Latent Process Decomposition of High-Dimensional Count Data

By

S. Srivastava and R.W. Doerge
Purdue University

Technical Report #11-03

Department of Statistics
Purdue University

September 2011

Latent Process Decomposition of High-Dimensional Count Data

Sanvesh Srivastava
Department of Statistics
Purdue University
sanvesh@gmail.com

R.W. Doerge
Department of Statistics
Purdue University
doerge@purdue.edu

September 9, 2011

Abstract

We present a novel approach to probabilistically model high-dimensional count data in an unsupervised way using a three-level hierarchical Bayesian model. Its application is explored in the context of next-generation sequencing data for the purpose of identifying subsets of genes with consistent expression patterns, and that explain a large portion of variability. Each sample is modeled as a finite mixture of Poisson random variables over an underlying set of latent variables that are assumed to correspond to biological functions. Each biological function is further modeled as an infinite mixture over an underlying set of biological function probabilities. We call this model *Latent Process Decomposition* (LPD). It combines ideas from machine learning and resampling-based methods, and uses a computationally efficient variational method for parameter estimation. The performance of LPD is investigated in both simulated and real data settings to demonstrate that it is a useful modular and extensible tool for identifying interesting genes for further exploration. LPD is implemented as an R/Bioconductor package called *themes*.

1 Introduction

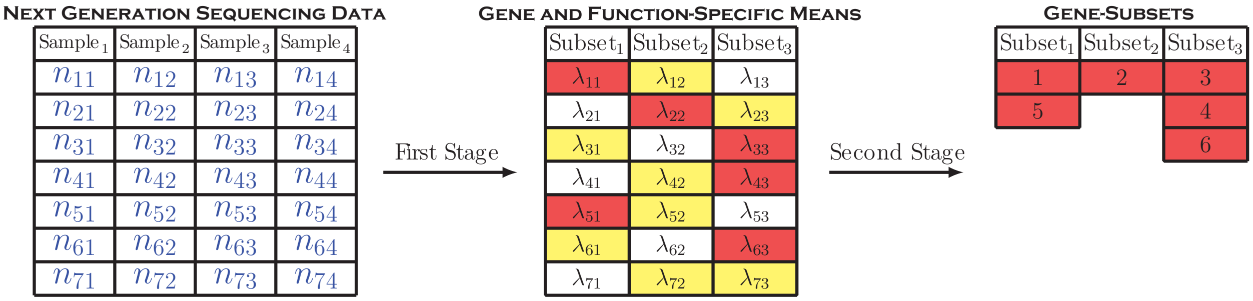
Next-generation sequencing (next-gen) technologies have emerged as a promising approach for exploring both cell organization and functionality, and have been used in a variety of fields, including genomics, transcriptomics, and epigenomics (Hayden, 2009; Metzker, 2009; Ng et al., 2009; Roach et al., 2010). Unlike data from earlier technologies (e.g., microarrays), data from next-gen technologies are highly replicable with little technical variation (Marioni et al., 2008), are in the form of discrete gene counts that represent the expression of each gene in the genome, and are similar to other high-dimensional data. Typically, there is a limited availability of samples (i.e., individuals) when compared to the number of predictors (i.e., genes), thus limiting the amount of sample-specific information relative to the feature-specific information. This is also known as the "large p small n" problem.

Research in high-dimensional genomic data gained momentum with the advent of microarrays. It has lead to significant advancements in the theory of multiple hypotheses testing (Efron et al., 2001; Efron, 2004, 2007, 2008), variable selection (Ishwaran and Rao, 2003; Efron et al., 2004; Yeung et al., 2005; Zou and Hastie, 2005; Friedman et al., 2010), and the use of false discovery rates (FDR) for multiple testing problems (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001; Genovese and Wasserman, 2002; Storey, 2003; Sun and Cai, 2007; Efron, 2008). In order to model high-dimensional genomic data Efron (2010) recommends approaches including empirical Bayesian methods that take advantage of information-borrowing across genes to compensate for limited availability of samples. Bayesian approaches (Baldi and Long, 2001; Broet et al., 2002; Ibrahim et al., 2002; Medvedovic et al., 2004; Newton et al., 2004) and penalized-likelihood based approaches (Tibshirani et al., 2005; Ma and Huang, 2007; Ma et al., 2007; Witten et al., 2009) have also been recommended to take advantage of information-borrowing amongst genes.

When contrasted with the issues found in microarray analysis, the issues for analyzing next-gen data are magnified simply because of the increased complexity of the data. Beyond the one central theme found in all differential gene expression approaches that calculate gene-wise test-statistics, shrinks them towards a common value, and adjusts the p-values for the modified test-statistics using FDR, next-gen data pose two main non-trivial problems. First, due to the non-normality of the data there are no equivalents of a t-test or an F-test (Casella and Berger, 2001). The approximate distribution of the test-statistic is determined by the asymptotic likelihood approximations, or by using exact tests (Agresti, 2002; Robinson and Smyth, 2007, 2008; Anders and Huber, 2010; Auer and Doerge, 2011). Second, as over-dispersion, small-counts, and zero-inflation are very common in next-gen data, the assumption of Poisson distribution on gene counts may not be justified (Vêncio et al., 2004; Thygesen and Zwinderman, 2006; Hardcastle and Kelly, 2010). And, even more frustrating is the fact that approaches that are very helpful tools for exploring structure in microarray data such as, the gene-shaving algorithm (Hastie et al., 2000), linear discriminant analysis (Dudoit et al., 2002; Guo et al., 2007), and nearest shrunken centroids algorithm (Tibshirani et al., 2003) have no equivalents for next-gen data. Pachter (2011) and Oshlack et al. (2010) provide excellent reviews of the current statistical methods.

2 Latent Process Decomposition

We present a probabilistic model – Latent Process Decomposition (LPD) – for unsupervised modeling of high-dimensional count data. Its application is explored for next-generation sequencing (herein next-gen) data, but is generalizable to many other applications. LPD is a generative three-level hierarchical Bayesian model that achieves a reduction of next-gen data into subsets of genes in two stages by combining ideas from machine learning and resampling-based methods. In the first stage, LPD adapts the Latent Dirichlet Allocation (LDA) algorithm (Blei et al., 2003) for count data and estimates the probability of individual genes belonging to each of the pre-specified latent classes (say K), assumed to correspond to biological functions. In the second stage, LPD adapts the Gap algorithm (Hastie et al., 2000) to obtain K subsets of genes corresponding to each of the functional classes (Figure 1). Although LDA has been adapted for microarray data (Rogers et al., 2005), approaches that combine methods from machine learning and classical statistics remain unexplored in next-gen applications. The benefit here is that LPD can be applied to count data of any sample size and is more flexible than classical clustering models. The model can be readily extended or embedded in models with the intent of analyzing more complex count data. Since the exact posterior distribution of the unknown parameters in LPD is intractable, we rely on empirical Bayesian methods to estimate the parameters of the model using computationally efficient variational algorithm (Bishop, 2006, chapter 10). Furthermore, LPD is easy to parallelize making it scalable for increasing sample size (Wang et al., 2009; Smola and Narayanamurthy, 2010). We will use the terminology of next-gen data throughout the paper to illustrate the biological motivation behind the model. This is especially useful when we introduce latent variables.


Next Generation Sequencing Data
First Stage
Gene and Function-Specific Means
Second Stage
Gene-Subsets

Sample_1	Sample_2	Sample_3	Sample_4
n_{11}	n_{12}	n_{13}	n_{14}
n_{21}	n_{22}	n_{23}	n_{24}
n_{31}	n_{32}	n_{33}	n_{34}
n_{41}	n_{42}	n_{43}	n_{44}
n_{51}	n_{52}	n_{53}	n_{54}
n_{61}	n_{62}	n_{63}	n_{64}
n_{71}	n_{72}	n_{73}	n_{74}

Subset_1	Subset_2	Subset_3
λ_{11}	λ_{12}	λ_{13}
λ_{21}	λ_{22}	λ_{23}
λ_{31}	λ_{32}	λ_{33}
λ_{41}	λ_{42}	λ_{43}
λ_{51}	λ_{52}	λ_{53}
λ_{61}	λ_{62}	λ_{63}
λ_{71}	λ_{72}	λ_{73}

Subset_1	Subset_2	Subset_3
1	2	3
5		4
		6

Figure 1: The Latent Process Decomposition algorithm (LPD) reduces the dimensionality of high-dimensional count data in two stages. This example illustrates four samples ($S = 4$) and seven genes ($G = 7$). In the first stage LPD estimates the probability of individual genes belonging to each of the apriori chosen number of subsets (three) that are assumed to correspond to biological functions. The gene and function-specific means matrix, $\boldsymbol{\Lambda}$, is estimated and contains the means of gene counts if they belong to the respective functional classes (discussed in Section 2.2.1). The row entries of $\boldsymbol{\Lambda}$ are proportional to the probability of individual genes belonging to the respective biological functions. The red cells have the largest means in the respective rows. In the second stage, LPD uses model-based and bootstrap-based methods to extract gene-subsets from $\boldsymbol{\Lambda}$ that have similar expression patterns and that explain a large amount of variability (discussed in Section 2.3).

2.1 Notation And Terminology

We assume that the total number of samples is S, and that there are G genes in each sample (Figure 1). Due to the unsupervised nature of the analysis, we ignore the treatment information associated with the samples. The observed data are the matrix of gene counts, $\boldsymbol{N}$. Following the widely used convention for genomic data, we assume that the columns of $\boldsymbol{N}$ represent the samples and rows of $\boldsymbol{N}$ correspond to the genes (Figure 1). We denote the gene counts for s^{th} sample by $\boldsymbol{N}_s$ (i.e., the s^{th} column of $\boldsymbol{N}$). The gene count for gene g in sample s is represented by n_{gs}. We assume apriori that there are K biological functions associated with the samples, and that the genes in each sample belong to one of the K biological functions. The association of a gene to a biological function can vary depending on the sample. Each sample has its own specific probability that its genes belong to the K biological functions. The biological functions, hereon referred to as functions, are modeled as latent variables in LPD.

1. Choose sample-specific functional probabilities $\boldsymbol{\theta}$ from Dirichlet($\boldsymbol{\alpha}$); $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$.

   For each gene, g, of the G genes:

2. Choose a biological function $k \in \{1, \ldots, K\}$ from Multinomial($\boldsymbol{\theta}$).

3. Given k, select the Poisson distribution with gene and function-specific mean, λ_{gk}, and generate gene count, n_{gs}, from Poisson(λ_{gk}).

Figure 2: The first stage of the Latent Process Decomposition (LPD) algorithm is a parametric hierarchical Bayesian model (Gelman et al., 2003) with three levels of hierarchy for generating the gene counts.

2.2 First Stage: The Generative Model

As a modified form of LDA, the first stage of LPD assumes gene counts in each sample, s, to be generated from an algorithm based on a parametric hierarchical Bayesian model with three levels of hieararchy (Figure 2). The functional probabilities, $\boldsymbol{\theta}$, are specific to sample s. They are generated from the underlying set of infinite functional probabilities of Dirichlet($\boldsymbol{\alpha}$). The algorithm assumes that we repeatedly sample functional memberships for genes in a sample, and generate the gene counts from Poisson distribution with mean chosen from the gene and function-specific matrix, $\boldsymbol{\Lambda}$ (Figure 1). The rows of $\boldsymbol{\Lambda}$ represent genes, the columns correspond to functions, and its dimensions are $G \times K$. The mean of gene g when it belongs to function k is represented by λ_{gk}, the $(g, k)^{th}$ element of $\boldsymbol{\Lambda}$.

The three-level hierarchical model (Figure 2) is a special case of a parametric hierarchical Bayesian model (Gelman et al., 2003). The first level consists of the K-dimensional Dirichlet distribution for sampling functional probabilities. The second level consists of the K-dimensional multinomial distribution that generates sample-specific functional associations of genes. The third level generates sample gene counts from a Poisson distribution with means depending on the genes' functional association. Due to the Hewitt-Savage theorem (Hewitt and Savage, 1955; Aldous, 1985), the hierarchical structure implies that the gene counts and functions are infinitely exchangeable within a sample. The first stage of LPD can be illustrated by using graphical models (Bishop, 2006, chapter 8); such representations are common in machine learning literature. The graphical model for first stage of LPD is same as the LDA model (see Figure 1, Section 3, Blei et al. (2003)).

We make several simplifying assumptions in the first stage of LPD. First, we assume that the dimensionality of the Dirichlet distribution, K, is fixed and known. We propose a method for validating K using N-fold cross-validation in Section A.5, and favor the choice of K that leads to gene-subsets with sparse number of genes. Second, the gene and function-specific mean parameters, $\boldsymbol{\Lambda}$, are assumed to be fixed. In Section 2.2.1, we treat $\boldsymbol{\Lambda}$ as random, and use a gamma prior to shrink the parameter estimates. Finally, we assume that the total number of gene counts in a sample (i.e., the population from which data samples are taken), or library-size, is fixed because library-size is not associated with other data-generating parameters ($\boldsymbol{\theta}$ and $\boldsymbol{\Lambda}$). Certainly, we could impose a layer of prior distributions for modeling the randomness, but we choose to ignore this for ease of parameter estimation and to facilitate interpretation.

2.2.1 Estimation of Parameters in the First Stage

Parameter estimation in the first stage of LPD algorithm (Figure 2) is motivated from the LDA algorithm (Blei et al., 2003) and the Latent Process Decomposition algorithm for microarrays (Rogers et al., 2005). The model parameters are the gene and function-specific matrix of mean parameters, $\boldsymbol{\Lambda}$, for generating gene counts from Poisson distribution, and the K dimensional parameter $\boldsymbol{\alpha}$ of the Dirichlet distribution for generating sample-specific functional probabilities. From hereon we refer $\boldsymbol{\Lambda}$ and $\boldsymbol{\alpha}$ as model parameters.

We assume the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\Lambda}$ are random, and that

$$p(\boldsymbol{\alpha}, \boldsymbol{\Lambda} | \boldsymbol{N}) \propto p(\boldsymbol{N} | \boldsymbol{\alpha}, \boldsymbol{\Lambda}) p(\boldsymbol{\alpha}, \boldsymbol{\Lambda}), \tag{1}$$

where $p(\boldsymbol{N}|\boldsymbol{\alpha}, \boldsymbol{\Lambda})$ is the likelihood of next-gen data given the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\Lambda}$. $p(\boldsymbol{\Lambda}, \boldsymbol{\alpha})$ is the prior distribution for $\boldsymbol{\alpha}$ and $\boldsymbol{\Lambda}$. Our aim is to obtain parameter estimates, $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\Lambda}}$, that maximize the posterior distribution of the parameters given next-gen data, $\boldsymbol{N}$

$$\begin{aligned}\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\Lambda}} &= \arg\max_{\boldsymbol{\alpha},\boldsymbol{\Lambda}} p(\boldsymbol{\alpha}, \boldsymbol{\Lambda}|\boldsymbol{N}), \\ &= \arg\max_{\boldsymbol{\alpha},\boldsymbol{\Lambda}} \log p(\boldsymbol{\alpha}, \boldsymbol{\Lambda}|\boldsymbol{N}), \qquad (2) \\ &= \arg\max_{\boldsymbol{\alpha},\boldsymbol{\Lambda}} [\log p(\boldsymbol{N}|\boldsymbol{\alpha}, \boldsymbol{\Lambda}) + \log p(\boldsymbol{\alpha}, \boldsymbol{\Lambda})]. \qquad (3)\end{aligned}$$

In (3), $\log p(\boldsymbol{N}|\boldsymbol{\alpha}, \boldsymbol{\Lambda})$ is found by using the fact that the first stage of LPD is a hierarchical Bayesian model. Specifically, if gene g in sample s belongs to function k, the gene count, n_{gs}, follows a conditional Poisson distribution with mean λ_{gk}. Recall, gene counts are generated from a finite Poisson mixture with function associations sampled from Multinomial($\boldsymbol{\theta}$). Therefore, marginalizing over $\boldsymbol{\theta}$ from the mixture distribution, $p(\boldsymbol{\theta}, \boldsymbol{N}|\boldsymbol{\alpha}, \boldsymbol{\Lambda})$, and taking the log gives the log-likelihood of $\boldsymbol{N}$ given the model parameters

$$\log p(\boldsymbol{N}_s|\boldsymbol{\alpha}, \boldsymbol{\Lambda}) = \log \int_{\boldsymbol{\theta}} \left\{ \prod_{g=1}^{G} \sum_{k=1}^{K} \theta_k \mathcal{P}(n_{gs}|k, \lambda_{gk}) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \right\} d\boldsymbol{\theta}, \qquad (4)$$

$$\log p(\boldsymbol{N}|\boldsymbol{\alpha}, \boldsymbol{\Lambda}) = \sum_{s=1}^{S} \log p(\boldsymbol{N}_s|\boldsymbol{\alpha}, \boldsymbol{\Lambda}). \qquad (5)$$

In (3), $\log p(\boldsymbol{\alpha}, \boldsymbol{\Lambda})$ depends on the choice of priors. We impose a gamma prior on the elements of $\boldsymbol{\Lambda}$, λ_{gk}. We should impose separate priors on each column, $\boldsymbol{\Lambda}_k$, but this leads to complicated and intractable posterior calculations. Therefore, we impose the same gamma prior on all the elements of matrix $\boldsymbol{\Lambda}$

$$p(\lambda_{gk}) \propto \lambda_{gk}^{\eta-1} e^{-\beta\lambda_{gk}} \quad \text{for } g = 1, \ldots, G \text{ and } k = 1, \ldots, K, \qquad (6)$$

$$\log p(\boldsymbol{\alpha}, \boldsymbol{\Lambda}) \propto \sum_{g=1}^{G} \sum_{k=1}^{K} \log p(\lambda_{gk}). \qquad (7)$$

A maximum likelihood approach would impose an uniform prior on λ_{gk} in (6). The use of gamma priors shrink the maximum likelihood estimates of λ_{gk} towards a common value, and acts as a penalty for model complexity, prevents over-fitting, and makes the estimates robust. We impose a uniform prior on $\boldsymbol{\alpha}$ since this level of hierarchy is for any next-gen experiment, and these parameters have little effect in the extraction of gene-subsets.

The model parameters are estimated from the observed data using a parametric empirical Bayesian approach (Morris, 1983). Since the log-posterior in (3) cannot be represented in an explicit form due to the coupling between the functional probabilities, $\boldsymbol{\theta}$, and gene and function-specific means, $\boldsymbol{\Lambda}$, it is impossible to find the theoretical estimates of the model parameters. Instead, we use approximate techniques for estimation of parameters. There are a host of approximate techniques that can be used. Specifically, Laplace approximations (Tierney et al., 1989), Markov chain Monte Carlo methods (Robert and Casella, 2004), and variational method (Bishop, 2006, chapter 10). We employ convexity-based and computationally efficient variational method since it provides interpretable parameter updates that are similar to EM algorithm updates (Dempster et al., 1977). These methods use Jensen's inequality to obtain an adjustable lower bound for the log-posterior and guarantee convergence to a local optimum. The variational method introduces a special set of parameters – variational parameters – $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$, to decouple $\boldsymbol{\theta}$ and $\boldsymbol{\Lambda}$ and to obtain a tractable family of lower bounds for the log-posterior in (3). The parameter $\boldsymbol{\Phi}$ is a $G \times S \times K$ array with entries, ϕ_{gsk}, that denote the probability that gene g in the s^{th} sample belongs to the k^{th} function. The $(s,k)^{th}$ entry, γ_{sk}, of the $S \times K$ matrix, $\boldsymbol{\Gamma}$, is proportional to the probability of s^{th} sample belonging to the k^{th} functional class.

The iterative updates to the estimates of variational parameters ϕ_{gsk} and γ_{sk}, are (Appendix, Section A.2.1, (37) – (41) and Section A.3.4, (49) – (54))

$$\phi_{gsk} = \frac{\mathcal{P}(n_{gs}|k, \lambda_{gk}) \exp[\boldsymbol{\Psi}(\gamma_{sk})]}{\sum_{k'=1}^{K} \mathcal{P}(n_{gs}|k', \lambda_{gk'}) \exp[\boldsymbol{\Psi}(\gamma_{sk'})]}, \tag{8}$$

$$\gamma_{sk} = \alpha_k + \sum_{g=1}^{G} \phi_{gsk}, \tag{9}$$

where $\mathcal{P}(n_{gs}|k, \lambda_{gk})$ denotes the Poisson density with mean, λ_{gk}, evaluated at n_{gs}, where α_k is assumed to be known, and where $\boldsymbol{\Psi}(z)$ is the digamma function (Abramowitz and Stegun, 1970). The iterative update for model parameter λ_{gk} is obtained as (Appendix, Section A.3.2, (46) – (48))

$$\hat{\lambda}_{gk} = \frac{\sum_{s=1}^{S} \phi_{gsk} n_{gs} + \eta - 1}{\sum_{s=1}^{S} \phi_{gsk} + \beta}. \tag{10}$$

The parameters η and β in (10) are obtained as the maximum likelihood estimates of shape and rate parameters of a gamma distribution from which $\boldsymbol{N}$ are sampled (Choi and Wette, 1969). Later, we use $\hat{\boldsymbol{\Lambda}}$ to extract K gene-subsets. The iterative updates for estimating model parameters, $\boldsymbol{\alpha}$, are derived by using Newton-Raphson method (Appendix, Section A.3.3)

$$\boldsymbol{\alpha}_{new} = \boldsymbol{\alpha}_{old} - \mathbf{H}(\boldsymbol{\alpha}_{old})^{-1}\mathbf{g}(\boldsymbol{\alpha}_{old}), \tag{11}$$

where $\mathbf{H}$ and $\mathbf{g}$ are the Hessian and gradient for the update. We calculate the update using operations of $\boldsymbol{O}(n)$ time complexity by taking advantage of the diagonal structure of the Hessian, and by avoiding the inversion of $\mathbf{H}$, which takes $\boldsymbol{O}(n^3)$ time (Appendix, Section A.3.3). Convergence and stability of this algorithm are discussed in Appendix, Section A.4.

A maximum-likelihood approach that uses a uniform prior on $\log p(\boldsymbol{\alpha}, \boldsymbol{\Lambda})$ in (3), ensures that all the parameter updates remain the same, except for λ_{gk} (Appendix, Section A.3.1, (28) – (43))

$$\hat{\lambda}_{gk} = \frac{\sum\limits_{s=1}^{S} \phi_{gsk} n_{gs}}{\sum\limits_{s=1}^{S} \phi_{gsk}}. \tag{12}$$

2.2.2 Interpreting the First Stage of LPD Estimated Parameters

The main motivation for relying on variational method for parameter estimation (Section 2.2.1) is that it facilitates interpretation of the parameter estimates. In the first stage of LPD, we estimate two sets of parameters – variational parameters and model parameters. Here we describe the interpretation of the two sets of parameters.

The probability that gene g in sample s belongs to the function k is denoted by ϕ_{gsk} (8). The probabilities, $\{\phi_{gsk}\}_{k=1}^{K}$, are a distribution over functions for gene g in sample s. Therefore, the sum of ϕ_{gsk} across all the K functions is 1, and the sum across all the genes is the expected number of genes in sample s from function k ($\sum_{g=1}^{G} \phi_{gsk}$). The variational parameter, γ_{sk}, is proportional to the probability that s^{th} sample belongs to the k^{th} function class. Using (9), $\gamma_{sk} - \alpha_k$ is a measure of the expected number of genes belonging to function k for the s^{th} sample (same as $\sum_{g=1}^{G} \phi_{gsk}$). This can be used as an unsupervised way of associating samples to functions, and for checking the convergence of iterative updates in parameter estimation (Section 2.2.1).

With respect to any next-generation sequencing experiment, the probability that a gene belongs to function k is proportional to α_k. The parameter, λ_{gk}, is the expected value of count of gene g when it belongs to the function k. It is the weighted average of n_{gs} across all samples with ϕ_{gsk} as weights (12). For a gene g, λ_{gk} is proportional to the probability that the gene belongs to the function k, hence the probability that gene g belongs to the function k is $\frac{\lambda_{gk}}{\sum_{k=1}^{K}\lambda_{gk}}$.

2.2.3 Relation With Clustering Procedures

Classical clustering models are hierarchical models with two levels of hierarchy (Blei et al., 2003). For example, in the Dirichlet-Poisson clustering model, the weights of the mixture (or probabilities of components) are generated only once from a Dirichlet distribution. Based on these weights a function (latent class) is chosen that is same for all the genes in the sample. Similar to the first stage of LPD, this function defines the gene-specific mean from which gene counts in the sample are generated. However, this approach results in each sample being associated with *only one* function, and this may be too restrictive for high-dimensional count data, including next-gen data. Our proposed alternative allows the first stage of LPD to associate genes in a sample to different functional associations by repeatedly sampling functions from Multinomial($\boldsymbol{\theta}$) for each gene in the sample (second level of hierarchy; Figure 2). It is the additional level of hierarchy in the first stage of LPD that makes it more flexible to adapt to the high-dimensional next-gen data.

2.3 Second Stage: Extraction of Subsets

In the second stage, LPD uses the columns of $\hat{\boldsymbol{\Lambda}}$ (10) to extract K gene-subsets. The second stage ensures that the expression patterns within each subset are consistent. Here, we describe an algorithm for extracting gene-subsets based on the Gap algorithm (Hastie et al., 2000). This method is computationally intensive, but has the advantage of being robust and independent of any restrictive distributional assumptions. In order to achieve this reduction, the elements of $\hat{\boldsymbol{\Lambda}}$ are divided by their row-means, and transformed to a matrix of probabilities, $\boldsymbol{P}^{'}$ (Section 2.2.2). Each element, $P^{'}_{gk}$, of the matrix $\boldsymbol{P}^{'}$ is further transformed to its logit, $\log\frac{P^{'}_{gk}}{1-P^{'}_{gk}}$, followed by centering and scaling of the columns by column means and standard deviation of columns, respectively, to yield $\boldsymbol{P}$. All our subsequent discussions are based on $\boldsymbol{P}^{'}$ and $\boldsymbol{P}$.

Gene g is selected in the subset, $\mathcal{S}_k$, if $P^{'}_{gk}$ is greater than a cutoff c_k; genes satisfying this condition are assumed to have similar expression patterns. The cutoffs, $\boldsymbol{c} = (c_1, \ldots, c_K)$, are selected based on the columns of $\boldsymbol{P}^{'}$ since its entries are bounded between 0 and 1. Once the genes are selected, their corresponding values from $\boldsymbol{P}$ are used.

Much like any analysis of variance (ANOVA), the within and between component variances can be partioned. Specifically,

$$V_W = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{|S_{c_k}|}\sum_{g\in S_{c_k}}(P_{gk}-\bar{P}_{.k})^2\right], \tag{13}$$

$$V_B = \frac{1}{K}\sum_{k=1}^{K}(\bar{P}_{.k}-\bar{P}_{..})^2, \tag{14}$$

$$V_T = V_W + V_B, \tag{15}$$

$$R^2 = \frac{V_B}{V_T} = \frac{\frac{V_B}{V_W}}{1+\frac{V_B}{V_W}}. \tag{16}$$

The measure V_W (13) is equivalent to the *within group variance* measure in ANOVA for K fixed effects. It measures the variability of each gene in a gene-subset about the subset-average, averaged over all the subsets. Similarly, V_B (14) is equivalent to the *between group variance*, and V_T (15) is the *total variance* in ANOVA. We define R^2, the fraction of variance explained by the K gene-subsets corresponding to $\boldsymbol{c}$ (16). A high value of R^2 implies that the K subsets explain a large fraction of variability in $\hat{\boldsymbol{\Lambda}}$, with similar expression patterns within each subset. R^2 uses $\boldsymbol{P}$, rather than $\boldsymbol{P}^{'}$, since the elements of $\boldsymbol{P}$ are closer to a normal distribution, similar to the ANOVA assumptions.

For a cutoff $\boldsymbol{c}$, let $D_{\boldsymbol{c}}$ be the R^2 measure for the corresponding gene-subsets. A modified form of Gap algorithm in the Gene-Shaving algorithm (Hastie et al., 2000) estimates the difference between $D_{\boldsymbol{c}}$ and R^2 obtained as a result of random association between the genes (rows) and functional classes (columns), $\bar{D}^*_{\boldsymbol{c}}$. Let $\boldsymbol{P}^{*b}$ be the bootstrapped $\boldsymbol{P}$ matrix obtained by sampling with replacement from the elements of each of the K columns of $\boldsymbol{P}$, respectively. B such matrices, indexed by $b = 1, \ldots, B$, are formed. Let $D^{*b}_{\boldsymbol{c}}$ be the R^2 measure for the gene-subsets obtained from $\boldsymbol{P}^{*b}$. The average of $D^{*b}_{\boldsymbol{c}}$ over b is $\bar{D}^*_{\boldsymbol{c}}$, and the Gap function for cutoff $\boldsymbol{c}$ is defined by

$$\text{Gap}(\boldsymbol{c}) = D_{\boldsymbol{c}} - \bar{D}^*_{\boldsymbol{c}}. \tag{17}$$

We obtain Gap($\boldsymbol{c}$) for all the cutoffs, $\boldsymbol{c}$, in the K dimensional bounded set $[0,1]^K$ (as $0 \le c_k \le 1,\ k \in \{1,\ldots,K\}$). The optimal cutoff value, $\hat{\boldsymbol{c}}$, produces the largest Gap

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \text{Gap}(\boldsymbol{c}). \tag{18}$$

This method estimates the value of $\hat{\boldsymbol{c}}$ that minimizes the chances of R^2 for the K gene-subsets as a consequence of random association between genes (rows) and functional classes (columns). In general, Gap($\boldsymbol{c}$) is non-smooth, so instead of using numerical optimization techniques for finding

$\hat{\boldsymbol{c}}$, we take advantage of the entries of $\boldsymbol{P}^{'}$ that lie in the bounded interval $[0, 1]$. We choose C values of cutoffs in the interval $(0, 1)$ for each of the K columns, and calculate C^K values of Gap($\boldsymbol{c}$) for the C^K possible values of $\boldsymbol{c}$. From these cutoffs, $\hat{\boldsymbol{c}}$ is the value that maximizes Gap($\boldsymbol{c}$). This is a grid search distributed across all possible C^K cutoffs in a K dimensional grid. For a large value of C, this is a good practical approximation to (18). This also illustrates the advantage of using $\boldsymbol{P}^{'}$ for selecting cutoffs because a grid search within a bounded interval $[0, 1]$ is more appropriate, and faster, than a grid search in $(-\infty, +\infty)$.

3 Examples of LPD

3.1 Application of LPD to Simulated Data

We rely on simulated high-dimensional count data with 4 functional classes, for 1000 genes, and 8 samples using the generative model in Figure 2. The specific parameter settings were chosen based on the Bioconductor (Gentleman et al., 2004) package *edgeR* (Robinson et al., 2010, Section 12) such that the gene and function-specific mean parameters were similar for all the genes, with the exception that the first 100 genes had means 10 times greater, across all samples, to make the data realistic. We chose K, the total number of functions in the data, to be 4 based on the uniformity of the size and sparsity of gene-subsets across functional classes.

The ratio, $\frac{\sum_{s=1}^{S}\sum_{g=1}^{G}\phi_{gsk}}{\sum_{k=1}^{K}\sum_{s=1}^{S}\sum_{g=1}^{G}\phi_{gsk}}$, estimates the expected proportion of genes that belong to the functional class k (Section 2.2.2). The true proportions are found by averaging the true gene and function-specific mean proportions across all samples. Because of the assumption of exchangeability of functions, the estimated functional classes (i.e., columns) of $\hat{\boldsymbol{\Lambda}}, \boldsymbol{P}$, and $\boldsymbol{P}^{'}$ represent a permutation of the true functional class associations. Based on this, the true and estimated gene-proportions in the four functional classes are ordered in Figure 3 to make the pattern clearer. The estimated function proportions are shrunk towards a common value, but retain the pattern of the true gene proportions across functional classes. Figure 4 illustrates the density plot of elements of $\boldsymbol{P}^{'}$ transformed to their logits (i.e., $\log\frac{P^{'}_{gk}}{1-P^{'}_{gk}}$), conditioned on the functional classes (columns of $\boldsymbol{P}^{'}$), for the true and estimated case. Although variational approximation is used for parameter estimation ($\hat{\boldsymbol{\Lambda}}$), the density estimates for the columns of $\boldsymbol{P}^{'}$ agree very closely with the truth, showing that variational method is reasonably accurate despite being an approximate method. The second

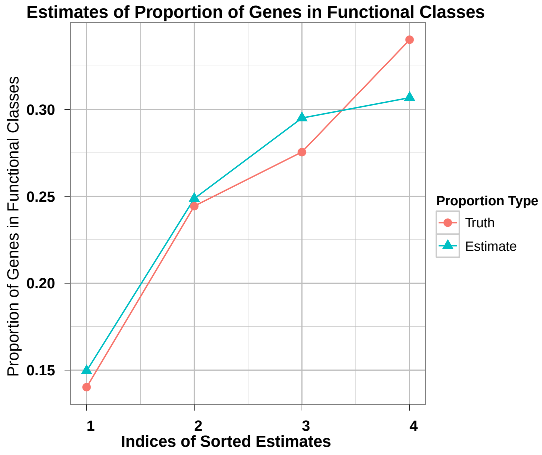


Figure 3: True and estimated proportion of genes in four functional classes. The estimated (aqua) and true (peach) proportion of genes in the respective functional classes are represented on the y-axis. The true functional classes are unidentifiable from the estimated functional classes, therefore the x-axis represents the indices of the sorted true and estimated gene-proportions for the four functional classes. This identifies the patterns for the true and estimated case. The overall pattern for the true and estimated cases look similar, except the estimated values are shrunk towards a common value due to the hierarchical modeling. The maximum value is shrunk the most, and it underestimates the true functional proportion by a small quantity. The remaining estimates slightly overestimate the true proportions.

stage of LPD uses $\hat{\boldsymbol{\Lambda}}$ to extract gene-subsets with $C = 9$ and $B = 500$ (Section 2.3). The four subsets contain 2, 4, 3, and 31 genes, respectively. The first three subsets contain genes that belonged to the first and second functional classes with high probability, and the fourth subset contains genes with high probability of belonging to the second and third functional classes.

This simulation and its results also serve to illustrate the limitations of LPD. Due to the exchangeability assumptions of LPD (Section 2.2), it is unable to associate genes with their true functional classes. Instead, LPD groups genes that belong to the same functional class with high probability. In cases where genes are associated with multiple functions with similar probabilities, LPD will not extract these genes because of similar probabilities of association to all the functional classes.

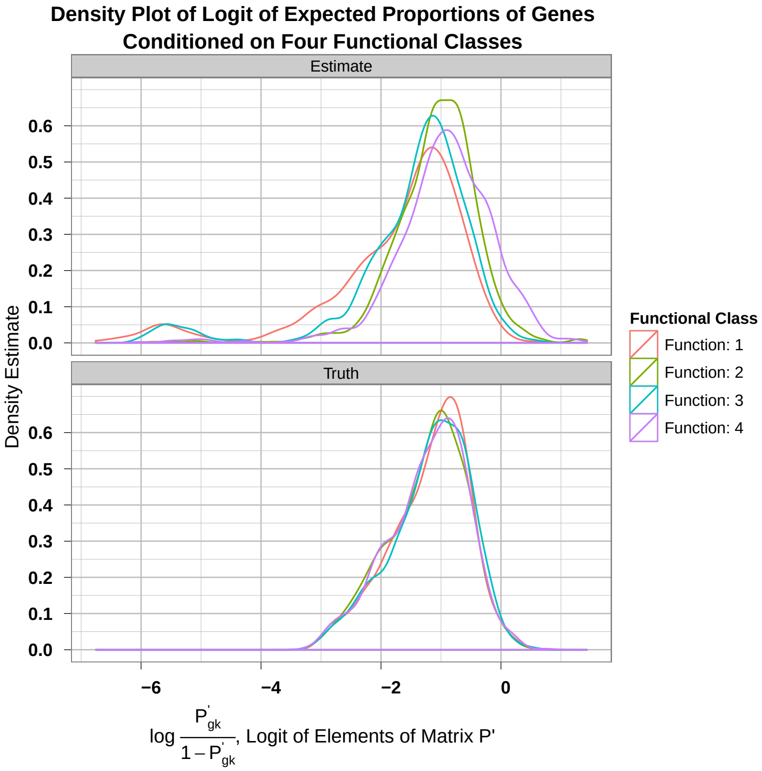


Figure 4: Density plot of the logit of elements of $\boldsymbol{P}'$ matrix ($\log \frac{P'_{gk}}{1-P'_{gk}}$) for the true and estimated case conditioned on the four functional classes. The plots are colored according to the columns of $\boldsymbol{P}'$, that are assumed to correspond to biological functions. The x-axis represents the logits of the entries of $\boldsymbol{P}'$ (Section 2.3) and the y-axis represents the density estimates of the columns of $\boldsymbol{P}'$, respectively. The estimated and true cases are in the two panels, showing that the approximate and true density estimates are very similar for the four functional classes.

3.2 Application of LPD to Yeast RNA-Sequencing Data

We apply LPD to RNA-sequencing data from *Saccharomyces cerevisiae* (yeast) (Nagalakshmi et al., 2008). The experiment consists of six samples, two library protocols (treatments), *dT* and *RH*, and three samples, respectively. Corresponding to the original analysis (labeled *dT_ori* and *RH_ori*) in each of the two treatments, there is a technical replicate (same culture, labeled *dT_tech* and *RH_tech*) and a biological replicate (different culture, labeled *dT_bio* and *RH_bio*). We will extract gene-subsets using LPD and then compare our results with the results from a differential gene expression analysis that used *DESeq* package (Anders and Huber, 2010). We will also compare our results to the results obtained from classification of samples to clusters using hierarchical clustering of $\log_2$-transformed data.

3.2.1 Application of LPD to Extract Gene-Subsets

LPD was fit to the data based on different values of K which in turn provided gene-subsets. We chose K based on the uniformity of the size of gene-subsets across functional classes, and rejected values of K for which any of the two gene-subset sizes varied by a factor of 10, or the size of any subset exceeded 1000 genes. For values of $K \leq 3$, at least one of the gene-subsets included approximately 1000 genes. For values of $K \geq 5$, the majority of gene-subset sizes were of the order of 1000. The differential between these values is related to under-parametrization and over-parametrization of LPD when $K \leq 3$ and $K \geq 5$, respectively. We chose K to be 4 since it led to gene-subsets with balanced number of genes across subsets. Interestingly, the method for validating K in Section A.5 failed to determine any unique value. Figure 5 illustrates the patterns of expected proportion of genes in the four functional classes. Samples *dT_ori*, *RH_ori*, and *RH_bio* show similar patterns across the four functional classes with the greatest proportion of genes belonging to functional class 3. The remaining three samples, *dT_bio*, *dT_tech*, and *RH_tech*, have the largest expected proportion of genes from functional classes 1, 4, and 2, respectively. The four subsets extracted using LPD contain 30, 14, 90, and 477 genes, respectively, out of the total 7,124 genes.

3.2.2 Comparison with *DESeq* Results

For comparison purposes we employed *DESeq* (Anders and Huber, 2010) to extract 196 differentially expressed genes (FDR was controlled at 5%), and then compared the results to the gene-subsets obtained using LPD. The number of genes, of the 196, in the four gene-subsets that are also differentially expressed are 1, 0, 2, and 23, respectively. Since the gene-subsets extracted using LPD, and the differentially expressed genes are interesting genes for further exploration, we recommend that the genes selected from both these analyzes are better candidates for further exploration (see Section 4.2, Supplements).

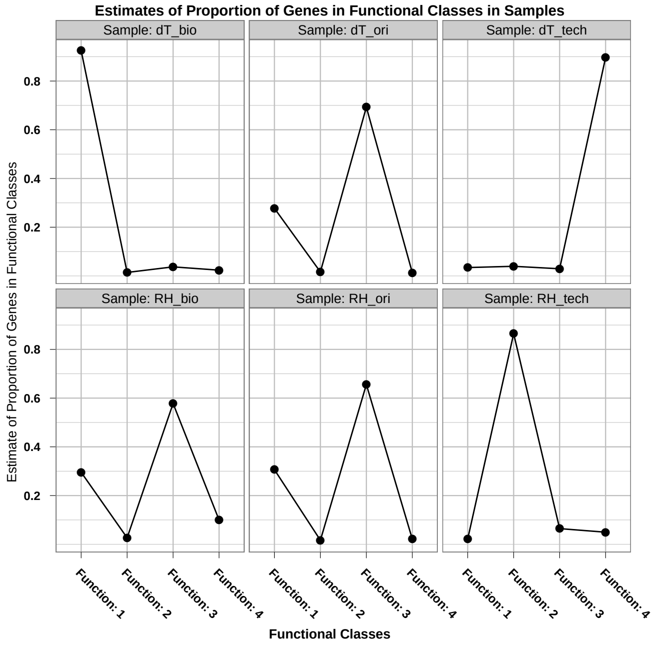


Figure 5: Expected proportions of genes in the four functional classes in the yeast data (Nagalakshmi et al., 2008), conditioned on samples. The functional classes vary along the x-axis, the proportions of genes belonging to the respective functional classes are on the y-axis denoted by a dot, and samples vary along the panels. The dots are connected by lines for visual clarity of patterns. The samples *dT_ori*, *RH_ori*, and *RH_bio* have similar patterns of expected gene proportion across the functional classes. Other samples have their maximum expected proportion of genes in the remaining functional classes.

3.2.3 Comparison with Hierarchical Clustering

Hierarchical clustering is a popular method of exploring the structure in genomic data, and has been used on $\log_2$-transformed next-gen data (Severin et al., 2010) for assigning samples to clusters. Although hierarchical clustering is unrelated to LPD, we used hierarchical clustering in R (R Development Core Team, 2011) to assign samples to four clusters to make the results of hierarchical clustering and LPD comparable. Table 1 shows the assignment of samples to four clusters using hierarchical clustering. The pattern of expected gene proportion in the four functional classes in Figure 5 is similar to the cluster assignments in Table 1. For example, *dT_ori* and *RH_ori* have the same profiles for expected gene proportions across functional classes in Figure 5, and are assigned

Sample	*dT_bio*	*dT_ori*	*dT_tech*	*RH_bio*	*RH_ori*	*RH_tech*
Cluster	1	2	3	4	2	3

Table 1: Results of performing hierarchical clustering on $\log_2$-transformed yeast data (Nagalakshmi et al., 2008). Samples are assigned to four clusters. The cluster assignments of samples have similar patterns as observed in Figure 5 using LPD. Specifically, samples *dT_ori* and *RH_ori* show similar patterns of estimated proportion of genes across the four functional classes, and both are assigned to cluster 2 using hierarchical clustering.

to the same cluster 2 by hierarchical clustering. In addition to the providing results similar to hierarchical clustering, LPD also provides estimates of variational parameters that provide valuable insights about the data (Section 2.2.2). Specifically, given the additional information, LPD provides more informative results compared to the hierarchical clustering of the same data.

4 Discussion

Count data have been analyzed using a variety of approaches and are often thought of as a special case of generalized linear models (glm) (Agresti, 2002). However, applying glm to high-dimensional count data such as next-gen data gives rise to non-trivial issues that are the result of a small number of samples when compared to the number of predictors (that is, genes). Latent Process Decomposition (LPD), provides a probabilistic approach for exploring the structure of high-dimensional count data, and is well suited to address next-gen data. LPD is a three-level hierarchical Bayesian model that assumes infinite exchangeability of biological functions and genes.

LPD obtains gene-subsets that explain a large portion of variability in next-gen data, with similar expression patterns between the members of a subset. Similar ideas about finding groups of differentially expressed gene-sets have been explored starting with gene-set enrichment analysis (Subramanian et al., 2005), and then generalized to gene-set analysis (GSA) (Efron and Tibshirani, 2007). We plan to investigate the relationship between enriched gene-subsets obtained from GSA and gene-subsets obtained from LPD. Furthermore, we plan to investigate the choice of K, the total number of biological functions, which is currently assumed to be fixed and known apriori. While we have proposed a method for validating K using N-fold cross-validation (Section A.5), we realize that it is better to choose K adaptively by methods used in Bayesian nonparametrics, specifically the Chinese Restaurant Process (Aldous, 1985; Ghosh and Ramamoorthi, 2003; Hjort et al., 2010).

The real power of LPD lies in its extensibility, flexibility, and modularity. LPD can be easily embedded in a hierarchy as a basic structure to produce complicated models for count data and next-gen data, in particular. This is particularly valuable when LPD-specific assumptions, like exchangeability of functions and genes, are not justified. For example, LPD can be embedded in appropriate hierarchical models to account for any biological annotation of the data such as treatment information or dependence between the functional classes, that was ignored in this work. As a further extension, time-course experiments can be easily modeled using multiple LPDs, with each LPD modeling next-gen data at a particular instant of time during the course of the experiment. This naturally captures the exchangeability of functions and genes at a particular time-point, but not across multiple time-points.

5 Conclusion

We have presented the Latent Process Decomposition (LPD) algorithm for the purpose of combining ideas from machine learning (Latent Dirichlet Allocation) and classical statistics (Gap algorithm) to explore the latent structure in high-dimensional count data, specifically next-generation sequencing data. LPD achieves a balance between theoretical and computational ideas. Due to the biological motivation behind the model LPD is able to provide interpretable model and variational parameters that other clustering algorithms cannot, and is able to extract subsets of genes that explain a large portion of variability, with similar expression patterns among the members of a subset. We anticipate that the advantage of the reduction in data dimensionality will only improve as the sample size increases with decreasing sequencing cost, and as projects such as, *1000 Genomes Project* (Stein, 2010) take hold.

We have implemented the LPD algorithm in the form of C++ optimized R/Bioconductor package called *themes*, that can be used in modeling next-generation sequencing data and extending the LPD model for other complex biological data.

6 Acknowledgment

We thank Doug Baumann for helpful comments on an earlier version of this manuscript. This work is funded in part by a National Science Foundation (DBI-0733857) grant to RWD and her colleagues. SS benefited immensely from discussions with Professor J.K. Ghosh about exchangeability and Bayesian analysis. All the plots in this paper are made using ggplot2 (Wickham, 2009). The R/Bioconductor package, *themes*, is available at `www.stat.purdue.edu/~doerge/`.

A Appendix

We provide the derivation for the variational inference algorithm that estimates the variational and model parameters in the first stage of Latent Process Decomposition (LPD) algorithm using iterative updates. The derivation adapts the Appendix of Blei et al. (2003) and Rogers et al. (2005) for Poisson (count) data.

A.1 Log-likelihood

The exchangeability of samples implies that the likelihood and log-likelihood for next-gen data are

$$p(\boldsymbol{N}|\boldsymbol{\alpha},\boldsymbol{\Lambda}) = \prod_{s=1}^{S} p(\boldsymbol{N}_s|\boldsymbol{\alpha},\boldsymbol{\Lambda}), \tag{19}$$

$$\log p(\boldsymbol{N}|\boldsymbol{\alpha},\boldsymbol{\Lambda}) = \sum_{s=1}^{S} \log p(\boldsymbol{N}_s|\boldsymbol{\alpha},\boldsymbol{\Lambda}). \tag{20}$$

For sample s, the likelihood marginalizes over the latent parameters, $\boldsymbol{\theta}$, makes LPD more flexible, and characterizes latent structure in the data.

$$p(\boldsymbol{N}_s|\boldsymbol{\alpha},\boldsymbol{\Lambda}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{N}_s|\boldsymbol{\theta},\boldsymbol{\Lambda})p(\boldsymbol{\theta}|\boldsymbol{\alpha})d\boldsymbol{\theta}. \tag{21}$$

Expanding $p(\boldsymbol{N}_s|\boldsymbol{\theta},\boldsymbol{\Lambda})$

$$p(\boldsymbol{N}_s|\boldsymbol{\theta},\boldsymbol{\Lambda}) = \prod_{g=1}^{G} p(n_{gs}|\boldsymbol{\theta},\boldsymbol{\Lambda}), \tag{22}$$

$$p(\boldsymbol{N}_s|\boldsymbol{\theta},\boldsymbol{\Lambda}) = \prod_{g=1}^{G}\sum_{k=1}^{K} \theta_k \mathcal{P}(n_{gs}|k,\lambda_{gk}), \tag{23}$$

$$\log p(\boldsymbol{N}_s|\boldsymbol{\theta},\boldsymbol{\Lambda}) = \sum_{g=1}^{G}\log\sum_{k=1}^{K} \theta_k \mathcal{P}(n_{gs}|k,\lambda_{gk}). \tag{24}$$

Under the assumption that gene g in sample s belongs to the function k, we denote the Poisson density with mean λ_{gk} evaluated at n_{gs} as $\mathcal{P}(n_{gs}|k, \lambda_{gk})$. Substituting (21) in (20), the log-likelihood of next-gen data reduces to

$$\begin{aligned}
\log p(\boldsymbol{N}|\boldsymbol{\alpha}, \boldsymbol{\Lambda}) &= \sum_{s=1}^{S} \log \int_{\boldsymbol{\theta}} p(\boldsymbol{N}_s|\boldsymbol{\theta}, \boldsymbol{\Lambda}) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) d\boldsymbol{\theta}, && (25)\\
&= \sum_{s=1}^{S} \log \int_{\boldsymbol{\theta}} \left(\frac{p(\boldsymbol{N}_s|\boldsymbol{\theta}, \boldsymbol{\Lambda}) p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{p(\boldsymbol{\theta}|\boldsymbol{\Gamma}_s)} \right) p(\boldsymbol{\theta}|\boldsymbol{\Gamma}_s) d\boldsymbol{\theta}, && (26)\\
&\geq \sum_{s=1}^{S} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} \left[\log \left(\frac{p(\boldsymbol{N}_s|\boldsymbol{\theta}, \boldsymbol{\Lambda}) p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{p(\boldsymbol{\theta}|\boldsymbol{\Gamma}_s)} \right) \right], && (27)\\
&= \sum_{s=1}^{S} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} [\log p(\boldsymbol{N}_s|\boldsymbol{\theta}, \boldsymbol{\Lambda})] + \sum_{s=1}^{S} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} [\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] - \sum_{s=1}^{S} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} [\log p(\boldsymbol{\theta}|\boldsymbol{\Gamma}_s)]. && (28)
\end{aligned}$$

We introduce the first set of variational parameters $\boldsymbol{\Gamma}$ to relate the sample and function levels. All the variational approximations are based on the fundamental concept of *Jensen's Inequality* which states that for a concave function $f(x)$, $f(\mathbb{E}_z[z]) \geq \mathbb{E}_z[f(z)]$, where $\mathbb{E}_z[z] = \int zp(z)dx$ or $\sum zp(z)$, for continuous and discrete z, respectively. We know that log is a concave function which enables (28), where the expectation is taken conditional on the variational parameter $\boldsymbol{\Gamma}$. We make another use of *Jensen's Inequality* in the first term of (28) and introduce variational parameters $\boldsymbol{\Phi}$, a $G \times S \times K$ array with the $(g, s, k)^{th}$ entry, ϕ_{gsk}, denoting $p(k|g, s)$.

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} [\log p(\boldsymbol{N}_s|\boldsymbol{\theta}, \boldsymbol{\Lambda})] &= \sum_{g=1}^{G} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} [\log \sum_{k=1}^{K} \theta_k \mathcal{P}(n_{gs}|k, \lambda_{gk})], && (29)\\
&= \sum_{g=1}^{G} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} \left[\log \sum_{k=1}^{K} \left(\frac{\theta_k \mathcal{P}(n_{gs}|k, \lambda_{gk})}{p(k|g, s)} \right) p(k|g, s) \right], && (30)\\
&= \sum_{g=1}^{G} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} \left[\log \mathbb{E}_{p(k|g,s)} \left[\frac{\theta_k \mathcal{P}(n_{gs}|k, \lambda_{gk})}{p(k|g, s)} \right] \right], && (31)\\
&= \sum_{g=1}^{G} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} \left[\log \mathbb{E}_{\phi_{gsk}} \left[\frac{\theta_k \mathcal{P}(n_{gs}|k, \lambda_{gk})}{\phi_{gsk}} \right] \right], && (32)\\
&\geq \sum_{g=1}^{G} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} \left[\mathbb{E}_{\phi_{gsk}} \left[\log \left(\frac{\theta_k \mathcal{P}(n_{gs}|k, \lambda_{gk})}{\phi_{gsk}} \right) \right] \right], && (33)\\
&= \sum_{g=1}^{G} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} \left[\sum_{k=1}^{K} \left[\log \left(\frac{\theta_k \mathcal{P}(n_{gs}|k, \lambda_{gk})}{\phi_{gsk}} \right) \right] \phi_{gsk} \right], && (34)\\
&= \sum_{g=1}^{G} \sum_{k=1}^{K} \left(\phi_{gsk} \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s} [\log \theta_k] + \phi_{gsk} \log \mathcal{P}(n_{gs}|k, \lambda_{gk}) - \phi_{gsk} \log \phi_{gsk} \right). && (35)
\end{aligned}$$

Substituting (35) in (28) yields (36), which is the likelihood of the observed data given the variational and model parameters; this equation is used throughout the remainder of this section. The optimal parameter estimates are found by optimizing

$$\log p(\boldsymbol{N}|\boldsymbol{\alpha},\boldsymbol{\Lambda}) \geq \sum_{g=1}^{G}\sum_{s=1}^{S}\sum_{k=1}^{K}\phi_{gsk}\log\mathcal{P}(n_{gs}|k,\lambda_{gk}) + \sum_{g=1}^{G}\sum_{s=1}^{S}\sum_{k=1}^{K}\phi_{gsk}\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\log\theta_k]$$
$$- \sum_{g=1}^{G}\sum_{s=1}^{S}\sum_{k=1}^{K}\phi_{gsk}\log\phi_{gsk} + \sum_{s=1}^{S}\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] - \sum_{s=1}^{S}\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\log p(\boldsymbol{\theta}|\boldsymbol{\Gamma}_s)]. \qquad (36)$$

A.2 Variational Parameters

A.2.1 Estimation of $\boldsymbol{\Phi}$

The first three terms of (36) contain $\boldsymbol{\Phi}$, therefore the optimal value of ϕ_{gsk} makes the derivative of (36) zero, under the constraint that $\sum_k \phi_{gsk} = 0$. Lagrange multipliers, (μ), are used to find the estimate $\hat{\phi}_{gsk}$

$$\phi_{gsk}\log\mathcal{P}(n_{gs}|k,\lambda_{gk}) + \phi_{gsk}\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\log\theta_k] - \phi_{gsk}\log\phi_{gsk} - \mu\left(\sum_{k=1}^{K}\phi_{gsk} - 1\right). \qquad (37)$$

Taking the partial derivatives with respect to ϕ_{gsk} and μ, and after setting them to zero and rearranging gives

$$\log\mathcal{P}(n_{gs}|k,\lambda_{gk}) - (\log\phi_{gsk} + 1) + \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\log\theta_k] - \mu = 0, \qquad (38)$$

implying

$$\hat{\phi}_{gsk} = \frac{\mathcal{P}(n_{gs}|k,\lambda_{gk})\exp[\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\boldsymbol{\Psi}(\gamma_{sk})]]}{\sum_{k'=1}^{K}\mathcal{P}(n_{gs}|k',\lambda_{gk'})\exp[\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\boldsymbol{\Psi}(\gamma_{sk'})]]}, \qquad (39)$$

where, using the results of Appendix A.1 of Blei et al. (2003)

$$\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\log p(\boldsymbol{\theta}|\boldsymbol{\Gamma}_s)] = \boldsymbol{\Psi}(\gamma_{sk}) - \boldsymbol{\Psi}\left(\sum_{k=1}^{K}\gamma_{sk}\right). \qquad (40)$$

Therefore, the final formula for estimating ϕ_{gsk} is

$$\hat{\phi}_{gsk} = \frac{\mathcal{P}(n_{gs}|k,\lambda_{gk})\exp[\boldsymbol{\Psi}(\gamma_{sk})]}{\sum\limits_{k'=1}^{K}\mathcal{P}(n_{gs}|k^{'},\lambda_{gk'})\exp[\boldsymbol{\Psi}(\gamma_{sk'})]}. \qquad (41)$$

A.3 Model Parameters

A.3.1 Estimation of $\boldsymbol{\Lambda}$

The first term of (36) contains $\boldsymbol{\Lambda}$, therefore differentiating this term with respect to λ_{gk}

$$\sum_{s=1}^{S} \phi_{gsk} \frac{\partial}{\partial \lambda_{gk}} \log \mathcal{P}(n_{gs}|k, \lambda_{gk}) = 0, \tag{42}$$

$$\sum_{s=1}^{S} \phi_{gsk} \frac{\partial}{\partial \lambda_{gk}} [-\lambda_{gk} + n_{gs} \log \lambda_{gk} - \log n_{ga}!] = 0, \tag{43}$$

$$\sum_{s=1}^{S} \phi_{gsk} \left[-1 + \frac{n_{gs}}{\hat{\lambda}_{gk}} \right] = 0, \tag{44}$$

and rearranging gives

$$\hat{\lambda}_{gk} = \frac{\sum_{s=1}^{S} \phi_{gsk} n_{gs}}{\sum_{s=1}^{S} \phi_{gsk}}. \tag{45}$$

A.3.2 Shrinkage of $\boldsymbol{\Lambda}$

We impose the following prior on all the elements of $\boldsymbol{\Lambda}$

$$\lambda_{gk} \sim \text{ Gamma}(\eta, \beta) \qquad g = 1, \ldots, G \text{ and } k = 1, \ldots, K. \tag{46}$$

Using (3), in addition to differentiating the first term of (36) (Section A.3.1), we also differentiate the gamma prior to obtain the optimal estimates

$$\sum_{s=1}^{S} \left[\phi_{gsk} \left[-1 + \frac{n_{gs}}{\hat{\lambda}_{gk}} \right] \right] + \frac{\eta - 1}{\lambda_{gk}} - \beta = 0, \tag{47}$$

$$\hat{\lambda}_{gk} = \frac{\sum_{s=1}^{S} \phi_{gsk} n_{gs} + \eta - 1}{\sum_{s=1}^{S} \phi_{gsk} + \beta}. \tag{48}$$

A.3.3 Estimation of $\boldsymbol{\alpha}$

The updates for $\boldsymbol{\alpha}$ are the same as in the of Appendix, Section A.2 of Blei et al. (2003) and Appendix, Section A.1 of Rogers et al. (2005).

A.3.4 Estimation of $\boldsymbol{\Gamma}$

There are three terms in (36) associated with $\boldsymbol{\Gamma}$, one of them is

$$\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\log p(\boldsymbol{\theta}|\boldsymbol{\Gamma}_s)] = \int_{\boldsymbol{\theta}} \log\left[\frac{\Gamma(\sum_k \gamma_{sk})}{\prod_k \Gamma(\gamma_{sk})} + \sum_{k=1}^{K}(\gamma_{sk}-1)\log\theta_k\right] p(\boldsymbol{\theta}|\boldsymbol{\Gamma}_s)d\boldsymbol{\theta}, \tag{49}$$

$$= \log\left[\frac{\Gamma(\sum_k \gamma_{sk})}{\prod_k \Gamma(\gamma_{sk})}\right] + \sum_k(\gamma_{ak}-1)\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{\Gamma}_s}[\log\theta_k], \tag{50}$$

$$= \log\left[\frac{\Gamma(\sum_k \gamma_{sk})}{\prod_k \Gamma(\gamma_{sk})}\right] + \sum_k(\gamma_{ak}-1)[\boldsymbol{\Psi}(\gamma_{sk}) - \boldsymbol{\Psi}(\sum_k \gamma_{sk})]. \tag{51}$$

Taking partial derivatives with respect γ_{ak} and setting those to zero leaves

$$\left(\alpha_k - \gamma_{sk} + \sum_g \phi_{gsk}\right)\left[\boldsymbol{\Psi}^{'}(\gamma_{sk}) - \boldsymbol{\Psi}^{'}(\sum_k \gamma_{sk})\right] - \log\left[\frac{\Gamma(\sum_k \gamma_{sk})}{\prod_k \Gamma(\gamma_{sk})}\right] = 0, \tag{52}$$

$$\left(\alpha_k - \gamma_{sk} + \sum_g \phi_{gsk}\right)\left[\boldsymbol{\Psi}^{'}(\gamma_{sk}) - \boldsymbol{\Psi}^{'}(\sum_k \gamma_{sk})\right] = 0. \tag{53}$$

gives the following update for γ_{ak}

$$\gamma_{sk} = \alpha_k + \sum_{g=1}^{G}\phi_{gsk}. \tag{54}$$

A.4 Convergence Issues

See Appendix A.3, Rogers et al. (2005).

A.5 Validation of K

Before describing this procedure, we describe a method for estimating the log-likelihood in the first stage of LPD given the model parameters; this will be used in the validation of K. Assuming that the model parameters are known, the log-likelihood for next-gen data is calculated using (5). However, the integral over $\boldsymbol{\theta}$ does not have a standard form. Therefore, it is approximated by its Monte Carlo integral

$$\sum_{s=1}^{S}\log p(\boldsymbol{N}_s|\boldsymbol{\alpha},\boldsymbol{\Lambda}) \approx \sum_{s=1}^{S}\log\sum_{b=1}^{B}\left\{\prod_{g=1}^{G}\sum_{k=1}^{K}\theta_{kb}\mathcal{P}(n_{gs}|k,\lambda_{gk})\right\} - S\log B. \tag{55}$$

This (55) that averages across B samples of functional probabilities, $\{\boldsymbol{\theta}_b\}_{b=1}^{B}$, drawn from the estimated distribution of function probabilities, Dirichlet($\hat{\boldsymbol{\alpha}}$). The value of K is validated by randomly partitioning the next-gen data into N subsets of samples, removing one of the partitions, and estimating the model and variational parameters, $\boldsymbol{\alpha}$, $\boldsymbol{\Lambda}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\Phi}$ using the first stage of LPD on the remaining $N - 1$ subsets. The log-likelihood for the excluded part is calculated using (55). This is repeated for each of the N partitions, and the value of K that maximizes the average of the log-likelihood across N partitions is chosen. This process is repeated many times and the value of K chosen maximum number of times is the validated value.

References

Abramowitz, M. and I. Stegun (Eds.) (1970). *Handbook of Mathematical Functions*. Dover Publications.

Agresti, A. (2002). *Categorical data analysis*, Volume 359. John Wiley and Sons.

Aldous, D. (1985). Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII—1983*, 1–198.

Anders, S. and W. Huber (2010). Differential expression analysis for sequence count data. *Genome Biology 11*(10), R106+.

Auer, P. L. and R. W. Doerge (2011). A Two-Stage Poisson Model for Testing RNA-Seq Data. *Statistical Applications in Genetics and Molecular Biology 10*(1), 26.

Baldi, P. and A. D. Long (2001). A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics 17*(6), 509–519.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological) 57*(1), 289–300.

Benjamini, Y. and D. Yekutieli (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics 29*, 1165–1188.

Bishop, C. M. (2006). *Pattern recognition and machine learning*, Volume 4. Springer New York.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research 3*, 993–1022.

Broet, P., S. Richardson, and F. Radvanyi (2002). Bayesian hierarchical model for identifying changes in gene expression from microarray experiments. *Journal of Computational Biology 9*(4), 671–683.

Casella, G. and R. L. Berger (2001). *Statistical inference*. Duxbury Press.

Choi, S. C. and R. Wette (1969). Maximum likelihood estimation of the parameters of the gamma distribution and their bias. *Technometrics 11*(4), pp. 683–690.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39*(1), 1–38.

Dudoit, S., J. Fridlyand, and T. P. Speed (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association 97*(457), 77–87.

Efron, B. (2004). Large-scale simultaneous hypothesis testing. *Journal of the American Statistical Association 99*(465), 96–104.

Efron, B. (2007). Size, power and false discovery rates. *The Annals of Statistics 35*(4), 1351–1377.

Efron, B. (2008). Microarrays, empirical Bayes and the two-groups model. *Statistical Science 23*(1), 1–22.

Efron, B. (2010). *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Cambridge Univ Pr.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of statistics 32*(2), 407–499.

Efron, B. and R. Tibshirani (2007). On testing the significance of sets of genes. *The Annals of Applied Statistics 1*(1), 107–129.

Efron, B., R. Tibshirani, J. D. Storey, and V. Tusher (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association 96*(456), 1151–1160.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2003). *Bayesian Data Analysis* (2 ed.). Boca Raton, Florida: Chapman & Hall/CRC.

Genovese, C. and L. Wasserman (2002). Operating characteristics and extensions of the false discovery rate procedure. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64*(3), 499–517.

Gentleman, R., V. Carey, D. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Yang, and J. Zhang (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology 5*(10), R80.

Ghosh, J. K. and R. V. Ramamoorthi (2003). *Bayesian nonparametrics*. Springer Verlag.

Guo, Y., T. Hastie, and R. Tibshirani (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics 8*(1), 86.

Hardcastle, T. J. and K. A. Kelly (2010). baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC bioinformatics 11*(1), 422.

Hastie, T., R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown (2000). Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biol 1*(2), 1–21.

Hayden, E. C. (2009). Genome sequencing: the third generation. *Nature 457*(7231), 768–9.

Hewitt, E. and L. J. Savage (1955). Symmetric measures on Cartesian products. *Trans. Amer. Math. Soc 80*(1), 470–501.

Hjort, N., C. Holmes, P. Müller, and S. Walker (Eds.) (2010). *Bayesian Nonparametrics*. Cambridge University Press.

Ibrahim, J. G., M. H. Chen, and R. J. Gray (2002). Bayesian models for gene expression with DNA microarray data. *Journal of the American Statistical Association 97*(457), 88–99.

Ishwaran, H. and J. S. Rao (2003). Detecting differentially expressed genes in microarrays using Bayesian model selection. *Journal of the American Statistical Association 98*(462), 438–455.

Ma, S. and J. Huang (2007). Clustering threshold gradient descent regularization: with applications to microarray studies. *Bioinformatics 23*(4), 466.

Ma, S., X. Song, and J. Huang (2007). Supervised group Lasso with applications to microarray data analysis. *BMC bioinformatics 8*(1), 60.

Marioni, J. C., C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad (2008). RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research 18*(9), 1509.

Medvedovic, M., K. Y. Yeung, and R. E. Bumgarner (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics 20*(8), 1222.

Metzker, M. L. (2009). Sequencing technologies—the next generation. *Nature Reviews Genetics 11*(1), 31–46.

Morris, C. N. (1983). Parametric empirical bayes inference: Theory and applications. *Journal of the American Statistical Association 78*(381), pp. 47–55.

Nagalakshmi, U., Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder (2008). The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science 320*(5881), 1344.

Newton, M. A., A. Noueiry, D. Sarkar, and P. Ahlquist (2004). Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics 5*(2), 155.

Ng, S. B., K. J. Buckingham, C. Lee, A. W. Bigham, H. K. Tabor, K. M. Dent, C. D. Huff, P. T. Shannon, E. W. Jabs, D. A. Nickerson, and Others (2009). Exome sequencing identifies the cause of a mendelian disorder. *Nature genetics 42*(1), 30–35.

Oshlack, A., M. D. Robinson, and M. D. Young (2010). From RNA-seq reads to differential expression results. *Genome Biology 11*(12), 220.

Pachter, L. (2011, May). Models for transcript quantification from RNA-Seq. *arXiv:1104.3889v2*.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Roach, J. C., G. Glusman, A. F. A. Smit, C. D. Huff, R. Hubley, P. T. Shannon, L. Rowen, K. P. Pant, N. Goodman, M. Bamshad, and Others (2010). Analysis of genetic inheritance in a family quartet by whole-genome sequencing. *Science 328*(5978), 636.

Robert, C. P. and G. Casella (2004). *Monte Carlo statistical methods*. Springer Verlag.

Robinson, M. D., D. J. McCarthy, and G. K. Smyth (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics 26*(1), 139.

Robinson, M. D. and G. K. Smyth (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics 23*(21), 2881.

Robinson, M. D. and G. K. Smyth (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics 9*(2), 321.

Rogers, S., M. Girolami, C. Campbell, and R. Breitling (2005). The latent process decomposition of cDNA microarray data sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 143–156.

Severin, A. J., J. L. Woody, Y. T. Bolon, B. Joseph, B. W. Diers, A. D. Farmer, G. J. Muehlbauer, R. T. Nelson, D. Grant, J. E. Specht, and Others (2010). RNA-Seq Atlas of Glycine max: a guide to the soybean transcriptome. *BMC plant biology 10*(1), 160.

Smola, A. J. and S. Narayanamurthy (2010). An Architecture for Parallel Topic Models. In *Very Large Databases (VLDB)*.

Stein, L. D. (2010). The case for cloud computing in genome informatics. *Genome Biology 11*, 207.

Storey, J. D. (2003). The positive false discovery rate: A Bayesian interpretation and the q-value. *The Annals of Statistics 31*(6), 2013–2035.

Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and Others (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America 102*(43), 15545.

Sun, W. and T. T. Cai (2007). Oracle and adaptive compound decision rules for false discovery rate control. *Journal of the American Statistical Association 102*(479), 901–912.

Thygesen, H. H. and A. H. Zwinderman (2006). Modeling Sage data with a truncated gamma-Poisson model. *BMC bioinformatics 7*(1), 157.

Tibshirani, R., T. Hastie, B. Narasimhan, and G. Chu (2003). Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Statistical Science 18*(1), 104–117.

Tibshirani, R., M. Saunders, S. Rosset, J. Zhu, and K. Knight (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(1), 91–108.

Tierney, L., R. E. Kass, and J. B. Kadane (1989). Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association 84*(407), 710–716.

Vêncio, R. Z. N., H. Brentani, D. F. C. Patrão, and C. A. B. Pereira (2004). Bayesian model accounting for within-class biological variability in Serial Analysis of Gene Expression(SAGE). *BMC bioinformatics 5*(1), 119.

Wang, Y., H. Bai, M. Stanton, W. Y. Chen, and E. Chang (2009). Plda: Parallel latent dirichlet allocation for large-scale applications. *Algorithmic Aspects in Information and Management*, 301–314.

Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. Springer New York.

Witten, D. M., R. Tibshirani, and T. Hastie (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics 10*(3), 515.

Yeung, K. Y., R. E. Bumgarner, and A. E. Raftery (2005). Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics 21*(10), 2394.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301–320.