A SPARSE NEGATIVE BINOMIAL CLASSIFIER WITH COVARIATE ADJUSTMENT FOR RNA-SEQ DATA

BY TANBIN RAHMAN†, HSIN-EN HUANG‡, YUJIA LI†, AN-SHUN TAI‡, WEN-PING HSIEH‡, COLLEEN A MCCUNGR† AND GEORGE TSENGB∗†

University of Pittsburgh† and National Tsing Hua University‡

Supervised machine learning methods have been increasingly used in biomedical research and clinical practice. In transcriptomic applications, RNA-seq data have become dominating and have gradually replaced traditional microarray due to their reduced background noise and increased digital precision. Most existing machine learning methods are, however, designed for continuous intensities of microarray and are not suitable for RNA-seq count data. In this paper, we develop a negative binomial model via generalized linear model framework with double regularization for gene and covariate sparsity to accommodate three key elements: adequate modeling of count data with overdispersion, gene selection and adjustment for covariate effect. The proposed sparse negative binomial classifier (snbClass) is evaluated in simulations and two real applications of multi-disease post-mortem brain tissue RNA-seq data and cervical tumor miRNA-seq data to demonstrate its superior performance in prediction accuracy and feature selection.

1. Introduction. In the past two decades, microarray and RNA sequencing (RNA-seq) are routine procedures to study the transcriptome of organisms in modern biomedical studies. In recent years, RNA-seq (Wang, Gerstein and Snyder, 2009; Chu and Corey, 2012) has become a popular experimental approach for generating a comprehensive catalog of protein-coding genes and non-coding RNAs (Lorenz et al., 2014), and it largely replaces the microarray technology due to its low background noise and increased precision. The most important difference between RNA-seq and microarray technology is that RNA-seq outputs millions of sequencing reads rather than continuous fluorescent intensities in microarray data. Unlike microarray, RNA-seq can detect novel transcripts, gene fusions, single nucleotide variants, and indels (insertion/deletion). It can also detect a higher percentage of differentially expressed genes than microarray, especially for genes with low expression (Zhao et al., 2014).
In machine learning, classification methods are used to construct a prediction model based on a training dataset with known class labels so that the future independent samples can be classified with high accuracy. For example, labels in clinical research can be case/control, disease subtypes, drug response or prognostic outcome. Many popular machine learning methods have been widely applied to microarray studies, such as linear discriminant analysis (Dudoit, Fridlyand and Speed, 2002), support vector machines (Brown et al., 2000) and random forest (Díaz-Uriarte and De Andres, 2006). However, for discrete data nature in RNA-seq, many powerful tools for microarray assuming continuous data input or Gaussian assumption may be inappropriate. A common practice to solve this problem is to transform RNA-seq data into continuous values such as FPKM or TPM (Conesa et al., 2016) and possibly take additional log-transformation. However, such data transformation can lead to loss of information from the original data (Marioni et al., 2008; Robinson and Oshlack, 2010), producing less accurate inference. Particularly, the transformation often produces greater loss of information for genes with lower counts (McCarthy, Chen and Smyth, 2012). To accommodate discrete data in RNA-seq, Poisson distribution and negative binomial distribution are two common distributions expected to better fit the data generation process and data characteristics.

Witten (2011) proposed a sparse Poisson linear discriminant analysis (sPLDA) based on Poisson assumption for the count data. However, Poisson distribution assumes equal mean and variance, which is often not true. In real RNA-seq data, the variance is often larger than the mean, leading to the need of an overdispersion parameter. Witten (2011) reconciled this problem by proposing a power transformation to the data for eliminating overdispersion. However, as we will see later, the power transformation can perform well when the overdispersions are small but performs poorly when overdispersions of some genes are large. Hence, direct modeling by negative binomial assumption rather than a Poisson distribution is more appropriate. To this end, Dong et al. (2016) proposed negative binomial linear discriminant analysis (denoted as NBLDAPE) by adding a dispersion parameter. They, however, borrowed the “point estimation” from sPLDA in Witten (2011) and did not pursue a principled inference such as maximum likelihood, consequently producing worse performance than the method we will propose later.

Since the number of genes is often much larger than the number of samples in transcriptomic studies (a standard “small-n-large-p” problem), feature selection is critical to achieve better prediction accuracy and model interpretation. Witten (2011) proposed a somewhat heuristic soft-thresholding
operator, similar to univariate lasso estimator in regression, for gene selection in sPLDA but the method is not applicable to the NBLDA\textsubscript{PE} model due to the addition of dispersion parameter. In the NBLDA\textsubscript{PE} model proposed by Dong et al. (2016), the feature selection issue was not discussed, except that they used “edgeR” package to reduce the number of genes in the input data. Such a two-step filtering method is well-known to have inferior performance than methods with embedded feature selection. In fact, Zararsız et al. (2017) have compared sPLDA and NBLDA\textsubscript{PE}, and showed that the power transformed sPLDA generally performed better than NBLDA\textsubscript{PE} in their simulations and the worse performance in NBLDA\textsubscript{PE} mainly came from the lack of feature selection. Finally, another critical factor to consider in transcriptomic modeling is the adjustment of covariates such as gender, race and age since it is well-known that many genes are systematically impacted by these factors. For example, Peters et al. (2015) have identified 1,497 genes that are differentially expressed with age in a whole-blood gene expression meta-analysis of 14,983 individuals. A classification model allowing for covariate adjustment is expected to provide better accuracy and deeper biological insight.

To account for all aforementioned factors, we propose a sparse negative binomial model (snbClass) for classification analysis with covariate selection and adjustment. The method is based on generalized linear model (GLM) with a first regularization for feature sparsity. The GLM framework also allows straightforward covariate adjustment and a second regularization term on covariates, facilitating further covariate selection. Such covariate adjustment is not possible through existing sPLDA or NBLDA\textsubscript{PE} methods.

The paper is structured as follows. In Section 2.1, we briefly describe the two existing methods sPLDA (Witten, 2011) and NBLDA\textsubscript{PE} (Dong et al., 2016) followed by our proposed methods sNBLDA\textsubscript{GLM} and sNBLDA\textsubscript{GLM,sC} in Section 2.2. Section 2.3 and 2.4 discuss parameter estimation and tuning parameter selection procedures of the proposed methods. Benchmarks for evaluation are described in Section 2.5. Section 3 presents simulation studies and Section 4 shows two real applications using multi-disease post-mortem brain RNA-seq data and cervical tumor miRNA-seq data. Conclusions and discussions are included in Section 5. An R package “snbClass” is available at https://github.com/mdr56/snbclass to implement the proposed method.

2. Existing and proposed methods. In this section, we first describe two existing methods for classification analysis of count data from RNA-seq and then propose our new method. To unify the notation, we denote by $\mathbf{X}$ the count data matrix with elements $X_{ij}$ referring to the sequence count
for the $j$th gene and the $i$th sample ($i = 1, 2, \ldots n$ and $j = 1, 2, \ldots p$). In addition, $\mathbf{x}_i = (X_{i1} \ldots X_{ip})^T$ denotes the $i$th row of $\mathbf{X}$, corresponding to feature measurements of observation $i$. Also, define $X_{j} = \sum_{i=1}^{n} X_{ij}$, $X_\cdot = \sum_{i,j} X_{ij}$. Moreover, in the classification setting where each observation belongs to one of the $K$ classes, we let disjoint sets $C_k \subset \{1, \ldots, n\}$ contain the indices of observations in class $k$. That is, class label $y_i = k$ if and only if $i \in C_k$. Furthermore, we denote $X_{C_k,j} = \sum_{i \in C_k} X_{ij}$.

2.1. Two existing methods for classification of RNA-seq data.

2.1.1. Sparse Poisson linear discrimination analysis (sPLDA). Witten (2011) introduced a log-linear Poisson model with feature selection, which resulted in a simple diagonal linear discriminant analysis suitable for count data (referred as “sPLDA” hereafter in this paper). Under the assumption of gene independence, the model is based on the following formulation,

$$X_{ij}|y_i = k \sim \text{Poisson}(N_{ij} \cdot d_{kj}), \quad N_{ij} = s_i \cdot g_j,$$

where $y_i$ is the class label for the $i$th subject, $s_i$ is the normalizing factor (a.k.a. size factor) for sample $i$ and $g_j$ is the grand mean for the $j$th gene, allowing for variations both in samples and genes. For a given gene $j$, $d_{1j}, \ldots, d_{Kj}$ allows the $j$th gene to be differentially expressed between the classes if any of $d_{kj} \neq 1$ ($1 \leq k \leq K$).

RNA-seq data often contain over-dispersion such that variances are larger than means, whereas an important constraint in Poisson model is the equivalence of mean and variance. To overcome this, Witten (2011) proposed a power transformation of count data $X_{ij}^u \leftarrow X_{ij}^u$ with a proper choice of $u$ such that,

$$\sum_{i=1}^{n} \sum_{j=1}^{p} \frac{(X_{ij}^u - X_{ij}/X_{i.}^u)^2}{X_{i.}^u X_{j.}^u/\bar{X}_{..}^u} \approx (n - 1)(p - 1).$$

From simulations of the original paper, this correction performed well in the presence of weak to moderate overdispersion.

Suppose $\mathbf{x}^* = (X_1^*, \ldots, X_p^*)^T$ be a future new sample for prediction. The discriminant score for assigning $\mathbf{x}^*$ to class $k$ is,

$$\log p(y^* = k|\mathbf{x}^*) = \sum_{j=1}^{p} X_j^* \cdot \log \hat{d}_{kj} - s^* \cdot \left( \sum_{j=1}^{p} \hat{g}_j \cdot \hat{d}_{kj} \right) + \log \hat{\pi}_k + c'$$

where $y^*$ is the predicted label, $\hat{g}_j = X_{j.}$, $\hat{\pi}_k$ is the estimated population prior probability of belonging to the $k$th class estimated by the fraction of samples
belonging to class $k$ if the training data are representative to the underlying population and $s^*$ is the estimated normalization factor for the new sample $x^*$ for which we do not know the class label. The classifier assigns $x^*$ to the class with the largest discriminant score. The paper also implemented a somewhat heuristic soft-thresholding operator for feature selection in the classifier, which is motivated by univariate lasso regularization in regression for feature selection: $\hat{d}_{kj} = 1 + S(a/b - 1, v/\sqrt{b})$, where $a = X_{C_kj} + \kappa$, $b = \sum_{i \in C_k} \hat{N}_{ij} + \kappa$, $\kappa$ is the hyperparameter pre-determined in the estimation of $d_{kj}$ fixed at 1 in this paper, $v$ is the tuning parameter chosen by cross-validation and $S(x, a) = \text{sign}(x)(|x| - a)_+$ is the soft thresholding parameter. $\hat{d}_{1j} = \hat{d}_{2j} = \cdots = \hat{d}_{Kj} = 1$ means gene $j$ is not differentially expressed across the classes and thus, is not selected in the classifier.

2.1.2. Negative binomial linear discrimination analysis (NBLDA$_{PE}$). Dong et al. (2016) extended sPLDA into a negative binomial model to explicitly allow overdispersion property in RNA-seq data:

$$X_{ij}|y_i = k \sim \text{NB}(\mu_{ij} \cdot d_{kj}, \phi_j), \quad \mu_{ij} = s_i \cdot g_j$$

Under the formulation, $E(X_{ij}) = \mu_{ij}$ and $\text{Var}(X_{ij}) = \mu_{ij} + \mu_{ij}^2/\phi_j$. Similar to sPLDA, for a new observation $x^*$, prediction is made by the maximized discriminant score:

$$\log P(y^* = k|x^*) = \sum_{j=1}^p X^*_j \left[ \log \hat{d}_{kj} + \log \hat{g}_j - \log(\phi_j + s^* \hat{g}_j \hat{d}_{kj}) \right] - \sum_{j=1}^p \phi_j \log(\phi_j + s^* \hat{g}_j \hat{d}_{kj}) + \log \hat{\pi}_k + c',$$

where $\phi_j$ is the dispersion parameter for the $j$th gene, $\hat{d}_{kj} = (\sum_{i \in C_k} X_{ij} + 1)/(\sum_{i \in C_k} s_i X_{ij} + 1)$ and $\hat{g}_j$ is the same as defined previously. We note that the point estimate of $\hat{d}_{kj}$ and $\hat{g}_j$ are borrowed directly from Witten’s sPLDA model without theoretical justification and the similar soft-thresholding in sPLDA cannot be easily incorporated into the procedure due to the increased complexity with $\phi_j$.

In the literature, several popular procedures have been used for estimating the size factor, including simple sum of counts, median ratio (Anders and Huber, 2010) and quantile method (Bullard et al., 2010). Witten (2011) and Dong et al. (2016) showed that the performance is comparable among the three methods. Here, we will use the quantile method for all methods.
for a fair comparison. In the quantile method, the normalization factor for sample $i$ ($1 \leq i \leq n$) is estimated as $s_i = q_i / \sum_{i=1}^{n} q_i$ (or equivalently some papers also use $s_i = n \cdot q_i / \sum_{i=1}^{n} q_i$, which is what we adopt in this paper), where $q_i$ is the 75th quantile of sequence counts of all genes for the $i$th sample. For a new sample $x^*$, the normalizing factor is estimated as $s^* = n \cdot q^* / \sum_{i=1}^{n} q_i$, where $q_i$ ($1 \leq i \leq n$) comes from training data and $q^*$ is the 75th count quantile for sample $x^*$. Note that the vectors of normalization factors and dispersion are denoted by $s$ and $\phi$ respectively and will be pre-estimated in all negative binomial models in this paper before inference. $\phi$ is estimated by the weighted likelihood empirical Bayes method using the edgeR package (Robinson, McCarthy and Smyth, 2010) with class label considered. We denote the method proposed by Dong et al. (2016) as “NBLDAPE” to emphasize the heuristic “point estimation” procedure inherited from sPLDA in Witten (2011).

2.2. Proposed method: sparse negative binomial classifier via generalized linear model. We first consider a model without covariates in Section 2.2.1. Then we extend to incorporate covariates in Section 2.2.2.

2.2.1. Sparse negative binomial classifier without covariate adjustment (sNBLDA$\text{GLM}$). Similar to NBLDA$\text{PE}$, we specify the following negative binomial model in a generalized linear model (GLM) setting:

$$X_{ij} | y_i = k \sim NB(\mu_{ijk}, \phi_j); \quad \log(\mu_{ijk}) = \log(s_i) + \beta_{jk},$$

where $s_i$ is the normalization factor of the $i$th sample, $\beta_{jk}$ is the mean count in log-scale of the $k$th class for the $j$th gene and $\phi_j$ is the dispersion parameter of the $j$th gene. Under the assumption of independence between genes, the corresponding log-likelihood can be written as,

$$\log L(\Theta, \phi; x, y) = \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{p} I(y_i = k) \cdot \log f(X_{ij}; \beta_{jk}, \phi_j),$$

where, $\Theta = \{ (\beta_k, \phi); k = 1, \ldots, K \}$, $\beta_k = (\beta_{1k}, \ldots, \beta_{pk})$, $\phi = (\phi_1, \ldots, \phi_p)$. $I(y_i = k)$ is the indicator function taking value 1 if $y_i = k$ and 0 otherwise, and $f(X_{ij}; \beta_{jk}, \phi_j)$ is the density function of $\text{NB}(s_i \exp(\beta_{jk}), \phi_j)$. Now, suppose we have a new observation $x^*$ for which we intend to predict the class.
label. By Bayes theorem, we can derive the discriminant score as (2.1)

$$\log P(y^* = k|\mathbf{x}^*) = \log \pi_k - \sum_{j=1}^{p} \phi_j \log (\phi_j + s^* \exp(\hat{\beta}_{jk})) + \sum_{j=1}^{p} X_j^* [\hat{\beta}_{jk} - \log (\phi_j + s^* \exp(\hat{\beta}_{jk})) + e']$$

Here, $\mathbf{x}^*$ is assigned to class $k$ for which the discriminant score is maximized.

Note that the form of the discriminant score in the current model is identical to that proposed in Dong et al. (2016), except that we reparametrize $\mu_{ijk} = s_i g_j d_{kj}$ to $\log(\mu_{ijk}) = \log(s_i) + \beta_{jk}$. The major difference is in the parameter estimation. Dong et al. (2016) directly borrowed the point estimation of $\mu_{ijk}$ from the Poisson model in Witten (2011), while we derive MLE of Equation (2.2) (see below) using the iteratively reweighted least squares (IRLS) method to be shown in Section 2.3.1.

In order to incorporate variable (gene) selection, we add a penalty term $h(\beta) = \sum_{k=1}^{K} \sum_{j=1}^{p} |\beta_{jk} - \bar{\beta}_j|$. Here, $\bar{\beta}_j$ is the average of $\beta_{jk}$’s over the $K$ classes for a given $j$th gene. Hence, the following penalized likelihood is maximized to obtain estimation of $\beta$ with pre-estimated $\phi$:

$$\log L(\beta; \mathbf{x}, y, \phi) = \sum_{i=1}^{n} \sum_{k=1}^{K} \left[ I(y_i = k) \cdot \sum_{j=1}^{p} \log f(X_{ij}; \beta_{jk}, \phi_j) \right] - \lambda h(\beta)$$

Here, $\beta$ is the collection of all $\beta_{jk}$ parameters and $\lambda$ is a tuning parameter controlling sparsity of the variable selection. The form of the discriminant scores for prediction is the same as in Equation 2.1.

**2.2.2. Sparse negative binomial classifier with covariate adjustment (sNBLDA$_{GLM.C}$ and sNBLDA$_{GLM.sC}$).** In real applications, information of multiple clinical variables is often available and some of them may be associated with subsets of genes. Commonly encountered clinical variables can include age, gender, race, etc. Failure of covariate adjustment can greatly reduce prediction accuracy and replicability. In our GLM framework, covariate adjustment can be straightforwardly incorporated in the linear regression term:

$$X_{ij}|y_i = k \sim NB(\mu_{ijk}, \phi_j); \log(\mu_{ijk}) = \log(s_i) + \beta_{jk} + \sum_{q=1}^{Q} \alpha_{qj} z_{iq},$$

Here, $z_q = (Z_{1q}, \ldots, Z_{nq})$ includes values of the $q$th covariate over $n$ samples and parameter $\alpha_{qj}$ corresponds to the coefficient of the $q$th covariate in the $j$th gene. Under the assumption of gene independence and adding penalty
terms for both genes and covariates, the problem can be presented as maximization of the following penalized log-likelihood with double regularization:

\[
\log L(\beta, \alpha; y, x, z_1, \ldots, z_Q, \phi) = \sum_{i=1}^{n} \sum_{k=1}^{K} I(y_i = k) \sum_{j=1}^{p} \log f(X_{ij}, Z_{i1}, \ldots, Z_{iQ}; \beta_{jk}, \alpha_{qj}, \phi_j) \\
- \lambda_1 h(\beta) - \lambda_2 \sum_{q=1}^{Q} \sum_{j=1}^{p} |\alpha_{qj}| ,
\]

(2.4)

where, \( \beta \) is the collection of all \( \beta_{jk} \) parameters and \( \alpha \) is the collection of all \( \alpha_{qj} \) parameters. \( \lambda_1 \) and \( \lambda_2 \) are tuning parameters controlling for levels of sparsity of variable selection in genes and covariates, respectively.

Similarly, for a new sample \( x^* \) with vector of clinical vectors \( z^* = (z^*_1, \ldots, z^*_Q) \) under this framework, we can derive the following discriminant score:

\[
\log P(y^* = k|x^*) = \log \hat{\pi}_k - \sum_{j=1}^{p} \phi_j \log[\phi_j + s^* \exp(\hat{\beta}_{jk} + \sum_{q=1}^{Q} z^*_q \hat{\alpha}_{qj})] \\
+ \sum_{j=1}^{p} X^*_j [\hat{\beta}_{jk} + \sum_{q=1}^{Q} z^*_q \hat{\alpha}_{qj} - \log(\phi_j + s^* \exp(\hat{\beta}_{jk} + \sum_{q=1}^{Q} z^*_q \hat{\alpha}_{qj}))] + c'
\]

(2.5)

As before, \( x^* \) is assigned to the class with the highest discriminant score.

We note that when \( \lambda_2 = 0 \), Equation 2.4 performs covariate adjustment using all covariates for all genes without regularization in covariate parameters \( \alpha_{qj} \). We will denote this method as “sNBLDA\_GLM.C”. In this case, when the number of covariates \( Q \) becomes large, performance of parameter estimation and prediction accuracy are expected to decline. With proper choice of \( \lambda_2 \) in Equation (2.4), the method can adequately select a subset of covariates in each gene to improve the performance. For illustration purposes, we refer to this method as “sNBLDA\_GLM.sC” in this paper, where “sC” means sparsity on covariates. This is the method we recommend in general applications when clinical covariates are available and will be referred to as “snbClass” in the R package and future applications. When clinical covariates do not exist, the method naturally reduces to “sNBLDA\_GLM”.

Remark: (distinction of estimation procedure of sNBLDA\_GLM compared to sPLDA) The soft-thresholding operator adopted in sPLDA’s estimation procedure (i.e. \( \hat{d}_{kj} \) in Section 2.1.1) borrows the idea from the nearest shrunken centroids (NSC) classifier (Tibshirani et al., 2002), where the class-specific
mean vectors of each gene under diagonal LDA are “shrunken” towards global means. Theoretically, however, this lasso “shrunken” soft-thresholding operator is derived from least square error minimization in Gaussian log-likelihood with lasso penalty. Directly plugging the lasso “shrunken” soft-thresholding operator to the Poisson log-likelihood is a convenient solution but is not a rigorous procedure. In other words, the sPLDA estimation procedure does not correspond to optimization of a regularized target function. In contrast, sNBLDA\textsubscript{GLM} optimizes the negative binomial log-likelihood with lasso regularization. The soft-thresholding formula in the iterative updates in Section 2.3.1 and 2.3.2 are consequences from the target function optimization and is fundamentally different from the soft-thresholding procedure of sPLDA in Section 2.1.1. Consequently, sNBLDA\textsubscript{GLM} can benefit from the GLM framework and naturally adjust for confounding covariates such as age, race or other clinical variables. On the other hand, the heuristic estimation and feature selection procedure in sPLDA prohibits its extension for this purpose.

2.3. \textit{Estimation in sNBLDA\textsubscript{GLM} and sNBLDA\textsubscript{GLM.sC}.}

2.3.1. \textit{Estimation of sNBLDA\textsubscript{GLM}.} Maximizing the log-likelihood derived in Equation (2.2) is equivalent to minimizing the following penalized weighted least square function,

\begin{equation}
\sum_{i=1}^{n} \sum_{k=1}^{K} I(y_i = k) \sum_{j=1}^{p} w_{ijk}(\tau_{ijk} - \log(s_i) - \beta_{jk})^2 + \lambda \sum_{j=1}^{p} \sum_{k=1}^{K} |\beta_{jk} - \bar{\beta}_j|,
\end{equation}

where \(w_{ijk} = \mu_{ijk}/(1 + \phi_{j}^{-1}\mu_{ijk})\) and \(\tau_{ijk} = \log(s_i) + \beta_{jk} + (x_{ij} - \mu_{ijk})/\mu_{ijk}\). Given the estimates at the \(t\)-th step, the updates of the \((t+1)\)-th step is:

1. Calculate \(w_{ijk}^{(t+1)} = \mu_{ijk}^{(t)}/(1 + \phi_{j}^{-1}\mu_{ijk}^{(t)})\)
2. Update \(\tau_{ijk}^{(t+1)} = \log(s_i) + \beta_{jk}^{(t)} + (x_{ij} - \mu_{ijk}^{(t)})/\mu_{ijk}^{(t)}\)
3. Solve \(\beta_{jk}^{(t+1)} = \arg\min \frac{1}{2} \sum_i I(y_i = k)w_{ijk}^{(t+1)}(\tau_{ijk}^{(t+1)} - \log(s_i) - \beta_{jk})^2 + \lambda|\beta_{jk} - \bar{\beta}_j|\)
4. Update \(\mu_{ijk}^{(t+1)} = \exp(\beta_{jk}^{(t+1)} + \log(s_i))\)

This is repeated until convergence of \(\hat{\beta}_{jk}\). The update of \(\hat{\beta}_{jk}\) in Step (3) is given by,
\[
\beta_{jk}^{(t+1)} = \tilde{\beta}_{j}^{(t+1)} + \text{sign}(\tilde{\beta}_{jk}^{(t+1)} - \bar{\beta}_{j}^{(t+1)}) \\
\left[ \sum_{i \in C_{k}} w_{ijk}^{(t+1)}(\tau_{ijk}^{(t+1)} - \log(s_{i})) - \lambda(1 - 1/K) \text{sign}(\tilde{\beta}_{jk}^{(t+1)} - \tilde{\bar{\beta}}_{j}^{(t+1)}) \right] - |\tilde{\beta}_{j}^{(t+1)}|
\]

Here, \( [\cdot]_{(+)} \) is the soft thresholding function such that \( [u]_{(+)} \) takes the value \( u \) when \( u \) is positive and 0 otherwise, \( \tilde{\beta}_{jk}^{(t+1)} \) is the estimate of \( \beta_{jk} \) under no penalization and \( \tilde{\bar{\beta}}_{j}^{(t+1)} = \sum_{k=1}^{K} \tilde{\beta}_{jk}^{(t+1)}/K \).

2.3.2. Estimation of sNBLDA\textsubscript{GLM,sC}. Similar to sNBLDA\textsubscript{GLM}, the problem of maximizing the penalized log-likelihood in Equation (2.4) can be represented as minimizing the penalized weighted least square function given below in Equation (2.7),

\[
\sum_{i=1}^{n} \sum_{k=1}^{K} \left[ I(y_{i} = k) \sum_{j=1}^{p} w_{ijk}(\tau_{ijk} - \log(s_{i}) - \beta_{jk} - \sum_{q=1}^{Q} z_{iq}\alpha_{qj})^{2} \right] \\
+ \lambda_{1} \sum_{j=1}^{p} \sum_{k=1}^{K} |\beta_{jk} - \tilde{\beta}_{j}| + \lambda_{2} \sum_{q=1}^{Q} \sum_{j=1}^{p} |\alpha_{qj}|
\]

where, \( w_{ijk} = \mu_{ijk}/(1 + \phi_{j}^{-1}\mu_{ijk}) \) and \( \tau_{ijk} = \log(s_{i}) + \beta_{jk} + \sum_{q=1}^{Q} z_{iq}\alpha_{qj} + (x_{ij} - \mu_{ijk})/\mu_{ijk} \). The estimation of each of the \( \beta_{jk} \) and \( \alpha_{qj} \) is given by the following algorithm. The steps involved in IRLS given the estimates obtained at the \( t \)th step is given below:

1. Calculate \( w_{ijk}^{(t+1)} = \mu_{ijk}^{(t)}/(1 + \phi_{j}^{-1}\mu_{ijk}^{(t)}) \)
2. Update \( \tau_{ijk}^{(t+1)} = \log(s_{i}) + \beta_{jk}^{(t)} + \sum_{q=1}^{Q} z_{iq}\alpha_{qj}^{(t)} + (x_{ij} - \mu_{ijk}^{(t)})/\mu_{ijk}^{(t)} \)
3. Solve \( \beta_{jk}^{(t+1)} = \arg \min \frac{1}{2} \sum_{i} I(y_{i} = k) w_{ijk}^{(t+1)}(\tau_{ijk}^{(t+1)} - \log(s_{i}) - \beta_{jk} - \sum_{q=1}^{Q} z_{iq}\alpha_{qj}^{(t)})^{2} + \lambda_{1} |\beta_{jk} - \tilde{\beta}_{j}| + \lambda_{2} \sum_{q=1}^{Q} \sum_{j=1}^{p} |\alpha_{qj}| \)
4. Solve \( \alpha_{qj}^{(t+1)} = \arg \min \frac{1}{2} \sum_{i} \sum_{k=1}^{K} I(y_{i} = k) w_{ijk}^{(t+1)}(\tau_{ijk}^{(t+1)} - \log(s_{i}) - \beta_{jk}^{(t+1)} - \sum_{q=1}^{Q} z_{iq}\alpha_{qj})^{2} + \lambda_{1} |\beta_{jk}^{(t+1)} - \tilde{\beta}_{j}^{(t+1)}| + \lambda_{2} \sum_{q=1}^{Q} \sum_{j=1}^{p} |\alpha_{qj}| \), where \( \tilde{\beta}_{j}^{(t+1)} = \sum_{k=1}^{K} \beta_{jk}^{(t+1)}/K \)
5. Update $\mu_{ijk}^{(t+1)} = \exp(\beta_{jk}^{(t+1)} + \sum_{q=1}^{Q} z_{iq} \alpha_{qj}^{(t+1)} + \log(s_i))$

The steps are repeated until convergence of the parameters $\beta_{jk}$ and $\alpha_{qj}$. The optimizations in step 3 and step 4 are respectively given by,

$$\beta_{jk}^{(t+1)} = \tilde{\beta}_{jk}^{(t+1)} + \text{sign}(\tilde{\beta}_{jk}^{(t+1)} - \tilde{\beta}_{jk}^{(t+1)})$$

$$\frac{\sum_{i \in C_k} w_{ijk}^{(t+1)} (\tau_{ijk} - \log(s_i) - \sum_{q=1}^{Q} \alpha_{qj}^{(t)} z_{qj}) - \lambda_1 (1 - 1/K) \text{sign}(\tilde{\beta}_{jk}^{(t+1)} - \tilde{\beta}_{jk}^{(t+1)})}{\sum_{i \in C_k} w_{ijk}^{(t+1)}}$$

and,

$$\alpha_{qj}^{(t+1)} = \text{sign}(\tilde{\alpha}_{qj}) \left[|\tilde{\alpha}_{qj}| - \frac{\lambda_2}{\sum_{i=1}^{n} \sum_{k=1}^{K} I(y_i = k) w_{ijk} z_{iq}^2}\right]$$

where,

$$\tilde{\alpha}_{qj} = \sum_{i=1}^{n} \sum_{k=1}^{K} I(y_i = k) w_{ijk}^{(t+1)} (\tau_{ijk} - \log(s_i)) - \beta_{jk}^{(t+1)} - \sum_{1 \leq m \leq Q, m \neq q} z_{im}^{(t+1)} \sum_{i=1}^{n} \sum_{k=1}^{K} I(y_i = k) z_{iq}^2 w_{ijk}^{(t+1)}$$

2.4. Selection of tuning parameters in regularization. Both sNBLDA$_{\text{GLM}}$ and sNBLDA$_{\text{GLM,sC}}$ methods involve selection of regularization parameters $\lambda$ or $(\lambda_1, \lambda_2)$. We apply V-fold cross-validation as a tool to determine the tuning parameter (Stone, 1974). For each given tuning parameter, we divide the dataset into $V$ equal folds and samples in the $K$ classes are split into $V$ folds as even as possible. In each iteration, one fold is set aside as the test set and the remaining ($V-1$) folds are used as the training set. The classifier is built from the training set and then validated in the test set for evaluating accuracy. This procedure is repeated until all $V$ folds have been chosen as the test set and the averaged accuracy is calculated. The tuning parameter corresponding to the highest averaged accuracy is chosen for the final model construction. We apply 10-fold ($V=10$) cross-validation for all simulations and real applications in this paper.

2.5. Benchmarks for evaluation. Performance of different methods will be judged by two major criteria: accuracy of prediction and accuracy of feature selection. For prediction performance, simple averaged accuracy is used when true class labels are known: Accuracy = \frac{\text{Number of test samples correctly classified}}{\text{Number of test samples}}.
For feature selection performance, we derive the area under the curve (AUC) (Bradley, 1997) values of the receiver operating characteristic (ROC) curves. For the simulated data, we calculated the AUC by varying the sparsity parameter in the training dataset. In real data, we use 10-fold cross-validation to compare the different methods. However for the sNBLDA_{GLM,sC}, we have two tuning parameters, $\lambda_1$ (for gene selection) and $\lambda_2$ (for covariate selection). For each $\lambda_1$, we choose the optimal $\lambda_2$ based on the best cross-validation accuracy.

3. Simulations. In this section, we will devise two simulation schemes to compare the performance of sPLDA and NBLDA_{PE} to our proposed model sNBLDA_{GLM} and sNBLDA_{GLM,sC} under different settings. In Simulation 1, there is no covariate effect over the expression levels of the genes. Here, we compare sPLDA, NBLDA_{PE} and sNBLDA_{GLM} over different levels of signal strength under different levels of dispersion in the data. In Simulation 2, we develop a simulation scheme where two covariates are introduced which can affect expression level of certain proportion of the genes. Here, we compare sPLDA, NBLDA_{PE}, sNBLDA_{GLM} and sNBLDA_{GLM,sC} in the presence of covariate effects. The first two simulations follow gene independence assumption. We finally extend both simulation schemes and incorporate gene correlation structure to conduct a sensitivity analysis.

In order to mimic real data, we investigate three published RNA-seq datasets and extract their gene-specific mean (i.e. $\bar{\beta}_j$) and dispersion (i.e. $\phi_j$) parameters to conduct the simulations. The three RNA-seq datasets include: (1) (rat brain) The GSE47474 RNA-seq dataset from Gene Expression Omnibus (GEO) contains 72 samples with 36 from HIV-1 transgenic and 36 from control rat strains (Li et al., 2013). (2) (cervical miRNA) The miRNA-seq dataset contains 29 controls and 29 cervical tumors measuring 714 miRNAs. This dataset will be evaluated in Section 4.2. (3) (Multi-disease post-mortem brain RNA-seq) This RNA-seq dataset contains 263 schizophrenia, 279 controls and 47 bipolar subjects which will be evaluated in Section 4.1. Figure S1 shows the genome-wide distribution of baseline means and dispersions for each dataset estimated by edgeR. The result shows generally greater overdispersion in the cervical miRNA dataset but smaller overdispersion for rat brain and schizophrenia brain studies. Table S1 shows the scale and shape parameters when the estimated dispersions are fitted to Pareto distribution family. To provide a better spectrum of simulation analyses, we first design Simulation Type A with baseline means $\hat{\beta}_j$ (denoted as $b_j$ in the simulation formulation below) sampled from the estimated empirical distribution in the rat brain study and then sample dispersions $1/\hat{\phi}_j \sim \text{Pareto}(\psi, \nu)$
where $\psi = \{5, 8\}$ and $\nu = \{0.1, 5, 10\}$. Here, $\psi$ refers to shape and $\nu$ refers to scale parameter. The values 0.1, 5, 10 for $\nu$ represent small, moderate and high level of dispersion respectively. In Simulation Type B, we directly sample paired $(\beta_j, \phi_j)$, $1 \leq j \leq p$, from the estimated empirical distributions of the three studies. Since different RNA-seq datasets may have different levels of overdispersion, we will only present simulation settings and results of Simulation Type A (1A and 2A) in the manuscript below. The result of Simulation Type B (1B and 2B) are shown in Supplement Material (Figure S7 and S8). Each simulation is repeated 100 times and the average result is reported.

3.1. Simulation settings.

Simulation 1A: Without covariate effect

In this simulation, we sample the count data by

$$X_{ij} | y_i = k \sim NB(s_i b_j \exp(\delta_{jk} \Delta_j), \phi_j)$$

for each gene $j (1 \leq j \leq 1000)$ and sample $i (1 \leq i \leq 1200)$ in class $k (1 \leq k \leq 3)$, where the number of informative features is 300. The notation of the parameters as well as the settings are given below:

- The library size factor $s_i$ is sampled from Unif(0.75, 1.25) for each sample $i$.
- $b_j$ is the baseline which is sampled from the empirical distribution of the mean expression described previously.
- $\delta_{jk}$ represents the pattern of gene $j$ in class $k$. For all $\delta_{jk} \in \{-1, 0, 1\}$, 1 indicating an up-regulated trend of genes in this class relative to other classes, -1 indicating down-regulation and 0 indicating no difference. There exists three gene patterns for the 300 informative genes: $(\delta_{j1}, \delta_{j2}, \delta_{j3}) = (1,0,-1), (0,1,1)$ and $(-1,-1,0)$. For non-informative genes, the pattern is $(0,0,0)$.
- Sample the main effect size parameter $\Delta_j$ for each gene $j$ from a truncated normal distribution $TN(\zeta, 0.1^2, \zeta/2, \infty)$, where $\zeta$ is the mean, standard deviation is set to 0.1 and values smaller than $\zeta/2$ are truncated.
- $1/\phi_j \sim Pareto(\psi, \nu)$ where $\psi = \{5, 8\}$ and $\nu = \{0.1, 5, 10\}$. Here, $\psi$ refers to shape and $\nu$ refers to scale parameter. The values 0.1, 5, 10 for $\nu$ represent small, moderate and high level of dispersion respectively.
- 200 of the samples are used as the training set and the remaining 1,000 samples are used as the testing set.

Simulation 2A: Incorporating covariate effect

We sample the count data by

$$X_{ij} | y_i = k \sim NB(s_i b_j \exp(\delta_{jk} \Delta_j + \sum_{q=1}^{2} \gamma_{qj} \epsilon_{qj} z_{qi})), \phi_j)$$

for each gene $j (1 \leq j \leq 1000)$ and sample $i (1 \leq i \leq 1200)$ in class
\(k(1 \leq k \leq 3)\) with two covariates \((z_1 \text{ and } z_2; \ Q=2)\), where the number of informative features is 300. The notation of parameters are as follows:

- We generate a binary covariate (e.g. gender) for each sample \(i\) from \(Ber(0.5)\) (i.e. \(z_1 \sim Ber(0.5)\)) and generate a continuous covariate (e.g. age) for each sample \(i\) from Gamma(5, 10)
- \(\gamma_{qj}\) represents the pattern of gene \(j\) in covariate \(q\) for all \(\gamma_{qj} \in \{0, 1\}\); there exists four patterns: \((\gamma_{1j}, \gamma_{2j}) = (1, 1), (1, 0), (0, 1), \text{ and } (0, 0)\) with probability \((\rho/3, \rho/3, \rho/3 \text{ and } 1-\rho)\) respectively. When \(\rho = 0\), none of the genes are not impacted by covariates. We choose the proportion of covariate-impacted genes \(\rho\) to be 0.125, 0.25 and 0.5.
- Sample the main effect size parameter \(\Delta_j\) for each gene \(j\) in class \(k\) from a truncated normal distribution \(TN(0.25, 0.1^2, 0.125, \infty)\)
- The effect size parameter of covariates \(\epsilon_{qj}\) for each gene \(j\) in covariate \(q\) is drawn from a truncated normal distribution \(TN(\eta, 0.1^2, \eta/2, \infty) \times \omega\) where \(\omega\) has equal probability of taking 1 or -1. We use the different values of \(\eta \in \{0.1, 0.3, 0.5, 0.7\}\) for different levels of signal strength of the covariates.
- Other parameters are set the same as Simulation 1 except that \(\zeta\) is set at 0.25.
- 200 of the samples are used as training set and the remaining 1,000 samples are used as testing set.

The above two simulation settings assume gene independence. To evaluate the impact of gene dependence, we extend Simulation 1A and 2A by incorporating gene-gene dependence structure and perform sensitivity analysis. The detailed simulation scheme is described in Appendix A.

### 3.2. Simulation results.

Results of Simulation 1A are summarized in Figure 1. In Figure 1(a), average prediction accuracies of the three models (sPLDA, NBLDA\textsubscript{PE} and sNBLDA\textsubscript{GLM}) are compared under six different levels of dispersions controlled by the parameters \(\psi = \{5, 8\}\) and \(\nu \in \{0.1, 5, 10\}\). As previously described, the choice of Pareto distribution family and parameters in this simulation are guided by three real datasets. The larger the value of \(\nu\), the larger the level of dispersion in the simulated datasets. In all different levels of \(\zeta\) and \(\nu\), sNBLDA\textsubscript{GLM} outperforms the other two methods with large margins in prediction accuracy (Figure 1(a)). As expected, NBLDA\textsubscript{PE} is superior to sPLDA when \(\nu\) is large (i.e. larger overdispersion). In our simulation, sPLDA has inferior prediction performance even when the scale parameter is small. Figure 1(b) shows results of variable selection by AUC. sNBLDA\textsubscript{GLM} also outperforms sPLDA in all cases while NBLDA\textsubscript{PE} cannot perform variable selection and is not appli-
Fig 1: Results for Simulation 1A without covariate effect. The vertical bar line represents the standard error.
cable in this plot. We further compare sNBLDA_{GLM} with three popular machine learning methods: support vector machines (SVM), random forest (RF) and classification and regression tree (CART). Figure S2 shows inferior performance of these classifiers due to ignorance of count data and transformation to continuous inputs. In this simulation setting as well as for the rest of the paper where we have used SVM, RF and CART, we transform the data to continuous values by using variance stabilizing transformation (VST) method in the DESeq2 package.

Figure 2 demonstrates results of Simulation 2A using sPLDA, NBLDA_{PE}, sNBLDA_{GLM} (no covariate adjustment) and sNBLDA_{GLM,sC} (with covariate adjustment and regularization) with varying percent of genes impacted by covariates $\rho = 0.125, 0.25$ and $0.5$. Figure 2(a) shows average prediction accuracy of varying $\eta$ and level of dispersion controlled by parameters $\psi = 5$ and $\nu = \{0.1, 5, 10\}$ as in Simulation 1 (result of $\psi = 8$ is shown in Figure S3). When $\nu = 10$ (high level of dispersion), sNBLDA_{GLM,sC} largely outperforms all other methods as the impact of covariates on gene expression $\eta$ increases. The prediction accuracy for sNBLDA_{GLM,sC} remains high with increased $\eta$ due to its capacity of adjusting for covariate effect, while prediction accuracies of the other three methods drop with increased $\eta$. sNBLDA_{GLM} still outperforms sPLDA and NBLDA_{PE}. When $\nu = 0.1$, a similar pattern is observed. The margin between sNBLDA_{GLM,sC} and sNBLDA_{GLM} becomes much smaller but sNBLDA_{GLM,sC} is still the best performer. Figure S4 includes comparison with SVM, RF and CART, all of which perform worse than sNBLDA_{GLM,sC}. Variable selection performance between sPLDA, sNBLDA_{GLM} and sNBLDA_{GLM,sC} is shown in Figure 2(b). Here we observe that when the covariate effect and proportion of genes influenced by covariates are small, sNBLDA_{GLM} has the best variable selection performance followed by sPLDA. However, once the proportion of covariates $\rho$ or the effect size $\eta$ increases, sNBLDA_{GLM,sC} outperforms the other methods.

Figure S5 shows sensitivity analysis result with gene-gene dependence structure in Simulation scheme 1A. Here we see that sNBLDA_{GLM} outperforms the other two methods in terms of accuracy. It also shows slightly superior variable selection compared to sPLDA under moderate to large level of dispersion. Similarly, the performance under gene dependence structure in Simulation 2A is given in Figure S6 ($\psi = 5$). sNBLDA_{GLM,sC} outperforms all the other methods in terms of prediction accuracy. However, for low level of dispersion ($\nu = 0.1$), the variable selection of sPLDA slightly outperforms sNBLDA_{GLM} and sNBLDA_{GLM,sC}. With increasing covariate effect, sNBLDA_{GLM,sC} outperforms the other two methods. The results for
Fig 2: Results for Simulation 2A with covariate effect when $\psi = 5$. The vertical bar represents standard error.
Simulation 1B and 2B is given in Figures S7 and S8 respectively showing similar trends as in Simulation 1A and 2A.

4. Real applications.

4.1. Multi-disease post-mortem brain RNA-seq dataset. This RNA-seq dataset (http://www.synapse.org/CMC) is obtained from the CommonMind Consortium (Fromer et al., 2016) using post-mortem human dorsolateral prefrontal cortex tissues from 263 schizophrenia, 47 bipolar and 285 control subjects. Three clinical variables are available: age of death, PMI and ethnicity (Caucasian, Asian, Hispanic, Multiracial or African American). We exclude 55 subjects with an age of death over 90 and remove subjects with multiracial (1 subject), Asian (4 subjects) and Hispanic (22 subjects) ethnicity. After removing these samples, we are left with 241 schizophrenia, 227 control and 45 bipolar subjects. We then perform routine data preprocessing and filtering to keep genes with at least 70% of the samples having gene expression counts greater than 0 and mean count across the samples greater than 10, producing a count data matrix with 17,962 genes for machine learning. Even though three of the four methods have embedded feature selection capability, the feature selection is usually difficult for ultra-high dimensionality. Hence, we further reduce the number of genes to the top 1,000 genes based on standard deviation (i.e., filter low-variance genes) of log(CPM) values.

We compare the models under four different scenarios. In scenario I, we keep all the 513 subjects with three classes corresponding to schizophrenia (SZ), bipolar (BP) and control subjects (Ctrl). For scenario II, we apply the methods to classify schizophrenia and control (SZ vs Ctrl) subjects, while scenarios III and IV are to predict bipolar versus control (BP vs Ctrl) and schizophrenia versus bipolar (SZ vs BP), respectively. In order to compare the prediction accuracy for the models, we compare the 10-fold cross-validation accuracy with a varying number of selected genes obtained by varying the tuning parameters for the corresponding models. Next, we perform pathway enrichment analysis by using Fisher’s exact test based on the Gene Ontology (GO), KEGG and Reactome pathway databases to evaluate the variable selection performance for each of the model under consideration. For pathway analysis, a subset of genes for each method are selected by varying the tuning parameter. This is only done for sPLDA, sNBLDA_{GLM} and sNBLDA_{GLM,sC} since there is no variable selection procedure proposed for the NBLDA_{PE} model.
SPARSE NEGATIVE BINOMIAL MODEL FOR RNA-SEQ

Fig 3: Number of pathways enriched (y-axis) in sPLDA, sNBLDA_{GLM} and sNBLDA_{GLM,sC} with varying number of selected genes (x-axis) in the multi-disease post-mortem brain RNA-seq data. Fig 5(A)-5(D) are results for scenarios I-IV, respectively.
Table 1
Cross-validation accuracy and the corresponding number of selected genes by each method under the four different scenarios. I: SZ vs BP vs Ctrl; II: SZ vs Ctrl; III: BP vs Ctrl; IV: SZ vs BP.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Model</th>
<th>Number of selected genes</th>
<th>Cross-validation accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>sPLDA</td>
<td>458.30</td>
<td>0.466</td>
</tr>
<tr>
<td></td>
<td>NBLDA_PPE</td>
<td>1000.00</td>
<td>0.475</td>
</tr>
<tr>
<td></td>
<td>sNBLDA_GLM</td>
<td>900.40</td>
<td>0.501</td>
</tr>
<tr>
<td></td>
<td>sNBLDA_GLM_sc</td>
<td>131.20</td>
<td>0.571</td>
</tr>
<tr>
<td>II</td>
<td>sPLDA</td>
<td>930.20</td>
<td>0.611</td>
</tr>
<tr>
<td></td>
<td>NBLDA_PPE</td>
<td>1000.00</td>
<td>0.599</td>
</tr>
<tr>
<td></td>
<td>sNBLDA_GLM</td>
<td>139.50</td>
<td>0.618</td>
</tr>
<tr>
<td></td>
<td>sNBLDA_GLM_sc</td>
<td>723.60</td>
<td>0.643</td>
</tr>
<tr>
<td>III</td>
<td>sPLDA</td>
<td>526.80</td>
<td>0.697</td>
</tr>
<tr>
<td></td>
<td>NBLDA_PPE</td>
<td>1000.00</td>
<td>0.699</td>
</tr>
<tr>
<td></td>
<td>sNBLDA_GLM</td>
<td>448.10</td>
<td>0.695</td>
</tr>
<tr>
<td></td>
<td>sNBLDA_GLM_sc</td>
<td>41.20</td>
<td>0.792</td>
</tr>
<tr>
<td>IV</td>
<td>sPLDA</td>
<td>184.50</td>
<td>0.755</td>
</tr>
<tr>
<td></td>
<td>NBLDA_PPE</td>
<td>1000.00</td>
<td>0.750</td>
</tr>
<tr>
<td></td>
<td>sNBLDA_GLM</td>
<td>971.90</td>
<td>0.809</td>
</tr>
<tr>
<td></td>
<td>sNBLDA_GLM_sc</td>
<td>952.20</td>
<td>0.859</td>
</tr>
</tbody>
</table>

Figure S9 shows the 10-fold cross-validation accuracies for the four models under different scenarios with varying number of selected genes. In all four scenarios, we can see slight improvement of sNBLDA_GLM over SPLDA and NBLDA_PPE, showing advantage of a rigorous negative binomial modeling. In addition, sNBLDA_GLM_sc shows a large (5-10% improvement) prediction improvement over all three other methods, which argues importance of covariate adjustment in the model. Table 1 summarizes the best cross-validation prediction accuracy and the corresponding number of selected genes for each method in each scenario. In the three-class prediction in scenario 1, for example, sNBLDA_GLM improves over SPLDA and NBLDA_PPE (accuracy=50.1% versus 46.6% and 47.5%) and sNBLDA_GLM_sc has the highest accuracy 57.1%. For scenario 3, we observe the most striking improvement with sNBLDA_GLM_sc achieving 79.2% accuracy while all the other three methods have ~ 70% accuracy. To assess statistical significance of the improvement and avoid potential overfitting from selecting the best prediction accuracy across different choices of number of selected genes, we randomly split the data into two equal halves in each class. We use half of the training samples to build the predictive model and then obtain the prediction accuracy by using the other half testing samples. We then compare the testing set prediction accuracy of sNBLDA_GLM_sc with each of the other three methods using two-sample proportion test. We repeat the random data splitting three
times and the result of test sample prediction accuracy and proportion test p-values are shown in Table S3. We observe that the prediction accuracy for sNBLDA_{GLM,sC} is always greater than the other methods for all repeats and scenarios. For scenario I, III and IV, we find that the performance of sNBLDA_{GLM,sC} is almost always significantly better than the other models for all the repeats at $\alpha = 0.05$ level of significance (see boldfaced p-values in Table S3). For scenario II, the statistical significance of the improvement is not that clear.

Next, we compare the methods in terms of pathway analysis to assess the gene selection performance by functional annotation. Here, the number of pathways enriched at p-value cutoff of 0.01 is selected and the result is given in Figure 3. In all four scenarios, we have found sNBLDA_{GLM,sC} to detect the highest number of enriched pathways, followed by sNBLDA_{GLM}, and sPLDA. Table S4 shows the union set of pathways with enrichment p-values < 0.01 by any of the three methods in the BP versus controls comparison. We observe multiple neural related pathways with smaller p-values for sNBLDA_{GLM,sC} compared to sNBLDA_{GLM} and sPLDA, such as “transmission of nerve impulse”, “synaptic transmission”, “neuropeptide hormone activity” and “G-protein coupled receptor binding” (see boldfaced p-values in Table S4). However, when enrichment p-values of sNBLDA_{GLM} and sPLDA are smaller, the pathways are mostly irrelevant to neural function or mental diseases.

To understand why sNBLDA_{GLM,sC} achieves much higher prediction accuracy than sNBLDA_{GLM} in the BP versus Ctrl classification (81.5% accuracy with 32 genes for sNBLDA_{GLM,sC} versus 71.6% accuracy with 30 genes), we show the differential expression analysis by DESeq2 of the 32 genes selected by sNBLDA_{GLM,sC} in Table S5 with disease p-values without covariate adjustment, disease p-values with covariate adjustment and p-values of the three potential confounders (age, ethnicity and PMI). We focus on five genes (LINC01470, RN7SL5P, LRP2, SCARNA5, CD44) with age p-value < 1E-10. Their corresponding disease p-values without covariate adjustment are very significant (1E-3 to 1E-8) but p-values with covariate adjustment are only marginally significant 0.04-0.001 (see boldfaced genes in Supplement Table S5). The exaggerated p-values without covariate adjustment likely come from the imbalanced age distributions in bipolar and control samples (see box-plots of age in BP versus controls in Figure S10). On the other hand, differential expression p-values of NGFR and IGF1 are 0.0001 and 0.007 when covariates are adjusted but their p-values dropped to 0.64 and 0.60 if covariates are not considered. Nerve growth factor (NGF) related genes have been found to provide potential neuroprotective effects.
Insulin-like growth factor 1 (IGF1) is a type of neuropeptide and has also been found to provide neuroprotective function in neurodegenerative disorders (Zheng et al., 2000). Together, these evidences explains why sNBLDA\textsubscript{GLM,sC} achieves a more accurate prediction accuracy than sNBLDA\textsubscript{GLM} and provides more insightful biological findings for the disease. For completeness, we also compare the accuracy of the models to SVM, RF and CART (Figure S11) and the cross-validation accuracies for all these methods are lower than the sNBLDA\textsubscript{GLM,sC} model.

4.2. Cervical tumor miRNA-seq data. This miRNA-seq dataset measures expression level of miRNAs in tumor and nontumor cervical tissues in human samples (Witten et al., 2010). The dataset contains information of over 714 miRNAs for 29 control samples (samples with no tumor) and 29 tumor samples. No clinical information (covariates) is available for adjustment. Although this dataset has small sample size, it has been used in both sPLDA and NBLDA\textsubscript{PE} papers and we demonstrate this application as a secondary example for completeness. Dong et al. (2016) found that NBLDA\textsubscript{PE} performed better than sPLDA in terms of prediction accuracy because of the high dispersion in this dataset. In Figure 4, we compare 10-fold cross-validation accuracy (y-axis) between sPLDA and sNBLDA\textsubscript{GLM} based on 10-fold cross-validation when different number of genes are selected (x-axis; by varying tuning parameter for sparsity) for the corresponding models. Since there is no variable selection in NBLDA\textsubscript{PE}, we only perform cross-validation considering all miRNAs (shown as “X” in the figure). sNBLDA\textsubscript{GLM} generally outperforms the other two methods in different number of selected miRNAs. Specifically, it achieves 95% prediction accuracy with a small number of 37 miRNAs while NBLDA\textsubscript{PE} and sPLDA achieved around 91% accuracy. Although the improvement in accuracy is not statistically significant given the small sample size, the result indicates a trend of improvement of sNBLDA\textsubscript{GLM} in prediction accuracy and variable selection. The 10-fold cross-validation accuracy for SVM, RF and CART are also compared in Figure S12. Again, those methods for continuous input data appear to have inferior prediction performance.

5. Conclusion and Discussion. In this paper, we propose a sparse negative binomial classifier (snbClass) based on a GLM framework with covariate adjustment. The method incorporates three key elements in RNA-seq machine learning modeling: adequate modeling for count data, feature selection and adjustment of covariate effects. Existing methods such as sPLDA do not consider overdispersion properly, NBLDA\textsubscript{PE} does not embed regularization for feature selection, and both methods cannot adjust for covariate
Fig 4: 10-fold cross-validation accuracy (y-axis) of sPLDA (dotted line) and sNBLDA_GLM (dashed line) with varying number of selected miRNAs (x-axis) in the cervical tumor application. NBLDA_PE does not allow variable selection and is shown with “X” symbol.
effect in gene expression. Our new approach assumes a negative binomial model to allow overdispersion, adopts GLM to allow covariate adjustment and facilitates double regularization for feature selection and covariate selection. Extensive simulations and two real applications show superior performance of snbClass (i.e. sNBLDA\textsubscript{GLM.sC}) in terms of prediction accuracy and feature selection.

A potential computational concern of the proposed snbClass method is the convergence issue for the IRLS algorithm and whether it converges to the global optimum. Figure S13 shows the proportions of non-converged simulations after 500 iterations in 100 repeats under different expression level, sample size and overdispersion (1/\(\phi\)). The result shows an increasingly severe convergence issue when sample size is small and overdispersion is large. A further analysis in Figure S14 shows that the non-converged simulations all have > 50% of zero counts in the data when the training set is small, which should be the main cause of non-convergence. Our current package has utilized a routine to automatically exclude non-convergent genes after 80 iterations. It has avoided any convergence issue in our extensive simulations and achieves close to the underlying truth as the global optimum. In real applications, it is a common practice to filter out genes with > 30% zero counts across samples and the convergence issue never happens in all computation and cross validations of the two real applications.

One major limitation of the four count data methods compared in this paper is that the methods are based on gene independence assumption. Due to the complex form of multivariate negative binomial model and the potentially heavy computational cost, it is not directly addressed in this paper but a sensitivity analysis has shown similar pattern and conclusion when the gene dependence structure exists. A extended model with flexible gene dependence will be a future direction. Because of the inherent iterative optimization procedures and larger search space of the tuning parameters, the proposed method is computationally more intensive but is proven practical. The computing time required by each method for the simulations and the two real datasets is given in Table S2. Computing time using the SZ/BP/Ctrl dataset with the largest number of samples (Scenario 1) is 0.515 minute and 1.07 minute for sNBLDA\textsubscript{GLM} and sNBLDA\textsubscript{GLM.sC} for each tuning parameter combination, compared to 0.003 minute and 0.006 minute for sPLDA and NBLDA\textsubscript{PE}. If considering the double regularization tuning parameter estimation, sNBLDA\textsubscript{GLM.sC} requires affordable 38.52 minutes (the number of combinations of the two tuning parameters considered is 540) for the schizophrenia application with 15 cores (under 2 x Sixteen-Core Intel Xeon E5-2683 and 128 GB RAM). An R package “snbClass” and all data/code
used in this paper are available in https://github.com/mdr56/snbclass for reproducibility and convenient application to future datasets.

Acknowledgements. The multi-disease post-mortem brain RNA-seq dataset was obtained from NIMH Repository & Genomics Resource, a centralized national biorepository for genetic studies of psychiatric disorders. Data were generated as part of the CommonMind Consortium supported by funding from Takeda Pharmaceuticals Company Limited, F. Hoffman-La Roche Ltd and NIH grants R01MH085542, R01MH093725, P50MH066392, P50MH080405, R01MH097276, R01MH075916, P50M096891, P50MH084053S1, R37MH057881, AG02219, MH06692, R01MH110921, R01MH109677, R01MH109897, U01MH103392, AG05138, and contract HHSN271201300031C through IRP NIMH. Brain tissue for the study was obtained from the following brain bank collections: the Mount Sinai NIH Brain and Tissue Repository, the University of Pennsylvania Alzheimer’s Disease Core Center, the University of Pittsburgh NeuroBioBank and Brain and Tissue Repositories, and the NIMH Human Brain Collection Core. CMC Leadership: Panos Roussos, Joseph Buxbaum, Andrew Chess, Schahram Akbarian, Vahram Haroutunian (Icahn School of Medicine at Mount Sinai), Bernie Devlin, David Lewis (University of Pittsburgh), Raquel Gur, Chang-Gyu Hahn (University of Pennsylvania), Enrico Domenici (University of Trento), Mette A. Peters, Solveig Sieberts (Sage Bionetworks), Thomas Lehner, Stefano Marenco, Barbara K. Lipska (NIMH).

TR, YL and GT are funded by NIH grant R01CA190766 and R21LM012752. GT and CM are funded by NIH R01MH111601 and P50DA046346. HHI, AT and WH are funded by MOST 107-2118-M-007-001.

Supplementary materials. Supplementary materials are available in the pdf file supplementary.pdf

References.


