Classification of RNA-Seq Data via Gaussian Copulas

Qingyang Zhang

Department of Mathematical Sciences, University of Arkansas, Fayetteville, AR 72701

Email: qz008@uark.edu

Abstract

RNA-sequencing (RNA-Seq) has become a preferred option to quantify gene expression, because it is more accurate and reliable than microarrays. In RNA-Seq experiments, the expression level of a gene is measured by the count of short reads that are mapped to the gene region. Although some normal-based statistical methods may also be applied to log-transformed read counts, they are not ideal for directly modeling RNA-Seq data. Two discrete distributions, Poisson distribution and negative binomial distribution, have been commonly used in the literature to model RNA-Seq data, where the latter is a natural extension of the former with allowance of overdispersion. Due to the technical difficulty in modeling correlated counts, most existing classifiers based on discrete distributions assume that genes are independent of each other. However, as we show in this paper, the independence assumption may cause non-ignorable bias in estimating the discriminant score, making the classification inaccurate. To this end, we drop the independence assumption and explicitly model the dependence between genes using Gaussian copula. We apply a Bayesian approach to estimate covariance matrix and the overdispersion parameter in negative binomial distribution. Both synthetic data and real data are used to demonstrate the advantages of our model.

Keywords: RNA-Seq, Negative binomial distribution, Gaussian copula, Sample classification, Correlated counts

1 Introduction

RNA-sequencing (RNA-Seq) is a revolutionary tool for the study of transcriptomes (Mardis [2008]; Wang et al. [2009]). Compared to hybridization-based microarrays, RNA-Seq eliminates the need for species-specific sequence information and provides more reliable measurements for gene expression (Marioni et al. [2008]). The huge number of reads produced by RNA-Seq experiment enables researchers to better detect novel transcripts and quantify the gene expression in ultra-high resolution. Essentially, RNA-Seq consists of three distinct phases: (1) RNA is isolated from tissue and segmented to an average length of 200 base pairs; (2) RNA segments are reverse transcribed to cDNAs; (3) The cDNAs are mapped to reference transcriptome or genome. An RNA-Seq experiment usually produces tens of millions of short reads between 25 and 300 base pairs in length. The number of reads mapped to each transcript provides a digital measure of transcript abundance.

Poisson distribution and negative binomial distribution are commonly used distributions to model RNA-Seq data. Based on these two distributions, numerous methods have been proposed to detect the differentially expressed genes, including but not limited to edgeR (Robinson & Smyth [2008]), DESeq2 (Love et al. [2014]), baySeq (Hardcastle & Kelly [2014]), BBSeq (Zhou et al. [2011]), and SAMseq (Li & Tibshirani [2013]). Despite the significant advances in differential expression analysis, the progress on classification of RNA-Seq data is relatively recent. Witten (Witten [2011]) developed a Poisson linear discriminant analysis (PLDA) by assuming that the data follow a Poisson distribution. However, in the presence of overdispersion, the Poisson assumption might not be appropriate. Dong et al. (Dong et al. [2016]) further extended the Poisson classifier to a negative binomial classifier (NBLDA), and explored how the dispersion affects the classifications. Other classifiers developed for RNA-Seq data include logistic regression model and partial least square method (Tan et al. [2014]). Due to the difficulty of modeling correlated counts, most existing classifiers assume that all genes are independent of each other. However, as pointed out by Dong et al. (Dong et al. [2016]), this assumption is very restrictive and may not be realistic in practice. The objective of the paper is to numerically assess the effect of independence assumption on classification of RNA-Seq data, and to develop a new classifier incorporating the dependence between genes using continuous latent variables and Gaussian copula. A Metropolis-Hasting algorithm in combination with Gibbs sampler (Lee

[2014], Liu & Daniels [2006]) is adopted to estimate the covariance matrix and overdispersion parameters in our model. Our new classifier explicitly models two important aspects of RNA-Seq data: overdispersion of read counts and correlation between genes, therefore provides accurate parameter estimate and sample classification.

Copula is an important tool in modeling the dependence between random variables of any type. It is especially useful for modeling multiple discrete variables whose joint distribution can be extremely complicated. To begin with, we provide a short review of copula function and Gaussian copula. Consider a vector of random variables $(X_1, X_2, ..., X_p)$, the copula function of $(X_1, X_2, ..., X_p)$, $C : [0,1]^p \to [0,1]$, is defined as the cumulative distribution function (cdf) of $(F(X_1), F(X_2), ..., F(X_p))$:

$$C(u_1, u_2, ..., u_p) = P(F(X_1) \leq u_1, F(X_2) \leq u_2, ..., F(X_p) \leq u_p).$$

By definition, a copula function is a multivariate distribution function where the marginal of each random variable is uniform. Sklar's Theorem guarantees that any multivariate distribution can be expressed with univariate marginals and a copula function which links the marginals. In practice, we can completely separate the choice of marginals and the choice of copula. Popular copulas include, but not limited to Gaussian copula, Student's t copula, Clayton's copula and Frank's copula (Nelson [1999]). Clayton's copula and Frank's copula both belong to the bivariate Archimedean copula family. For more than two dimensions, Gaussian copula is convenient to model the complex correlation structure (both positive and negative correlations). The Gaussian copula is based on multivariate normal distribution:

$$C(u_1, ..., u_p|\Omega) = \Phi_p(\Phi^{-1}(u_1), ..., \Phi^{-1}(u_p)|\Omega),$$

where Φ represents the cdf of standard normal distribution, $\Phi_p(...|\Omega)$ represents the cdf of p-dimension normal distribution with correlation matrix Ω. To connect discrete marginals and continuous copula function, we introduce a latent variable and treat the observed count as the discretized value of the continuous latent variable (in spirit, it is same as multivariate Probit model).

The remainder of this paper is structured as follows. In Section 2, we formally describe the statistical framework for classification of RNA-Seq data and introduce a Bayesian approach to estimate unknown

parameters. Numerical studies are conducted to compare different models and classifiers in Section 3. In Section 4, we apply the proposed method to two real data sets including the cervical cancer data and HapMap data. We discuss and conclude this paper in Sections 5 and 6.

2 Methods

In this section, we propose a new classifier for RNA-Seq data based on copula function. We assume that the data follow a complex multivariate distribution with negative binomial marginals. The correlation between genes can be described by a Gaussian copula. A general Bayesian framework for estimating parameters in each class is discussed.

2.1 Negative binomial distributions for marginal model

First, we consider only one class. Let $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^T$ be the $n \times p$ data matrix, where x_{ij} denotes the observed number of reads mapped to gene i in sample j, $i = 1, 2, ..., p$ and $j = 1, 2, ..., n$. We consider the following negative binomial distribution $F_{ij}, i = 1, 2, ..., p, j = 1, ..., n$ for marginals (Dong et al. [2016]):

$$X_{ij} \sim \text{NB}(\mu_{ij}, \delta_i),\ \mu_{ij} = s_j \lambda_i, \tag{1}$$

where $\mu_{ij} = E(X_{ij})$, s_j is the size factor to scale read counts for the jth sample due to different sequencing depth, λ_i is the total number of reads for gene i, and δ_i is the overdispersion parameter for gene i, i.e., $V(X_{ij}) = \mu_{ij} + \mu_{ij}^2 \delta_i$. The estimates of λ_i and s_j in (1) are straightforward:

$$\hat{\lambda}_i = \sum_{j=1}^{n} x_{ij},$$

$$\hat{s}_j = \frac{\sum_{i=1}^{p} x_{ij}}{\sum_{j=1}^{n} \sum_{i=1}^{p} x_{ij}}.$$

For overdispersion parameter δ_i, the moment estimate and shrinkage estimate (Yu et al. [2013]) are commonly used estimates. However, both methods suffer from instability, e.g., the moment method sometimes

gives a negative value. In this paper, we treat δ_i as unknown parameter, which is to be estimated jointly with other parameters in a Bayesian framework.

2.2 Gaussian copula for dependence between genes

In general, the analysis of correlated counts might be difficult because of the lack of suitable discrete multivariate distribution that can model complex correlation structures. To surmount this difficulty, we model the correlation via Gaussian copula, so that the correlation between read counts can be created through the correlation of the continuous latent variables. Let $\boldsymbol{Z}_j = (Z_{1j}, ..., Z_{pj})^T$ be the Gaussian latent variables (with unit variance) of $\boldsymbol{X}_j$, and $\boldsymbol{Z}_j \sim N_p(\boldsymbol{0}, \Omega)$, where Ω represents the covariance or correlation matrix. The observed counts $\boldsymbol{x}_j$ are the discretized values of $\boldsymbol{z}_j$ by quantile matching. The relation between X_{ij} and Z_{ij} can be interpreted as follows:

$$X_{ij} = x_{ij} \iff F_{ij}(x_{ij} - 1|\delta_i) < \Phi(Z_{ij}) \leq F_{ij}(x_{ij}|\delta_i), \tag{2}$$

where $F_{ij}()$ represents the cumulative distribution function of variable X_{ij}, x_{ij} takes nonnegative integer values $0, 1, 2, ...$ and $F_{ij}(-1|\delta_i) = 0$ by definition of cdf.

The Gaussian copula of latent variables has the following simple form:

$$C(\Phi(z_{1j}), ..., \Phi(z_{pj})|\Omega) = \Phi_p(z_{1j}, ..., z_{pj}|\Omega). \tag{3}$$

Based on (2) and (3), the likelihood function can be obtained immediately:

$$f(\boldsymbol{x}_j|\delta, \Omega) = \int_{R_{pj}} \cdots \int_{R_{1j}} \phi_p(\boldsymbol{z}_j|\Omega) d\boldsymbol{z}_j, \tag{4}$$

where $\delta = (\delta_1, ..., \delta_p)$, ϕ_p denotes the multivariate normal density of dimension p, with mean vector $\boldsymbol{0}$ and unit marginal variance. The endpoints defining integration region $R_{ij} = (L_{ij}, U_{ij}]$ are specified as a function of the parameter δ_i,

$$L_{ij} = \Phi^{-1}(F_{ij}(x_{ij} - 1|\delta_i)),$$

$$U_{ij} = \Phi^{-1}(F_{ij}(x_{ij}|\delta_i)).$$

2.3 Bayesian estimation of parameters

Finding the maximizer of (4) is impractical due to the complexity and non-convexity of the likelihood function. Here, we consider the Bayesian approach proposed by Lee (Lee [2014]; Liu & Daniels [2006]) for parameter estimate. The posterior distribution of the parameters and latent variables can be written as follow:

$$f(\delta, \Omega|\boldsymbol{x}) \propto f(\Omega)\prod_{i=1}^{p} f(\delta_i)\prod_{j=1}^{n}\int f(\boldsymbol{x}_j, \boldsymbol{z}_j|\delta, \Omega)d\boldsymbol{z}_j,$$

where the priors are specified as $f(\delta_i) = IG(\alpha_0, \beta_0)$, and $f(\Omega) = IW_p(\Psi_0, v_0)$. We use $IG(\alpha_0, \beta_0)$ to denote the inverse-gamma distribution with shape parameter α_0 and rate parameter β_0, and use $IW(\Psi_0, v_0)$ to denote the inverse-Wishart distribution with scale matrix Ψ_0 and degree of freedom v_0. The Gibbs sampling can be implemented based on the following conditional distributions:

$$f(z_{ij}|z_{-i,j}, \delta_i, \Omega),$$

$$f(\delta_i|z_{-i,\cdot}, \delta_{-i}, \Omega),$$

$$f(\Omega|\boldsymbol{z}, \delta),$$

where $z_{-i,j} = (z_{1j}, ..., z_{(i-1)j}, z_{(i+1)j}, ..., z_{pj})$, $z_{-i,\cdot} = \{z_{-i,j}, j = 1, ..., n\}$, $\delta_{-i} = (\delta_1, ..., \delta_{i-1}, \delta_{i+1}, ..., \delta_p)$, $\boldsymbol{z} = (\boldsymbol{z}_1, ..., \boldsymbol{z}_n)^T$. Conditioning on $\{z_{-i,j}, \delta_i, \Omega\}$, latent variable z_{ij} follows a truncated normal distribution, where the mean and variance depend on $\{z_{-i,j}, \Omega\}$, and the endpoints depend on $\{x_{ij}, \delta_i\}$. We use a routine Metropolis-Hasting algorithm to sample δ_i and use a parameter-expanded reparameterization and Metropolis-Hasting algorithm (PXMH, Lee [2014]; Liu & Daniels [2006]) to sample Ω. Details about these algorithms are provided in the Appendix.

2.4 Classification

We consider the classification problem when the RNA-Seq were conducted over multiple classes, i.e., $K \geq 2$. Let $y_j \in \{1,2,...,K\}$ denotes the class label of sample j, i.e., $y_j = k \iff j \in C_k$. The marginal distribution of X_{ij} in class k can be formulated in a similar way to (1):

$$X_{ij}|y_j = k \sim \text{NB}(\mu_{ij}d_{ik}, \delta_{ik}),\ \mu_{ij} = s_j\lambda_i, \tag{5}$$

where d_{ik} and δ_{ik} are gene- and class-specific parameters among the K classes. The overdispersion parameter δ_{ik} can be estimated in the PXMH algorithm and d_{ik} can be estimated by

$$\hat{d}_{ik} = \frac{\sum_{j\in C_k} x_{ij}}{\sum_{j\in C_k} \hat{s}_j\hat{\lambda}_i}.$$

Based on the trained models for all the K classes, the class label for new observation $\boldsymbol{x}^*$ can be predicted. By Bayes' rule:

$$P(y^* = k|\boldsymbol{x}^*) \propto \pi_k f_k(\boldsymbol{x}^*),$$

where f_k is the probability density function for class k. The prior probability π_k can be estimated by $\hat{\pi}_k = \frac{\sum_{j=1}^n I_{\{y_j=k\}}}{n}$. We assign the new observation $\boldsymbol{x}^*$ to class k that maximizes the following discriminant score (posterior probability):

$$\hat{P}(y^* = k|\boldsymbol{x}^*) = \frac{\hat{\pi}_k f_k(\boldsymbol{x}^*)}{\sum_{l=1}^{K} \hat{\pi}_l f_l(\boldsymbol{x}^*)}.$$

3 Simulation study

We conducted two simulation studies with $K = 2$ to benchmark our new classifier. In Simulation I, we evaluated the performance of the Bayesian approach in estimating the discriminant score $P(y^* = 1|\boldsymbol{x}^*)$ under different correlation settings. In Simulation II, we compared the performance of six classifiers under different settings of dispersion and correlation strength.

3.1 Simulation I

Given the correlation matrices $\Omega_k, k = 1, 2$, we generated data in two steps:

- Step 1: Simulate latent variables $\boldsymbol{z}_{jk} \sim N_p(\mathbf{0}, \Omega_k), j = 1, ..., n_k, k = 1, 2$, where Ω is the correlation matrix or covariance matrix, $n_1 = n_2 = 50$ and $p = 50$
- Step 2: Transform $\boldsymbol{z}_{jk}$ to $\boldsymbol{x}_{jk}, j = 1, ..., n$ using (2), where $\mu_{i1} = 20, \mu_{i2} = \mu_{i1} + \Delta_i, \Delta_i \sim Unif(-15, 15)$ and $\delta_{i1} = \delta_{i2} \sim Unif(1, 10)$ for $i = 1, ..., p$

We compared the copula-based model with Dong et al.'s independence model under two settings of autoregressive correlation structure $\Omega_1(i, j) = \Omega_2(i, j) = \exp(-a|i - j|)$: $(1)a = 0.5$; $(2)a = 1.5$. For each class, we trained the model using half of the samples (25 samples randomly chosen in each class) and then calculated the discriminant score $P(y^* = 1|\boldsymbol{x}^*)$ for each of the rest samples. For both models, we estimated μ_{ik} by $\hat{\mu}_{ik} = \sum_{j=1}^{n_k} x_{ijk}/n_k$. For independence model, we estimated the overdispersion parameter δ_{ik} using R package *sSeq*, which implements the shrinkage method by Yu et al. The score $P(y^* = 1|\boldsymbol{x}^*)$ was then estimated under independence assumption. For the copula-based model, we jointly estimated δ_{ik} and Ω_k using PXMH algorithm with the following priors: $\delta_{ik} \sim IG(0.5, 0.5)$ and $\Sigma \sim IW(5, I_{50})$. A chain of 15,000 iterations was generated and the last 10,000 samples were kept for calculating the posterior mean.

Figures 1 shows the estimation bias, i.e., $\hat{P}(y^* = 1|\boldsymbol{x}^*) - P(y^* = 1|\boldsymbol{x}^*)$, by two models under two settings. In both settings, the copula-based method shows its superiority over the independence model, and the improvement is more significant in the presence of stronger correlation. Since moderate and strong co-expression between genes were commonly seen in real data, the ignorance of such information may lead to lower prediction accuracy.

3.2 Simulation II

In the second simulation, we compared our copula-based classifier with the other five classifiers including Poisson linear discriminant analysis (PLDA), negative binomial linear discriminant analysis (NBLDA),

k-nearest neighbors (KNN), partial least square method (PLS) and logistic regression method. We implemented PLDA using R package *PoiClaClu* and implemented NBLDA using the R source code provided by the Dong et al. (https://github.com/yangchadam/NBLDA). In NBLDA, the overdispersion parameters were estimated by R package *sSeq*. For KNN, we chose parameter $K = 1,3,5$. To implement PLS, we used the function 'spls()' provided in R package *spls*. For our new classifier, same priors were used as in Simulation I. A chain of 15,000 iterations was generated and the last 10,000 samples are retained for estimation.

The data are generated under four different settings (for all settings, $\mu_{i1} = 20, \mu_{i2} = \mu_{i1} + \Delta_i, \Delta_i \sim Unif(-15,15)$):

- Setting 1 (weaker correlation, smaller dispersion): $\Omega_k(i,j) = \exp(-1.5|i-j|)$, $\delta_{ik} \sim Unif(1,5)$, $k = 1,2$
- Setting 2 (weaker correlation, larger dispersion): $\Omega_k(i,j) = \exp(-1.5|i-j|)$, $\delta_{ik} \sim Unif(5,20)$, $k = 1,2$
- Setting 3 (stronger correlation, smaller dispersion): $\Omega_k(i,j) = \exp(-0.5|i-j|)$, $\delta_{ik} \sim Unif(1,5)$, $k = 1,2$
- Setting 4 (stronger correlation, larger dispersion): $\Omega_k(i,j) = \exp(-0.5|i-j|)$, $\delta_{ik} \sim Unif(5,20)$, $k = 1,2$

The comparison results under different sample sizes ($n = 10,30,60,100$) are shown in Figures 2 and 3. Due to the independence assumption, other five classifiers including PLDA and NBLDA failed to model the dependence structure between genes, therefore the estimated probabilities were biased (see Simulation I). It is observed that the copula-based model performs consistently better than the other classifiers in terms of classification accuracy, especially in setting 4 with stronger correlation and larger dispersion. When the correlation between genes are very weak, our model has similar performance with Dong et al.'s independence model.

4 Real data analysis

In this experiment, we considered two real data sets including the cervical cancer data (Witten et al. [2010], available in Gene Expression Omnibus (GEO) with access number GSE20592) and the HapMap data (Montgomery et al. [2010]; Pickrell et al. [2010], available at ftp://ftp.ncbi.nlm.nih.gov/hapmap). The cervical cancer data contains 58 samples (29 tumor samples and 29 normal controls), and 714 microRNAs which were differentially expressed in cancer group and normal group. The HapMap data contains 129 samples (60 CEU samples and 69 YRI samples) and a total number of 52,580 genes. For both data sets, we removed genes with less than 10 reads across all samples. Three different classifiers including Poisson classifier, negative binomial classifier and the copula-based classifier were compared in terms of the misclassification rate. In our classifier, same priors were used as in the simulation studies.

We noted that the real data sets often contain large portion of irrelevant and redundant genes. A gene screening could greatly reduce the computing time and improve the classification accuracy. We conducted gene selections using R package *edgeR* (available in *Bioconductor*, www.bioconductor.org), as suggested by Dong et al. The algorithm implemented in *edgeR* is based on negative binomial model and takes overdispersion into account, therefore it is suitable for our problem. This method first estimates the overdispersion parameter for each gene by maximizing the combination of gene-specific conditional likelihood and the overall conditional likelihood, and then constructs an exact test using negative binomial distribution.

For Cervical cancer data, 40 samples were randomly assigned to the training set and the rest 18 samples were assigned to the testing set. A total of 20, 50, 100, 300 genes were selected, respectively. For HapMap data, the samples were randomly split into training set and testing set, with 70 samples and 59 samples, respectively. A total of 50, 100, 300, 500 genes were selected, respectively. Three different classifiers were then trained by the training data and applied to the testing data. The whole procedure were repeated for 100 times and the average misclassification rate were recorded.

The comparison results are shown in Figure 4. For both data sets, the copula-based model is more accurate than the other two classifiers. Figure 5 displays the distribution of correlation coefficients between every pair of genes in the cervical cancer data (log-transformed). It can be seen that the vast majority of the correlations are positive, and half of them are fairly strong (above 0.5), indicating that the independence

assumption in PLDA and NBLDA is violated.

5 Discussion

In this paper, we have proposed a new classifier for RNA-Seq data. Different from other classifiers, it incorporates the dependence between genes in the supervised classification problem. To the best of our knowledge, this is the first work that applies copula model to the classification of count data. Numerical comparisons show that our new model achieves better estimate of discriminant scores than existing methods, therefore results in more accurate sample classification. The improvement is more significant in the presence of stronger correlation between genes. In addition to RNA-Seq data, this classifier can be generally applied to other digital gene expression data to improve the classification accuracy.

The copula-based classifier introduced in this paper assumes that the reads count of each gene follows a negative binomial distribution. This assumption has been widely used in practice since the negative binomial distribution is flexible to model and quantify the overdispersion in RNA-seq data. However, the negative binomial assumption still might be violated in some real data sets. In the copula-based method, one could choose alternative marginal models (e.g., Poisson mixture model) which better fits the data, and the Bayesian estimation introduced in this paper still can be applied to estimate the covariance matrix since the estimation of copula only depends on the cumulative distribution functions of the marginals. It is also noteworthy that the current PXMH algorithm for parameter estimation is time-consuming when the number of selected genes is large. For example, in the analysis of HapMap data, a single run (out of 100 runs in total) with 500 selected genes takes about 2.5-3 minutes with C++ implementation. In the future study, we would like to explore other optimizations such as EM-type method for computational efficiency. For example, rather than joint estimating $\{\delta, \Omega\}$ in Bayesian framework, we may first estimate δ using stable method based on read counts, and then estimate Ω by EM update where latent variables $\boldsymbol{Z}$ can be treated as missing values. The EM-type method may greatly reduce the computing time. To model the correlation between genes, we use Gaussian copula as it is convenient for multivariate problem. Another possible future work is to compare different latent variables and copula functions, e.g., Student's t copula (Nelson [1999]) and Gaussian mixture copula (Zhang & Shi [2016]), in a model comparison framework (Mai & Zhang [2016], Zhang et al. [2014],

Matveeva et al. [2016]).

6 Conclusion

RNA-sequencing experiment quantify gene expression by the count of short reads mapped to the gene region. When biological replicates are available, the negative binomial distribution allowing overdispersion is better suited for modeling RNA-Seq data than Poisson distribution. Recently, Dong et al. (Dong et al. [2016]) developed a classifier based on negative binomial distribution, which outperforms previous methods including the Poisson classifier and K-nearest neighbors classifier. However, due to the difficulty of modeling the dependence in discrete data, most existing classifiers assume that all the genes are independent of each other. In this paper, we systematically investigate the effect of independence assumption on discriminant score calculation and classification. In addition, we developed a copula-based classifier for RNA-Seq data that incorporates the dependence structure between genes, while maintaining the negative binomial marginals. Our numerical comparisons and real data analysis demonstrate that the new classifier performs better than existing methods including Dong et al.'s negative binomial classifier.

Acknowledgement

Support has been provided in part by the Arkansas Biosciences Institute, the major research component of the Arkansas Tobacco Settlement Proceeds Act of 2000.

Competing Interests

The author has declared that no competing interests exist.

Abbreviations

PLDA: Poisson linear discriminant analysis; NBLDA: Negative binomial linear discriminant analysis; KNN: K-nearest neighbors; PXMH: Parameter-expanded reparameterization and Metropolis-Hasting

References

Dong, K, Zhao, H, Tong, T & Wan, X (2016), NBLDA: negative binomial linear discriminant analysis for RNA-seq data, *BMC Bioinformatics*, **17**, 1–7.

Hardcastle, TJ & Kelly, KA (2014), baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data, *BMC Bioinformatics*, **11**(422).

Mai, K & Zhang, Q (2016), Identification of biomarkers for predicting the overall survival of ovarian cancer patients: a sparse group LASSO approach, *International Journal of Statistics and Probability*, **5**(6).

Lee, EH (2010), Copula analysis of correlated counts, *Advances in Econometrics*, **34**, 325–48, Emerald Group Publishing Limited.

Li, J & Tibshirani, R (2013), Finding consistent patterns: A nonparametric approach for identifying differential expression in RNA-Seq data, *Statistical Methods in Medical Research*, **22**, 19–36.

Liu, X & Daniels, MJ (2006), A New Algorithm for Simulating a Correlation Matrix Based on Parameter Expansion and Reparameterization, *Journal of Computational and Graphical Statistics*, **15**, 897–914.

Love, MI, Huber, W & Anders, S (2014), Moderated estimation of fold change and dispersion for rna-seq data with deseq2, *Genome Biology*, **15**, 1–21.

Mardis, ER (2008), Next-generation DNA sequencing methods, *Annual Review of Genomics and Human Genetics*, **17**, 1–7.

Marioni, JC, Mason, CE, Mane, SM, Stephens, M & Wan, Y (2008), RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays, *Genome Research*, **18**, 1509–17.

Matveeva, E, Maiorano, J, Zhang, Q, Eteleeb, A, Converting, P, Chen, J, Infantino, V, Stamm, S, Rochka, E, Wang, JP, & Fondufe-Mittendorf, Y (2016), Involvement of PARP1 in the regulation of alternative splicing, *Cell Discovery*, **2**(15046)

Montgomery, SB, Sammeth, M, Gutierrez-Arcelus, M, Lach, RP, Ingle, C, Nisbett, J, Guigo, R & Dermitzakis, ET (2010), Transcriptome genetics using second generation sequencing in a Caucasian population, *Nature*, **464**, 773–7.

Nelson, RB (1999), *An Introduction to Copulas*, Springer, New York.

Pickrell, JK, Marioni, JC, Pai, AA, Degner, JF, Engelhardt, BE, Nkadori, E, Veyrieras, JB, Stephens, M & Gilad, Y (2010), Understanding mechanisms underlying human gene expression variation with RNA-sequencing, *Nature*, **464**, 768–72.

Robinson, MD & Smyth, GK (2008), Small-sample estimation of negative binomial dispersion with applications to SAGE data, *Biostatistics*, **9**, 321–32.

Tan, KM, Petersen, A & Witten, D (2014), Classification of RNA-seq data, *Statistical Analysis of Next Generation Sequencing Data*, 219–46, Springer, New York.

Wang, Z, Gerstein, M & Snyder, M (2009), RNA-Seq: a revolutionary tool for transcriptomics, *Nature Review of Genetics*, **10**, 57–63.

Witten, DM (2011), Classification and clustering of sequencing data using a Poisson model, *Annals of Applied Statistics*, **5**, 2493–518.

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

Yu, D, Huber, W & Vitek, O (2013), Shrinkage estimation of dispersion in Negative Binomial models for RNA-Seq experiments with small sample size, *Bioinformatics*, **8**, 1–18.

Zhang, Q & Shi, X (2016), A Mixture Copula Bayesian Network Model for Multimodal Genomic Data, *Cancer Informatics*, In Press.

Zhang, Q, Burdette, JE, & Wang, JP (2014), Integrative Network Analysis of TCGA data for Ovarian Cancer, *BMC Systems Biology*, **8**(1338), 1–18

Zhou, Y, Xia, K & Wright, FA (2011), A powerful and flexible approach to the analysis of RNA sequence count data, *Bioinformatics*, **27**, 72–78.

Appendix

Sampling z_{ij} conditioning on $\{z_{-i,j}, \delta_i, \Omega\}$

We sample z_{ij} from univariate normal distribution $\phi(z_{ij}|\mu_{ij}, \omega_{ij}^2)$ truncated between L_{ij} and U_{ij}:

$$\mu_{ij} = \Omega_{i,-i}\Omega_{-i,-i}^{-1} z_{-i,j},$$

$$\omega_{ij}^2 = \Omega_{ii} - \Omega_{i,-i}\Omega_{-i,-i}^{-1}\Omega_{-i,i},$$

$$L_{ij} = \Phi^{-1}(F_{ij}(x_{ij} - 1|\delta_i)),$$

$$U_{ij} = \Phi^{-1}(F_{ij}(x_{ij}|\delta_i)),$$

where $\Omega_{i,-i} = \{\Omega_{hl}, h = i, l \neq i\}$, $\Omega_{-i,i} = \{\Omega_{hl}, h \neq i, l = i\}$, $\Omega_{-i,-i} = \{\Omega_{hl}, h \neq i, l \neq i\}$ and $z_{-i,j} = (z_{1j}, ..., z_{(i-1)j}, z_{(i+1)j}, ..., z_{pj})$.

Sampling δ_i conditioning on $\{z_{-i,\cdot}, \delta_{-i}, \Omega\}$

We sample δ_i based on the following density function:

$$f(\delta_i|x_{i,\cdot}, \delta_{-i}, \Omega, z_{-i,\cdot}) \propto f(\delta_i|\delta_{-i}) \prod_{j=1}^{n} f(x_{ij}|z_{-i,j}, \delta_i, \Omega),$$

where $f(x_{ij}|z_{-i,j}, \delta_i, \Omega) = \int \phi(z_{ij}|\Omega, z_{-i,j}) f(x_{ij}|z_{ij}, \delta_i) dz_{ij} = \Phi(\frac{U_{ij}-\mu_{ij}}{\omega_{ij}}) - \Phi(\frac{L_{ij}-\mu_{ij}}{\omega_{ij}})$.

Suppose δ_i is the current value and δ_i^* is the generated value from the proposal distribution. The Metropolis-Hasting probability of moving from δ_i to δ_i^* is:

$$A(\delta_i \to \delta_i^*|x_{i,\cdot}, z_{-i,\cdot}, \Omega) = \min\left\{1, \frac{f(\delta_i^*|x_{i,\cdot}, \delta_{1:(i-1)}^*, \delta_{(i+1):p}, z_{i,\cdot}, \Omega)}{f(\delta_i|x_{i,\cdot}, \delta_{1:(i-1)}^*, \delta_{(i+1):p}, z_{i,\cdot}, \Omega)} \frac{f_T(\delta_i|\hat{\delta}_i, v)}{f_T(\delta_i^*|\hat{\delta}_i, v)}\right\},$$

where the proposal $f_T(\delta_i|\hat{\delta}_i, v)$ is a t distribution which dominates the normal tails, $\hat{\delta}_i$ and v represents location and degree of freedom, respectively.

Sampling Ω conditioning on $\{z, \delta\}$

Sampling of Ω can be problematic due to the constraint $\Omega_{ii} = 1, i = 1, ..., p$. Here we sample Ω using parameter-expanded reparameterization and Metropolis-Hasting algorithm (PXMH, Lee [2014]; Liu & Daniels [2006]). The conditional density can be written as follows:

$$f(\Omega|z, \delta) \propto f(\Omega) \prod_{j=1}^{n} f(\boldsymbol{x}_j|\delta, \Omega).$$

The PXMH algorithm first simulates a covariance matrix Σ and then transforms it to a correlation matrix Ω. For convenience, define $D = \text{diag}(\sqrt{\Sigma_{11}}, ..., \sqrt{\Sigma_{pp}})$, then $\Omega = D^{-1}\Sigma D^{-1}$. Since Ω has p fewer parameters than Σ, an additional constraint is imposed:

$$\sum_{j=1}^{n} \Sigma_{ii} z_{ij}^2 = n, i = 1, 2, ..., p.$$

The PXMH Algorithm consists of three steps:

- PX step: Sample $\Sigma \sim IW_p(v, \Psi)$, where $v = v_0 + n$ and $\Psi = [\Psi_0^{-1} + \sum_{j=1}^{n} D z_j z_j^T D]^{-1}$
- MH step: Move to the new value Σ^* with probability:

$$A(\Sigma \to \Sigma^*|z) = \min\left\{1, \frac{f(\Sigma^*|v_0, \Psi_0)}{f(\Sigma|v_0, \Psi_0)} \frac{\prod_{i=1}^{n} f(\boldsymbol{y}_j|\delta, \Sigma^*)}{\prod_{i=1}^{n} f(\boldsymbol{y}_j|\delta, \Sigma)} \frac{f(\Sigma|v, \Psi)}{f(\Sigma^*|v, \Psi)}\right\}$$

- Transform covariance matrix Σ to correlation matrix Ω, $\Omega^* = D^{*-1}\Sigma^* D^{*-1}$

Figures

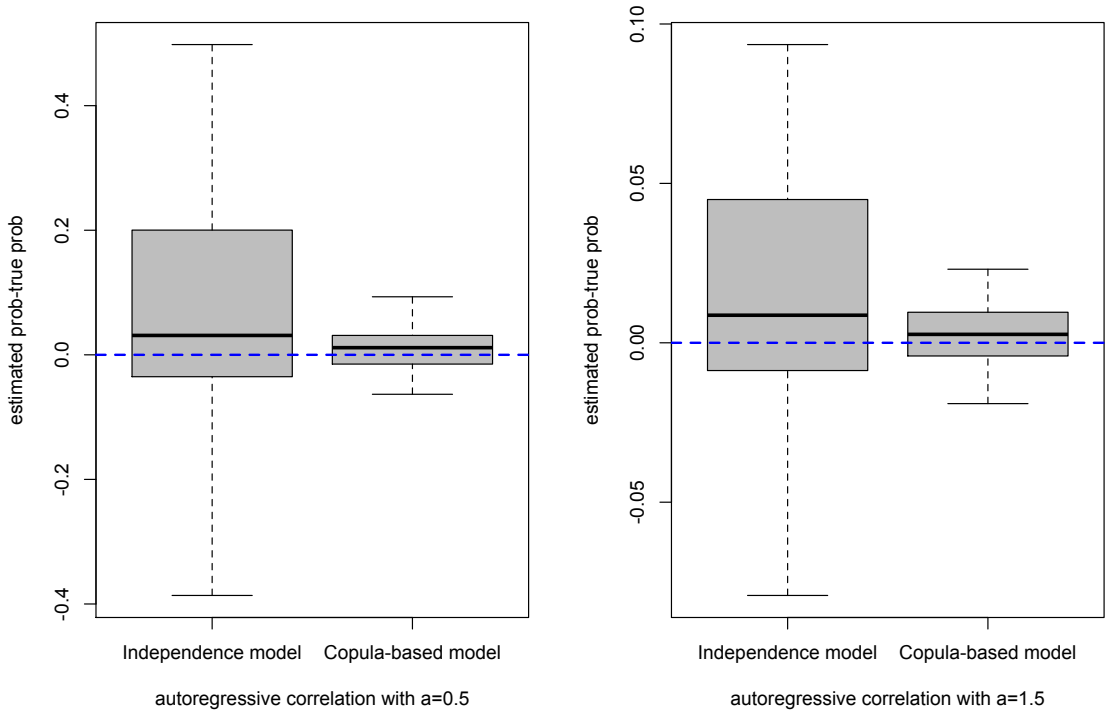

Figure 1: Comparison between Dong et al.'s independence model and the copula-based model. The y-axis represents the estimation bias in the discriminant score, i.e., the estimated score minus the true score.

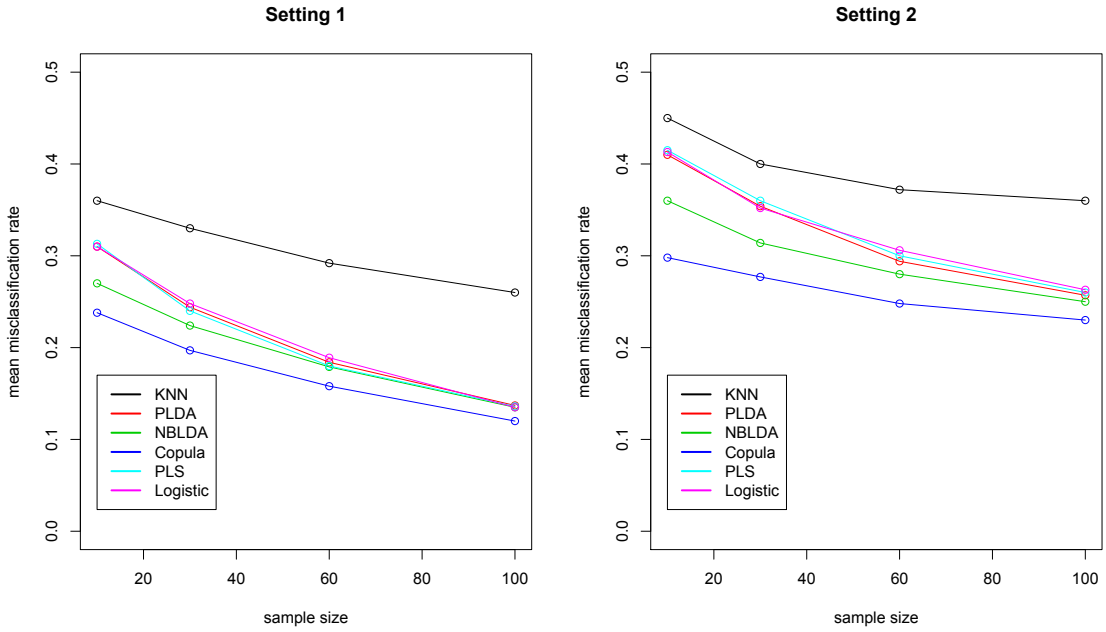

Figure 2: Comparison of six different classifiers under setting 1 (weaker correlation and smaller dispersion) and setting 2 (weaker correlation and larger dispersion). The y-axis is mean misclassification rate based on 100 independent simulation runs. The x-axis is sample size, $n = 10, 30, 60, 100$.

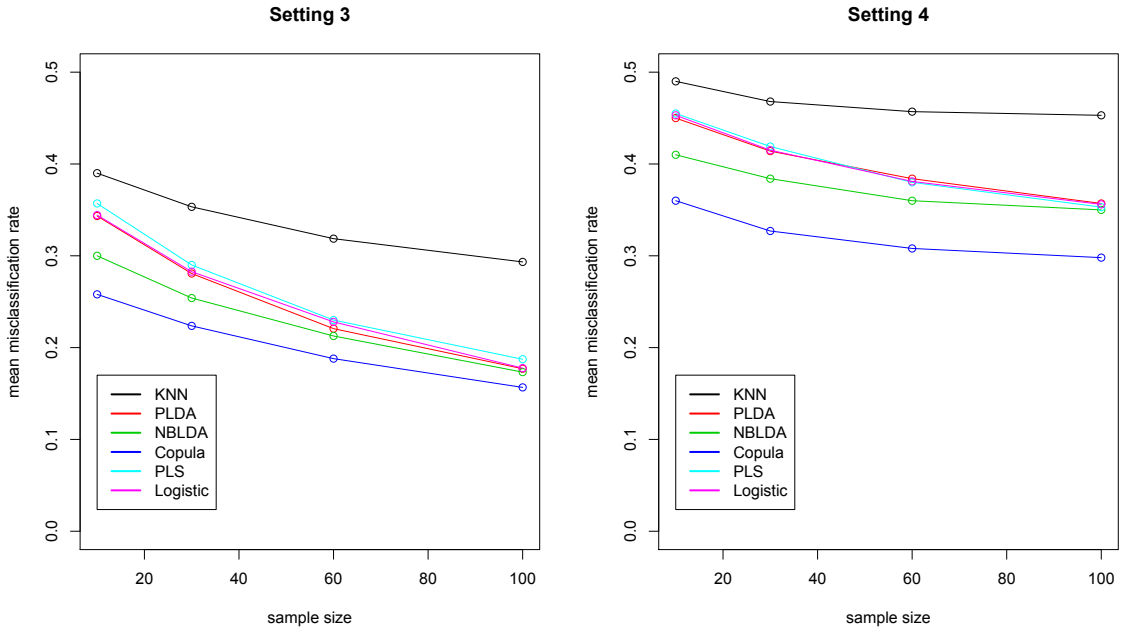

Figure 3: Comparison of six different classifiers under setting 3 (stronger correlation and smaller dispersion) and setting 4 (stronger correlation and larger dispersion). The y-axis is mean misclassification rate based on 100 independent simulation runs. The x-axis is sample size, $n = 10, 30, 60, 100$.

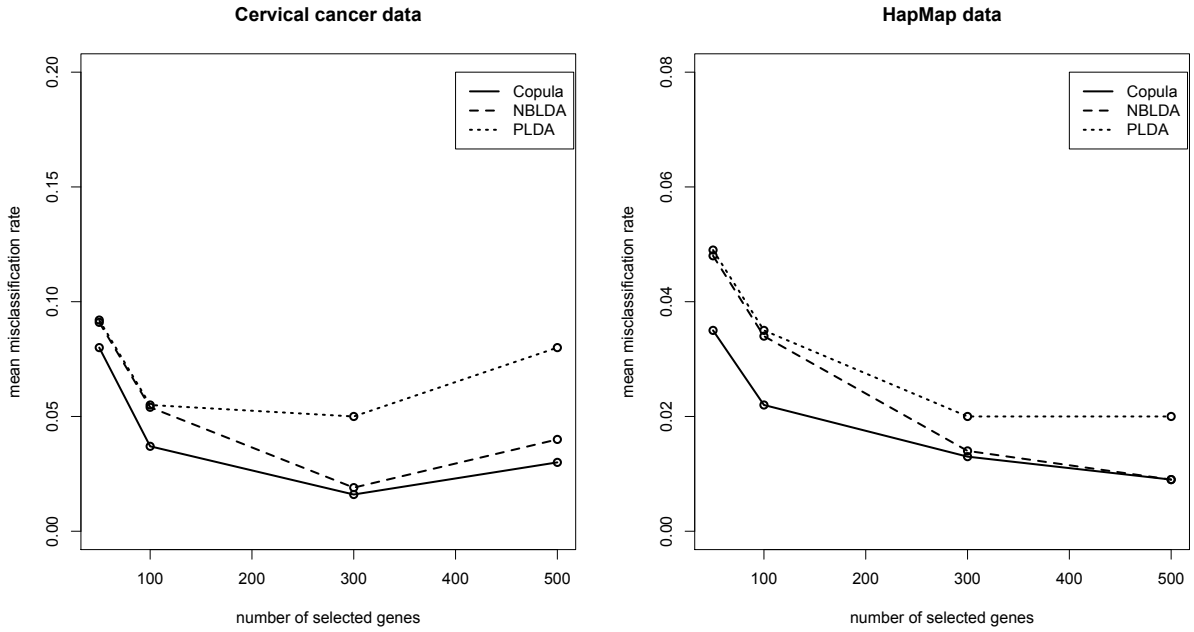

Figure 4: Comparison of three classifiers on two real data sets: cervical cancer data and HapMap data. The y-axis is mean misclassification rate based on 100 runs. The x-axis is number of selected genes, $p = 50, 100, 300, 500$

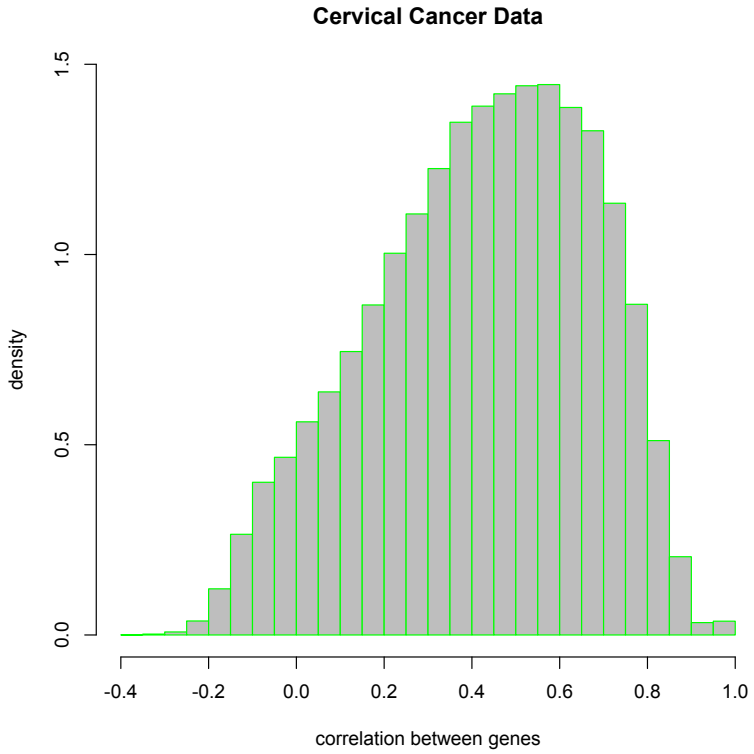

Figure 5: Distribution of correlations between every pair of genes in the cervical cancer data. The Pearson's correlation coefficients are based on the log-transformed data. About 93.49% of the gene pairs show positive correlation and only 6.51% of gene pairs show negative correlation.