Submitted to the Annals of Applied Statistics

MODEL-BASED FEATURE SELECTION AND CLUSTERING OF RNA-SEQ DATA FOR UNSUPERVISED SUBTYPE DISCOVERY

BY DAVID K. LIM, NAIM U. RASHID, AND JOSEPH G. IBRAHIM

University of North Carolina at Chapel Hill, NC, USA

Clustering is a form of unsupervised learning that aims to uncover latent groups within data based on similarity across a set of features. A common application of this in biomedical research is in delineating novel cancer subtypes from patient gene expression data, given a set of informative genes. However, it is typically unknown *a priori* what genes may be informative in discriminating between clusters, and what the optimal number of clusters are. Few methods exist for performing unsupervised clustering of RNA-seq samples, and none currently adjust for between-sample global normalization factors, select cluster-discriminatory genes, or account for potential confounding variables during clustering. To address these issues, we propose the Feature Selection and Clustering of RNA-seq (FSCseq): a model-based clustering algorithm that utilizes a finite mixture of regression (FMR) model and utilized the quadratic penalty method with a SCAD penalty. The maximization is done by a penalized Classification EM algorithm, allowing us to include normalization factors and confounders in our modeling framework. Given the fitted model, our framework allows for subtype prediction in new patients via posterior probabilities of cluster membership. Based on simulations and real data analysis, we show the advantages of our method relative to competing approaches.

1. Introduction. Clustering is a form of unsupervised learning that aims to uncover latent groups across subjects based on their similarity with respect to a common set of features. A common application of clustering is in the discovery of novel molecular subtypes in cancer based upon the patients' subject-specific RNA sequencing (RNA-seq) gene expression profiles. Many studies have demonstrated the clinical utility of such molecular subtypes, where significant differences in patient prognosis and treatment response between subtypes have been observed (Perou et al., 2000; Chia et al., 2012; Mao et al., 2017).

However, few methods are able to cluster samples whose genes (features) are measured via RNA-seq, a common platform for measuring gene expression based upon high-throughput sequencing (Mo et al., 2013; Li et al., 2018).

Keywords and phrases: clustering, RNA-seq, confounders, batch

Gene expression in RNA-seq studies is typically quantified in terms of gene-level read counts, defined by the number of sequencing reads mapping to protein coding regions in a particular gene following sequence alignment. Gene expression quantitation software such as **RSEM** and **Salmon** can further account for multimapped reads, or sequences that map to multiple regions of the genome (Li and Dewey, 2011; Patro et al., 2017), in addition to other technical biases in quantifying gene-level read counts. Such counts are often overdispersed, and are therefore commonly modeled by the negative binomial distribution (Robinson, McCarthy and Smyth, 2009; Love, Huber and Anders, 2014). This is in contrast to Gaussian distributional assumptions made by earlier methods designed for clustering subjects with microarray-based gene expression measurements, such as **mclust** (Scrucca et al., 2016).

This shift in distributional assumption has practical implications. For example, between-sample variation in technical factors such as sequencing depth can cause global shifts in sample-specific read count distributions, necessitating the use of between-sample correction methods (Li et al., 2015). However, due to the count nature of RNA-seq data, these correction factors are often directly incorporated into statistical modeling procedures, for example, as offset terms in negative binomial regression-based differential expression methods (Robinson, McCarthy and Smyth, 2009; Love, Huber and Anders, 2014). Correction for other confounders, such as batch effects (Leek et al., 2010) or specific clinical variables, is done by including the relevant factors as regression covariates. However, to the best of our knowledge, no analogous approach exists to correct for such factors in RNA-seq read count-based clustering analyses. Therefore, an approach that is able to correct for such factors may have significant utility in improving clustering performance.

Furthermore, a large number of genes is often available for clustering subjects (typically 20,000 or more), and it is often unknown *a priori* which subset of genes may best discriminate between latent clusters. Utilizing all available genes for clustering may negatively impact the accuracy and usefulness of derived clusters (Tritchler, Parkhomenko and Beyene, 2009), cause methods to underestimate the number of latent groups (Pan and Shen, 2007), and add significantly to computation time. As a result, it is common practice to filter a large proportion of genes prior to clustering, for example, by removing low variance genes with respect to their median absolute deviation (MAD) across samples (Chung et al., 2008; Wu et al., 2013). However, it is often unclear how many genes should be kept to ensure optimal clustering performance, as a poor threshold can decrease power of detecting differen-

tially expressed genes (Bourgon, Gentleman and Huber, 2010). Lastly, the identification of such cluster-discriminatory genes may improve biological understanding of identified clusters via gene set enrichment analyses (Subramanian et al., 2005), and may aid in the development of future subtype classification methods (Golub et al., 1999).

Current methods that seek to cluster RNA-seq data have several limitations with respect to the issues mentioned above. We divide these methods into three general classes: transformation-based, count-based, and non-parametric clustering methods. Transformation-based methods first apply a variance-stabilizing transformation to RNA-seq gene read counts, allowing for the application of clustering methods developed for microarray data (Zwiener, Frisch and Binder, 2014) by assuming the transformed data are distributed approximately Gaussian. However, prior work has shown that the choice of transformation can greatly influence clustering performance, as the quality of this Gaussian approximation varies by condition, and the optimal choice may not be readily apparent (Noel-MacDonnell et al., 2018). Examples of such transformations include the log, variance stabilizing (Anders and Huber, 2010), and regularized log (rlog) (Love, Huber and Anders, 2014) transforms. The second class of methods directly models gene-level read counts assuming some type of count distribution. One such method is *iCluster+* (Mo et al., 2013), which assumes a Poisson distribution on the gene read counts. *iCluster+* also performs feature selection by inducing sparsity via the lasso penalty, but ignores potential overdispersion, or extra-Poisson variation, in counts. NBMB (Li et al., 2018) accounts for overdispersion via the negative-binomial distribution, but cannot identify cluster-discriminatory genes. In addition, neither method is able to adjust for confounding factors, such as batch effects or differences in sequencing depth. The third class of methods includes non-parametric clustering approaches such as the average-linkage hierarchical clustering (HC) and K-medoids (KM) (Jaskowiak, Costa and Campello, 2018). However, these algorithms cannot perform feature selection, nor can they adjust for batch effects or differences in sequencing depth. Lastly, none of these methods can be directly utilized to predict discovered subtypes in new subjects, limiting their direct translational utility.

To address these issues, we propose FSCseq (Feature Selection and Clustering of RNA-seq): a penalized finite mixture regression model that assumes negative-binomially distributed mixture components. In addition to directly clustering samples with respect to RNA-seq gene read counts, FSCseq also performs simultaneous feature selection to select cluster-discriminatory genes during the clustering process. We achieve this by imposing a fusion penalty

on the differences in estimated means within each gene, simultaneously shrinking the estimates closer together while inducing sparsity on the differences across clusters. By modeling cluster means with weighted negative binomial regression models, FSCseq can directly adjust for differences in sequencing depth through the inclusion of size factors in a manner similar to common differential gene expression methods. Our method can also correct for potential confounders, such as batch effects, by including them as regression covariates. We demonstrate the utility of these features in improving clustering performance via simulation studies and a real data application to a TCGA breast cancer RNA-seq dataset. Lastly, we illustrate how FSCseq performs subtype prediction in new samples, given a new set of gene expression profiles and a prior fitted FSCseq object. In all, we address critically lacking features in clustering samples whose gene expression profiles are measured via RNA-seq, and demonstrate improved performance with respect to existing methods over a range of conditions.

2. Methods. The main utility of FSCseq is two-fold, where we perform 1) unsupervised clustering of subjects based on subject-specific RNA-seq gene expression profiles, and 2) simultaneous selection of cluster-discriminatory genes. Our modeling approach can correct for sources of technical variation like sequencing depth, adjust for effects of potential confounders during clustering such as batch effects, and directly predict discovered subtypes in new patients based upon a previously fitted FSCseq model. We describe our model implementation below, and a visualization of the FSCseq workflow can be found in Section A of the Supplementary Material [Lim, 2020].

2.1. Model likelihood. Gene-level RNA-seq read counts are utilized as the basis of our model, which can be obtained from RNA-seq gene expression quantification software such as Salmon (Patro et al., 2017). Such approaches account for read multi-mapping and other biases and, as a result, often provide non-integer “expected read counts” adjusting for these factors. We round these expected count values to integers, similar to previous approaches (Love, Huber and Anders, 2014). Let $\mathbf{y}$ denote an $n \times G$ matrix of n subjects (or “samples”) and G genes (or “features”), with each element y_{ij} denoting the read count of the j^{th} gene for the i^{th} subject, where $i = 1, \dots, n$ and $j = 1, \dots, G$.

We assume that the RNA-seq gene expression profile of subject i , $\mathbf{y}_i = (y_{i1}, \dots, y_{iG})$, can be modeled by a mixture of K G -dimensional multivariate negative binomial regression models. The likelihood of this mixture model

may be written as

$$(1) \quad L(\boldsymbol{\mu}, \boldsymbol{\phi} | \mathbf{y}) = f(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\phi}) = \prod_{i=1}^n \sum_{k=1}^K \pi_k f_k(\mathbf{y}_i; \boldsymbol{\mu}_{ik}, \boldsymbol{\phi}),$$

where K is the “order” of the mixture model (i.e. the number of assumed clusters in the data), π_k is the mixture proportion pertaining to cluster k , and $\sum_{k=1}^K \pi_k = 1$. We denote μ_{ijk} as the mean of the j^{th} gene for the i^{th} subject in the k^{th} cluster, after adjusting for subject-specific sequencing depth, and we denote ϕ_j as the gene-level dispersion parameter pertaining to gene j . Then, $\boldsymbol{\mu}_{ik} = (\mu_{i1k}, \dots, \mu_{iGk})$ is the mean expression profile pertaining to sample i and cluster k , and $\boldsymbol{\phi} = (\phi_1, \dots, \phi_G)$ is a vector of the G gene-level dispersion parameters.

We utilize a gene-gene independence assumption, which has been successfully implemented in prior RNA-seq read count-based clustering methods (Si et al., 2013; Mo et al., 2013; Li et al., 2018), as a reasonable approximation to simplify (1). Under this assumption, $f_k(\mathbf{y}_i; \boldsymbol{\mu}_{ik}, \boldsymbol{\phi}) = \prod_{j=1}^G f_{jk}(y_{ij}; \mu_{ijk}, \phi_j)$, where $f_{jk}(\cdot)$ is the negative binomial probability mass function, given by $f_{jk}(y_{ij}; \mu_{ijk}, \phi_j) = \binom{y_{ij} + \phi_j^{-1} - 1}{\phi_j^{-1} - 1} \left[\frac{1}{1 + \phi_j \mu_{ijk}} \right]^{\phi_j^{-1}} \left[\frac{\phi_j \mu_{ijk}}{1 + \phi_j \mu_{ijk}} \right]^{y_{ij}}$, equivalent to the “NB2” parameterization by Hilbe (2009). Although the multivariate log-normal Poisson or multivariate negative binomial model may alternatively account for potential correlation between genes, the specification and estimation of such a covariance structure across genes is intractable in very high dimensions (Inouye et al., 2017).

For the j^{th} gene, we may specify the link and variance functions with respect to μ_{ijk} as

$$(2) \quad \log_2(\mu_{ijk}) = \beta_{jk} + \sum_{p=1}^P x_{ip} \gamma_{jp} + s_i \quad V(\mu_{ijk}) = \mu_{ijk} + \phi_j \mu_{ijk}^2,$$

where β_{jk} is the log base 2 ($\log_2$) mean read count for gene j and cluster k , γ_{jp} is the effect of covariate p on the expression of gene j , and s_i is the $\log_2$ scaled size factor of subject i calculated by DESeq2 to correct for subject-specific differences in sequencing depth. In other words, we model the $\log_2$ mean of the j^{th} gene for subjects in the k^{th} cluster via a cluster-specific mean parameter (on the $\log_2$ scale) and a set of P subject-level covariates ($x_{i1}, \dots, x_{iP}$), while also accounting for sequencing depth. We allow covariate effects to be distinct across genes, similar to prior batch correction methods (Johnson, Li and Rabinovic, 2006; Leek, 2014), as, biologically, certain sets of genes may be affected differently by a particular covariate. Examples of

such covariates may include technical factors such as batch, or potential demographic variables such as age or tumor grade. Through the inclusion of covariates in the model, we will later demonstrate FSCseq’s ability to adjust cluster mean estimates, thus improving clustering performance in the presence of such factors.

We note that in the given parametrization, we specify gene-level dispersion parameters ϕ_j that are shared across the K clusters within gene j (Robinson, McCarthy and Smyth, 2009; Love, Huber and Anders, 2014). In contrast, NBMB (Li et al., 2018) specifies cluster-specific dispersion parameters $\phi_j = (\phi_{j1}, \dots, \phi_{jK})$ within each gene j . Conceptually, when n is small, the estimate of each ϕ_{jk} may not be accurate due to the smaller average per-cluster sample size (Piegorisch, 1990; Al-Khasawneh, 2010). In FSCseq, we utilize gene-level dispersions as the default in order to avoid this issue, and later show that the cluster-specific dispersion scheme by NBMB performs poorly in clustering analyses.

2.1.1. Selection of cluster-discriminatory genes via fusion penalty. FSCseq identifies cluster-discriminatory genes simultaneously during clustering. To do this, within a given gene j , we impose the SCAD penalty (Fan and Li, 2001) on the pairwise differences between $\log_2$ cluster means, in order to induce shrinkage on these differences and enforce equality of similar group means within gene j . That is, we introduce the penalty term

$$(3) \quad p_\lambda(\beta_j) = \sum_{k < l}^K \text{SCAD}_\lambda(|\beta_{jk} - \beta_{jl}|),$$

where $\beta_j = (\beta_{j1}, \dots, \beta_{jK})$, λ is a tuning parameter to control the amount of penalty that is introduced, and k and l are cluster indices such that $1 \leq k < l \leq K$. The explicit form of the original SCAD penalty can be found in Section A1 of the Supplementary Material [Lim, 2020].

Then, we aim to maximize the penalized log-likelihood, which may be written as

$$(4) \quad \begin{aligned} \log[L_p(\Psi)] &= \log[L(\mu, \phi | \mathbf{y})] + \text{Penalty} \\ &= \sum_{i=1}^n \log \left\{ \sum_{k=1}^K \pi_k \prod_{j=1}^G f_{jk}(y_{ij}; s_i, \beta_{jk}, \gamma_j, \phi_j) \right\} \\ &\quad + \sum_{j=1}^G p_\lambda(\beta_j), \end{aligned}$$

after incorporating the link function in (2) and the SCAD penalty. Here, $\Psi = (\pi, \beta, \gamma, \phi)$ pertains to the vector of model parameters, with $\pi =$

$(\pi_1, \dots, \pi_K)$, $\beta = (\beta_1, \dots, \beta_G)$, and $\gamma = (\gamma_1, \dots, \gamma_G)$. Here, $\gamma_j = (\gamma_{j1}, \dots, \gamma_{jP})$ denotes the regression coefficients pertaining to the P covariates in gene j , $j = 1, \dots, G$. The first term of (4) is the log of the model likelihood in (1), with μ_{ijk} replaced by $(s_i, \beta_{jk}, \gamma_j)$, since μ_{ijk} is a function of $(s_i, \beta_{jk}, \gamma_j)$ by (2).

However, because the penalty is imposed on the differences in cluster $\log_2$ mean parameters in gene j , it is not separable in β_j . As a result, the penalized log-likelihood may not converge to a stationary point during maximization (Friedman et al., 2007; Wu and Lange, 2008). Similar to Pan, Shen and Liu (2013), we address this by introducing a new variable $\theta_{j,kl} = \beta_{jk} - \beta_{jl}$ for each gene $j = 1, \dots, G$ and each pair of clusters k and l such that $1 \leq k < l \leq K$. We then apply the quadratic penalty method (Nocedal and Wright, 2000) leveraging this new constraint. In particular, for a given gene j , let θ_j be a $K \times K$ upper triangular matrix denoting the pairwise differences in β_j , with entries $(\theta_j)_{kl} = \theta_{j,kl}$. Then, under the quadratic penalty method, (3) can be rewritten as

$$(5) \quad p_{\lambda,\alpha}(\beta_j) = \frac{\lambda(1-\alpha)}{2} \sum_{k<l} (\beta_{jk} - \beta_{jl} - \theta_{j,kl})^2 + \sum_{k<l} \text{SCAD}_{\lambda\alpha}(|\theta_{j,kl}|),$$

where the first term is the added quadratic penalty term as described by Nocedal and Wright (2000), pertaining to the constraint $\theta_{j,kl} = \beta_{jk} - \beta_{jl}$. This first term in (5) is convex and differentiable, while the second term is non-smooth, but separable and convex, and so a coordinate-wise descent algorithm is guaranteed to converge to the global optimum (Tseng, 2001). In FSCseq, we utilize this property to implement a coordinate-wise descent algorithm (CDA) to estimate cluster $\log_2$ means and covariate effects within each given gene $j = 1, \dots, G$ (See Section 2.2.2 for more details). We also introduce the hyperparameter α to control the balance between these two terms. Although similar in form, we note that this is not the traditional elastic net penalty (Zou and Hastie, 2005). Rather, this penalty is similar to the L1 penalty with the quadratic penalty method proposed by (Pan, Shen and Liu, 2013), with the L1 replaced by the SCAD penalty.

We also note that $\beta_{jk} - \beta_{jl} \rightarrow 0$ as $\theta_{j,kl} \rightarrow 0$. However, similar to Pan, Shen and Liu (2013), we observe that $\beta_{jk} - \beta_{jl} - \theta_{j,kl} \rightarrow 0$ as $\lambda(1-\alpha) \rightarrow \infty$, but $\beta_{jk} - \beta_{jl} = 0$ cannot be guaranteed. In particular, let $\hat{\theta}_{j,kl}$ and $\hat{\beta}_{jk}$ denote the estimates of $\theta_{j,kl}$ and β_{jk} , respectively. Then, although $\hat{\theta}_{j,kl} = 0$ can be attained by the SCAD penalty, $\hat{\beta}_{jk} = \hat{\beta}_{jl}$ cannot be guaranteed for two distinct clusters k and l . This is because the term introduced by the quadratic penalty method (Nocedal and Wright, 2000) to enforce the constraint $\theta_{j,kl} = \beta_{jk} - \beta_{jl}$ is a ridge-like term, and thus does not perform strict thresholding,

much like the classic L2 penalty (Hoerl and Kennard, 1970). Increasing the first term by increasing $\lambda(1 - \alpha)$ more closely approximates equality in the constraint, but this introduces bias in the estimation of β_j (Pan, Shen and Liu, 2013).

In FSCseq, we address this by fusing clusters k and l together into one joint cluster for gene j if $\hat{\theta}_{j,kl} = 0$, denoting the index of this new cluster as k_* . Specifically, we estimate $\hat{\beta}_{jk_*}$ using data from both clusters k and l in gene j , and we strictly enforce the quadratic penalty method constraint by setting $\hat{\beta}_{jk} = \hat{\beta}_{jl} = \hat{\beta}_{jk_*}$. If all clusters are fused together in this way for a particular gene j , then gene j is determined to be “nondiscriminatory” across clusters. Otherwise, gene j is considered “cluster-discriminatory.” We call this penalty the “SCAD fusion penalty”. See Section 2.2 and Section A1 in the Supplementary Material for more details on this procedure [Lim, 2020].

We select optimal values of the three tuning parameters (K , α , and λ) by searching over a grid of possible values for α and λ with K fixed, and then repeat this procedure for every candidate value of K . We utilize the Bayesian Information Criterion (BIC) for selection (Schwarz, 1978). Details of this tuning parameter selection procedure is given in Section 2.2.6.

2.2. Computation. We maximize the penalized log-likelihood from (4) via a Classification Expectation-Maximization (CEM) algorithm with simulated annealing (Celeux and Govaert, 1992) to reduce the likelihood of converging to a local maximum. To reduce computational overhead per iteration, we also apply “mini-batching” of genes. That is, we update the estimates of parameters pertaining to a random subset (or a ‘mini-batch’) of the genes in the M-step, and then perform the E-step after each ‘mini-batched’ M-step. This is in contrast to the standard EM algorithm, where we would only perform the E-step after estimating parameters for all genes in the M-step. Thus, mini-batching allows the E-step to be updated more quickly with M-step updates from a subset of the parameter space, similar to the multicycle ECM algorithm (Meng and Rubin, 1993) which has been shown to decrease overall computation time (Meng, 1994). We show later that this mini-batching approach works well in our simulation and real data analyses.

The complete data penalized log-likelihood, which corresponds to the pe-

nalized log-likelihood in (4), can be written as

$$(6) \quad \log[L_{cp}(\Psi)] = \sum_{i=1}^n \sum_{k=1}^K z_{ik} \left\{ \log(\pi_k) + \log \prod_{j=1}^G f_{jk}(y_{ij}; s_i, \beta_{jk}, \gamma_j, \phi_j) \right\} + \sum_{j=1}^G p_{\lambda, \alpha}(\beta_j),$$

where $z_{ik} = I(z_i = k)$ is the indicator of whether subject i belongs to cluster k . In this context, z_{ik} is a latent variable denoting cluster membership of sample i in cluster k . In the standard EM algorithm, the so-called “Q-function” is then calculated as the conditional expectation of $\log[L_{cp}(\Psi)]$, given the current parameter estimates and observed data (McLachlan and Krishnan, 2008; Garcia, Ibrahim and Zhu, 2009). This expectation is calculated in the E-step, and parameter estimates are updated in the M-step by maximizing the Q-function.

The rest of this section is organized as follows. To simplify the presentation of our method, we first describe the general E and M-steps for our algorithm without mini-batching, given some fixed values of K , α , and λ , in Sections 2.2.1 and 2.2.2. Then, we describe the modifications of these steps to support gene mini-batching in Section 2.2.3. We then show how the FSCseq algorithm is initialized in Section 2.2.4, outline convergence criteria in Section 2.2.5, and describe our procedure for selecting the optimal values of K , α and λ in Section 2.2.6. We conclude this section by describing a method for subtype prediction using the converged FSCseq model in Section 2.2.7.

2.2.1. E-Step. We first describe the E-step with no mini-batching of genes. Let m denote the current EM iteration, $\hat{\Psi}^{(m)} = (\hat{\pi}^{(m)}, \hat{\beta}^{(m)}, \hat{\gamma}^{(m)}, \hat{\phi}^{(m)})$ denote the estimates of $(\pi, \beta, \gamma, \phi)$ in the current iteration, and $\hat{z}_{ik}^{(m)} = E[z_{ik} \mid \mathbf{y}, \mathbf{s}, \hat{\Psi}^{(m)}]$ denote the conditional expectation of z_{ik} , given the observed data and the current parameter estimates, where $\mathbf{s} = (s_1, \dots, s_n)$ is the vector of fixed size factor offsets calculated *a priori* by DESeq2 (Love, Huber and Anders, 2014). Given the form of (6), evaluation of the Q-function requires only the calculation of $\hat{z}_{ik}^{(m)}$, which can be interpreted as the posterior probability of subject i belonging to cluster k given the parameter estimates at the m^{th} EM iteration. Thus, in the standard EM algorithm, the m^{th} E-step reduces to computing $\hat{z}_{ik}^{(m)}$ for $i = 1, \dots, n$ and $k = 1, \dots, K$ by the following expression:

$$(7) \quad \hat{z}_{ik}^{(m)} = \frac{\hat{\pi}_k^{(m)} \prod_{j=1}^G f_{jk}(y_{ij}; s_i, \hat{\beta}_{jk}^{(m)}, \hat{\gamma}_j^{(m)}, \hat{\phi}_j^{(m)})}{\sum_{k'=1}^K \hat{\pi}_{k'}^{(m)} \prod_{j=1}^G f_{jk'}(y_{ij}; s_i, \hat{\beta}_{jk'}^{(m)}, \hat{\gamma}_j^{(m)}, \hat{\phi}_j^{(m)})}.$$

These posterior probabilities may then be used as cluster-specific observation weights in the M-step. Following convergence, it is also common to assign cluster membership to a subject by determining the cluster with the maximum posterior probability in that subject. That is, we can assign subject i to the cluster corresponding to $\arg \max_k (\hat{z}_{ik})$ at convergence. We note from (7) that the contribution of cluster-nondiscriminatory genes cancel out in the posterior probability calculations.

However, the standard EM algorithm is known for its susceptibility to converge to a local optimum, rather than the global optimum (Dellaert, 2002). To reduce the likelihood of convergence to a local optimum, we modify the E-step in a manner similar to the Classification EM (CEM) algorithm with simulated annealing (Celeux and Govaert, 1990; Si et al., 2013). In the original CEM with simulated annealing algorithm (Celeux and Govaert, 1990), the E-step update in (7) is modified (denoted as the “AE-step”), and an additional sampling step (denoted as the “C-step”) is added after the AE-step.

In the AE-step, a “temperature” term is incorporated into the E-step update, and is gradually decreased (or “annealed”) such that the values of $\hat{z}_{ik}$ approach 0 or 1. The AE-step update corresponding to (7) is written as

$$(8) \quad \hat{z}_{ik}^{(m)} = \frac{[\hat{\pi}_k^{(m)} \prod_{j=1}^G f_{jk}(y_{ij}; s_i, \hat{\beta}_{jk}^{(m)}, \hat{\gamma}_j^{(m)}, \hat{\phi}_j^{(m)})]^{1/\tau^{(m)}}}{\sum_{k'=1}^K [\hat{\pi}_{k'}^{(m)} \prod_{j=1}^G f_{jk'}(y_{ij}; s_i, \hat{\beta}_{jk'}^{(m)}, \hat{\gamma}_j^{(m)}, \hat{\phi}_j^{(m)})]^{1/\tau^{(m)}}},$$

where $\tau^{(m)}$ is the temperature of annealing in the m^{th} AE-step iteration. As $\tau^{(m)} \rightarrow \infty$, $\hat{z}_{ik}^{(m)} \rightarrow 1/K$, maximizing the entropy of the partition. As $\tau^{(m)} \rightarrow 0$, we approach a hard partition for cluster membership (Rose, 1998) because the elements of the vector of posterior probabilities $\hat{\mathbf{z}}_i^{(m)} = (\hat{z}_{i1}^{(m)}, \dots, \hat{z}_{iK}^{(m)})$ tend to 0 or 1, and $\hat{z}_{ik} \in (0, 1)$ with $\sum_{k=1}^K \hat{z}_{ik}^{(m)} = 1$ for all subjects $i = 1, \dots, n$. Traditionally, one allows $\tau^{(m)} \rightarrow 0$ as $m \rightarrow \infty$ at a pre-specified annealing rate r such that $0 < r < 1$ and $\tau^{(m+1)} = r\tau^{(m)}$. In this way, the AE-step modification allows the CEM to avoid a local optimum by gradually reducing the stochasticity introduced by the sampling in the C-step.

In the C-step, a sample $\mathbf{c}_i = (c_{i1}, \dots, c_{iK})$ is drawn from the multinomial distribution with probabilities equal to $\hat{\mathbf{z}}_i^{(m)}$ from (8). Then, we set $\hat{\mathbf{z}}_i^{(m)} \leftarrow \mathbf{c}_i$, such that the $\hat{\mathbf{z}}_i^{(m)}$ now specify a hard classification of sample i , i.e. for a given sample i , $\hat{z}_{ik}^{(m)}$ is now exactly 1 for one value of k , and 0 for all other values of k . This is repeated for all samples $i = 1, \dots, n$, yielding a hard classification partition of samples.

The combination of sampling in the C-step and the inclusion of annealing in the AE-step together helps the algorithm escape potential local optima.

This is achieved by initializing $\tau^{(m)}$ to be large in the first iterations, such that the $\hat{z}_{ik}^{(m)}$ from the AE-step are close to $1/K$ for all i and k . Then, the probability of yielding a random partition of subjects in the C-step is high in the first iterations. By decreasing the value of $\tau^{(m)}$, the probability of random partitioning approaches 0. Thus, a transition corresponding to a decrease in the objective function can be accepted with non-zero probability, and this probability approaches 0 as the algorithm proceeds (Celeux and Govaert, 1992). This results in the useful property for the CEM of not terminating when reaching the first local optimum.

Following the guidelines by Celeux and Govaert (1992) and Rose (1998), we set the annealing rate $r = 0.9$, similar to other methods (Si et al., 2013; Li et al., 2018). It is suggested that the entropy should be sufficiently large in early iterations, eventually shrinking to 0 in later iterations (Klein and Dubes, 1989; van Laarhoven and Aarts, 1987; Rose, 1998). Additionally, if the initial temperature $\tau^{(1)}$ is too small, annealing may terminate at a suboptimal solution (Celeux and Govaert, 1992). We address this by setting the value of $\tau^{(1)} = G$. We find that this works well in our simulations and real data analyses.

2.2.2. M-step. In the M-step, we maximize the Q-function, given the current E-step values $\hat{\mathbf{z}}^{(m)} = (\hat{\mathbf{z}}_1^{(m)}, \dots, \hat{\mathbf{z}}_n^{(m)})$ and some specified values of α and λ , updating the current estimates of the model parameters. From (6), the estimation of π_k is separable from the remaining model parameters, and the corresponding updates for $\hat{\boldsymbol{\pi}} = (\hat{\pi}_1, \dots, \hat{\pi}_K)$ are given by $\hat{\pi}_k^{(m+1)} = \sum_{i=1}^n \hat{z}_{ik}^{(m)} / n$ for $k = 1, \dots, K$. We also observe that the updates of the remaining model parameters are separable across genes. That is, we may obtain corresponding estimates $(\hat{\boldsymbol{\beta}}_j^{(m+1)}, \hat{\boldsymbol{\gamma}}_j^{(m+1)}, \hat{\phi}_j^{(m+1)})$ for each gene j separately of all other genes at each M-step, as these parameters are not shared across genes. This observation also facilitates the mini-batching approach described in Section 2.2.3. In this section, we first outline the M-step without mini-batching.

Specifically, for a given gene j , we utilize Iteratively Reweighted Least Squares (IRLS) with an embedded Coordinate Descent Algorithm (CDA) that updates $(\hat{\boldsymbol{\beta}}_j^{(m+1)}, \hat{\boldsymbol{\gamma}}_j^{(m+1)})$, given specified values of α , λ , $\hat{\mathbf{z}}^{(m)}$, and $\hat{\boldsymbol{\Psi}}^{(m)}$. The update scheme for a particular gene j consists of two nested loops: an inner loop corresponding to the CDA and an outer loop corresponding to the IRLS. The IRLS weights and working responses are updated in the outer loop, and fed into the inner loop to update the parameter estimates in a coordinate-descent fashion. After convergence of IRLS, we estimate $\hat{\phi}_j^{(m+1)}$ using Newton-Raphson. The entire M-step procedure is outlined in

Algorithm 1.

Algorithm 1 M-step

```

1: Input  $\hat{\mathbf{z}}^{(m)}$ 
2: Update  $\hat{\boldsymbol{\pi}}^{(m+1)}$ 
3: for  $j \in \{1, \dots, G\}$  do ▷ Cycle through all genes
4:   Input  $(\hat{\boldsymbol{\beta}}_j^{(m)}, \hat{\boldsymbol{\gamma}}_j^{(m)}, \hat{\phi}_j^{(m)})$ 
5:   while IRLS not converged do ▷ IRLS loop
6:     Compute IRLS weights and working response
7:     while CDA not converged do ▷ CDA loop
8:       Update  $\hat{\boldsymbol{\theta}}_j$  using previous CDA updates of  $\hat{\boldsymbol{\beta}}_j$ 
9:       for  $k \in \{1, \dots, K\}$  do
10:        Update  $\hat{\beta}_{jk}$ 
11:       for  $p \in \{1, \dots, P\}$  do
12:        Update  $\hat{\gamma}_{jp}$ 
13:       Replace  $(\hat{\boldsymbol{\beta}}_j^{(m+1)}, \hat{\boldsymbol{\gamma}}_j^{(m+1)}) \leftarrow (\hat{\boldsymbol{\beta}}_j, \hat{\boldsymbol{\gamma}}_j)$  ▷ Final CDA updates
14:       Update  $\hat{\phi}_j^{(m+1)}$  ▷ via Newton-Raphson
15: Output  $(\hat{\boldsymbol{\pi}}^{(m+1)}, \hat{\boldsymbol{\beta}}^{(m+1)}, \hat{\boldsymbol{\gamma}}^{(m+1)}, \hat{\boldsymbol{\phi}}^{(m+1)})$ 

```

$\hat{\beta}_{jk}$, $\hat{\gamma}_{jp}$, and $\hat{\boldsymbol{\theta}}_j$ denote the CDA updates of β_{jk} , γ_{jp} , and $\boldsymbol{\theta}_j$, respectively, and $\hat{\boldsymbol{\theta}}_j$ facilitates coordinate-wise descent updates of $\hat{\boldsymbol{\beta}}_j$. Notably, the fusion SCAD penalty is imposed on the updates of $\hat{\boldsymbol{\beta}}_j$, but updates of all other parameters are not penalized. Additional details of our CDA within IRLS implementation, including the forms of IRLS weights and working response updates, as well as the closed-form CDA updates pertaining to $\hat{\beta}_{jk}$, $\hat{\gamma}_{jp}$, and $\hat{\boldsymbol{\theta}}_j$ are given in Sections A2 and A3 of the Supplementary Material [Lim, 2020].

2.2.3. Mini-batching of genes. In order to make our algorithm more scalable to large datasets and to speed up the computation time, we mini-batch genes in the M-step, such that the parameters pertaining to only a subset of genes are estimated during a particular M-step iteration (Neal and Hinton, 1998). Mini-batching of genes is done by randomly drawing a prespecified proportion of the genes whose parameters are updated in that M-step iteration. By default, we set the mini-batch size at each iteration to $G/5$, or 20% of the total genes. Specifically, in the M-step, we run Algorithm 1 on the pre-specified subset of genes corresponding to the current mini-batch. That is, we update $\hat{\boldsymbol{\pi}}^{(m+1)}$ as before, but we update $(\hat{\boldsymbol{\beta}}_j^{(m+1)}, \hat{\boldsymbol{\gamma}}_j^{(m+1)}, \hat{\phi}_j^{(m+1)})$ only if gene j is in the mini-batch for that M step iteration. This is straightforward to implement due to the separability across genes in parameter updates. If gene j is not in the mini-batch for the cur-

rent M-step, current estimates are replaced with the previous iteration's estimates, i.e. $(\hat{\beta}_j^{(m+1)}, \hat{\gamma}_j^{(m+1)}, \hat{\phi}_j^{(m+1)}) \leftarrow (\hat{\beta}_j^{(m)}, \hat{\gamma}_j^{(m)}, \hat{\phi}_j^{(m)})$.

The E-step calculations are then done as in Section 2.2.1 using the parameter estimates of all genes, with the updated values pertaining to the included genes of the mini-batch. Although mini-batching requires more CEM iterations to converge, we found that it yields slightly faster convergence than updating all genes at every M-step.

2.2.4. Initialization. For each candidate value of K , we initialize our algorithm with very small penalty (small values of α and λ) from two sets of informative initial cluster labels (from naïve K-means and hierarchical clustering), and one set of random initial cluster labels drawn from the multinomial distribution with equal probabilities for the K clusters. We run the CEM with mini-batching on these 3 initializations until convergence, and select the converged model that yields the lowest BIC. Then, we run the standard EM algorithm from this selected starting point to convergence. This is similar to the “xCEM-EM” strategy proposed by Biernacki, Celeux and Govaert (2003), using the BIC for selection rather than the log-likelihood. Then, we use the estimates and posterior probabilities of the converged model, and perform selection of α and λ using EM. We repeat this initialization procedure for each candidate value of K , tuning α and λ jointly given K via warm starts. This procedure is described in more detail in Section 2.2.6.

2.2.5. Convergence. The stopping criterion for both the CEM and EM algorithms is based upon a threshold on the relative change in the Q function. Convergence is determined when $|(Q^{(m)} - Q^{(m-n_{mb})})/Q^{(m-n_{mb})}| < \epsilon_1$ where $Q^{(m)}$ is the value of the total Q-function conditional on current estimates at the m^{th} EM or CEM iteration, and n_{mb} is the minimum number of mini-batches that the data can be divided into (rounded up to the nearest integer). By default, $n_{mb} = 5$ (corresponding to mini-batching 20% of genes), and the left hand side (LHS) would be interpreted as the relative change in the Q-function across 5 iterations. Without mini-batching ($n_{mb} = 1$), the LHS simplifies to the relative change in the Q-function across one iteration.

In FSCseq, we set $\epsilon_1 = 10^{-6}$ as the default convergence threshold. Convergence of IRLS and CDA are described in Section A2 and A3 of the Supplementary Material [Lim, 2020].

2.2.6. Tuning the order and penalty parameters. In this section, we outline how optimal values of tuning parameters are selected in FSCseq via “warm starts” (Friedman et al., 2007; Friedman, Hastie and Tibshirani,

2010). Let $(K^*, \alpha^*, \lambda^*)$ denote the optimal values of (K, α, λ) , respectively. Denote candidate values of α as $(\alpha_1, \dots, \alpha_A)$ with $\alpha_1 < \dots < \alpha_A$, and candidate values of λ as $(\lambda_1, \dots, \lambda_L)$ with $\lambda_1 < \dots < \lambda_L$, such that there are $A \cdot L$ possible combinations of values for (α, λ) . Also, let $\hat{\Psi}_{a,\ell}$ denote the final estimates of the converged results pertaining to $\alpha = \alpha_a$ and $\lambda = \lambda_\ell$, for a given candidate value of K . Then, the tuning parameter selection proceeds as follows. First, initialize a model with fixed candidate value of K as described in Section 2.2.4 to obtain $\hat{\Psi}_{1,1}$. Then, leaving α fixed, increase λ to the next smallest value, λ_2 . Next, we run the penalized EM algorithm to convergence, utilizing the prior converged result $\hat{\Psi}_{1,1}$ as the initial value for Ψ , to obtain $\hat{\Psi}_{1,2}$. Then, we increase λ again and repeat the previous procedure, utilizing $\hat{\Psi}_{1,2}$ as the initial value for Ψ , to obtain $\hat{\Psi}_{1,3}$. Repeat until all L candidate values of λ are searched. Then, set $\alpha = \alpha_2$ and $\lambda = \lambda_1$, and run EM to convergence from $\hat{\Psi}_{1,1}$ to obtain $\hat{\Psi}_{2,1}$. Repeat this process for each value of α , until all $A \cdot L$ combinations of λ and α are searched. Finally, we repeat this entire procedure for each candidate value of K .

For a given combination of order K and penalty parameters (α, λ) , we may calculate the Bayesian Information Criterion (BIC) pertaining to the converged result, which has the form:

$$(9) \quad BIC = -2 \log \hat{L} + q \log(n \cdot G).$$

Here, $\hat{L}$ is the marginal likelihood given in (1) calculated with the estimates $\hat{\Psi}$ of the converged model, and q is the total number of estimated parameters. Let K_j denote the number of distinct cluster mean parameters for gene j , with each fused cluster considered as one distinct cluster. Then, $q = G + (K - 1) + \sum_{j=1}^G K_j$, corresponding to G gene-level dispersion parameters, $(K - 1)$ free mixing proportions (with $\pi_K = 1 - \sum_{k=1}^{K-1} \pi_k$), and a total of $\sum_{j=1}^G K_j$ distinct $\log_2$ cluster means. It is clear that $1 \leq K_j \leq K$ for all j , since $K_j = 1$ when all clusters are fused together in gene j (gene is determined to be cluster-nondiscriminatory), and $K_j = K$ when no pair of clusters are fused together in gene j . Additionally, given the gene-gene independence assumption from Section 2.1, we have an effective sample size of $n \cdot G$, corresponding to the number of entries in the gene expression matrix.

To select the optimal tuning parameter values, we calculate the BIC of each converged model using (9), and select $(K^*, \alpha^*, \lambda^*)$ as the unique combination of input values of (K, α, λ) corresponding to the converged model that yielded the lowest BIC.

2.2.7. Prediction of discovered subtypes in new patients. FSCseq is able to perform prediction on new samples based on the converged model with

input $(K^*, \alpha^*, \lambda^*)$. Given the RNA-seq read counts of cluster-discriminatory genes in a new sample, we may derive the posterior probabilities of subtype (cluster) membership for this new sample based on (7), where we assign subtypes as done in Section 2.2.1. Specifically, we calculate the posterior probability of cluster k membership in this new sample, $\hat{z}_{new,k}$, utilizing the estimated parameters from the converged FSCseq model based upon the original dataset (training set). This can be useful for predicting the subtype of new samples (prediction set) without having to re-cluster patients in future studies. We see that as in Section 2.2.1, the contribution of cluster-nondiscriminatory genes would cancel out in the posterior probability calculations of new samples.

However, correcting for sequencing depth in the new samples is nontrivial, since the size factors are specific to the training set samples used to fit the FSCseq model. To calculate the size factor of a new sample, we compare the counts of the new sample to the geometric mean of the counts of the training set samples, using the information in the training set as a pseudo-reference (Anders and Huber, 2010). Then, the estimated size factor of a new sample ($\hat{s}_{new}$) is given by:

$$(10) \quad \hat{s}_{new} = \underset{j}{\text{median}} \left[\frac{y_{new,j}}{(\prod_{v=1}^{n_{train}} y_{vj}^{train})^{1/n_{train}}} \right],$$

where n_{train} denotes the training set sample size, and $y_{new,j}$ and y_{vj}^{train} denote the count of gene j for the new sample and for training set sample v , respectively. As before, the size factors are calculated with the DESeq2 package.

3. Numerical Examples.

3.1. *Simulations.* To evaluate our proposed method, we simulated data across an extensive set of conditions, varying the following factors: number of true underlying clusters (K_{true}), sample size (n), $\log_2$ fold change between cluster means (LFC), proportion of cluster-discriminatory genes (p_{DE}), baseline $\log_2$ mean count (β_0) of genes, and overdispersion (ϕ_0) of counts across samples for a given gene. We fix LFC , β_0 , and ϕ_0 to be the same across all simulated genes in order to demonstrate the performance of FSCseq in a controlled setting with respect to these factors. We also fixed the number of simulated genes at $G = 10000$. Of these, we specified the first $(p_{DE} \cdot G)$ genes to be cluster-discriminatory, and the rest of the genes to be nondiscriminatory, and we denote the set of cluster-discriminatory genes as G_d . In cancer subtyping data, it is common to see specific genes

upregulated or downregulated for just one subtype (Yersal, 2014). Thus, for each cluster-discriminatory gene $j \in G_d$, we randomly select one cluster k_* , $1 \leq k_* \leq K_{true}$, and randomly up-regulate ($\beta_{jk_*} = \beta_0 + LFC$) or down-regulate ($\beta_{jk_*} = \beta_0 - LFC$) the $\log_2$ mean for that cluster, while keeping the expression of the remaining clusters the same as the baseline ($\beta_{jk'} = \beta_0$, for all $k' \neq k_*$), where β_{jk} denotes the $\log_2$ mean of gene j for a sample in cluster k . For each nondiscriminatory gene $j \notin G_d$, we set $\beta_{jk} = \beta_0$ for all $k = 1, \dots, K_{true}$. In this way, each cluster-discriminatory gene is expected to have one cluster with $\log_2$ mean different from β_0 , while each nondiscriminatory gene is expected to have $\log_2$ mean equal to β_0 across all clusters.

We simulated datasets with $K_{true} = (2, 4)$ underlying groups with 25 and 50 samples per cluster, such that $n = (50, 100)$ for $K_{true} = 2$, and $n = (100, 200)$ for $K_{true} = 4$. Additionally, we considered $p_{DE} = (0.025, 0.050)$, $\beta_0 = (8, 12)$, $\phi_0 = (0.15, 0.35, 0.50)$ and $LFC = (1, 2)$ in our simulations. These values were determined using RNA-seq data from the TCGA Breast Cancer project (Grossman et al., 2016). Specifically, after removing outlier genes, we fit a negative binomial regression model to expression counts from each gene one-by-one, utilizing $\log_2$ size factors calculated from DESeq2 as offsets and the 5 annotated PAM50 subtypes as covariates, such that a cluster mean is estimated for each subtype. The corresponding $n \times 5$ design matrix (via cell-means coding) has i^{th} row and k^{th} column entry $x_{ik} = 1$ if tumor sample i has annotated PAM50 subtype k , and $x_{ik} = 0$ otherwise, for $i = 1, \dots, n$ and $k = 1, \dots, 5$. Based upon the results from these models, the 50th and 75th quantiles of the estimated LFC s in the TCGA Breast Cancer dataset were 0.875 and 1.72, and the median estimated $\hat{\beta}_{jk}$ and $\hat{\phi}_j$ were 10.2 and 0.354, respectively. These values informed our simulation parameters above. We also simulated realistic between-sample technical variation due to sequencing depth by simulating size factors for each sample $i = 1, \dots, n$ from $s_i \sim N(1, 0.25)$, based upon the estimates acquired by DESeq2 on the same TCGA samples.

To prevent our simulation model from being identical to our model framework, we also introduced Gaussian noise to the $\log_2$ expression of each count, drawn from $\sigma_{ij} \sim N(0, 0.1)$ for each sample i and gene j . Then, the expression of each gene $j = 1, \dots, G$ for subjects $i = 1, \dots, n$ in cluster $k = 1, \dots, K_{true}$ was simulated from the negative binomial distribution $NB(\mu_{ijk}, \phi_0)$ with mean μ_{ijk} , such that

$$(11) \quad \log_2(\mu_{ijk}) = s_i + \beta_{jk} + \sigma_{ij},$$

and dispersion parameter ϕ_0 . We simulated 100 datasets for the main simulation analyses in Section 3.1.1, and 25 datasets for the rest of the analyses.

In order to mimic what is done in real data in each simulated dataset, we used thresholds on the median count and the MAD value to pre-filter low-count and low-variable genes, respectively, keeping features that yielded median count > 100 and MAD score in the top 50^{th} quantile. We ran FSC-seq on each dataset after pre-filtering. Optimal values for tuning parameters $(K^*, \alpha^*, \lambda^*)$ were found by searching candidate values of $K = \{2, \dots, 6\}$; $\alpha = \{0.01, 0.05, 0.10, \dots, 0.50\}$; and $\lambda = \{0.25, 0.50, \dots, 5.00\}$, as described in Section 2.2.6.

n	LFC	p _{DE}	ϕ_0	$\bar{K}^*$	$\overline{ARI}$	$\overline{pARI}$	$\overline{TPR}$	$\overline{FPR}$
100	1.0	0.025	0.15	4.05	0.994	0.998	0.703	0.0007
			0.35	3.78	0.926	0.922	0.554	0.0012
			0.50	2.28	0.233	0.196	0.077	0.0003
		0.050	0.15	4.02	0.998	1.000	0.685	0.0006
			0.35	4.01	0.994	0.994	0.608	0.0015
			0.50	3.80	0.941	0.923	0.432	0.0013
	2.0	0.025	0.15	4.06	0.996	0.999	0.780	0.0004
			0.35	4.01	0.999	1.000	0.757	0.0004
			0.50	4.02	0.998	1.000	0.726	0.0014
		0.050	0.15	4.05	0.995	0.999	0.781	0.0012
			0.35	4.04	0.997	0.999	0.747	0.0010
			0.50	4.01	0.999	1.000	0.725	0.0026
200	1.0	0.025	0.15	4.17	0.988	0.993	0.741	0.0014
			0.35	4.07	0.989	0.992	0.700	0.0018
			0.50	3.97	0.975	0.974	0.645	0.0012
		0.050	0.15	4.15	0.992	0.996	0.727	0.0017
			0.35	4.04	0.997	0.997	0.700	0.0034
			0.50	4.00	1.000	1.000	0.662	0.0014
	2.0	0.025	0.15	4.11	0.992	0.999	0.803	0.0019
			0.35	4.07	0.998	1.000	0.798	0.0004
			0.50	4.09	0.995	0.999	0.782	0.0007
		0.050	0.15	4.05	0.996	0.999	0.809	0.0019
			0.35	4.04	0.997	0.999	0.778	0.0007
			0.50	4.04	0.998	0.998	0.767	0.0005

TABLE 1

Results of FSCseq clustering, feature selection, and prediction on subset of combinations of simulation conditions with $K_{\text{true}} = 4$ true number of clusters and $\beta_0 = 12$ baseline $\log_2$ mean. Clustering is measured by average optimal order ($\bar{K}^*$) and adjusted rand index ($\overline{ARI}$). Prediction is done on an independently simulated dataset with $n_{\text{pred}} = 25$ samples, and is measured by the average ARI between predicted and simulated cluster labels, and is denoted $\overline{pARI}$. Feature selection performance is measured by mean true positive rate ($\overline{TPR}$) and false positive rate ($\overline{FPR}$) of cluster-discriminatory gene discovery.

3.1.1. *FSCseq performance across varied conditions.* In this section, we evaluate the clustering, feature selection, and prediction performance of FSCseq under a subset of simulated conditions that most closely reflected the estimates from the TCGA data. Thus, this subset represents conditions that are most similar to what one may expect from real data.

Accuracy of derived clustering results was measured by two metrics: optimal order obtained (K^*) and concordance (or agreement) in cluster assignment with the truth, measured by the Adjusted Rand Index (ARI). Feature selection performance was measured by the true positive rate (TPR) and false positive rate (FPR) of discovering true simulated cluster-discriminatory genes. Specifically, TPR is the proportion of true simulated cluster-discriminatory genes ($\{j : j \in G_d\}$) that were correctly determined to be discriminatory by FSCseq, while FPR is the proportion of true simulated nondiscriminatory genes ($\{j : j \notin G_d\}$) that were incorrectly determined to be discriminatory. To assess the prediction performance of FSCseq in new data, we also independently simulated a prediction set of $n_{pred} = 25$ samples for each simulated dataset, based upon the same set of simulated conditions. We then took the estimates obtained from the converged FSCseq model on the original simulated dataset (training set), and then applied the procedure from Section 2.2.7 to the corresponding test set samples to obtain the predicted cluster labels for these samples. We again measured concordance of predicted clusters with the true simulated cluster labels in the test set by ARI , and we denote this “prediction” ARI , or $pARI$. In Table 1, all performance metrics are averaged across the 100 simulated datasets, with each row denoting a unique combination of conditions. The averaged value is denoted by bar notation, e.g. $\bar{K}^*$ is the average K^* across the 100 datasets corresponding to a particular set of simulated conditions.

We found that the clustering performance of FSCseq was generally robust to the magnitude of ϕ_0 , even when n is small (Table 1). One exception was found when $n = 100$, $p_{DE} = 0.025$ and $LFC = 1$ with $K_{true} = 4$, i.e. when both the proportion of discriminatory genes and the LFC across these discriminatory genes were small. In such a case, the clusters may be poorly separated and clustering results may be confounded by higher ϕ_0 , compounded by the small average per-cluster sample size (25 per cluster). We note that none of the compared methods perform well in this case (see Section C2 of the Supplementary Material [Lim, 2020]), although FSCseq yields higher mean ARI .

As n , LFC , or p_{DE} is increased, we also found that the sample clusters become more distinct from one another, yielding better clustering results. Increasing n and LFC also tended to result in an increase in

TPR , reflecting the increasing sensitivity of our variable selection method under larger sample size and fold change differences between clusters. However as ϕ_0 increased, we found a decrease in the TPR , where the higher overdispersion confounded the true differences in expression of the cluster-discriminatory genes and reduced our sensitivity to select cluster discriminatory genes. Interestingly, the FPR did not vary significantly across changes in these conditions, showing the reliability of cluster-discriminatory features discovered by FSCseq, and the FPR was relatively low in general. Figure 1 shows a scatterplot of TPR vs FPR from each individual simulated dataset from Table 1, stratified by simulated LFC and n . In addition to showing that larger n or LFC and a smaller value of ϕ_0 yielded higher TPR , we also show that selecting the correct order was important in attaining both high TPR and low FPR . This is because FSCseq more accurately selected cluster-discriminatory genes if the correct number of groups is identified.

FSCseq’s prediction performance was also found to be highly dependent on its initial clustering performance during model training, i.e., $pARI$ on simulated test subjects was correlated to ARI during FSCseq training. This reflects the fact that FSCseq models that correctly clustered samples also tended to yield more accurate parameter estimates, which allowed for more accurate prediction of cluster identity in new samples. Since FSCseq generally yielded high ARI , we also generally found high average $pARI$ for simulated conditions, reflecting the accuracy of predictions via FSCseq. Overall, we found favorable performance of FSCseq over a variable series of conditions.

Results of FSCseq analyses on all simulated datasets encompassing all simulation conditions can be found in Section B of the Supplementary Material [Lim, 2020].

3.1.2. Performance comparison across methods. We compared performance of FSCseq with 7 competing methods: `iCluster+` (iCl), average-linkage hierarchical clustering (HC), K-medoids (KM), `NB.MClust` (NBMB), and `mclust` on log, variance stabilizing, and rlog transformations of the normalized data (IMC, vMC, and rMC). FSCseq can intrinsically adjust for differences in sequencing depth using derived size factors as described earlier. For all other methods, we use either normalized counts (iCl, HC, KM, NBMB, IMC) or transformed normalized counts (vMC, rMC) for clustering to adjust for differences in sequencing depth, and these counts are computed from DESeq2 (Love, Huber and Anders, 2014). We also note that KM was performed on log-transformed normalized counts (Jaskowiak, Costa and Campello, 2018) as in IMC. Because there was no default method of select-

FIG 1. Scatterplot of true positive rate (TPR) vs false positive rate (FPR) in discovering cluster-discriminatory genes in simulated datasets via FSCseq. Displayed points correspond to simulated datasets with $K_{true} = 4$ underlying clusters, with $n = 100$ (top) or 200 (bottom) and simulated LFC = 1 ('Low', left) or 2 ('Moderate', right). Red points indicate that the correct order was uncovered ($K^* = K_{true}$), and blue points indicate that an incorrect order was uncovered. Squares, circles, and X's indicate low ($\phi_0 = 0.15$), moderate ($\phi_0 = 0.35$), and high ($\phi_0 = 0.50$) levels of overdispersion, respectively. In general, TPR is higher for larger n and larger LFC. TPR is higher and FPR is lower for results that yielded the correct order, or for smaller values of ϕ_0 . For ease of visualization, a total of 15 outlier points (of 2400 total) were removed in this figure with $0.015 < FPR < 0.060$. The figure without excluding outlier points can be found in Supplementary Figure 2, Section B of the Supplementary Material [Lim, 2020].

ing K^* for HC, we chose to utilize the gap statistic, as proposed by Tibshirani, Walther and Hastie (2001), and implemented the order selection in this setting via the NbClust R package (Charrad et al., 2014). For KM, we selected K^* to be the value of K that maximized the average silhouette width (Reynolds, Richards and Rayward-Smith, 2004). For NBMB and the transformed MC methods, we used default selection procedures for K^* directly available from the respective NB.MClust and mclust R packages. For iCluster+, we describe an automated procedure for selecting K^* in Section C1 of the Supplementary Material [Lim, 2020].

Cluster-discriminatory gene selection performance was not compared across methods as only FSCseq had an automated procedure for feature (gene) selection. We note that iCluster+ does perform automatic gene selection via the L1 penalty; however, the iCluster+ manual recommends thresholding an arbitrary proportion of genes for selection rather than utilizing its automatic feature selection process (Mo and Shen, 2019). One could similarly perform thresholding on the estimated coefficients from the mclust or NB.MClust results, but this would also depend on an arbitrary threshold. We compared the clustering performance of these methods under different numbers of underlying groups $K_{true} = (2, 4)$ with 25 samples per cluster. We also varied ϕ_0 to test each method’s robustness to the level of overdispersion in the simulated counts. Here, we define order accuracy (OA) as the proportion of simulated datasets (within a given simulated condition) that yielded the correct order, such that $K^* = K_{true}$.

Figure 2 shows violin plots of order accuracy OA and ARI . Here, we fixed $LFC = 1$, $p_{DE} = 0.05$, $n/K_{true} = 25$, and $\beta_0 = 12$, and we varied $\phi_0 = \{0.15, 0.35, 0.50\}$. Generally, FSCseq yielded clusters that had the highest concordance with the simulated clusters. Many of the methods performed competitively in clustering when $K_{true} = 2$, but FSCseq and iCluster+ performed markedly better than competing methods when $K_{true} = 4$. Throughout all conditions, iCluster+ tended to select a larger optimal order $K^* > K_{true}$, which caused performance to be significantly better under $K_{true} = 4$ than $K_{true} = 2$. KM performed sporadically throughout, suggesting lack of robustness in clustering performance to the magnitude of ϕ_0 . Generally, NBMB yielded the lowest ARI compared to all other methods. We postulate that this is because NBMB utilizes cluster-specific dispersion parameters for each gene in their model, which may yield unstable estimates due to smaller sample sizes in estimating each cluster-level dispersion parameter.

In general, we observed poorer clustering performance of most methods in $K_{true} = 4$ compared to $K_{true} = 2$, with markedly lower average ARI

FIG 2. Violin plots of Order Accuracy (OA, in red) and average cluster concordance with the truth (ARI, in blue) from simulation results with $K_{true} = 2$ and $K_{true} = 4$. We compared performance of FSCseq (FSC), *iCluster+* (iCI), hierarchical clustering (HC), K-medoids (KM), *NB.MClust* (NBMB), and *mclust* on log/variance-stabilizing/rlog transformed data (lMC/vMC/rMC). Simulated overdispersion was varied with $\phi_0 = (0.15, 0.35, 0.50)$ to test each method's robustness to the magnitude of overdispersion. Other simulation parameters were fixed at $LFC = 1$, $p_{DE} = 0.05$, $\beta_0 = 12$, and $n/K_{true} = 25$ for both $K_{true} = 2$ (top) and $K_{true} = 4$ (bottom). We fix n/K_{true} here to show the effect of varying K_{true} on performance, with a fixed number of samples per cluster. Many methods perform competitively when $K_{true} = 2$, but FSCseq attains the highest average ARI overall, and yields high performance that is very robust to the magnitude of ϕ_0 .

for HC and the transformed MC runs (Figure 2). When K_{true} is larger, there is a smaller proportion of samples within each cluster-discriminatory gene with $\log_2$ mean different from β_0 , since only one cluster is differentially regulated by LFC for these genes. Thus, we generally observe a significant decrease in performance for larger K_{true} , although FSCseq’s performance is most robust to this effect. In addition, similarly sporadic performance of HC in RNA-seq was also observed previously (Vidman, Källberg and Rydén, 2019), suggesting unreliability of HC for clustering RNA-seq gene expression. Finally, although the MC methods performed similarly on average, the best performing method between these three transformations varied for each set of simulation conditions, reflecting the fact that the optimal transformation may not be known in advance. Comparisons between FSCseq and these competing methods across the entire extensive set of simulation conditions can be found in Section C2 of the Supplementary Material [Lim, 2020].

	γ_0	$\bar{K}_2^*$	$\overline{ARI}_2$	$\bar{K}_4^*$	$\overline{ARI}_4$
FSC	2	2.00	1.000	4.16	0.971
	3	2.00	1.000	4.00	0.965
iCl	2	4.68	0.450	5.00	0.779
	3	3.00	0.287	4.00	0.332
HC	2	2.00	0.000	2.00	0.001
	3	2.00	-0.002	2.00	-0.001
KM	2	2.00	0.989	2.00	0.002
	3	2.00	-0.002	2.00	-0.001
NBMB	2	2.28	0.055	2.32	0.004
	3	2.32	0.043	2.04	-0.001
lMC	2	3.92	0.498	2.00	0.001
	3	4.00	0.499	2.00	-0.001
vMC	2	3.92	0.498	2.00	0.001
	3	4.00	0.499	2.00	-0.001
rMC	2	2.04	0.024	2.00	0.001
	3	3.36	0.368	2.00	-0.001

TABLE 2

Average obtained order $\bar{K}^*$ and average ARI for competing methods with $K_{true} = 2$ and $K_{true} = 4$ underlying groups, in the presence of simulated batch effects γ_0 . We shorten notation in this table by denoting the $\bar{K}^*$ value as $\bar{K}_2^*$ for $K_{true} = 2$, and as $\bar{K}_4^*$ for $K_{true} = 4$. Similarly, we denote ARI as $\overline{ARI}_2$ for $K_{true} = 2$, and as $\overline{ARI}_4$ for $K_{true} = 4$. Tabulated results are from datasets with $n/K_{true} = 50$ simulated samples per cluster, with fixed $LFC = 2$, $p_{DE} = 0.05$, and $\beta_0 = 12$, $\phi_0 = 0.35$. Simulated effect across batches was varied, such that $\gamma_0 = 2, 3$.

3.1.3. *Performance under simulated batch effects.* We also evaluated the clustering and order selection performance of these methods in the presence

of batch effects. Here we simulate two batches in a subset of simulation conditions, where batch effects were imposed on a subset of genes for each simulated dataset. Simulated RNA-seq read counts were generated in a manner similar to (11), except now we assume $\log_2(\mu_{ijk}) = s_i + \beta_{jk} + \sigma_{ij} + \{I(j \in G_b) \cdot \gamma_0 \cdot Batch_i\}$, where G_b is the set of genes that are simulated to be batch-affected, γ_0 is a fixed batch effect on the $\log_2$ scale, and $Batch_i = \{-0.5, 0.5\}$. We randomly assign $Batch_i = -0.5$ or $Batch_i = 0.5$ for $i = 1, \dots, n$ with equal probability, and G_b is comprised of a randomly selected 50% subset of the G genes. For each simulation we performed FSCseq with the batch variable as a covariate, and compared performance via each method on simulation conditions with $n/K_{true} = 50$, $LFC = 2$, $p_{DE} = 0.05$, $\beta_0 = 12$, and $\phi_0 = 0.35$. We chose this subset to isolate the confounding effect of batch, since these simulated conditions yielded good clustering performance when batch effects were not simulated.

Results on batch-simulated datasets are shown in Table 2. FSCseq is the only method that performs robustly to the magnitude of γ_0 , showing its ability to properly adjust for the confounding batch effects. Although KM does not perform any correction for batch, it surprisingly performed well under $K_{true} = 2$ and $\gamma_0 = 2$. However, because KM does not actually correct for batch, KM performs expectedly poorly under larger simulated batch effects $\gamma_0 = 3$.

3.2. Application to TCGA Breast Cancer RNA-seq dataset. In this section, we compared performance of these clustering methods on the TCGA Breast Cancer (BRCA) dataset. The dataset and annotations were obtained from the National Cancer Institute GDC Portal (Grossman et al., 2016). Gene expression RNA-seq data was sequenced via the Illumina HiSeq 2000 platform, and alignment and quantification were done using the GDC’s mRNA Analysis Pipeline, which can be found in the “GDC Data User’s Guide” online (Grossman et al., 2016). The raw read counts were downloaded using the TCGAbiolinks R package (Colaprico et al., 2015).

Subtyping was done previously using the PAM50 classifier: a set of 50 genes that has been heavily investigated as driving genes of breast cancer subtypes (Koboldt et al., 2012). These genes are also known to be primarily expressed in tumor cells. However, tumor bulk samples are typically comprised of heterogeneous mixtures of cell types, and may poorly reflect the expression profile of the tumor if the sample is of low tumor purity, i.e. if it is composed of a small proportion of tumor cells. Attempts at unsupervised clustering of low-purity samples based on subtype may emphasize genes that are associated with purity and confound results (Aran, Sirota and Butte,

2015). Thus, we first included in our analysis only samples whose estimated ABSOLUTE purity (Carter et al., 2012) was greater than 0.9. This is done in order to (1) properly cluster samples via each compared method based upon tumor subtypes, and (2) compare each clustering method fairly with the annotated subtypes, which are based upon tumor-intrinsic PAM50 classifier genes. We also performed our analyses on all samples without purity filtering, and compared our results with those from Koboldt et al. (2012), which performed similar clustering analyses on all samples.

After filtering samples by purity, we found very low incidence of the HER2-enriched and normal-like subtypes, and thus removed these subtypes from our analysis. Then, we pre-filtered low read count genes by low median normalized count, and low-variable genes by low MAD scores. For reproducibility, we detail the pre-processing steps in Section D1 of the Supplementary Material [Lim, 2020]. After all filtering steps, the dataset contained 123 samples from 3 distinct subtypes, and 4038 genes. We measured concordance with annotated sample groupings (tumor subtypes in the TCGA BRCA dataset) by four clustering metrics: ARI, normalized mutual information (NMI) (Strehl and Ghosh, 2003), normalized variation of information (NVI) (Reichart and Rappoport, 2009), and normalized information distance (NID) (Vinh, Epps and Bailey, 2010).

The RNA-seq experiments were performed on separate wells of multiwell plates, with multiple samples sequenced on the same plate at different times. Therefore, samples across plates may express variability similar to batch effects (Reese et al., 2013). We performed FSCseq analysis with (FSC_{adj}) and without (FSC) adjustment for this confounding effect, and compared the results with competing methods. For proper analysis via FSC_{adj} , we agglomerated singleton plates (plates with just one sample) into one joint plate.

For this data, we expanded the grid of candidate values of K to $\{2, \dots, 8\}$. All clustering metrics were calculated with respect to the mRNA PAM50 subtype annotations ('anno') by the GDC. Potential confounders like age, ethnicity, and tumor grade were not included here because they were not of major interest in our analysis, but they can be incorporated as additional covariates in FSCseq. Results are given in Table 3.

FSC selected order $K^* = 5$, but after adjusting for plate effects, FSC_{adj} uncovered the true order of $K^* = 3$. Similar to the trends seen in simulations, iCluster+ selected a larger number of clusters $K^* = 8$, while HC and KM selected a smaller $K^* = 2$. The transformed MC methods and NBMB selected a slightly larger $K^* = 4$. Of the competing methods, FSC_{adj} yielded the clusters with the highest concordance with the annotations, with an ARI

	K^* (3)	ARI (1)	NMI (1)	NVI (0)	NID (0)
anno	3				
FSC	5	0.375	0.422	0.667	0.578
FSC_{adj}	3	0.594	0.642	0.523	0.358
iCl	8	0.204	0.32	0.737	0.68
HC	2	0.481	0.476	0.559	0.524
KM	2	0.486	0.489	0.539	0.511
NBMB	4	0.447	0.461	0.658	0.539
IMC	4	0.437	0.464	0.649	0.536
vMC	4	0.428	0.465	0.65	0.535
rMC	4	0.436	0.464	0.649	0.536

TABLE 3

Selected order (K^*) and clustering concordance between compared methods and annotated TCGA Breast Cancer subtypes. FSC_{seq} was run with adjustment (FSC_{adj}) and without adjustment (FSC) for plate effect, and each of the clustering labels were compared to annotated subtypes (anno). For each column, the value of the best performing metric is colored in red. The values in parentheses in the column headings represent the optimal value for that metric. For ARI and NMI, values close to 1 indicate good clustering, and values close to 0 indicate poor clustering. For NVI and NID, values close to 0 indicate good clustering, and values close to 1 indicate poor clustering.

of 0.594, followed by KM and HC with ARI of 0.486 and 0.481, respectively. FSC_{adj} additionally yielded the best NMI, NVI, and NID.

Feature selection via FSC and FSC_{adj} respectively determined a total of 2693 and 2238 genes to be cluster-discriminatory, while the automatic feature selection by the L1 penalty in iCluster+ determined significantly more genes (4035) to be cluster-discriminatory. Of the 41 PAM50 genes that were included in our analysis after pre-filtering, FSC determined all 41 likely to be cluster-discriminatory, and FSC_{adj} determined 38 of them to be likely. This shows FSC_{seq} 's ability to identify subtype-discriminating genes in real cancer data, as the relevance of PAM50 genes in RNA-seq has been shown by previous studies (Picornell et al., 2019; Raj-Kumar et al., 2019). Figure 3 shows a heatmap of the PAM50 genes with notations on their inclusion/exclusion through the pre-filtering step, and through FSC_{seq} 's simultaneous feature selection. Column (samples) ordering is based on annotated subtypes, and samples are ordered within subtypes by decreasing order of maximum posterior probability from FSC_{adj} results. KM performs well in terms of ARI, but it only grouped samples by basal vs. non-basal subtypes and didn't distinguish between the Luminal A and Luminal B subtypes. Similarly, IMC performs well in grouping basal samples, but over-selected the order ($K^* = 4$) and was not able to distinguish between Luminal A and B samples. Overall, the cluster labels from FSC_{adj} most accurately clustered the samples according to the 3 underlying subtypes. Additional

results from all cluster-discriminatory genes with all compared methods can be found in Section D2 of the Supplementary Material [Lim, 2020].

In addition to the PAM50 genes, other significant cluster-discriminatory genes found by both FSC and FSC_{adj} included genes like CDH1, which has been linked to lobular or ER+ tumor carcinomas (Yang, Ding and Huang, 2015), and CHEK2, which has been studied for its association with the luminal subtypes (Huszno and Kolosza, 2019). Additionally, we performed gene ontology analysis on the discovered sets of cluster-discriminatory genes from both FSC and FSC_{adj} . Enriched pathway analysis showed that genes from both sets were involved in known key pathways associated with cancer. Such pathways included signaling pathways like EIF2 and AHR, as well as pathways involved in mitosis and other molecular mechanisms of cancer. Both sets of genes also contained many gene ontology (GO) biological processes pertaining to the cell cycle, consistent with previous studies that found such enrichment in basal-like subtypes of breast cancer (Yang et al., 2017; Yang, Gao and Luo, 2019). We further validated our results by testing for overlaps with known gene sets via GSEA analysis (Mootha et al., 2003; Subramanian et al., 2005). We first grouped cluster-discriminatory genes from FSC_{adj} using *MBCluster.Seq* (Si et al., 2013), then separately analyzed the subset of genes that were upregulated (*basalUP*), and those that were downregulated (*basalDOWN*) for the most distinct basal subtype. Significantly overlapping gene sets with *basalUP* and *basalDOWN* were sets of genes known to differentiate subtypes of breast cancer. Additionally, the top overlapping gene set for *basalUP* and *basalDOWN* corresponded to the specific collections of genes known to be upregulated and downregulated, respectively, in the basal subtype of breast cancer (Smid et al., 2008). Detailed results of our gene enrichment analyses can be found in Section D3 of the Supplementary Material [Lim, 2020].

We measured performance in prediction via leave-one-out cross-validation by training a model on cluster-discriminatory genes from FSC with one sample held out, and using this model to predict the cluster label of the held-out sample. In this way, prediction was performed on each sample once. We observed very high overall accuracy of 0.951, showing robustness of our prediction framework in real data settings.

Finally, we also clustered the TCGA samples without pre-filtering samples for purity and including all 5 original subtypes ($K_{true} = 5$), as done previously by Koboldt et al. (2012). We anticipated that there would exist a larger number of underlying groups in this dataset due to heterogeneity caused by low purity samples, which is also reflected by the results in Koboldt et al. (2012). Thus, we expanded the search range of K to $K = \{2, \dots, 15\}$.

FIG 3. Heatmap of the PAM50 genes included in FSCseq analyses, with row annotations for feature selection and pre-filtering (left) and column annotations for clustering labels (top). Column ordering is based on annotated subtypes, and samples are ordered within subtypes by decreasing order of maximum posterior probability from FSCadj results. 9 PAM50 genes did not pass the pre-filtering (PF) threshold: TMEM45B, MDM2, FGFR4, ACTR3B, FOXC1, MIA, EGFR, CCNE1, and ORC6L. Of the 41 remaining PAM50 genes, FSC and FSCadj found 41 and 38 of the PAM50 genes were cluster-discriminatory, respectively. Additionally, all clustering labels distinguish well between Basal and Luminal subtypes, but FSCadj best distinguishes between Luminal A and Luminal B samples.

As expected, all methods yielded clusters of poor concordance with the annotated PAM50 subtypes, with FSC and FSC_{adj} yielding larger orders of $K^* = 7$ and $K^* = 8$, respectively, and $ARIs$ of 0.316 and 0.245, respectively. Koboldt et al. (2012) performed unsupervised clustering on the same samples and found 13 clusters, and they performed semi-supervised clustering with an intrinsic list of significant genes that similarly yielded 14 clusters. Their results were similarly poor in agreement with the annotated subtypes (ARI of 0.272 and 0.258 for unsupervised and semi-supervised clusters, respectively). Compared to these results, FSCseq was able to select K^* that is closer to the true number of subtypes, however the potential confounding in expression due to variable sample purity appears to cloud the ARI performance of all methods in this setting. All results from this analysis can be found in Section D4 of the Supplementary Material [Lim, 2020].

4. Discussion. Our findings from our simulations and real data applications give evidence to the utility of our method across a varied set of conditions. In our simulations, we found very good feature selection performance throughout simulated conditions, and FSCseq outperformed existing clustering methods for RNA-seq data. In addition, the markedly low $FPRs$ in gene discovery may help researchers attain a higher degree of confidence in the discovered list of genes, while limiting expended resources on validation studies in clinical settings. In the TCGA BRCA dataset, FSCseq clusters aligned best to previously discovered subtypes when correcting for batch effects (plate), although differing levels of heterogeneity in samples confounded the results. The annotated subtypes were discovered by analyzing just the genes that were clinically validated to be significant in discriminating across breast cancer subtypes. We showed that FSCseq is able to perform comparably, despite using no such *a priori* knowledge of significant genes. Moreover, of the PAM50 genes included in our TCGA BRCA analysis, the FSCseq workflow identified most of these genes as discriminatory across the resulting clusters. As these genes have been validated to be significant on both microarray and RNA-seq platforms, this emphasizes FSCseq’s ability to uncover significant genes in real data settings.

SUPPLEMENTARY MATERIAL

Supplement A: Supporting Information for Model-based Feature Selection and Clustering of RNA-seq Data for Unsupervised Subtype Discovery

(doi: [COMPLETED BY THE TYPESETTER](#); .pdf). Contains links to code and implementation of FSCseq and additional details of our algorithm in

Section A, extensive results from simulations in Sections B-C, and details and results from the TCGA BRCA dataset in Section D.

References.

- AL-KHASAWN, M. F. (2010). Estimating the Negative Binomial Dispersion Parameter. *Asian Journal of Mathematics & Statistics* **3** 1–15.
- ANDERS, S. and HUBER, W. (2010). Differential expression analysis for sequence count data. *Genome Biology* **11** R106.
- ARAN, D., SIROTA, M. and BUTTE, A. J. (2015). Systematic pan-cancer analysis of tumour purity. *Nature Communications* **6**.
- BIERNACKI, C., CELEUX, G. and GOVAERT, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis* **41** 561–575.
- BOURGON, R., GENTLEMAN, R. and HUBER, W. (2010). Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences* **107** 9546–9551.
- CARTER, S. L., CIBULSKIS, K., HELMAN, E., MCKENNA, A., SHEN, H., ZACK, T., LAIRD, P. W., ONOFRIO, R. C., WINCKLER, W., WEIR, B. A., BEROUKHIM, R., PELLMAN, D., LEVINE, D. A., LANDER, E. S., MEYERSON, M. and GETZ, G. (2012). Absolute quantification of somatic DNA alterations in human cancer. *Nature Biotechnology* **30** 413–421.
- CELEUX, G. and GOVAERT, G. (1990). Stochastic Algorithms for Clustering. In *Compstat* (K. MOMIROVIĆ and V. MILDNER, eds.) 3–8. Physica-Verlag HD, Heidelberg.
- CELEUX, G. and GOVAERT, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis* **14** 315–332.
- CHARRAD, M., GHAZZALI, N., BOITEAU, V. and NIKNAFS, A. (2014). NbClust: An R-Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software* **61**.
- CHIA, S. K., BRAMWELL, V. H., TU, D., SHEPHERD, L. E., JIANG, S. and ET AL., T. V. (2012). A 50-Gene Intrinsic Subtype Classifier for Prognosis and Prediction of Benefit from Adjuvant Tamoxifen. *Clinical Cancer Research* **18** 4465–4472.
- CHUNG, N., ZHANG, X. D., KREAMER, A., LOCCO, L., KUAN, P. F. and ET AL., S. B. (2008). Median Absolute Deviation to Improve Hit Selection for Genome-Scale RNAi Screens. *Journal of Biomolecular Screening* **13** 149–158.
- COLAPRICO, A., SILVA, T. C., OLSEN, C., GAROFANO, L., CAVA, C., GAROLINI, D. and ET AL. (2015). TCGAblinks: an R/Bioconductor package for integrative analysis of TCGA data. *Nucleic Acids Research* **44** e71–e71.
- DELLAERT, F. (2002). The expectation maximization algorithm Technical Report, Georgia Institute of Technology.
- FAN, J. and LI, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association* **96** 1348–1360.
- FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **33**.
- FRIEDMAN, J., HASTIE, T., HÖFLING, H. and TIBSHIRANI, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics* **1** 302–332.
- GARCIA, R. I., IBRAHIM, J. G. and ZHU, H. (2009). Variable Selection in the Cox Regression Model with Covariates Missing at Random. *Biometrics* **66** 97–104.
- GOLUB, T. R., SLONIM, D. K., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J. P., COLLIER, H., LOH, M. L., DOWNING, J. R., CALIGIURI, M. A.,

- BLOOMFIELD, C. D. and LANDER, E. S. (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science* **286** 531–537.
- GROSSMAN, R. L., HEATH, A. P., FERRETTI, V., VARMUS, H. E., LOWY, D. R., KIBBE, W. A. and STAUDT, L. M. (2016). Toward a Shared Vision for Cancer Genomic Data. *New England Journal of Medicine* **375** 1109–1112.
- HILBE, J. M. (2009). *Modeling Count Data*. Cambridge University Press.
- HOERL, A. E. and KENNARD, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* **12** 55–67.
- HUSZNO, J. and KOLOSZA, Z. (2019). Molecular characteristics of breast cancer according to clinicopathological factors. *Molecular and Clinical Oncology*.
- INOUE, D. I., YANG, E., ALLEN, G. I. and RAVIKUMAR, P. (2017). A review of multivariate distributions for count data derived from the Poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics* **9** e1398.
- JASKOWIAK, P. A., COSTA, I. G. and CAMPELLO, R. J. G. B. (2018). Clustering of RNA-Seq samples: Comparison study on cancer data. *Methods* **132** 42–49.
- JOHNSON, W. E., LI, C. and RABINOVIC, A. (2006). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8** 118–127.
- KLEIN, R. W. and DUBES, R. C. (1989). Experiments in projection and clustering by simulated annealing. *Pattern Recognition* **22** 213–220.
- KOBOLDT, D. C., FULTON, R. S., MCLELLAN, M. D., SCHMIDT, H., KALICKIVEIZER, J. and ET AL., J. F. M. (2012). Comprehensive molecular portraits of human breast tumours. *Nature* **490** 61–70.
- LEEK, J. T. (2014). svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Research* **42** e161–e161.
- LEEK, J. T., SCHARPF, R. B., BRAVO, H. C., SIMCHA, D., LANGMEAD, B., JOHNSON, W. E., GEMAN, D., BAGGERLY, K. and IRIZARRY, R. A. (2010). Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics* **11** 733–739.
- LI, B. and DEWEY, C. N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* **12**.
- LI, P., PIAO, Y., SHON, H. S. and RYU, K. H. (2015). Comparing the normalization methods for the differential analysis of Illumina high-throughput RNA-Seq data. *BMC Bioinformatics* **16**.
- LI, Q., NOEL-MACDONNELL, J. R., KOESTLER, D. C., GOODE, E. L. and FRIDLEY, B. L. (2018). Subject level clustering using a negative binomial model for small transcriptomic studies. *BMC Bioinformatics* **19**.
- LOVE, M. I., HUBER, W. and ANDERS, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **15**.
- MAO, J.-H., VAN DIEST, P. J., PEREZ-LOSADA, J. and SNIJDERS, A. M. (2017). Revisiting the impact of age and molecular subtype on overall survival after radiotherapy in breast cancer patients. *Scientific Reports* **7**.
- McLACHLAN, G. J. and KRISHNAN, T. (2008). *The EM Algorithm and Extensions, 2E*. John Wiley & Sons, Inc.
- MENG, X.-L. (1994). On the Rate of Convergence of the ECM Algorithm. *The Annals of Statistics* **22** 326–339.
- MENG, X.-L. and RUBIN, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* **80** 267–278.
- MO, Q. and SHEN, R. (2019). iClusterPlus: Integrative clustering of multi-type genomic data R package version 1.20.0.

- MO, Q., WANG, S., SESHAN, V. E., OLSHEN, A. B., SCHULTZ, N. and ET AL., C. S. (2013). Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proceedings of the National Academy of Sciences* **110** 4245–4250.
- MOOHTA, V. K., LINDGREN, C. M., ERIKSSON, K.-F., SUBRAMANIAN, A., SIHAG, S., LEHAR, J., PUIGSERVER, P., CARLSSON, E., RIDDERSTRÅLE, M., LAURILA, E., HOUSTIS, N., DALY, M. J., PATTERSON, N., MESIROV, J. P., GOLUB, T. R., TAMAYO, P., SPIEGELMAN, B., LANDER, E. S., HIRSCHHORN, J. N., ALTSHULER, D. and GROOP, L. C. (2003). PGC-1-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics* **34** 267–273.
- NEAL, R. and HINTON, G. E. (1998). A View Of The Em Algorithm That Justifies Incremental, Sparse, And Other Variants. In *Learning in Graphical Models* 355–368. Kluwer Academic Publishers.
- NOCEDAL, J. and WRIGHT, S. J. (2000). *Numerical Optimization*. Springer Science & Business Media.
- NOEL-MACDONNELL, J. R., USSET, J., GOODE, E. L. and FRIDLEY, B. L. (2018). Assessment of data transformations for model-based clustering of RNA-Seq data. *PLOS ONE* **13** e0191758.
- PAN, W. and SHEN, X. (2007). Penalized Model-Based Clustering with Application to Variable Selection. *J. Mach. Learn. Res.* **8** 1145–1164.
- PAN, W., SHEN, X. and LIU, B. (2013). Cluster Analysis: Unsupervised Learning via Supervised Learning with a Non-convex Penalty. *Journal of Machine Learning Research* **14**.
- PATRO, R., DUGGAL, G., LOVE, M. I., IRIZARRY, R. A. and KINGSFORD, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods* **14** 417–419.
- PEROU, C. M., SØRLIE, T., EISEN, M. B., VAN DE RIJN, M., JEFFREY, S. S. and ET AL., C. A. R. (2000). Molecular portraits of human breast tumours. *Nature* **406** 747–752.
- PICORNELL, A. C., ECHAVARRIA, I., ALVAREZ, E., LÓPEZ-TARRUELLA, S., JEREZ, Y., HOADLEY, K., PARKER, J. S., DEL MONTE-MILLÁN, M., RAMOS-MEDINA, R., GAYARRE, J., OCAÑA, I., CEBOLLERO, M., MASSARRAH, T., MORENO, F., SAENZ, J. A. G., MORENO, H. G., BALLESTEROS, A., BORREGO, M. R., PEROU, C. M. and MARTIN, M. (2019). Breast cancer PAM50 signature: correlation and concordance between RNA-Seq and digital multiplexed gene expression technologies in a triple negative breast cancer series. *BMC Genomics* **20**.
- PIEGORSCH, W. W. (1990). Maximum Likelihood Estimation for the Negative Binomial Dispersion Parameter. *Biometrics* **46** 863.
- RAJ-KUMAR, P.-K., LIU, J., HOOKE, J. A., KOVATICH, A. J., KVECHER, L., SHRIVER, C. D. and HU, H. (2019). PCA-PAM50 improves consistency between breast cancer intrinsic and clinical subtyping reclassifying a subset of luminal A tumors as luminal B. *Scientific Reports* **9**.
- REESE, S. E., ARCHER, K. J., THERNEAU, T. M., ATKINSON, E. J., VACHON, C. M., DE ANDRADE, M., KOCHER, J.-P. A. and ECKEL-PASSOW, J. E. (2013). A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. *Bioinformatics* **29** 2877–2883.
- REICHART, R. and RAPPOPORT, A. (2009). The NVI Clustering Evaluation Measure. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning. CoNLL '09* 165–173. Association for Computational Linguistics, Stroudsburg, PA, USA.
- REYNOLDS, A. P., RICHARDS, G. and RAYWARD-SMITH, V. J. (2004). The Application

- of K-Medoids and PAM to the Clustering of Rules. In *Intelligent Data Engineering and Automated Learning - IDEAL 2004, 5th International Conference* 173–178.
- ROBINSON, M. D., MCCARTHY, D. J. and SMYTH, G. K. (2009). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26** 139–140.
- ROSE, K. (1998). Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE* **86** 2210–2239.
- SCHWARZ, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics* **6** 461–464.
- SCRUCCA, L., FOP, M., MURPHY, T. B. and RAFTERY, A. E. (2016). mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R journal* **8** 289–317.
- SI, Y., LIU, P., LI, P. and BRUTNELL, T. P. (2013). Model-based clustering for RNA-seq data. *Bioinformatics* **30** 197–205.
- SMID, M., WANG, Y., ZHANG, Y., SIEUWERTS, A. M., YU, J., KLIJN, J. G. M., FOEKENS, J. A. and MARTENS, J. W. M. (2008). Subtypes of Breast Cancer Show Preferential Site of Relapse. *Cancer Research* **68** 3108–3114.
- STREHL, A. and GHOSH, J. (2003). Cluster Ensembles — a Knowledge Reuse Framework for Combining Multiple Partitions. *J. Mach. Learn. Res.* **3** 583–617.
- SUBRAMANIAN, A., TAMAYO, P., MOOTHA, V. K., MUKHERJEE, S., EBERT, B. L., GILLETTE, M. A., PAULOVICH, A., POMEROY, S. L., GOLUB, T. R., LANDER, E. S. and MESIROV, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* **102** 15545–15550.
- TIBSHIRANI, R., WALTHER, G. and HASTIE, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63** 411–423.
- TRITCHLER, D., PARKHOMENKO, E. and BEYENE, J. (2009). Filtering Genes for Cluster and Network Analysis. *BMC Bioinformatics* **10**.
- TSENG, P. (2001). Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization. *Journal of Optimization Theory and Applications* **109** 475–494.
- VAN LAARHOVEN, P. J. M. and AARTS, E. H. L. (1987). Simulated annealing. In *Simulated Annealing: Theory and Applications* 7–15. Springer Netherlands.
- VIDMAN, L., KÄLLBERG, D. and RYDÉN, P. (2019). Cluster analysis on high dimensional RNA-seq data with applications to cancer research - An evaluation study. *PLOS ONE* **14** e0219102.
- VINH, N. X., EPPS, J. and BAILEY, J. (2010). Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.* **11** 2837–2854.
- WU, T. T. and LANGE, K. (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics* **2** 224–244.
- WU, A. R., NEFF, N. F., KALISKY, T., DALERBA, P., TREUTLEIN, B., ROTHENBERG, M. E., MBURU, F. M., MANTALAS, G. L., SIM, S., CLARKE, M. F. and QUAKE, S. R. (2013). Quantitative assessment of single-cell RNA-sequencing methods. *Nature Methods* **11** 41–46.
- YANG, F., DING, P. and HUANG, R. (2015). Clinicopathological significance and potential drug target of CDH1 in breast cancer: a meta-analysis and literature review. *Drug Design, Development and Therapy* 5277.
- YANG, K., GAO, J. and LUO, M. (2019). Identification of key pathways and hub genes in basal-like breast cancer using bioinformatics analysis. *OncoTargets and Therapy* **Vol-**

ume **12** 1319–1331.

YANG, L., SHEN, Y., YUAN, X., ZHANG, J. and WEI, J. (2017). Analysis of breast cancer subtypes by AP-ISA biclustering. *BMC Bioinformatics* **18**.

YERSAL, O. (2014). Biological subtypes of breast cancer: Prognostic and therapeutic implications. *World Journal of Clinical Oncology* **5** 412.

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

ZWIENER, I., FRISCH, B. and BINDER, H. (2014). Transforming RNA-Seq Data to Improve the Performance of Prognostic Gene Signatures. *PLoS ONE* **9** e85150.

D. K. LIM

DEPARTMENT OF BIOSTATISTICS

UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

CHAPEL HILL, NC, 27599

USA

E-MAIL: deelim@live.unc.edu

N. U. RASHID

DEPARTMENT OF BIOSTATISTICS

UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

CHAPEL HILL, NC, 27599

USA

E-MAIL: naim@unc.edu

J. G. IBRAHIM

DEPARTMENT OF BIOSTATISTICS

UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

CHAPEL HILL, NC, 27599

USA

E-MAIL: ibrahim@bios.unc.edu