Journal of
Computer Science & Systems Biology

Vavoulis and Gough, J Comput Sci Syst Biol 2013, 7:1
DOI: 10.4172/jcsb.1000131

Research Article Open Access

# Non-Parametric Bayesian Modelling of Digital Gene Expression Data

**Dimitrios V Vavoulis* and Julian Gough***

*Department of Computer Science, University of Bristol, Bristol, United Kingdom*

## Abstract

Next-generation sequencing technologies provide a revolutionary tool for generating gene expression data. Starting with a fixed RNA sample, they construct a library of millions of differentially abundant short sequence tags or "reads", which constitute a fundamentally discrete measure of the level of gene expression. A common limitation in experiments using these technologies is the low number or even absence of biological replicates, which complicates the statistical analysis of digital gene expression data. Analysis of this type of data has often been based on modified tests originally devised for analysing microarrays; both these and even *de novo* methods for the analysis of RNA-seq data are plagued by the common problem of low replication.

We propose a novel, non-parametric Bayesian approach for the analysis of digital gene expression data. We begin with a hierarchical model for modelling over-dispersed count data and a blocked Gibbs sampling algorithm for inferring the posterior distribution of model parameters conditional on these counts. The algorithm compensates for the problem of low numbers of biological replicates by clustering together genes with tag counts that are likely sampled from a common distribution and using this augmented sample for estimating the parameters of this distribution. The number of clusters is not decided *a priori*, but it is inferred along with the remaining model parameters. We demonstrate the ability of this approach to model biological data with high fidelity by applying the algorithm on a public dataset obtained from cancerous and non-cancerous neural tissues.

Source code implementing the methodology presented in this paper takes the form of the Python Package DGEclust, which is freely available at the following link: https://bitbucket.org/DimitrisVavoulis/dgeclust.

**Keywords:** DGEclust; Stick-breaking priors; Negative binomial distribution

## Introduction

It is a common truth that our knowledge in Molecular Biology is only as good as the tools we have at our disposal. Next-generation or high-throughput sequencing technologies provide a revolutionary tool in the aid of genomic studies by allowing the generation, in a relatively short time, of millions of short sequence tags, which reflect particular aspects of the molecular state of a biological system. A common application of these technologies is the study of the transcriptome, which involves a family of methodologies, including RNA-seq ([1]), CAGE (Cap Analysis of Gene Expression; [2]) and SAGE (Serial Analysis of Gene Expression; [3]). When compared to microarrays, this class of methodologies offers several advantages, including detection of a wider level of expression levels and independence on prior knowledge of the biological system, which is required by hybridisation-based technologies, such as microarrays.

Typically, an experiment in this category starts with the extraction of a snapshot RNA sample from the biological system of interest and it's shearing in a large number of fragments of varying lengths. The population of these fragments is then reversed-transcribed to a c-DNA library and sequenced on a high- throughput platform, generating large numbers of short DNA sequences known as "reads". The ensuing analysis pipeline starts with mapping or aligning these reads on a reference genome. At the next stage, the mapped reads are summarised into gene-, exon- or transcript-level counts, normalised and further analysed for detecting differential gene expression [4].

It is important to realize that the normalised read (or tag) count data generated from this family of methodologies represents the number of times a particular class of c-DNA fragments has been sequenced, which is directly related to their abundance in the library and, in turn, the abundance of the associated transcripts in the original sample. Thus, this count data is essentially a discrete or digital measure of gene expression, which is fundamentally different in nature (and, in general terms, superior in quality) from the continuous fluorescence intensity measurements obtained from the application of microarray technologies. Due to their better quality, next-generation sequence assays tend to replace microarray- based technologies, despite their higher cost [5].

One approach for the analysis of count data of gene expression is to transform the counts to approximate normality and then apply existing methods aimed at the analysis of microarrays [6,7]. However, as noted in McCarthy et al. [8], this approach may fail in the case of very small counts (which are far from normally distributed) and also due to the strong mean-variance relationship of count data, which is not taken into account by tests based on a normality assumption. Proper statistical modelling and analysis of count data of gene expression requires novel approaches, rather than adaptation of existing methodologies, which aimed from the beginning at processing continuous input.

Formally, the generation of count data using next-generation sequencing assays can be thought of as random sampling of an underlying population of cDNA fragments. Thus, the counts for each tag describing a class of cDNA fragments can, in principle, be modelled using the Poisson distribution, whose variance is, by definition, equal to its mean. However, it has been shown that, in real count data of gene expression, the variance can be larger than what is predicted by the Poisson distribution [9-12]. An approach that accounts for the so-called "over-dispersion" in the data is to adopt quasi-likelihood methods, which augment the variance of the Poisson distribution with

***Corresponding author:** Dimitrios V Vavoulis, Department of Computer Science, University of Bristol, Bristol, United Kingdom, Tel: +44 (0)117 331573; E-mail: Dimitris.Vavoulis@bristol.ac.uk

**Received** October 20, 2013; **Accepted** November 18, 2013; **Published** November 25, 2013

**Citation:** Vavoulis DV, Gough J (2013) Non-Parametric Bayesian Modelling of Digital Gene Expression Data. J Comput Sci Syst Biol 7: 001-009. doi:10.4172/jcsb.1000131

a scaling factor, thus by-passing the assumption of equality between the mean and variance [13-16]. An alternative approach is to use the Negative Binomial distribution, which is derived from the Poisson, assuming a Gamma-distributed rate parameter. The Negative Binomial distribution incorporates both a mean and a variance parameter, thus modelling over-dispersion in a natural way [17,18]. An overview of existing methods for the analysis of gene expression count data can be found in Oshlack et al. and Kvam et al. [4,19].

Despite the decreasing cost of next-generation sequencing assays (and also due to technical and ethical restrictions), digital datasets of gene expression are often characterised by a small number of biological replicates or no replicates at all. Although this complicates any effort to statistically analyse the data, it has led to inventive attempts at estimating as accurately as possible the biological variability in the data given very small samples. One approach is to assume a locally linear relationship between the variance and the mean in the Negative Binomial distribution, which allows estimating the variance by pooling together data from genes with similar expression levels [17]. Alternatively, one can make the rather restrictive assumption that all genes share the same variance, in which case the over-dispersion parameter in the Negative Binomial distribution can be estimated from a very large set of data points [11]. A further elaboration of this approach is to assume a unique variance per gene and adopt a weighted-likelihood methodology for sharing information between genes, which allows for an improved estimation of the gene-specific over-dispersion parameters [8]. Another yet distinct empirical Bayes approach is implemented in the software *baySeq*, which adopts a form of information sharing between genes by assuming the same prior distribution among the parameters of samples demonstrating a large degree of similarity [18].

In summary, proper statistical modelling and analysis of digital gene expression data requires the development of novel methods, which take into account both the discrete nature of this data and the typically small number (or even the absence) of biological replicates. The development of such methods is particularly urgent due to the huge amount of data being generated by high-throughput sequencing assays. In this paper, we present a method for modelling digital gene expression data that utilizes a novel form of information sharing between genes (based on non-parametric Bayesian clustering) to compensate for the all-too-common problem of low or no replication, which plagues most current analysis methods.

## Approach

We propose a novel, non-parametric Bayesian approach for the analysis of digital gene expression data. Our point of departure is a hierarchical model for over-dispersed counts. The model is built around the Negative Binomial distribution, which depends, in our formulation, on two parameters: the mean and an over-dispersion parameter. We assume that these parameters are sampled from a Dirichlet process with a joint Inverse Gamma - Normal base distribution, which we have implemented using stick breaking priors. By construction, the model imposes a clustering effect on the data, where all genes in the same cluster are statistically described by a unique Negative Binomial distribution. This can be thought of as a form of information sharing between genes, which permit pooling together data from genes in the same cluster for improved estimation of the mean and over-dispersion parameters, thus bypassing the problem of little or no replication. We develop a blocked Gibbs sampling algorithm for estimating the posterior distributions of the various free parameters in the model. These include the mean and over-dispersion for each gene and the number of clusters (and their occupancies), which does not need to be fixed *a priori*, as in alternative (parametric) clustering methods. In principle, the proposed method can be applied on various forms of digital gene expression data (including RNA-seq, CAGE, SAGE, Tag-seq, etc.) with little or no replication and it is actually applied on one such example dataset herein.

## Modelling Over-Dispersed Count Data

The digital gene expression data we are considering is arranged in an $M \times N$ matrix, where each of the $N$ rows corresponds to a different gene and each of the $M$ columns corresponds to a different sample. Furthermore, all samples are grouped in $L$ different classes (i.e. tissues or experimental conditions). It holds that $L \leq M$, where the equality is true if there are no replicas in the data.

We indicate the number of reads for the $i^{th}$ gene at the $j^{th}$ sample with the variable $y_{ij}$. We assume that $y_{ij}$ is Poisson-distributed with a gene- and sample-specific rate parameter $r_{ij}$. The rate parameter $r_{ij}$ is assumed random itself and it is modelled using a Gamma distribution with shape parameter $\alpha_{i\lambda(j)}$ and scale parameter $s_{ij.}$ The function $\lambda(\cdot)$ in the subscript of the shape parameter maps the sample index $j$ to an integer indicating the class this sample belongs to. Thus, for a particular gene and class, the shape of the Gamma distribution is the same for all samples. Under this setup, the rate $r_{ij}$ can be integrated (or marginalised) out, which gives rise to the Negative Binomial distribution with parameters $\alpha_{i\lambda(j)}$ and $\mu_{ij}=\alpha_{i\lambda(j)} s_{ij}$ for the number of reads $y_{ij}$:

$$y_{ij} \big| \alpha_{i\lambda(j)}, \mu_{ij} \sim \frac{\Gamma(y_{ij}+\alpha_{i\lambda(j)})}{\Gamma(\alpha_{i\lambda(j)})\Gamma(y_{ij}+1)} \left(\frac{\alpha_{i\lambda(j)}}{\alpha_{i\lambda(j)}+\mu_{ij}}\right)^{\alpha_{i\lambda(j)}} \left(\frac{\mu_{i\lambda(j)}}{\alpha_{i\lambda(j)}+\mu_{ij}}\right)^{y_{ij}} \quad (1)$$

Where $\mu_{ij}$ is the mean of the Negative Binomial distribution and $\mu_{ij} + \alpha^{-1}_{i\lambda(j)} \mu^2_{ij}$ is the variance. Since the variance is always larger than the mean by the quantity $\alpha^{-1}_{i\lambda(j)} \mu^2_{ij}$, the Negative Binomial distribution can be thought of as a generalisation of the Poisson distribution, which accounts for over-dispersion. Furthermore, we model the mean as $\mu_{ij} = c_j e^{\beta i\lambda(j)}$, where the offset $c_j = \sum_{i=1}^{N} y_{ij}$ is the depth or exposure of sample $j$ and $\beta_{i\lambda(j)}$ is, similarly to $\alpha_{i\lambda(j)}$, a gene- and class-specific parameter. This formulation ensures that $\mu_{ij}$ is always positive, as it ought to.

Given the model above, the likelihood of observed reads $y_{ij}$ ={ $y_{ij}$ : $\lambda(j)=l$} for the $i^{th}$ gene in class $l$ is written as follows:

$$p(Y_{il} \mid \alpha_{il}, \beta_{il}) = \prod_j (Y_{ij} \mid \alpha_{i\lambda(j)}, \beta_{i\lambda(j)})$$

$$= \prod_j NegBinomial(Y_{ij} \mid \alpha_{i\lambda(j)}, c_j e^{\beta_{i\lambda(j)}}) \quad (2)$$

Where the index $j$ satisfies the condition $\lambda(j)=l$. By extension, for the $i^{th}$ gene across all sample classes, the likelihood of observed counts $y_i$={ $y_{ij}$ : $\lambda(j)=l$, $l$=1,…. , L} is written as:

$$p(Y_i \mid \alpha_{il}, \beta_{il}, \ldots\ldots, \alpha_{iL}, \beta_{iL}) = \prod_l p(Y_{il} \mid \alpha_{il}, \beta_{il}) \quad (3)$$

where the class indicator $l$ runs across all $L$ classes.

### Information sharing between genes

A common feature of digital gene expression data is the small number of biological replicates per class, which makes any attempt to estimate the gene- and class-specific parameters $\theta_{il}$={$\alpha_{il}$ , $\beta_{il}$ } through standard likelihood methods a futile exercise. In order to make robust estimation of these parameters feasible, some form of information sharing between different genes is necessary. In the present context, information sharing between genes means that not all values of $\theta_{il}$ are distinct; different genes (or the same gene across different sample classes) may share the same values for these parameters. This idea can be expressed formally by assuming that $\theta_{il}$ is random with an infinite mixture of discrete random measures as its prior distribution:

$$\theta_{il} \sim \sum_{k=1}^{\infty} w_k \delta_{\theta_k^*}, 0 \le w_k \le 1, \sum_{k=1}^{\infty} w_k = 1$$

where $\delta_{\theta_k^*}$ indicates a discrete random measure centered at $\theta_k^* = \{\alpha_k^*, \beta_k^*\}$ and $w_k$ is the corresponding weight. Conceptually, the fact that the above summation goes to infinity expresses our lack of prior knowledge regarding the number of components that appear in the mixture, other than the obvious restriction that their maximum number cannot be larger than the number of genes times the number of sample classes. In this formulation, the parameters $\theta_k^*$ are sampled from a prior base distribution $G_0$ with hyper-parameters $\phi, i.e. \theta_k^* \big| \phi \sim G_0(\phi)$. We assume that $\alpha_k^*$ is distributed according to an inverse Gamma distribution with shape $a_\alpha$ and scale $s_\alpha$, while $\beta_k^*$ follows the Normal distribution with mean $\mu_\beta$ and variance $\sigma_\beta^2$. Thus, $G_0$ is a joint distribution as follows:

$$\overbrace{\alpha_k^*, \beta_k^*}^{\theta_k^*} \Big| \overbrace{a_\alpha, s_\alpha, \mu_\beta, \sigma_\beta^2}^{\phi} \sim \overbrace{InvGamma(a_\alpha, s_\alpha) \cdot Normal(\mu_\beta, \sigma_\beta^2)}^{G_0(\phi)}, \text{ k=1,2,....} \tag{5}$$

Given the above, $\alpha_k^*$ can take only positive values, as it ought to, while $\beta_k^*$ can take both positive and negative values.

What makes the mixture in Eq. 4 special is the procedure for generating the infinite sequence of mixing weights. We set $w_1=V_1$ and $w_k = V_k \prod_{m=1}^{k-1}(1-V_m)$ for $k \ge 2$ where $\{V_1,..., V_k\}$ are random variables following the Beta distribution, i.e., $Vk \sim Beta(a_k, b_k)$. This constructive way of sampling new mixing weights resembles a stick-breaking process; generating the first weight w1 corresponds to breaking a stick of length 1 at position $V_1$; generating the second weight $w_2$ corresponds to breaking the remaining piece at position $V_2$ and so on. Thus, we write:

$$w_k \big| a_k, b_k \sim Stick(a_k, b_k), \text{ k=1,2,....} \tag{6}$$

There are various ways for defining the parameters $a_k$ and $b_k$. Here, we consider only the case where $a_k=1$ and $b_k=\eta$, with $\eta > 0$. This parametrisation is equivalent to setting the prior of $\theta_{il}$ to a Dirichlet Process with base distribution $G_0$ and concentration parameter $\eta$. By construction, this procedure leads to a rapidly decreasing sequence of sampled weights, at a rate which depends on $\eta$. For values of $\eta$ much smaller than 1, the weights $w_k$ decrease rapidly with increasing $k$, only one or few weights have significant mass and the parameters $\theta_{il}$ share a single or a small number of different values $\theta_k^*$. For values of the concentration parameter much larger than 1, the weights $w_k$ decrease slowly with increasing $k$, many weights have significant mass and the values of $\theta_{il}$ tend to be all distinct to each other and distributed according to $G_0$. Below, we set $\eta$=1, which results in a balanced decrease of the weight mass with increasing $k$. In particular, for $\eta$=1, log $(w_k)$ decreases (on average) in an unbiased manner with increasing $k$.

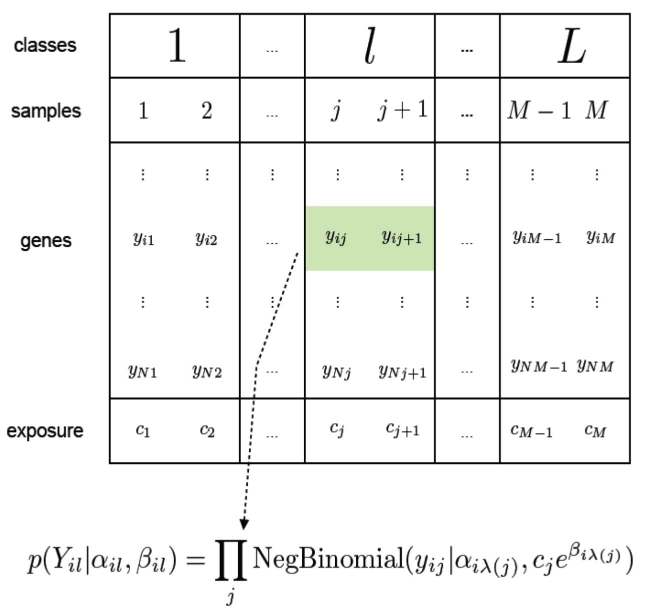
Given the above formulation, sampling $\theta_{il}$ from its prior distribution is straightforward. First, we introduce an indicator variable $z_{il} \in \{1, 2, ...\}$, which points to the value of $\theta_k^*$ corresponding to the $i^{th}$ gene in class l. We sample such indicator variables for each gene in each class from the Categorical distribution, i.e. $z_{il} \sim$ Categorical $(w_1, w_2, ...)$, and set $\theta_{il} \equiv \theta_{z_{il}}^*$. Although $G_0$ is continuous, the distribution of $\theta_{il}$ is almost surely discrete and, therefore, its values are not all distinct. Different genes may share the same value of $\theta^*$ and, thus, all genes are grouped in a finite (unknown) number of clusters, according to the value of $\theta_k^*$ they share. Modeling digital gene expression data using this approach is one way to bypass the problem of few (or the absence of) technical replicates, since the data from all genes in the same cluster are pooled together for estimating the parameters that characterize this cluster. The clustering effect described in this section is illustrated in Figure 2.

## Generative model

The description in the previous paragraphs suggests a hierarchical model, which presumably underlies the stochastic generation of the data matrix in Figure 1. This model is explicitly described below:

$$\begin{aligned}
&\theta_k^* \big| a_\alpha, s_\alpha, \mu_\beta, \sigma_\beta^2 \sim InvGamma(a_\alpha, s_\alpha) \cdot Normal(\mu_\beta, \sigma_\beta^2) \\
&w_1, w_2, ... \big| \eta \sim Stick(1, \eta) \\
&z_{\mathrm{i}\lambda(\mathrm{j})} \big| w_1, w_2, .... \sim Categorical(\big| w_1, w_2, ....) \\
&\theta_{\mathrm{i}\lambda(\mathrm{j})} \equiv \theta_{z_{\mathrm{i}\lambda(\mathrm{j})}}^* \\
&y_{ij} \big| \theta_{\mathrm{i}\lambda(\mathrm{j})} \sim Negbinomial(\theta_{\mathrm{i}\lambda(\mathrm{j})})
\end{aligned} \tag{7}$$

At the bottom of the hierarchy, we identify the measured reads $y_{ij}$ for each gene in each sample, which follow a Negative Binomial distribution with parameters $\theta_{\mathrm{i}\lambda(\mathrm{j})} = \{\alpha_{\mathrm{i}\lambda(\mathrm{j})}, \beta_{\mathrm{i}\lambda(\mathrm{j})}\}$. The parameters of the Negative Binomial distribution $\theta_{\mathrm{i}\lambda(\mathrm{j})}$ are gene- and class-specific and they are completely determined by an also gene- and class-specific indicator variable $z_{i\lambda(j)}$ and the centers $\theta_k^*$ of the infinite mixture of



$$p(Y_{il}|\alpha_{il}, \beta_{il}) = \prod_j \mathrm{NegBinomial}(y_{ij}|\alpha_{i\lambda(j)}, c_j e^{\beta_{i\lambda(j)}})$$

**Figure 1:** Format of digital gene expression data. Rows correspond to genes and columns correspond to samples. Samples are grouped into classes (e.g. tissues or experimental conditions). Each element of the data matrix is a whole number indicating the number of counts or reads corresponding to the $i^{th}$ gene at the $j^{th}$ sample. The sum of the reads across all genes in a sample is the depth or exposure of that sample.

point measures in Eq. 4. These centers are distributed according to a joint inverse Gamma and Normal distribution with hyper-parameters $\phi=\{a_\alpha,s_\alpha,\mu_\beta,\sigma_\beta^2\}$, while the indicator variables are sampled from a Categorical distribution with weights $\{w_1, w_2, \ldots\}$. These are, in turn, sampled from a stick-breaking process with concentration parameter $\eta$. In this model, $\phi$, $w_k$, $\theta_k^*$ and $z_{i\lambda(j)}$ are latent variables, which are subject to estimation based on the observed data.

## Inference

At this point, we introduce some further notation. We indicate the $N \times L$ matrix of indicator variables with the letter $Z$; $\Theta^* = \{\theta_1^*,\theta_2^*,\ldots\}$ lists the centers of the point measures in Eq. 4 and $W=\{w_1, w_2, \ldots\}$ is the vector of mixing weights. We are interested in computing the joint posterior density $p(Z,W,\Theta^*,\phi|Y)$, where $Y$ is a matrix of count data as in Figure 1. We approximate the above distribution through numerical (Monte Carlo) methods, i.e. by sampling a large number of $\{\Theta^*,\mathrm{W},\mathrm{Z},\phi\}$ -tuples from it. One way to achieve this is by constructing a Markov chain, which admits $p(Z,W,\Theta^*,\phi|Y)$ as its stationary distribution. Such a Markov chain can be constructed by using Gibbs sampling, which consists of alternating repeated sampling from the full conditional posteriors $p(\Theta^*|Y,Z,\phi)$, $p(W|Z)$, $p(Z|Y,\Theta^*,W)$ and $p(\phi|\Theta^*,Z)$. Below, we explain how to sample from each of these conditional distributions.

### Sampling from the conditional posterior $p(\Theta^* \mid Y,Z,\varphi)$

In order to sample from the above distribution it is convenient to truncate the infinite mixture in Eq. 4 by rejecting all terms with index larger than K and setting $w_k = 1-\sum_{k-1}^{K-1} w_k$, which is equivalent to setting $\mathrm{V}_\mathrm{K} = 1$. It has been shown that the error associated with this approximation when $\mathrm{V}_\mathrm{k} \sim Beta(1,\eta)$ is less than or equal to $4\mathrm{NM}\ \exp\left(-\frac{k-1}{\eta}\right)$ ([8]). For example, for $N=14\times10^3, M=6, k=200$ and $\eta=1$, the error is minimal (less than $10^{-80}$). Thus, the truncation should be virtually indistinguishable from the full (infinite) mixture.

Next, we distinguish between $k_{ac}$ active clusters $\Theta_{ac}^*$ and $k_{in}$ inactive clusters $\Theta_{in}^*$, such that $\Theta^*=\{\Theta_{ac}^*,\Theta_{\mathrm{in}}^*\}$ and $k=\mathrm{k}_{\mathrm{ac}}+k_{in}$. Active clusters are those containing at least one gene, while those containing no genes are considered inactive. We write:

$$p(\Theta^*|Y,Z,\Phi) = p(\Theta_{ac}^*,\Theta_{\mathrm{in}}^*|Y,Z,\Phi) = p(\Theta^*{}_{ac}|Y,Z,\Phi)p(\Theta_{\mathrm{in}}^*|\Phi)$$

Updating the inactive clusters is a simple matter of sampling $\mathrm{k}_{\mathrm{in}}$ times from the joint distribution in Eq. 5 given the hyper-parameters ϕ. Sampling the active clusters is more complicated and involves sampling each active cluster center $\Theta_{ac,k}^*$ individually from its respective posterior $(\Theta_{ac,k}^* p|Y_{ac,k})$, where $Y_{ac,k}$ is a matrix of measured count data for all genes in the $k^{th}$ active cluster. Sampling $\theta_{ac,k}^* = \{\alpha_{ac,k}^*,\beta_{ac,k}\}$ is done using the Metropolis algorithm with acceptance probability:

$$P_{acc} = \min\left(1,\frac{p(Y_{ac,k}|\theta_{ac,k}^+)p(\theta_{ac,k}^+|\Phi)}{p(Y_{ac,k}|\theta_{ac,k}^*)p(\theta_{ac,k}^*|\Phi)}\right) \qquad (8)$$

Where the superscript + indicates a candidate vector of parameters. Each of the two elements (α and β) of this vector is drawn from a symmetric proposal of the following form:

$$q(x^+ \mid x^*) = x^* \exp\,(0.01 \cdot r) \qquad (9)$$

Where the random number $r$ is sampled from the standard Normal distribution, i.e., r ~ Normal(0,1). The prior of is a joint Inverse Gamma - Normal distribution, as shown in Equation 5, while the likelihood function $p(Y_{ac,k}|\theta^*{}_{ac,k})$ is a product of Negative Binomial probability distributions, similar to those in Equation 2 and 3.

### Sampling from the conditional posterior $p(Z \mid Y,\Theta^*,W)$

Each element $z_{il}$ of the matrix of indicator variables $Z$ is sampled from a Categorical distribution with weights $\pi_{il}=\{\pi_{il}^1,\ldots,\pi_{il}^K\}$, where $\pi_{il}^K = \prod_{il}^K / \sum_{m=1}^K \prod_{il}^m$ and:

$$\{\pi_{il}^1,\ldots,\pi_{il}^K\} \propto \{w_1 p(Y_{il}|\theta_1^*),\ldots\ldots,w_K p(Y_{il}|\theta_1^*)\} \qquad (10)$$

In the above expression, $Y_{il}$ is the data for the $i^{th}$ gene in class $l$, as mentioned in a previous section. Notice that $z_{il}$ can take any integer value between 1 and $K$ and that the weights $\pi_{il}$ depend both on the cluster weights $w_k$ and on the value of the likelihood function $p(Y_{il}|\theta_1^*)$.

### Sampling from the conditional posterior p(w | z)

The mixing weights $W$ are generated using a truncated stick-breaking process with $\eta=1$. As pointed out in Engström et al. [20], this implies that $W$ follows a generalised Dirichlet distribution. Considering the conjugacy between this and the multinomial distribution, the first step in updating $W$ is to generate $K - 1$ Beta-distributed random numbers:

$$V_k \sim \mathrm{Beta}(1+N_k,\eta+N-\sum_{m=1}^{k}N_m) \qquad (11)$$

for $k=1, \ldots, K-1$, where $N_k$ is the total number of genes in the $k^{th}$ cluster. Notice that $N_k$ can be inferred from $Z$ by simple counting and $\sum_{m=1}^K N_k = N$, where $N$ is the total number of genes. $V_K$ is set equal to 1, in order to ensure that the weights add up to 1. These are simply generated by setting $V_1=w_1$ and $w_k = \mathrm{V}_\mathrm{k}\prod_{m=1}^{\mathrm{k}-1}(1-\mathrm{V}_\mathrm{m})$, as mentioned in a previous section.

### Sampling from the conditional posterior $p(\phi \mid \Theta^*,\mathrm{Z})$

The hyper-parameters $\phi=\{a_\alpha,s_\alpha,\mu_\beta,\sigma_\beta^2\}$ influence indirectly the observations $Y$ through their effect on the distribution of the active cluster centres, $\Theta_{ac}^* = \{\alpha_{ac}^*,\beta_{ac}^*\}$ where $\alpha_{ac}^* = \{\alpha_{ac,1}^*,\ldots.\alpha_{ac,K_{ac}}^*\}$ and $\beta_{ac}^* = \{\beta_{ac,1}^*,\ldots.\beta_{ac,K_{ac}}^*\}$. If we further assume independence between $\alpha_{ac}^*$ and $\beta_{ac}^*$ we can write $p(\phi \mid \Theta^*,\mathrm{Z}) = p(a_\alpha,s_\alpha,\mu_\beta,\sigma_\beta^2|\alpha_{ac}^*,\beta_{ac}^*) = p(a_\alpha,s_\alpha|\alpha_{ac}^*)p(\mu_\beta,\sigma_\beta^2|\beta_{ac}^*)$.

Assuming $K_{ac}$ active clusters and considering that the prior for α* (see Equation 5), it follows that the posterior $p(a_\alpha,s_\alpha|\alpha_{ac}^*)$ is:

$$p(a_\alpha,s_\alpha|\alpha_{ac}^*) \propto \frac{\gamma_1^{a_\alpha-1}\exp(-s_\alpha\gamma_2)s_\alpha^{a_\alpha\gamma_3}}{\Gamma(a_\alpha)^{\gamma_4}} \qquad (12)$$

The parameters $\gamma_1$ to $\gamma_4$ are given by the following expressions:

$$\gamma_1 = \gamma_1^{(0)}\prod_{k=1}^{K_{ac}}\frac{1}{\alpha_{ac,k}^*}$$

$$\gamma_2 = \gamma_2^{(0)} + \prod_{k=1}^{K_{ac}}\frac{1}{\alpha_{ac,k}^*}$$

$$\gamma_3 = \gamma_3^{(0)} + K_{ac}$$

$$\gamma_4 = \gamma_4^{(0)} + K_{ac}$$

where the initial parameters $\gamma_1^{(0)},\gamma_2^{(0)},\gamma_3^{(0)},and\gamma_4^{(0)}$ are all positive. Since sampling from Equation 12 cannot be done exactly, we employ a Metropolis algorithm with acceptance probability

$$P_{acc} = \min(1,\frac{p(a_\alpha^+,s_\alpha^+|a_{ac}^*)}{p(a_\alpha,s_\alpha|a_{ac}^*)}) \qquad (13)$$

where the proposal distribution *q(•|•)* for sampling new candidate points has the same form as in Eq. 9. Furthermore, taking advantage of the conjugacy between a normal likelihood and a Normal-Inverse Gamma prior, the posterior probability for parameters $\mu_\beta$ and $\sigma_\beta^2$ becomes:

$$p(\mu_\beta,\sigma_\beta^2 \middle| \beta_{ac}^*) = NormalInverseGamma(\delta_1,\delta_2,\delta_3,\delta_4) \qquad (14)$$

The parameters $\delta_1$ to $\delta_4$ (given initial parameters $\delta_3^{(0)}$ to $\delta_4^{(0)}$) are as follows:

$$\delta_1 = \frac{\delta_1^{(0)}\delta_2^{(0)} + K_{ac}\bar{\beta}_{ac}^*}{\delta_2^{(0)} + K_{ac}}$$

$$\delta_2 = \delta_2^{(0)} + K_{ac}$$

$$\delta_3 = \delta_3^{(0)} + \frac{K_{ac}}{2}$$

$$\delta_4 = \delta_4^{(0)} + \frac{1}{2}\sum_{k=1}^{K_{ac}}(\beta_{ac,k}^* - \bar{\beta}_{ac}^*) + \frac{1}{2}\frac{\delta_2^{(0)}K_{ac}}{\delta_2^{(0)} + K_{ac}}(\bar{\beta}_{ac}^* - \delta_1^{(0)})$$

where $\bar{\beta}_{ac}^* = \frac{1}{K_{ac}}\sum_{k=1}^{K_{ac}}\beta_{ac,k}^*$. Sampling a $\{\mu_\beta,\sigma_\beta^2\}$ -pair from the above posterior takes place in two simple steps: first, we sample $\sigma_\beta^2 \sim InverseGamma(\delta_3,\delta_4)$ where $\delta_3$ and $\delta_4$ are shape and scale parameters, respectively. Then, we sample $\mu_\beta \sim Normal(\delta_1,\sigma_\beta^2 / \delta_2)$.

## Algorithm

We summarise the algorithm for drawing samples from the posterior $p(\Theta^*,Z,W,\phi \mid Y)$ below. Notice that $x^{(t)}$ indicates the value of *x* at the *t*<sup></sup>$^{th}$ iteration of the algorithm. $x^{(0)}$ is the initial value of *x*.

1. Set $\gamma^{(0)} = \{\gamma_1^{(0)},\gamma_2^{(0)},\gamma_3^{(0)},\gamma_4^{(0)}\}$
2. Set $\delta^{(0)} = \{\delta_1^{(0)},\delta_2^{(0)},\delta_3^{(0)},\delta_4^{(0)}\}$
3. Set $\phi^{(0)} = \{a_\alpha^{(0)},b_\alpha^{(0)},\mu_\beta^{(0)},\sigma_\beta^{2(0)}\}$
4. Set $K$, the truncation level
5. Sample $\Theta^{*(0)}$ from its prior (Eq. 5) conditional on $\phi^{(0)}$
6. Set all *K* elements of $W^{(0)}$ to the same value i.e *1/K*
7. Sample $Z^{(0)}$ from the Categorical distribution with weights $W^{(0)}$
8. For *t*=1,2,3,.....T

a. Sample $\Theta_{ac}^{*(t)}$ given $Z^{(t-1)},\phi^{(t-1)}$ and the data matrix *Y* using a single step of the Metropolis algorithm for each active cluster (see Eq. 8)

b. Sample $\Theta_{in}^{*(t)}$ from its prior given $\phi^{(t-1)}$ (see Eq. 5)

c. Sample $\mathrm{Z}^{(\mathrm{t})}$ given $\Theta^{*(\mathrm{t})},\mathrm{W}^{(\mathrm{t}-1)}$ and the data matrix *Y* (see Eq. 10)

d. Sample $\mathrm{W}^{(\mathrm{t})}$ given $\mathrm{Z}^{(\mathrm{t})}$ (see Eq. 11)

e. Sample $\phi^{(\mathrm{t})}$ given $\Theta_{ac}^{*(t)}$ and $\phi^{(t-1)}$ (see Eqs. 12 and 14)

9. Discard the first $T_0$ samples, which are produced during the burn-in period of the algorithm (i.e. before equilibrium is attained), and work with the remaining $T - T_0$ samples.

The above procedure implements a form of blocked Gibbs sampling with embedded Metropolis steps for impossible to directly sample from distributions.

## Results and Discussion

We applied the methodology described in the preceding sections on publicly available digital gene expression data (obtained from control and cancerous tissue cultures of neural stem cells; [20]) for evaluation purposes. The data we used in this study can be found at the following URL: http://genomebiology.com/content/supplementary/gb-2010-11-10-r106-s3.tgz. As shown in Table 1, this dataset consists of four libraries from glioblastoma-derived neural stem cells and two from non- cancerous neural stem cells. Each tissue culture was derived from a different subject (with the exception of GliNS1 and G144, which came from the same patient). Thus, the samples are divided in two classes (cancerous and non-cancerous) with four and two replicates, respectively.

We implemented the algorithm presented above in the programming language Python, using the libraries NumPy, SciPy and MatplotLib. The most recent version of the software can be found at the following link: https://bitbucket.org/DimitrisVavoulis/dgeclust. Calculations were expressed as operations between arrays and the multiprocessing Python module was utilised in order to take full advantage of the parallel architecture of modern multicore processors. The algorithm was run for 200K iterations, which took approximately two days to complete on a 12-core desktop computer. Simulation results were saved to the disk every 50 iterations.

| | Cancerous | | | Non-Cancerous | | |
|---|---|---|---|---|---|---|
| **Genes** | **GliNS1** | **G144** | **G166** | **G179** | **CB541** | **CB660** |
| 13CDNA73 | 4 | 0 | 6 | 1 | 0 | 5 |
| 15E1.2 | 75 | 74 | 222 | 458 | 215 | 167 |
| 182-FIP | 118 | 127 | 555 | 231 | 334 | 114 |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |

**Table 1:** Format of the data [6].

The first four samples are from glioblastoma neural stem cells, while the last two are from non-cancerous neural stem cells. The dataset contains a total of 18760 genes (i.e. rows).

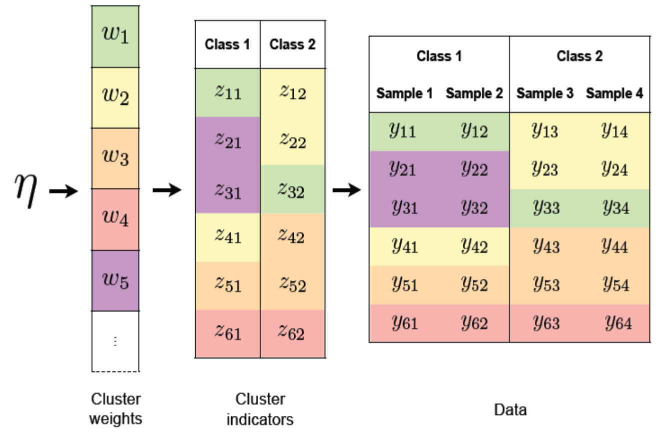


**Figure 2:** The clustering effect that results from imposing a stick-breaking prior on the gene and class- specific model parameters, $\theta_{ij}$. A matrix of indicator variables is used to cluster the observed count data into a finite number of groups, where the genes in each group share the same model parameters. The number of clusters is not known a priori. The distribution of weight mass among the various clusters in the model is determined by parameter $\eta$.

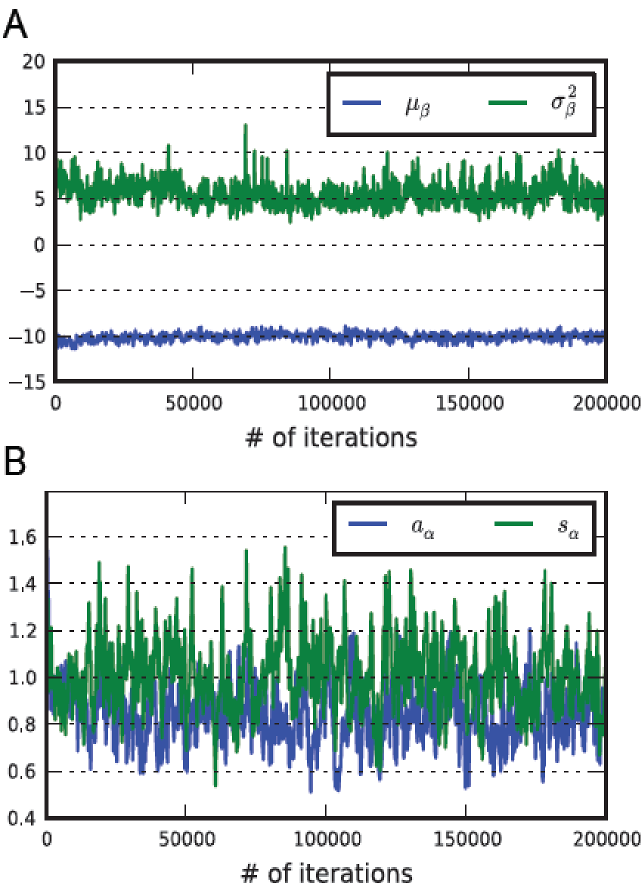


**Figure 3:** Simulation results after 200K iterations. The chains of random samples correspond to the components of the vector of hyper-parameters $\phi$ , i.e. $\mu_\beta$ and $\sigma^2_\beta$ . (Panel A) and $a_\alpha$ and $s_\alpha$ (panel B). The former determines the Normal prior distribution of the cluster center parameters β*, while the latter pair determines the Inverse Gamma prior distribution of the cluster center parameters α*. The random samples in each chain are approximately sampled (and constitute an approximation of) the corresponding posterior distribution conditional on the data matrix *Y*.

The raw simulation output includes chains of random values of the hyper-parameters $\phi$ , the gene- and class-specific indicators *Z* and the active cluster centres $\Theta^*_{ac}$ , which constitute an approximation to the corresponding posterior distributions given the data matrix *Y*. The chains corresponding to the four different components of $\phi = \{a_\alpha, s_\alpha, \mu_\beta, \sigma^2_\beta\}$ are illustrated in Figure 3. It may be observed that these reached equilibrium early during the simulation (after less than 20K iterations) and they remained stable for the remaining of the simulation. As explained earlier, these hyper-parameters are important, because they determine the prior distributions of the cluster centres α* and β* (hyper-parameters $\{a_\alpha, s_\alpha\}$ and $\{\mu_\beta, \sigma^2_\beta\}$ , respectively) and, subsequently, of the gene- and class-specific parameters α and β. It follows from analysis of the chains in Figure 3 that the estimates for these hyper-parameters are (indicating the mean and standard deviation of the estimates): $a_\alpha = 0.83 \pm 0.13, s_\alpha = 1.00 \pm 0.16, \mu_\beta = -10.01 \pm 0.39, \sigma^2_\beta = 5.41 \pm 1.32$. The corresponding Inverse Gamma and Normal distributions, which are the priors of the cluster centres α* and β*, respectively, are illustrated in Figure 4.

A major use of the methodology presented above is that it allows us to estimate the gene and class-specific parameters α and β, under the assumption that the same values for these parameters are shared between different genes or even by the same gene among different sample classes. This form of information sharing permits pulling together data from different genes and classes for estimating pairs of *α and β* parameters in a robust way, even when only a small number of replicates (or no replicates at all) are available per sample class. As an example, in Figure 5 we illustrate the chains of random samples for *α and β* corresponding to the non-cancerous class of samples for the tag with ID 182-FIP (third row in Table 1). These samples constitute approximations of the posterior distributions of the corresponding parameters. Despite the very small number of replicates (*n=4*), the variance of the random samples is finite. Similar chains were derived for each gene in the dataset, although it should be emphasised that the number of such estimates is smaller than the total number of genes, since more than one genes share the same parameter estimates.

It has already been mentioned that the sharing of *α and β* parameter values between different genes can be viewed as a form of clustering (Figure 2), i.e. there are different groups of genes, where all genes in a particular group share the same *α and β* parameter values. As expected in a Bayesian inference framework, the number of clusters is not constant, but it is itself a random variable, which is characterised by its own posterior distribution and its value, fluctuates randomly from

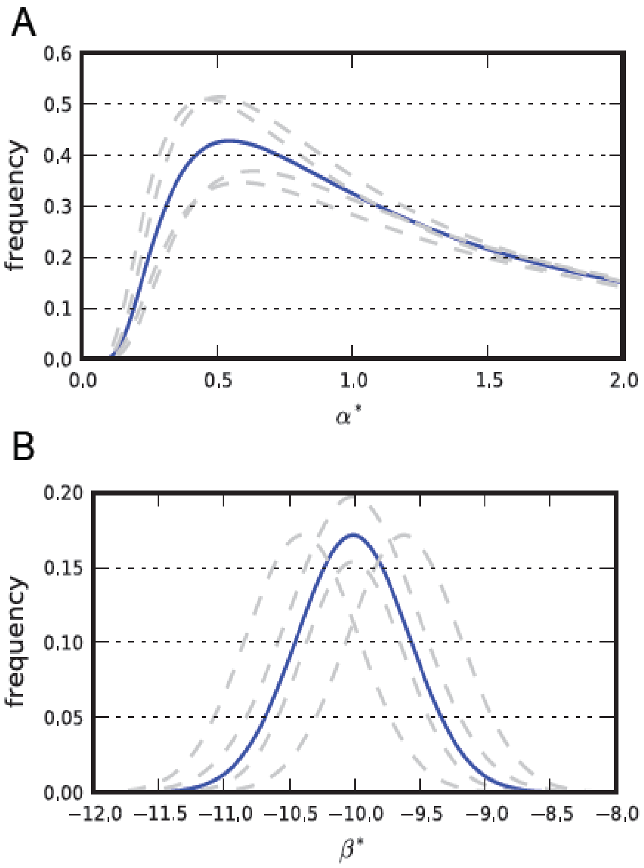


**Figure 4:** Estimated Inverse Gamma (panel A) and Normal (panel B) prior distributions for the cluster parameters α* and β*, respectively. The solid lines indicate mean distributions, i.e. those obtained for the mean values of the hyper-parameters $a_\alpha, s_\alpha, \mu_\beta, \sigma^2_\beta$ . The dashed lines are distributions obtained by adding or subtracting individually one standard deviation from each relevant hyper-parameter.

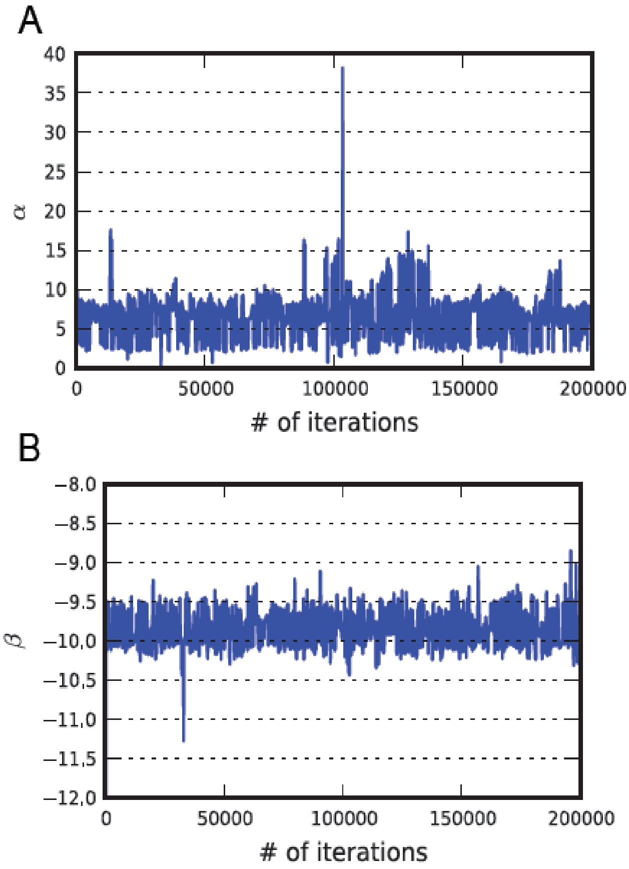


**Figure 5:** Chains of random samples approximating the posterior distributions of the parameters *α* (panel A) and *β* (panel B) corresponding to the non-cancerous class of samples for the tag with ID 182-FIP (third row in Table 1). These samples were generated after 200K iterations of the algorithm. A similar pair of chains exists for each gene at each sample class (i.e. cancerous and non-cancerous), although not all pairs are distinct to each other due to the clustering effect imposed on the data by the algorithm.

one iteration to the next. In Figure 6, we illustrate the chain of sampled cluster numbers during the course of the simulation (panel A). The first 75K iterations were discarded as burn-in and the remaining samples were used for plotting the histogram in panel B, which approximates the posterior distribution of the number of clusters given the data matrix *Y*. It may be observed that the number of clusters fluctuates between 35 and 55 with a peak at around 42 clusters. The algorithm we present above does not make any particular assumptions regarding the number of clusters, apart from the obvious one that this number cannot exceed the number of genes times the number of sample libraries. Although the truncation level *K*=200 sets an artificial limit in the maximum number of clusters, this is never a problem in practise, since the actual estimated number of clusters is typically much smaller that the truncation level *K* (see the y-axis in Figure 6A). The fact that the number of clusters is not decided a priori, but rather inferred along with the other free parameters in the model sets the described methodology in an advantageous position with respect to alternative clustering algorithms, which require deciding the number of clusters at the beginning of the simulation [21].

Similarly to the stochastic fluctuation in the number of clusters, the cluster occupancies (i.e. the number of genes per cluster) are a random vector. In Figure 7, we illustrate the cluster occupancies at two different stages of the simulation, i.e. after 100K and 200K iterations, respectively. We may observe that, with the exception of a single super-cluster (containing more than 6000 genes), cluster occupancies range from between around 3000 and less than 1000 genes. It should be clarified that each cluster includes many (potentially, hundreds of) genes and it may span several classes. An individual cluster represents a Negative Binomial distribution (with concrete *α and β* parameters), which models with high probability the count data from all its member genes. This is illustrated in Figure 8, where we show the histogram of the log of the count data from the first sample (sample GliNS1 in Table 1) along with a subset of the estimated clusters after 200K iterations (gray lines) and the fitted model (red line). It may be observed that each cluster models a subset of the gene expression data in the particular sample. The complete model describing the whole sample is a weighted sum of the individual clusters/Negative Binomial distributions. Formally,

$$p(Y_j \middle| \alpha_{1\lambda(j)}, \beta_{1\lambda(j)}, \ldots\ldots.\alpha_{N\lambda(j)}, \beta_{N\lambda(j)}) = \frac{1}{N} p(Y_{ij} \middle| \alpha_{i\lambda(j)}, \beta_{i\lambda(j)}) \qquad (15)$$

where $Y_j$ is the $j^{th}$ sample and the index *i* runs over all N genes. We

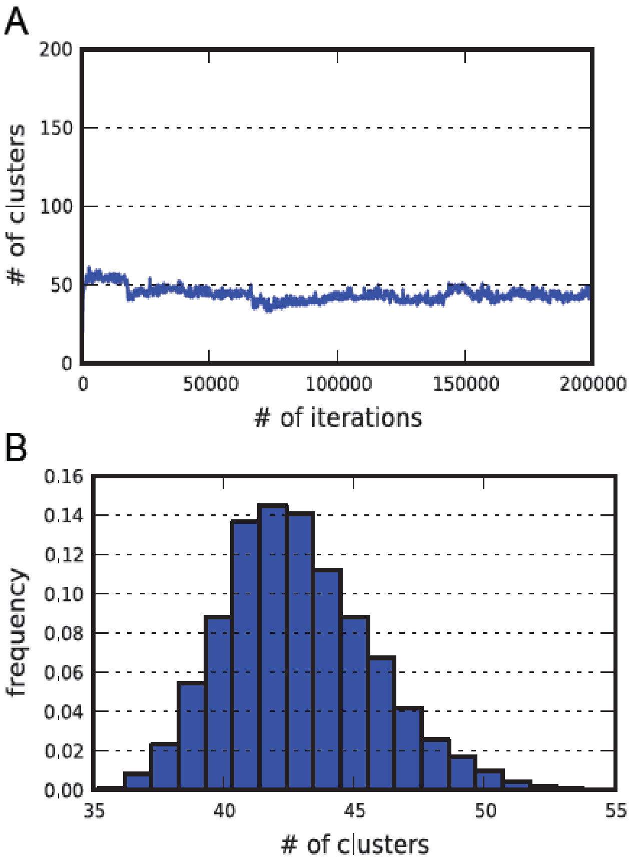


**Figure 6:** Stochastic evolution of the number of clusters during 200K iterations of the simulation (panel A) and the resulting histogram after discarding the first 75K iterations as burn-in (panel B). After reaching equilibrium, the number of clusters fluctuates around a mean of approximately 43 clusters. In general, the estimated number of clusters is much smaller than the truncation level (K = 200, see y-axis in panel A). The histogram in panel B approximates the posterior distribution of the number of clusters given the data matrix *Y*.

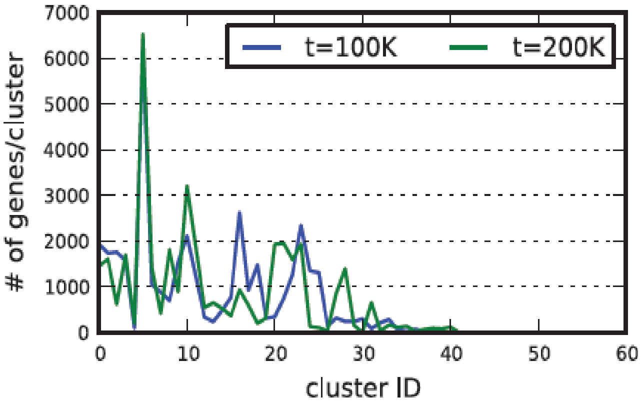


**Figure 7:** Cluster occupancies after 100K and 200K iterations of the algorithm. A single super-cluster (including more than 6000 genes) appears at both stages of the simulation. The occupancy of the remaining clusters demonstrates some variability during the course of the simulation, with clusters containing between 3000 and less than 1000 genes.

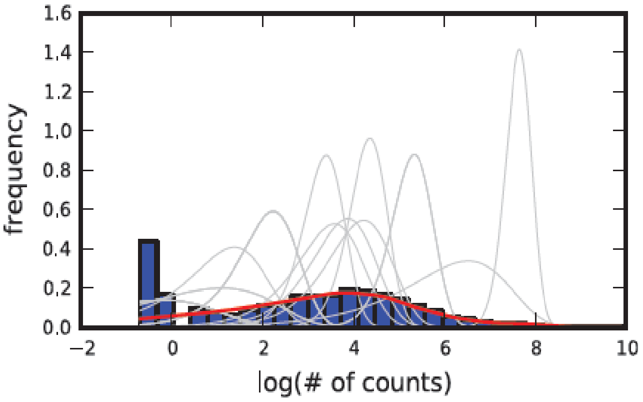


**Figure 8:** Histogram of the log of the number of reads from sample GliNS1, a subset of the estimated clusters (gray lines) and the estimated model of the sample at the end of the simulation. Each cluster (gray line) represents a Negative Binomial distribution with specific *α and β* parameters, which models a subset of the count data in this particular sample. The complete model (red line) is the weighted sum of all component clusters.

repeat that not all $\{\alpha_{i\lambda(j)}, \beta_{i\lambda(j)}\}$ pairs are distinct. Also, clusters with larger membership (i.e. including a larger number of genes) have larger weight in determining the overall model.

The proposed methodology provides a compact way to model each sample in a digital gene expression dataset following a two-step procedure: first, the dataset is partitioned into a finite number of clusters, where each cluster represents a Negative Binomial distribution (modelling a subset of the data) and the parameters of each such distribution are estimated. Subsequently, each sample in the dataset can be modelled as a weighted sum of Negative Binomial distributions. In Figure 9, we show the log of count data for each sample in the dataset shown in Table 1 along with the fitted models (red lines) after 200 K iterations of the algorithm.

## Conclusion

Next-generation sequencing technologies are routinely being used for generating huge volumes of gene expression data in a relatively short time. This data is fundamentally discrete in nature and their analysis requires the development of novel statistical methods, rather than modifying existing tests that were originally aimed at the analysis of microarrays. The development of such methods is an active area of research and several papers have been published on the subject [4,19].

In this paper, we present a novel approach for modelling over-dispersed count data of gene expression (i.e. data with variance

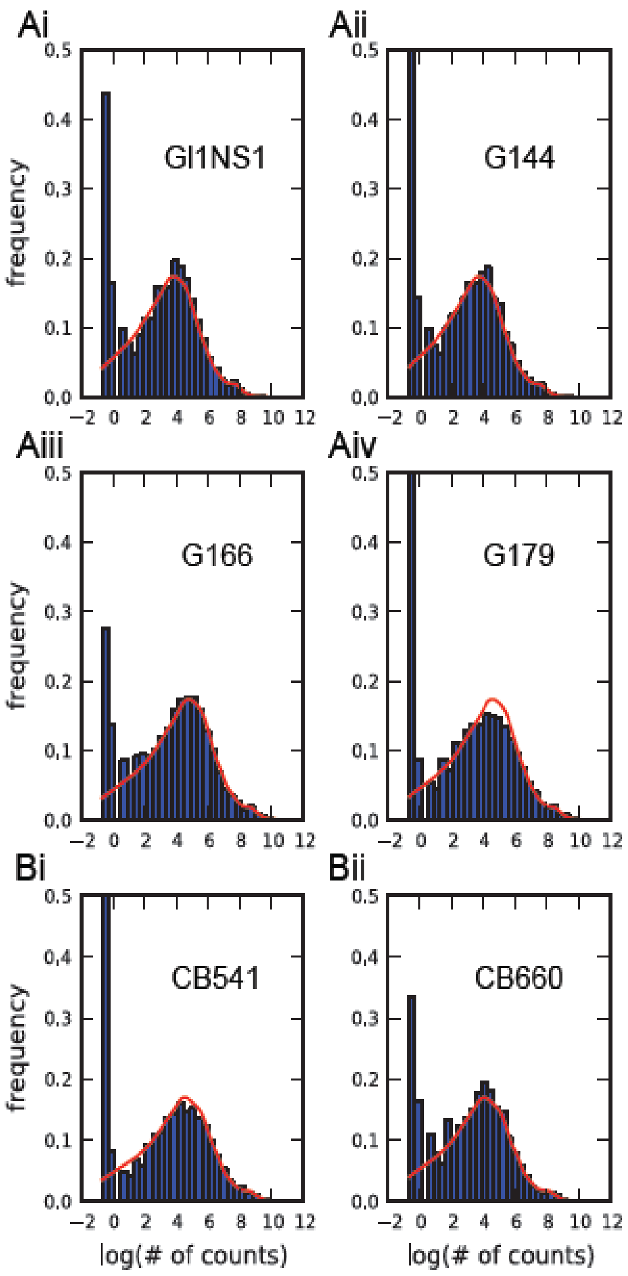


**Figure 9:** Histograms of the log of the number of reads from cancerous (panels Ai-iv) and non-cancerous (panels Bi,ii) samples and the respective estimated models after 200K iterations of the algorithm. As already mentioned, each red line is the weighted sum of many component Negative Binomial distributions / clusters, which model different subsets of each data sample. We may observe that the estimated models fit tightly the corresponding data samples.

larger than the mean predicted by the Poisson distribution) using a hierarchical model based on the Negative Binomial distribution. The novel aspect of our approach is the use of a Dirichlet process in the form of stick breaking priors for modelling the parameters (mean and over-dispersion) of the Negative Binomial distribution. By construction, this formulation forces clustering of the count data, where genes in the same cluster are sampled from the same Negative Binomial distribution, with a common pair of mean and over-dispersion parameters. Through this elegant form of information sharing between genes, we compensate for the problem of little or no replication, which often restricts the analysis of digital gene expression datasets. We have demonstrated the ability of this approach to model accurately actual biological data by applying the proposed methodology on a publicly available dataset obtained from cancerous and non-cancerous cultured neural stem cells [20].

We show that inference is achieved in the proposed model through the application of a blocked Gibbs sampler, which includes estimating, among others, the gene- and class-specific mean and over-dispersion of the Negative Binomial distribution. Similarly, the number of clusters and their occupancies are inferred along with the rest free parameters in the model.

Currently, the software implementing the proposed method remains relatively computationally expensive. In particular, 200 K iterations require approximately two days completing on a 12-core desktop computer. This time scale is not disproportionate to the production time of experimental data and it is mainly due to the high volume of the tested data (> 15 K genes per sample) and the need to obtain long chains of samples for a more accurate estimation of posterior distributions. Long execution times are a characteristic, more generally, of all Monte Carlo approximation methods. Our implementation of the algorithm is completely parallelised and calculations are expressed as operations between vectors in order to take full advantage of modern multi-core computers. Ongoing work towards reducing execution times aims at the application of variation inference methods [22], instead of the blocked Gibbs sampler we currently use. The algorithm can be further improved by avoiding truncation of the infinite summation described in Equation 4, as described in Papaspiliopoulos and Roberts [23] and in Walker [24].

This non-parametric Bayesian approach for modelling count data has thus shown great promise in handling over-dispersion and the all-too-common problem of low replication, both in theoretical evaluation and on the example dataset. The software that has been produced (DGE clust ) will be of great utility for the study of digital gene expression data and the statistical theory will contribute to leading the development of non-parametric methods in general for modelling all forms of count data of gene expression.

## Acknowledgement

The authors would like to thank Prof. Peter Green and Dr. Richard Goldstein for useful discussions. Also, we would like to thank P. G. Engstrom and colleagues for producing the public data we used in this paper. Source code implementing the methodology presented in this paper can be downloaded from the following link: https://bitbucket.org/DimitrisVavoulis/dgeclust.

**Funding:** This work was supported by grants EPSRC EP/H032436/1 and BBSRC G022771/1.

## References

1. Anders S, Huber W (2010) Differential expression analysis for sequence count data. Genome Biol 11: R106.
2. Auer PL, Doerge RW (2011) A Two-Stage Poisson Model for Testing RNA-Seq Data. Statistical Applications in Genetics and Molecular Biology 10:1- 26.
3. David M Blei, Michael I Jordan (2006) Variational inference for dirichlet process mixtures. Bayesian Analysis 1: 121-144.
4. Carninci P (2009) Is sequencing enlightenment ending the dark age of the transcriptome? Nat Methods 6: 711-713.
5. Nicole Cloonan, Alistair RRF, Gabriel Kolle, Brooke BAG, Geoffrey JF, et al. (2008) Stem cell transcriptome profiling via massive-scale mRNA sequencing. Nat Methods 5: 613-619.
6. Engström PG, Tommei D, Stricker SH, Ender C, Pollard SM, et al. (2012) Digital transcriptome profiling of normal and glioblastoma-derived neural stem cells identifies genes associated with patient survival. Genome Med 4: 1-76.
7. Hardcastle TJ, Kelly KA (2010) baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. BMC Bioinformatics 11: 422.
8. Ishwaran H, Lancelot FJ (2001) Gibbs Sampling Methods for Stick-Breaking Priors. Journal of the American Statistical Association 96:161-173.
9. Daxin Jiang, Chun Tang, Aidong Zhang (2004) Cluster analysis for gene expression data: A survey. IEEE Trans Knowl Data Eng 16: 1370-1386.
10. Kvam VM, Liu P, Si Y (2012) A comparison of statistical methods for detecting differentially expressed genes from RNA-seq data. Am J Bot 99: 248-256.
11. Langmead B, Hansen KD, Leek JT (2010) Cloud-scale RNA-sequencing differential expression analysis with Myrna. Genome Biol 11: R83.
12. Lu J, Tomfohr JK, Kepler TB (2005) Identifying differential expression in multiple SAGE libraries: an overdispersed log-linear model approach. BMC Bioinformatics 6: 165.
13. McCarthy DJ, Chen Y, Smyth GK (2012) Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. Nucleic Acids Res 40: 4288-4297.
14. Nagalakshmi U, Wang Z, Waern K, Shou C, Raha D, et al. (2008) The transcriptional landscape of the yeast genome defined by RNA sequencing. Science 320: 1344-1349.
15. Oshlack A, Robinson MD, Young MD (2010) From RNA-seq reads to differential expression results. Genome Biol 11: 220.
16. Papaspiliopoulos, Roberts GO (2008) Retrospective mcmc for dirichlet process hierarchical models. Biometrika 95: 169-186.
17. Robinson MD, Smyth GK (2007) Moderated statistical tests for assessing differences in tag abundance. Bioinformatics 23: 2881-2887.
18. Robinson MD, Smyth GK (2008) Small-sample estimation of negative binomial dispersion, with applications to SAGE data. Biostatistics 9: 321-332.
19. Shiraki T, Kondo S, Katayama S, Waki K, Kasukawa T, et al. (2003) Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. Proc Natl Acad Sci U S A 100: 15776-15781.
20. Srivastava S, Chen L (2010) A two-parameter generalized Poisson model to improve the analysis of RNA-seq data. Nucleic Acids Res 38: e170.
21. ‘t Hoen PA, Ariyurek Y, Thygesen HH, Vreugdenhil E, Vossen RH, et al. (2008) Deep sequencing-based expression analysis shows major advances in robustness, resolution and inter-lab portability over five microarray platforms. Nucleic Acids Res 36: e141.
22. Velculescu VE, Zhang L, Vogelstein B, Kinzler KW (1995) Serial analysis of gene expression. Science 270: 484-487.
23. S Walker (2007) Sampling the dirichlet mixture model with slices. Comm Statist Sim Comput 36: 45-54.
24. Wang L, Feng Z, Wang X, Wang X, Zhang X (2010) DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. Bioinformatics 26: 136-138.
25. Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet 10: 57-63.