathbf{u}, \mathbf{v})\}], \tag{7}$$

with $\boldsymbol{\theta}(c, \tau, \mathbf{u})$ and $\mathbf{w}(c, \tau, \mathbf{u}, \mathbf{v})$ as in equations (5) and (6) respectively.

Moreover, we stress that, if the permutation τ were not uniquely defined according to definition 2, then we would have had to take into account all the possible permutations within the dead and alive subsets. However, such an approach would lead to an increased modelling complexity without making any difference on the inference. That said, the conditional prior distribution of τ given c is Dirac:

$$f(\tau|c) = \mathbf{1}_{\tau}\{\tau(c)\}, \tag{8}$$

where $\tau(c)$ denotes the unique permutation (given c) in definition 2.

At this point we state our prior assumptions for the free parameters, given a state vector c. We assume that *a priori* $\mathbf{u}$ and $\mathbf{v}$ are independent random variables distributed according to a Dirichlet distribution, i.e.

$$\mathbf{u}|c \sim \mathcal{D}_{K-1}(\alpha_1, \ldots, \alpha_K), \tag{9}$$

$$\mathbf{v}|c \sim \mathcal{D}_{c_+ - 1}(\gamma_1, \ldots, \gamma_{c_+}). \tag{10}$$

In the applications, we shall furthermore assume that $\alpha_k = 1$ for all $k = 1, \ldots, K$ and $\gamma_l = 1$ for all $l = 1, \ldots, c_+$, to assign a uniform prior distribution over $\mathcal{P}_{K-1} \times \mathcal{P}_{c_+ - 1}$. Now, the following theorem holds.

Theorem 1. Assume that distributions (9) and (10) hold true and furthermore $\alpha_k = \gamma_k = \alpha$ for all $k = 1, \ldots, K$. Then, $\boldsymbol{\theta}$ and $\mathbf{w}$ are marginally identical random variables following the $\mathcal{D}_{K-1}(\alpha, \ldots, \alpha)$ distribution.

For a proof of theorem 1, see appendix C in the on-line supplementary material.

Note here that theorem 1 does not imply that $\boldsymbol{\theta}$ and $\mathbf{w}$ are *a priori* independent. As shown in Fig. 1, θ_k is exactly equal to w_k with probability $P(c_k = 0) > 0$, $k = 1, \ldots, K$.

The model definition is completed by considering the latent allocation variables of the mixture model. Let $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_r\}$ and $\mathbf{z} = \{z_1, \ldots, z_s\}$ with

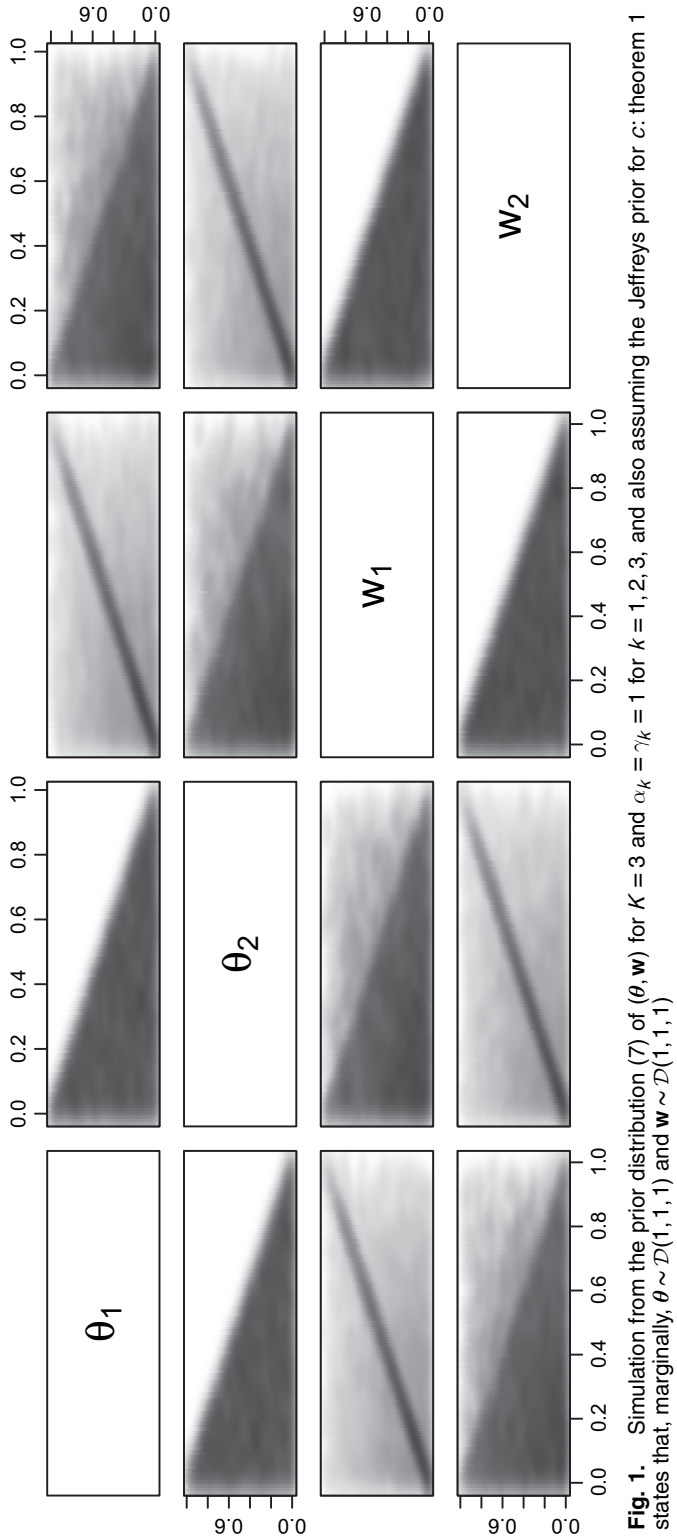

Fig. 1. Simulation from the prior distribution (7) of $(\boldsymbol{\theta}, \mathbf{w})$ for $K = 3$ and $\alpha_k = \gamma_k = 1$ for $k = 1, 2, 3$, and also assuming the Jeffreys prior for c: theorem 1 states that, marginally, $\boldsymbol{\theta} \sim \mathcal{D}(1, 1, 1)$ and $\mathbf{w} \sim \mathcal{D}(1, 1, 1)$

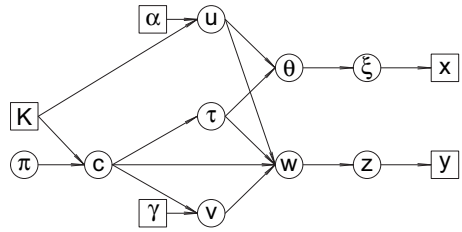

Fig. 2. Directed acyclic graph representation of the hierarchical model (12)

$$P(\xi_i = k|\boldsymbol{\theta}) = \theta_k, \qquad \text{independent for } i = 1, \ldots, r,$$
$$P(z_j = k|\mathbf{w}) = w_k, \qquad \text{independent for } j = 1, \ldots, s,$$

for $k = 1, \ldots, K$. Moreover, $\boldsymbol{\xi}$ and $\mathbf{z}$ are assumed conditionally independent given $\boldsymbol{\theta}$ and $\mathbf{w}$, i.e. $P(\boldsymbol{\xi}, \mathbf{z}|\boldsymbol{\theta}, \mathbf{w}) = P(\boldsymbol{\xi}|\boldsymbol{\theta})\, P(\mathbf{z}|\mathbf{w})$. Now, the joint distribution of the complete data $(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}, \mathbf{z})$ factorizes as follows:

$$f(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}, \mathbf{z}|\boldsymbol{\theta}, \mathbf{w}) = \prod_{i=1}^{r} \theta_{\xi_i} f_{\xi_i}(x_i) \prod_{j=1}^{s} w_{z_j} f_{z_j}(y_j). \tag{11}$$

Let $\mathbf{g} = (\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}, \mathbf{z}, \boldsymbol{\theta}, \mathbf{w}, \mathbf{u}, \mathbf{v}, c, \tau, \pi)$. From equations (2), (3) and (7)–(11), the joint distribution of $\mathbf{g}$ is defined as

$$f(\mathbf{g}|\boldsymbol{\alpha}, \gamma, K) = f(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}, \mathbf{z}|\boldsymbol{\theta}, \mathbf{w})\, f(\mathbf{u}|\boldsymbol{\alpha}, K)\, f(\mathbf{v}|c, \gamma)\, f(\boldsymbol{\theta}|\tau, \mathbf{u})\, f(\mathbf{w}|c, \tau, \mathbf{u}, \mathbf{v})\, f(\tau|c)\, f(c|K, \pi)\, f(\pi). \tag{12}$$

Equation (12) defines a hierarchical model whose graphical representation is given in Fig. 2 with circles and squares denoting respectively unobserved and observed or known variables.

2.3. Full conditional distributions for the Gibbs updates

In this section, the full conditional distributions are derived. Let $h|\cdots$ denote the conditional distribution of a random variable h given the values of the rest of the variables. We also denote by $\mathbf{x}_{[-i]}$ all remaining members of a generic vector after excluding its ith item.

It is straightforward to show that $\pi|\cdots \sim \text{beta}(c_+ + \frac{1}{2}, K - c_+ + \frac{1}{2})$. For the allocation variables it follows that

$$P(\xi_i = k|\cdots) \propto \theta_k f_k(x_i), \qquad k = 1, \ldots, K, \tag{13}$$

$$P(z_j = k|\cdots) \propto w_k f_k(y_i), \qquad k = 1, \ldots, K \tag{14}$$

independent for $i = 1, \ldots, r$ and $j = 1, \ldots, s$. Now, given $(\mathbf{u}, \mathbf{v}, c, \tau)$, it is again trivial to see that the full conditional distribution of $\boldsymbol{\theta}, \mathbf{w}|\cdots$ is the same as in equation (7). Let $\text{GD}(\cdot, \cdot)$ denote the generalized Dirichlet distribution (see appendix B in the on-line supplementary material) and also define

$$s_k(\boldsymbol{\xi}) := \sum_{i=1}^{r} I(\xi_i = k),$$
$$s_k(\mathbf{z}) := \sum_{j=1}^{s} I(z_j = k)$$

for $k=1,\ldots,K$. Regarding the full conditional distribution of the free parameters, we have the following result.

Lemma 2. The full conditional distribution of $(\mathbf{u},\mathbf{v}|\cdots)$ is

$$\mathbf{u}|\cdots\sim \mathrm{GD}(\lambda_1,\ldots,\lambda_{K-1};\beta_1,\ldots,\beta_{K-1}), \tag{15}$$

$$\mathbf{v}|\cdots\sim \mathcal{D}_{c_+-1}[\{\gamma_l+s_{\tau_{l+k^*}}(\mathbf{z});l=1,\ldots,c_+\}], \tag{16}$$

with $k^*:=K-c_+$, conditionally independent (given all other variables), where

$$\lambda_k:=\begin{cases}\alpha_k+s_{\tau_k}(\boldsymbol{\xi})+s_{\tau_k}(\mathbf{z}), & k=1,\ldots,k^*,\\ \alpha_k+s_{\tau_k}(\boldsymbol{\xi}), & k=k^*+1,\ldots,K-1\end{cases}$$

and

$$\beta_k:=\begin{cases}\sum\limits_{j=k+1}^{K}\{\alpha_j+s_{\tau_j}(\boldsymbol{\xi})+s_{\tau_j}(\mathbf{z})\}, & k=1,\ldots,k^*,\\ \sum\limits_{j=k+1}^{K}\{\alpha_j+s_{\tau_k}(\boldsymbol{\xi})\}, & k=k^*+1,\ldots,K-1.\end{cases}$$

For a proof of lemma 2, see appendix D in the on-line supplementary material.

Here, we underline that we have essentially derived an alternative construction of the generalized Dirichlet distribution. Assuming that two vectors of weights share some common elements, and independent Dirichlet prior distributions are assigned to the free parameters of these weights, the posterior distribution of the first free-parameter vector is a generalized Dirichlet distribution. Finally, note that, if $\mathbf{v}=\emptyset$ (this is the case when the corresponding elements of the weights of the two mixtures are all equal to each other), the generalized Dirichlet distribution (15) reduces to the distribution $\mathcal{D}_{K-1}[\{\alpha_k+s_k(\boldsymbol{\xi})+s_k(\mathbf{z});k=1,\ldots,K\}]$, as expected, since in such a case $(\mathbf{x},\mathbf{y})$ forms a random sample of size $r+s$ from the same population. However, if all weights are different, the full conditional distribution of $\mathbf{u}$ and $\mathbf{v}$ becomes a product of two independent Dirichlet distributions, as expected. Next we show that we can integrate out the parameters that are related to transcript expression and directly sample from the marginal posterior distribution of $\boldsymbol{\xi},\mathbf{z},c|\mathbf{x},\mathbf{y}$.

Theorem 2. Integrating out the transcript expression parameters $\mathbf{u}$ and $\mathbf{v}$, the full conditional distributions of allocation variables are written as

$$\begin{aligned}f(\boldsymbol{\xi},\mathbf{z}|\mathbf{x},\mathbf{y},c)\propto{}&\frac{\Gamma\left\{\sum\limits_{k\in C_1}\tilde{\alpha}_k+s_k(\boldsymbol{\xi})+s_k(\mathbf{z})\right\}}{\Gamma\left\{\sum\limits_{k\in C_1}\tilde{\alpha}_k+s_k(\boldsymbol{\xi})\right\}\Gamma\left\{\sum\limits_{k\in C_1}\gamma_{l(k)}+s_k(\mathbf{z})\right\}}\\ &\times\prod_{k\in C_1}\Gamma\{\tilde{\alpha}_k+s_k(\boldsymbol{\xi})\}\Gamma\{\gamma_{l(k)}+s_k(\mathbf{z})\}\\ &\times\prod_{k\in C_0}\Gamma\{\tilde{\alpha}_k+s_k(\boldsymbol{\xi})+s_k(\mathbf{z})\}\prod_{i=1}^{r}f_{\xi_i}(x_i)\prod_{j=1}^{s}f_{z_j}(y_j),\end{aligned} \tag{17}$$

$$P(\xi_i=k|\boldsymbol{\xi}_{[-i]},\mathbf{z},c,\mathbf{x})\propto\begin{cases}\{\tilde{\alpha}_k+s_k^{(i)}(\boldsymbol{\xi})+s_k(\mathbf{z})\}f_k(x_i), & k\in C_0,\\ \dfrac{\sum\limits_{t\in C_1}\tilde{\alpha}_k+s_t^{(i)}(\boldsymbol{\xi})+s_t(\mathbf{z})}{\sum\limits_{t\in C_1}\tilde{\alpha}_t+s_t^{(i)}(\boldsymbol{\xi})}\{\tilde{\alpha}_k+s_k^{(i)}(\boldsymbol{\xi})\}f_k(x_i), & k\in C_1,\end{cases} \tag{18}$$

Table 1. Workflow for the two samplers

RJMCMC sampler	*Collapsed sampler*
(a) Update $(\boldsymbol{\xi}, \mathbf{z})\|\boldsymbol{\theta}, \mathbf{w}$	(a) Update $\xi_i\|\boldsymbol{\xi}_{[-i]}, \mathbf{z}, \mathbf{c}$, $i = 1, \ldots, r$
(b) Update $(\mathbf{u}, \mathbf{v})\|c, \boldsymbol{\xi}, \mathbf{z}$	(b) Update $z_j\|\boldsymbol{\xi}, \mathbf{z}_{[-j]}, c$, $j = 1, \ldots, s$
(c) Update $(\boldsymbol{\theta}, \mathbf{w})\|c, \tau, \mathbf{u}, \mathbf{v}$	(c) Update a block of $c\|\boldsymbol{\xi}, \mathbf{z}$
(d) Propose update of $(c, \tau, \mathbf{v})\|\ldots$	(d) Update $\pi\|c$
(e) Update $\pi\|c$	(e) Update $(\boldsymbol{\theta}, \mathbf{w}, \tau, \mathbf{u}, \mathbf{v})\|c, \boldsymbol{\xi}, \mathbf{z}$ (optional)

$$P(z_j = k|\mathbf{z}_{[-j]}, \boldsymbol{\xi}, c, \mathbf{y}) \propto \begin{cases} \{\tilde{\alpha}_k + s_k(\boldsymbol{\xi}) + s_k^{(j)}(\mathbf{z})\} f_k(y_j), & k \in C_0, \\ \dfrac{\sum_{t \in C_1} \tilde{\alpha}_t + s_t(\boldsymbol{\xi}) + s_t^{(j)}(\mathbf{z})}{\sum_{t \in C_1} \gamma_{l(t)} + s_t^{(j)}(\mathbf{z})} \{\gamma_{l(k)} + s_k^{(j)}(\mathbf{z})\} f_k(y_j), & k \in C_1, \end{cases} \tag{19}$$

where $\tilde{\alpha}_k = \alpha_{\tau_k^{-1}}$, $l(k) = \tau_k^{-1} - k^*$, $s_k^{(i)}(\boldsymbol{\xi}) = \Sigma_{t \neq i} I(\xi_i = k)$ and $s_k^{(j)}(\mathbf{z}) = \Sigma_{t \neq j} I(z_i = k)$ for $k = 1, \ldots, K$, $i = 1, \ldots, r$ and $j = 1, \ldots, s$.

For a proof of theorem 2, see appendix E in the on-line supplementary material.

Once again, note the intuitive interpretation of our model in the special cases where $C_0 = \emptyset$ or $C_1 = \emptyset$. If $C_0 = \emptyset$ (all transcripts are differentially expressed) then the denominator in the first line of equation (17) becomes equal to $\Gamma(\Sigma_k \alpha_k + r + s)$, i.e. independent of $\boldsymbol{\xi}$ and $\mathbf{z}$. Hence, equation (17) reduces to the conditional distribution of the allocation variables when independent Dirichlet prior distributions are imposed on the mixture weights. In contrast, when $C_1 = \emptyset$ (all transcripts are equally expressed), the distribution reduces to the product appearing in the last row of equation (17). This is the marginal distribution of the allocations when considering that $(\mathbf{x}, \mathbf{y})$ arise from the same population and after imposing a Dirichlet prior on the weights, as expected.

2.4. Markov chain Monte Carlo samplers

In this section we consider the problem of sampling from the posterior distribution of model (12). We propose two (alternative) MCMC sampling schemes, depending on whether the trans-dimensional random variable $\mathbf{v}$ is updated before or after c.

Given c everything has fixed dimension. However, as c varies on the set of its possible values, then $\mathbf{v} \in \cup_{k \in \{0,2,\ldots,K\}} \mathcal{P}_{k-1}$. This means that, whenever c is updated, $\mathbf{v}$ should change dimension. To construct a sampler that switches between different dimensions, an RJMCMC method (Green, 1995) can be implemented (see also Richardson and Green (1997) and Papastamoulis and Iliopoulos (2009)). However, this step can be avoided since we have already shown that the transcript expression parameters can be integrated out. Thus, a collapsed sampler is also available. Given an initial state, the general workflow for the samplers proposed is shown in Table 1 (we avoid explicitly stating that all distributions appearing in Table 1 are conditionally defined on the observed data $\mathbf{x}$ and $\mathbf{y}$, although they should be understood as such).

Note that step (e) is optional for the collapsed sampler. It is implemented only to derive the estimates of transcript expression but it is not necessary for the previous steps. The next paragraphs outline the workflow for step (d) of the RJMCMC sampler and step (c) of the collapsed sampler. For full details the reader is referred to appendices F and G in the on-line supplementary material.

(a) *Reversible jump sampler*: models of different dimensions are bridged by using two move types, namely 'birth' and 'death' of an index. The effect of a birth or death move is respectively to increase or decrease the number of differentially expressed transcripts. These moves are complementary in the sense that the one is the reverse of the other. Note that this step proposes a candidate state which is accepted according to the acceptance probability.

(b) *Collapsed sampler*: in this case we randomly choose two transcripts (j_1 and j_2) and perform an update from the conditional distribution $c_{j_1,j_2}|c_{-[j_1,j_2]}\boldsymbol{\xi}, \mathbf{z}, \mathbf{x}, \mathbf{y}, \pi$, which is detailed in equations (G.1)–(G.4) in section G of the on-line supplementary material. The random selection of the block $\{j_1, j_2\} \subseteq \{1, \ldots, K\}$ and the corresponding update of c_{j_1,j_2} from its full conditional distribution is a valid MCMC step because it corresponds to a Metropolis–Hastings step in which the acceptance probability equals 1 (see lemma 2 in appendix G of the supplementary material).

2.5. Clustering of reads and transcripts

In real RNA-seq data sets the number of transcripts could be very large. This imposes a great obstacle for the practical implementation of the approach proposed: the search space of the MCMC sampler consists of 2^K elements (state vectors) and convergence of the sampler may be very slow. This problem can be alleviated by a cluster representation of aligned reads to the transcriptome. High quality mapped reads exhibit a sparse behaviour in terms of their mapping places: each read aligns to a small number of transcripts and there are groups of reads mapping to specific groups of transcripts. Hence, we can take advantage of this sparse representation of alignments and break the initial problem into simpler problems, by performing MCMC sampling per cluster.

This clustering representation introduces an efficient way to perform parallel MCMC sampling by using multiple threads for transcript expression estimation. For this purpose we used the GNU parallel (Tange, 2011) tool, which effectively handles the problem of splitting a series of jobs (MCMC sampling per cluster) into the available threads. The jobs are ordered according to the number of reads per cluster and those containing more reads are queued first. GNU parallel efficiently spawns a new process when one finishes and keeps all available central processor units active, thus saving time compared with an arbitrary assignment of the same amount of jobs to the same number of available threads. For further details see the on-line appendix H.

2.6. False discovery rate

Controlling the FDR (Benjamini and Hochberg, 1995; Storey, 2003) is a crucial issue in multiple-comparisons problems. Under a Bayesian perspective, any probabilistic model that defines a positive prior probability for DE and expression estimation yields that $\mathbb{E}(\text{FDR}|\text{data}) = \Sigma\{1 - \hat{P}(c_k = 1|\mathbf{x}, \mathbf{y})\}d_k/D$ (see for example Müller *et al.* (2004, 2006)), where $d_k \in \{0, 1\}$ and $D = \Sigma d_k$ denote the decision for transcript k, $k = 1, \ldots, K$, and the total number of rejections respectively. Consequently, the FDR can be controlled at a desired level α by choosing the transcripts that $\hat{P}(c_k = 1|\mathbf{x}, \mathbf{y}) > 1 - \alpha$, which is also the approach that was proposed by Leng *et al.* (2013). We have found that this rule achieves small FDRs compared with the desired level α, but sometimes results in a small true positive rate.

A less conservative choice is as follows. Let $q_1 \geqslant \ldots \geqslant q_K$ denote the ordered values of $\hat{P}(c_k = 1|\mathbf{x}, \mathbf{y})$, $k = 1, \ldots, K$, and define $G_k := \Sigma_{j=1}^{k}(1 - q_k)/k$, $k = 1, \ldots, K$. For any given $0 < \alpha < 1$, consider the decision rule

$$d_k = \begin{cases} 1, & 1 \leqslant k \leqslant g, \\ 0, & g+1 \leqslant k \leqslant K, \end{cases} \tag{20}$$

where $g := \max\{k = 1, \ldots, K : G_k \leqslant \alpha\}$. It is quite straightforward to see that expression (20) controls the expected FDR at the desired level α, since by direct substitution we have that

$$\mathbb{E}(\text{FDR}|\text{data}) = \frac{\sum_{k=1}^{K}\{1 - \hat{P}(c_k = 1|\mathbf{x}, \mathbf{y})\}d_k}{D} = \frac{\sum_{k=1}^{g}(1 - q_k)}{g} \leqslant \alpha.$$

An alternative is to use a rule optimizing the posterior expected loss of a predefined loss function. For example, the threshold $c/(c+1)$ is the optimal cut-off under the loss function $L = c\overline{\text{FD}} + \overline{\text{FN}}$, where $\overline{\text{FD}}$ and $\overline{\text{FN}}$ denote the posterior expected counts of false discoveries and false negative discoveries respectively. Note that L is an extension of the $(0, 1, c)$ loss functions for traditional hypothesis testing (Lindley, 1971), whereas a variety of alternative loss functions can be devised as discussed in Müller *et al*. (2004).

3. Results

A set of simulation studies is used to benchmark the proposed methodology by using synthetic RNA-seq reads from the *Drosophila melanogaster* transcriptome. The Spanki software (Sturgill *et al*., 2013) is used for this. In addition to the simulated data study we also perform a comparison for two real data sets: a low and high coverage sequencing experiment using human data and a data set from drosophila. In all cases, the reads are mapped to the reference transcriptome by using Bowtie (version 2.0.6), allowing up to 100 alignments per read. TopHat (version 2.0.9) is also used for Cufflinks.

3.1. Evaluation of samplers

We used a simulated data set from $K = 630$ transcripts (more details are described in the on-line appendix H) and compare the posterior mean estimates between short and long runs. As shown in Fig. 3, the collapsed sampler exhibits faster convergence than the RJMCMC sampler; hence in what follows we shall present only results corresponding to the collapsed sampler. The reader is referred to the on-line supplementary material (appendices J and K) for further comparisons (including auto-correlation function estimation and prior sensitivity) between our two MCMC schemes.

3.2. Simulated data

The input of the Spanki simulator is a set of reads per kilobase values per sample. This file is provided under a variety of generative scenarios. Given the input files, Spanki simulates RNA-seq reads (in 'fastq' format, a text-based format for storing nucleotide sequences with the corresponding quality scores) according to the specified reads per kilobase values. Seven scenarios are used to generate the data: two Poisson replicates per condition (scenario 1), three negative binomial replicates per condition (scenario 2), nine negative binomial replicates (scenario 3), three negative binomial replicates per condition with five times higher variability among replicates compared with scenario 2 (scenario 4) and the same variability as scenario 4 but a smaller range for the mean reads per kilobase values (scenario 5). The last two scenarios are revisions of the first scenario with smaller fold changes (scenario 6) and large differences in the number of reads between conditions (scenario 7). See the on-line supplementary Fig. 9 and appendix K for the details of the ground truth that was used in our simulations.

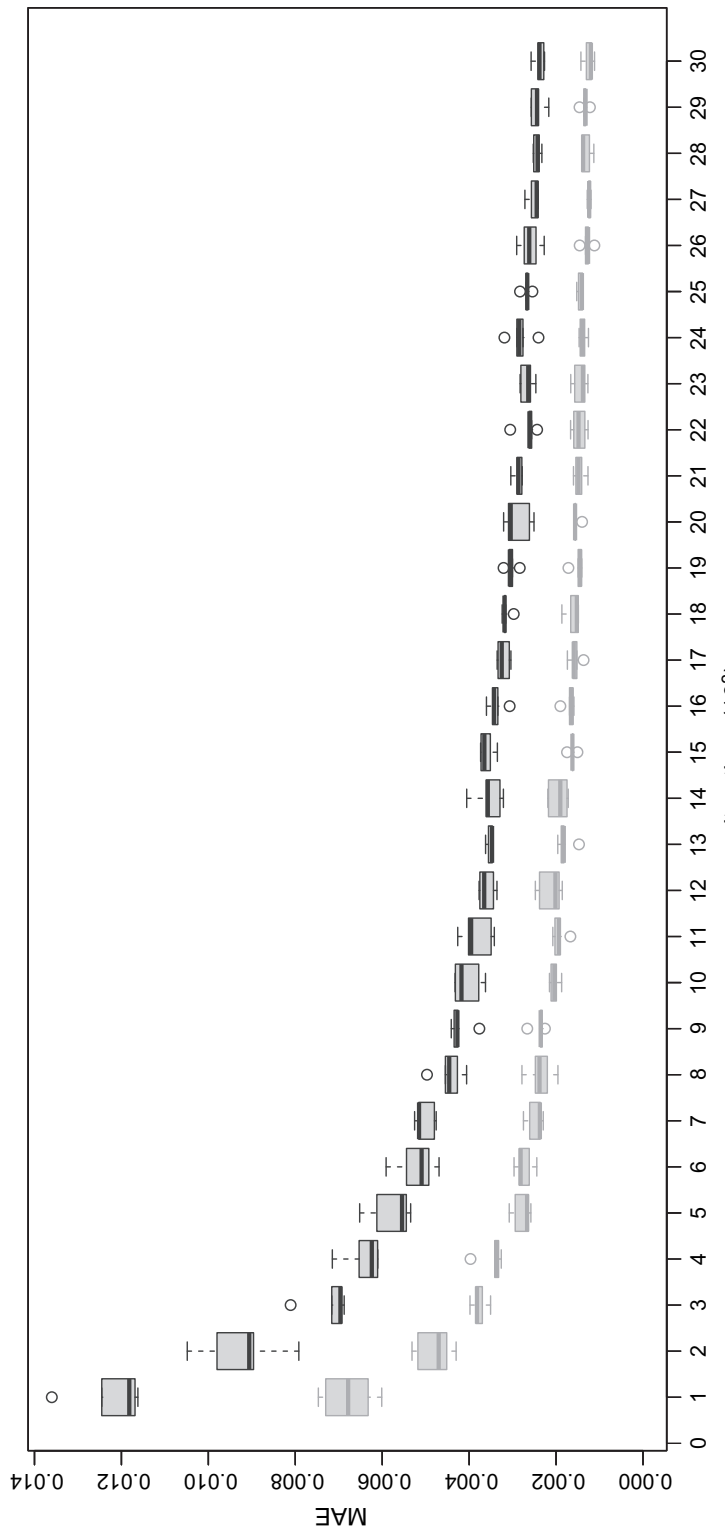

Fig. 3. Convergence of the ergodic means of posterior probabilities of DE for a toy example of $K = 630$ transcripts (the 'ground truth' for the posterior mean estimates $(\hat{P}_g(c_k=1);\ k=1,\ldots,K)$ of these probabilities was inferred by running each sampler for 500000 iterations; then, each sampler ran for a smaller number of m iterations resulting in the posterior mean estimates $\hat{P}_m(c_k=1),\ k=1,\ldots,K$, for $m = 1000, 2000, \ldots, 30000$; finally, the averaged mean absolute error of the posterior mean estimates was computed as $(1/K)\sum_{k=1}^{K}|\hat{P}_m(c_k=1) - \hat{P}_g(c_k=1)|$; the boxplots correspond to five replications of the previous procedure): ——, collapsed sampler; ——, RJMCMC sampler

Next, we applied the method proposed and compared our results against Bitseq, Cuffdiff and EBSeq, using

(a) the receiver operating characteristic,
(b) the squared error, accuracy receiver operating characteristic area measure, SAR (Sing *et al.*, 2005), and
(c) the power to achieved FDR curves, as shown in Fig. 4.

For the comparison in (c) the FDR decision of our model is based on rule (20). Moreover, only methods that control the FDR are taken into account in (c); hence BitSeq stage 2 is excluded. In addition to this FDR control procedure, we also provide adjusted rates after imposing a threshold to the log-fold change of the cjBitSeq sampler: all transcripts with estimated absolute $\log_2$-fold change less than 1 are filtered out (results correspond to the broken lines in Fig. 4). A typical behaviour of the methods compared is illustrated in Fig. 5, displaying true expression values used in scenario 3. We conclude that our method infers an almost ideal classification, which is not something that applies to the other methods despite the large number of replicates used.

To summarize our findings, Fig. 6 displays the complementary area under the curve for each scenario. Averaging across all simulation scenarios, we conclude that our method is almost twice as good as BitSeq stage 2, three times better than EBSeq and 3.2 times better than Cuffdif. Finally, we compare the estimated relative abundance of transcripts against the true values that were used to generate the data, using the average across all replicates of a given condition. Fig. 6(b) displays the mean absolute error between the logarithm of true transcript expression and the corresponding estimates according to each method. We see that cjBitSeq, BitSeq stage 1 and RSEM exhibit similar behaviour, and all perform significantly better than Cufflinks. Although there is no consistent ordering between the first three methods, averaging across all experiments we conclude that cjBitSeq is ranked first.

We have also tested the sensitivity of our method with respect to the prior distributions of DE (3) by setting $\pi = 0.5$ (see the on-line supplementary Fig. 11 and the corresponding discussion in appendix K). We conclude that the prior distribution does not affect the ranking of methods either for DE or expression estimation.

3.3. Human data

This example demonstrates the proposed algorithm to differential analysis of lung fibroblasts in response to loss of the developmental transcription factor HOXA1; see Trapnell *et al.* (2013) for full details. There are three biological replicates in the two conditions. The experiment is carried out by using two sequencing platforms: 'HiSeq' and 'MiSeq', where MiSeq produced only 23% of the number of reads in the HiSeq data. Here, these reads are mapped to hg19 (University of California, Santa Cruz, gerome browser annotation) using Bowtie 2, consisting of $K = 48\,009$ transcripts. In total, there are 96 969 106 and 21 271 542 mapped reads for HiSeq and MiSeq sequencers respectively. Trapnell *et al.* (2013) demonstrated the ability of Cuffdiff2 to recover the transcript dynamics from the HOXA1 knockdown when using the significantly smaller amount of data generated by MiSeq compared with HiSeq.

Applying cjBitSeq to the MiSeq data recovers 50.2% of the DE transcripts from HiSeq. In contrast, 183 transcripts are reported as differentially expressed with the MiSeq data but not the HiSeq data (Figs 7(a) and 7(b)). The corresponding percentages for BitSeq stage 2, EBSeq and Cuffdiff are 43.3%, 40.6% and 15.7% respectively (see Figs 7(b), 7(c) and 7(d)). We conclude that the model proposed returns the largest proportion of consistently differentially expressed transcripts between platforms. The number of transcripts which are simultaneously reported as differentially expressed is equal to 2173 and 390 for HiSeq and MiSeq data respectively (Figs

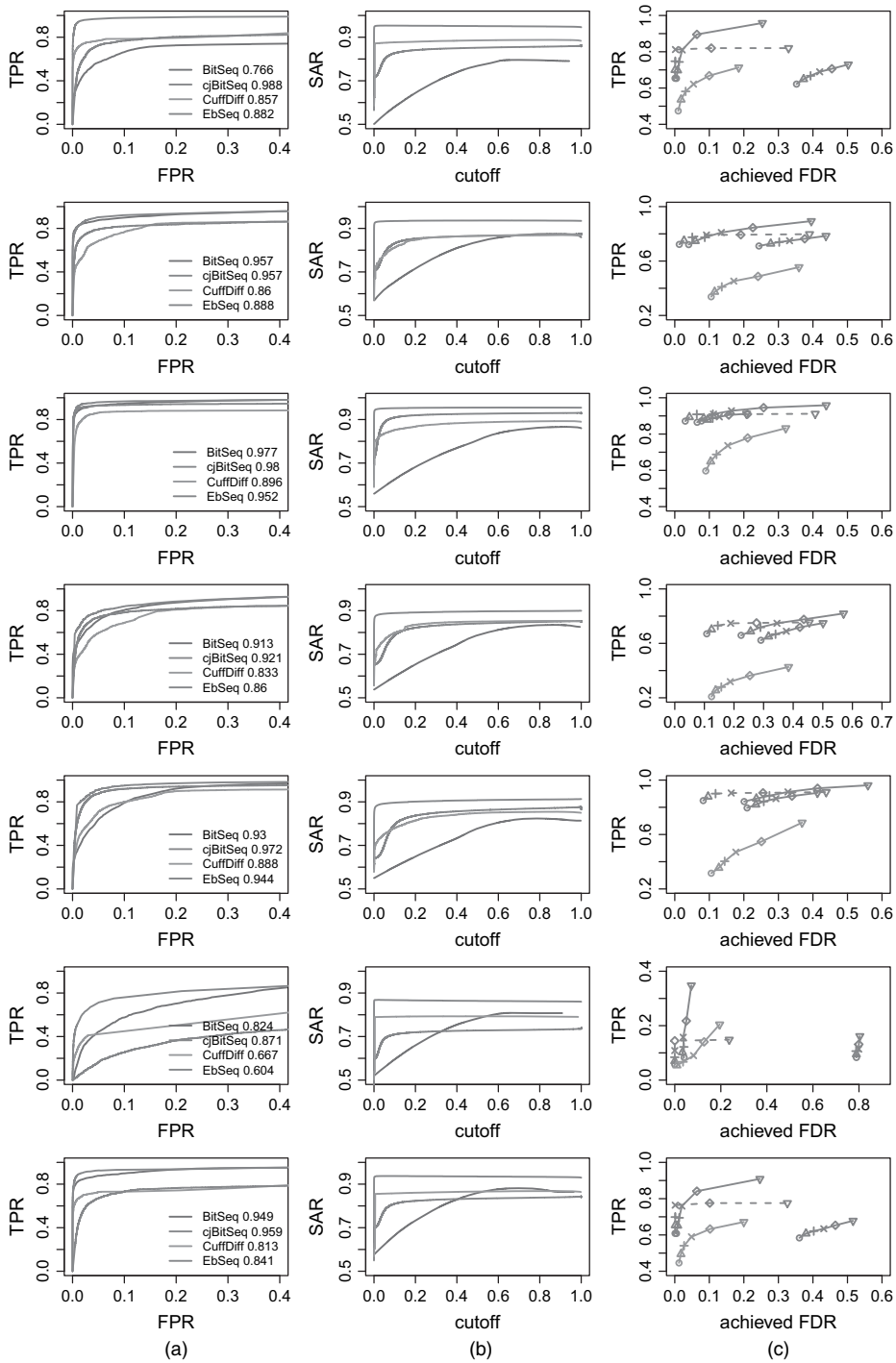

Fig. 4. (a) Receiver operating characteristic, (b) SAR-measure and (c) power to achieved FDR curves for scenarios 1–7 (from top to bottom): – – –, filtered cjBitSeq output by discarding transcripts with absolute $\log_2$-fold change less than 1; ○, eFDR = 0.01; △, eFDR = 0.025; +, eFDR = 0.05; ×, eFDR = 0.1; ◇, eFDR = 0.2; ▽, eFDR = 0.4

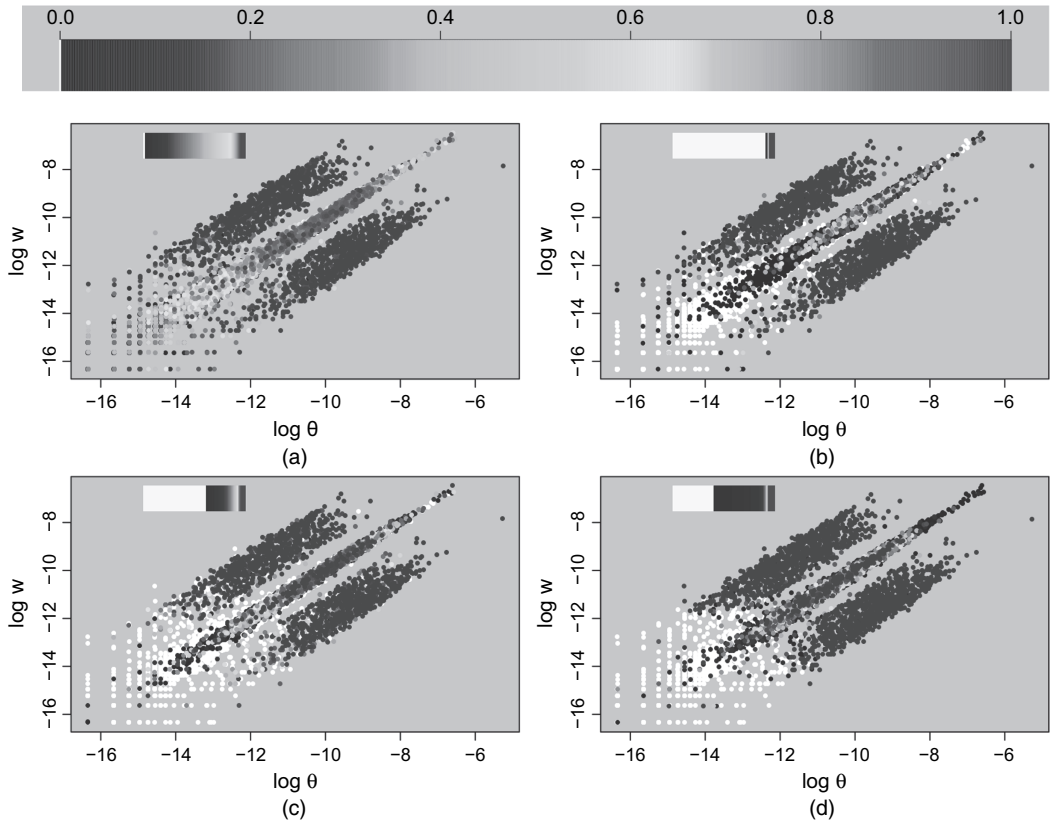

Fig. 5. True log-relative expression values for scenario 3 (average of nine replicates per condition: approximately 24 million reads in total) (the colour corresponds to the evidence of DE according to each method and the keys show the relative frequency of colours): (a) BitSeq; (b) cjBitSeq; (c) Cuffdiff; (d) EbSeq

8(a) and 8(b)). Finally, cjBitSeq and EBSeq provide the most highly correlated classifications (see Table 1 of the on-line supplementary material).

4. Discussion

We have proposed a probabilistic model for the simultaneous estimation of transcript expression and DE between conditions. Building on the BitSeq framework, the new Bayesian hierarchical model is conjugate for fixed dimension variables. A by-product is a new interpretation of the generalized Dirichlet distribution, which naturally appears in equation (15) as the full conditional distribution of a random variable describing one of the free parameters corresponding to two proportion vectors under the constraint that some of the weights are equal to each other. We implemented two MCMC samplers, a reversible jump and a collapsed Gibbs sampler, and we found that the collapsed Gibbs sampler converged faster. To reduce the dimensionality of the parameter space greatly for inference we developed a transcript clustering approach which allows inference to be carried out independently on subsets of transcripts that share aligned reads. According to lemma 3 in the on-line supplementary material (appendix H), this clustered version of the ordinary algorithm converges to the proper marginal distribution for each cluster. Thus, the algorithm has the nice property that it can be run in parallel for each cluster, and the memory requirements are quite low, providing a simple parallelization option.

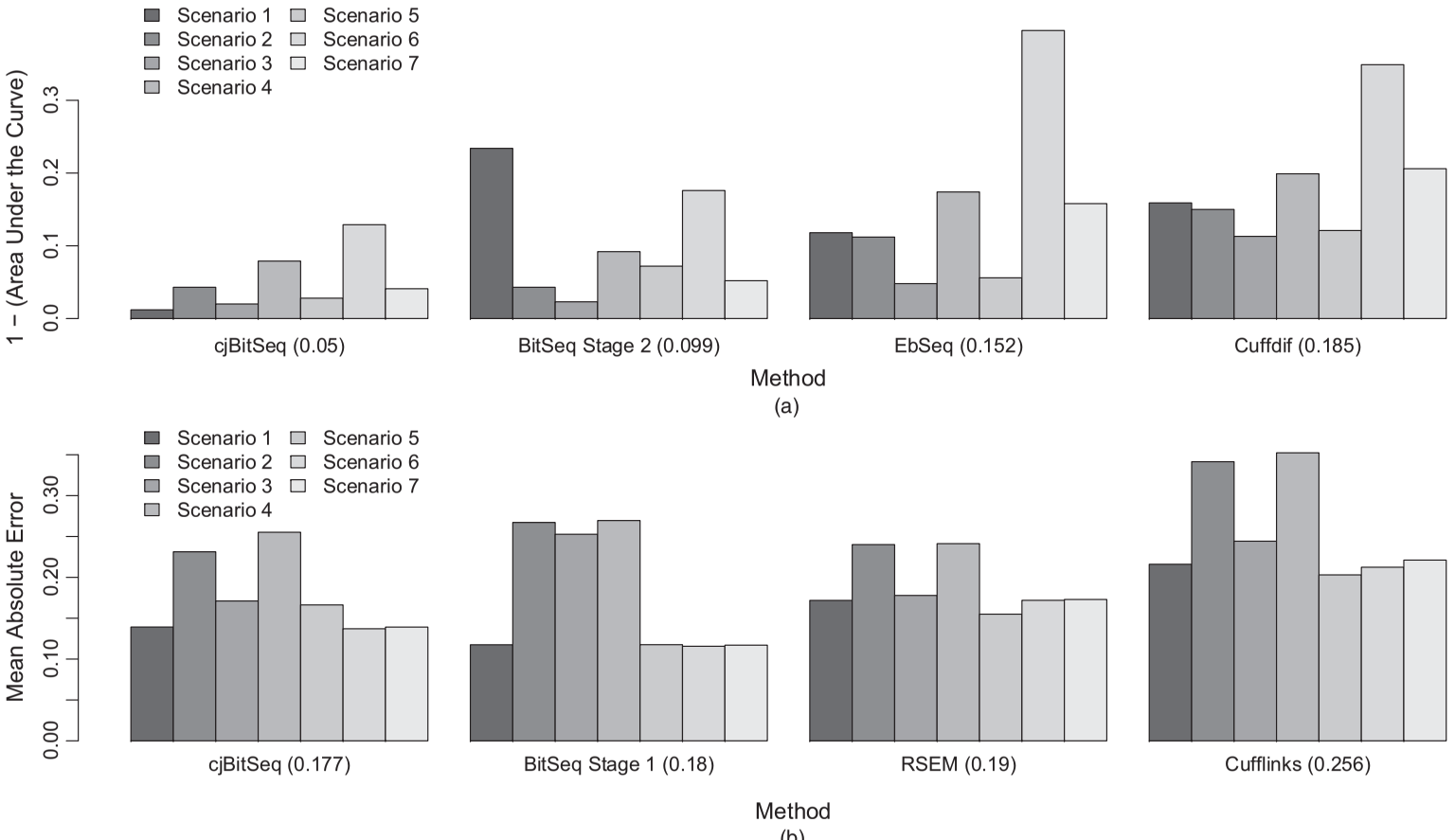

Fig. 6. Simulated data: ranking of methods with respect to estimation of (a) DE and (b) the logarithm of relative expression (the methods are ordered according to the averaged complementary area under the curve and mean absolute error (shown in parentheses))

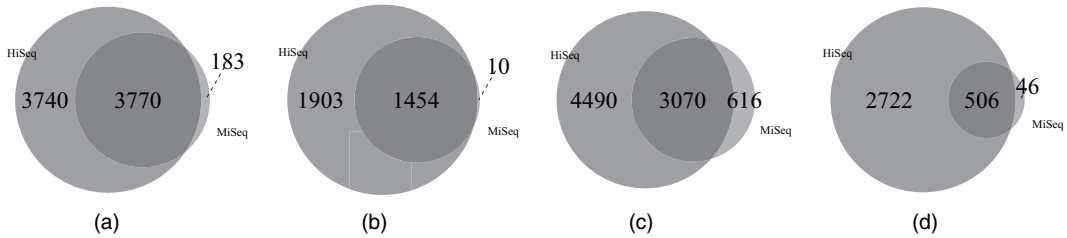

Fig. 7. HOXA1 knockdown data set: significant transcript list returned by (a) cjBitSeq (50.2%), (b) BitSeq (43.3%), (c) EBSeq (40.6%) and (d) Cuffdiff (15.7%) when using HiSeq (●) and MiSeq (●) data (the FDR for cjBitSeq, EBSeq and Cuffdiff were set to 0.05, whereas, for BitSeq, PPLR < 0.025 or PPLR > 0.975)

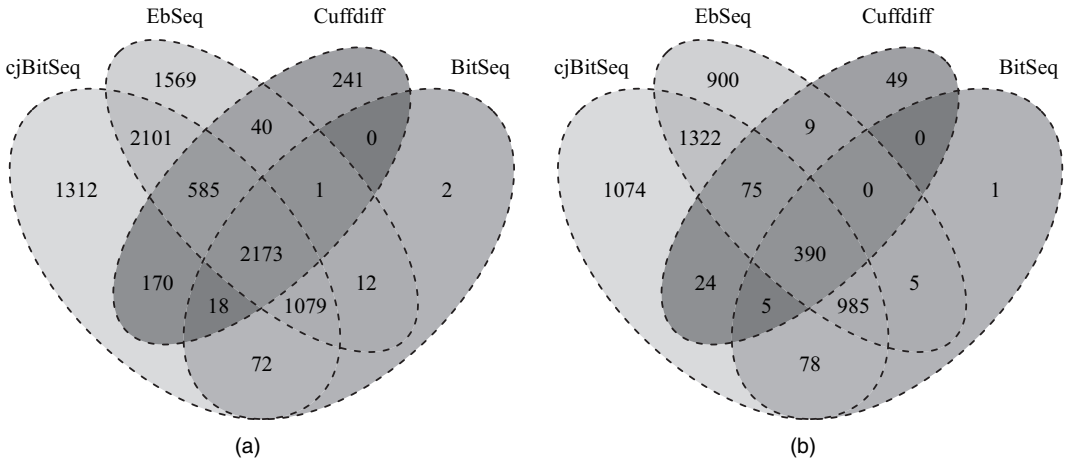

Fig. 8. HOXA1 knockdown data set: contiguity of methods when using (a) HiSeq and (b) MiSeq data (the FDR for cjBitSeq, EBSeq and Cuffdiff were set to 0.05, whereas, for BitSeq, PPLR < 0.025 or PPLR > 0.975)

The applications to simulated and real RNA-seq data reveal that the method proposed is highly competitive with the current state of the art software dealing with DE analysis at the transcript level. Note that the simulated data were generated under a variety of scenarios and including different levels of replication and biological variation. We simulated transcript reads per kilobase values with variability following either the Poisson or the negative binomial distribution with various levels for the dispersion around the mean. We conclude that our method is quite robust in expression estimation and in classifying transcripts as differentially expressed or not. Compared with standard two-stage pipelines it is ranked as the best method under a wide range of generative scenarios.

RNA-seq data are usually replicated such that more than one data set is available for each condition. In such a way, biological variability between repetitions of the same experiment can be taken into account. The amount of variability between replicates can be quite high depending on the experimental conditions. Two-stage approaches for estimating DE are strongly focused on modelling this interreplicate variability. This is not so for our method at present and all replicates of a given condition are effectively pooled before inference. Modelling the variability between replicates would significantly increase the complexity of our approach as it is technically challenging to retain conjugacy. However, according to our simulation studies, we have found that pooling replicates and jointly estimating expression and DE balances the loss through ignoring variability between replicates in many cases. Nevertheless, an extension also

to model interreplicate variability would be very interesting and could be expected to improve performance when there is high interreplicate dispersion.

The method proposed was developed focusing on a comparison of two conditions and its extension to more general settings is another interesting area for future research. A remarkable property of the parameterization that was introduced in equations (5) and (6) is that its extension is straightforward when $J > 2$: it can be shown that in this case there is one parameter of constant dimension and $J-1$ parameters of varying dimension. Let $\mathbf{u} = \mathbf{u}^{(1)}$ be the vector of relative abundances for condition 1. For a given condition $j = 2, \ldots, J$ define a vector $\mathbf{v}_j$ containing the expression of transcripts not being equal to any of the previous conditions $1, \ldots, j-1$. Note that $\mathbf{v}_j$ is a random variable with varying length (between 0 and K). Furthermore, for $j \geqslant 2$ define the vectors $\mathbf{u}_k^{(j)}$, $k = 1, \ldots, j-1$, containing the expression of transcripts that are shared with condition k but not with $1, \ldots, k-1$. It follows that $\mathbf{u}_k^{(j)}$ can be written as a function of $\mathbf{u}^{(1)}$ and $\mathbf{v}_k$, $k = 1, \ldots, j-1$. Hence, the relative transcript expression vector for condition j can be expressed as a suitable permutation of $(\mathbf{u}_1^{(j)}, \ldots, \mathbf{u}_{j-1}^{(j)}, \mathbf{v}_j)$. However, the question of whether the model stays conjugate for fixed dimension updates remains an open problem. If yes, the design of more sophisticated move types between different models would also be crucial to the convergence of the algorithm since the search space is increased.

The source code of the proposed algorithm is compiled for Linux distributions and it is available from `https://github.com/mqbssppe/cjBitSeq`. The simulation pipeline is available from `https://github.com/ManchesterBioinference/cjBitSeq_benchmarking`. Cluster discovery and MCMC sampling are coded in R and C++ respectively. Parallel runs of the MCMC scheme are implemented by using the GNU parallel (Tange, 2011) shell tool. The computing times that are needed for our data sets are reported in the on-line supplementary Table 2.

5. Supplementary material

In the on-line supplementary material we provide the proofs of our lemmas and theorems, a detailed description of the reversible jump proposal and the Gibbs updates of the state vector of the collapsed sampler. Also included are details of alignment probabilities and some useful properties of the generalized Dirichlet distribution. We also perform various comparisons between the RJMCMC and collapsed samplers and examine their prior sensitivity. Finally we describe the generative schemes for the simulation study and some guidelines for the practical implementation of the algorithm.

Acknowledgements

The research was supported by Medical Research Council award MR/M02010X/1, Biotechnology and Biological Sciences Research Council award BB/J009415/1 and European Union FP7 project RADIANT (grant 305626). The authors acknowledge the assistance given by IT Services and the use of the Computational Shared Facility at the University of Manchester. We also thank the Joint Editor and two reviewers for their helpful comments and suggestions which helped us to improve the paper.

References

Anders, A. and Huber, W. (2010) Differential expression analysis for sequence count data. *Genome Biol.*, **11**, article R106.

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

Connor, R. J. and Mosimann, J. E. (1969) Concepts of independence for proportions with a generalization of the Dirichlet distribution. *J. Am. Statist. Ass.*, **64**, 194–206.
Gelfand, A. and Smith, A. (1990) Sampling-based approaches to calculating marginal densities. *J. Am. Statist. Ass.*, **85**, 398–409.
Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattn Anal. Mach. Intell.*, **6**, 721–741.
Glaus, P., Honkela, A. and Rattray, M. (2012) Identifying differentially expressed transcripts from RNA-Seq data with biological variation. *Bioinformatics*, **28**, 1721–1728.
Green, P. J. (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711–732.
Gu, J., Wang, X., Halakivi-Clarke, L., Clarke, R. and Xuan, J. (2014) BADGE: a novel Bayesian model for accurate abundance quantification and differential analysis of RNA-Seq data. *BMC Bioinform.*, **15**, suppl. 9, article S6.
Hensman, J., Papastamoulis, P., Glaus, P., Honkela, A. and Rattray, M. (2015) Fast and accurate approximate inference of transcript expression from RNA-seq data. *Bioinformatics*, **31**, 3881–3889.
Jeffreys, H. (1946) An invariant form for the prior probability in estimation problems. *Proc. R. Soc. Lond.* A, **186**, 453–461.
Jia, C., Guan, W., Yang, A., Xiao, R., Tang, W. H. W., Moravec, C. S., Margulies, K. B., Cappola, T. P., Li, C. and Li, M. (2015) MetaDiff: differential isoform expression analysis using random-effects meta-regression. *BMC Bioinform.*, **16**, 1–12.
Langmead, B., Trapnell, C., Pop, M. and Salzberg, S. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**.
Leng, N., Dawson, J. A., Thomson, J. A., Ruotti, V., Rissman, A. I., Smits, B. M., Haag, J. D., Gould, M. N., Stewart, R. M. and Kendziorski, C. (2013) EBSeq: an empirical Bayes hierarchical model for inference in RNA-Seq experiments. *Bioinformatics*, **29**, 1035–1043.
Li, B. and Dewey, C. N. (2011) RSEM: accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinform.*, **12**, article 323.
Lindley, D. V. (1971) *Making Decisions*. New York: Wiley.
Mortazavi, A., Williams, B., McCue, K., Schaeffer, L. and Wold, B. (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Meth.*, **5**, 621–628.
Müller, P., Parmigiani, G. and Rice, K. (2006) FDR and Bayesian multiple comparisons rules. In *Proc. 8th Wrld Meet. Bayesian Statistics, Benidorm, June 1st–6th*.
Müller, P., Parmigiani, G., Robert, C. and Rousseau, J. (2004) Optimal sample size for multiple testing. *J. Am. Statist. Ass.*, **99**, 990–1001.
Nariai, N., Hirose, O., Kojima, K. and Nagasaki, M. (2013) TIGAR: transcript isoform abundance estimation method with gapped alignment of RNA-Seq data by variational Bayesian inference. *Bioinformatics*, **18**, 2292–2299.
Nicolae, M., Mangul, S., Mandoiu, I. and Zelikovsky, A. (2011) Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Alg. Molec. Biol.*, **6**, no. 1, article 9.
Papastamoulis, P., Hensman, J., Glaus, P. and Rattray, M. (2014) Improved variational Bayes inference for transcript expression estimation. *Statist. Applic. Genet. Molec. Biol.*, **13**, 213–216.
Papastamoulis, P. and Iliopoulos, G. (2009) Reversible jump MCMC in mixtures of normal distributions with the same component means. *Computnl Statist. Data Anal.*, **53**, 900–911.
Richardson, S. and Green, P. J. (1997) On Bayesian analysis of mixtures with an unknown number of components. *J. R. Statist. Soc.* B, **59**, 731–758; correction, **60** (1998), 661.
Robinson, M., McCarthy, D. and Smyth, G. (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.
Rossell, D., Attolini, C. S.-O., Kroiss, M. and Stocker, A. (2014) Quantifying alternative splicing from paired-end RNA-Sequencing data. *Ann. Appl. Statist.*, **8**, 309–330.
Sing, T., Sander, O., Beerenwinkel, N. and Lengauer, T. (2005) ROCR: visualizing classifier performance in R. *Bioinformatics*, **21**, 3940–3941.
Storey, J. D. (2003) The positive false discovery rate: a Bayesian interpretation and the q-value. *Ann. Statist.*, 2013–2035.
Sturgill, J., Malone, J. H., Sun, X., Smith, H. E., Rabinow, L., Samson, M. L. and Oliver, B. (2013) Design of RNA splicing analysis null models for post hoc filtering of drosophila head RNA-Seq data with the splicing analysis kit (Spanki). *BMC Bioinform.*, **14**, article 320.
Tange, O. (2011) GNU parallel—the command-line power tool. ;*login:*, **36**, 42–47.
Trapnell, C., Hendrickson, D. G., Sauvageau, M., Goff, L., Rinn, J. L. and Pachter, L. (2013) Differential analysis of gene regulation at transcript resolution with RNA-Seq. *Nat. Biotechnol.*, **31**, 46–53.
Trapnell, C., Pachter, L. and Salzberg, S. (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, **25**, 1105–1111.
Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J. and Pachter, L. (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.*, **28**, 511–515.

Wong, T. (1998) Generalized Dirichlet distribution in Bayesian analysis. *Appl. Math. Computn*, **97**, 165–181.
Wong, T. (2010) Parameter estimation for generalized Dirichlet distributions from the sample estimates of the first and the second moments of random variables. *Computnl Statist. Data Anal.*, **54**, 1756–1765.

Supporting information

Additional 'supporting information' may be found in the on-line version of this article:

'Supplementary material for the article: "A Bayesian model selection approach for identifying differentially expressed data from RNA-seq data"'.