Classification and Clustering for RNA-seq data with variable selection

by

Md Tanbin Rahman

MS in Applied Statistics, University of Dhaka, Bangladesh, 2014

BS in Applied Statistics, University of Dhaka, Bangladesh, 2013

Submitted to the Graduate Faculty of

the Department of Biostatistics

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This dissertation was presented

by

Md Tanbin Rahman

It was defended on

June 7th, 2019

and approved by

George C. Tseng, ScD, Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Abdus S. Wahed, PhD, Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Ying Ding, PhD, Assistant Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Hyun Jung Park, PhD, Assistant Professor, Department of Human Genetics, Graduate School of Public Health, University of Pittsburgh

Dissertation Director: **George C. Tseng**, ScD, Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Classification and Clustering for RNA-seq data with variable selection

Md Tanbin Rahman, PhD

University of Pittsburgh, 2019

Abstract

Clustering and classification play an important role in identifying sub-types of complex diseases as well as building a predictive model in the field of medicine. In recent years, lowering of cost and high accuracy has made RNA-seq widely popular which is expected to continue to grow over the next few years. One of the important features of RNA-seq data is its count data structure. While there has been a great deal of literature in both clustering and classification method, most of them are either heuristic or suitable for continuous data and does not directly generalize to count data.

In Chapter 2, we propose a classifier for the count structure of the RNA-seq data with variable selection and covariate adjustment. 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 effects. The proposed sparse negative binomial classifier (snbClass) is evaluated in simulations and two real applications using cervical tumor miRNA-seq data and schizophrenia post-mortem brain tissue RNA-seq data to demonstrate its superior performance in prediction accuracy and feature selection.

In Chapter 3, we discuss a model-based clustering method which can use the count structure of the data. In this paper, we develop a negative binomial mixture model with gene regularization to cluster samples (small n) with high-dimensional gene features (large p). The method is compared with the sparse Gaussian mixture model and sparse K-means using extensive simulations and two real transcriptomic applications in breast cancer and rat brain studies. The result shows superior performance of the proposed count data model in clustering accuracy, feature selection and biological interpretation by pathway enrichment analysis.

Contribution to public health: Transcriptomic data play an important role in identifying genes that are differentially expressed under various external conditions and diseases. RNA-seq data are now the most popular method when measuring the expression level in transcriptomic data. The method proposed in this thesis is tailor-made for classification and clustering in count structure of RNA-seq data.

Table of Contents

List of Tables

List of Figures

1.0 Introduction

1.1 Gene regulation and transcriptomics

1.1.1 Central dogma and gene regulation

Deoxyribonucleic acid (DNA) believed to be holding the absolute information of an organism is central to the theory developed in molecular biology (Schneider-Poetsch and Yoshida, 2018). It is a self-replicating material that contains the genetic information and holds key to all the attributes of an organism. It is also where the central dogma in molecular biology begins. The central dogma in molecular biology hypothesizes the path through which protein synthesis occurs. It begins with transcription of DNA to ribonucleic acid (RNA). The information from DNA is transferred to RNA by complementary base pairing which occurs in the nucleus. In transcription, enzyme RNA polymerase delivers the correct complementary base. In the start of transcription, pre-mRNA is created which includes exon (expression sequences) and introns (insertion sequences). This is followed by the deletion of the introns from the pre-mRNA. Once, the introns are removed, a cap is added to the start site and poly A++ tail is added to the termination site and the resulting messenger RNA (mRNA) is known as transcripts. Once the transcription process is completed, the process of translation occurs in the cytoplasm of ribosomes. This results in the synthesis of the proteins. The process requires mRNA, translation RNA (tRNA), ribosomal RNA(rRNA) where each of them carries out different functions. The mRNA having the codon determines the amino acid to be produced while tRNA joins the correct amino acid to the correct codon sequence by codon-anticodon base pairing. Here rRNA forms the two subunits namely small ribosomal subunits responsible for reading the RNA and large subunits which joins the amino acids to build the polypeptide chain. The initiation process of translation starts with binding the mRNA to the small ribosome. Meanwhile, the initiator tRNA is attached to the start (AUG) codon on the mRNA. The large subunit is then attached to the small subunit in the ribosome. At A site of the ribosome, the second tRNA binds bringing an amino acid with it. Once, the

amino acid brought by tRNA is attached to form polypeptide, the ribosome moves down the mRNA chain allowing the tRNA to read each codon in the mRNA and bring an amino acid accordingly. Hence, the polypeptide chain begins to elongate. This continues until tRNA reaches one of the three stop codons which signals the end of translation. When a polypeptide chain folds into a 3-D shape, it is called a protein. The expression of proteins is controlled by a cellular process known as gene regulation. It controls the rate of gene expression. This occurs through the convoluted relationship between the genes, RNA molecules, proteins and other components of expression system which determines the set of genes to be expressed as well as the regions where they ought to be expressed. Gene regulation involves regulation of transcription, regulating the processing of RNA molecules, regulating the stability of mRNA as well as the regulation of the rate of translation.

1.1.2 Transcriptomic data

In molecular biology, there is a branch of studies ending with -omics. They are essential in understanding the different component of the central dogma. Genomics studies the genetical material. Transcriptomics studies the RNA transcripts while proteins are studied in depth in the field of proteomics. All of them together play an important role in gaining insight into the inner workings of the biological process in organisms. In transcriptomics, the subject of study is all the different RNA molecules in contrast to DNA in genomics and proteins in proteomics. While, the genome which is generally fixed, the expression level of RNA can change with different diseases and external condition in which the organism is in. Hence it allows us to understand the dynamics of gene expression as well as identify the genes that are differentially expressed in response to differing levels of stress, illness and stimulants thereby making progress in genomic and molecular biology research. RNA molecules can be primarily classified as mRNA (protein-coding RNA) and non-coding RNA (ncRNA). The roles of ncRNAs include gene regulation which involves both transcriptional and post-transcriptional regulation, regulation of alternative splicing, controlling transcription factor binding, chromatin modification and mRNA stabilization (Qu and Adeson, 2012). Complicated Diseases like cancer and neurological, developmental, and cardiovascular diseases can

be often traced to inadequate functioning of the non-coding RNAs (Lahtz and Pfeifer, 2011). In contrast, mRNA plays a key role in transcription and reflects the information of almost all expressed genes. More than 90% of the genome is found to be transcribed and 66% of the RNA is estimated to be mRNA (Pertea, 2012). Several different high-throughput technologies exist for gene expression quantification in transcriptomes. The two most widely popular types of data are microarray and RNA-seq data. Note that proteins are the key elements of most molecular activities in a living organism and conceptually proteomics provide better dynamic information than transcriptomics. However, the double helix structure of DNA and reverse transcription technique from RNA to DNA facilitate transcriptomic high-throughput technologies such as microarray and RNA-Seq. On the other hand, techniques to measure proteins such as mass spectrometry and protein microarray are much more difficult and expensive. This leads to popularity of transcriptomic analysis in the field.

1.1.2.1 Microarray data Introduction of microarray DNA in early 1990 allowed the measurement of expression level of thousands of gene simultaneously which led to the inception of the omics era. Technologies predating microarray technology were expensive, time-consuming as they could only detect approximately tens of genes at a time. Here, a glass slide known as microarray contains thousands of spots or locations where DNA molecules are fixed in an orderly manner. Each spot contains millions of identical copies of DNAs corresponding to identical genes. These DNAs are either oligo-nucleotide strands or genomic DNA. When comparing samples from different conditions i.e.(condition A vs condition B), the RNA are first extracted from the samples. They are then reverse transcribed into complementary DNA (cDNA) using enzyme reverse transcriptase and nucleotides labeled with different fluorescent dyes. That is the dyes to condition A and condition B will be different to distinguish between them. The cDNA produced then will be hybridized to the specific spots on the glass slide containing the complementary sequence. Once the hybridization is done, the spots will be excited with a laser to detect the different dyes. The color at each spot will depend on the relative expression level having the color associated with condition A for spots where genes associated with condition A is comparatively highly expressed and vice versa.

1.1.2.2 RNA-seq experiment While microarray technology revolutionized the study of genomics having provided the ability to monitor thousands of genes at the same time, next-generation sequencing had taken it to a whole new level. Two major disadvantages in the microarray are cross-hybridization where cDNA hybridizes to a spot for which it may not perfectly match, leading to high level of background noise and that it can only measure relative expression for transcripts which are present in the spots thereby failing to discover novel mRNAs. In RNA-seq data, at first, the cellular RNAs are isolated and purified. This can be done either by liquid-liquid partitioning or solid-phase extraction. Generally, mRNA is of main interest. Therefore, the total RNA is enriched for mRNA removing the rRNA which is done by selectively removing rRNA or just selecting the mRNA directly. Then the mRNA is fragmented to sizes ranging from 30-300 base pairs known as reads. Once, RNA-seq is prepared appropriately, it is converted to double-stranded cDNA. In order to sequence this cDNAs, specific adapter sequences must be present at the ends of the fragments. The DNA is then amplified for sequencing. Once sequencing is done, the reads are reassembled. When the genome information is known, reads can be directly aligned onto the reference. In the case where this is not known, the transcripts are first needed to be reconstructed from the reads which is known as the de novo assembly. The typical workflow for an RNA-seq is given in figure 1. RNA-Seq experiment generates discrete (count) data by nature and require novel statistical modeling different from continuous data in microarray. The two papers in this thesis will develop machine learning methods for RNA-Seq count data with variable selection.

1.2 Supervised and Unsupervised machine learning

Two main branches in machine learning are supervised and unsupervised machine learning. In supervised machine learning, also known as classification, we know the ground truth of class labels for a set of training data. This training data is used to build a function of data points or classifier which can predict the class for a future object for which the class is unknown. In contrast, in unsupervised machine learning, we want to know the structure

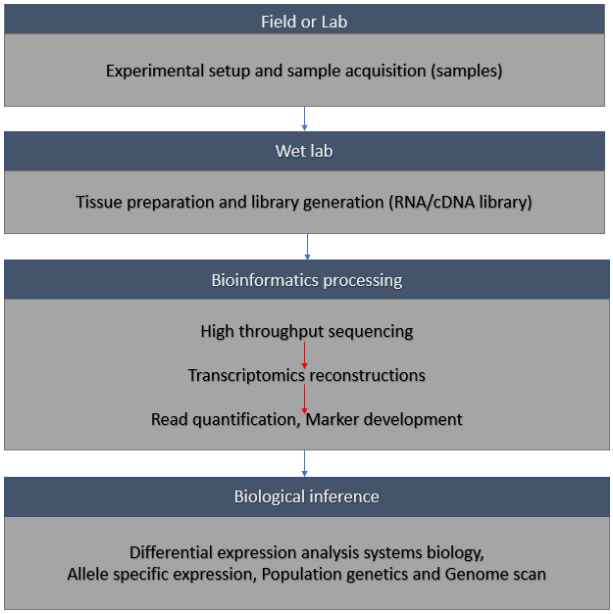


Figure 1: Workflow in RNA-seq data analysis.

of the data without knowing the true class labels. The purpose in unsupervised machine learning is exploratory to get an initial insight and testing individual hypotheses. Popular methods are discussed in details in the following two sections.

1.2.1 Supervised Machine learning

In high dimensional data such as transcriptomic data, predicting class label based on a model remains an important field of study. In microarray data and RNA-seq data, the number of genes is often too high leading to complexities in building the classifier. In response, there have been quite a few methods that were developed (Golub et al., 1999; Hedenfalk et al., 2001). With transcriptomic data, one of the main problems is that often only a small fraction of genes are truly predictive. In addition, the number of samples is often very small compared to the number of genes, giving rise to the small n large p problem. Therefore, it is imperative to identify the genes which truly contribute to the classification. Hence, the sparsity of the model is desirable in order to build a good classifier. Nearest

shrunken centroids (Tibshirani et al., 2003) is a good example since it allows for feature selection in microarray data.

However, for RNA-seq data none of the above are theoretically correct since the data structure, in this case, is count in nature. Hence, a more appropriate classifier which can take account of the count nature of the data is desirable. Two methods proposed for the RNA-seq data for classification are sparse Poisson linear discriminant analysis (sPLDA) (Witten, 2011) and Negative binomial linear discriminant analysis (NBLDA) (Dong et al., 2016). While sPLDA allows for feature selection, NBLDA method does not have sparsity and depend on DE analysis for gene selection in the training data. The advantage of NBLDA lies in the fact that it allows for the overdispersion parameter which sparse PLDA is not able to consider when building the classifier.

Here, we will explore the two classification methods directed towards RNA-seq data taking the count data structure into account. For the purpose of the next section, the following notations are used. In this section, we will first describe two existing methods for classification analysis of count data from RNA-seq and then propose our new method. To unify the notation, denote by $\mathbf{X}$ the count data matrix with elements X_{ij} referred 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 $i-$th row of $\mathbf{X}$, corresponding to feature measurements of observation i. Also, define $X_{.j} = \sum_{i=1}^{n} X_{ij}$, $X_{i.} = \sum_{j=1}^{p} X_{ij}$ and $X_{..} = \sum_{i,j} X_{ij}$. 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 $X_{C_k j} = \sum_{i \in C_k} X_{ij}$.

1.2.2 Two existing methods for classification of RNA-seq data

1.2.2.1 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 ground 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 equivalent mean and variance. To overcome this, Witten (2011) proposed a transformation of count data $X'_{ij} \leftarrow X^u_{ij}$ with a proper choice of u such that, $\sum_{i=1}^{n}\sum_{j=1}^{p} \frac{(X'_{ij} - X'_{i.}X'_{.j}/X'_{..})^2}{X'_{i.}X'_{.j}/X'_{..}} \approx (n-1)(p-1)$. From simulations of the original paper, this correction performs 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 estimate of population prior probability of belonging to the kth 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 $\mathbf{x}^*$ for which we do not know the class label. The classifier assigns $\mathbf{x}^*$ to the class with the largest discriminant score. The paper also implemented a somewhat ad hoc soft-thresholding operator for feature selection in the classifier, which is motivated from univariate lasso regularization in regression for feature selection: $\hat{d}_{kj} = 1 + S(a/b - 1, v/\sqrt{b})$, where $a = X_{C_k j} + \kappa$, $b = \sum_{i \in C_k} \hat{N}_{ij} + \kappa$, κ 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.

1.2.2.2 Negative binomial linear discrimination analysis ($\mathbf{NBLDA_{PE}}$)

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

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

where ϕ_j is the dispersion parameter for the jth gene, $\hat{d}_{kj} = (\sum_{i \in C_k} X_{ij} + 1)/(\sum_{i \in C_k} \hat{s}_i X_{.j} + 1)$ 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 ϕ_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 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 ith sample. For a new sample $\mathbf{x}^*$, the normalizing factor is estimated as $s^* = n \cdot q^* / \sum_{i=1}^{n} q_i$, where q_i ($1 \leq i \leq n$) come from training data and q^* is the 75th count quantile for sample $\mathbf{x}^*$. Note that the vector of normalization factors and dispersion denoted by $\mathbf{s}$ and $\boldsymbol{\phi}$ respectively will be pre-estimated in all negative binomial models in this paper before inference. $\boldsymbol{\phi}$ are estimated by weighted likelihood empirical Bayes method using the edgeR package (Robinson

et al., 2010) with class label considered. We denote the method proposed by Dong et al. (2016) as "NBLDA_{PE}" to emphasize the ad hoc "point estimation" procedure inherited from sPLDA in Witten (2011).

1.2.3 Unsupervised machine learning

Complex diseases like leukemia (Golub et al., 1999) , lymphoma (Rosenwald et al., 2002) , glioblastoma (Parsons et al., 2008), breast cancer (Lehmann et al., 2011) and ovarian cancer (Tothill et al., 2008) were once thought to be a single disease. It was revealed later that each of them had different subtypes based on different gene expression levels. Unsupervised machine learning (also known as clustering) can be useful in detecting identifying those subtypes.

In omics applications, many popular methods such as K-means clustering (MacQueen et al., 1967), hierarchical clustering (Eisen et al., 1998), self-organizing map (SOM; (Kohonen, 1998)) and model-based clustering (Fraley and Raftery, 2002) have been widely used. In transcriptomic data measured in microarray, for example, genes can be clustered into gene modules that suggest co-regulated or co-expressed genes with related biological function. In complex diseases, patients can be clustered to identify novel disease subtypes with distinct disease mechanism or drug responses, which often forms basis for personalized medicine. When clustering such high-dimensional data, hierarchical clustering and SOM are heuristic in nature while model-based clustering assumes the data come from a mixture distribution of two or more clusters. Although the heuristic clustering algorithms are often effective, they lack formal inference (Fraley and Raftery, 2002). Model-based clustering, on the other hand, incorporates distributional features of the data through the density functions and can make inference on the assignment of samples to the clusters. In microarray, model based clustering has been found with superior performance compared to heuristic methods such as hierarchical clustering or SOM (Thalamuthu et al., 2006).

When clustering patients in omics data with thousands of genes, it is biologically reasonable that only a small subset of genes (e.g. 50-200 genes) are cluster predictive. For this purpose, Pan and Shen (2007) proposed a Gaussian mixture model-based clustering

with lasso penalty. Witten and Tibshirani (2010) proposed a sparse K-means algorithm extended from K-means for feature selection. These methods can serve well for clustering transcriptomic data from the old microarray platforms.

Here we will discuss sparse Kmeans and sparse Gaussian model based clustering in more details. We denote x_{ij} for gene expression for jth gene in ith sample.

1.2.3.1 sparse Gaussian clustering model (sgClust) Pan and Shen (2007) proposed a penalized likelihood approach by extending from conventional Gaussian mixture model with a penalty term for feature selection. By assuming zero mean for each gene vector, the penalty term is simply the sum of l_1-norm of all cluster means in all genes. Specifically, the likelihood to be maximized is

$$\log L(\theta; x) = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} p_k f_k(x_i; \theta_k)] - \lambda b(\theta), \tag{1.1}$$

where $f_k(x_i; \theta_k)$ is the density function of multivariate normal distribution with cluster means and variances $\theta_k = \{\mu_k, \Sigma_k\}$, $x_i = (x_{i1}, \cdots, x_{iG})$, p_k is the mixing probability of the k-th cluster and $b(\theta) = \sum_{j=1}^{G} \sum_{k=1}^{K} |\mu_{jk}|$ is the penalty term for regularization. We note that this method assumes diagonal (i.e. independence across genes) and equal covariance matrices across all clusters (i.e. $\Sigma_k = \sigma^2 \cdot I, \forall k$). In real applications, each gene vector is standardized to zero mean before applying the method.

1.2.3.2 sparse K-means Clustering (sKmeans) One of the most popular clustering method, K-means clustering is a classical, efficient and powerful clustering algorithm that seeks to minimize the within cluster sum-of-squares (WCSS). While this is a heuristic method, it is related to Gaussian mixture model-based clustering with equal and spherical covariance matrices in each cluster (Tseng, 2007). In calculating distances for WCSS, traditional K-means adopts equal contribution from each gene feature. In genomic applications, however, the input dataset contains thousands of genes and biologically only a small gene set (sometimes called "intrinsic genes") are relevant to sample clustering. Witten and Tibshirani (2010) proposed a sparse K-means approach to allow feature selection and to improve clustering performance. While K-means minimizes the WCSS, sparse K-means equivalently

seeks to maximize the between cluster sum of squares (BCSS) with gene-specific weight w_j for gene j and an l_1 lasso penalty on w_j. Specifically, sparse K-means seeks to optimize the following weighted target function:

$$\max \sum_{j=1}^{G} w_j \cdot BCSS_j = \sum_{j=1}^{G} w_j \cdot (TSS_j - WCSS_j)$$

subject to $||w||^2 \leq 1$, $||w||_1 \leq s$, and $w_j \geq 0, \forall j$. Here, $TSS_j = \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_j(x_i, x_{i'})$ is the total sum-of-squares, $WCSS_j = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_j(x_i, x_{i'})$ is the within cluster sum-of-squares for gene j, and $d_j(x_i, x_{i'}) = (x_{ij} - x_{i'j})^2$. Note that s is the tuning parameter to control feature selection (i.e. sparsity) and is chosen by gap statistics in the original paper

1.3 Motivation and Overview of the thesis

This thesis focuses on the supervised and unsupervised machine learning for RNA-Seq data. In Chapter 2, a sparse clustering method based on negative binomial distribution is proposed. While Clustering plays an important role in gaining deep insights into the possibility of subtypes within a group, not much work has been done with regards to the RNA-Seq data. The motivation in this chapter is to be propose an appropriate method for clustering with sparsity in the case of count dataset.

In the third chapter, we will discuss a new methodology in terms of classification of RNA-seq data. Here, we will propose a classifier based on negative binomial model with sparsity, which to our knowledge has not yet been developed. Our model also has the advantage that it allows incorporation of information from clinical variables which is not possible in the other two methods which were developed for RNA-Seq dataset.

2.0 A sparse negative binomial classifier with covariate adjustment for RNA-seq data

2.1 Introduction

In the past two decades, microarray and RNA sequencing (RNA-seq) are routine procedures to study transcriptome of organisms in modern biomedical studies. In recent years, RNA-seq (Wang et al., 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 the 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 label so 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 et al., 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 taking 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 et al., 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 overdispersion is small but performs poorly when overdispersion becomes 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 NBLDA_{PE}) 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 ad hoc soft-thresholding operator, similar to univariate lasso estimator in regression, for gene selection in sPLDA but the method is not applicable to the NBLDA_{PE} model due to the addition of dispersion parameter. In the NBLDA_{PE} model proposed by Dong et al. (2016), 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_{PE}, and showed that the power transformed sPLDA generally performed better than NBLDA_{PE} in their simulations and the worse performance in NBLDA_{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_{PE} methods. In Section 1.2.2, we briefly discussed the two existing methods sPLDA (Witten, 2011) and NBLDA_{PE} (Dong et al., 2016). The paper is structured as following. We will discuss our proposed models $\text{sNBLDA}_{\text{GLM}}$ and $\text{sNBLDA}_{\text{GLM.sC}}$ in Section 2.1.1. Section 2.1.2 and 2.1.3 will discuss parameter estimation and model selection of the proposed method. Benchmarks for evaluation are described in Section 2.1.4. Section 2.2 presents simulation studies and Section 2.3 shows two real applications of cervical tumor miRNA-seq data and schizophrenia RNA-seq data. Conclusion and discussion are included in Section 2.4. An R package "snbClass is available at https://github.com/mdr56/snbclass to implement the proposed method.

2.1.1 Proposed method: sparse negative binomial classifier via generalized linear model

We first consider a model without covariates in section 2.1.1.1. Then we extend to incorporate covariates in section 2.1.1.2. The notation will follow from Section 1.2.1.

2.1.1.1 Sparse negative binomial classifier without covariate adjustment ($\text{sNBLDA}_{\text{GLM}}$)

Similar to NBLDA_{PE}, we specify the following negative binomial model in a generalized linear model (GLM) setting:

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

where s_i is the normalization factor of the i-th sample, β_{jk} is the mean count in log-scale of the k-th class for the j-th gene and ϕ_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,

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

where, $\Theta = \{(\boldsymbol{\beta}_k, \boldsymbol{\phi}); k = 1, \ldots, K\}$, $\boldsymbol{\beta}_k = (\beta_{1k}, \ldots, \beta_{pk})$, $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)$, $I(y_i = k)$ 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 $\mathbf{x}^*$ for which we intend to predict the class label. By Bayes theorem, we can derive the discriminant score as

$$\log P(y^* = k|\mathbf{x}^*) \propto \log \hat{\pi}_k - \sum_{j=1}^{p} \phi_j \log[\phi_j + s^* \exp(\hat{\beta}_{jk})] + \sum_{j=1}^{p} X_j^* [\hat{\beta}_{jk} - \log(\phi_j + s^* \exp(\hat{\beta}_{jk}))] \tag{2.1}$$

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 borrows the point estimation of μ_{ijk} from the Poisson model in Witten (2011), while we will derive MLE of Equation (2.2) (see below) using iteratively reweighted least squares (IRLS) method to be shown in Section 2.1.2.1 .

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

$$\log L(\beta; \mathbf{x}, \mathbf{y}, \boldsymbol{\phi}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \left[I(y_i = k) \cdot \sum_{j=1}^{p} \log f(X_{ij}; \beta_{jk}, \phi_j) \right] - \lambda h(\beta) \tag{2.2}$$

Here, β is the collection of all β_{jk} parameters and λ 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.

2.1.1.2 Sparse NB classifier with covariate adjustment sNBLDA$_{\mathbf{GLM.sC}}$

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:

$$X_{ij}|y_i = k \sim NB(\mu_{ijk}, \phi_j);\ \log(\mu_{ijk}) = \log(s_i) + \beta_{jk} + \sum_{q=1}^{Q} \alpha_{qj} z_{iq}, \tag{2.3}$$

Here, $\mathbf{z_q} = (Z_{1q}, \ldots, Z_{nq})$ includes values of the q-th covariate over n samples and parameter α_{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:

$$\begin{aligned}\log L(\beta, \alpha; \mathbf{y}, \mathbf{x}, \mathbf{z_1}, \ldots, \mathbf{z_Q}, \boldsymbol{\phi}) = \sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i = k) \sum_{j=1}^{p} \log f(X_{ij}, Z_{i1}, \ldots, Z_{iQ}; \beta_{jk}, \vec{\alpha}_j, \phi_j) \\ -\lambda_1 h(\beta) - \lambda_2 \sum_{q=1}^{Q}\sum_{j=1}^{p} |\alpha_{qj}|,\end{aligned} \tag{2.4}$$

where, β is the collection of all β_{jk} parameters and α is the collection of all α_{qj} parameters. λ_1 and λ_2 are tuning parameters controlling for levels of sparsity of variable selection in genes and covariates, respectively.

Similarly, for a new sample $\mathbf{x}^*$ with vector of clinical vector $\mathbf{z}^* = (z_1^*, \ldots z_Q^*)$ under this framework, we can derive the following discriminant score:

$$\begin{aligned}\log P(y^* = k|\mathbf{x}^*) \propto \log \hat{\pi}_k - \sum_{j=1}^{p} \phi_j \log[\phi_j + s^* \exp(\hat{\beta}_{jk} + \sum_{q}^{Q} z_q^* \hat{\alpha}_{qj})] \\ + \sum_{j=1}^{p} X_j^* [\hat{\beta}_{jk} + \sum_{q}^{Q} z_q^* \hat{\alpha}_{qj} - \log(\phi_j + s^* \exp(\hat{\beta}_{jk} + \sum_{q}^{Q} z_q^* \hat{\alpha}_{qj})]\end{aligned} \tag{2.5}$$

As before, $\mathbf{x}^*$ 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 α_{qj}. We will denote this method as "sNBLDA$_{\text{GLM.C}}$". 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 λ_2 in Equation (2.4), the method can adequately select a subset of covariates in each gene to improve the performance. For illustration purpose, we refer to this method as "sNBLDA$_{\text{GLM.sC}}$ " 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$_{\text{GLM}}$.

2.1.2 Estimation in sNBLDA$_{\text{GLM}}$ and $sNBLDA_{GLM.sC}$

2.1.2.1 Estimation of sNBLDA$_{\text{GLM}}$ Maximizing the log-likelihood derived in Equation (2.2) is equivalent to minimizing the following penalized weighted least square function,

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\left[I(y_i=k)\sum_{j=1}^{p}w_{ijk}(\tau_{ijk}-\log(s_i)-\beta_{jk})^2\right]+\lambda\sum_{j=1}^{p}\sum_{k=1}^{K}|\beta_{jk}-\bar{\beta}_j|, \tag{2.6}$$

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 (t+1) 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 $\beta_{jk}^{(t+1)}=\text{argmin}\ \frac{1}{2}\sum_i I(y_i=k)w_{ijk}^{(t+1)}(\tau_{ijk}^{(t+1)}-\log(s_i)-\beta_{jk})^2+\lambda|\beta_{jk}-\bar{\beta}_j|$
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,

$$\beta_{jk}^{(t+1)} = \ddot{\beta}_j^{(t+1)} + \text{sign}(\tilde{\beta}_{jk}^{(t+1)} - \ddot{\beta}_j^{(t+1)}) \left[\left| \frac{\sum_{i \in C_k} w_{ijk}^{(t+1)} (\tau_{ijk}^{(t+1)} - \log(s_i)) - \lambda(1 - \frac{1}{K}) \text{ sign}(\tilde{\beta}_{jk}^{(t+1)} - \ddot{\beta}_j^{(t+1)})}{\sum_{i \in C_k} w_{ijk}^{(t+1)}} \right| - |\ddot{\beta}_j^{(t+1)}| \right]_{(+)}$$

Here, $[\ ]_{(+)}$ is 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 β_{jk} under no penalization and $\ddot{\beta}_j^{(t+1)} = \sum_{k=1}^{K} \tilde{\beta}_{jk}^{(t+1)}/K$.

2.1.2.2 Estimation of $sNBLDA_{GLM.sC}$ Similar to $\text{sNBLDA}_{\text{GLM}}$, 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),

$$\sum_{i=1}^{n} \sum_{k=1}^{K} \left[I(y_i = k) \sum_{j=1}^{p} w_{ijk} (\tau_{ijk} - \log(s_i) - \beta_{jk} - \sum_{q=1}^{Q} z_{iq} \alpha_{qj})^2 \right] + \lambda_1 \sum_{j=1}^{p} \sum_{k=1}^{K} |\beta_{jk} - \bar{\beta}_j| + \lambda_2 \sum_{q=1}^{Q} \sum_{j=1}^{p} |\alpha_{qj}| \tag{2.7}$$

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 β_{jk} and α_{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 $\beta_{jk}^{(t+1)} = \text{argmin } \frac{1}{2} \sum_i I(y_i = k) w_{ijk}^{(t+1)} (\tau_{ijk}^{(t+1)} - \log(s_i) - \beta_{jk} - \sum_{q=1}^{Q} z_{iq} \alpha_{qj}^{(t)})^2 + \lambda_1 |\beta_{jk} - \bar{\beta}_j| + \lambda_2 \sum_{q=1}^{Q} \sum_{j=1}^{p} |\alpha_{qj}^{(t)}|$
4. Solve $\alpha_{qj}^{(t+1)} = \text{argmin } \frac{1}{2} \sum_i \sum_{k=1}^{K} I(y_i = k) w_{ijk}^{(t+1)} (\tau_{ijk}^{(t+1)} - \log(s_i) - \beta_{jk}^{(t+1)} - \sum_{q=1}^{Q} z_{iq} \alpha_{qj})^2 + \lambda_1 |\beta_{jk}^{(t+1)} - \bar{\beta}_j^{(t+1)}| + \lambda_2 \sum_{q=1}^{Q} \sum_{j=1}^{p} |\alpha_{qj}|$, where $\bar{\beta}_j^{(t+1)} = \sum_{k=1}^{K} \beta_{jk}^{(t+1)}/K$
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 β_{jk} and α_{qj}. Then the penalized estimate of the parameters in step 3 and step 4 are respectively given by,

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

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

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

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

2.1.3 Selection of tuning parameters in regularization

Both sNBLDA$_{\text{GLM}}$ and sNBLDA$_{\text{GLM.sC}}$ methods involve selection of regularization parameters λ or (λ_1, λ_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. We note that nested cross validation is used for the schizophrenia example (Figure 10) for an unbiased accuracy evaluation. In this case, the outer loop of 10-fold cross validation is conventionally used to estimate accuracy. In each cross validation, the 9 folds of training set undergo an inner loop of 10-fold cross validation to determine λ or (λ_1, λ_2).

2.1.4 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. We also evaluate the performance of $\text{sNBLDA}_{\text{GLM}}$, $\text{sNBLDA}_{\text{GLM.sC}}$ and sPLDA in terms of estimating the true parameters β_{jk} when the gene expression is affected by covariates in Simulation 2 in Section 2.2. Here, we define RMSE$=\sqrt{(1/BpK)\sum_{b=1}^{B}\sum_{j=1}^{p}\sum_{k=1}^{K}(\hat{\beta}_{jk}^{(b)}-\beta_{jk})^2}$ where B is the number of datasets simulated.

2.2 Simulations

In this section, we will devise two simulation schemes to compare the performance of sPLDA and NBLDA_{PE} to our proposed model $\text{sNBLDA}_{\text{GLM}}$ and $\text{sNBLDA}_{\text{GLM.sC}}$ 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 $\text{sNBLDA}_{\text{GLM}}$ over different level of signal strength under three 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}, $\text{sNBLDA}_{\text{GLM}}$ and $\text{sNBLDA}_{\text{GLM.sC}}$ in the presence of covariate effects.

In order to mimic real data, we use a real RNA-seq dataset downloaded from Gene Expression Omnibus (GEO, GSE47474) to retrieve key parameters and perform the simulation. The dataset includes 72 samples with 36 coming from HIV-1 transgenic and 36 from control rat strains (Li et al., 2013). We compute the mean counts of each gene over all samples to obtain an empirical distribution of mean counts, which will be used for obtaining baseline expression levels in all the simulations. Each simulation is repeated 100 times and the average result is reported.

2.2.1 Simulation settings

Simulation 1: 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 $i(1 \leq i \leq 1100)$ in class $k(1 \leq k \leq 3)$, where the number of informative feature 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.
- δ_{jk} represents the pattern of genes j in class k. For all $\delta_{jk} \in \{-1, 0, 1\}$, 1 indicating a 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 Δ_j for each gene j from a truncated normal distribution $TN(\zeta, 0.1^2, \zeta/2, \infty)$, where ζ is the mean, standard deviation is 0.1 and values smaller than $\zeta/2$ are truncated.
- $\phi_j \sim TN(\nu, 0.1, 0, \infty)$ and ν is chosen as 1, 5 and 10 .
- 100 of the samples are used as training set and the remaining 1,000 samples are used as testing set

Simulation 2: 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 $i(1 \leq i \leq 1100)$ in class $k(1 \leq k \leq 3)$ with two covariates (z_1 and z_2; Q=2), where the number of informative feature is 300. The notation of parameters are as follows:

- We generate a binary covariate (e.g. gender) for each sample i from $Ber(0.5)(i.e. z_{1i} \sim Ber(0.5))$ and generate a continuous covariate for each sample i from Gamma(5, 10)
- $\phi_j \sim TN(\nu, 0.1, 0, \infty)$ where $\nu \in \{10, 1\}$

- γ_{qj} represents the pattern of gene j in covariate q for all $\gamma_{qj} \in \{0, 1\}$; there exist three patterns: $(\gamma_{1j}, \gamma_{2j}) = (1, 1), (1, 0), (0, 1),$ and $(0, 0)$ with probablity ($\rho/3$,$\rho/3$,$\rho/3$ and 1-ρ) respectively. When $\rho = 0$, all genes are not impacted by covariates. We choose the proportion of covariate-impacted genes ρ to be 0.125, 0.25 and 0.5.
- Sample the main effect size parameter Δ_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 ϵ_{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 ω has equal probablity of taking 1 or -1. We use the different values of $\eta \in \{0.1, 0.3, 0.5, 0.7\}$ for different level of signal strength of the covariates.
- Other parameters are set the same as Simulation 1 except that ζ is set at 0.25.
- 100 of the samples are used as training set and the remaining 1,000 samples are used as testing set.

2.2.2 Simulation results

Results of Simulation 1 are summarized in Figure 2. In Figure 2(a), average prediction accuracy of the three models sPLDA, NBLDA_{PE} and $\text{sNBLDA}_{\text{GLM}}$ were compared over three different levels of dispersions $\nu \in \{1, 5, 10\}$. The larger the value of ν, the smaller the level of dispersion in the simulated datasets. In all different levels of ζ and ν, $\text{sNBLDA}_{\text{GLM}}$ outperformed the other two methods. As expected, NBLDA_{PE} was superior to sPLDA when ν was small (large overdispersion) but their performances became comparable when ν was large, confirming good performance of power transformation to correct dispersion in sPLDA only for small overdispersion. Figure 2(b) shows results of variable selection by AUC. $\text{sNBLDA}_{\text{GLM}}$ clearly outperformed sPLDA in all cases while NBLDA_{PE} could not perform variable selection and was not applicable in this plot. The new method was also compared to three popular classification methods such as support vector machines (SVM), random forest (RF) and classification and regression tree (CART) in Figure 3. The result showed inferior performance in these methods 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

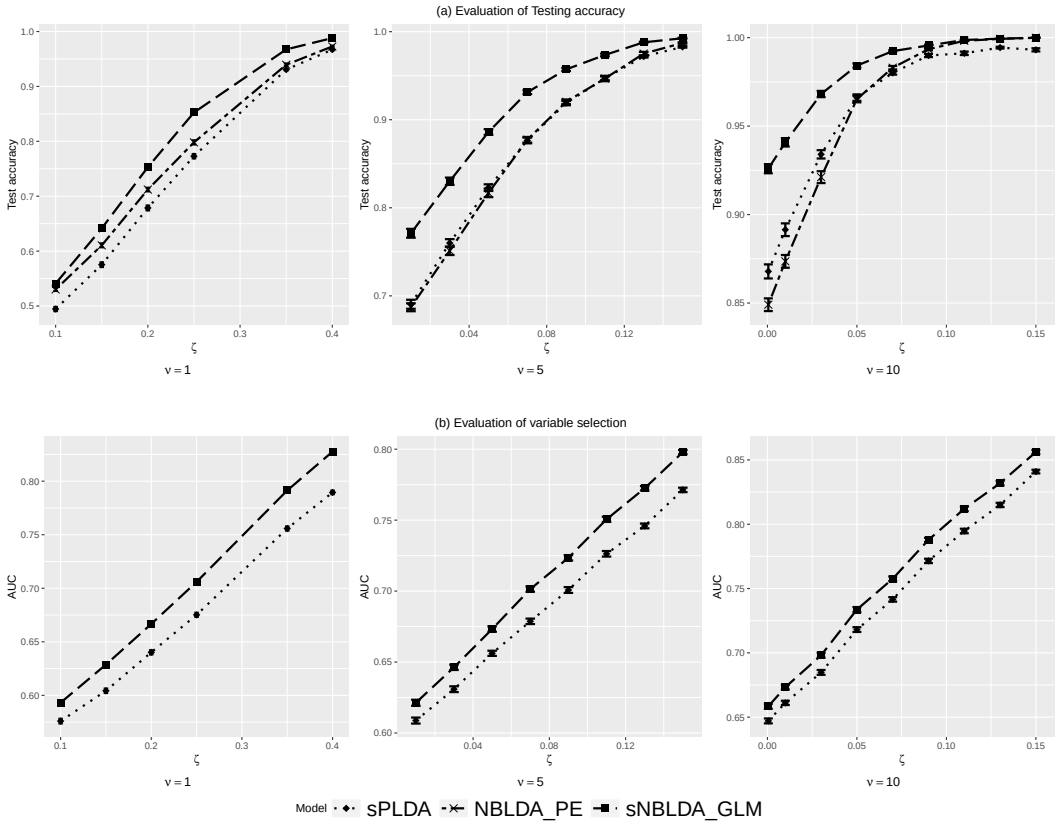


Figure 2: Results for Simulation 1 without covariate effect.

we have used SVM, RF and CART we transform the the data to continuous value by using variance stabilizing transformaton (VST) using package DESeq2 Figure 2 demonstrates results of Simulation 2 using sPLDA, NBLDA_{PE}, $\text{sNBLDA}_{\text{GLM}}$ (no covariate adjustment) and $\text{sNBLDA}_{\text{GLM.sC}}$ (with covariate adjustment and regularization) with varying percent of genes impacted by covariates $\rho = 0.125, 0.25$ and 0.5. Figure 4(a) shows averaged prediction accuracy of varying η and $\nu = 1$ or 10. When $\nu = 1$ (high level of dispersion), $\text{sNBLDA}_{\text{GLM.sC}}$ largely outperformed all other methods as the impact of covariates on gene expression η increased. The prediction accuracy for $\text{sNBLDA}_{\text{GLM.sC}}$ remained high with increased η due to its capacity of adjusting for covariate effect, while prediction accuracy of the other three methods dropped with increased η although $\text{sNBLDA}_{\text{GLM}}$ still outperformed sPLDA and NBLDA_{PE}. When $\nu = 10$, similar pattern was observed. The margin between $\text{sNBLDA}_{\text{GLM.sC}}$ and $\text{sNBLDA}_{\text{GLM}}$ became much smaller but $\text{sNBLDA}_{\text{GLM.sC}}$ was still the best performer. Figure 5 includes comparison with SVM, RF and CART, all of which performed much worse than $\text{sNBLDA}_{\text{GLM.sC}}$.

Variable selection performance between sPLDA, $\text{sNBLDA}_{\text{GLM}}$ and $\text{sNBLDA}_{\text{GLM.sC}}$ is shown in 4(b). Similarly, we observed stable and high performance of $\text{sNBLDA}_{\text{GLM.sC}}$ with increasing η, while performance of $\text{sNBLDA}_{\text{GLM}}$ dropped for increased η due to the lack of covariate adjustment. sPLDA performed the worst in all cases under high level of dispersion. However, when the covariate effect is strong under moderate dispersion, it is seen to outperform $\text{NBLDA}_{\text{GLM}}$ in terms of variable selection. It is intriguing that the variable selection gap between $\text{NBLDA}_{\text{GLM.sC}}$ and $\text{NBLDA}_{\text{GLM}}$ was larger in $\nu = 10$ than in $\nu = 1$, which is contrary to the prediction accuracy in Figure 4(a).

An evaluation of the parameter estimates between the four models sPLDA, $\text{sNBLDA}_{\text{GLM}}$ and $\text{sNBLDA}_{\text{GLM.sC}}$ was carried out in terms of RMSE in Figure 6 under the simulation scheme 2, where $\text{sNBLDA}_{\text{GLM.sC}}$ performed the best. Note that estimation of the class center means for both sPLDA and NBLDA_{PE} were the same in this plot since NBLDA_{PE} borrows the estimating strategy from the sPLDA directly.

To examine the advantage of covariate regularization, we compared $\text{sNBLDA}_{\text{GLM.C}}$ (i.e. $\lambda_2 = 0$ in Equation 2.4; all covariates are used) with $\text{sNBLDA}_{\text{GLM.sC}}$ (with covariate regularization) in Figure 7 . Here, we fixed the $\rho = 0.125$ meaning that only 12.5% of the

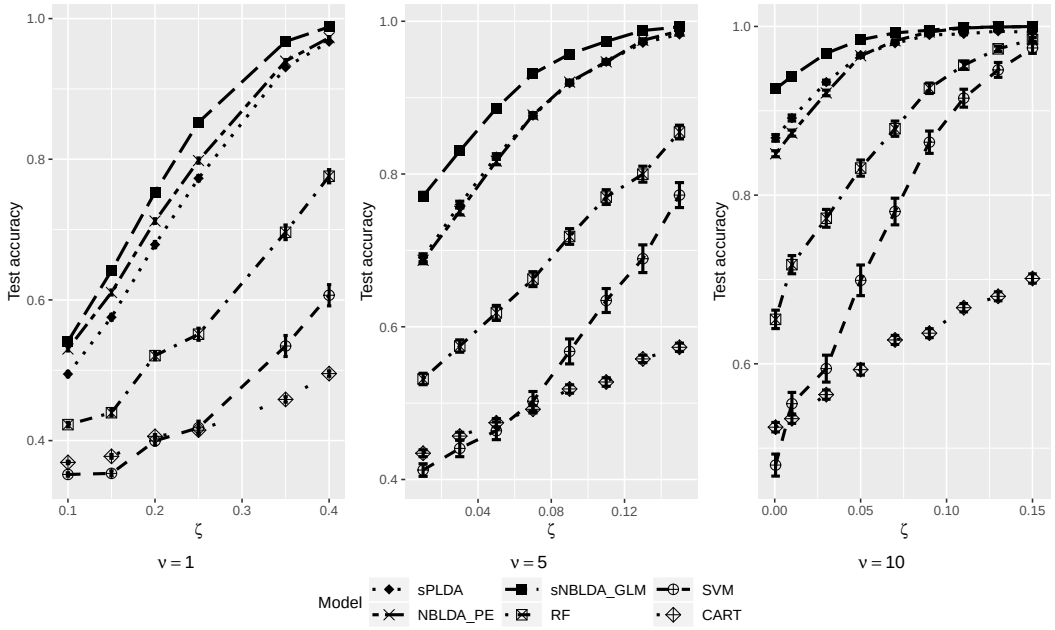


Figure 3: Evaluation of prediction accuracy in Simulation 1 scheme when compared with SVM, RF and CART. RNA-seq count data are transformed by VST method using package to generate input for these machine learning methods.

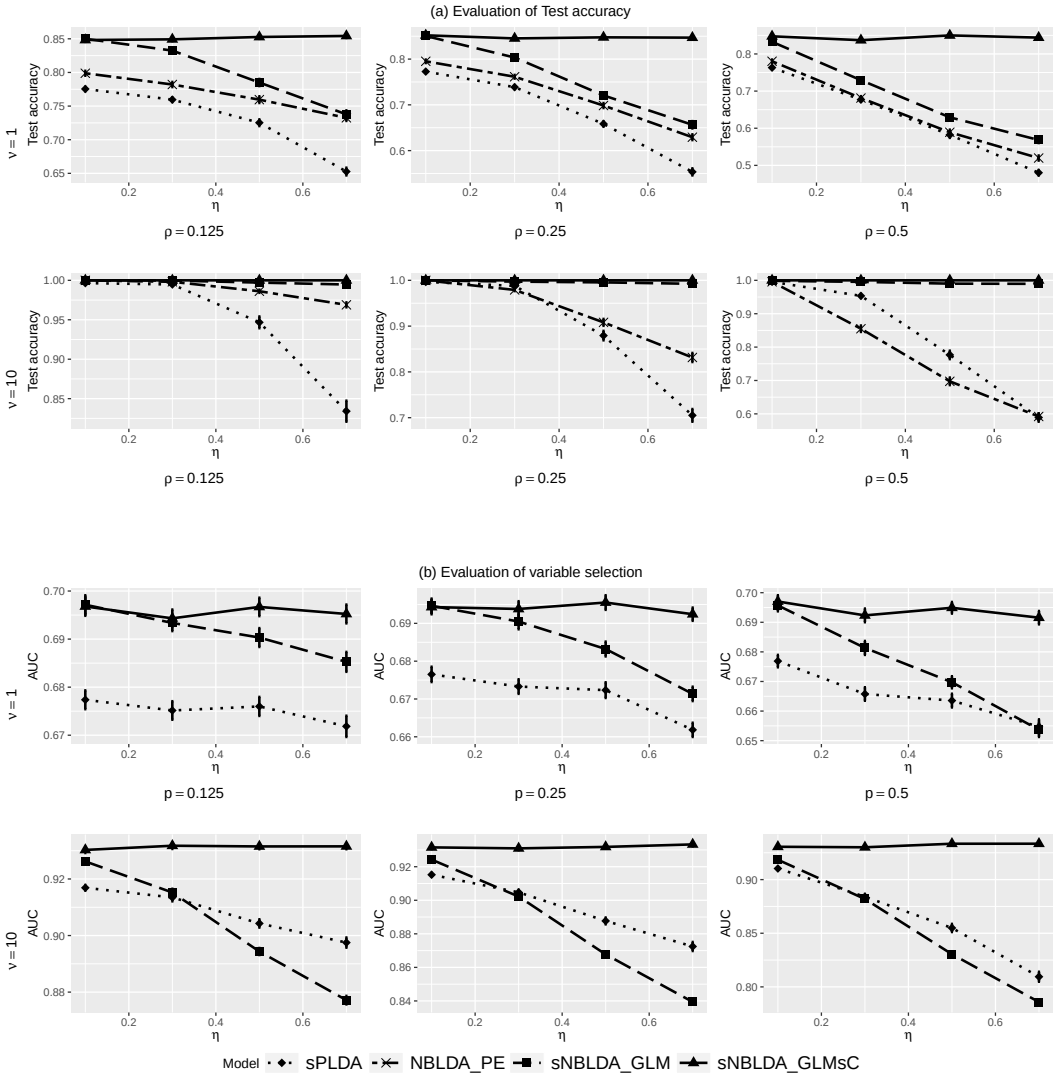


Figure 4: Results for Simulation 2 with covariate effect.

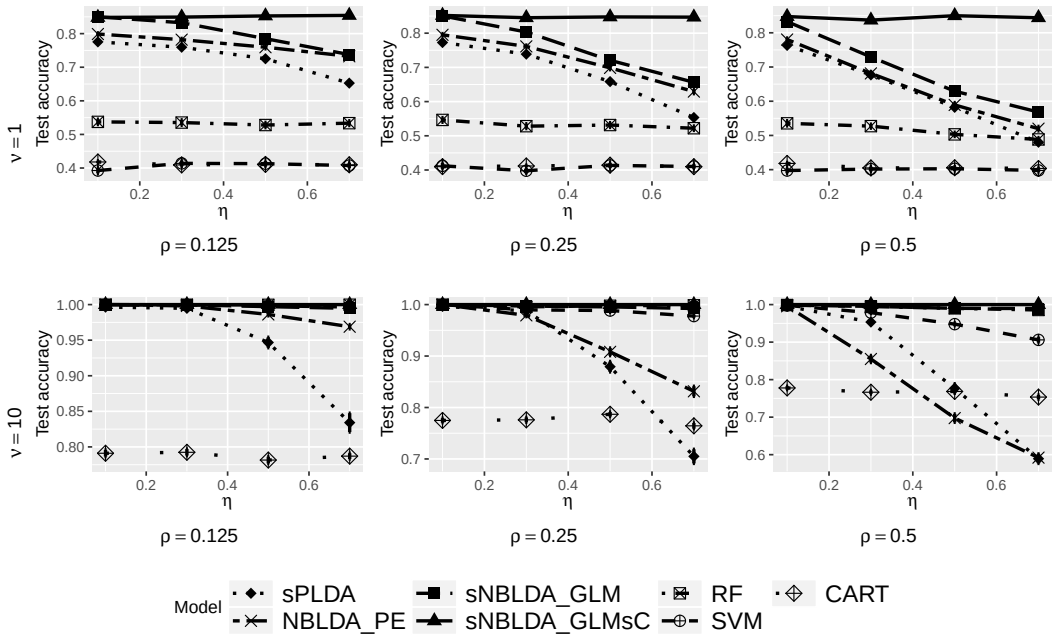


Figure 5: Evaluation of prediction accuracy in Simulation 2 scheme when compared with SVM, RF and CART.

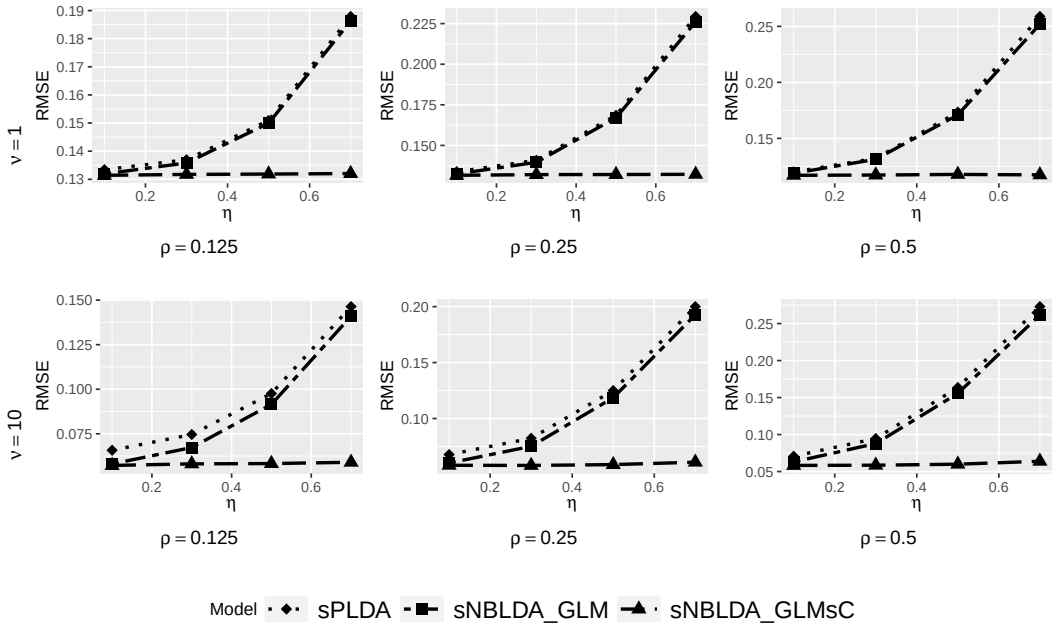


Figure 6: Evaluation of estimation of parameters by RMSE with varying coefficient effect and dispersion in Simulation 2.

genes were actually affected by the covariate expression level. We varied gene strength $\zeta \in \{0.15, 0.20, 0.35, 0.50\}$ with $\nu = 0.5$.. We compared the performance of $\text{sNBLDA}_{\text{GLM.C}}$ to $\text{sNBLDA}_{\text{GLM.sC}}$ over different level of $\eta \in \{0.05, 0.1, 0.5\}$ and compared performance in terms of both prediction accuracy and feature selection. The result showed clear improvement in prediction accuracy due to covariate regularization for larger level of dispersion. However, the gene selection performance was seen to be comparable in this setting.

2.3 Real applications

2.3.1 Cervical tumor miRNA-seq data

This miRNA-seq dataset measured expression level of miRNAs in tumor and nontumor cervical tissues in human samples (Witten et al., 2010). The dataset contains information of over 714 microRNAs for 29 control samples (samples with no tumor) and 29 tumor samples. No clinical information (covariates) is available for adjustment. This dataset has been used in both sPLDA and NBLDA_{PE} papers and thus is a good dataset to evaluate our new method. Dong et al. (2016) found that NBLDA_{PE} performed better than sPLDA in terms of prediction accuracy because of high dispersion estimate in this dataset. In Figure 8 , we compare prediction accuracy (y-axis) between sPLDA and $\text{sNBLDA}_{\text{GLM}}$ based on 10-fold cross-validation when different number of genes are selected (x-axis; by varying tuning parameter for sparsity) as proposed for the corresponding models. Since there is no variable selection in NBLDA_{PE}, we only performed cross-validation considering all miRNAs. $\text{sNBLDA}_{\text{GLM}}$ generally outperformed the other two methods in different number of selected miRNAs. Specifically, it achieved 95% prediction accuracy with a small number of 37 miRNAs while NBLDA_{PE} and sPLDA achieved around 91% accuracy. Although the improvement in accuracy is marginal given the small sample size, the result indicates a trend of improvement of $\text{sNBLDA}_{\text{GLM}}$ in prediction accuracy and variable selection. The 10-fold cross-validation accuracy for SVM, RF and CART are also compared in Figure 9. Again, those methods for continuous input data appeared to have inferior prediction performance.

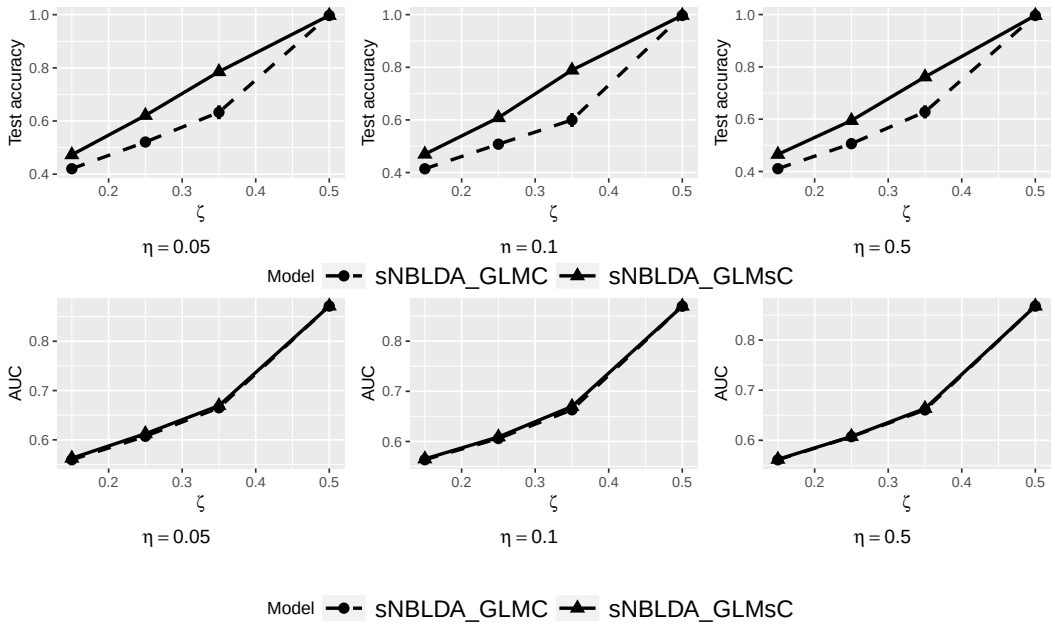


Figure 7: Comparison between covariate selection ($\text{sNBLDA}_{\text{GLM.sC}}$) versus no covariate selection ($\text{sNBLDA}_{\text{GLM.C}}$).

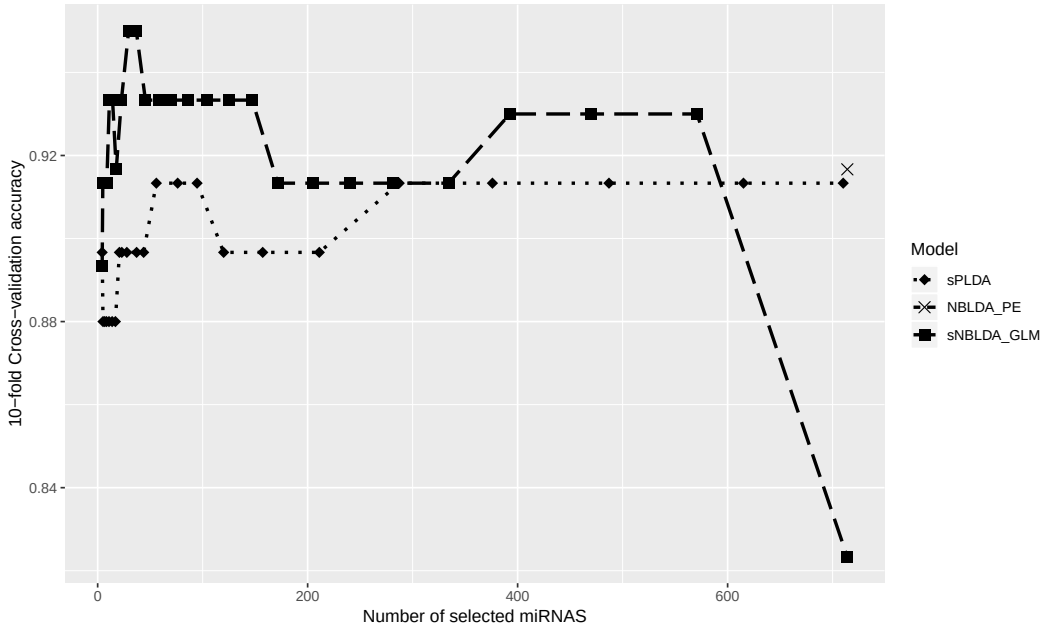


Figure 8: Prediction accuracy (y-axis) of sPLDA (dotted line) and $\text{sNBLDA}_{\text{GLM}}$ (dashed line) with varying number of selected miRNAs (x-axis) in the cervical tumor application. NBLDA_{PE} does not allow variable selection and is shown with "X" symbol.

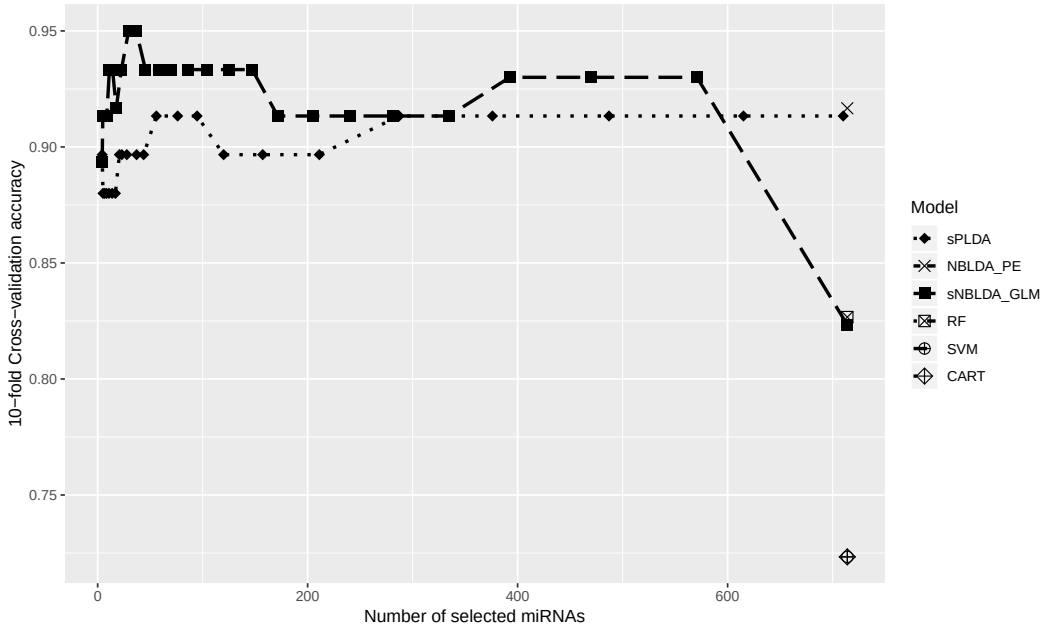


Figure 9: Prediction accuracy (y-axis) of count data models compared to SVM, RF and CART with varying input gene number after DE preprocessing (x-axis) in the cervical miRNA-seq data.

2.3.2 Schizophrenia RNA-seq dataset

The schizophrenia RNA-seq dataset (http://www.synapse.org/CMC) was obtained from the CommonMind Consortium (Fromer et al., 2016) using post-mortem human dorsolateral prefrontal cortex tissues from 258 schizophrenia patients and 279 controls. Here we restricted our analysis to patients with age below 50 and post-mortem interval (PMI; time relapsed from the person has died to the tissues are frozen) less than 30 hours, producing 142 subjects where 96 were controls and 46 suffered from schizophrenia. Five clinical variables were available: age of death, gender, PMI, pH level and ethnicity (Caucasian or African American). At first, we ran a differential expression analysis on each covariate and found a higher percentage of DE genes affected by age of death, ethnicity, PMI and pH. However, since pH had some missing values, we only considered the other three clinical variables in the $\text{sNBLDA}_{\text{GLM.sC}}$ model. We performed 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 16989 genes for machine learning. Similar to simulation and previous application, 10-fold cross-validation was performed to evaluate sPLDA, NBLDA_{PE}, $\text{sNBLDA}_{\text{GLM}}$ and $\text{sNBLDA}_{\text{GLM.sC}}$. We further performed DE analysis to narrow down to 50-1000 genes with a step of 50 in each training set before adopting the four machine learning methods. Even though three of the four methods have embedded feature selection capacity, the feature selection is usually difficult for ultra-high dimensionality (e.g. 16,989 gene features in our case). We performed a pre-screening by differential expression analysis to reduce dimensionality to 50-1000. This procedure is similar to the sure independence screening idea in Fan and Lv (2008) and can usually improve prediction performance.

Figure 10 shows the 10-fold cross validation accuracy of the four methods for different gene size after DE analysis pre-screening. For the three methods with embedded feature selection (sPLDA, $\text{sNBLDA}_{\text{GLM}}$ and $\text{sNBLDA}_{\text{GLM.sC}}$), varied tuning parameters for feature selection were applied and the best prediction accuracy (using nested cross validation to avoid assessment bias) was reported in Figure 10. The result clearly demonstrates better prediction performance of $\text{sNBLDA}_{\text{GLM.sC}}$, especially when the pre-screening by DE analysis reduced

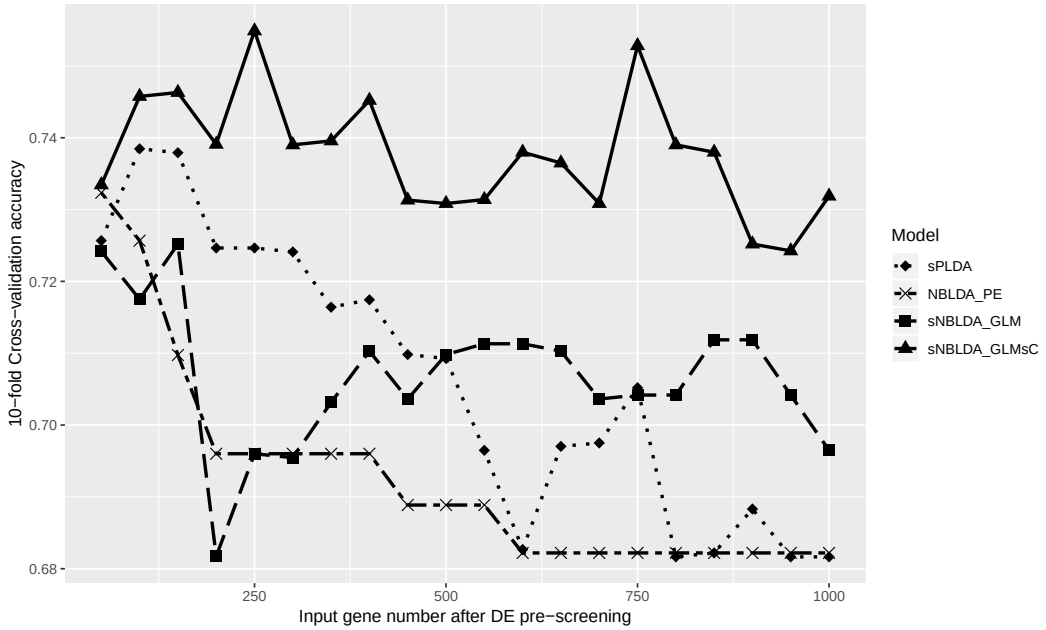


Figure 10: Prediction accuracy (y-axis) of $\text{sNBLDA}_{\text{GLM.sC}}$, $\text{sNBLDA}_{\text{GLM}}$, NBLDA_{PE} and sPLDA with varying input gene number after DE analysis pre-screening (x-axis) in the schizophrenia post-mortem brain RNA-seq data.

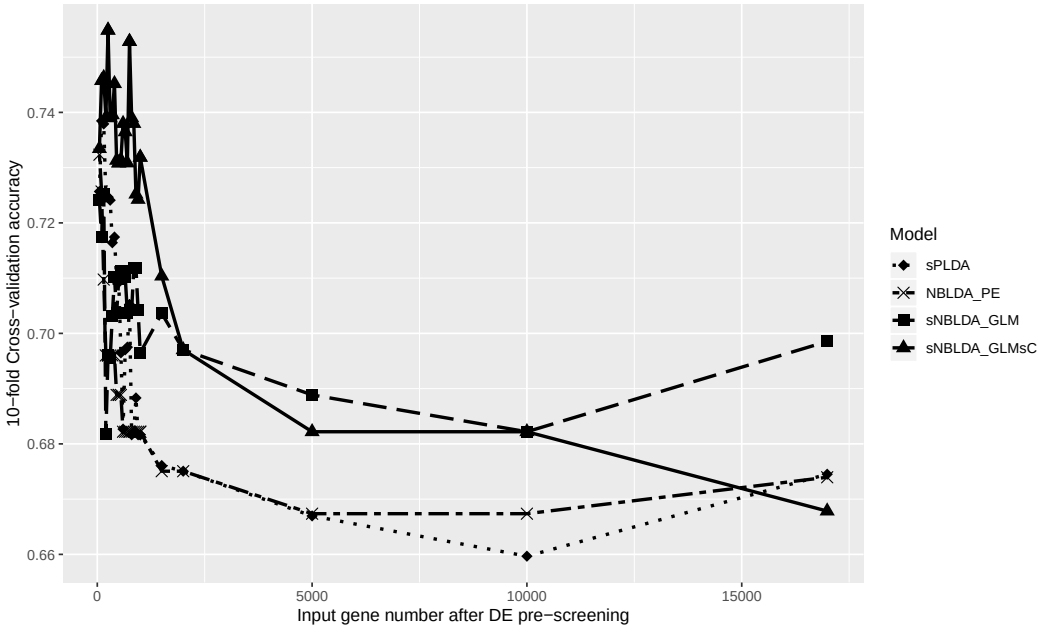


Figure 11: 10-fold Cross-validation by top DE genes in Schizophrenia data.

the input gene size to 50-1000. However, when large number of genes were input to the $\text{sNBLDA}_{\text{GLM.sC}}$ algorithm (e.g. more than 2000 genes after pre-screening), its performance dropped to close to $\text{sNBLDA}_{\text{GLM}}$ and the advantage of covariate adjustment was diminished (Figure 11). As a result, we recommend pre-screening down to 50-1000 genes for ultra-high dimensional data, such as the RNA-seq data in this example, before applying $\text{sNBLDA}_{\text{GLM.sC}}$. For completeness, we also compared $\text{sNBLDA}_{\text{GLM.sC}}$ with $\text{sNBLDA}_{\text{GLM.C}}$ (adjustment with all three covariates without sparsity of covariates) in Figure 12. The result shows inferior performance of $\text{sNBLDA}_{\text{GLM.C}}$, indicating necessity of covariate regularization. We also performed SVM, RF and CART (Figure 13) and the prediction accuracies were generally lower than $\text{sNBLDA}_{\text{GLM.sC}}$.

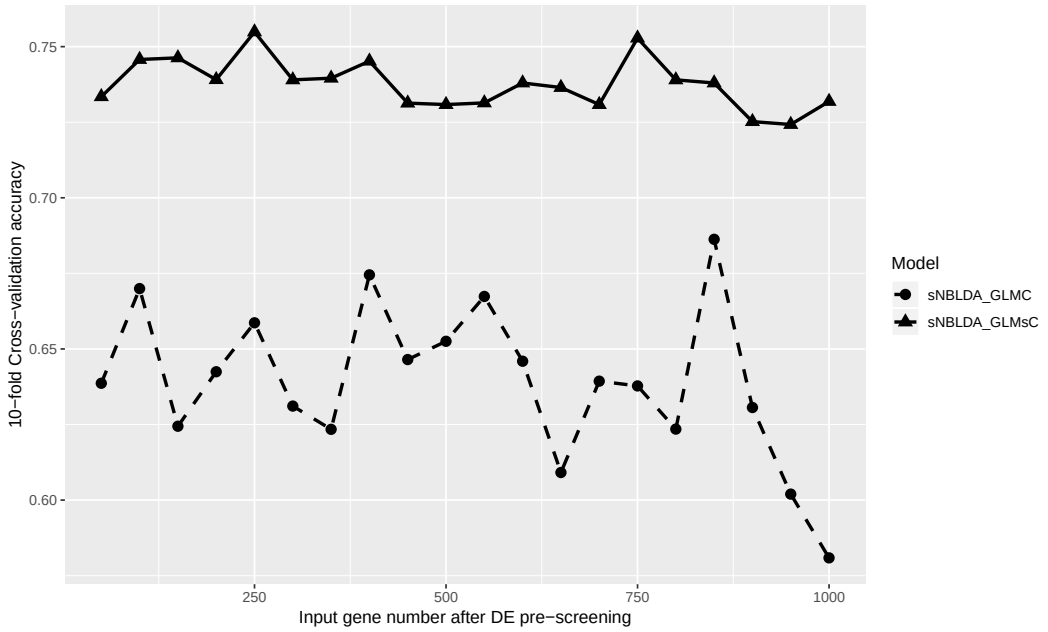


Figure 12: Comparison between covariate selection ($\text{sNBLDA}_{\text{GLM.sC}}$) versus no covariate selection ($\text{sNBLDA}_{\text{GLM.C}}$) in schizophrenia application.

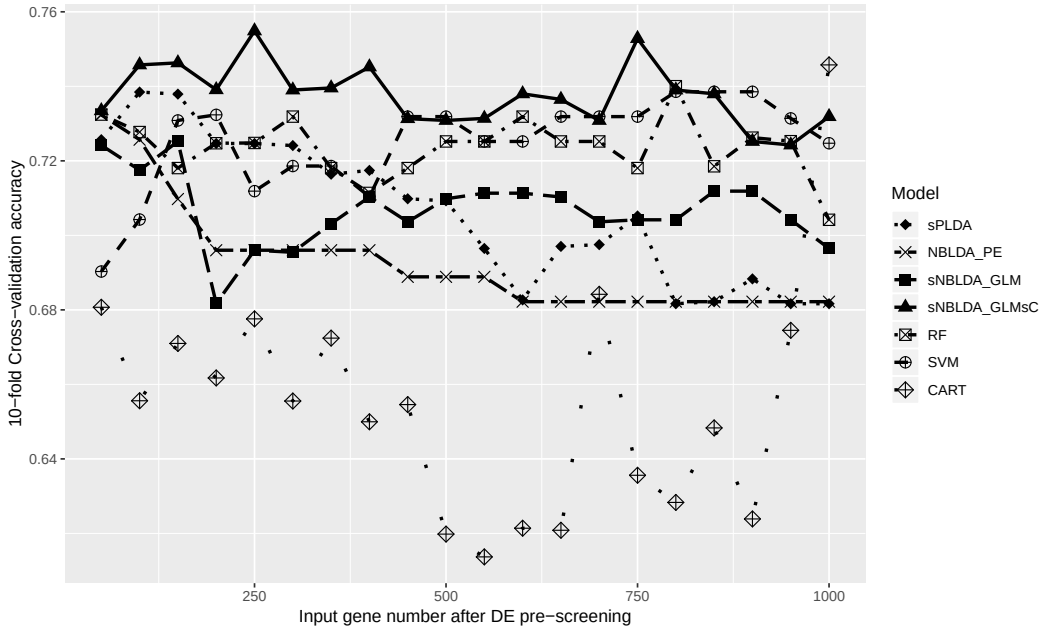


Figure 13: Prediction accuracy (y-axis) of count data models compared to SVM, RF and CART with varying input gene number after DE preprocessing (x-axis) in the schizophrenia RNA-seq data.

2.4 Conclusion and Discussion

In this paper, we propose a sparse negative binomial classifier (snbClass) based on a GLM framework with and without 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 does not consider overdispersion properly, NBLDA_{PE} does not embed regularization for feature selection, and both methods cannot adjust for covariate 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 showed superior performance of snbClass (i.e. $\text{sNBLDA}_{\text{GLM.sC}}$) in terms of prediction accuracy and feature selection. Particularly, snbClass achieved higher prediction accuracy with smaller number of selected genes or miRNAs in 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 independent assumption. Due to the complex form of multivariate negative binomial model and the potentially heavy computational cost, it is not addressed in this paper but will be a future direction. Because of the inherent iterative optimization procedures and large search space of the tuning parameters, the proposed method is computationally more intensive. To analyze the schizophrenia dataset (100 controls and 50 cases with 1,000 genes after pre-screening), $\text{sNBLDA}_{\text{GLM.sC}}$ required 7.12 hours for 10-fold cross validation, compared to 20.85 minutes for sPLDA and NBLDA_{PE} (under Intel Xeon Phi Processor 7210 CPU with 64 cores and 192GB RAM). An R package "snbClass" and all data/code used in this paper are available in https://github.com/mdr56/snbclass for easy reproducibility and convenient application to future datasets.

3.0 Sparse negative binomial model-based clustering for RNA-seq count data

3.1 Introduction

Cluster analysis is a powerful exploratory tool for high-dimensional data. In omics applications, many popular methods such as K-means clustering (MacQueen et al., 1967), hierarchical clustering (Eisen et al., 1998), self-organizing map (SOM) (Kohonen, 1998) and model-based clustering (Fraley and Raftery, 2002) have been widely used. In transcriptomic data measured in microarray, for example, genes can be clustered into gene modules that suggest co-regulated or co-expressed genes with related biological function. In complex diseases, patients can be clustered to identify novel disease subtypes with distinct disease mechanism or drug responses, which often forms basis for personalized medicine and such sample clustering is the focus of this paper. When clustering such high-dimensional data, methods such as hierarchical clustering and SOM are heuristic in nature while model-based clustering assumes that data come from a mixture distribution of two or more clusters. Although the heuristic clustering algorithms are easy to implement and popular, they lack formal inference (Fraley and Raftery, 2002). Model-based clustering, on the other hand, incorporates distributional features of the data through the density functions and can make inference on the assignment of samples to the clusters. In microarray, model based clustering has been found with superior performance compared to heuristic methods such as hierarchical clustering or SOM (Thalamuthu et al., 2006).

When clustering patients in omics data with thousands of genes, it is biologically reasonable that only a small subset of genes (e.g. 50-200 genes) are cluster predictive. For this purpose, Pan and Shen (2007) proposed a Gaussian mixture model-based clustering with lasso penalty. Witten and Tibshirani (2010) proposed a sparse K-means algorithm extended from K-means for feature selection. These methods can serve well for clustering transcriptomic data from the old microarray platforms. In the past ten years, the rapid development of RNA sequencing (RNA-seq) technology has revolutionized the transcriptomic research. Unlike the continuous florescent measurements from microarray, one important feature of

RNA-seq is the (discrete) count-based data after alignment of millions of sequencing reads. In the literature, a common practice is to transform RNA-seq count data into continuous normalized values and directly apply methods that were developed for microarray. This leads to significant loss of information, particularly for genes with lower counts. Methods directly modeling count data are expected to better fit the data generation process and essential data characteristics, and thus perform better.

In the literature, Si et al. (2013) has proposed a count-based model for clustering genes, where variable selection is not needed since n is usually small compared to p. In this paper, we focus on the problem of clustering samples with feature selection using transcriptomic data from RNA-seq. The data are count-based and usually contain $\sim$50-200 samples and $>$10,000 genes (features), which necessitates effective feature selection while performing clustering. We develop a penalized model-based clustering method for RNA-seq count data. Our approach directly deals with the count data without loss of information from transformation to continuous data. Further, we introduce a penalty term in the likelihood to shrink the cluster specific means of each feature towards its global mean across all clusters.

The paper is structured as follows. In Section 3.2, we will summarize two existing methods, sparse Gaussian clustering and sparse K-means, and then propose the sparse negative binomial clustering model. Optimization for the penalized likelihood, Bayesian information criterion (BIC) for model selection and performance benchmarks will be presented. Section 3.3 will cover extensive simulations to benchmark and justify improved performance of the proposed methods. In Section 3.4, two real applications using RNA-seq data from rat brain and breast cancer subtype examples will be evaluated to illustrate improvement of the new method. Section 3.5 contains final conclusion and discussion.

3.2 Existing and proposed methods

We will present two existing methods sparse Gaussian model-based clustering and sparse K-means in 3.2.1. To simplify discussion hereafter, we will abbreviate the sparse Gaussian clustering model as "sgClust" and abbreviate the sparse K-means method as "sKmeans".

We will then present our method sparse negative binomial model-based clustering, snbClust in section 2.2. Section 3.2.3 discusses EM algorithm for optimizing the penalized likelihood function of "snbClust". Section 3.2.4 and 3.2.5 will discuss Bayesian Information Criterion (BIC) for model selection and benchmarks for evaluation, respectively. We assume the raw sequencing reads from RNA-seq experiment are properly preprocessed, aligned and summarized. Denote by y_{ij} the observed counts for gene j ($1 \leq j \leq G$) in sample i ($1 \leq i \leq n$). Our proposed snbClust model will utilize the count data as input. For the two existing methods, sgClust and sKmeans, Gaussian assumption is explicitly or implicitly assumed and only continuous input data are allowed. We will generate log-transformed (base 10) CPM (Counts per Million) values using the edgeR package (Robinson et al., 2010). The resulting log-CPM continuous values are denoted as x_{ij} and are the input data for sgClust and sKmeans.

3.2.1 Two existing methods using continuous data input

3.2.1.1 sparse Gaussian clustering model (sgClust) Pan and Shen (2007) proposed a penalized likelihood approach by extending from conventional Gaussian mixture model with a penalty term for feature selection. By assuming zero mean for each gene vector, the penalty term is simply the sum of l_1-norm of all cluster means in all genes. Specifically, the likelihood to be maximized is

$$\log L(\theta; x) = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} p_k f_k(x_i; \theta_k)] - \lambda b(\theta), \tag{3.1}$$

where $f_k(x_i; \theta_k)$ is the density function of multivariate normal distribution with cluster means and variances $\theta_k = \{\mu_k, \Sigma_k\}$, $x_i = (x_{i1}, \cdots, x_{iG})$, p_k is the mixing probability of the k-th cluster and $b(\theta) = \sum_{j=1}^{G} \sum_{k=1}^{K} |\mu_{jk}|$ is the penalty term for regularization. We note that this method assumes diagonal (i.e. independence across genes) and equal covariance matrices across all clusters (i.e. $\Sigma_k = \sigma^2 \cdot I, \forall k$). In real applications, each gene vector is standardized to zero mean before applying the method. Since no R package are available to the best of our knowledge, we wrote the R functions to carry out the algorithm and include it in our R package.

3.2.1.2 sparse K-means Clustering (sKmeans)

K-means clustering is a classical, efficient and powerful clustering algorithm that seeks to minimize the within cluster sum-of-squares (WCSS). The method is related to Gaussian mixture model-based clustering with equal and spherical covariance matrices in each cluster (Tseng, 2007). In calculating distances for WCSS, traditional K-means adopts equal contribution from each gene feature. In genomic applications, however, the input dataset contains thousands of genes and biologically only a small gene set (sometimes called " informative genes") are relevant to sample clustering. Witten and Tibshirani (2010) proposed a sparse K-means approach to allow feature selection and to improve clustering performance. While K-means minimizes the WCSS, sparse K-means equivalently seeks to maximize the between cluster sum of squares (BCSS) with gene-specific weight w_j for gene j and an l_1 lasso penalty on w_j. Specifically, sparse K-means seeks to optimize the following target function:

$$\max \sum_{j=1}^{G} w_j \cdot BCSS_j = \sum_{j=1}^{G} w_j \cdot (TSS_j - WCSS_j)$$

subject to $||w||^2 \leq 1$, $||w||_1 \leq s$, and $w_j \geq 0, \forall j$. Here, $TSS_j = \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_j(x_i, x_{i'})$ is the total sum-of-squares, $WCSS_j = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_j(x_i, x_{i'})$ is the within cluster sum-of-squares for gene j, and $d_j(x_i, x_{i'}) = (x_{ij} - x_{i'j})^2$. Note that s is the tuning parameter to control feature selection (i.e. sparsity) and is chosen by gap statistics in the original paper. In this paper, the method is implemented using the R package "sparcl".

3.2.2 sparse Negative binomial clustering with varying library size (snbClust)

Since RNA-seq experiment generates count data by nature, the common practice is to transform count data to continuous measures (e.g. logCPM), thereby reducing the statistical power. In the literature, negative binomial model has been widely used for RNA-seq differential expression analysis due to its better model fitness than Poisson model with an additional over-dispersion parameter. Assume,

$$y_{ij}|C_i = k \sim NB(\mu_{ijk}, \phi_j); \;\; \log(\mu_{ijk}) = \log(s_i) + \beta_{jk}, \tag{3.2}$$

where C_i is the cluster assignment for the ith sample, s_i is the normalization size factor of the i-th sample a priori estimated by edgeR (Robinson et al., 2010) to control for the library size variation among samples, β_{jk} is the cluster mean of the k-th cluster for the j-th gene on the log scale after controlling for the library size variation and ϕ_j is the dispersion parameter for the jth gene.

Using the density function for the negative binomial distribution instead of the Gaussian distribution in the mixture model defined before, we can use the true structure of the count data rather than using transformation of the count data. Let $\vec{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iG})$ be the observed counts in sample i with G features. The penalized log-likelihood is given by,

$$\log L(\Theta_1) = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} p_k f_k^{(nb)}(\vec{y}_i; s_i \exp(\vec{\beta}_k), \vec{\phi})] - \lambda h(\beta), \tag{3.3}$$

where $\Theta_1 = \{(p_k, \vec{\beta}_k), \vec{\phi}; k = 1, \ldots, K\}$ is the set of all unknown parameters, $f_k^{(nb)}(\vec{y}_i; s_i \exp(\vec{\beta}_k), \vec{\phi})$ is the density function of $\text{NB}(s_i \exp(\vec{\beta}_k), \vec{\phi})$ with $\vec{\beta}_k = (\beta_{1k}, \beta_{2k}, \ldots \beta_{Gk})$ being the cluster means of cluster k, $\vec{\phi} = (\phi_1, \ldots, \phi_G)$ is the vector of gene-specific dispersion parameters and p_k is the probability of belonging to cluster k. In the penalty term, λ is the tuning parameter and $h(\beta) = \sum_{k=1}^{K} \sum_{j=1}^{G} |\beta_{jk} - \beta_j^*|$ with β_j^* being the MLE of global mean of j-th gene on a log-scale assuming no cluster effect after controlling for the library size variation (see section 3.2.3 for estimate of β_j^*). We note that unlike the Gaussian model in sgClust in Section 3.2.1.1, the count data can not be standardized in each gene row. The subtraction of overall global cluster mean β_j^* for each gene j in $h(\beta)$ is necessary. Maximization of the above likelihood can be achieved by using EM algorithm (Dempster et al., 1977). Here, we introduce a latent variable $z_{ik} = I\{i \in C_k\}$ as the indicator function of cluster assignment for sample i to be assigned to cluster k and the problem becomes maximizing the following complete penalized log-likelihood:

$$\log L_c(\Theta_2) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}[\log(p_k) + \sum_{j=1}^{G} \log(f_k^{(nb)}(y_{ij}; \exp(\log(s_i) + \beta_{jk}), \phi_j))] - \lambda h(\beta), \tag{3.4}$$

where $\Theta_2 = \{(p_k, \vec{\beta}_k, \vec{z}_k); k = 1, \ldots, K)\}$ and $\vec{z}_k = (z_{1k}, \ldots, z_{nk})$. Details of optimization will be illustrated in the next subsection.

3.2.3 Optimization using EM algorithm

Expectation-Maximization (EM) algorithm is a method iterating between an expectation and a maximization step to find the maximum likelihood estimates of parameters in a model with unobserved latent variables (e.g. a mixture model with unknown cluster assignments in our case) (Dempster et al., 1977). In the literature, McLachlan (1997) discussed the estimation of mixture of generalized linear models using iteratively reweighted least square algorithm. Friedman et al. (2010) proposed the estimation of generalized linear model with convex penalties for variable selection using coordinate descent algorithm. For the estimation of snbClust model, we combined the above two ideas to derive a new EM algorithm to estimate the parameters in a mixture of generalized linear model with convex penalties in Equation 3.4. For the gene-specific dispersion parameters ϕ_j's, we estimated a priori by edgeR (Robinson et al., 2010) and plugged into the model. For simplicity, $\vec{\phi}$ will be ignored as we introduce the algorithms below.

We first pre-estimate β_j^* (i.e. the global mean of non-informative feature j) and considered it known during the EM algorithm. β_j^* is estimated by maximizing the following likelihood using iteratively reweighted least square (IRLS) algorithm,

$$\sum_{i=1}^{n}\sum_{j=1}^{G}\log f(y_{ij};\exp(\log(s_i)+\beta_j))$$

Once the vector β_j^* is estimated, we carry out the EM algorithm as follows. The E-step yields:

$$Q(\Theta_2;\Theta_2^{(m)}) = E_{\Theta_2^{(m)}}(\log L_{c,\Theta_2}|Y) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}^{(m)}[\log p_k + \sum_{j=1}^{G}\log f(y_{ij};\exp(\log(s_i)+\beta_{jk}))] - \lambda\sum_j^G\sum_k^K|\beta_{jk}-\beta_j^*|,$$

$$z_{ik}^{(m)} = \frac{p_k^{(m)}\prod_{j=1}^{G} f_k^{(nb)}(y_{ij};\exp(\log(s_i)+\beta_{jk}^{(m)}))}{\sum_{l=1}^{K} p_l^{(m)}\prod_{j=1}^{G} f_{(nb)l}(y_{ij};\exp(\log(s_i)+\beta_{jl}^{(m)}))}$$

In the M-step, the updating function of p is given by,

$$p_k = \sum_{i=1}^{n} z_{ik}/n$$

The updating function of β cannot be easily derived by maximizing the above Q function. We can solve it by using IRLS algorithm, a similar idea recently applied in Wang et al. (2016) under a regression setting. Suppose t is the current iteration of IRLS, we will repeat the following four steps until convergence and return the final estimates of β_{jk} as $\beta_{jk}^{(m+1)}$:

1. Calculate $w_{ijk}^{(t+1)} = \mu_{ijk}^{(t)}/(1 + \phi_j^{-1}\mu_{ijk}^{(t)})$
2. Update $\tau_{ijk}^{(t+1)} = \log(s_i) + \beta_{jk}^{(t)} + (y_{ij} - \mu_{ijk}^{(t)})/\mu_{ijk}^{(t)}$
3. Solve $\beta_{jk}^{(t+1)} = \text{argmin } \frac{1}{2}\sum_i z_{ik}^{(m)} w_{ijk}^{(t+1)}(\tau_{ijk}^{(t+1)} - \log(s_i) - \beta_{jk})^2 + \lambda|\beta_{jk} - \beta_j^*|$
4. Update $\mu_{ijk}^{(t+1)} = \exp(\beta_{jk}^{(t+1)} + \log(s_i))$

The solution in step 3 is given by:

$$\beta_{jk}^{(t+1)} = \beta_j^* + \text{sign}(\tilde{\beta}_{jk} - \beta_j^*)[|\frac{\sum_i z_{ik}^{(m)} w_{ijk}^{(t+1)}(\tau_{ijk}^{(t+1)} - \log(s_i)) - \lambda \text{ sign}(\tilde{\beta}_{jk} - \beta_j^*)}{\sum_i z_{ik}^{(m)} w_{ijk}^{(t+1)}}| - |\beta_j^*|]_+$$

where $\tilde{\beta}_{jk} = \sum_{i=1}^{n} z_{ik}^{(m)} w_{ijk}^{(t+1)}(\tau_{ijk}^{(t+1)} - \log(s_i))/\sum_{i=1}^{n} z_{ik}^{(m)} w_{ijk}^{(t+1)}$ is the estimate of β_{jk} without penalization and f_+ is the soft-thresholding function which takes the value f if $f_+ > 0$ and 0 otherwise.

Once we obtain the estimates $\beta_{jk}^{(m+1)}$ from the IRLS algorithm, we can continue to iteratively carry out E step and M step until convergence to obtain the final MPLE.

3.2.4 Model selection

For all clustering methods, we need to determine the number of clusters, K. This is usually done by first fitting various models with different K, and then using a model selection criterion to select the best K. BIC criterion (Schwarz et al., 1978) is one of the more common method to determine the number of clusters by minimizing the criterion. A modified version

of the BIC was introduced by Pan and Shen (2007) for the sgClust model. Here, we propose a similar BIC approach for estimating K:

$$BIC = -2 \log L(\theta) + \log(n) d_e, \tag{3.5}$$

where $d_e = (K-1) + KP - q$ is the effective number of parameters. In determining d_e, the first term $K-1$ refers to the number of parameters in the mixing probabilities with constraint $\sum p_k = 1$, the second term KP is the number of parameters in cluster means. Finally, q refers to the number of estimates (among the $K \cdot P$ cluster mean parameters) which are shrunken to the global mean. The dispersion parameters are pre-estimated, therefore they are considered known and therefore not included in the BIC criterion.

For the snbClust and sgClust model, BIC criteria is used for both selecting K and the penalty tuning parameter, which determines the number of selected features (i.e. sparsity). Here, we select the tuning parameter in such a way that the corresponding BIC is minimized. In order to select the number of clusters K, we choose the one with minimum BIC over a sequence of tuning parameter. Once we have chosen the number of cluster, we use the BIC criterion to select the tuning parameter. As for sKmeans, gap statistics was proposed in the original paper and software package 'sparcl' is used for model selection.

3.2.5 Benchmarks for evaluation

In a high-dimensional clustering problem, the clustering performance is first benchmarked by the clustering accuracy using adjusted Rand index (ARI) when the true cluster labels are known in simulations and real applications. We next consider performance on feature (variable) selection. In simulation, since the true cluster-predictive features are known, we use receiver operating characteristic (ROC) curve and its area under curve (AUC) for evaluation. In real data, the true cluster-predictive features are unknown. We perform pathway enrichment analysis using Fisher's exact test under different degrees of sparsity to evaluate statistical significance of biological annotation on selected features.

3.3 Simulation

In this section, we conducted three simulations to show the advantages of snbClust while compared to sKmeans and sgClust methods. In simulation 1, we assumed all genes were informative and all samples had equal library sizes. No variable selection was performed so we only assessed the clustering performance. In simulation 2, we assumed only a proportion of genes was informative and assessed both the clustering and variable selection performance. In simulation 3, we performed additional sensitivity analysis by simulating gene-gene dependency structure to examine whether the performance would be affected and whether our independence assumption was valid in general. We repeated 100 times for each simulation and evaluated the averaged results.

To mimic real data structure, we extracted the main characteristics of The Cancer Genome Atlas (TCGA) breast cancer RNA-seq data, which is also used in the second real data example in Section 3.4.2, to perform the simulation. The dataset contains 610 female patients. We first computed the mean counts of each gene over all samples and obtained an empirical distribution of mean counts, which will be used for obtaining baseline expression levels in all three simulations. Since RNA-seq data are usually skewed with many highly expressed house-keeping genes which are irrelevant to cluster analysis, we excluded the top 30% mean counts when forming the empirical distribution. In addition, we also pre-estimated the gene-specific dispersion parameter $\vec{\phi}$ from the data using edgeR (Robinson et al., 2010) and plugged in the estimate.

3.3.1 Simulation settings

Simulation 1: no feature selection and equal library size

1. Sample the baseline expression level of $G = 150$ independent genes μ_j $(1 \leq j \leq 150)$ from the empirical distribution of mean counts constructed above.
2. Use $\delta_{jk} \in \{-1, 0, 1\}$ to represent the pattern of gene j $(1 \leq j \leq 150)$ in cluster k $(1 \leq k \leq 3)$, with 1 indicating the gene is up-regulated in this cluster relative to baseline, -1 indicating down-regulation and 0 indicating no difference. Assume there exist three

gene patterns: $(\delta_{j1}, \delta_{j2}, \delta_{j3}) = (-1, 0, 1)$ for $1 \leq j \leq 50$, $(\delta_{j1}, \delta_{j2}, \delta_{j3}) = (0, 1, 1)$ for $51 \leq j \leq 100$, and $(\delta_{j1}, \delta_{j2}, \delta_{j3}) = (1, -1, 0)$ for $101 \leq j \leq 150$.

3. Sample the log2 fold change (effect size) parameter Δ_j for each gene j ($1 \leq j \leq 150$) and cluster k ($1 \leq k \leq 3$) from a truncated normal distribution $TN(\gamma, 1, \gamma/2, \infty)$ with mean γ, standard deviation 1 and $\gamma/2$ the lower truncation of the distribution (i.e. the minimal effect size). We vary the value of $\gamma \in \{0.5, 0.75, 1, 1.5\}$ for a thorough comparison with the other methods.
4. Denote class label $C_i = k$ for $1 + (k-1) \cdot 15 \leq i \leq k \cdot 15$ (i.e. 15 samples per cluster and 45 samples in total). Sample the count data by $y_{ij}|C_i = k \sim NB(\mu_j \times 2^{\Delta_j \times \delta_{jk}}; \phi)$ for each gene j ($1 \leq j \leq 150$) and sample i ($1 \leq i \leq 45$) in cluster k ($1 \leq k \leq 3$).

Simulation 2: with feature selection

1. As in simulation 1, sample the baseline expression level of $G = 1000$ independent genes μ_j ($1 \leq j \leq 1000$) from the empirical distribution of mean counts.
2. Assume 150 genes are informative and there exist three gene patterns for these informative genes (50 genes in each): $(\delta_{j1}, \delta_{j2}, \delta_{j3}) = (-1, 0, 1)$, $(0, 1, 1)$ or $(1, -1, 0)$. For non-informative genes, the pattern is $(0, 0, 0)$.
3. Sample the log2 fold change (effect size) parameters Δ_j for each gene j ($1 \leq j \leq 1000$) and cluster k ($1 \leq k \leq 3$) from a truncated normal distribution $TN(\gamma, 1, \gamma/2, \infty)$. Here, $\gamma \in \{0.50, 0.75, 1.50\}$.
4. Sample the library size scaling factor a_i from Unif (LB,UB) for each sample i ($1 \leq i \leq 45$), where LB and UB indicate the lower and upper bounds of the uniform distribution. Here, we choose (LB,UB) to be $(0.9, 1.1)$ and $(0.5, 1.5)$ to compare snbClust to the other methods.
5. Sample the count data by $y_{ij}|C_i = k \sim NB(a_i\mu_j \times 2^{\Delta_j \times \delta_{jk}}; \phi)$ for each gene j ($1 \leq j \leq 1000$) and sample i ($1 \leq i \leq 45$) in cluster k ($1 \leq k \leq 3$).

Simulation 3: sensitivity analysis under gene dependency

1. For a total of $G = 1000$ genes, assume 150 genes are informative and there exist three gene patterns for these informative genes (50 genes in each): $(\delta_{j1}, \delta_{j2}, \delta_{j3}) = (-1, 0, 1)$, $(0, 1, 1)$ or $(1, -1, 0)$. For non-informative genes, the pattern is $(0, 0, 0)$.

2. Sample the $\log_2$ fold change (effect size) parameters Δ_j for each gene j ($1 \leq j \leq 1000$) and cluster k ($1 \leq k \leq 3$) from a truncated normal distribution $TN(0.5, 1, 0.25, \infty)$.
3. Sample the baseline expression level μ_j ($1 \leq j \leq 1000$) from the empirical distribution of mean counts. For each gene j in cluster k, obtain $\theta_{jk} = \log_2(\mu_j) + \Delta_j \times \delta_{jk}$.
4. For each gene pattern, sample five gene modules, so there are a total of $M = 15$ modules with $d = 10$ genes in each module ($1 \leq m \leq 15$) for informative genes.
5. Sample the covariance matrix Σ_{mk} for genes in module m ($1 \leq m \leq 15$), cluster k ($1 \leq k \leq 3$). First sample $\Sigma'_{mk} \sim W^{-1}(\vec{\psi}, 60)$, where $\vec{\psi} = (1-\alpha)I_{d\times d} + \alpha J_{d\times d}$ and $\alpha \in \{0, 0.25, 0.5\}$ controls the correlation. Σ_{mk} is calculated by standardizing Σ'_{mk} so that the diagonal elements are all 1's. Here, $I_{d\times d}$ is an identity matrix of dimension $d \times d$ and J_d is a matrix of 1 with dimension $d \times d$.
6. Sample the expression levels of all genes in each module m as $(\beta_{i,(m-1)d+1}, \ldots, \beta_{i,md})|C_i = k \sim MVN((\theta_{(m-1)(d+1)}, \ldots, \theta_{md,k})^T, \Sigma_{mk})$ for sample i ($i \leq i \leq 45$) in cluster k ($1 \leq K \leq 3$).
7. Sample the library size scaling factor a_i from Unif(0.9,1.1) for each sample i ($1 \leq i \leq 45$).
8. Sample $y_{ij} \sim NB(a_i 2^{\beta_{ij}}, \phi)$ for $1 \leq j \leq 150$ and sample i ($1 \leq i \leq 45$). For $151 \leq j \leq 1000$, $y_{ij} \sim NB(a_i 2^{\mu_j}, \phi)$.

3.3.2 Simulation results

Figure 14 shows the mean and standard error of ARI values over 100 replications for the three methods in Simulation 1. Here, the purpose is to evaluate whether using negative binomial distribution to model the count data outperforms other Gaussian-based methods in a simple situation. Here, we considered all the genes to be informative; therefore, only clustering performance in terms of ARI is assessed in this case. For simplicity, we only considered 150 genes and the library size to be constant over all the samples. Compared to Kmeans and gClust methods, our method nbClust had better clustering performance (larger ARI) and the advantage is consistent as we vary the minimal effect size $\gamma/2$. We also show the result for estimation of the number of clusters by BIC criterion in Supplementary Figure S1. Here, we see that the proportion of estimation of correct number of cluster increases

with increasing strength signal γ. In Simulation 2, we evaluate how the performance varies when there are non-informative genes as well as varying γ. The clustering performance is measured using the ARI as before while the variable selection is assessed using the AUC value. The result for this simulation scheme is summarized in Figure 2. In Figure 2(a) we see the comparison of performance between the three methods when the variation of library size is moderate (normalization size factor varies from 0.90 to 1.10). The ARI value of snbClust is higher on average compared to both sKmeans and sgClust. The variable selection performance in terms of AUC in Figure 2(b) is also higher for snbClust compared to the other two methods. When the signal strength γ increases, we observe improved performance for ARI and AUC as expected for all.

Table 1 shows the results in Simulation 3 for varying gene dependnce correlation α. As we can see, the performance of snbClust sligtly decreased but is stable even when α increases up to 0.50, partially justifying the gene-gene independence assumption in our model. Intuitively, in high dimensionality, points are much better separated and ignoring gene dependence structure may not greatly impact the clustering performance. (Donoho, 2000).

3.4 Real data application

3.4.1 Multiple brain regions of rat

In the first example, we applied our method to a RNA-seq dataset studying the brain of HIV transgenic rat from Gene Expression Omnibus (GEO) database (Li et al., 2013). RNA samples from three brain regions (hippocampus, striatum and prefrontal cortex) were sequenced for both control strains and HIV infected strains. Only the 36 control strains (12 samples in each brain region) were used here to see whether samples from the three brain regions can be correctly identified ($K = 3, n_1 = n_2 = n_3 = 12$). After standard preprocessing and filtering out genes with mean counts smaller than 10 based on the guidance in edgeR (Robinson et al., 2010), 10280 genes remained for clustering analysis. In this application, the true cluster labels (brain regions) are known and ARI can be evaluated for clustering

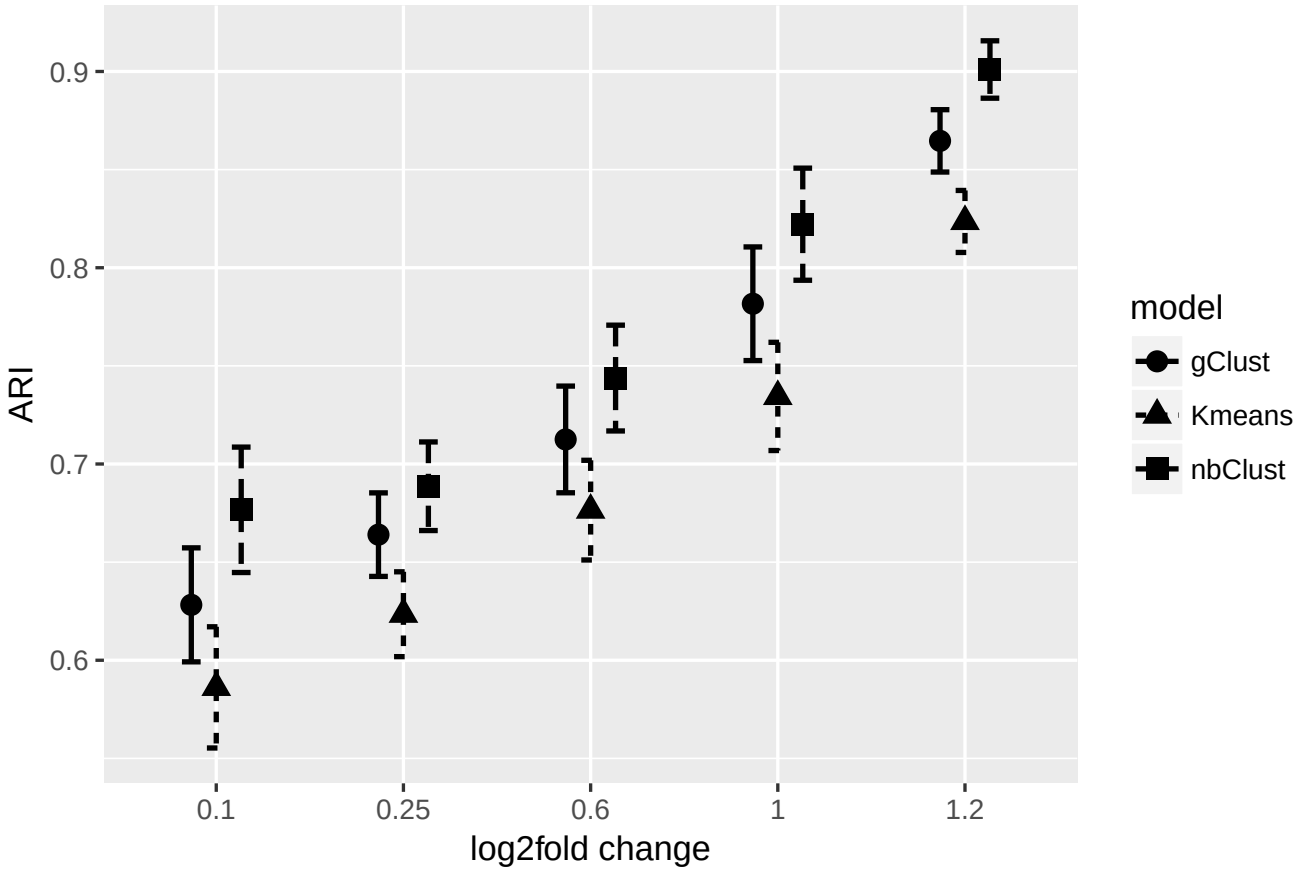


Figure 14: ARI by signal strength γ for simulation scheme 1 when no feature selection is needed.

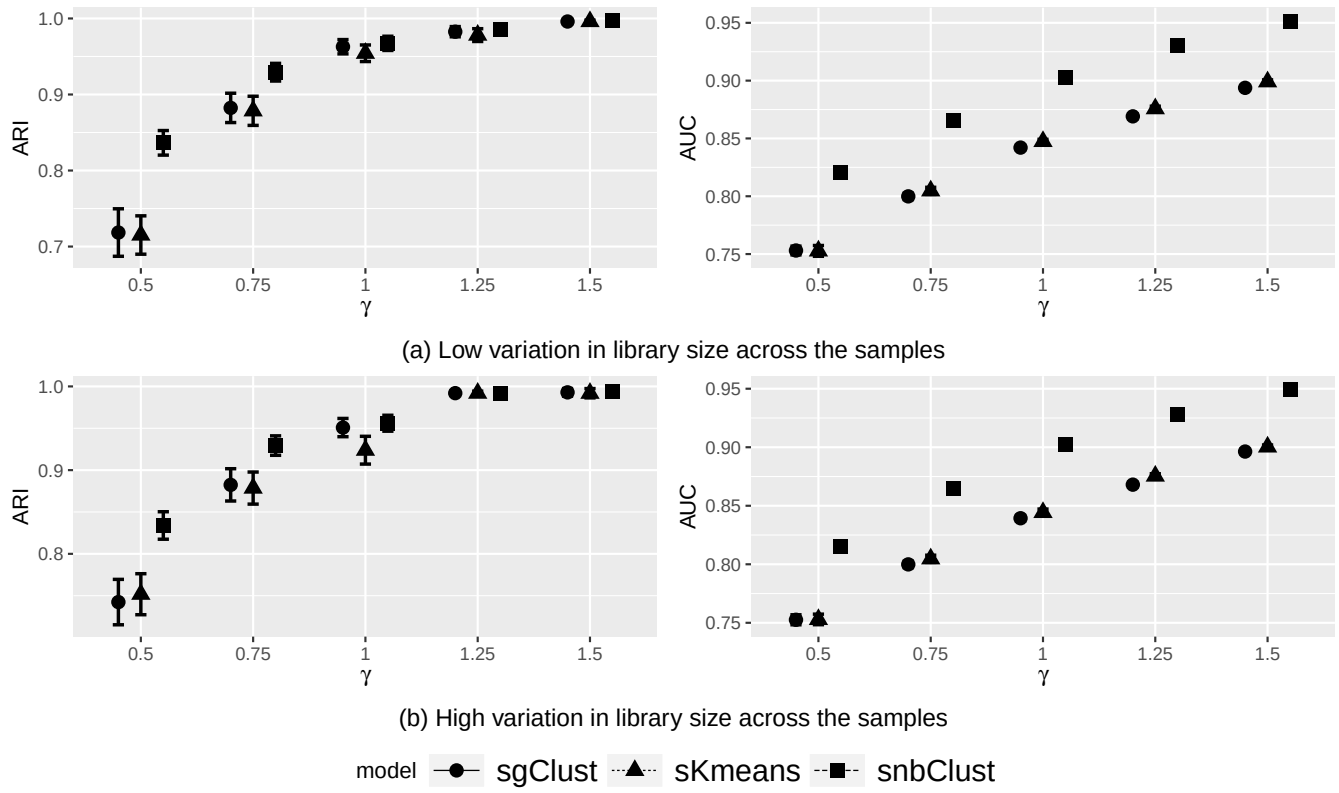


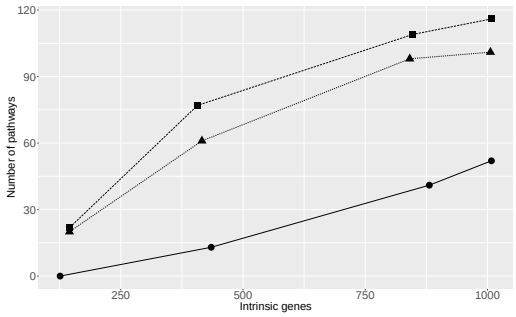
Figure 15: Clustering accuracy by ARI and feature selection accuracy by AUC for Simulation scheme 2.

Table 1: ARI and AUC performance when gene-gene correlation α exists

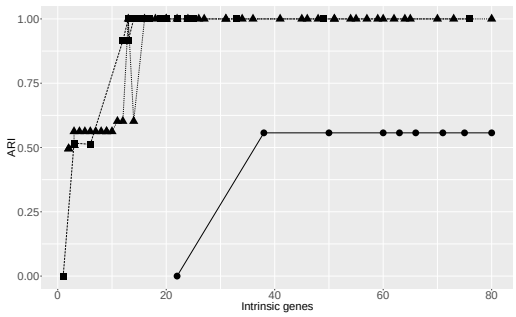
Correlation	Model	ARI	AUC
0.00	snbClust	0.817(0.017)	0.837(0.003)
	sgClust	0.551(0.037)	0.732(0.005)
	sKmeans	0.635(0.022)	0.738(0.004)
0.25	snbClust	0.832(0.015)	0.832(0.005)
	sgClust	0.447(0.037)	0.722(0.004)
	sKmeans	0.635(0.023)	0.739(0.004)
0.50	snbClust	0.748(0.020)	0.818 (0.004)
	sgClust	0.585(0.030)	0.710(0.006)
	sKmeans	0.572(0.028)	0.725(0.006)

accuracy. However, the true informative genes are unknown and the AUC cannot be assessed for feature selection accuracy, as was done in simulation. Instead, we obtain a sequential number of selected genes (around 50-1000) by varying tuning parameters. We then performed pathway enrichment analysis by using Fisher's exact test based on the Gene Ontology (GO), KEGG and Reactome pathway databases to assess the biological relevance of selected genes.

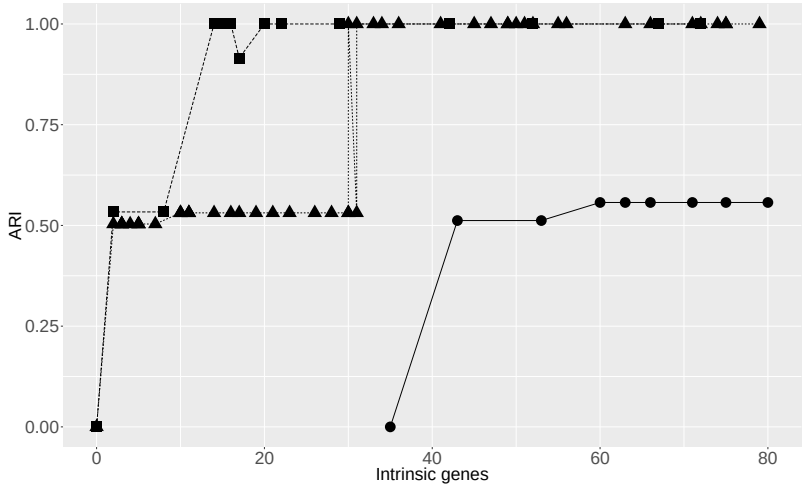
Figure 16(a) showed the number of enriched pathways ($FDR = 0.05$) when different numbers of genes (by tuning λ) were selected. Compared to sKmeans and sgClust methods, snbClust had more enriched pathways at all selected gene numbers, implying the better functional association of selected genes by snbClust. Figure 16(b) shows the ARI value of each method. Both snbClust and sgClust demonstrated a perfect clustering performance (ARI=1) when more than 20 genes were selected while sgClust performed poorly below the top 80 informative genes. To distinguish performance of different methods further, we randomly subsampled the sequencing counts and examined the performance of shallower sequencing data. Figure 16(c) and 16(d) shows the ARI results when we downsampled the sequences to only 50% and 20% of their original total reads. At 50% subsampling sKmeans and snbClust required more than 30 selected genes to achieve perfect ARI and snbClust



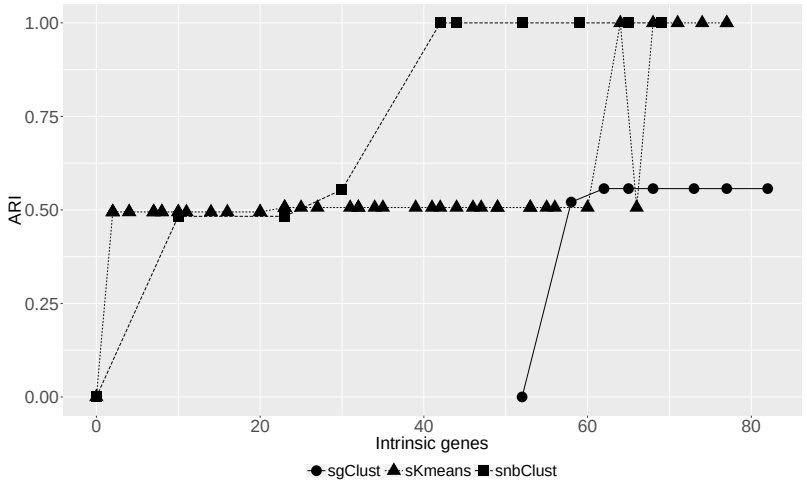
(a) Number of Pathways by Informative genes



(b) ARI by Informative genes (complete data)



(c) ARI by Informative genes (Downsampled to 50%)



(d) ARI by Informative genes (Downsampled to 20%)

Figure 16: Comparison of snbClust, sgClust and sKmeans in rat dataset.

only needed 20. When sequencing depth reduced further to 20%, sKmeans needed 70 genes and sgClust required 120 genes to achieve ARI=1. snbClust only needed around 40 genes. The performance for sgClust have been found to be quite poor compared to the other two methods. Since the input for the sgClust is standardized to mean 0 and variance 1 for genes, we found the the top informative genes have means almost identical. Hence, finding tuning parameter to have smaller subset of top genes have been found difficult. When we used BIC or gap method to select the fixed tuning parameter, sKmeans and sgClust selected 9846 and 10280 genes respectively. The BIC of snbClust selected a more reasonable 1,311 gene set for clustering.

3.4.2 Breast Cancer dataset

Next, we applied the three methods to the Cancer Genome Atlas (TCGA) Breast cancer dataset. The dataset contains patients with 610 female patients with four differnt subtypes of breast cancer: Basal (116 subjects), Her2 (63 subjects), LumA (257 subjects) and LumB (174 subjects). After standard preprocessing and using the criteria of filtering out genes with mean count less than 5 and variance less than the median variance, 8789 genes were retained. LumA and LumB expression patterns were known to be similar, hence, three clusters were considered for evaluation are Basal, Her2 and LumA+LumB. The evaluation was performed similarly to the rat brain example. As shown in Figure 17(a), snbClust reached the highest clustering accuracy at 77.3% when 642 genes were selected and generally outperformed sgClust and sKmeans. Performance of sgClust dropped dramatically when the number of selected genes increased. In terms of pathway analysis, snbClust also performed the best with larger number of enriched pathways compared to the other two methods when selecting 127$\sim$1,000 top genes. This is illustrated in figure 17(b).

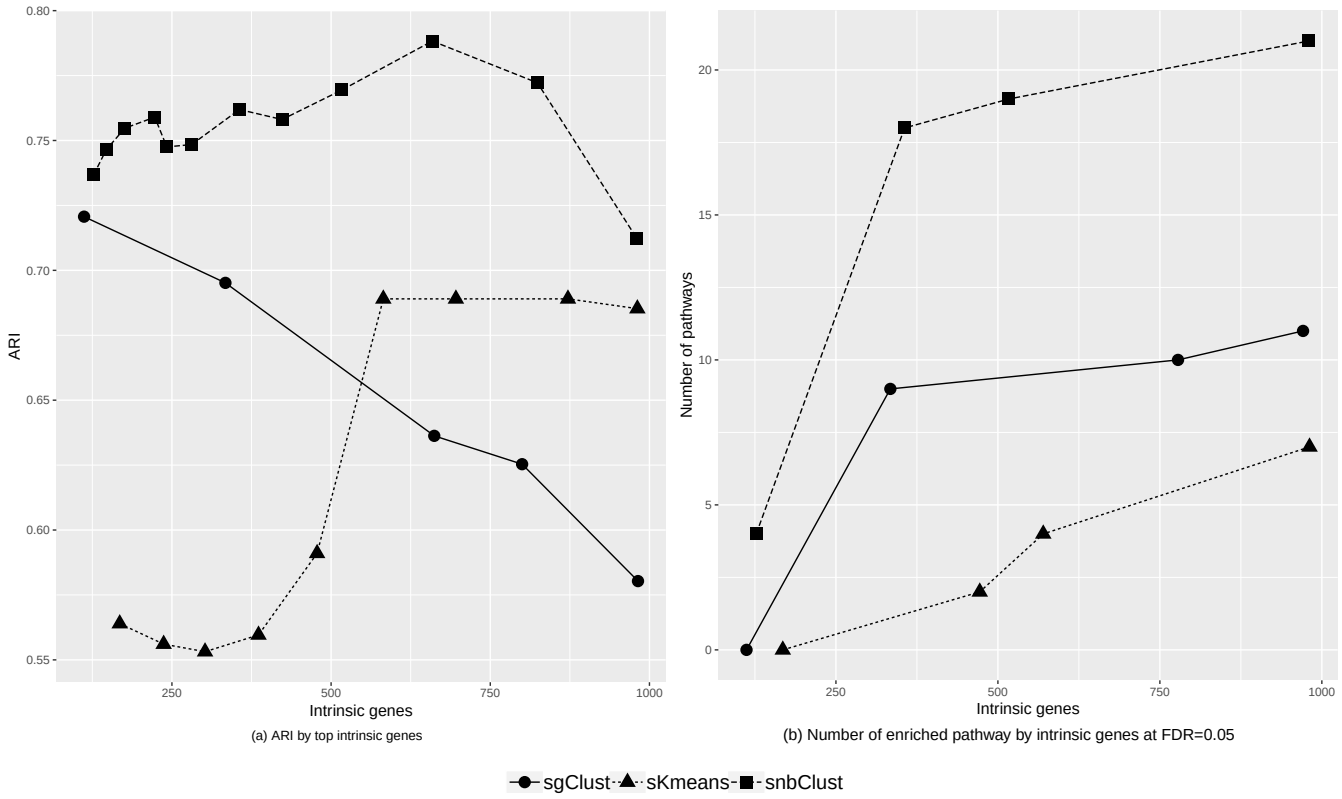


Figure 17: Comparison of snbClust, skmeans and sgClust model in Breast cancer data.

3.5 Discussion and Conclusion

In this paper, we proposed a sparse model-based clustering analysis with negative binomial mixture distribution. Since RNA-seq data are known to be discrete and skewed, negative binomial is a more appropriate distribution to capture the data characteristics, while normalizing counts to continuous and applying Gaussian-based models lose information and efficiency. The extensive simulations and two real applications clearly confirmed this intuition.

There are two limitations in the current model. Firstly, The new count data model requires heavier computing than Gaussian-based models although still in an affordable range for general omics application. Time needed for each simulation scheme is given in Table 2. Similar to all optimization-based clustering algorithms, initial value plays an important role for successful clustering of all three methods. Secondly, the new model does not consider gene correlation structure that may be prevalent among the genes. In high dimensional data where the number of the features is considerably larger compared to the number of samples and the fact of complex multivariate negative binomial distribution, incorporating the correlation structure in the model is not addressed in this paper and will be a future direction. However, we performed sensitivity analysis to examine performance impacted by existence of varying level of correlation structure. We found generally robust clustering and feature selection result in our model.

Table 2: Average time per run in each simulation scheme (in minutes)

	Simulation scheme 1	Simulation scheme 2	Simulation scheme 3
snbClust	7.81(0.951)	186.11(25.2)	180.21(26.3)
sgClust	2.05(1.95)	154.00(24.3)	163.00(30.12)
sKmeans	0.02(0.01)	0.05(0.01)	0.06(0.01)

4.0 Discussion and future work

The research work comprising this dissertation focuses on developing machine learning methods for the count data structure in RNA-seq data. In the first paper, we propose a classifier based on the negative binomial distribution and generalized linear modeling framework with embedded feature selection for the count data structure of the RNA-seq data. The GLM framework proposed in the paper enables us to control for the effect of clinical variables on the gene expression in building the classifier. As a result, the classifier is shown to have superior performance to the other methods proposed for classification in RNA-seq data.

In the second paper, we develop a clustering algorithm for identifying samples of similar expression with embedded feature selection. This is achieved by using penalized negative binomial mixture model with lasso penalty. Through simulations and real data we showed that the clustering algorithm can outperform the popular sparse kmeans and sparse Gaussian mixture model proposed for high-dimensional continuous data.

However, there are some challenges that are not addressed in this paper and remains a future direction. Both of the models proposed are based on the assumption of independence among the genes which in practice is quite restrictive. Due to the complexity of the multivariate negative binomial distribution, incorporating interdependence between genes directly remains a very difficult task. One possible direction is incorporating the correlation between the genes through a Bayesian framework. Other than that, extensions of the methodology in this paper could be applicable for other types of -omics data such as single-cell RNA-seq.

Pan et al. (2006) proposed a semi-supervised machine learning method for application in microarray data. This could easily be extended to RNA-seq data using the methodology discussed in the research work.

Bibliography

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

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recogn.*, 30(7):1145–1159.

Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences*, 97(1):262–267.

Bullard, J. H., Purdom, E., Hansen, K. D., and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. *BMC Bioinformatics*, 11(1):94.

Chu, Y. and Corey, D. R. (2012). Rna sequencing: Platform selection, experimental design, and data interpretation. *Nucleic Acid Therapeutics*, 22(4):271–274. PMID: 22830413.

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., et al. (2016). A survey of best practices for rna-seq data analysis. *Genome Biology*, 17(1):13.

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

Díaz-Uriarte, R. and De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3.

Dong, K., Zhao, H., Tong, T., and Wan, X. (2016). Nblda: negative binomial linear discriminant analysis for rna-seq data. *BMC Bioinformatics*, 17(1):369.

Donoho, D. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. pages 1–32.

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

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631.

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

Fromer, M., Roussos, P., Sieberts, S. K., Johnson, J. S., Kavanagh, D. H., Perumal, T. M., Ruderfer, D. M., Oh, E. C., Topol, A., Shah, H. R., et al. (2016). Gene expression elucidates functional impact of polygenic risk for schizophrenia. *Nature Neuroscience*, 19(11):1442.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, 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(5439):531–537.

Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Raffeld, M., Yakhini, Z., Ben-Dor, A., Dougherty, E., Kononen, J., Bubendorf, L., Fehrle, W., Pittaluga, S., Gruvberger, S., Loman, N., Johannsson, O., Olsson, H., Wilfond, B., Sauter, G., Kallioniemi, O.-P., Borg, ., and Trent, J. (2001). Gene-expression profiles in hereditary breast cancer. *New England Journal of Medicine*, 344(8):539–548. PMID: 11207349.

Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, 21(1-3):1–6.

Lahtz, C. and Pfeifer, G. P. (2011). Epigenetic changes of dna repair genes in cancer. *Journal of Molecular Cell Biology*, 3(1):51–58.

Lehmann, B. D., Bauer, J. A., Chen, X., Sanders, M. E., Chakravarthy, A. B., Shyr, Y., and Pietenpol, J. A. (2011). Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. *The Journal of Clinical Investigation*, 121(7):2750–2767.

Li, M. D., Cao, J., Wang, S., Wang, J., Sarkar, S., Vigorito, M., Ma, J. Z., and Chang, S. L. (2013). Transcriptome sequencing of gene expression in the brain of the hiv-1 transgenic rat. *PLoS One*, 8(3):e59582.

Lorenz, D. J., Gill, R. S., Mitra, R., and Datta, S. (2014). Using rna-seq data to detect differentially expressed genes. In *Statistical analysis of next generation sequencing data*, pages 25–49. Springer.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M., and Gilad, Y. (2008). Rna-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research*, 18(9):1509–1517.

McCarthy, D. J., Chen, Y., and Smyth, G. K. (2012). Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10):4288–4297.

McLachlan, G. (1997). On the em algorithm for overdispersed count data. *Statistical Methods in Medical Research*, 6:76–98.

Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8(May):1145–1164.

Pan, W., Shen, X., Jiang, A., and Hebbel, R. P. (2006). Semi-supervised learning via penalized mixture model with application to microarray sample classification. *Bioinformatics*, 22(19):2388–2395.

Parsons, D. W., Jones, S., Zhang, X., Lin, J. C.-H., Leary, R. J., Angenendt, P., Mankoo, P., Carter, H., Siu, I.-M., Gallia, G. L., Olivi, A., McLendon, R., Rasheed, B. A., Keir, S., Nikolskaya, T., Nikolsky, Y., Busam, D. A., Tekleab, H., Diaz, L. A., Hartigan, J., Smith, D. R., Strausberg, R. L., Marie, S. K. N., Shinjo, S. M. O., Yan, H., Riggins, G. J., Bigner, D. D., Karchin, R., Papadopoulos, N., Parmigiani, G., Vogelstein, B., Velculescu, V. E., and Kinzler, K. W. (2008). An integrated genomic analysis of human glioblastoma multiforme. *Science*, 321(5897):1807–1812.

Pertea, M. (2012). The human transcriptome: An unfinished story. *Genes*, 3(3).

Peters, M. J., Joehanes, R., Pilling, L. C., Schurmann, C., Conneely, K. N., Powell, J., Reinmaa, E., Sutphin, G. L., Zhernakova, A., Schramm, K., et al. (2015). The transcriptional landscape of age in human peripheral blood. *Nature Communications*, 6:8570.

Qu, Z. and Adeson, D. L. (2012). Evolutionary conservation and functional roles in ncrna. *frontiers in Genetics*, 3(205):391–420. PMID: 29727582.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.

Robinson, M. D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biology*, 11(3):R25.

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B., Giltnane, J. M., Hurt, E. M., Zhao, H., Averett, L., Yang, L., Wilson, W. H., Jaffe, E. S., Simon, R., Klausner, R. D., Powell, J., Duffey, P. L., Longo, D. L., Greiner, T. C., Weisenburger, D. D., Sanger, W. G., Dave, B. J., Lynch, J. C., Vose, J., Armitage, J. O., Montserrat, E., Lpez-Guillermo, A., Grogan,

T. M., Miller, T. P., LeBlanc, M., Ott, G., Kvaloy, S., Delabie, J., Holte, H., Krajci, P., Stokke, T., and Staudt, L. M. (2002). The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine*, 346(25):1937–1947. PMID: 12075054.

Schneider-Poetsch, T. and Yoshida, M. (2018). Along the central dogmacontrolling gene expression with small molecules. *Annual Review of Biochemistry*, 87(1):391–420. PMID: 29727582.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Si, Y., Liu, P., Li, P., and Brutnell, T. P. (2013). Model-based clustering for rna-seq data. *Bioinformatics*, 30(2):197–205.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Roy. Stat. Soc.*, 36:111–147.

Thalamuthu, A., Mukhopadhyay, I., Zheng, X., and Tseng, G. C. (2006). Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405–2412.

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

Tothill, R. W., Tinker, A. V., George, J., Brown, R., Fox, S. B., Lade, S., Johnson, D. S., Trivett, M. K., Etemadmoghadam, D., Locandro, B., Traficante, N., Fereday, S., Hung, J. A., Chiew, Y.-E., Haviv, I., Gertig, D., deFazio, A., and Bowtell, D. D. (2008). Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clinical Cancer Research*, 14(16):5198–5208.

Tseng, G. C. (2007). Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data. *Bioinformatics*, 23(17):2247–2255.

Wang, Z., Gerstein, M., and Snyder, M. (2009). Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57.

Wang, Z., Ma, S., Zappitelli, M., Parikh, C., Wang, C.-Y., and Devarajan, P. (2016). Penalized count data regression with application to hospital stay after pediatric cardiac surgery. *Statistical methods in medical research*, 25(6):2685–2703.

Witten, D., Tibshirani, R., Gu, S. G., Fire, A., and Lui, W.-O. (2010). Ultra-high throughput sequencing-based small rna discovery and discrete statistical biomarker analysis in a collection of cervical tumours and matched controls. *BMC Biology*, 8(1):58.

Witten, D. M. (2011). Classification and clustering of sequencing data using a poisson model. *Ann. Appl. Stat.*, 5(4):2493–2518.

Witten, D. M. and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726.

Zararsız, G., Goksuluk, D., Korkmaz, S., Eldem, V., Zararsiz, G. E., Duru, I. P., and Ozturk, A. (2017). A comprehensive simulation study on classification of rna-seq data. *PlOS one*, 12(8):e0182507.

Zhao, S., Fung-Leung, W.-P., Bittner, A., Ngo, K., and Liu, X. (2014). Comparison of rna-seq and microarray in transcriptome profiling of activated t cells. *PloS one*, 9(1):e78644.